Extreme Data Workshop 2018

Forschungszentrum Jülich, 18 – 19 September 2018

Proceedings

Martin Schultz, Dirk Pleiter, Peter Bauer (Editors)

IAS Series
Band / Volume 40
ISBN 978-3-95806-392-1

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Schriften des Forschungszentrums Jülich
IAS Series Band / Volume 40

Forschungszentrum Jülich GmbH
Institute for Advanced Simulation (IAS)
Jülich Supercomputing Centre (JSC)

Extreme Data Workshop 2018

Forschungszentrum Jülich, 18 – 19 September 2018
Proceedings

Martin Schultz, Dirk Pleiter, Peter Bauer (Editors)

Schriften des Forschungszentrums Jülich
IAS Series Band / Volume 40

ISSN 1868-8489 ISBN 978-3-95806-392-1

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über http://dnb.d-nb.de abrufbar.

Herausgeber und Vertrieb:	Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag 52425 Jülich Tel.: +49 2461 61-5368 Fax: +49 2461 61-6103 zb-publikation@fz-juelich.de www.fz-juelich.de/zb
Umschlaggestaltung:	Grafische Medien, Forschungszentrum Jülich GmbH
Druck:	Grafische Medien, Forschungszentrum Jülich GmbH
Copyright:	Forschungszentrum Jülich 2019

Schriften des Forschungszentrums Jülich
IAS Series, Band / Volume 40

ISSN 1868-8489
ISBN 978-3-95806-392-1

Persistent Identifier: urn:nbn:de:0001-2019032102

The complete volume is freely available on the Internet on the Jülicher Open Access Server (JuSER)
at www.fz-juelich.de/zb/openaccess

Table of Contents

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Extreme data: Demands, technologies, and services – A community workshop

Martin G. Schultz
Federated Systems and Data Division
Jülich Supercomputing Centre
Forschungszentrum Jülich
Jülich, Germany
m.schultz@fz-juelich.de

Dirk Pleiter
Jülich Supercomputing Centre
Forschungszentrum Jülich
Jülich, Germany
d.pleiter@fz-juelich.de

Peter Bauer
European Centre for Medium Range
Weather Forecast
Shinfield Park
Reading RG2 9AX, UK
peter.bauer@ecmwf.int

***Abstract*—We report on a two-day workshop in September 2018 which brought together scientists from different disciplines, computer engineers, and hardware vendors to discuss the challenges of data handling in the upcoming exascale era. Several aspects of extreme data management were brought up in the participants' presentations, and some common expectations and issues were identified in a plenary discussion. This paper summarizes the workshop presentations and discussions and reflects on likely and necessary developments in HPC infrastructures and services for meeting the domain sciences demands in the future. The paradigm change from compute centric to data centric HPC applications and the increasing complexity of HPC systems will also require much greater efforts in training next generation scientists and computer engineers.**

Keywords—exascale computing, large-scale data infrastructures, big data storage, fast data staging

I. Introduction

Science relies on complex numerical simulations on high performance computers (HPC) and is becoming increasingly data-driven. Advancing compute capabilities and capacities from current petascale systems to the exascale poses new challenges, because existing system architectures cannot simply be scaled up but instead require new concepts resulting in systems of higher complexity and different compute and storage layers. This in turn calls for a radical re-engineering of software and workflows. The staging of data, i.e. ensuring that the right data are at the right location at the right time and without undue delays, will therefore be a very important aspect of the design of future HPC systems.

Exascale computing will require exascale data handling. Various application areas (e.g. Earth system sciences, astrophysics, genomics, photon science) are facing rapidly growing challenges due to the fact that observing systems and advanced numerical models will produce data of unprecedented volumes and at unprecedented rates. For example, weather and climate models will run at much finer resolution, include more physical and chemical processes and as ensembles to also predict forecast uncertainty. Specialised observatories such as new telescopes, medical imagers, and satellites, but also other instruments in mobile phones and cars will produce data at finer resolutions, higher frequencies with increasing numbers. Scientific progress and the development of services based on modelling and observation (e.g. environmental prediction) hinge on the ability of the next-generation compute and data infrastructures to cope with these data streams without compromising service quality and delivery schedules. Present workflows often decouple data generation, data processing, downstream service data usage, and data archiving and curation. The anticipated dimension of extreme data will require the development of new hardware architectures, new workflows and services, and a better integration across various storage systems.

In recognition of these ongoing developments, a 2-day workshop was organized at the Jülich Supercomputing Centre in September 2018 to discuss extreme data demands, technologies, and services across different scientific disciplines. The workshop was attended by 25 participants from several European countries, who represented four natural science disciplines, computer scientists, and two representatives of major storage system vendors (Figure 1). Three sessions with 17 oral presentations aimed at identifying common demands across research fields and at developing prospects how future extreme data systems may unfold.

Fig. 1. Group photograph of the Extreme Data workshop participants (not all participants took part of the photo shooting)

The workshop focussed primarily on HPC-related extreme data generation and handling. However, topics such as data integration and sharing between HPC and cloud systems, and big data analytics concepts were also covered.

II. Workshop Programme

The three workshop sessions were grouped according to the three aspects, which the workshop aimed to address, i.e. demands, technologies, and services. Due to varying responses from the different communities who were invited to participate, the overall focus was somewhat skewed towards extreme data demands in Earth system science. Nevertheless,

as shown below, participation from other science disciplines and from technology and service providers was large enough to allow the identification of interdisciplinary challenges and potential solutions.

The following subsections provide brief summaries of the workshop presentations. The subsection titles refer to the session in which the talks were presented. However, there is considerable overlap between the three workshop topics and therefore the session titles are merely indicative. Thirteen workshop invitees also provided extended abstracts after the workshop. These are published in this report.

A. Extreme data demands

The opening presentation was given by Markus Reichstein (Max Planck Institute for Biogeochemistry, Jena, Germany) on "Challenges and perspectives of data-driven Earth system science". He presented examples of how machine learning is transforming the field of Earth system science and expects that this trend will continue.

Steve Aplin (DESY, Hamburg, Germany) reflected on "Data challenges in serial femtosecond crystallography", which represent one of the greatest data processing tasks in photon science. Several thousand frames are generated each second from the experimental facility, thus requiring supercomputing infrastructure that is usually dedicated to numerical simulations for analysing experimental data.

A perspective from the material science community was given by Giovanni Pizzi (EPFL Lausanne, Switzerland). His presentation was titled "Extreme-data demands in materials science: Dealing with high-throughput calculations towards the exascale". Unlike in other scientific disciplines, material simulations are usually relatively small but large in numbers. The total data volume reaches petabyte scale. Challenges are primarily related to management of the huge number of files and the efficient throughput for data sharing within the international community.

Dörte Handorf (Alfred Wegener Institute, Bremen, Germany) gave a presentation on "Current approaches and future challenges for analysing atmospheric circulation from climate model big data" [1]. Analyzing output from multi-decadal ensemble simulations of large community intercomparison projects and reanalysis datasets from numerical weather prediction centers requires careful planning of the data management. Due to increasing model resolution, retrieving, storing, and processing of such data from distributed sources will become more difficult in the future.

Jan Erik Sundermann (KIT, Karlsruhe, Germany) explained the "The challenge of the data demands of the high luminosity LHC experiments for the GridKa WLCG tier-1 center at KIT" [2]. The Large Hadron Collider at CERN produces more than 50 PBytes per year of data from four dedicated experiments. These data are managed in a distributed infrastructure consisting of about 170 sites in 42 countries. An essential element of this infrastructure is the integration of storage and compute services.

B. Extreme data technologies

Guido Juckeland (HZDR, Dresden, Germany) described real life experiences of generating and evaluating extreme data sets around the world [3]. He noticed in particular that modern experimental systems are now achieving data rates similar to advanced numerical simulations, so that it becomes impossible to store all raw data and some post-processing needs to take place in or near the instrument. They explore compression and in-situ visualisation techniques and advocate interactive access to simulations and data to help the selection of the data that shall be preserved.

Sadaf Alam (CSCS, Lugano, Switzerland) gave an overview of the hybrid cloud and HPC services for extreme data workflows [4]. She discussed the challenges of managing the enormous data amounts from the Swiss free electron laser and the particle accelerators and detectors at CERN, and weighed the pros and cons of providing extreme scale data services on shared HPC facilities. A hybrid HPC and cloud environment was proposed, which however requires further standardisation of interfaces so that community platform services can be built and maintained regardless of the underlying system architecture changes.

Stephan Kindermann (DKRZ, Hamburg, Germany) made a claim for dedicated large scale data infrastructures for the climate research community who already operate a global federated data infrastructure (Earth System Grid Federation, ESGF) [5]. Data handling challenges mainly arise from the geospatial and organizational separation of simulation centers and data centers conflicting with the user needs to co-locate data or perform complex data-intensive data analysis at several sites. He emphasized the importance of linking basic data management services (e.g. data identification, citation, and replication) with high performance data processing capabilities.

Tiago Quintino (ECMWF, Reading, UK) discussed the extreme data challenges on the HPC and cloud systems of a major operational weather prediction center [6]. Currently, more than 100 GByte/day of observational data from ca. 80 satellites and countless other sources and 130 TByte/day of model output need to be ingested, processed, managed, and stored at ECMWF. These data volumes are expected to grow by a factor of 100 by 2025. The weather centre responds to these challenges by rigorously controlling data workflows and developing object store facilities, which introduce flexibility to varying patterns of data access and allow configuration of hardware to minimize data latency and maximize throughput.

Bryan Lawrence (University of Reading, UK) observed that current workflows are often inadequate to solve large analysis tasks in climate science [7]. He pointed out that "user education, smarter compression, better use of tiered storage, and smarter workflows are all necessary – but far from sufficient." Dedicated HPC/cloud data analysis facilities together with smarter data storage software may levitate some problems of exascale data management, but without fundamental rethinking of the scientific analysis concepts there is a great risk of losing momentum in the discovery of fundamental mechanisms driving climate dynamics and the assessment of climate change and its impacts.

Dirk Pleiter (Forschungszentrum Jülich, Germany) provided an overview of technology roadmaps for future HPC infrastructures [8] and noted that both storage capacity and storage performance need to be increased. While HPC centers need to adopt more open policies with respect to data use from distributed sources, the different user communities need to develop better ways for estimating data storage and staging demands. His proposal is to make use of annotated use case

diagrams which characterize the data workloads at the different workflow stages.

At the end of the second session, Oliver Oberst (IBM, Zürich, Switzerland) and Jean-Thomas Acquaviva (DDN, Paris, France) presented some insights into new technological developments on the vendor side and emphasized the increasing need to monitor the performance of all storage tiers and use this information for the definition of scientific workflows.

C. *Extreme data services*

Peter Bauer (ECMWF, Reading, UK) complemented Tiago Quintino's description of ECMWF's data management strategy by explaining the upcoming developments of the weather prediction model and data assimilation system and the ensuing challenges for next-generation storage tiers [6]. He anticipated that next generation models will be run at scales that are fine enough that many of the current physical parameterisations can be replaced by the explicit simulation of physical laws, thereby alleviating a great part of current model biases. However, the necessary resolution increase will lead to enormously increased demands on the computational power and storage capacity of next generation HPC systems.

Jeannot Trampert (University Utrecht, The Netherlands) explained the "Data flow and assimilation in computational seismology". Although this application also requires large realtime data acquisition capability when it is employed operationally, the current challenges lie more on the computational side and the need to adapt codes to the more complex next generation HPC architectures.

Stefan Kollet (Forschungszentrum Jülich, Germany) presented the ongoing research at Jülich and its partner universities "Towards big data-enabled terrestrial systems modelling at HPSC TerrSys" [11]. He highlighted the need for large-scale and performant data services of researchers who may not always be directly involved in large community projects, but can contribute and wish to publish relevant data from their simulations. Future data solutions should therefore be developed with a view of being accessible to every scientist.

Jens Bröder (Forschungszentrum Jülich, Germany) talked about "the AiiDA framework for data generation and processing in materials science" [9] and made a point that there are various frameworks in the electron structure community, which are designed to manage large numbers of simulation jobs together with heterogeneous data. The community currently faces major challenges with regard to data sharing, because this requires petascale bandwidth solutions between distributed centers.

Ugur Cayoglu (Karlsruhe Institute for Technology, Germany): presented a "Flexible toolkit for climate data applications: Compression and tensor frameworks" [10], thus highlighting an example for an important technology for managing large simulation output, which is however only slowly adopted by the climate modelling community.

The final presentation was given by Kai Krajsek (Forschungszentrum Jülich, Germany) who described "The Helmholtz Analytics Toolkit (HeAT) - A scientific big data library for HPC" [12]. This initiative develops a software package that will reduce the complexity of scaling complex data analysis tasks on multi-node HPC systems. Through coding of a parallelized tensor and by offering a numpy-like interface to users, their applications can be ported almost effortlessly.

The proceedings volume contains an additional paper from Matthias Becker (University of Bonn, Germany), who unfortunately could not attend in person, on "Personalized medicine: the need for exascale data handling" [13]. Here, large genomic datasets and high resolution images need to be processed at high speed to enable computer-aided decision making in future hospitals. Special challenges in this field are the strict data privacy rules that must be securely implemented without jeopardizing the data handling efficiency.

III. Concluding Discussion

The participants of the Extreme Data workshop agreed that it was very productive to assemble scientists with different backgrounds and from different fields for exploring commonalities between domain-specific challenges of exascale data handling. A number of issues were identified which appear in more or less all scientific communities, and some common solutions could also be discussed.

Data throughput may be more problematic than storage capacity: from current experiences with handling large datasets it appears that bandwidth is more of a bottleneck than storage capacity, and this problem will likely exacerbate in the future. Bandwidth limitations occur at all levels of data storage hierarchies from storing simulation data during production to delivery of data to external users. As a rule of thumb various data centers observe that the amount of data moved across storage systems is about ten times the amount of data produced.

Future data storage will be hierarchical and use object stores: there is a clear recognition by computer engineers and vendors that the days of POSIX file systems for large scale data centers are counted and that the only viable option to achieve optimal performance and flexibility is the transition to hierarchical storage tiers and object store technologies. However, this implies a severe disruption of most scientific workflows operated today. Therefore, this transition needs to be accompanied by training and the development of tools, which should ideally make the underlying storage technology completely transparent to users. At least for some time, users will still want to retrieve for example a netCDF file instead of some objects when they request data. It needs to be investigated if such transformations can still be performed on the fly and efficiently.

Data services will be distributed: data production and data use are often spatially separated. The classical workflow of downloading data from data centers to local storage for analysis becomes infeasible. This implies that data centers must provide increased capacities for analysing data on their systems and adopt usage models which allow scientists to be involved. It is considered crucial to understand that data services are not only about technology, but also concern use policies, a business model and sustainability.

There is a chasm between large data and complex data: Extreme data is not only about huge data volumes, but must appreciate data complexity. While in some applications extreme data volumes originate from a relatively small amount of data sources, e.g. a numerical model, there are several use cases where these huge data sets are accompanied by many small and heterogeneous data sets, or where the

entire collection consists of many different data sets ("long-tail data"). Standardisation of the data model (for example expression of different types of weather and climate data as subsets or slices of a multi-dimensional data cube) can help managing the data complexity. However, current hardware and software stacks are either built for user-friendly and FAIR management of relatively small data amounts, or for the efficient handling of large, but more or less homogeneous data with limited support in terms of data documentation, data publication, data sharing, and user-driven complex analyses. The demand for better integration of HPC and cloud services is expected to have implications on the achievable performance increases in the future. A transition from the current file-based data organisation to object stores promises a more flexible management of different data needs and the more efficient merging of data from different sources. Performant data staging procedures could possibly be developed largely independent of the science application ("separation of concerns"). It remains to be seen to what extent storage system complexity can and should be hidden from the users.

Users demand more flexible access to data on HPC systems: In various disciplines ongoing initiatives explore the use of Jupyter notebooks or other collaborative tools for accessing data and running analyses on HPC systems. These tools greatly facilitate the usage of large data, but they require changes in the usage model of HPC installations and make it more difficult to achieve sufficient performance with full workloads on the machines. New technologies are needed to allow resource sharing and efficient data staging.

Large datasets should carry more metadata: traditionally, large (simulation) datasets are often poorly documented, or documentation is provided externally and not in machine-readable form. Enhanced metadata would allow users to make better decisions about which data they need [14]. Through provenance tracking it may become possible to identify simulation output that is rarely or never used so that the output strategies of models can be optimized. It can already be observed that moving simulation output around can be slower and more expensive than re-running a simulation. However, in practice these pathways for reproducing a simulation result are rarely at the choice of the user. Besides technical issues of integrating these two different workflows into a coherent framework they might involve different usage policies and accounting procedures, which might prevent access to either data or software for some users.

Extreme data requires collaboration and training: scientists are trained to do science, software engineers are trained to write good software. However, exascale computing and data handling urgently require people with different backgrounds working together so that data needs can be clearly expressed and data infrastructures can be designed to meet the requirements within and across the different science domains. More frequently than in the past it will be necessary to compromise on the data precision or completeness and adapt the analysis goals or analysis strategies in order to make scientific problems tractable. Collaboration between software engineers and scientists is required to find the best procedures and define the most efficient workflows. For example, it is often necessary to balance accuracy and performance, while traditionally scientists are concerned about accuracy and computer scientists worry about performance. There is an evident lack of people with adequate training to address such issues both in general, but even more so with respect to extreme data applications.

Machine learning will influence data patterns and production rates in the future: classical simulations and machine learning are brought together to replace parameterisations, expand the search space, or simply save computing resources when understanding the causality between inputs and outputs is not the most important requirement. This has important consequences for the design of future HPC systems. Many machine learning applications lack data locality and thus need to be able to read the data fast enough to cope with the speed of data processing. With GPUs becoming faster and faster also bandwidth requirements go up. Furthermore, training of neural networks typically involves repeated use of the same data, so that high-performance "cache" storage can be of great value. Some HPC centers are now employing SSD devices for this purpose. However, the ideal layout of hierarchical exascale storage systems still needs to be found and one must investigate whether capacity and throughput can be scaled from current systems or if new storage layouts will be needed for the exascale. In any case it is very likely that first generation exascale systems will be equipped with rather heterogeneous and complex data storage architectures. It will be important to educate users about these architectures and to work with the users on the development of performant workflows for their specific application on these architectures.

All in all the workshop participants observed a paradigm change occurring in almost all scientific disciplines with emphasis shifting from primarily computational problems to primarily data analysis problems. This has significant consequences for the planning of exascale HPC facilities in the future. Further discussions are needed to fully understand the implications of these changes and top prepare software and hardware technology implementation roadmaps that evolve at the same pace as the application requirements.

REFERENCES

[1] Handorf, D., Dethloff, K., Rinke, A., and Jaiser, R., Current approaches and future challenges for analysing atmospheric circulation from climate model big data, *this volume*, 2019.

[2] Petzold, A., and Sundermann, J. E., The Challenge of the Data Demands of the High Luminosity LHC Experiments for the GridKa WLCG Tier-1 Center at KIT, *this volume*, 2019.

[3] Juckeland, G., Huebl, A., and Bussmann, M., Is it here/there yet? Real Life Experiences of Generating/Evaluating Extreme Data Sets Around the World, *this volume*, 2019.

[4] Alam, S. R., Martinasso, M., Schulthess, T. C., Hybrid Cloud and HPC Services for Extreme Data Workflows, *this volume*, 2019.

[5] Kindermann, S., Stockhause, M., Thiemann, H., Towards exascale climate data handling: infrastructure, data management, data services, *this volume*, 2019.

[6] Quintino, P., Lean, P., Bauer, P., and Smart, S., Extreme data and computing in numerical weather prediction, *this volume*, 2019.

[7] Lawrence, B. N., Kunkel, J. M., Churchill, J., Massey, N., Kershaw, P., Pritchard, M., Beating data bottlenecks in weather and climate science, *this volume*, 2019.

[8] Pleiter, D., Future I/O Architectures and Infrastructures for Extreme-Scale Data Analytics, *this volume*, 2019.

[9] Bröder, J., Wortmann, D., Blügel, S., Using the AiiDA-FLEUR package for All-electron *Ab initio* Electronic Structure Data Generation and Processing in Materials Science, *this volume*, 2019.

[10] Cayooglu, U., Meyer, J., Kerzenmacher, T., Braesicke, P., Streit, A., Flexible Tool Development for Climate Data Applications: A Compression Framework, *this volume*, 2019.

[11] Görgen, K., Brdard, S., Furusho-Percot, C., Kulkarni, K. B., Naz, B., Vanderborght, J., Hendricks-Franssen, H.-J., Towards big data-enabled terrestrial systems modeling at HPSC TerrSys, *this volume*, 2019.

[12] Krajsek, K., Comito, C., Götz, M., Hagemeier, B., Knechtges, P., Siggel, M., The Helmholtz Analytics Toolkit (HeAT) - A Scientific Big Data Library for HPC, *this volume*, 2019.

[13] Becker, M., Schultze, H., and Schultze, J. L., Personalized medicine: the need for exascale data handling, *this volume*, 2019.

[14] see definition of FAIR digital objects in Final Report and Action Plan from the European Commission Expert Group on FAIR Data, Turning FAIR into reality, available at , last accessed 2019-02-19.

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Current approaches and future challenges for analysing atmospheric circulation from climate model big data

Dörthe Handorf
Research Department Potsdam
Alfred Wegener Institute, Helmholtz
Center for Polar and Marine Research
Potsdam, Germany
doerthe.handorf@awi.de

Klaus Dethloff
Research Department Potsdam
Alfred Wegener Institute, Helmholtz
Center for Polar and Marine Research
Potsdam, Germany
klaus.dethloff@awi.de

Annette Rinke
Research Department Potsdam
Alfred Wegener Institute, Helmholtz
Center for Polar and Marine Research
Potsdam, Germany
annette.rinke@awi.de

Ralf Jaiser
Research Depertment Potsdam
Alfred Wegener Institute, Helmholtz
Center for Polar and Marine Research
Potsdam, Germany
ralf.jaiser@awi.de

***Abstract*— A large part of low-frequency variability in the climate system on sub-seasonal to decadal timescales can be described in terms of so-called atmospheric teleconnection patterns. To provide reliable climate predictions and projections it is necessary to advance the understanding of past, recent and future changes in the spatial/temporal structure of atmospheric teleconnections and to assess the impact of internal climate dynamics versus external forcing. To tackle these questions we exploit large global, gridded data sets, either from different reanalysis data sets or from model simulations with state of the art climate models mostly performed in the framework of CMIP (Coupled model intercomparison project) initiatives. The current and next generation climate models will produce unprecedented volumes of data. We will identify specific questions to deal with the challenge of climate big data.**

Keywords—atmospheric teleconnections, climate models, analysis of big data

I. Introduction

The extra-tropical atmospheric flow is characterized by large-scale spatial patterns of correlated anomalies of the climate fields (e.g. pressure, temperature and precipitation) with time-varying amplitude and phase. These patterns are called atmospheric teleconnection patterns and represent a considerable portion of the low-frequency atmospheric variability on sub-seasonal to decadal time-scales.

Atmospheric teleconnections influence the long-term weather prediction. A prominent example is the North Atlantic Oscillation (NAO) representing the dominant teleconnection pattern for the North Atlantic-European region. It is mainly constituted by a seesaw between Iceland and the Azores [1] and most pronounced during winter. The NAO-index measures the phase and strength of the NAO and indicates the strength of the westerlies over the North Atlantic and Western Europe, which determines the winter climate in Europe. Fig 1 shows the mean sea-level pressure anomalies during the NAO in its positive phase which results in anomalous stronger north-south pressure gradient over the North-Atlantic. This leads to stronger westerly winds in general and stronger and more frequent storms across the Atlantic in particular. The positive NAO phase supports mild, stormy and wet winter conditions in northern and central Europe and eastern US, whereas northern Canada, Greenland and southern Europe experience cold and dry winter conditions.

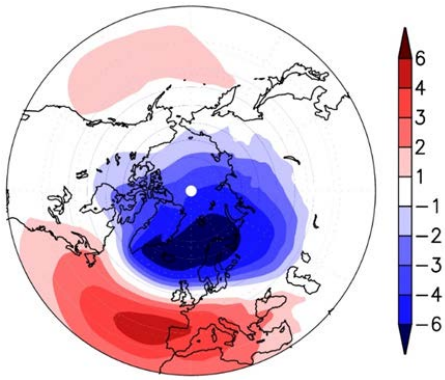


Fig. 1. The pattern of the North Atlantic Oscillation in terms of sea-level pressure anomalies (in hPa), showing the deviation from mean pressure distribution during the positive NAO phase.

II. Data and Methods

A. Data

This study uses data from the Coupled Model Intercomparison Project phase 3 (CMIP3) [2] and phase 5 (CMIP5) [3] multi-model data sets. We have used a suite of 23 CMIP3 and 46 CMIP5 models. We analysed coupled atmosphere-ocean simulations of the climate of the 20th century with observed anthropogenic and natural forcing from 1958 to 1999 (CMIP3 and CMIP5). For comparison, gridded data fields for the period 1958-1999 from three re-analysis data sets have been analysed. These datasets comprise the 40-yr re-analysis ERA-40 provided by the European Centre for Medium-Range Weather Forecast [4] and its precessor ERA-Interim [5] and the re-analysis of the National Center for Environmental Prediction/National Center for Atmospheric Research (NCEP/NCAR) [6].

B. *Methods*

Multivariate statistical methods for single and coupled pattern recognition have been applied. Atmospheric teleconnection patterns in fields of single climate variables have been determined by empirical orthogonal function (EOF) analysis [7]. EOF analysis is optimizing for orthogonal basis functions (patterns) determined by the dataset to account for as much variance as possible. Additionally, we applied a basis vector rotation to EOFs (varimax rotation; [8]) with the aim to get more localised patterns with a better separation of the EOF for each spatial point.

Coupled patterns between pairs of climate fields have been detected by the maximum covariance analysis (MCA; [9], [10]). The results of this analysis method are pairs of patterns coupled through a maximised covariance of their associated time series.

To quantify similarity between different atmospheric patterns Taylor diagrams [11] have been used. A Taylor diagram delivers a compact summary of pattern statistics in terms of pattern correlation, root-mean-square difference and ratio of variances.

III. Evaluation of Climate Models with Respect to Atmospheric Teleconnection Patterns

Exemplarily we evaluate the performance of CMIP3 and CMIP5 models in reproducing the spatial pattern of the atmospheric teleconnection pattern of the NAO for the boreal winter months December, January, February (DJF). The NAO patterns have been calculated by a rotated EOF analyses of the DJF monthly mean gridded geopotential height fields at 500 hPa from 20ºN to 90ºN.

Fig. 2 displays the performance of CMIP5 models in reproducing the NAO patterns by means of a Taylor plot.

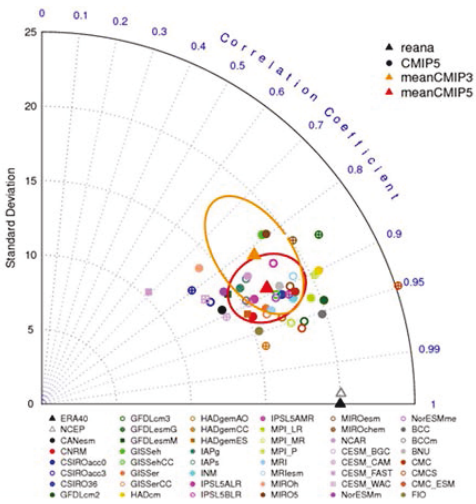


Fig. 2. Taylor plot to quantify similarity of the NAO pattern simulated by CMIP5 models and ERA40 and NCEP/NCAR reanalyses data. Ensemble mean and spread are indicated for the CMIP3 (orange) and CMIP5 ensembles.

First, it is striking that the CMIP5 models reproduce the NAO pattern well. Whereas 20 out of 23 CMIP3 models had pattern correlations larger than 0.7 for the period 1958-1999, for CMIP5 45 out of 46 CMIP5 models have pattern correlations larger than 0.7. The comparison of the CMIP3 and CMIP5 runs reveals improvement in reproducing the spatial structure of the NAO patterns. The spread in the skill in simulating the spatial NAO patterns across the CMIP5 ensemble has been decreased. Further investigations (e.g. [12]) have shown, that the skill in simulating the spatial teleconnection patterns is largely determined by the quality of the simulated zonal wind variability.

IV. Impact of Sea-Ice Changes on Teleconnection Patterns

Based on the results of the evaluation we developed the hypothesis that the zonal wind variability and hence the patterns and the state of teleconnections (NAO) are influenced by external forcing factors. Here we focus on the impact of changes in Arctic sea-ice on the atmospheric circulation.

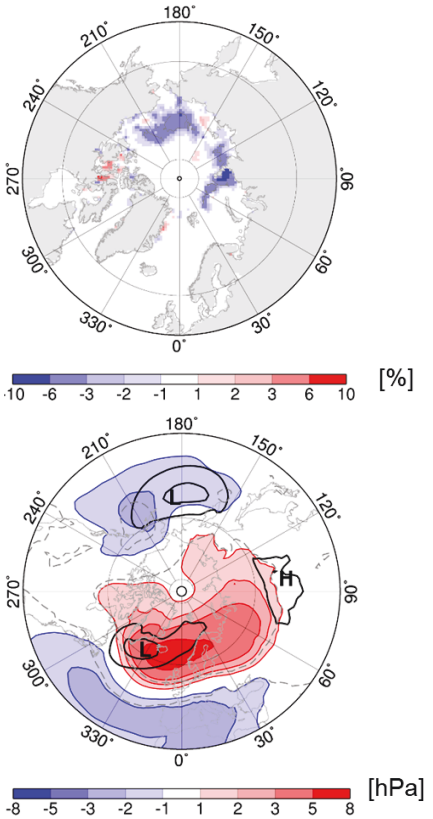


Fig. 3. Coupled patterns obtained by MCA of HadISST1 sea ice concentration (SIC) in September with ERA-Interim SLP in the following February from 1979 to 2015. The leading MCA patterns explain 39% of the squared covariance fraction. The SIC anomaly maps (top) are shown as heterogeneous regression map, the corresponding anomaly map for SLP (bottom) as homogeneous regression maps. Thin grey contours show the significance of the regressions at the 95% level. Black contours show the climatological mean (1980–2015) field of SLP. All data have been linearly detrended before calculating the MCA.

We start by searching for statistical relationships between large-scale patterns of September sea ice concentration and atmospheric circulation structures in the following February by applying a MCA. Fig. 3 displays the first pair of coupled MCA patterns of Arctic sea ice concentration in September and fields of sea level pressure (SLP) in February for the period 1979–2015. The coupled patterns describe diminishing sea ice over the northern edge of the Barents Sea, Kara Sea, Laptev Sea, Chukchi Sea, and Beaufort Sea co-varying with a pressure anomaly pattern resembling the negative phase of the NAO.

V. Study of Underlying Mechanisms by Exploiting New Specific Model Experiments

For these simulations, the atmospheric component of the GCM is constrained by realistic sea surface temperature (SST) and sea-ice distributions. Thus, the AMIP experiments allow us to focus on the atmospheric model without the added complexity and freedoms of ocean-atmosphere feedbacks.

In order to study the impact of Arctic sea ice reduction on the atmospheric circulation, in particular on the excitation of the NAO- pattern, respective sensitivity experiments with an atmospheric general circulation model (AGCM) have been carried out. The used model is the AGCM for Earth Simulator (AFES) version 4.1 with a spectral resolution of T79, 56 vertical levels, and a model top of about 60 km. Two perpetual model integrations labelled CNTL and NICE with 60 years each have been performed, where only the prescribed sea ice conditions in the Arctic are different [13]. High-ice conditions in the CNTL experiment are obtained from the observed 1979 to 1983 average of sea ice concentration, whereas low-ice conditions in the NICE experiment are obtained from the 2005 to 2009 period. Sea surface temperature (SST) data are kept constant to its 1979 to 1983 mean value in both model runs.

This design of the model experiment allows for a dedicated analysis of the impact of Arctic sea ice concentration anomalies on the atmosphere. The studies of [13] and [14] have shown that in winter the negative phase of the NAO appears more frequently following low Arctic sea ice conditions in ERA-Interim and AFES. As a possible mechanism, [13], [15], and [16] suggested a stratospheric pathway, in which vertically propagating planetary waves in early winter interact with the stratospheric polar vortex and weaken it. This leads to positive temperature and negative zonal wind anomalies in the vortex. These anomalous signals propagate downward into the troposphere and favor a negative phase of the NAO in February and March.

The good agreement in Arctic regions between the model experiment and reanalysis in terms of vertical planetary wave propagation in the troposphere and stratosphere provides strong evidence that the more frequent occurrence of a negative phase of the NAO in winter can be associated with changes in Arctic sea ice via a stratospheric pathway.

VI. Future Challenges

Here we presented a three-step hypothesis-driven approach to get improved understanding, how Arctic sea-ice retreat can impact atmospheric teleconnections, in particular the dominant teleconnection pattern for the North Atlantic-European region, the NAO.

To apply such hypothesis-driven research based on scientific theory in the field of climate research also in future, challenges will arise due to the availability of climate big data. The current and next generation climate models will produce unprecedented volumes of data mainly due to higher spatial resolution and the performance of ensemble simulations (to allow for robust signal detection against the background of strong atmospheric internal variability). It is expected that our current technological and data-analytical approaches will probably be not applicable in future. To adequately deal with the challenges of climate big data the following specific problems have to be tackled: (i) the analytical bottleneck in scientific data analysis has to be reduced, (ii) new approaches for the visualization of the results from large ensembles of model simulations have to be developed, (iii) big data analytics has to be integrated into hypothesis-driven climate research.

Acknowledgment *(Heading 5)*

We acknowledge the modelling groups, the Program for Climate Model Diagnosis and Intercomparison (PCMDI) and the WCRP?s Working Group on Coupled Modelling (WGCM) for their roles in making available the WCRP CMIP3 multi-model data set. Support of this dataset is provided by the Office of Science, US Department of Energy. We would like to thank NOAA, Earth System Research Laboratory, Physical Sciences Division, Boulder for providing the NCEP Reanalysis derived data and the European Centre for Medium Range Weather Forecast for providing the ERA40 Reanalysis data. The ERA interim data were obtained from the ECMWF web site (http://data-portal.ecmwf.int/). Merged Hadley-NOAA/OI SST and SIC data were obtained from the Climate Data Guide provided by the National Center for Atmospheric Research and University Corporation for Atmospheric Research (https://climatedataguide.ucar.edu/). The authors acknowledge the support by the project QUARCCS "QUAntifying Rapid Climate Change in the Arctic: regional feedbackS and large-scale impacts" funded by the German Federal Ministry for Education and Research (BMBF) under grant agreement 03F0777A and by the Helmholtz Climate Initiative REKLIM.

References

[1] Hurrell, J. W., "Influence of variations in extratropical wintertime teleconnections on Northern Hemisphere temperature", Geophys. Res. Lett., vol. 23, pp. 665-668, 1996

[2] Meehl, G., C. Covey, T. Delworth, M. Latif, B. McAvaney, and co-authors, "The WCRP CMIP3 multi-model dataset: a new era in climate change research", Bull. Amer. Meteor. Soc., vol. 88, pp. 1383-1394, 2007.

[3] Taylor, K. E., R. J. Stouffer, and G. A. Meehl, "An overview of CMIP5 and the experiment design", Bull. Amer. Meteor. Soc., vol. 93, pp. , 485-498, 2012.

[4] Uppala, S. M., P. W. Kaellberg, A. J. Simmons, U. Andrae, V. Da Costa Bechtold, and co-authors, "The ERA-40 re-analysis", Quart. J. Roy. Meteor. Soc., vol. 131, pp. 2961-3012, 2005.

[5] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Holm, L. Isaksen, P. Kallberg, M. Koehler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thepaut, F. Vitart, "The ERA-Interim reanalysis: configuration and performance of the

data assimilation system", Q. J. Roy. Meteor. Soc., vol. 137, pp. 553-597, 2011.

[6] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, and co-authors, "The NCEP/NCAR 40-year reanalysis project", Bull. Amer. Meteor. Soc., vol. 77, pp. 437-470, 1996.

[7] R. Preisendorfer, Principal Component Analysis in Meteorology and Oceanography (Developments in Atmospheric Science, Vol. 17). Elsevier, Amsterdam, p. 425, 1988

[8] M. B. Richman, "Rotation of principal components", J. Climatol., vol. 6, pp. 293-335, 1986.

[9] H. von Storch, and F. W. Zwiers, Statistical Analysis in Climate Research, Cambridge University Press, Cambridge, UK, p. 484, 1999.

[10] C. S. Bretherton, C. Smith, and J. M. Wallace, J. M. "An intercomparison of methods for finding coupled patterns in climate data", J. Climate, vol. 5, pp. 541-560, 1992.

[11] K. E. Taylor, K. E. "Summarizing multiple aspects of model performance in a single diagram", J. Geophys. Res., vol. 106, pp. 7183-7192, 2001.

[12] D. Handorf, and K. Dethloff, "How well do state-of-the-art atmosphere-ocean general circulation models reproduce atmospheric teleconnection patterns?", Tellus A, vol. 64, pp. 19777, 2012.

[13] T. Nakamura, K. Yamazaki, K. Iwamoto, M. Honda, Y. Miyoshi, Y. Ogawa, and J. Ukita, "A negative phase shift of the winter AO/NAO due to the recent Arctic sea-ice reduction in late autumn", J. Geophys. Res., vol. 120, pp. 3209-3227, 2015.

[14] B. Crasemann, D. Handorf, R. Jaiser, K. Dethloff, T. Nakamura, J. Ukita, and K. Yamazaki, "Can preferred atmospheric circulation patterns over the North-Atlantic-Eurasian region be associated with arctic sea ice loss?", Polar Science, vol. 14, pp. 9-20, 2017.

[15] T. Nakamura, K. Yamazaki, K. Iwamoto, M. Honda, Y. Miyoshi, Y. Ogawa, Y. Tomikawa, J. Ukita, "The stratospheric pathway for Arctic impacts on mid-latitude climate", Geophys. Res. Lett., vol. 43, pp. 3494-3501, 2016.

[16] R. Jaiser, T. Nakamura, D. Handorf, K. Dethloff, J. Ukita, and K. Yamazaki, "Atmospheric winter response to Arctic sea ice changes in reanalysis data and model simulations", J. Geophys. Res., vol. 121, pp. 7564-7577, 2016.

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

The challenge of the data demands of the high luminosity LHC experiments for the GridKa WLCG Tier-1 center at KIT

Andreas Petzold
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Karlsruhe, Germany
andreas.petzold@kit.edu

Jan Erik Sundermann
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Karlsruhe, Germany
jan.sundermann@kit.edu

***Abstract*—In this article we discuss the WLCG computing model, the role of GridKa as a WLCG Tier-1 center, and plans for the envisaged evolution of WLCG and GridKa towards the future HL-LHC computing.**

I. Introduction

The Large Hadron Collider (LHC) at CERN is one of the largest machines ever built. It enables physicists to study the basic constituents forming matter and their interactions at highest energies. The LHC consists of two large rings with a circumference of $27\,\mathrm{km}$ located in an underground tunnel near Geneva, $100\,\mathrm{m}$ below the border of Switzerland and France. The rings consist of a number of structures to accelerate two beams of protons or heavy ions to highest energies and more than 1200 super-conducting dipole magnets to bend the beams. Along the accelerator ring the beams are colliding with a center-of-mass energy of $13\,\mathrm{TeV}$ at four locations corresponding to the positions of the four LHC experiments ATLAS, CMS, ALICE and LHCb. About 1 billion protons collide every second inside each of the particle detectors. These detectors measure and record detailed information of particles and their decay products produced in the proton collisions. After filtering the most interesting collisions, events are recorded in each experiment with a rate of $200\,\mathrm{Hz}$ to permanent storage. This corresponds to data rates between $800\,\mathrm{MB/s}$ and $10\,\mathrm{GB/s}$ per experiment or $25\,\mathrm{GB/s}$ for all four experiments in total. Since 2009 the LHC experiments at CERN have been producing data volumes of more than $50\,\mathrm{PB}$ per year.

II. The LHC Computing Model

Today LHC data is being processed and analyzed in a relatively uniform distributed computing infrastructure. The Worldwide LHC Computing Grid (WLCG) is a global collaboration of computing centers composed of more than 170 sites in 42 countries. The WLCG integrates compute and storage resources of those centers to store, distribute and analyze the LHC data. The data is distributed by the experiments to the participating sites and jobs are usually run where the data is located. In 2017 WLCG sites provided more than $350\,\mathrm{PB}$ of online (or disk) storage and peak computing capacities of more than 750000 CPU cores. In addition, the experiments are using $450\,\mathrm{PB}$ of offline storage on magnetic tape. Compute and storage resources are typically connected with dedicated Ethernet links with speeds ranging from $10\,\mathrm{Gbit/s}$ up to $100\,\mathrm{Gbit/s}$ to the other WLCG sites.

The WLCG is following a tiered approach. Computing sites are organized in four different layers. Sites of each layer provide a specific set of services to the LHC experiments. The Tier-0, distributed between CERN and the Wigner Research Center for Physics in Budapest, is responsible for the initial data recording, the first pass reconstruction and the distribution of raw and reconstructed data to the Tier-1 sites. The WLCG has 13 large Tier-1 data centers providing storage (disk and tape) and compute resources for data reprocessing, simulation production, data analysis as well as the distribution of data to the smaller Tier-2 sites. Tier-2 sites are typically smaller university centers providing resources for data analysis and simulation production. There are currently 160 Tier-2 sites worldwide. Tier-3 sites provide local resources to the community without formal engagement in WLCG. From 2021 onwards, GridKa will serve as raw data center for the BELLE-II experiment.

III. The Grid Computing Centre Karlsruhe

The Grid Computing Centre Karlsruhe (GridKa) is a data and computing center for particle and astroparticle physics experiments. It is operated by the Steinbuch Centre for Computing (SCC) at the Karlsruhe Institute of Technology (KIT) in Germany. It was founded in 2002 initially supporting the four particle physics experiments BaBar, D0, CDF and COMPASS. Since 2006 GridKa is providing resources for the Pierre Auger Observatory. In 2008 GridKa started its full production service with 24/7 coverage before the anticipated start of LHC supporting all four LHC experiments as the German Tier-1 centre. At the moment GridKa is responsible for about 14% of raw LHC data. It is the largest of the 13 Tier-1 centers in terms of CPU and storage resources pledged to the LHC experiments.

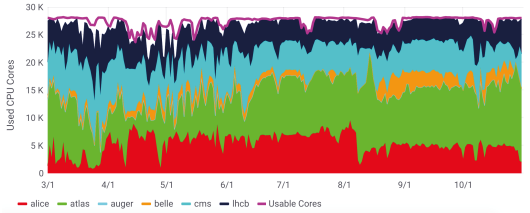


Fig. 1: Average utilization of the GridKa compute farm per day and per virtual organization between March 2018 and October 2018. The used cores per VO are stacked on top of each other. They are compared to the total number of available cores (magenta line). Fluctuations in the total number of available cores appear due to maintenance periods in which security updates are applied to the worker node installations.

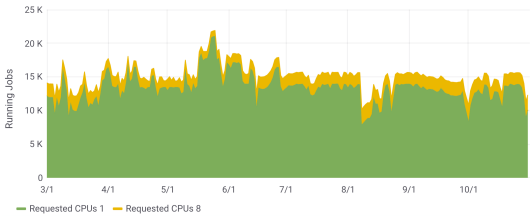
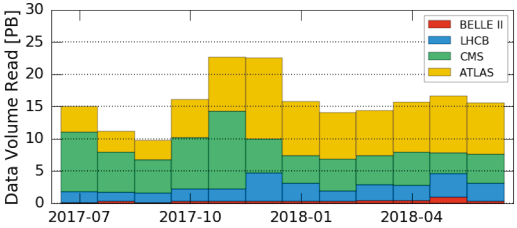


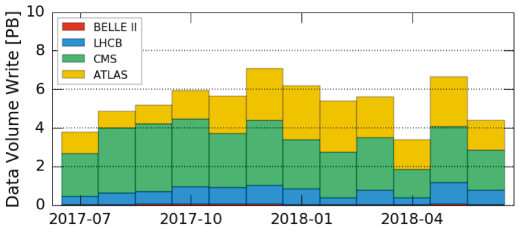
Fig. 2: Number of running jobs per number of requested CPU cores between March 2018 and October 2018. Jobs are typically single core or multi-core on the same CPU but do not require fast network interconnects between the worker nodes.

GridKa consists of a high-throughput compute farm, large installations of disk and tape (offline) storage as well as a dedicated network infrastructure. The compute farm consists of 1000 worker nodes with 18000 cores in a classical high-throughput setup. GridKa uses cost-efficient and reliable hardware in terms of power consumption, rack occupancy, and network infrastructure usage. The typical analysis and simulation production workload requires one disk spindle per 10 jobs and $10\,\mathrm{Gbit/s}$ Ethernet connections from the worker node server. The GridKa farm has 29000 job slots which were utilized on average by 98% (see figure 1). Jobs are typically single core or multi-core on the same CPU (see figure 2) but do not require fast network interconnects between the worker nodes. During the past 12 months, 24 Million jobs were running in GridKa corresponding to 176 Million CPU hours.

In 2016 a new disk storage system was put into operation in GridKa. The new system is a GxFS storage appliance from NEC. The design of the system allows for a flexible scaling of the storage infrastructure both in size and performance. The system uses IBM Spectrum Scale as software defined storage layer. The storage is partitioned into few very large file systems which enable the operator to manage the storage efficiently



(a) Data volume read



(b) Data volume write

Fig. 3: Data volume read (a) and written (b) to the GridKa online storage system per month and per experiment in 2017.

and make it possible to optimize for different scenarios, e.g. different experiments or workloads such as tape buffers. The online storage system currently has a usable capacity of $34\,\mathrm{PB}$ with a combined maximum read-write performance measured to be $100\,\mathrm{GB/s}$. The storage system is connected redundantly with 8 $100\,\mathrm{Gbit/s}$ Ethernet lines to the GridKa network backbone and subsequently to the high-throughput compute farm. In 2017 $7.5\,\mathrm{PB}$ and $4.2\,\mathrm{PB}$ of data were read on average per month from the GridKa compute farm and from remote sites, respectively (see figure 3a). In the same time period $1.1\,\mathrm{PB}$ and $3.0\,\mathrm{PB}$ of data were written on average per month from the GridKa compute farm and from remote sites, respectively (see figure 3b).

The tape storage system of GridKa is actively used by the experiments as distributed backup of LHC data. Data is frequently recalled from tape for the reprocessing of the raw data. Tape operations are initiated from the GridKa disk storage system. At the moment $49\,\mathrm{PB}$ of experiment data is stored on tape in two libraries.

In order to serve as a data distribution hub in WLCG, GridKa is connected with dedicated $100\,\mathrm{Gbit/s}$ network connections to CERN, the German research network and private high-energy physics networks. Upgrades of those networks to $200\,\mathrm{Gbit/s}$ will be done once required and financially viable.

In the upcoming years, resource increases of 20% per year are envisaged. The available storage capacity of the disk storage system is expected to grow till 2021 to $50\,\mathrm{PB}$. The design of the storage infrastructure already proved to be transparently expandable both in size and performance.

IV. Towards the High-Luminosity LHC

The data taking rates are expected to increase only moderately with the current experimental setup till the end of 2023 when the LHC accelerator and detectors will be shut down for two years to allow for a fundamental upgrade of the equipment with the objective to increase the data rates by a factor of up to 10 corresponding to a factor of up to 30 of data required to be collected, stored and distributed. The so called High-Luminosity Large Hadron Collider (HL-LHC) is expected to get in operation in 2025 with a dramatically increased discovery potential posing new unique challenges also for software, data analytics and computing.

Considering the expected technology and price evolution, the shortfall between requirements of the experiments and bare technology gains is significant: in the case of the ATLAS experiment a factor 4 in CPU and a factor 7 in online storage in 2027 can be expected. Also the event complexity and size will increase dramatically. With HL-LHC conditions more than 200 interactions are expected to happen simultaneously posing new challenges for the reconstruction of such complex event structures. HEP software is traditionally, until today, mostly single-threaded. Since single core CPU performance has been stalling already for several years, HEP software needs to evolve to utilize many cores or even specialized hardware like GPUs. Both, experiments and resource providers, considerably need to improve software and efficiency of resource usage to cover this gap in order to minimize the additional funding requirements [1], [2].

Especially online storage will be the relevant cost factor to consider when designing a new computing model for the HL-LHC era. This means in particular that HL-LHC computing will make extensive use of faster network connections between few very large and storage heavy sites and compute providers. The whole system will behave like a single large data lake. Data will either be accessed directly via the network, or will be pre-placed in dedicated caches to enable the use of inhomogeneous computing resources comprising existing WLCG sites, HPC centers, commercial or private clouds and other opportunistic resources. Users will access the "HEP cloud" remotely with a transparent view on the data. In addition, new software enabled to use specialized hardware, for instance GPUs, will improve the efficiency of compute resource usage by the experiments.

V. R&D Activities

In the following some selected examples for research and development activities undertaken in the proximity of GridKa or elsewhere towards the HL-LHC will be discussed.

A. Opportunistic Tier 1 for a Day

The on-demand usage of additional opportunistic compute resources, i.e. resources that are available only temporarily, does allow to satisfy short term compute requirement of users. The software package *Responsive On-Demand Cloud Enabled Deployment* (ROCED) [3]–[6] developed at the Institute of Experimental Particle Physics (ETP) at the KIT is able to monitor compute demands of an existing batch system and to dynamically provision additional cloud compute resources to fulfill those demands. Resources are transparently added and removed by starting and stopping virtual machines on supported resource providers. ROCED is suitable for CPU intense work flows. It was tested intensively by dynamically extending the Tier-3 cluster of the KIT CMS group with resources provided by the remote HPC centre at the University of Freiburg. During testing the system has demonstrated the ability to provision and manage resources on the same scale as those provided by the GridKa Tier-1 center. Further research on the dynamic usage of opportunistic resources and caching strategies to improve and speed up analysis work flows in the context of HEP and GridKa is performed in a number of different other projects [7]–[10].

B. Helix Nebula Science Cloud

One possible future provider of compute resources could be commercial cloud providers. The Helix Nebula Science Cloud (HNSciCloud) project is a pre-commercial procurement tender co-funded by the H2020 Programme (2016-18) for the establishment of a European hybrid cloud platform to support the deployment of high-performance computing and big-data capabilities for scientific research [11]. In the currently running second phase, the pilot phase, the HNSciCloud public-private partnership project has 10 procurers (CERN, CNRS, DESY, EMBL-EBI, ESRF, IFAE, INFN, KIT, STFC, SURFSara) and two contractor consortia (T-Systems/Huawei/Cyfronet/Divia and RHEA Group/Exoscale/SixSq). The HNSciCloud projects aims to develop IaaS level cloud services for scientific work loads and provide solutions in the areas of compute and storage, network connectivity, and service payment models.

C. Machine Learning

Different Machine learning techniques like neural networks or boosted decision trees have been used in HEP analyses already for a while. Main fields of applications have been primarily the identification of particles or the separation of signal and background events. Towards the HL-HLC additional use cases might prove value to increase the efficiency of the then limited compute and storage resources [12]. Research and development activities focus on topics like improved fast detector simulations using e.g. generative models, real-time analytics and triggering, object or track reconstruction, anomaly detection in detector operations or the reduction of the data footprint (event size).

VI. Summary

Over the previous decades computing models have been proven to be very successful for the global HEP community, collaborations and experiments. Nevertheless, new challenges with the upcoming HL-LHC require significant improvements wrt. software and computing models. HL-LHC will not be alone facing these new challenges as with the start of the project other communities like the upcoming FAIR or SKA will have to solve similar problems toward a successful future computing model.

References

[1] I. Bird et al., Update of the Computing Models of the WLCG and the LHC Experiments (2014), CERN-LHCC-2014-014.

[2] Antonio Augusto Alves et al., A Roadmap for HEP Software and Computing R&D for the 2020s arXiv:1712.06982v3, HSF-CWP-2017-001, FERMILAB-PUB-17-607-CD

[3] M. Schnepf et al., Mastering Opportunistic Computing Resources for HEP, to be published (2017).

[4] T. Hauth et al., On-demand provisioning of HEP Compute Resources on Clouds Sites and Shared HPC Centers, Journal of Physics 898, 5 (2017)

[5] T. Hauth et al., Dynamic provisioning of a HEP computing infrastructure on a shared hybrid HPC system, Journal of Physics 762, 1 (2016)

[6] T. Hauth et al., Dynamic provisioning of local and remote compute resources with OpenStack, Journal of Physics 664, 2 (2015)

[7] COBalD: http://cobald.readthedocs.io/en/latest/

[8] NaviX: https://gitlab.ekp.kit.edu/ETP-XRootD/NaviX

[9] ROCED: https://github.com/roced-scheduler/ROCED

[10] HTDA: https://bitbucket.org/kitcmscomputing/hpda/src/master/

[11] Helix Nebula Science Cloud: https://www.hnscicloud.eu/

[12] K. Albertsson et al., Machine Learning in High Energy Physics Community White Paper, J. Phys. Conf. Ser. 1085 (2018) no.2, 022008.

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Is it here/there yet? Real life experiences of generating/evaluating extreme data sets around the world

Guido Juckeland, Axel Huebl, Michael Bussmann
Helmholtz-Zentrum Dresden Rossendorf (HZDR)
Bautzner Landstr. 400, 01328 Dresden, Germany; Email: {g.juckeland,a.huebl,m.bussmann}@hzdr.de

Introduction

Large scale simulations have always been able to produce data at enormous rates to record the simulation progress. The current state of practice is to select a tiny subset of the simulation data prior to running the simulation and have only this subset written to disk for post-mortem analysis, since the I/O bandwidth per node even in the most advanced supercomputers today is limited to about the speed of a USB 2.0 thumbdrive. The same practice has long been established with the output of scientific experiments as well, albeit, in this case the subset of data to store is much larger, in most cases even the complete set. Most interestingly, the latest detectors and sensors have improved to a point where they can flood the file system in a similar fashion as simulations. As a result, storing all raw data is just not possible for continuous data streams, but at the same time lessons learned from dealing with the I/O of large simulations can also be applied to these experiment data.

This extended abstract first provides a detailed look at the actual numbers behind this data reduction challenge. Next it presents solutions that the authors successfully employed to (partially) solve these problems.

Putting the Challenges into Numbers

The background of our extreme data sets stems from particle-in-cell simulations run at various HPC sites around the world. The code has proven to scale to to the largest supercomputers available while maintaining an unprecedented performance [1]. The application is used to explore and/or verify phenomena that are observed in experiments as well as to prepare experiment setups. As such it is very difficult to determine the data structures to be stored for offline analysis beforehand. In this regard the challenges faced by PIConGPU are similar to those of novel, highly-dynamic high-bandwidth experiments enabled by the latest generation of detectors.

As a result both our simulations and upcoming experiments face multiple data challenges: First, the raw data from the simulation/experiment needs to be transferred into main memory. PIConGPU produces on Titan per GPU about 60 GByte/s while the PCIe bandwidth into main memory is limited to about 6 GByte/s. Even individual detectors in today's experiments or individual high-rate cameras in self-driving cars can produce raw data at more than 30 GByte/s with a similar transfer bandwidth limitation into main memory. Classical in-transit evaluation or visualization methods are already facing an order-of-magnitude data loss in this first step.

TABLE I
Comparison of three HPC Sites wrt. Storage Capacity and Speed

	Site A	Site B	Site C
Capacity (PByte)	250	6	3
Capacity per Node (TByte)	50	1	25
Capacity per FLOP (Byte/FLOP/s)	2	0.3	10
Bandwidth (GB/s)	2500	100 (estd.)	40
Bandwith per Node (MB/s)	42	1.7	33
Bandwidth per FLOP (μByte/FLOP)	20	5	133
Retention Time (days)	90 + archive	30	∞

Second, the traditional offline analysis of simulation or experiment data needs to transfer the data to a permanent storage location. The actual bandwidth available per node even for the fastest parallel file systems today is below that of a local disk, as shown in Table I. As such there is another data loss of multiple orders of magnitude for this storage process. To put these numbers into perspective, no leadership class HPC site (Site A and Site B) can even store the output of one single floating-point operation of all its compute units. Hence, disks can only be involved with already reduced primary data.

An even more dramatic problem with file I/O is the achievable performance depending on the file format. The numbers presented in Table I actually reflect peak performance using raw binary I/O with mutliple I/O streams. When using standardized file formats, such as the widely used HDF5, the actual performance is much lower and does not even scale to the whole HPC system as shown in Figure 1. HDF5 output stalls slightly about 10 GByte/s bandwidth, storing one snapshot of 15 TByte in this case takes 25 minutes. The ADIOS[1] library is able to overcome these limitations and to provide an overall performance increase for scaling up to the whole of Titan with an I/O bandwidth of about 50% of peak bandwidth.

The rather limited retention times at some compute facilities make it necessary to move the primary data off site for later evaluation. A typical PIConGPU primary data set is of 1.5 PBytes in size, thus, creating a race against the retention clock.

[1] https://www.olcf.ornl.gov/center-projects/adios/

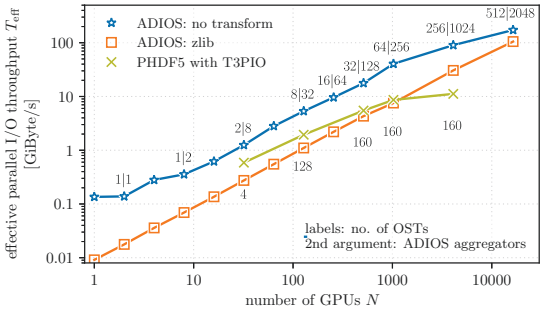


Fig. 1. PIConGPU I/O weak scaling on Titan from 1 to 16384 K20x GPUs (nodes). Zlib was only supported serially with compression mode fast. MPI Info hints for parallel HDF5 set via T3PIO (v2.3). For ADIOS, labels denote number of OSTs—aggregators, resulting for N 32 in a striping of each aggregated process group over four OSTs. Lustre filesystem limits enforced 160 OSTs for (single-file) parallel HDF5 writes. [2]

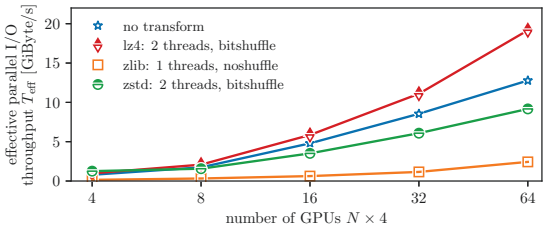


Fig. 2. Weak scaling of PIConGPU with implemented I/O methods on Hypnos from 4 to 64 K20m GPUs (16 nodes). In contrast to Titan and Summit nodes, on Hypnos only two physical CPU cores are available per GPU, resulting in I/O performance with zlib and zstd [3] below the untransformed output. [2]

Transferring such a large data set at line speed to a site with "only" a 10Gbit/s internet connection still requires over 17 days of transfer time at line speed. As a result, one is forced to start the download of the data set almost immediately after its generation. The need for transfers might not be relevant for every use case. However, when using simulations to verify experiment results or when using multiple supercomputing sites, it becomes necessary to transfer all data sets to one site for comparative analysis, thus, raising the transfer challenges again.

Potential Solutions

With all these challenges making an efficient usage of the raw data difficult, what options are available to lessen the implications of those limitations. It turns out that this is not a general solution. The compression/decompression of the data also takes time—usually a higher compression takes longer to compute. As a result, the achievable benefit is highly dependent on the choosen compression scheme and in some cases even the number of CPU threads used for compression (as shown in Figure 2).

Every I/O problem can be treated as a data reduction problem with a specific data reduction ratio and correspondig data throughput [2]. Viable alternatives to plain compression are in-situ visualization/analysis with very high reduction (e.g. from 3D data sets to 1D graphs or rendered images). ISAAC [4] enables direct in-memory visualization without any data transfers. The visualization kernels use the same data structures as the simulation directly on the GPU, also using the GPUs rendering capabilities for remote visualization. Combined with a mechanism to control the actual simulation (pause, rewind, restart), a user can explore and modify the simulation parameters interactively and, thus, acually find the 0.1-1% of data they want to store for later analysis. The actual finding process might even involve rewriting simulation/evaluation kernels. This will in the future be more convienient by deploying e.g. JIT CUDA cling to replace whole analysis kernels on the fly. [5].

The data transfer challenge between multiple sites has led to a rediscovery of well established grid computing tools. GridFTP and as a larger framework Globus Online[2] provide the best option for transferring large data sets asynchronously and without human supervision.

Summary and Outlook

Explorative simulations and experiments using novel high bandwidth data sources present a number of large challenges for traditional data analysis workflows. Offline analysis of raw data is simple impossible and even highly reduced sets of primary data can easily exceed one PByte in size. Only very few HPC sites are ready to enable scientists the evaluation of such large data sets over a longer period of time. As such, the sizing of I/O capabilities of a tier 0 and 1 site needs to be enpar with the actual compute capabilities. Interactive access to even large scale simulations helps mitigate the data reduction problem and enables a selection of the "right" data. Sufficient retention times and off-site transfer capabilities complete the list of requirements towards a modern supercomputing center.

References

[1] M. Bussmann, H. Burau, T. E. Cowan, A. Debus, A. Huebl, G. Juckeland, T. Kluge, W. E. Nagel, R. Pausch, F. Schmitt, U. Schramm, J. Schuchart, and R. Widera, "Radiative signatures of the relativistic kelvin-helmholtz instability," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. New York, NY, USA: ACM, 2013, pp. 5:1–5:12. [Online]. Available: http://doi.acm.org/10.1145/2503210.2504564

[2] A. Huebl, R. Widera, F. Schmitt, A. Matthes, N. Podhorszki, J. Y. Choi, S. Klasky, and M. Bussmann, "On the scalability of data reduction techniques in current and upcoming hpc systems from an application perspective," in *High Performance Computing*, J. M. Kunkel, R. Yokota, M. Taufer, and J. Shalf, Eds. Cham: Springer International Publishing, 2017, pp. 15–29.

[3] Y. Collet, P. Skibinski, N. Terrell, and S. Purcell, "Contributors: Zstandard (zstd) 1.1.4 - fast real-time compression algorithm," https://github.com/facebook/zstd, March 2017.

[4] A. Matthes, A. Huebl, R. Widera, S. Grottel, S. Gumhold, and M. Bussmann, "In situ, steerable, hardware-independent and data-structure agnostic visualization with ISAAC," *Supercomputing Frontiers and Innovations*, vol. 3, no. 4, pp. 30–48, 2016. [Online]. Available: {http://dx.doi.org/10.14529/jsfi160403}

[5] A. Huebl, S. Ehrig, and M. Bussmann, "Data Analysis and Simulations in Exascale Computing: Quo vadis?" ROOT User's Workshop, 10.-13.09.2018, Sarajevo, Bosnien und Herzegowina, 2018. [Online]. Available: {https://doi.org/10.5281/zenodo.1412537}

[2] https://www.globus.org/data-transfer

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Hybrid cloud and HPC services for extreme data workflows

Sadaf R Alam
Swiss National Supercomputing Centre
Lugano, Switzerland
alam@cscs.ch

Maxime Martinasso
Swiss National Supercomputing Centre
Lugano, Switzerland
martinasso@cscs.ch

Thomas C Schulthess
Swiss National Supercomputing Centre
Zürich, Switzerland
schulthess@cscs.ch

***Abstract*— Large scale experimental facilities such as the Swiss Light Source and the free-electron X-ray laser SwissFEL at the Paul Scherrer Institute, and the particle accelerators and detectors at CERN are experiencing unprecedented data generation growth rates. Consequently, management, processing and storage requirements of data are increasing rapidly. The Swiss National Supercomputing Centre, CSCS, provides computing and storage capabilities, specifically related to a dedicated archiving system for scientific data, for the Paul Scherrer Institute. Moreover, CSCS operates for the Swiss Institute of Particle Physics the Swiss portion of the Worldwide LHC Computing Grid, which has recently been provisioned on a shared supercomputing environment (Piz Daint) for the first time. While successful and cost-effective, porting of custom middleware in a shared environment is not a sustainable and scalable solution for diverse communities with their unique requirements for data management and computing services. We present and discuss a hybrid service provisioning approach by leveraging cloud and HPC services for addressing data and workflow management challenges at scale. One of the goals is to develop standard interfaces for the developers of community platform services for accessing HPC and cloud infrastructure in a transparent manner.**

Keywords—Supercomputing, cloud technologies, containers, storage, authentication and authorization.

I. Introduction

Due to ongoing and future progress in accelerator and detector technologies, large-scale experimental facilities such as the Swiss Light Source (SLS) and the free-electron X-ray laser SwissFEL at the Paul Scherrer Institute (PSI) [1], and the particle accelerators and detectors at CERN, the Large Hadron Collider (LHC), have been projecting unprecedented increases in performance. These developments are leading to rapid growth of data generated during experiments that will be difficult, if not impossible to manage with traditional IT infrastructures. For instance, until recently PSI operated alongside its large-scale experimental research facilities mostly workstation and HPC cluster-based computing and storage systems that were attached to individual experiments. Increasing data rates of new experiments are expected to substantially outgrow these IT systems. Hence, innovative solutions are needed to store and to process data that are beyond the capabilities and capacities of these individual large-scale experimental IT infrastructures.

An obvious solution strategy is to systematically couple the IT infrastructure of these large experiments with scalable compute and data systems of large academic data centers such as the Swiss National Supercomputing Centre (CSCS). For more than 25 years, CSCS developed and operates cutting-edge high-performance computing (HPC) systems as an essential service facility for Swiss and international researchers and contractual partners such as PSI. These computing and storage systems are used by scientists for a diverse range of purposes – from high-resolution simulations to the analysis of complex data. CSCS supercomputing and storage services are versatile to accommodate diverse needs of scientific simulation workflows while leverage economy of scale of high-end computing and storage environments. Examples of services provided by CSCS include the traditional HPC user program with allocations via a transparent peer review process of the Swiss (Tier 1) User Lab and the Tier 0 allocation process of the Partnership for Advanced Computing in Europe (PRACE), as well as dedicated HPC services for Universities and other public research facilities and organizations such as MeteoSwiss; or the analysis of data from LHC at CERN by the Swiss Institute for Particle Physics (CHIPP); and data archives for Climate scientists as well as data from PSI. In recent years addition CSCS has begun to provision more generic infrastructure services for the platforms of the Human Brain Project (HBP) [2], the MaterialsCloud of the MARVEL project [3], and the Swiss Data Science Centre (SDSC) [4].

Cloud technologies and public cloud service providers have successfully introduced a service-driven, on-demand model where customers have flexibility to configure infrastructure as well as platform services, namely Infrastructure-as-a-service (IaaS) and Platform-as-a-service (PaaS), respectively. It is unclear though whether public cloud provisioned services are cost-effective for scientific workflows and whether they can yield cost-to-performance ratios for large-scale computing and storage needs comparable to a large academic HPC data center environment. A hybrid solution leveraging cloud technologies can potentially transform service delivery at HPC data centers by introducing key features such as interactivity, on-demand provisioning and service federation between experimental IT facilities and HPC data centers. Hybrid cloud solutions rely on transparent transferability of workloads between different IT infrastructure, for high availability, load balancing and elasticity need. In short, a hybrid solution can address needs for different operating modes at scale without a need to resort to custom middleware implementation for individual community platforms and workflows.

II. Background

Until 2012 CSCS, like many other supercomputing centers, provided each institutional customer with a dedicated service along with dedicated IT infrastructure, in

order to accommodate the customer's unique workflow, user and data management requirements. With the introduction of the new data center facility in 2012 in Lugano and independently of any technological development in the cloud, CSCS began to introduce a horizontal supercomputing platform in order to consolidates services for large research communities such as User Lab with the Swiss Tier 1 and PRACE Tier 0 allocations, as well as HPC cluster services that can be provisioned by a conventional batch scheduling mode for Universities of Zurich and Lugano and PSI. Even the LHC data analysis workloads of CHIPP could recently be integrated seamlessly into the flagship supercomputer "Piz Daint", a hybrid Cray XC40/XC50 system. Only the complex workflows and high availability requirements of platforms such as the MaterialsCloud and the HBP Collaboratory currently require operations of a separate OpenStack cluster.

CSCS HPC ecosystem and OpenStack cloud deployments are currently on discrete IT infrastructure due to fundamental differences that include storage systems, authentication and authorization infrastructure (AAI) and service provisioning methods. Our proposed approach aims at bridging this gap by aligning technologies used for system and software stacks. For instance, at the level of management and provisioning of hardware components, we introduce software defined infrastructure concepts. This essentially means introducing standard API interfaces to compute, storage and network. Currently, HPC systems are considered bare-metal deployments where system and quality of service (QoS) configuration of hardware resources is deterministic and fixed at the system provisioning level. As a result, neither the migration of services and applications nor automation is typically possible in a bare-metal environment, which is tuned for performance and scaling.

A service-oriented architecture that exploits concepts of hybrid cloud technologies such as containers and web services can address complex and growing needs for scientific IT communities. An overview of such an environment is shown in figure 1. Community platform developers and users can develop their services using Infrastructure-as-a-service (IaaS) and Platform-as-a-service (PaaS) and manage them independently on their local IT infrastructure, shared infrastructure like CSCS or public cloud. End users of community platforms are generally agnostic to underlying platform and infrastructure implementation. Cloud technologies such as OpenStack and Kubernetes for container orchestration offer different levels of control for scheduling and resource management for application and platform developers. These technologies are however oriented for enterprise service environment, where availability and consistency of service is achieved through different layers of virtualization and abstraction. Same is true for storage technologies. Cloud storage technologies are typically based on object storage concepts, which is fundamentally different from storage concepts in HPC parallel file systems. RESTful interfaces to HPC services are therefore needed to bridge access to HPC resources from external services and vice versa.

Figure 1 lists some services that need to be extended in today's bare metal HPC ecosystem to incorporate interactivity. A reservation service is an example. It can deliver interactivity for tightly coupled experiments where workflow stages are spread between an experimental facility an HPC data center. A level of control and interactivity is needed to ensure timely execution of tightly-coupled workflows without waiting in a job queue. Another underlying service at the HPC data centers that needs to be extended is the AAI service, particularly AAI federation by accepting external identity providers, which is essential to facilitate implementation of community platforms and complex workflows across distributed IT infrastructure.

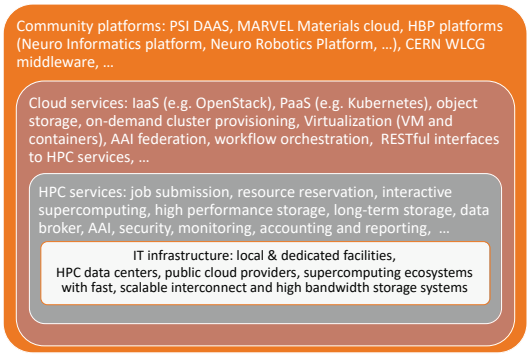


Fig. 1. A service-oriented environment for developing community platforms for hybrid cloud and supercomputing environments

There are several challenges in realizing a hybrid cloud infrastructure for supercomputing and large-scale data management services. Introducing interactivity and on-demand provisioning via external web services require fundamental technological and policy changes in an HPC data center. For instance, traditional HPC services are accessed via command line (ssh) and use batch job submission mechanisms, whereas interactivity in a traditional public cloud environment is achieved by overprovisioning resources. Subsequently, on average resource utilization is low in cloud infrastructure. Supercomputing systems tend to have very high node utilization, often over 95%, at an expense of long job waiting times. Technical solutions will include introduction of secure web accessible services (through so called RESTful APIs) for back-end HPC services, role-based access controls for fine grain access to services, hybrid resource management and scheduling systems for batch and service-oriented accesses and interoperability of high-performance POSIX file systems and object-based files systems.

III. Example of Service Based Implementation

A local e-infrastructure platform called Data Analysis as A Service (DaaS) was built at PSI in 2017, focusing on data storage and data analysis for Swiss academic users [5]. The DaaS infrastructure implements a research data policy, which is consistent with European open research data requirements to make data generated at PSI findable, accessible, interoperable and reusable (FAIR principles). The massive growth of data requires scalable and extensible services for data management, data processing, and data analysis. DaaS, therefore, needs to be able to utilize scalable storage and computing resources within a data center environment such as CSCS in a manner that is transparent to users, for online (during an experiment) and offline (after researchers leaves PSI) analysis.

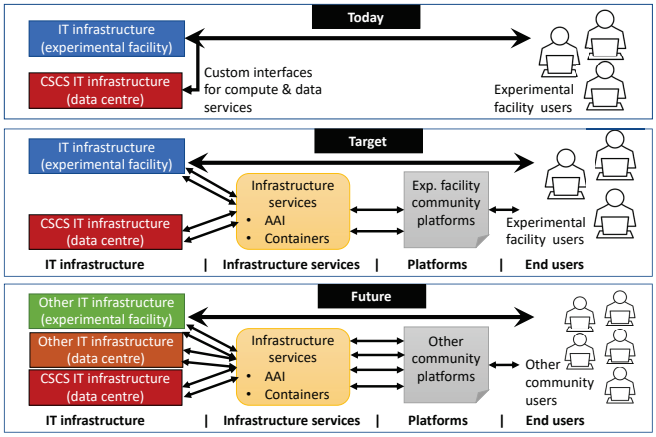


Fig. 2. Evolution of IT infrastructure services for PSI to enable hybrid cloud, HPC and federated IT service. A service-oriented architecture for developing community platforms for hybrid cloud and supercomputing environments

Today, PSI users transparently access CSCS HPC computing (in a legacy batch manner) and storage resources using custom workflows and tools. An example is the PetaBytes archive project that uses custom tools for packaging, archiving and retrieving the datasets within a tape based long-term storage system [6]. Data sets are stored with Persistent Identifiers (PIDs) to be compliant with the FAIR principles. Today's access model used by PSI is *not scalable* and sustainable due to a high level of customization. Each custom instantiation poses an overhead for development and maintenance of services, both for CSCS and for platform developers. Therefore, our target model is to delineate IT infrastructure, infrastructure services, platform services and end user-facing services. Figure 2 highlights the steps that need to be taken to introduce a service-oriented, hybrid HPC and cloud model. Example of infrastructure services include AAI services, APIs for web services to access HPC resources, and virtualization as well as quality of service for interactive, on-demand access of resources. Community platforms target these infrastructure services independent of the underlying IT infrastructure. Experimental users (e.g. PSI) will continue accessing resources as before while PSI platforms can take advantage of both their local IT infrastructure and shared resources at CSCS for scalable computing and storage resources.

The hybrid architecture can be extended in the future to support a diverse range of community platforms to utilize services in a hybrid cloud model i.e. local (experimental facilities) IT plus shared services in an HPC data center or a federated IT infrastructure that has been proposed by a European consortium called Fenix [7].

The Fenix project defines a set of core services and concepts that will be implemented at five leading European HPC data centers. These centers include BSC (Spain), CEA (France), CINECA (Italy), CSCS (Switzerland), and JSC (Germany). The distinguishing characteristic of Fenix service portfolio is that scalable data repositories and supercomputing systems are in close proximity and are well-integrated for optimal performance. High availability and diversity of service portfolio is achieved through the federation of services across five sites by leveraging cloud technologies.

The core service concepts are listed in Table 1. There are a set of traditional HPC services such as the scalable computing services and active data repositories or scratch file systems. Fenix services introduce concept of interactive computing service for on-demand provisioning of clusters and storage systems similar to a cloud environment. Data federation concepts will be accomplished through object storage as part of archival data repositories. Currently, OpenStack Swift protocol has been chosen as an implementation technology across Fenix sites for data federation. Data mover service aims at moving data securely and consistently between archival (POSIX) files to Swift objects. There are supporting services for user management and accounting as well as for networking. Additional data services for compliance such PIDs can be introduced at different sites to support workflows in order to comply with the FAIR principles.

TABLE I. LIST OF SELECTED FENIX SERVICES FOR ENABLING A HYBRID CLOUD AND HPC ECOSYSTEM FOR DATA FEDERATION

Service	Description
Interactive Computing Services	On-demand and interactive access to computing resources at scale. For example, Jupyter and interactive visualization services
Scalable Computing Services	Massively parallel and distributed applications that require a high bandwidth and low latency interconnect system
Virtual Machine Services	Community platforms can use these services for deploying virtual machines. Examples include web servers, DB servers, etc.
Active Data Repositories	Site-local data repositories located close to computational and/or visualization resources, used for storing temporary slave replicas of large data sets
Archival Data Repositories	Federated data store optimized for capacity, reliability and availability, used for long-term storage of large data sets that cannot be easily regenerated
Authentication and Authorization Services	Set of services needed for authentication and authorization with all relevant Fenix services
Data Mover Services	Site-local service for moving data between Archival and Active Data Repositories

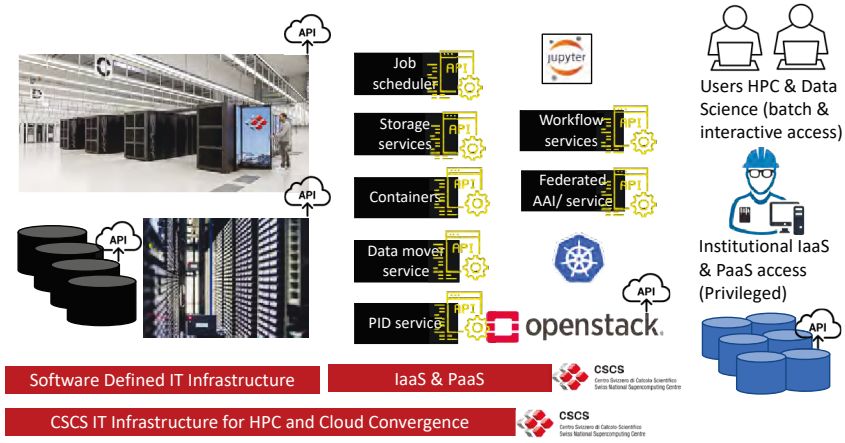


Fig. 3. Ecosystem for extreme-scale data and HPC workflows being developed at CSCS using consolidated IT infrastructure. The goal is to realize hybrid cloud and HPC services.

IV. Evolution of Hybrid Cloud and HPC Services

In order to realize the vision for a research infrastructure that is presented in Figure 2, CSCS has been in the process of deploying software defined IT infrastructure services by using OpenStack technologies, interactive supercomputing service using Jupyter and storage services for HPC and cloud use cases using object storage and data mover technologies. At the time of writing this extended abstract, these services are on discrete systems. Research and development efforts are underway to consolidate scalable IT infrastructure (computing, storage and networking) at CSCS to exploit economy of scale. An ecosystem for hybrid cloud and HPC infrastructure that we would like to realize at CSCS is shown in Figure 3. Existing HPC users can access resources through traditional batch-oriented resource management system as before. New workflows that require data streaming from experimental facilities such as PSI require privileged access for PSI DevOps teams that have flexibility to exploit IaaS and PaaS resources at CSCS. CSCS IT infrastructure therefore can be presented to PSI users transparently as an extension to PSI IT services. PSI compute and data streaming services can use standard APIs such as OpenStack and Kubernetes to access IT resources for a variety of computing and data services including scalable computing services. Federated AAI services allow PSI DevOps teams to manage role-based access control.

Data science workflows and frameworks share similarities with data-driven workflows of experimental facilities namely interactivity, on-demand access, access to external services from HPC services, etc. Hence, the proposed hybrid ecosystem can enable data science services in addition to scientific workflows. The Jupyter service is shown as an example of an interactive computing service that is widely used by data science communities. Typically, a user accesses the frontend in a service-oriented environment, which could be an OpenStack or a Kubernetes cluster (a service-oriented resource management and scheduling system). Data science applications are typically containerized (Docker service) and access to HPC resources are provided via secure APIs. Storage services likewise can be provisioned, on-demand, within an HPC system or a hierarchical storage target.

V. Conclusions and Future Outlook

On-demand, service-driven cloud computing together with extreme-scale computing and data analysis services can potentially support a wide range of emerging scientific workflows by leveraging a combination of HPC and cloud technologies. Extreme-scale data workflow examples include data from experimental facilities such as PSI, data generated by simulation engines such as a weather forecasting system and physics simulations, and data consumed by complex workflow for communities such as HBP and the MaterialsCloud platform. Discrete technology solutions that exist today incur both cost and performance overheads, in most instances by overprovisioning of resources at local IT or public cloud infrastructure and by compromising on user access modes (e.g. interactivity) at HPC data centers. Software defined infrastructure concepts together with a service-driven model that is supported by standard APIs for IaaS and PaaS resources could bridge this gap. CSCS has been developing use case driven solutions for a diverse range of scientific communities with a goal of providing hybrid cloud and HPC services within a consolidated, scalable IT infrastructure. In parallel, efforts are underway to update user access, management and security policies for a hybrid HPC and cloud ecosystem to support a service-oriented architecture.

References

[1] SwissFEL: The Swiss X-ray Free Electron Laser, by Christopher J. Milne, et.a al. *Appl. Sci.* 2017, 7(7), 720; https://doi.org/10.3390/app7070720

[2] HBP platforms: https://www.humanbrainproject.eu/

[3] Materials Cloud: https://www.materialscloud.org

[4] Swiss Data Science Centre (SDSC): https://datascience.ch/solutions/

[5] PSI DAAS: https://www.psi.ch/science/daas-project-swissuniversities

[6] PSI PetaByte archive: https://www.psi.ch/photon-science-data-services/data-catalog-and-archive

[7] Fenix Research Infrastructure: https://fenix-ri.eu

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Towards exascale climate data handling: infrastructure, data management, data services

Stephan Kindermann
Data Management Department
(DKRZ)
Hamburg, Germany
kindermann@dkrz.de

Martina Stockhause
Data Management Department
(DKRZ)
Hamburg
stockhause@dkrz.de

Hannes Thiemann
Tobias Weigel
Sofiane Bendouka
(DKRZ)
Hamburg
thiemann,weigel,bendouka@dkrz.de

***Abstract*—There is a strong requirement to evaluate and compare climate model data originating from modeling centers around the world in international coordinated inter-comparison projects (e.g. the CMIP efforts coordinated by the WCRP-WGCM). Additionally climate model data is increasingly used in interdisciplinary studies. Dedicated data infrastructure as well as data services are necessary to support climate scientists in data analysis activities involving data from this distributed Multi-PByte climate data archive hosted in the Earth System Grid Federation (ESGF). In this paper we summarize the infrastructural challenges the Earth System Science community is confronted with when preparing for future large climate model data inter-comparison projects from the perspective of a large climate data center. We describe specific solution approaches taken to be able to adapt to future technological and infrastructural changes (e.g. object based data management) as well as future climate data volumes. The fundamental underlying data handling problems are related to the geospatial and organizational separation of data centers on the one hand and the need to co-locate data and move computation to the data to be able to efficiently perform large data analysis experiments on the other hand. Data centers are in charge to establish coordinated infrastructural services to hide the related complexities from end users. A set of such services supporting data identification, citation, replication and data processing are introduced.**

Keywords—distributed data infrastructure, data management, climate science

I. INTRODUCTION

The climate model data life cycle starts with the HPC based generation of high volumes of "raw" climate model simulation data. Portions of this data which are subject to large international coordinated model inter-comparison projects (e.g. the WCRP Coupled Model Inter-comparison Project [1]) are standardized according to a set of well defined (project) rules and conventions (see e.g. [15]) and stored at larger national climate research related facilities. These nodes are interconnected in an international peer-to-peer network based on a software stack developed as part of the Earth System Grid Federation [1] providing a consistent set of data search and download services to end users. Some data nodes also take over federation wide responsibilities e.g. by acting as data replication centres. Besides acting as a replication node the German Climate Computing Centre (DKRZ) also provides federation wide long term data storage services as well as data citation and identification services e.g. by integrating the World Data Centre for Climate (WDCC) hosted at DKRZ with ESGF. The long term archival of data e.g. as part of the Reference Data Archive for climate model output of the IPCC DDC (Intergovernmental Panel on Climate Change Data Distribution Centre [21]) is additionally associated to metadata curation steps integrating ancillary metadata information like model related metadata provided by ES-DOC [26].

Providing this large data repository DKRZ is now confronted with the growing end user requirements to support data centre near evaluation experiments by providing access to compute resources co-located to the archive. As also other ESGF data centres are faced with this challenge coordinated efforts are currently starting to be able to jointly respond to these end user needs. An ESGF compute working team started to define an interface to basic functionalities like data sub-setting and re-gridding to be provided by large ESGF (replica) nodes [16]. In Europe the currently starting IS-ENES3 project establishes a service activity with respect to data near processing at centres in Germany (DKRZ), France (IPSL), England (CEDA), Italy (CMCC) as well as Netherlands (KNMI). Yet to enable consistent federation wide data processing services involving existing Petabyte archives and future exascale archives new distributed data management infrastructure and services need to be developed and established.

In this paper we start with these core data handling components currently being established operationally to hide the complexity of high volume distributed climate data handling from end-users. These include data replication (section II), data identification and versioning (section III) as well as data citation (section IV). After this (section V) the integration of these components to support end-to-end climate data analysis workflows is presented.

II. DATA REPLICATON

A. Data access load and failover

Based on overall ESGF experiences from CMIP5 the current estimate for the data access volume to data storage volume ratio is about 10. Thus ~50PBytes from the overall ~5 PByte distributed CMIP5 archive were downloaded and accessed. With an estimation of ~50 PByte for CMIP6 (20 PByte originals + older versions + replicas) we expect ~500 PBbyte download volume to support CMIP6, To be able to handle this load ESGF sites organized themselves in two tiers: Tier 1 nodes act as replication centres, replicating data collections from each other and are thus sharing the data access traffic and are also acting as failover nodes. Tier2 sites are providing original data from modelling centres around the world. Tier1 sites invest in optimizing their

overall network performance as well as ensure high-bandwidth interconnects to their tier1 partner sites for data replication. They also try to ensure that every tier2 site hosted dataset has at least one replica at a tier1 site. Based on current estimates on replica disk space availability at sites (around 4PBytes at DKRZ, IPSL, CEDA and around 20 PBytes at LLNL) these are conflicting requirements as important "core" data will be replicated to each site to support local user communities. In addition to this a large part of CMIP6 data will be transferred into the long term archive hosted at DKRZ (based on tape storage), thus providing long term data curation, preservation and access as part of the ISC (former ICSU) World Data System (WDS) and IPCC DDC.

B. Data replication challenges

The challenges related to data replication can be separated into:

- Scientific and organisational issues (what, when, priorities)
- networking issues (bandwidth, interface to local storage, network providers)
- and related data transfer issues (protocol, failure management)

The prioritization of replica datasets at replica centres is mostly driven by national and institutional responsibilities to support national research groups. On the other hand federation wide replication requirements (with respect to load balancing and reliability) need to be addressed in an ad hoc basis based on inter-institutional agreements and as part of funded projects like IS-ENES3 in Europe. Therefore an international working team was established to address these organizational aspects together with the networking aspects. In this context local infrastructure experts at data centres also collaborate with network providers to establish a monitoring infrastructure (e.g. using perfSONAR [3] deployments at sites) based on which bandwidth bottlenecks can be located and resolved.

To manage parallel large data collection transfer activities (based on http and gridftp) between sites a community replication tool (synda [4]) was developed based on a transfer status management solution (using a relational database). To support future replication loads this community solution needs to be integrated with evolving research data management solutions like globus online [18].

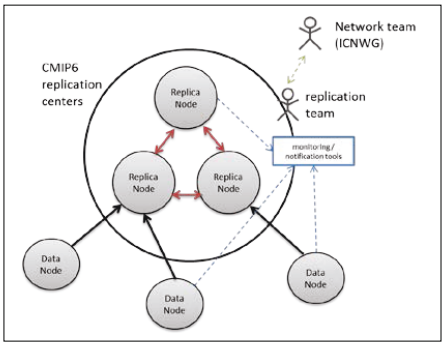


Fig. 1. Overview of replication infrastructure in support of CMIP6

III. Data Objects, Data Identification and Versioning

In this distributed environment climate data sets are stored at different locations using different technologies (e.g. POSIX files systems, object storage, tape storage) during their lifecycle. Additionally data sets get replaced with new versions and have replicas at other sites. To enable reproducible data analysis experiments in such a context and to refer to the input data used (and data products generated) in a stable, technology and location independent manner a persistent identification mechanism needs to be deployed and used. In support of CMIP6 and other future ESGF projects DKRZ developed a Handle [5] based PID infrastructure supporting scalable and reliable PID registration by deploying a distributed message broker (based on rabbitMQ [6]). This scalable PID registration service was then integrated into the ESGF data publication software. Key motivations resulting in the message broker as additional component were to provide fail-over capacity and cashing of burst requests.

This enables the automatic PID registration as part of the ESGF data publication procedure. PIDs are automatically assigned at different granularities: individual data files as well as data-sets referring to complete time series of a specific variable. Data versioning and data replication based relationship information is recorded in the PID metadata records. This way a stable persistent data referencing layer is established as a basis for the provisioning of data location and storage technology independent higher level data management services as well as end user services. As an example the assignment of PIDs to end user defined data collections ("data shopping baskets") is supported. It can also enable better detection of replication failures and replication failure recovery. Sustainability issues for this "core layer" are critical and are addressed in the context of larger international collaborations: the provisioning of an operational distributed Handle based PID infrastructure supporting stable resolution services (based e.g. on PID metadata replication) is ensured by European collaborations as part of the ePIC consortium [12] and EUDAT [24] as well as EOSC-hub H2020 projects. PID metadata agreements and PID collection aspects are discussed and agreed on as part of RDA interest and working groups [25].

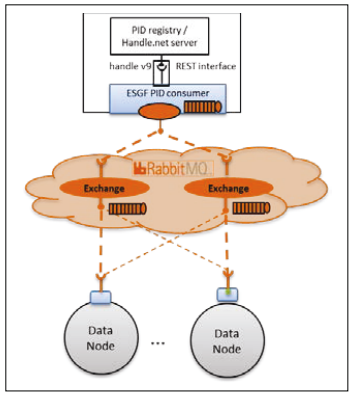


Fig. 2. PID registration infrastructure in support of CMIP6/ESGF

The current expectation is that the deployed PID infrastructure as part of ESGF/CMIP6 data publication scales up to ~50 PByte (and ~50 billion data entities) and probably with tuning by an additional factor of 10. Yet at Exa-Byte scale the core PID registration and resolution infrastructure (currently based on the Handle system) has to be redesigned based on state of the art distributed key-value store and block-chain technology, see e.g. [11].

IV. Data Citation and Long Term Archiving

Data citation based on DataCite DOIs relies on persistent data references as well as data centres taking over responsibility in preserving the referenced data collections. To support data citation of data collections stored in geographically and organisationally separated data centres as in the case of ESGF/CMIP6 an evolving data citation service was developed [8,27]. As ESGF data nodes have no long-term data strategy, it is important to transfer the data after project ending into a long-term data archive such as the IPCC DDC for CMIP6 data underlying the corresponding IPCC AR6. During data archival ancillary metadata is added to the data descriptions of the ESGF extracted from the NetCDF data headers. Examples are ES-DOC [26] documents describing experiments, models, and model set-ups, or data citations including further paper and data references as well as ORCID researcher PIDs.

The CMIP6 Citation Service consists of a database to store citation and relation details, data provider interfaces (GUI and API) to maintain the information, an automatic DataCite DOI registration and metadata update service, and interfaces to other CMIP6 infrastructure components, the ESGF, ES-DOC, PCMDI's publication hub [27], and the long-term archival. Future work is required for the full implementation of the recommendations of the RDA WGDC (Working Group on Data Citation, [26]). A second PID to refer to the subset used in the article is recommended. Such a solution could be developed based on the PID data cart approach (see section III), after solving the long-term data availability issue for the data cart content, i.e. inclusion in the replication and long-term archival strategies. The integration of data usage information in literature based on Scholix API [28] is under development. Integration into data download statistics (ESGF dashboard developed by CMCC) and PCMDI's publication hub are planned as part of IS-ENES3.

V. Data Processing and E-Science Infrastructure Integration

A. Data processing requirements

High volumes of data stored in files in Network Common Data Form (NetCDF [9], and thus HDF5) have to be accessed to support climate model data evaluation as well as data analysis experiments. Efficient processing thus requires data collection at central places with efficient paths to associated compute resources. Whereas in the past these central places were set up and organized more in an institutional uncoordinated fashion, expected data volumes are forcing data centres to establish new optimized environments supporting client scientists in efficient processing of high volume, multidimensional data cubes. The environment is composed of a set of layers: As a storage layer large (parallel) posix file systems need to be exploited by now as cloud storage solutions supporting efficient high dimensional netcdf4/HDF5 file processing are an open research question (see e.g. the pangeo experiences [10]). As processing layer high bandwidth interconnected compute resources need to support core optimized processing software packages like iris, dask and xarray as well as established community tools like nco and cdo or host dedicated backend data cube processing solutions like e.g. Ophidia [19,20].

Higher software layers then exploit these core features to support e.g. larger community climate data evaluation frameworks like the ESMValTool [7] which integrates collections of diagnostics routines in a common input provisioning, output storage and data provenance recording environment.

Additionally different interfaces to the compute layer have to be provided, supporting different usage scenarios:

- Direct access interfaces (e.g. using ssh) to dedicated interactive nodes or virtual machines
- Web based interactive interfaces like jupyter notebooks hosted e.g. in a jupyterhub environment
- Web service interfaces (e.g. based on the OGC WPS standard) to remotely call predefined analysis capabilities.

As a concrete example in the following current work on the data analysis environment at DKRZ is described.

As illustrated in Fig. 3 DKRZ provides a large community data pool for model data (e.g. replicated from ESGF), which is associated to a compute platform. Besides providing direct access to DKRZ HPC resources this compute platform currently evolves to a flexible, integrated service layer relying heavily on virtualisation and containerisation of infrastructural components. Two concrete application scenarios are shown: The so called ECAS service providing an ENES compute service as part of the EOSC-hub project. This compute service provides access to an Ophidia big data processing backend. Additionally OGC WPS conforming web processing services are provided, which are deployed based on the Birdhouse framework [23] and will provide compute interfaces e.g. for the Copernicus climate data store [28].

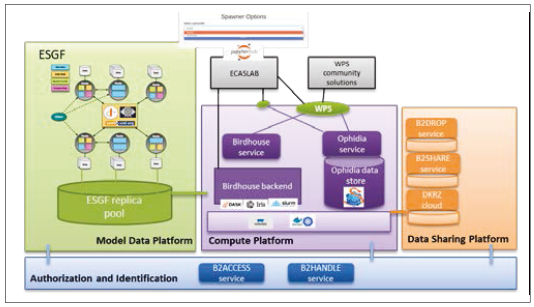


Fig. 3. Processing infrastructure developments at DKRZ

To better support end-to-end analysis workflows and collaboration in projects DKRZ also tries to integrate data import, export and data sharing services developed as part of the EUDAT and EOSC-Hub projects (especially the

B2Drop, B2Share and B2Find services). This enables scientists to add external data sources to their (e.g. interdisciplinary) climate analysis experiment and export and share generated results. At the web interface level Jupyter hub notebooks support the integrated interaction with different workflow steps and documentation of end-to-end workflows.

This newly provided service platform is also useful for many user groups which are currently not supported by national service providers. They have to be supported e.g. as part of international collaborations providing shared resources to scientists. Current examples are the IS-ENES3 H2020 project where processing resources hosted at Germany (DKRZ), France (IPSL), England (CEDA) and Italy (CMCC) are offered to users as part of a new service activity. As part of this e.g. IPCC authors are supported to generate derived data products needed for the next IPCC Assessment report.

As part of the EOSC-Hub project the ENES climate analytics service (ECAS) hosted at DKRZ as well as CMCC is promoted to users. At an international level large ESGF data nodes are preparing the provisioning of OGC WPS based processing services, providing basic data reduction functionalities and thus reducing the need to directly access large data volumes as a first pre-processing step for larger data analysis activities.

The efforts above are described from the perspective of one large climate data centre (DKRZ), yet they are in line with the efforts taken by other climate data centre to support future usage scenarios: Thus the JASMIN analysis platform hosted at STFC Rutherford Appleton Laboratory supports the provisioning of VMs with pre-installed climate software packages and access to the data centre archive to user groups. At NCAR (Boulder, Colorado) a dedicated CMIP analysis platform is provided to US user groups providing access to a large climate data replica pool as well as high performance data analysis servers. NCAR also initiated an (Earthcube program related) effort which evolved to the Pangeo community platform initiative to collaboratively develop software and infrastructure to enable Big Data geoscience research [23]. Also the national computational infrastructure (NCI) located in Canberra, Australia is working towards the provisioning of a consistent climate model data storage and processing environment integrated with existing observational data archives.

VI. Summary and Conclusion

The before described components of data ingest/replication, persistent identification/citation, data processing and result data sharing have to be integrated in the future at data centres and across data centres to provide climate scientists with seamless services to support their workflow. A core requirement here is to establish an integrated provenance service layer enabling the automatic provisioning of provenance records for data analysis results which were generated using the data centre processing layer. These provenance records have to be based on standards like W3C PROV [13], PROV Templates [14], and should be harmonized across climate data centres.

References

[1] WCRP Coupled Model Intercomparison Project, https://www.wcrp-climate.org/wgcm-cmip

[2] Earth System Grid Federation (ESGF), https://esgf.llnl.gov/

[3] PerfSONAR network measurement toolkit, https://www.perfsonar.net

[4] Synda ESGF downloader, https://github.com/Prodiguer/synda

[5] Handle.Net, https://www.handle.net

[6] RabbitMQ message broker, https://www.rabbitmq.com

[7] Eyring, V., Righi, M., Lauer, A., Evaldsson, M., Wenzel, S., Jones, C., Anav, A., Andrews, O., Cionni, I., Davin, E. L., Deser, C., Ehbrecht, C., Friedlingstein, P., Gleckler, P., Gottschaldt, K.-D., Hagemann, S., Juckes, M., Kindermann, S., Krasting, J., Kunert, D., Levine, R., Loew, A., Mäkelä, J., Martin, G., Mason, E., Phillips, A. S., Read, S., Rio, C., Roehrig, R., Senftleben, D., Sterl, A., van Ulft, L. H., Walton, J., Wang, S., and Williams, K. D.: ESMValTool (v1.0) – a community diagnostic and performance metrics tool for routine evaluation of Earth system models in CMIP, Geosci. Model Dev., 9, 1747-1802, doi:10.5194/gmd-9-1747-2016, 2016.

[8] Stockhause, M. and Lautenschlager, M., CMIP6 Data Citation of Evolving Data. *Data Science Journal*, 16, p.30. DOI: http://doi.org/10.5334/dsj-2017-030, 2017

[9] Unidata : *NetCDF* [software]. Boulder, CO: UCAR/Unidata Program Center. doi:10.5065/D6H70CW6, 2018

[10] HDF in the Cloud: http://matthewrocklin.com/blog/work/2018/02/06/hdf-in-the-cloud

[11] Golodoniuc, P., Car, N.J. and Klump, J., 2017. Distributed Persistent Identifiers System Design. *Data Science Journal*, 16, p.34. DOI: http://doi.org/10.5334/dsj-2017-034

[12] The ePIC consortium, https://www.pidconsortium.eu/

[13] W3C PROV-Overview: https://www.w3.org/TR/prov-overview/

[14] Moreau, L., Batlajery, B.V., Huynh, T.D., Michaelides, D., Packer, H.: A templating system to generate provenance. IEEE Trans. Softw. Eng., 2017

[15] CMIP6 Participation guidance for Modelers, https://pcmdi.llnl.gov/CMIP6/Guide/modelers.html

[16] ESGF Compute WPS, https://github.com/ESGF/esgf-compute-wps

[17] World Data System (WDS), http://www.icsu-wds.org/organization

[18] Globus, https://www.globus.org/what-we-do

[19] Ophidia ,http://ophidia.cmcc.it/

[20] S. Fiore, C. Palazzo, A. D'Anca, D. Elia, E. Londero, C. Knapic, S. Monna, N. M. Marcucci, F. Aguilar, M. Płóciennik, J. E. M. De Lucas, G. Aloisio, "Big Data Analytics on Large-Scale Scientific Datasets in the INDIGO-DataCloud Project". In Proceedings of the ACM International Conference on Computing Frontiers (CF '17), Siena, Italy, pp. 343-348, May 15-17, 2017

[21] The IPCC Data Distribution Centre, http://www.ipcc-data.org/

[22] The Birdhouse framework, https://birdhouse.readthedocs.io/en/latest/

[23] PANGEO: A community platform for Big Data geoscience, http://pangeo.io/

[24] EUDAT H2020 Project: https://eudat.eu/

[25] Research Data Alliance (RDA) working groups and interest groups: https://www.rd-alliance.org/groups

[26] ES-DOC Earth System Documentation, https://es-doc.org/

[27] CMIP6 Citation Service, http://cmip6cite.wdc-climate.de

[28] Copernicus Climate Data Store, https://cds.climate.copernicus.eu/#!/home

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Extreme data and computing in numerical weather prediction

Tiago Quintino
Forecast Department
ECMWF
Reading, UK
tiago.quintino@ecmwf.int

Simon Smart
Forecast Department
ECMWF
Reading, UK
simon.smart@ecmwf.int

Peter Lean
Research Department
ECMWF
Reading, UK
peter.lean@ecmwf.int

Peter Bauer
Research Department
ECMWF
Reading, UK
peter.bauer@ecmwf.int

***Abstract*—This paper describes extreme-scale data challenges in numerical weather prediction for both observational input as well as model output data. At ECMWF, significant steps towards much enhanced data processing capabilities at both ends have been achieved and are currently being prepared for operational implementation. Keys to success have been (i) optimized workflows that treat and stream data through the production chain as early as they become available and (ii) the use of object-based datastore solutions that introduce flexibility to varying patterns of data access and flexibility to different hardware options which can be configured to minimize data latency and maximize throughput.**

Keywords—numerical weather prediction, high-performance computing, big data handling

I. Introduction

Numerical weather forecasts are based on tens of millions of observations made every day around the globe and on physically based numerical models that represent processes acting on scales from hundreds of metres to thousands of kilometres in the atmosphere, the ocean, the land surface and the cryosphere. Forecast production and product dissemination to users is always time critical. Forecasting systems are run on O(1000) CPU node allocations and forecast output data volumes already reach petabytes per week today.

For achieving a qualitative change of models, Earth-system simulations need to represent significantly finer scales than today, and – with a focus on enhanced prediction of environmental extremes – with much larger ensembles. Data assimilation methods need to follow this trend to provide accurate initial conditions at such scales [1].

Meeting these requirements translates into at least 1000 times larger high-performance computing and data management resources than today. Achieving a simulation throughput of 1 simulated year per wall-clock day for a single 1-km resolution simulation of the atmosphere already produces a short-fall factor of 100 with present-day models and on present-day computing architectures [2].

The required simulation upgrades immediately translate into similarly enhanced data handling footprints that affect the entire workflow between observational data collection, model output data handling and post-processing, and data archiving and dissemination to users. As shown in Table I, while model output volumes will grow faster than the observational input data volumes, both streams need to efficiently and resiliently operate along tight schedules.

This paper summarizes recent progress of accelerating data handling at the European Centre for Medium-Range Weather Forecasts (ECMWF) exploring new workflow elements and new technologies. These developments will be crucial for achieving ECMWF's strategic goal of operating 5-km global ensemble simulations by 2025.

TABLE I. Present and Future Data Volumes

	Data source	
	Observation input	***Model output***
2018	3×10^8 (2×10^7[PL1]) observations received (assimilated) daily from ca. 80 satellites and conventional sources; >100 GByte / day	1×10^{11} degrees of freedom (15 km model = 5 million grid points x 100 vertical levels x 10 prognostic variables) x 50 ensemble members; 130 TByte / day
2030	1×10^{10}[PL2] observations received daily from O(100) satellites with more complex instrumentation, and commodity devices (phones, cars); 10 TByte / day	1×10^{15} degrees of freedom (1 km model = 500 million grid points x 200 vertical levels x 100 prognostic variables) x 100 ensemble members; 1 EByte / day
Expected growth factor	$O(10^2)$ / day	$O(10^4)$ / day

II. Observational Data

A. Data pre-processing and screening

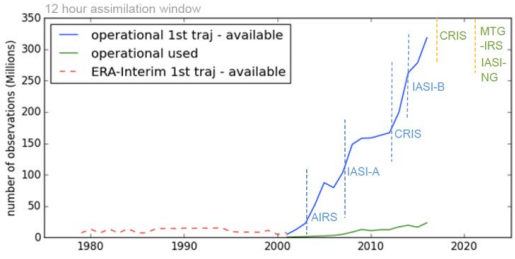


Fig. 1. Time series of observations received by ECMWF within 12 hours. Total number (blue), used number after screening (green) and total number in ERA-Interim reanalysis prior to 2001 [4]. Note volume boosts produced by spectrometer-type instruments such as the Atmospheric Infrared Sounder (AIRS), the Infrared Atmospheric Sounding Interferometer (IASI), the

Cross-track Infrared Sounder (CrIS), and those expected from the Meteosat Third Generation (MTG) Infrared Sounder (IRS) and the IASI-Next Generation (NG).

Observational data is received from a wide range of instruments onboard satellites (98% of total volume), from ground-based stations, balloons, ships aircraft and buoys. The data quality varies between instruments and over time, which requires an effective quality control (screening) mechanism and a methodology that assigns uncertainties to observations in the data assimilation framework [3].

Increasingly, satellite instruments become more complex and provide high data volumes. Only in the recent decade, so-called spectrometers with thousands of spectral channels have contributed to a significant growth in data volumes as well as complexity to extract information for model initialisation (see Fig. 1). The latter requires running compute intensive operators that translate model fields (temperature, moisture etc.) into observed quantities (radiances, reflectivities etc.).

B. *Performance enhancements*

As the largest challenges for observational data handling are data latency (the time spent between reception at instrument and availability at the forecasting centre) and data diversity rather than data volume (see Table I), the largest efficiency gains can be obtained from optimising the pre-processing chain. Apart from operational forecasting, this requirement also applies to research experimentation because numerical experiments rerun data processing, which greatly affects computing capacity.

Optimising the pre-processing targets the workflow by removing static quality checks and reformatting from the time-critical path and by using object-based data stores that allow fast and flexible access to large volumes of data. The data assimilation process itself will access the same database, but additional efficiency gains can be achieved from efficient load balancing as data is heterogeneously distributed over the globe.

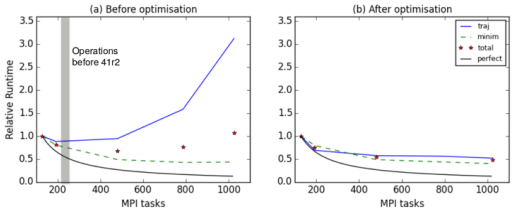


Fig. 2. Scalability of the 4D-Var trajectory (traj) calculation and minimisation (minim) across MPI tasks before (a) and after (b) optimisation. '41r2' refers and grey-shaded area refer to performance at operational nodel allocation. More details are in the text.

Fig. 2 shows the combined results of several optimisations that have been applied to the data assimilation process which produces the initial conditions for the forecasts. This so-called four-dimensional variational assimilation (4D-Var, [5]) compares a model forecast with observations (=trajectory) and then performs an iterative optimisation that computes corrections to the trajectory forecast.

The optimisations mostly comprised the refactoring of message passing and I/O loops to minimize time spent at barriers. The redistribution of active observations across MPI tasks aiming to counter-balance sub-optimal load balancing following the data screening proved to be too communication intensive. However, an overall speed-up of a factor of 2 was achieved when running on 1000 MPI tasks. As Fig. 2 indicates, both efficiency and scalability for the trajectory calculations have been significantly enhanced. Optimal load balancing is key to further improvements as model grid points and observations are not co-located. This can be achieved by better domain decomposition so that data transfers become shorter.

In the future, the main remaining scalability bottlenecks in data assimilation and observational data handling will be the minimisation algorithm itself, because 4D-Var is a sequential method, and the forecast model. Observational data volumes per se will not be a limiting factor for the efficiency of forecasting systems.

III. Model Data Output and Post-Processing

ECMWF forecasts are produced in two one-hour time windows on the critical production path per day. Meteorological output data is generated as a stack of two-dimensional slices of the atmosphere surrounding the globe, known as fields and their size grows quadratically as spatial resolution increases. In bursts of one-hour time-critical windows, the forecast system currently generates around 130 TiB of output per day (22M fields).

The data is stored in byte streams in the GRIB format. This format is intricate, requiring specialised tools to decode and interpret but self-describing, such that metadata can be extracted from a field. This metadata takes the form of a structured set of key-value pairs. Globally, the available metadata space is extremely sparsely occupied, but also includes very dense regions.

The Fields Database (FDB) is the main tool for model output data management. The FDB is a software library and an internally provided service used as part of the weather forecasting software stack. It operates as a domain-specific object store for byte streams of meteorological data such that the output from ECMWF's Integrated Forecasting System (IFS) is written into the FDB, from where it is retrieved by the various post-processing and archival tasks. In this capacity, the FDB also operates as the highest layer within an application controlled hierarchical storage manager.

The Meteorological Archival and Retrieval System (MARS) is a primary service offered by ECMWF that makes many decades of meteorological observations and forecasts available to a wide range of end users and operational systems. At the base of the stack is the tape archive presently built on the IBM High Performance Storage System (HPSS) which is supported by the MARS disk-based cache. The FDB sits between the HPC systems and the rest of the MARS infrastructure, absorbing the forecast output and making it available throughout the post-processing pipeline and elsewhere, efficiently working as a first line cache within the workflow of the HPC.

In practice, operational weather forecasts decay in value very rapidly after they are made (being superseded by forecasts made on later occasions). The FDB exists to make this data quickly and cheaply available while it is broadly useful and thus accessed frequently. Operational data in the FDB has a lifetime of between 3 and 5 days. Because the data flows through this system are predictable, they are application controlled and thus can be optimised for performance.

In current operation production, roughly 200 TiB of data pass through the FDB per day (including both operations and research experiments). More than 100 TiB of this data is then moved to MARS for archival. At any given time, the total contents of the operational FDB are estimated to be between 4 and 5 PiB.

The MARS infrastructure uses the key-value pair nature of the metadata to structure its operational language. Data-write requests use the complete metadata associated with a field to index it. Data accesses use either complete or partial sets of key-value pairs to navigate the indexing information and access the data objects. Which keys are required is well defined according to a schema that guides the data collocation policy.

There are a number of extreme data challenges on the horizon. The amount of data being generated and processed is not only large, it is growing exponentially by, on average, 40% per year. In 1995, operations generated a total of 14 TiB per year, whereas by the end of 2018 just the forecast model was generating 20 TiB in one hour. This is currently made up of 7M fields, each between 1 and 20MiB. To achieve the scientific objectives and goals of ECMWF's 2016-2025 strategy both the resolution and the diversity of the generated data will increase substantially. Given past trends, one third of the growth has resulted from increases in resolution and two thirds from an increase in diversity due to the addition of more physical parameters, vertical levels or more complex ensemble configurations in both analysis and forecast.

The I/O system performance growth has not kept pace with the available computational capacity. Significant software engineering and changes to the use pattern of HPC hardware are required to support the expected data volumes in the future along with the increased metadata load implied by the increase of data diversity. A number of new technologies are becoming available which will help with these challenges. ECMWF is already involved in projects that are looking to develop this capacity, for example NextGenIO [1], which assesses the performance of novel NVRAM technology to enhance, for example, model output post-processing near memory and on the fly. However, the distribution of these resources within an HPC environment is likely to be very different from the large-scale parallel filesystems that currently dominate in all applications. There is not only a significant need for software evolution to support such technologies, but such a system must also be able to cope with varying distributions of storage capacity and the potential lack of global namespaces in the logic of current parallel filesystems.

In addition to the scalability challenges caused by increasing data volumes, the number and diversity of consumers of data are also growing. Commercial customers and member states are increasingly wanting to access the full high-resolution output data set to perform their own computations and run their own limited-area models rather than simply receiving a set of post-processed products from ECMWF. This trend is visible in the emerging European Weather Cloud[2] which showcases the gradual trend to bring compute closer to the data as well as the continuing convergence of HPC and big data technologies for the numerical weather prediction community. A consequence of this is the need to provide significantly better access to the high-resolution model output data within a cloud environment. ECMWF needs to increase not only the flexibility and capability of the output components of the model's I/O stack, but also the ability for processes running outside the HPC environment to access data within the FDB, and the full configurational flexibility of both data storage and access. An overview of the overall computational infrastructure including potential future cloud services is given in Figure 3.

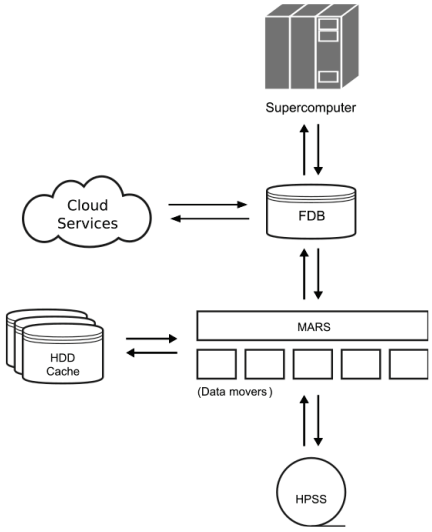


Fig. 3. Overview of potential future infrastructure for model data provisioning at ECMWF.

Within the NextGenIO project, ECMWF has developed a new version of its domain specific object-store FDB for this purpose. This version features a new architecture that separates the front-end from the back-end storage technology. It supports multiple back-end implementations, from a strict POSIX for current operational workload on top of Lustre filesystem to NVRAM based back-ends to support upcoming very low latency and high-density devices interfaced by the PMDK library. The front-end is a plug-in architecture that introduces multiple composable single-purpose components. These implement data routing and collocation functions such as distribution across resources, remote dispatch and data filtering. This flexibility introduced by the front-end and back-end separation of concerns, serves two main goals for ECMWF's plans in the context of exascale data handling: (1) the front-end provides design options of "*what*" to do with the data (e.g. distribute, shard, collocate) and (2) the back-end provides multiple implementations based on different storage technologies. This separation will future-proof the evolution of the data handling at ECMWF as it covers the flexibility needs for the increasing need of data access diversity and the flexibility options for where and how to perform the actual data extraction and distribution on both centralized HPC systems and different layers in cloud-based processing environments.

[1] www.nextgenio.eu

[2] https://www.ecmwf.int/en/newsletter/156/editorial/cloud

IV. CONCLUSION

ECMWF is well aware of the extreme-scale data handling needs in the wake of the anticipated exascale HPC capabilities. The increasing diversity and volume of data and the need to access, manage, compute and distribute data by/to a growing user community for scientific research, service provision and private sectors create unprecedented requirements for workflow management and hardware.

On the observational data input side, ECMWF has created a much leaner workflow that allows data pre-processing at a much earlier stage in the data assimilation process. Previous bottlenecks in the critical processing path have been eliminated and data pre-processing has been parallelised such that the expected growth of O(100) in data volume introduced my more and enhanced satellite instruments will not severely affect the analysis and forecast production efficiency any more. Introducing an object-store data management has been instrumental for reaching this goal.

On the model output data handling side the challenges are actually bigger as data volumes are expected to increase by O(10,000) and data diversity will grow alongside because of more complex Earth-system models, advanced ensemble products and more user communities wanting to access and post-process data closer to the native model output generation. ECMWF has implemented a flexible data handling software - largely supported by the European Commission funded project NextGenIO - that allows a much more flexible handling of data both on the what-to-do-with-which-data at the front-end as well as the where-and-how-to-process back-end. Again, this software is based on an object-store type data management. The software is being tested with new NVRAM-based hardware at present and initial results indicate substantial efficiency gains.

Both developments will support ECMWF's forecast production in the future but also serve as key components for data handling in selected Copernicus services such as the Copernicus Climate Change and the Copernicus Atmospheric Monitoring Service.

ACKNOWLEDGMENT

ECMWF has received funding for the NEXTGenIO project from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement 671951.

REFERENCES

[1] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," Nature, vol. 525, 2015, pp. 47-55.

[2] T. Schulthess, P. Bauer, O. Fuhrer, T. Hoefler, C. Schär, and N. Wedi, "Reflecting on the goal and baseline for exascale computing: a roadmap based on weather and climate simulations," 2019, in press.

[3] K. Ide, P. Courtier, M. Ghil, and A. C. Lorenc, "Unified notation for data assimilation: Operational sequential and variational," J. Meteor. Soc. Japan, vol. 75 (1B), 1997, pp. 71–79.

[4] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and F. Vitart, "The ERA-Interim reanalysis: configuration and performance of the data assimilation system," Q. J. Roy. Meteor. Soc. vol. 137, 2011, pp. 553-597.

[5] P. Courtier, J.-N. Thépaut, and A. Hollingsworth, "A strategy for operational implementation of 4D-Var, using an incremental approach," Quart. J. Roy. Meteor. Soc., vol. 120, 1994, pp. 1367–1387.

[6] S. Smart, T. Quintino, and B. Raoult, "A Scalable Object Store for Meteorological and Climate Data", Proceed. of the Platform for Advanced Scientific Computing Conf. (PASC'17), article 13, DOI: http://dx.doi.org/10.1145/3093172.3093238

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Beating data bottlenecks in weather and climate science

Bryan N. Lawrence*†‡, Julian M. Kunkel‡, Jonathan Churchill§, Neil Massey¶, Philip Kershaw¶, Matt Pritchard¶
*National Centre for Atmospheric Science, †Department of Meteorology,‡Department of Computer Science
University of Reading, Reading, U.K.
§Scientific Computing Department, ¶Centre for Environmental Data Analysis, RALSpace
STFC Rutherford Appleton Laboratory, Didcot, U.K
Corresponding Author: bryan.lawrence@ncas.ac.uk

***Abstract*—The data volumes produced by simulation and observation are large, and growing rapidly. In the case of simulation, plans for future modelling programmes require complicated orchestration of data, and anticipate large user communities. "Download and work at home" is no longer practical for many use-cases. In the case of simulation, these issues are exacerbated by users who want simulation data at grid point resolution instead of at the resolution resolved by the mathematics, and/or who design numerical experiments without knowledge of the storage costs.**

There is no simple solution to these problems: user education, smarter compression, and better use of tiered storage and smarter workflows are all necessary – but far from sufficient. In this paper, we introduce two approaches to addressing (some) of these data bottlenecks: dedicated data analysis platforms, and smarter storage software. We provide a brief introduction to the JASMIN data storage and analysis facility, and some of the storage tools and approaches being developed by the ESiWACE project. In doing so, we describe some of our observations of real world data handling problems at scale, from the generic performance of file systems to the difficulty of optimising both volume stored and performance of workflows. We use these examples to motivate the two-pronged approach of smarter hardware and smarter software – but recognise that data bottlenecks may yet limit the aspirations of our science.

***Index Terms*—HPC, exascale, big data, extreme data, POSIX object store, NetCDF**

I. Introduction

Weather and climate science exploit vast amounts of observational data and generate vast amounts of simulation data. Data volumes and velocity are increasing rapidly. This growth in data is driven by computing capacity – both within instruments and in supercomputing. Major weather centres will approach an exabyte of data in the near future, years before they have access to exascale computing, and so we believe the *first* exascale challenge for the scientific community is a data challenge, and the computing challenge [1] will follow! In this paper, we concentrate on the bottlenecks introduced into the relevant workflows by the volume and velocity of that data and describe some existing and proposed solutions.

II. Context

The growth in data volumes arises from the inexorable exploitation of computing in instruments and simulation. In particular, both the weather and climate communities seek to develop ever higher resolution models of the earth system, and run them in ensembles (e.g. see Figure 4 in [2]) Such extra resolution leads directly to larger volume output, and handling that in a timely manner brings velocity issues – can the input/output, storage, and workflow systems deal with the data in a timely manner?

An immediate question when faced with such issue is "Do we really need all that data?". The answer is almost certainly not, insofar as some of the data being written out has little meaningful information content — being beyond the meaningful resolution [3] — yet it is written because people think it *might* be useful in the future. Similarly, perhaps not all ensemble members need to be fully written out, and both temporal resolution, and opportunities for online analysis before writing data out should all be considered. However, in many cases where the eventual analysis is not yet determined, the full resolution data may be needed since re-running models may be too expensive, or even impossible. In any event, reductions in output from "one-off" decisions will only postpone bottlenecks being introduced by the drive

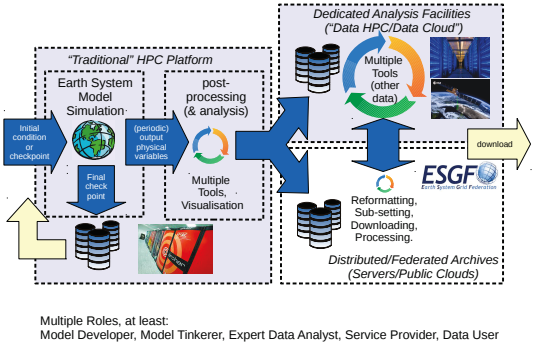


Fig. 1. Heterogeneity in the workflow platforms and requirements: from traditional HPC platforms, to dedicated analysis facilities, and data management and distribution systems, all with different requirements, serving users with a multitude or roles.

The workflow environment involved is complicated. Traditional HPC platforms have been augmented by dedicated

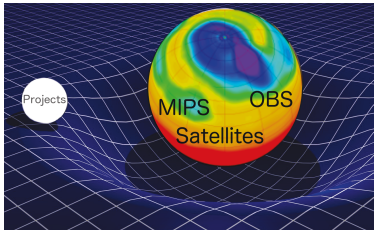


Fig. 2. Data Gravity: the JASMIN concept of a data commons is predic providing a large managed archive of data from ground based obser satellites, as well as simulations from major modelling campaigns Model Intercomparison Projects (MIPs). That provides an incentive f to bring and share their own project data.

data analysis facilities, and complicated systems fo distribution such as the Earth System Grid Federation [in use (figure 1). The storage capacity, types, and perfor requirements are very different, and each has a di class of storage bottleneck to consider. On HPC pla the main issue is often performance — reaching sus I/O performance from and to disk. On analysis pla there is I/O performance and storage to consider, and in data distribution systems, network [5] and software issues dominate to the point where most groups rely on dedicated local archives rather than personal downloading (e.g. see [6], in particular their figure 2).

III. Customised Hardware

In the UK academic community, large weather and climate simulations are primarily carried out on one of two national supercomputers: ARCHER (in Edinburgh) or NEXCS (a portion of the Met Office supercomputer, in Exeter). Neither have large storage and/or analysis systems, and data output is migrated to JASMIN, a data analysis supercomputer for environmental supercomputing (near Didcot, in Oxfordshire). Dedicated high bandwidth network links are available to supplement backbone networks for data transfer.

JASMIN has been designed for environmental data analysis. As of September 2018 it has over 40 PB of storage, and over 10,000 cores distributed between a batch cluster and a community cloud. JASMIN implements a data commons (fig. 2) utilising the managed archive from the Centre for Environmental Data Analysis (CEDA,https://ceda.ac.uk) to underpin the services provided by JASMIN (fig. 3).

JASMIN is configured with a high performance storage environment [7], which is heavily used – data read rates exceed 1 PB/day for multi-day periods (fig. 4). However, despite the heavy use, there is considerable performance "left-on-the-floor", as not all user codes can make effective use of the input/output performance available.

As of early 2018, the storage was divided into five classes: home, user scratch, group work space (GWS), and archive; with most of the space allocated to the archive (5 PB) and GWSs (>12PB). Users are generally assigned access to one

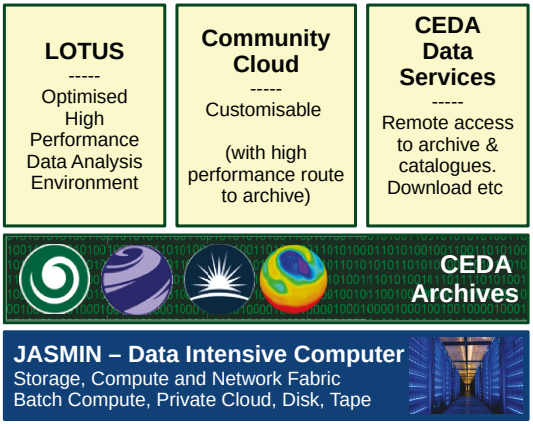


Fig. 3. JASMIN provides a range of services which exploit the CEDA archives and the customised hardware, the most important of which are the LOTUS batch cluster, the Community Cloud, and the CEDA data services, which together provide "Platform-as-a-Service", "Infrastructure-as-a-Service" and "Software-as-a-Service".

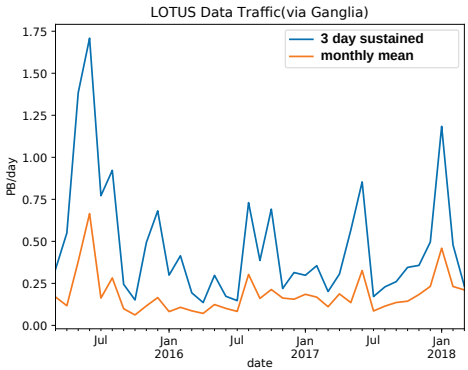


Fig. 4. Network traffic from storage into LOTUS showing data movement over several years. The blue line is the three day sustained average, the orange line, the monthly mean.

or more GWS, and the GWS allocations are constrained within consortium allocations controlled by an external board.

During most of the previous years, growth on disk was almost linear (fig 5, top panel[1]). However that linear growth did not represent user demand, which was heavily constrained: the middle panel of fig 5 shows the archive on disk, and how it has been constrained by the allocation cap, despite the higher underlying growth in most of the archive – one example of which is the Sentinel data, shown in the bottom panel of figure 5. The group workspaces were also constrained: Figure 6 shows that much of the user growth was within GWS

[1]Note that the early 2018 increase was due to data replication associated with an upcoming upgrade.

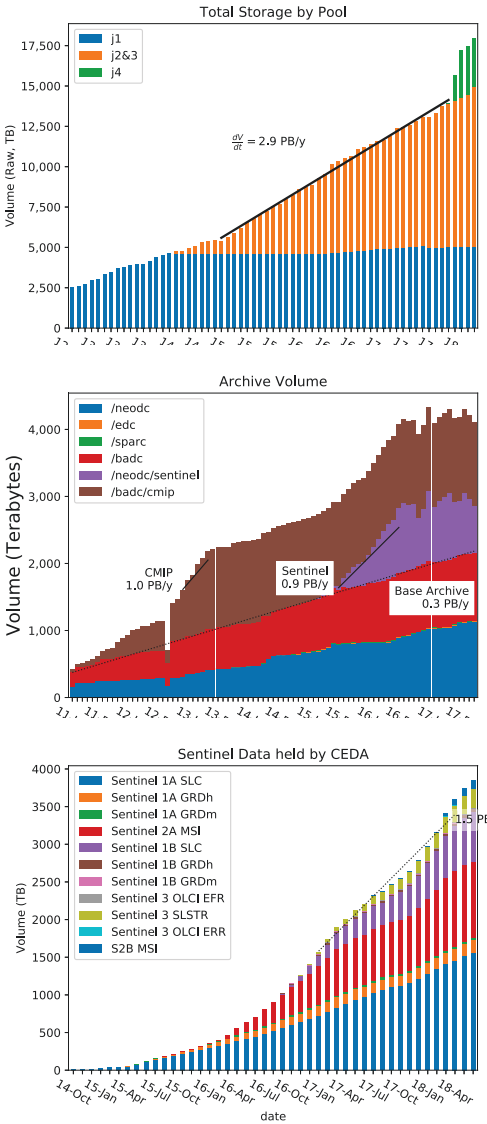


Fig. 5. Three aspects of storage volume growth: the total disk usage by storage pool (see text); total archive volume on disk, and total of the Sentinel data held in the archive (on tape and disk). Key points to note are that the overall linear growth (as opposed to exponential growth) is because of constraints on the archive and group work spaces sizes on disk (see text).

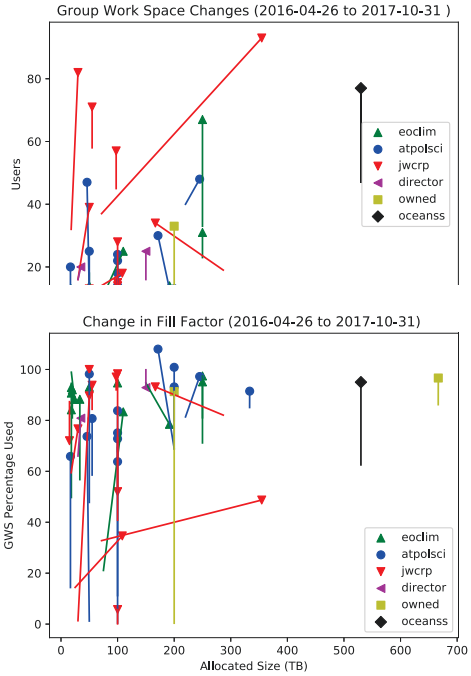


Fig. 6. Usage and fill factor on selected Group Work Spaces over 18 months to October 2017. Lines show change during this period, from beginning to end (denoted with the icons, which indicate the consortium). The top panel shows how the number of users has grown with most workspaces, while the bottom shows how the group work spaces have filled up to their allocations. In both cases, some GWS have also changed in size over that period.

and that users were constraining themselves to fit within their GWS allocations.

The split between archive and GWS, and the constraints which are applied to both, provide key mechanisms used by JASMIN to turn what would have otherwise been exponential data growth into relatively manageable linear growth (although even linear growth will not be affordable on disk if it exceeds the Kryder rate [8]).

IV. Customised Software

Some of the solutions to volume and velocity need to be addressed in both hardware and software.

Where volume and velocity combine, performance becomes an issue. As already noted, not all existing workflows make good use of parallel file systems, and may be more suitable for other storage media. However, even where workflows are well suited for parallel file systems, the file sytems themselves bring limitations which arise from the failure of POSIX at scale to handle high volume concurrent metadata look-ups and very large numbers of processes attempting to access a handful of files. Solutions generally involve application level tuning to local systems, resulting in poor performance portability.

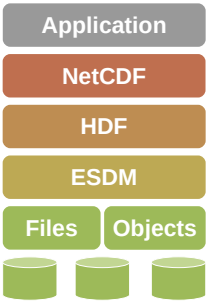


Fig. 7. The Earth System Data Middleware, ESDM, lies between the HDF5 library and storage volumes.

Migrating to object stores is one possible solution [9], but only as part of a plan which addresses higher performance at write time using traditional interfaces. However, object stores are subject to a declining Kryder rate too, so tape is an integral part of planning at most sites, including JASMIN, providing lower performance (and cheaper) storage where the "coldest" data can still be accessed quickly.

Object stores and tape bring another set of issues in that there is little available portable software in the weather and climate community which can easily exploit such storage in workflows. Data placement and appropriate metadata are key, but hierarchical namespaces are limiting, users generally do not have control over data placement, and system controls are often blind to expected usage and workflow requirements. While sophisticated solutions for tape usage exist at major sites (e.g. the ECMWF MARS system [10]), they do not yet incorporate object stores, or if they can (or will soon), they do not deliver portable solutions.

The Centre of Excellence in Simulation of Weather and Climate in Europe (ESiWACE, https://esiwace.eu) is addressing these issues in a focused attempt to develop portable software. Currently there are two strands of activity:

1) The *Earth System Data Middleware* (ESDM), which provides a library which sits between traditional HDF and NetCDF interfaces and storage to deliver performance portability; and
2) The *Semantic Storage Tools* which are aimed at providing suitable portable interfaces to both tape and object storege and providing users the ability to manage their own placement on tiered storage, without losing visibility of their metadata.

A. Earth System Data Middleware

The ESDM targets performance portability by providing software that can be linked into existing applications, but take advantage of knowledge of the local storage environment. Design goals include: (1) Ease of use and deployment; (2) Exploiting knowledge of data structures and scientific metadata to provide efficiency, (3) Supporting multiple read-patterns

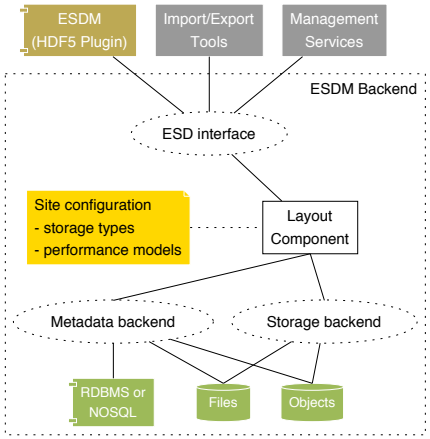


Fig. 8. Core architecture of the ESDM. The HDF plugin and external tools interface to the layout component which is configured with information about available storage types and their expected performance. Metadata and storage backends can use whatever is available.

efficiently, and; (4) Reducing the penalties of shared file access (i.e. deliver "lock-free" writes in parallel applications).

Ease of use is delivered by providing a library which can be linked into existing applications using HDF or NetCDF (figure 7 along with configuration which involved site-specific optimised data layout schema, figure 8). Administration and user tools will provide import export and monitoring.

Performance is delivered by exploiting knowledge of the scientific structures to deliver the necessary lock-free writes by handling data as atomic fragments.

The current status of the ESDM software is that prototypes have been built on a number of systems, and it has been demonstrated to perform signficantly better on Lustre file systems than the native HDF5 writing to Lustre. Details of that performance, and results on other systems will appear elsewhere. User management tools are not yet available.

Future plans include exploiting internal ESD backend daemons to rearrange data for multiple different access patterns requested by "usage hints" delivered at write-time, or via the user-tools interface. These daemons will also be able to rearrange data on the fly for export to remote sites (for example, via Globus).

B. Semantic Storage Tools

The semantic storage tools target direct use of object stores by user software, as well as user-controlled data management in a tiered storage environment.

Currently it is not easy to exploit object stores directly in normal user workflows, most software is predicated on systems and libraries which expect to be working with POSIX filesystems. S3NetCDF (fig 9) addresses this for Python users by providing a drop-in-replacement for NetCDF-python. NetCDF datasets are fragmented using the Climate Forecast

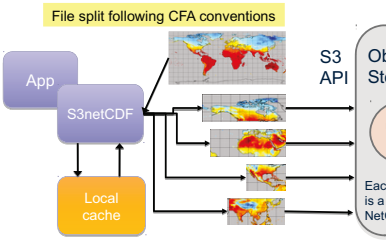


Fig. 9. S3NetCDF involves a splitter which utilises th split a multi-field NetCDF file into a mater file and sma are all stored separately (on tape, or on disk in an object

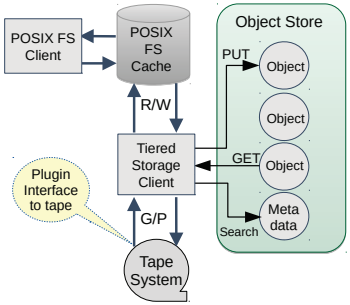


Fig. 10. CacheFace will provide a POSIX front end which manages and migrates data between storage tiers while exposing NetCDF and other metadata to the user regardless of where the data is stored, whether on tape, or disk.

Aggregation conventions [11] resulting in a set of individual NetCDF files which can be stored as objects or files, along with a master array describing how they are aggregated. Users can keep the master array on normal disk, and S3NetCDF simply opens that file, and reads/writes the fragments into/from memory from/into storage.

S3NetCDF exists as a functional prototype, but it accesses fragments in serial, and performance is relatively poor. It is currently being rewritten to exploit the available parallelism to delivery what is hoped to be even better performance than is available using normal POSIX disk access.

Even without direct access to object stores from user codes, object stores can be treated like tape, and used for stashing "colder" data for later use. However, where users are managing this process, the major problem is maintaining information about what is on such storage. Lists of filenames are inadequate, and local bespoke solutions do not allow users to mange their data across multiple sites. These issues are being addressed by the development of cache facing software that (1) manages data migrations, and (2) allow users to manage their own metadata about what is where.

This software, currently known as CacheFace, depends on three key internal components: a data migration utility, a cache management utility, and a metadata system. Development on each is underway, with the data migration tool reaching a sufficient level of maturity so as to be deployed on JASMIN (as the JASMIN Data Migration App) in the final quarter of calendar year 2018. The other two only have rudimentary prototypes, but have the same fundamental requirements as the ESDM and S3NetCDF, so development is expected to be relatively swift. Exploiting these underlying similarities will be one of the goals of the ESiWACE2 project beginning in 2019, with long-term maintenance of the tools being picked up by institutional partners.

V. Related Work

A number of sites are developing hybrid HPC/cloud solutions, and some are at a similar scale in terms of compute, e.g [12]. However, we believe JASMIN is unique in terms of the co-location with a managed archive crossing a wide range of environmental data, although the Polish Innovation Testbed hosts a range of earth observation data [13], and a number of sites are providing computational facilities alongside ESGF climate data.

The ESDM is built on a middleware heritage that some may argue began with ADIOS [14], and has many characteristics in common with sophisticated solutions for buffering data flow in tiered disk storage (e.g. [15], [16] and managing scientific workflows [17]. The ESDM differs from these more generic solutions, by attempting to make use of our domain specific knowledge about the contents of NetCDF data.

There are broadly two current approaches to exploiting object stores: attempts to use middleware to unify I/O stacks (as we are doing with the ESDM) or providing "POSIX-like" or "POSIX-light" file system interfaces that drop some of the full POSIX requirements in order to exploit object stores efficiently. Differing examples from the research and commercial sphere include MarFS [18] and QuoByte[2]. Our approach is somewhat different, by again using our domain specific knowledge (specifically, the Climate Forecast conventions [19]) and the ability to split metadata and data to effectively exploit object stores. The same domain specific knowledge provides the key point of difference for our CacheFace development from other data migration and caching systems (of which there are too many to reference here).

VI. Summary

Modern weather and climate workflows demand customised data analysis environments with specialised hardware and user configurable software environments (virtualisation, containerisation etc). These requirements are met in the UK by the JASMIN facility which co-locates a "community cloud" with a managed archive, a large batch cluster, and a sophisticated tiered storage system. Over the last few years, most user workflows have been able to be accommodated on JASMIN disk, with tape used only for backup and long-term archive, however, projections of future demand suggest that "disk-only"

[2]https://www.quobyte.com

workflows will need to be supplanted by workflows which include more tiers of storage, including tape.

Within those workflows more parallelism will be necessary to avoid unreasonable wall clock times, but such parallelism will not eventuate without both new algorithms and approaches by users and the widespread availability of more efficient and smarter storage middleware and data management software. The European ESiWACE project is addressing these middleware and data management software requirements by developing two families of products: high performance middleware to lie beneath commonly used software libraries like HDF and NetCDF4 (the "Earth System Data Middleware, ESDM"), and user deployable portable tools to manage data in a tiered storage environment.

These two approaches to beating data bottlenecks, smarter hardware and software, will not be enough on their own. The reality of storage economics coupled with feasible data production volumes and velocity mean that despite technology innovations, the most important approach to these data bottlenecks will be avoiding the problem in the first place by writing less data! This means that experimental design and analysis workflows will need radical rethinking — a process that will inevitably involve the entire scientific community, not just the technical experts.

Acknowledgements

JASMIN is supported by the UK Natural Environment Research Council and Science and Technology Facilities Council. The ESiWACE project is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 675191. The authors are grateful to the entire JASMIN and CEDA teams and the ESiWACE team involved with WP4: Exploitability (of storage).

References

[1] B. N. Lawrence, M. Rezny, R. Budich, P. Bauer, J. Behrens, M. Carter, W. Deconinck, R. Ford, C. Maynard, S. Mullerworth, C. Osuna, A. Porter, K. Serradell, S. Valcke, N. Wedi, and S. Wilson, "Crossing the chasm: How to develop weather and climate models for next generation computers?" *Geoscientific Model Development*, vol. 11, no. 5, pp. 1799–1821, May 2018.

[2] J. Mitchell, R. Budich, S. Jousssame, B. Lawrence, and J. Marotzke, "Infrastructure strategy for the European Earth system modelling community 2012–2022," 2012.

[3] S. Abdalla, Lars Isaksen, Peter A.E.M. Janssen, and Nils Wedi, "Effective spectral resolution of ECMWF atmospheric forecast models." *ECMWF Newsletter*, vol. 137, pp. 19–23, 2013.

[4] D. N. Williams, B. N. Lawrence, M. Lautenschlager, D. Middleton, and V. Balaji, "The Earth System Grid Federation: Delivering globally accessible petascale data for CMIP5," in *Proceedings of the 32nd Asia-Pacific Advanced Network Meeting*, New Delhi, Dec. 2011, pp. 121–130.

[5] E. Dart, M. F. Wehner, and Prabhat, "An Assessment of Data Transfer Performance for Large-Scale Climate Data Analysis and Recommendations for the Data Infrastructure for CMIP6," *ArXiv e-prints*, vol. abs/1709.09575, 2017, primaryClass: cs.DC.

[6] V. Balaji, K. E. Taylor, M. Juckes, B. N. Lawrence, P. J. Durack, M. Lautenschlager, C. Blanton, L. Cinquini, S. Denvil, M. Elkington, F. Guglielmo, E. Guilyardi, D. Hassell, S. Kharin, S. Kindermann, S. Nikonov, A. Radhakrishnan, M. Stockhause, T. Weigel, and D. Williams, "Requirements for a global data infrastructure in support of CMIP6," *Geoscientific Model Development*, vol. 11, pp. 3659–3680, Sep. 2018.

[7] B. Lawrence, V. Bennett, J. Churchill, M. Juckes, P. Kershaw, S. Pascoe, S. Pepler, M. Pritchard, and A. Stephens, "Storing and manipulating environmental big data with JASMIN," in *2013 IEEE International Conference on Big Data*, Oct. 2013, pp. 68–75.

[8] P. Gupta, A. Wildani, E. L. Miller, D. Rosenthal, I. F. Adams, C. Strong, and A. Hospodor, "An economic perspective of disk vs. flash media in archival storage," in *2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*. IEEE, 2014, pp. 249–254.

[9] D. Goodell, S. J. Kim, R. Latham, M. Kandemir, and R. Ross, "An Evolutionary Path to Object Storage Access," in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, Nov. 2012, pp. 36–41.

[10] M. Grawinkel, L. Nagel, M. Masker, F. Padua, and A. Brinkmann, "Analysis of the ECMWF Storage Landscape," in *Proceedings of the 13th USENIX Conference on File and Storage Technologies*. Santa Clara: USENIX Association, 2015, pp. 15–27.

[11] D. Hassell, "The CFA-netCDF conventions," http://www.met.reading.ac.uk/˜david/cfa/0.4/cfa.html.

[12] Y. Li, X. Zhang, A. Srinath, R. B. Getman, and L. B. Ngo, "Combining HPC and Big Data Infrastructures in Large-Scale Post-Processing of Simulation Data: A Case Study," in *Proceedings of the Practice and Experience on Advanced Research Computing - PEARC '18*. Pittsburgh, PA, USA: ACM Press, 2018, pp. 1–7.

[13] A. Romeo, S. Pinto, S. Loekken, and A. Marin, "Cloud Based Earth Observation Data Exploitation Platforms," New Orleans, Dec. 2017.

[14] J. Lofstead, F. Zheng, S. Klasky, and K. Schwan, "Adaptable, metadata rich IO methods for portable high performance IO," in *IEEE International Symposium on Parallel & Distributed Processing, 2009. IPDPS 2009*. IEEE, 23-29 May 2009, pp. 1–10.

[15] A. Kougkas, H. Devarajan, and X.-H. Sun, "Hermes: A Heterogeneous-aware Multi-tiered Distributed I/O Buffering System," in *Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '18. New York, NY, USA: ACM, 2018, pp. 219–230.

[16] B. Dong, S. Byna, K. Wu, Prabhat, H. Johansen, J. N. Johnson, and N. Keen, "Data Elevator: Low-Contention Data Movement in Hierarchical Storage System," in *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*. Hyderabad, India: IEEE, Dec. 2016, pp. 152–161.

[17] J. Wang, D. Huang, H. Wu, J. Yin, X. Zhang, X. Chen, and R. Wang, "SideIO: A Side I/O system framework for hybrid scientific workflow," *Journal of Parallel and Distributed Computing*, vol. 108, pp. 45–58, Oct. 2017.

[18] J. Inman, W. Vining, G. Ransom, and G. Grider, "MarFS, a Near-POSIX Interface to Cloud Objects," *;login:*, vol. 42, no. 1, p. 6, 2017.

[19] D. Hassell, J. Gregory, J. Blower, B. N. Lawrence, and K. E. Taylor, "A data model of the Climate and Forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1)," *Geoscientific Model Development*, vol. 10, no. 12, pp. 4619–4646, Dec. 2017.

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Future I/O architectures and infrastructures for extreme-scale data analytics

Dirk Pleiter
Jülich Supercomputing Centre
Forschungszentrum Jülich
Jülich, Germany
d.pleiter@fz-juelich.de

***Abstract*—Future supercomputing infrastructures have to accommodate the needs of scientific applications and workflows that have in common that they involve large data volumes as well as the need for scalable compute resources. Based on the roadmaps of the underlying technologies new approaches are needed to satisfy both the requirements for storage capacity as well as storage performance capabilities. Furthermore, supercomputing centres need to become more open in order to facilitate external data injection and access to different, geographically dispersed data sources. One of the challenges of designing future supercomputing infrastructures that meet the needs of the aforementioned class of applications is the lacking ability of defining their needs. We will advocate one method for capturing these needs mainly focussing on data transport and storage requirements. We will furthermore report in this talk about results from different projects aiming on realising new I/O architectures and supercomputing infrastructures. This includes in particular the EC-funded projects SAGE and ICEI.**

***Index Terms*—I/O, workload characterisation, e-infrastructures**

I. Introduction

At centres like the Jülich Supercomputing Centre we observe an increasing number of scientific workloads[1] which have in common that they need scalable compute resources and involve large amounts of data. We use the term "data-intensive HPC workloads" for these type of workloads. The characteristics of these workloads and their needs often deviate significantly from the more traditional simulation-based HPC workloads. Data-intensive HPC workloads include workloads that consume large amounts of experimental data and process this data using scalable compute resources or simulations that produce large amounts of results that may need to be processed while data is still in transit.[2] These work flows may furthermore include steps that require different kind of architectures, e.g. simulation and deep learning steps.

One of the challenges in this field is the lack of established approach to characterise these applications. Many of the simulations applications can be categorised in terms of the Berkeley dwarfs [2]. For any of these dwarfs a significant amount of knowledge is widely available on the relation of application characteristics and system architecture characteristics. Also for data-intensive applications categorisation schemes have been proposed (see, e.g., [3]), but the take-up of these schemes is so far limited. Unlike the Berkeley dwarfs they are not yet a good instrument for designing future systems and infrastructures.

Giving the key features of the workloads, namely the use of scalable HPC systems with a high throughput of arithmetic operations and the extreme size of the consumed and/or produced data sets, providing sufficient data store and data transport capabilities are the obvious challenges for future infrastructures for data-intensive HPC workloads. Emerging technologies, like new memory hardware technologies or data store software technologies, will help to address these challenges. However, significant efforts are needed to integrate these into future infrastructures and to enable applications to efficiently exploit these.

The remaining part of this paper is organised as follows: In section II we explore different approaches to characterising the needs of data-intensive HPC workloads. We then look in section III into a selected choice of future technologies and architectures, which we consider particularly interesting in this context. In section IV we present the architecture of a federated e-infrastructure, which is in the process of being realised and will be optimised for extreme-scale data analysics workloads, before presenting summary and conclusions in section V.

II. Application Characterisation and Requirements

In this section we explore different strategies for characterising data-intensive applications and to derive requirements for designing future I/O architectures and infrastructures for extreme scale data analytics.

a) Annotated use case diagrams: Annotated use case diagrams are obtained by mapping the different steps of the workload onto an abstracted machine architecture. The abstracted machine architecture is defined as a graph comprising the following components:

- Data ingest nodes
- Data repository nodes
- Data processing nodes
- Data transport edges

As an example we consider the processing of brain images within a deep learning workflow [4] for which the

[1] We use the term "workloads" instead of "applications" to include cases where only single applications are executed as well as more complex work flows. We furthermore assume a workload definition to also include the specification of a problem size.

[2] We use the term "in transit" as defined in [1].

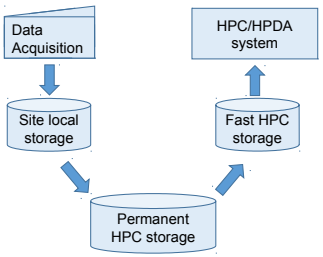


Fig. 1. Example for an annotated use case diagram for processing brain image data.

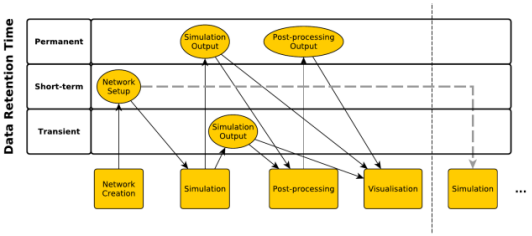


Fig. 2. Example for an retention time analysis for a brain simulation work flow.

corresponding use case diagram is shown in Fig. 1. Data, which is produced by high-performance microscopes at one site, is first buffered in a site local storage before being transferred to a large-capacity archive. From there the data is staged into a fast HPC storage tier such that data can be read at a sufficiently high rate by a high-performance data analytics (HPDA) system. After creating the diagram, an in-depth analysis of the workload is required, which involves, e.g., analysis of the rate at which data is produced by the microscopes or consumed by the HPDA system. Based on this information, the components of the diagrams can be annotated with capability and capacity requirements. The analysis for the specific use case considered here goes beyond the scope of this paper.

b) Retention time analysis: Another dimension to workload characterisation can be added by identifying the retention time of the involved data objects. This has been proposed for HPC jobs [5], for which taking job duration as reference time scale is a natural choice. In this case data objects used within a work flow can be classified as follows:

- *Transient*: Data discarded on HPC job completion or when later processing steps are concluded;
- *Short-term*: Data used throughout the execution of the work flow;
- *Permanent*: Data outliving the system producing it.

For other workloads the choice of another reference time scale may be more appropriate.

Retention time analysis is useful in the context of hierarchical storage architectures comprising a high-performance tier, which will typically be smaller in capacity, and a large-capacity tier, which will provide lower performance. Permanent data objects within will accumulate over time and thus need to be stored in a large-capacity tier. To save limited bandwidth to the large-capacity tier and to ensure rather fast access, only the high-performance tier should be used for transient data objects. As an illustrative example we show a brain simulation work flow in Fig. 2 as it might be realised using the NEST simulator [6]. During an initial stage this application creates a network, which might be re-used for other simulations, i.e. it is retained after the life of the job that produced this data object. During the simulation stage other data objects are produced, which need partially be kept permanently and that are partially analysed or visualised while the simulation is running, i.e. these data objects are both permanent and transient.

c) Functional requirements: Workloads do not only come with performance requirements, also functional requirements need to be met. In this contribution we would only highlight some selected requirements, which are expected to have a significant impact on the way HPC systems are operated in the future:

- *Enable use of various data sources*: HPC systems today tend to be designed with limited capabilities for transferring data into or out of the data centre. Workloads as described before, involving, e.g., processing of vast amounts of experimental data, require the boundaries between HPC systems and the outside world becoming more permeable.
- *Facilitate data sharing following the FAIR principles [7] and collaborative data processing*: With costs for creating data assets increasing, the need for exploiting its value as much as possible becomes even more important. Data sharing and collaborative data processing capabilities enable exploitation of data assets by a larger number of scientists. More value is created by connecting different data assets. This may require federation of services to support data localisation, data access and data transport when data is distributed over different sites. As access to such data assets is typically protected, this furthermore requires the ability to manage access rights to data objects through different control domains.
- *Provide interactive access to data and compute resources*: Today HPC systems are predominantly used in batch mode, i.e. a central resource manager schedules jobs without any user involvement. For various steps within the scientific discovery process and data analytics work flows interactive frameworks, like Jupyter notebooks, have become popular. Interactivity may also be needed for analysing and visualising data of running HPC applications, possibly combined with steering of these applications. All these cases require changes in access and operation of HPC infrastructures, including changes of allocation mechanisms.

III. Future Technologies and Architectures

A key technology for improving support of data-intensive HPC workloads are memory technologies. The size of the data sets typically results in a need for large memory capacity C_{mem}, while the speed at which data is processed, e.g. using modern GPUs, results in a need for high performance, e.g. large bandwidth B_{mem}. Different memory technologies differ significantly in terms of $\Delta\tau = C_{\mathrm{mem}}/B_{\mathrm{mem}}$, which indicates that high bandwidth comes with smaller capacity and vice versa. For high-performance memory technologies like HBM, which are used for current high-end GPUs, we observe $\Delta\tau \simeq 20 \ldots 40\,\mathrm{ms}$. For high-end SSD we have $\Delta\tau = \mathcal{O}(10^3\,\mathrm{s})$, i.e. an orders of magnitude larger value.

Despite the necessary compromises in terms of bandwidth, non-volatile memory devices like SSDs are interesting because of their high memory capacity within a small footprint and the much higher performance compared to spinning disks. An interesting open question: What is the best interface for accessing such non-volatile memory (NVM) devices? The options can be categorised as follows:

- *POSIX file system interface*: NVM devices are in the context of HPC most often used as a block device with a (near) POSIX compliant file system on top. The main advantage is the use of an interface that is still most popular within the relevant user communities. The disadvantage is that in most cases the capabilities of the underlying hardware cannot be fully exploited, e.g. the ability to perform a very large number of small write and read operations using random addresses. Furthermore, such a setup may suffer from limitations of POSIX due to slow metadata operations and consistency requirements.
- *Object store interface*: Object stores allow to overcome this limitation of POSIX as updates of a namespace are avoided. Individual objects are rather addressed through unique keys avoiding the need for updating a shared metadata structure. While object stores would in principle allow for a better exploitation of the underlying hardware capabilities, currently available solutions like Ceph still come with too much overhead.
- *Memory interface*: Due to the underlying technology being an addressable memory, using a memory interface would be a natural choice. Indeed, such solutions have been explored and compared to the other interfaces best use of the underlying hardware for the case of small, random transfers could be demonstrated. Lacking good solutions for managing this memory and the need for users to adapt their applications are currently probably the most important factors preventing wider uptake.

At Jülich Supercomputing Centre we used the JURON cluster for exploring some of the outlined interface options. The cluster comprises 18 IBM S822LC servers (also known as Minsky), each comprising 2 IBM POWER8 processors, 4 NVIDIA P100 GPUs, 1 HGST Ultrastar SN100 Series NVMe SSD and 1 Infiniband EDR card. We configured the system such that the SSDs were accessible as a shared storage resource, i.e. each node could access the SSD mounted at another node via the network. Two solutions providing such a setup have been tested: BeeGFS [8], a parallel file system, and Distributed Shared Storage (DSS) [9]. DSS is an interface developed by IBM Research, which allows accessing SSDs available within a RMDA-capable network using an RDMA CM managed communication. Benchmark results on this system are publicly available [9], [10]. We observed that for large transfer sizes of 4 MiByte both solutions allow to transfer at maximum bandwidth using 16 nodes, where maximum bandwidth is defined as 16 times the maximum bandwidth measured for the single devices using the ezFIO benchmark [11]. The latter slightly exceeds the specifications of the vendor. For small transfer sizes of 8 kiByte only using DSS the same performance level can be maintained, while for the parallel file system solution the effective bandwidth is at least $3\times$ below maximum performance. This is an indication that a memory interface towards the non-volatile memory is interesting for cases where small random read and write operations are the dominating access pattern.

In this context we would like to highlight that there are various emerging solutions for integrating non-volatile memory into the I/O architecture of modern supercomputers in such a way that the performance of the underlying memory technology can be efficiently exploited. One of the first solutions that reached product level was DDN's Infinite Memory Engine (IME), which is a kind of burst buffer [12]. (See, e.g., [13] for an early evaluation using brain simulations as a use case.) Another approach is taken in the SAGE project [14], which has in recent years been developing an hierarchical storage architecture based on Seagate's object store technology Mero. It's native support for hierarchical storage architectures allow for integration of fast storage technologies based on non-volatile memory. It furthermore provides a native object store interface designed for high performance. Due to the limited capabilities of the so far available prototype system, a performance evaluation for data-intensive HPC workloads would be premature.

IV. Fenix: Infrastructures for Extreme Scale Data Analytics

Recently, in the context of the Human Brain Project[3], a new initiative has been started to establish a federated infrastructure involving several European supercomputing centres, which is called Fenix and is initially realised by the ICEI project. This infrastructure is among others committed to be co-designed for data-intensive HPC workloads from brain research, like those described in section II.

The architecture is based on the concept that different sites provide similar class of compute and data services in a way that they can be federated and made accessible to the users as a single infrastructure. This means, e.g., that a common Authentication and Authorisation Infrastructure

[3] http://humanbrainproject.eu/

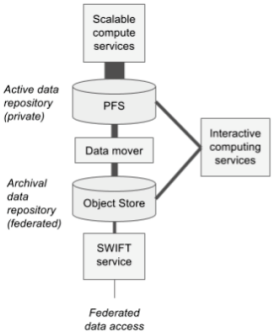


Fig. 3. Fenix storage architecture.

(AAI) is created. Among others the following services are foreseen:

- Scalable compute services
- Interactive compute services
- Virtual machine services
- Active data repositories
- Archival data repositories
- Data mover, location and transport services

To enable within the Fenix infrastructure both, sharing of large-scale data volumes as well as fast access to the data, two classes of data repositories are being introduced:

- *Archival Data Repositories*: Data stores optimised for capacity, reliability and availability, which is used for storing large data products permanently that cannot be easily regenerated. To facilitate federation of the Archival Data Repositories, these will all be accessible via a SWIFT interface.
- *Active Data Repositories*: Data repositories optimised for performance that are localised close to systems that consume or produce this data, e.g. HPC or visualisation systems. They are meant to be used for storing temporary slave replica of data objects. No specific interface for accessing these repositories are mandated as they are expected to be optimised for the system from which this data is being accessed. For HPC systems this will typically mean that a parallel file system is used.

In Fig. 3 we show a schematic overview of this architecture. To simplify staging of data from archival to active data repositories or migration of data from active to archival data repositories data mover services will be deployed that facilitate asynchronous data transport.

Scalable compute resources,which will become available within Fenix, will in parts be optimised for data analytics workloads. Furthermore, interactive compute services are becoming available such that users can access large-scale data volumes, e.g., for interactive data analytics steps.

V. Summary and Conclusions

Supercomputing centres are facing the need of providing resources to an increasing number of data-intensive HPC workloads and thus have to take this into account for the design of future I/O architectures and infrastructures suitable for extreme-scale data analytics. Realisation of such architectures and infrastructures need to take the key characteristics of data-intensive HPC workloads into account. In this contribution we outlined a few methodologies for this purpose.

New technologies, in particular in the area of memory, help us to address the challenges created by these data-intensive HPC workloads. While the best way of integrating these technologies into our future architectures remain to be explored, performance results are very encouraging.

Experience with emerging data-intensive HPC workloads have been used to design the initial architecture of the Fenix infrastructure. This infrastructure is being co-designed primarily with scientists from brain research. It is important to have guidance from the science domains that are supposed to benefit from this future infrastructure to exploit new technical opportunities efficiently.

Acknowledgements

Funding for the work is received from the European Commission Framework Programme FP7/2007-2013 under Grant Agreement No. 604102 (HBP) as well as the European Commission H2020 program under Specific Grant Agreement No. 720270 (HBP SGA1), Specific Grant Agreement No. 800858 (ICEI), and Grant Agreement No. 671500 (SAGE).

References

[1] K. Moreland, R. Oldfield, P. Marion, S. Jourdain, N. Podhorszki, V. Vishwanath, N. Fabian, C. Docan, M. Parashar, M. Hereld, M. E. Papka, and S. Klasky, "Examples of in transit visualization," in *Proceedings of the 2Nd International Workshop on Petascal Data Analytics: Challenges and Opportunities*, ser. PDAC '11. New York, NY, USA: ACM, 2011, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2110205.2110207

[2] K. Asanovi, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick, "The landscape of parallel computing research: A view from berkeley," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-183, Dec 2006. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html

[3] G. C. Fox, S. Jha, J. Qiu, and A. Luckow, "Towards an understanding of facets and exemplars of big data applications," in *Proceedings of the 20 Years of Beowulf Workshop on Honor of Thomas Sterling's 65th Birthday*, ser. Beowulf '14. New York, NY, USA: ACM, 2015, pp. 7–16. [Online]. Available: http://doi.acm.org/10.1145/2737909.2737912

[4] H. Spitzer, K. Amunts, S. Harmeling, and T. Dickscheid, "Parcellation of visual cortex on high-resolution histological brain sections using convolutional neural networks," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, April 2017, pp. 920–923.

[5] J. Lujan *et al.*, "APEX workflows," LANL, NERSC, SNL, Tech. Rep., 2016. [Online]. Available: https://www.nersc.gov/assets/apex-workflows-v2.pdf

[6] M.-O. Gewaltig and M. Diesmann, "NEST (NEural Simulation Tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.

[7] M. D. Wilkinson *et al.*, "The FAIR guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, p. 160018, Mar 2016. [Online]. Available: https://doi.org/10.1038/sdata.2016.18

[8] Fraunhofer-Institut ITWM, "BeeGFS, the parallel cluster file system." [Online]. Available: https://www.beegfs.io

[9] D. Pleiter *et al.*, "Assessment of "test series" progress (phase 3 of PCP) (evaluation in month 30) and definition of further process for the big data pre-exascale system"," Human Brain Project, Tech. Rep. Deliverable D7.7.7, 2017.

[10] A. Eekhoff, B. Tweddell, and D. Pleiter, "BeeGFS benchmarks on JURON," 2018. [Online]. Available: https://www.beegfs.io/docs/whitepapers/JURON_OpenPOWER_NVMe_by_ThinkParQ_FZ-Juelich.pdf

[11] E. F. Philhower, "ezFIO powerful, simple NVMe SSD benchmark tool," 2016. [Online]. Available: https://nvmexpress.org/ezfio-powerful-simple-nvme-ssd-benchmark-tool/

[12] N. Liu, J. Cope, P. Carns, C. Carothers, R. Ross, G. Grider, A. Crume, and C. Maltzahn, "On the role of burst buffers in leadership-class storage systems," in *Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on*, April 2012, pp. 1–11.

[13] W. Schenck, S. El Sayed, M. Foszczynski, W. Homberg, and D. Pleiter, "Early evaluation of the "Infinite Memory Engine" burst buffer solution," in *High Performance Computing*, M. Taufer, B. Mohr, and J. M. Kunkel, Eds. Cham: Springer International Publishing, 2016, pp. 604–615.

[14] S. Narasimhamurthy, N. Danilov, S. Wu, G. Umanesan, S. W. D. Chien, S. Rivas-Gomez, I. B. Peng, E. Laure, S. D. Witt, D. Pleiter, and S. Markidis, "The SAGE project: a storage centric approach for exascale computing: invited paper," in *Proceedings of the 15th ACM International Conference on Computing Frontiers, CF 2018, Ischia, Italy, May 08-10, 2018*, 2018, pp. 287–292. [Online]. Available: https://doi.org/10.1145/3203217.3205341

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Using the AiiDA-FLEUR package for all-electron ab initio electronic structure data generation and processing in materials science

Jens Broeder
Institute for Advanced Simulation
Forschungszentrum Jülich GmbH
Jülich, Germany
j.broeder@fz-juelich.de

Daniel Wortmann
Institute for Advanced Simulation,
Peter Grünberg Institute
Forschungszentrum Jülich GmbH
Jülich, Germany
d.wortmann@fz-juelich.de

Stefan Blügel
Institute for Advanced Simulation,
Peter Grünberg Institute
Forschungszentrum Jülich GmbH
Jülich, Germany
s.bluegel@fz-juelich.de

***Abstract*—Materials informatics tools [1] for tackling the data and computing challenges of materials design are rapidly evolving in the electronic structure community. Frameworks like AFLOW [2], AiiDA [3], ASE [4], Fireworks [5] and others are capable of managing thousands to millions of differently sized simulation jobs together with their heterogeneous data. If the computing infrastructure permits it, this is performed in a high-throughput fashion. Making high quality ab initio data and services from different data sources available inside and outside the scientific community is a challenge. Initiatives prepare for data sharing services scaling to petabyte [6]. While high-throughput studies [7] are not new in the electronic structure community, reference all-electron methods studies and data are rare. We present the open source AiiDA-FLEUR python package, enabling the management of many simulations with FLEUR [8], an all-electron code, through the Automated Interactive Infrastructure and Database for Computational Science (AiiDA) framework. AiiDA-FLEUR provides the user with FLEUR specific workflows, property calculators and tools to ease, and automatize everyday scientific work. Through AiiDA the connection to community data structure formats, databases, interactions with other community codes, and full provenance tracking of queryable curated data and logic is ensured.**

***Index Terms*—high-throughput; materials informatics; scientific workflows; materials science; ab initio; all-electron; electronic structure; density functional theory; AiiDA; FLAPW**

I. Introduction

Data challenges in materials science arise not in a local single large petabyte data producing facility, but in a large distributed community of terabyte producers with heterogenous data, data quality, services and sharing. In ab initio computational materials science the amount of data accumulates due to the number of calculations needed to accomplish a certain task. As the material properties one needs to simulate arise from the underlying QMA-complete [9] problem of solving the Schroedinger equation, two different aspects are of particular interest. On the one hand, the structural configuration space itself is enormous [10] and hence screening type tasks through material property space are fit for high-throughput computing (HTC). On the other hand, simulations of large systems with many atoms require computational effort at the frontiers of high-performance computing (HPC). Both of these aspects lead to specific challenges for the simulation codes used as well as for the job and data handling framework. At the core of future research will be the tools created to manage simulations, workflows, data processing and to ensure data to be reproducible, searchable, reliable, shareable, curated and provenance tracked along the principles of FAIR [11] and the open provenance model [12].

While many methods in materials science rely on experimental or empirical parameters, ab initio methods aim at a description of materials properties from first principles. Hence, they are of particular value in materials design by the direct calculation of properties of interest and by providing input for other applications e.g. for multi scale models. A sufficiently large database of high quality ab initio data can accelerate materials design, through machine learning and other tools. The most successful of these methods is Density Functional Theory (DFT) in its various incarnations and numerical implementations. In this paper we will focus on the FLEUR code, an implementation of the all-electron full-potential linearized augmented plane waves (FLAPW) method [13] known for its universality and high accuracy.

Among the tool sets used in computational materials design in the field of material informatics the AiiDA framework stands out as a dedicated tool for DFT calculations with interfaces to a wide variety of codes and a strong focus on data provenance. In this paper we will present and discuss the AiiDA-FLEUR package that allows to deploy the FLEUR code productively through the AiiDA framework.

II. The AiiDA-FLEUR package

The AiiDA framework features a flexible design in which a plug-in system enables the use of different DFT codes. Such a plug-in must take care of the basic tasks of providing interfaces translating the data structures, of handling the calculation

We acknowledge partial support from the EU Centre of Excellence "MaX Materials Design at the Exascale" (Grant No. 676598).

setup and resources, of parsing results, and of enabling the construction of code specific workflows.

The open source AiiDA-FLEUR python package[1] provides such plug-ins and utility for the FLEUR [8] code. The AiiDA-FLEUR package contains AiiDA plug-ins for FLEUR itself, its input generator (inpgen) and a data structure representing the FLEUR input. Further, it contains workflows, property calculator protocols and utility to create a high-level work environment. The package is open source under MIT license, released on github and the python package index (PyPI). The visualization function plot_fleur allows for quick default visualizations of any database node(s) produced by FLEUR calculations or workflows.

A. Calculation plug-ins

AiiDA calculation plug-ins, as fundamental building blocks, contain instructions how to create valid input from information in the database and what information to parse from output files and store in the database. For FLEUR we have implemented two calculation plug-ins. The input for an input generator calculation plug-in (FleurinputgenCalculation) consists of up to three database nodes. A crystal structure is provided via an AiiDA StructureData node. The executable of the code is known through a Code node. Further FLAPW parameters can be specified optionally in an additional ParameterData node. An input generator calculation returns a FleurinpData node representing the input files for a FLEUR calculation. Other calculation output nodes represent files in the repository (Folder) or on a remote machine (RemoteData). Fig. 1 shows a node graph of a FleurCalculation. Input nodes are a Code, a FleurinpData node plus an optional RemoteData node from a previous parent FleurCalculation to continue from the its output results. In the output node of a FleurCalculation basic calculation results are stored, for example the total energy, Fermi energy, band gap, charge distance and meta data information of the run.

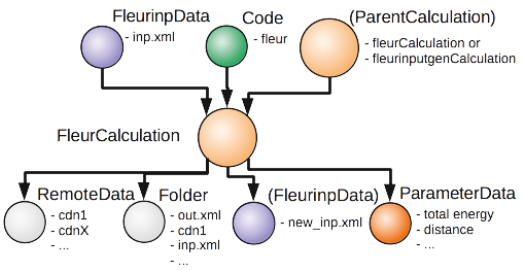


Fig. 1. Database input and output nodes in the directed acyclic provenance graph for a single run of the FLEUR code.

[1]Code: https://github.com/JuDFTteam/aiida-fleur, Documentation: https://aiida-fleur.readthedocs.io

B. Data structure plug-in

As a typical FLEUR calculation needs a significant amount of additional parameters beyond the crystal structure represented in AiiDA plus functionality to efficiently manage and manipulate these, we used the possibility to extend AiiDA by new data structures [3]. We implemented a new data structure, FleurinpData, to represent the FLEUR input files and to provide user friendly methods for processing input or extracting information from it. The input files are stored in the file repository while in the database a footprint of the full inp.xml file is stored. The FleurinpModifier class ensures that provenance is kept through all input modifications and allows for previews and undo of changes.

C. Workflows

A powerful feature of the AiiDA framework is the ability to develop, run and share workflows [3]. AiiDA workflows are a way to automatically launch time consuming calculations that logically depend on each other without the user having to wait for each of them. The workflow developer encodes expert knowledge and ensures the provenance of data and logic while having access to the python universe. Workflows are powerful property calculator protocols with complex series of calculations able to be launched with a small snippet of python code. In workflows additional logic can be encoded like how to best run and converge calculations, fining reasonable parameter sets, determine optimal computing resources, automatic error treatment and restarts.

AiiDA-FLEUR comes with a set of workflows. The basic ones converge a FLEUR calculation, calculate a density of states, electronic band structure or an equation of state. AiiDA-FLEUR contains additional workflows to manage core-hole simulations and calculate core-level electron energy shifts. Workflows to perform structure relaxation or calculate magnetic properties are under development. A typical run of the basic FLEUR self-consistent field workflow creates about 20 database nodes and around 10 files of different size to be long-term stored. Advanced workflows spawn a few to hundreds of self-consistent field subworkflows.

A rather naive high-throughput example of a python launch code piece is shown in Fig. 2 as demonstration. Beforehand we have imported all structures (more than 800000 entries) from the OQMD [14] into an AiiDA database. Then for each structure we prepared a node with some specific FLAPW parameters we like to adjust beyond the FLEUR defaults. The python code snippet would load the structures and their parameter nodes from their two groups in our database and launch a self-consistent field workflow for each of them. The launched workflow could be interchanged with any other workflow with a similar interface. Further we have to specify the code and the machine to run on plus optionally specify some maximum resources per job among other options. While this code piece will run quickly through in minutes to hours, it will command the AiiDA daemon to manage all these workflows resulting in over 1.6 million jobs to be calculated. For an infrastructure with a throughput of 2000 jobs per day

this would take well over 2 years to complete. It is obvious that this naive demonstrative example will probably result in a very high failure rate. A realistic high-throughput project has to be handled more carefully and more verbosely while slowly scaling up if the error rate of the infrastructure is sufficiently low. Also splitting the project in smaller similar parts, predicting and controlling the work load and understanding if the quantum engine together with the workflows are robust enough for your project is necessary.

```
from aiida.orm import WorkflowFactory,load_group,Code
from aiida.work.launch import submit
fleur_scf = WorkflowFactory('fleur.scf')

inpgen = Code.get_from_string('inpger@otherhost')
fleur = Code.get_from_string('fleur@cluster')

strucs = load_group(label='oqmd_strucs').nodes.dbnodes
paras = load_group(label='oqmd_paras').nodes.dbnodes

for i,struc in enumerate(strucs):
    res = submit(fleur_scf, structure=struc,
                 calc_parameter=paras[i],
                 fleur=fleur, inpgen=inpgen)
```

Fig. 2. Minimal Python code to launch certain FLEUR workflows for a set of crystal structures. This naive code example spawns a self-consistent field workflow for each structure in the Open Quantum Materials Database (OQMD) resulting in over 1.6 million diverse jobs to be managed.

Besides the robustness of the underlying quantum engine the robustness of the workflow is important. The log-log plot in Fig. 3 shows the convergence result of the charge density and the total energy for over 1700 different bulk binary crystal structures (from the Materials Project [15] and Inorganic Crystal Structure Database (ICSD) [16]) run with the self-consistent field AiiDA workflow for FLEUR with spin orbit coupling. For over 86 % of the systems the workflow managed to achieve convergence in charge density and total energy. While 7 % did not converge at all for different reasons and another 7 % converged partially. Magnetic systems (red) are harder to converge then non-magnetic (blue) systems.

Code interoperability allows to exploit individual strengths of different electronic structure methods. It is convenient to chain different workflows and different electronic structure calculations through the reuse of common AiiDA data structures in various databases, workflows, plug-ins for electronic structure codes and utilities. We designed all FLEUR specific workflows to have a similar interface. For example one can import a crystal structure through AiiDA from any common database source, or file format. On this crystal structure one could run a structure relaxation workflow with FLEUR or any other code. The output structure can again become an input for any further workflow one wants to run. This interoperability of FLEUR specific workflows is shown in Fig. 4. Also each workflow comes with its quick default visualization. A single plot function allows for a visualization of any single database node or of a list of nodes.

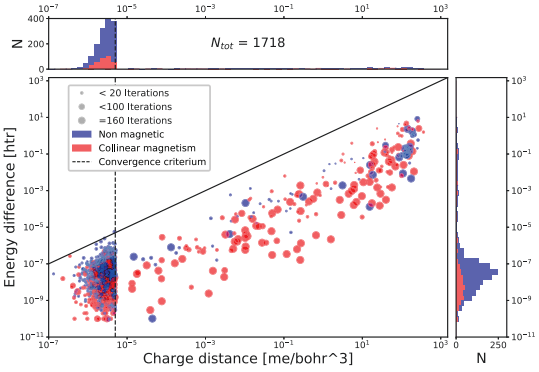


Fig. 3. Small example of the charge density and total energy convergence result of 1718 different binary crystal systems managed by the FLEUR self-consistent field workflow. The workflow succeed in over 86 % of the simulations to fully converge the systems.

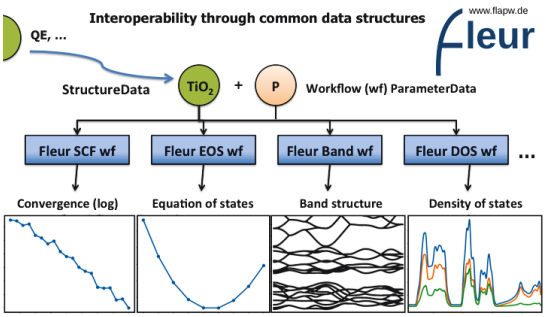


Fig. 4. Common AiiDA data structures (i.e StructureData) enable chaining of different electronic structure codes (example Quantum Espresso (QE), ...) and workflows profiting from their individual strengths. Also each AiiDA data structure and FLEUR specific workflow comes with its own quick default visualizations.

III. Data handling

AiiDA tracks the data and logic provenance in form of a directed acyclic graph in a queryable database. Files for long term storage are stored in a file repository or object store [3]. When running many complex workflows or a material screening task one ends up with millions of files on disk and databases with easily tens of millions of nodes. A database with one million nodes is about three gigabyte and more in size. To get an impression in Fig. 5 the full database provenance graph/network of a small AiiDA database is shown. The graph depicts about 4000 self-consistent field workflows with different codes, versions and computing resources, resulting in about 130000 nodes (black dots). The graph is layouted with a parallel multi force atlas graph layout algorithm using Gephi [17]. Clusters of nodes evolve around different highly connected FLEUR code nodes on divers computing

resources. Crystallographic Information File (CIF) data nodes from which crystal structures have been extracted dangle loosely connected around the edges.

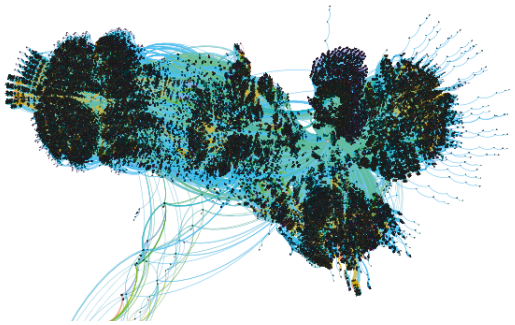


Fig. 5. A directed acyclic provenance graph of a small SQL database containing over 4000 self-consistent field cycles of different codes resulting in around 130000 nodes (black dots) to provide a brief impression on complexity and scalability. (Produced with Gephi [17], Multi force-directed graph layout)

A. Data services

Besides the predictive power of `ab initio` results themselves, access to `ab initio` data is of interest to other experiments, other communities and industry [18]. Every use case may have different demands, expectations, data quality, or accuracy requirements. Example one: For the evaluation or comparison of a certain experiment the user is only interested in a small subset of physical quantities, their accuracy, and some meta data information. Example two: A larger subset of data and meta data is required for training a machine learning application [19]. Not only successes but also in some respect failures are of interest for a good training set. For the first example a notebook or app with a very specialized query and data access tailored to the specific use case of the community is a good solution. Whereas for the second example a general access interface for the data might suffice. Therefore, data hosting platforms are needed which allow for deployment of special apps and services on data sets. A beginning along these lines is seen on Materialscloud [20], the Materials Project [15], AFLOWlib [21], OQMD [14], the NOMAD analytic toolkit [22] and EPS [23]. Given that structure configuration space is so vast, we can only accumulate data on a small fraction of it leading to the need of on demand computational services through the deployment of robust automated workflows.

IV. Conclusion

We have presented the AiiDA-FLEUR package, providing an automated high-level work environment for users of the FLEUR code. This is accomplished through specific implementations of workflows which can be run and managed thousand- to millionfold with the AiiDA framework. All data is provenance tracked in a directed acyclic graph along the open provenance model [12]. We pointed out that in computational materials science we have data challenges due to distributed heterogenous data producers with a wide range of possible applications. Materials informatics is evolving to progress on these challenges. Tailored data services, apps and on demand property calculations may be essential to provide data access to communities with certain use cases.

Acknowledgment

We acknowledge the open source community working on materials informatics, in detail the AiiDA team for their support, work, and cooperation. This work has been obtained as part of the research requirements of a PhD degree to be awarded by the RWTH Aachen University.

References

[1] L. Ward and C. Wolverton, “Atomistic calculations and materials informatics: A review,” Current Opinion in Solid State and Materials Science, 21, 3, Pages 167-176, 2017

[2] S. Curtarolo et al., “AFLOW: An automatic framework for high-throughput materials discovery,” Comput. Mater. Sci. 58, 2012,218-226, ISSN 0927-0256, https://doi.org/10.1016/j.commatsci.2012.02.005.

[3] G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, “AiiDA: automated interactive infrastructure and database for computational science,” Comput. Mater. Sci. 111, 218-230 (2016)

[4] A. H. Larsen et al., “The Atomic Simulation EnvironmentA Python library for working with atoms,”, J. Phys.: Condens. Matter Vol. 29 273002, 2017

[5] A. Jain et al., “FireWorks: a dynamic workflow system designed for highthroughput applications.” Concurrency Computat.: Pract. Exper., 27: 50375059. (2015) doi: 10.1002/cpe.3505.

[6] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, and I. Foster.“The Materials Data Facility: Data services to advance materials science research.” JOM 68, no. 8 (2016): 2045-2052.

[7] S. Curtarolo, G.L.W. Hart, M.B. Nardelli, N. Mingo, S. Sanvito and O. Levy, “The high-throughput highway to computational materials design,” Nat. Mater. 12, 191201, 2013

[8] http://www.flapw.de

[9] N. Schuch, and F. Verstraete, “Computational complexity of interacting electrons and fundamental limitations of density functional theory,” Nat. Phys. 5, pages 732735, 2009

[10] O.A. von Lilienfeld, “First principles view on chemical compound space: Gaining rigorous atomistic control of molecular properties.”, Int. J. Quantum Chem., 113: 1676-1689. (2013) doi:10.1002/qua.24375

[11] M. D. Wilkinson et al., “The FAIR Guiding Principles for scientific data management and stewardship.”, Sci. Data 3:160018 doi: 10.1038/sdata.2016.18 (2016).

[12] L. Moreau et al., “The Open Provenance Model core specification (v1.1)”, Future Gener. Comput. Syst., 27 (2011), pp. 743-756

[13] M. Weinert, E. Wimmer, and A.J. Freeman, “Total-energy all-electron density functional method for bulk solids and surfaces,” Phys. Rev. B 26, 4571 (1982)

[14] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, “Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD)”, JOM 65, 1501-1509, 2013.

[15] A. Jain*, S.P. Ong* et al. (*=equal contributions) “The Materials Project: A materials genome approach to accelerating materials innovation”, APL Materials, 1(1), 011002, 2013

[16] A. Belsky, M. Hellenbrandt, V. L. Karen and P. Luksch, “New developments in the Inorganic Crystal Structure Database (ICSD): accessibility in support of materials research and design”, (2002). Acta Cryst. B58, 364-369.

[17] M. Bastian, S. Heymann and M. Jacomy, “Gephi : An Open Source Software for Exploring and Manipulating Networks”, International AAAI ICWSM 2009, http://gephi.org

[18] Kristian S. Thygesen and Karsten W. Jacobsen, “Making the most of materials computations”, Science, 354, 6309, 180–181, 2016, doi:10.1126/science.aah477

[19] B. Meredig and A. Agrawal et al. “Combinatorial screening for new materials in unconstrained composition space with machine learning”, Phys. Rev. B 89, 094104 (2014)
[20] http://www.materialscloud.org
[21] S. Curtarolo et al., “AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations” Comput. Mater. Sci. 58, 227-235, 2012
[22] https://nomad-coe.eu
[23] C. Ortiz, O. Eriksson and M. Klintenberg. “Data mining and accelerated electronic structure theory as a tool in the search for new functional materials”, Comput. Mater. Sci. 44, 1042-1049 (2009).

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Flexible tool development for climate data applications: A compression framework

Ugur Cayoglu[†*], Jörg Meyer[*], Tobias Kerzenmacher[†], Peter Braesicke[†], and Achim Streit[*]

[*]Steinbuch Centre for Computing
Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
Email: {Ugur.Cayoglu, Joerg.Meyer2, Achim.Streit}@kit.edu

[†]Institute of Meteorology and Climate Research - Atmospheric Trace Gases and Remote Sensing
Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
Email: {Peter.Braesicke, Tobias.Kerzenmacher}@kit.edu

***Abstract*—Among the scientific communities that generate the largest amount of data today are the climate sciences. New climate models enable model integrations at unprecedented resolution and can simulate centuries of climate change that include many complex interactions in the Earth system. Previously, the numerical integrations used to be the bottleneck. Nowadays, limited storage space and data analysis are becoming the main bottlenecks, because of ever increasing simulation output data. To tackle this challenge, we propose a prediction-based lossless compression framework.**

The framework supports: (1) the creation of individual predictors, which can be adjusted to the available data, (2) strict interfaces and customisable components, which are building blocks of the compression modules that are optimised for particular applications as well as (3) the execution of benchmarks and validity tests for sequential and parallel processing of compression algorithms.

***Index Terms*—Compression, floating-point, finite-context.**

I. Introduction

Through the introduction of next-generation models the climate sciences have experienced a breakthrough in high-resolution simulations, which calculate global simulations with a resolution of five kilometres (e.g. ICON-ART [1]). The new models produce an unprecedented volume of data in climate research, so that future studies are limited by the storage capacity rather than numerical calculations.

These models are validated with, for example, reanalysis datasets. One of them, the current **E**uropean **Re****A**nalysis (ERA5) dataset outputs hourly data starting from 1979 to the present on a 1440×721 (about 31 km) horizontal and 137 level vertical (up to 0.01 hPa = 80 km) grid[1]. If we assume 16-Bit Integer values for each variable this amounts to 2.26 TiB p.a. per variable with support for 120 variables[2]. One way to tackle the storage problem is to use compression [2].

We propose a modular lossless compression framework (LSCF) for the development of customized prediction-based compression algorithms for structured spatio-temporal data.

The framework helps with the development of a prediction-based compression method by providing a strictly defined interface, concurrent compression support for fast testing, implementation of already established prediction models, the possibility to generate ensemble predictors, and fast iteration via multi-dimensional subsetting of datasets.

In the next section we will give a brief overview of prediction-based compression. Afterwards we will introduce LSCF and take a closer look at the implementation. In the concluding section, we will outline how the community can contribute to the framework and give recommendations for future work.

II. Prediction-based compression

Compression algorithms can be classified in two categories: lossless and lossy compression. A lossless compression algorithm creates a reconstruction that is identical to the source data on bit level. A lossy compression algorithm is not able to do this. The reconstruction generated by the lossy algorithm is an approximation of the source data.

Both types of algorithms work by first decorrelating and then encoding the data. A lossy compression algorithm has additionally an approximation step in between. The correlation steps reduces redundancy in the data being it autocorrelated or cross correlated information. The approximation step reduces the complexity of the data by using e.g. methods of quantized representation of data values. In the encoding step the actual compression happens and the data is written on disk in a compact form.

The last couple of years the development of compression algorithms for scientific data experienced a renaissance [3]–[8]. Although these methods are based on the same principle of prediction-based compression, there is currently no easy way to test and adapt them to different datasets.

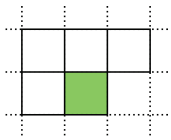


Fig. 1. Stencil of neighbouring data points being used by the predictor. The green data point is being predicted by the predictor using information from the encircled neighbouring data points

[1]European Centre for Medium-Range Weather Forecasts (ECMWF) Newsletter No. 147 – Spring 2016 (p.7)

[2]While some of these variables are simulated, others can be deduced from simulated variables. For reference http://apps.ecmwf.int/codes/grib/param-db

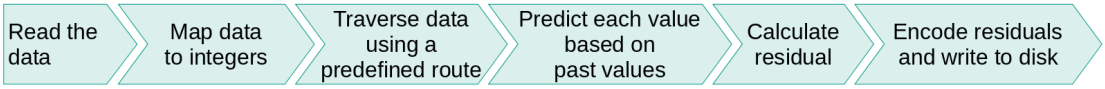


Fig. 2. Steps of a prediction-based compression algorithm.

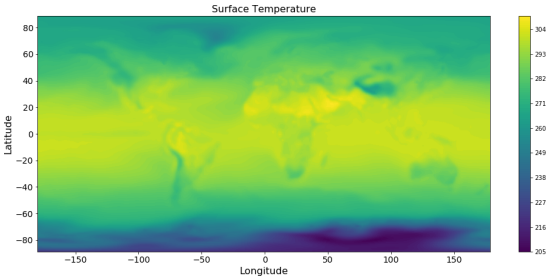


Fig. 3. Temperature as an example why to use prediction-based compression in climate research.

A prediction-based compression algorithm consists of the following steps: First the data is being read into memory. In case the data is floating-point data they are mapped to integers to avoid floating-point operations. Floating-point operations might cause numerical inaccuracies and prevent successful reconstruction of the compressed data. This concludes the preprocessing steps and the decorrelation starts. In the next steps a traversal sequence through the spatio-temporal data is chosen so that each point is visited once. The predictor then moves along this traversal path and uses the points which have been already visited in the past to predict the new data. Each predictor defines a stencil like the one in Figure 1 to chose which points to use. Finally the residual between predicted and the true values is calculated.

The better the prediction is, the more leading zeros has the residual. This is called the leading zero count (LZC) of the data. If lossless compression is used, the LZC and the residual are encoded and stored on disk. If lossy compression is used, the deviation from the true value is compared with an error tolerance determined in advance and a decision is made whether or not the residual should be stored [2]. The true value can then be reconstructed using the prediction model and residual. These steps are depicted in Figure 2.

Figure 3 gives an illustrative example for why prediction-based compression can be successfully used with climate data. The figure depicts surface temperature across the globe. While on global level there might be spots where neighbouring values are quite different (especially mountainous regions like the Himalayas or coasts) most of the data has a smooth gradient. This is especially true for the oceans and the equator. Prediction-based compression uses these regularities to improve the prediction and with this the compression rate.

In the next section we will introduce LSCF and explain how each step of the prediction-based compression algorithm is mapped to its components.

III. Framework

In this section we will first describe the two core components of LSCF. Afterwards we will present several features of LSCF which help the scientist during the design and validation process of a compression algorithms. For an in-depth explanation of the framework please refer to [2].

A. Core components

LSCF has two core components: objects and modifiers (Fig. 4).

a) Objects: Objects represent the current state of the data to be compressed. Each result of the steps given in Fig. 2 is represented by an object. They are immutable and may include metadata information about previous states. There are three main object classes: array objects, data objects and unique objects. The array objects are `floatarray`,`integerarray`, `predictionarray` and `residualarray`. The data objects are the input and output files. The unique objects classify objects which do not share similarities with any other object. These are the `sequence object` and the `coded object`. An overview of these objects are depicted in Fig. 4.

b) Modifier: Modifiers execute the steps of the compression algorithm depicted in Fig. 2. There are five types of modifiers [2]:

- **Mapper** *Mapping floating-point values to integers*
- **Sequencer** *Transforms an array into a data stream*
- **Predictor** *Predicts next datum on the data stream, based on past values*
- **Subtractor** *Calculates the residual between prediction and true value*
- **Encoder** *Prepares residuals to be written on disk*

They operate on objects and generate new objects. Each modifier is only allowed to operate on a specific kind of object (see Table I) and has a strict interface (see Table II. These properties are guarantee interoperability and modularity of the framework. A list of possible modifiers provided by LSCF is given in Fig. 5.

B. Additional features

The framework provides several additional methods for the design of a compression algorithm:

- Ensemble predictors
- Quality assessment
- Parallel processing
- Multidimensional random subsetting
- Strict interface

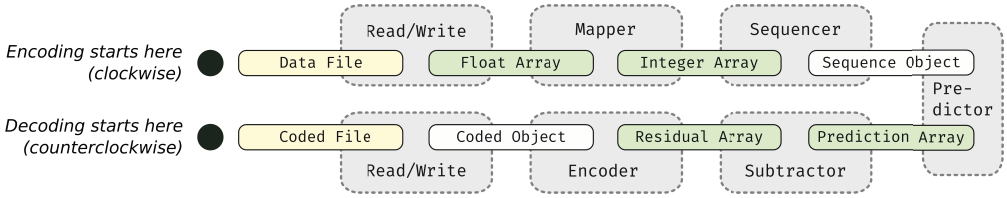


Fig. 4. Implementation of the steps described in Fig. 2. The modifier are depicted with dotted lines and the objects with through lines. The colouring of the objects emphasizes their class: yellow for file objects, green for array object and white for unique objects.

TABLE I
INPUT UND OUTPUT OBJECTS OF THE MODIFIERS

modifier	input	output
mapper	`floatarray`	`integerarray`
sequencer	`integerarray`	`sequence object`
predictor	`integerarray` `sequence object`	`predictionarray`
subtractor	`integerarray` `predictionarray`	`residualarray`
encoder	`residualarray`	`encoded object`

TABLE II
INTERFACE OF THE MODIFIERS

modifier	function	inv. function
mapper	`map()`	`rev_map()`
sequencer	`flatten()`	–
predictor	`predict()`	–
subtractor	`subtract()`	–
encoder	`encode()`	`decode()`

1) Ensemble predictors: An ensemble predictor is defined by a group of predictors, a cost function and a consolidation method. The predictors are run in parallel during compression. The cost function determines the rank of these predictors. The consolidation method defines how each single prediction from the group members should be consolidated and merged into a single prediction. The goal of using an ensemble predictor is to combine the knowledge of several predictors and generate a superior prediction.

2) Quality assessment: The quality assessment provides information about the achievable compression rate of the dataset. The framework calculates the information theoretical lower bound of the dataset provided by the Shannon Entropy (SE) [9]. The SE quantifies the average amount of information represented by a random datum of the dataset. The SE, denoted $H(X)$, is defined by

$$H(X) = -\sum_i P(x_i) \cdot \log_b P(x_i)$$

with $X = \{x_0, x_1, \ldots, x_n - 1, x_n\}$ representing all possible values of the dataset, P the probability mass function and b the base of the logarithm. Since we are interested in the information content in bits, we will use $b = 2$.

3) Parallel processing: LSCF provides further the possibility for parallel processing of compression algorithms. There are two possible ways of parallel processing: The data can be chunked into several blocks and compressed in parallel using a single compression algorithm or several compression algorithms are run in parallel to compress a single dataset.

4) Multidimensional random subsetting: The framework supports random subsetting of multidimensional data. The subsetting is defined by its size, error margin and dimension constraints. This feature is necessary for parameter fine-tuning.

5) Strict interface: The interoperability between the modifiers and objects can only be guaranteed if the interfaces are standardized. These interfaces are the function calls of the modifiers and attributes of the objects. The interfaces to the modifiers are given in Table II and the input and output objects are given in Table I.

C. Implementation

An implementation of the framework is available at [10]. The provided framework is implemented in Python 3 and uses as backend modules `scipy` [11], `pandas` [12] and `xarray` [13]. It has been tested with files in NetCDF format with Climate and Forecast Metadata Conventions. The use of established open source software provides a good basis for future co-operations and possible extensions of the framework.

IV. SUMMARY

Higher resolutioned simulation output provides a more accurate representation of the simulation run. This in turn depicts a better picture of the underlying model and enables more fine-granular model improvements than before. Our proposed lossless compression framework (LSCF) is a first step in this direction and provides the necessary tools for the development of a compression algorithm. LSCF provides an easy understanding of prediction-based compression through the usage of a modular architecture and supports concurrent testing and rapid development of custom methods. The usage of strictly defined interfaces provides a reusable framework and a clearly defined structure for future additions.

We hope the open source nature of the framework helps us to gather an active user base and enable collaboration in the field of compression for scientific datasets.

CODE AVAILABILITY

An implementation of the framework described above will be made available under GNU GPLv3 license at [10].

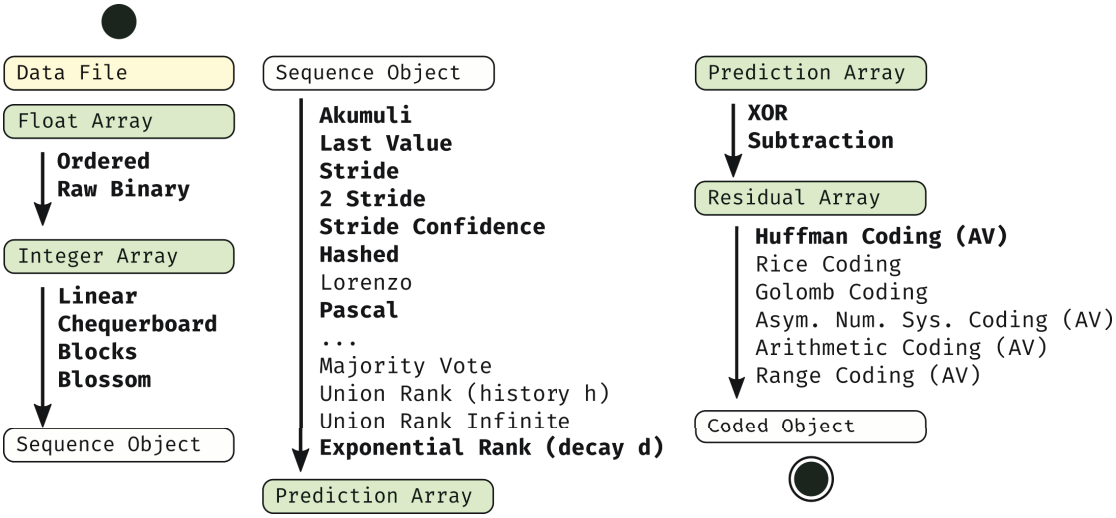


Fig. 5. Diagram of possible modifiers in the design of a prediction-based compression algorithms with LSCF. The label on the arrows define transitions applied to the previous object. Emphasised are the transitions which are implemented and part of the framework. The colouring of the objects emphasise the similarity of the states. Data objects are yellow, array objects are green and unique objects are white [2].

REFERENCES

[1] J. Schröter, D. Rieger, C. Stassen, H. Vogel, M. Weimer, S. Werchner, J. Förstner, F. Prill, D. Reinert, G. Zängl, M. Giorgetta, R. Ruhnke, B. Vogel, and P. Braesicke, "ICON-ART 2.1 – A flexible tracer framework and its application for composition studies in numerical weather forecasting and climate simulations," *Geoscientific Model Development Discussions*, vol. 2018, pp. 1–37, 2018. [Online]. Available: https://www.geosci-model-dev-discuss.net/gmd-2017-286/

[2] U. Cayoglu, J. Schröter, J. Meyer, A. Streit, and P. Braesicke, "A Modular Software Framework for Compression of Structured Climate Data," in *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '18)*, 2018. [Online]. Available: https://doi.org/10.1145/3274895.3274897

[3] S. Liu, X. Huang, Y. Ni, H. Fu, and G. Yang, "A High Performance Compression Method for Climate Data," in *2014 IEEE Int. Symp. Parallel Distrib. Process. with Appl.* IEEE, aug 2014, pp. 68–77. [Online]. Available: http://ieeexplore.ieee.org/document/6924431/

[4] X. Huang, Y. Ni, D. Chen, S. Liu, H. Fu, and G. Yang, "Czip: A Fast Lossless Compression Algorithm for Climate Data," *Int. J. Parallel Program.*, vol. 44, no. 6, pp. 1248–1267, dec 2016. [Online]. Available: http://link.springer.com/10.1007/s10766-016-0403-z

[5] A. H. Baker, D. M. Hammerling, S. A. Mickleson, H. Xu, M. B. Stolpe, P. Naveau, B. Sanderson, I. Ebert-Uphoff, S. Samarasinghe, F. De Simone, F. Carbone, C. N. Gencarelli, J. M. Dennis, J. E. Kay, and P. Lindstrom, "Evaluating Lossy Data Compression on Climate Simulation Data within a Large Ensemble," *Geosci. Model Dev. Discuss.*, no. July, pp. 1–38, jul 2016. [Online]. Available: http://www.geosci-model-dev-discuss.net/gmd-2016-146/

[6] U. Cayoglu, P. Braesicke, T. Kerzenmacher, J. Meyer, and A. Streit, "Adaptive Lossy Compression of Complex Environmental Indices Using Seasonal Auto-Regressive Integrated Moving Average Models," in *2017 IEEE 13th Int. Conf. e-Science*. IEEE, oct 2017, pp. 315–324. [Online]. Available: http://ieeexplore.ieee.org/document/8109150/

[7] P. Lindstrom and M. Isenburg, "Fast and Efficient Compression of Floating-Point Data," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 1245–1250, sep 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4015488

[8] P. Ratanaworabhan, J. Ke, and M. Burtscher, "Fast Lossless Compression of Scientific Floating-Point Data," in *Data Compression Conf.*, no. August. IEEE, 2006, pp. 133–142. [Online]. Available: http://ieeexplore.ieee.org/document/1607248/

[9] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, jul 1948. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773024

[10] U. Cayoglu, "Prediction-based Compression Framework," https://github.com/ucyo/cframework, 2018, [Online; accessed 27-May-2018].

[11] T. E. Oliphant, "Python for scientific computing," *Computing in Science Engineering*, vol. 9, no. 3, pp. 10–20, May 2007.

[12] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[13] S. Hoyer and J. Hamman, "xarray: N-D labeled arrays and datasets in Python," *Journal of Open Research Software*, vol. 5, no. 1, 2017. [Online]. Available: http://doi.org/10.5334/jors.148

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Towards big data-enabled terrestrial systems modeling at HPSC TerrSys

Klaus Goergen
Agrosphere (IBG-3)
Research Centre Jülich
52425 Jülich, Germany
k.goergen@fz-juelich.de

Slavko Brdar
Jülich Supercomputing Centre
Research Centre Jülich
52425 Jülich, Germany
s.brdar@fz-juelich.de

Carina Furusho-Percot
Agrosphere (IBG-3)
Research Centre Jülich
52425 Jülich, Germany
c.furusho@fz-juelich.de

Ketan B. Kulkarni
Altair Engineering GmbH
71034 Böblingen, Germany
kulkarni@altair.de

Bibi Naz
Agrosphere (IBG-3)
Research Centre Jülich
52425 Jülich, Germany
b.naz@fz-juelich.de

Jan Vanderborght
Agrosphere (IBG-3)
Research Centre Jülich
52425 Jülich, Germany
j.vanderborght@fz-juelich.de

Harrie-Jan Hendricks-Franssen
Agrosphere (IBG-3)
Research Centre Jülich
52425 Jülich, Germany
h.hendricks-franssen@fz-juelich.de

Stefan Kollet
Agrosphere (IBG-3)
Research Centre Jülich
52425 Jülich, Germany
s.kollet@fz-juelich.de

***Abstract*—This manuscript to the proceedings of the Extreme Data – Demands, Technologies, and Services – Workshop[1], gives an overview on the characteristics, components, steps and methods of a complex data flow path around a fully coupled regional Earth system model and highlights the challenges we face working with very large data, as well as the concepts and strategies towards a big data-enabled modeling chain.**

***Index Terms*—Big Data, HPC, Geoscience, Integrated Model, Terrestrial Systems Modeling Platform, TSMP, Terrestrial Water Cycle**

I. Introduction

Enabled by the steady increase of computational capacity through HPC developments towards massively parallel, increasingly heterogeneous supercomputers [1], Earth system modeling (ESM), including terrestrial systems modeling, is used to understand processes and feedbacks between the compartments of the geo-ecosystem, that are impacted by global environmental change, climate change and anthropogenic use of ecosystem services.

ESM in general is currently characterized by (i) a resolution increase to convection permitting simulations below 4 km grid spacing [2], (ii) multiphysics, fully coupled (regional) model systems [3], (iii) enlarged, high-resolution model domains [4], and (iv) long integration times and/or many ensemble members in either climate change [5] or (v) data assimilation experiments [6].

A consequence are unprecedented data volumes that have been a point of concern in [7] already. In a more recent opinion paper on the future of climate system modeling, [8] calls for a Flagship European Programme on Extreme Computing and Climate, which promotes exascale climate modeling at 1 km global resolution and also emphasizes the need for dedicated tools and strategies to cope with the associated big data volumes. A demonstrator of the feasibility of such simulations using the MPAS model is [9]; continental to global hydrology and land surface models are also advanced towards hyper-resolution [10]; [11] show, e.g., the added value of such simulations for water resources modeling.

The Centre for High-Performance Scientific Computing in Terrestrial Systems[2] (HPSC TerrSys) ultimately wants to provide predictions (including uncertainty estimates) of the hydrologic, energy, and biogeochemical cycles of the terrestrial system at scales that are relevant for science, stakeholders, and society (i.e., neighborhoods to continents). As terrestrial systems exhibit heterogeneity and non-linear exchange processes at all scales, high resolution models over large space and time scales are required. This is why HPSC TerrSys takes part in the above developments, that confront us with big data challenges along the complete modeling chain.

Included under the term "modeling" is the complete data flow path or data life cycle, from data acquisition (e.g., as model input data), over pre-processing, the model simulation itself, post-processing, analysis, visualization, storage, archival to dissemination. In this manuscript we present the status and plans from a purely user driven perspective within HPSC TerrSys to make our modeling chain big data-capable using off-the-shelf technical solutions. We do not report on the big data capabilities of the model systems themselves, e.g., to simulate larger problem sizes by using accelerators, novel solvers, or Deep or Machine Learning techniques.

II. Origin of Big Data Challenge

HPSC TerrSys' big data challenge is driven to a lesser extend by "data variety" (e.g., through integration of sensor networks) or "data velocity" (e.g., incorporating near real-time measurements in simulations), but rather the increase in – primarily numerical model – "data volumes". The data volumes, e.g., from observational data, used for data validation

[1]Held at Jülich Supercomputing Centre in Germany on 18 and 19 September 2018

[2]http://www.hpsc-terrsys.de/hpsc-terrsys/EN/

or in data assimilation, also steadily increase, but are not covered here.

A. Terrestrial Systems Modelling Platform, TSMP

One of the model systems most extensively used within HPSC TerrSys is the massively parallel, scale-consistent, fully coupled multiphysics Terrestrial Systems Modelling Platform (TSMP) [3], [12], a modular multiple program, multiple data (MPMD) code. Component models in TSMP v1.1.0 are: the hydrologic model ParFlow, the CLM land model [13], and the COSMO numerical weather prediction (NWP) model [14], alternatively the ICON atmospheric model may be used, each in multiple versions; TSMP uses the OASIS3-MCT coupler [15]. The Parallel Data Assimilation Framework (PDAF) [16] is also implemented and used so far primarily to assimilate soil moisture [6]. Fig. 1 illustrates some variables of the terrestrial water cycle as simulated by TSMP.

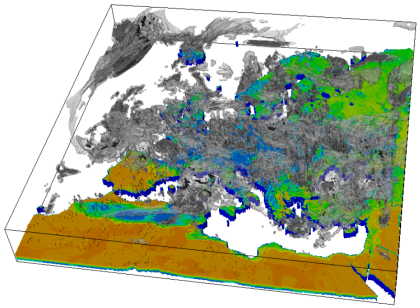


Fig. 1. Snapshot of fully coupled simulations with TSMP of 3D groundwater and soil moisture (orange: dry; blue: wet) and liquid/ice cloud water content (gray) in the summer of 2013 over the widely used European CORDEX domain at 12 km lateral resolution.

B. Typical Data Volumes for Common Experiments

Common numerical experiments conducted with TSMP or one of its component models, or a combination thereof, range from individual catchments to high resolution continental model domains and target, e.g., process- and sensitivity studies [17], hindcasts [11], forecasts [18], or climate change simulations [19]. To a much lesser extent, within the Coordinated Regional Downscaling Experiment (CORDEX) initiative, the WRF regional climate model is used to simulate time slices at 3 km convection permitting resolution, as a contribution to climate change assessments [20], [21]. Tab. I gives an overview of some typical raw model output data volumes of past and ongoing HPSC TerrSys experiments.

Data volumes as listed in Tab. I render conventional data handling, data movements, I/O operations, analysis and storage with respect to the temporal effort already very inefficient; for example post-processing wall clock times may eventually approach simulation times. As model resolution steadily increases, we are clearly expecting that large experiments, such as demonstrated for the CONUS domain at 1 km with ParFlow [11], or with global MPAS simulations at 3 km [9], become the default; for example, time-slice climate change experiments at about 2 km resolution for large parts of Europe are feasible already [23].

TABLE I
EXAMPLES OF TYPICAL RAW MODEL OUTPUT DATA VOLUMES PER EXPERIMENT.

Experiment	Model	Domain	Resolution (km)	Length	Volume (TB)
[19]	TSMP	EU[e]	12	30yrs	84
Planned[a]	TSMP	EU	12	141yrs	924
[18][b]	TSMP	EU, NRW	12, 1/0.5	daily	66, 28
[22][c]	CLM	EU	3	20x7	21
[11][d]	ParFlow	CONUS	1	n.a.	0.5
[21]	WRF	EU, Rhine	12, 3	48yrs	288

[a]Climate change projection, 1961-2100, 3 RCPs
[b]Forecasting simulation, run once per day, two domains
[c]20 ensemble members, 7 years, daily data
[d]Per output interval, run until equilibrium is reached
[e]European model domain, see Fig. 1

III. Current Status

This section addresses some important data-related aspects along our established TSMP workflows within HPSC TerrSys. The overall goal and requirement is to keep the data volume low and avoid data movement as much as possible.

A. Data Formats

The netCDF file format is at the core of our big data strategy (version 4 on top of HDF5). netCDF has established itself as the quasi-standard for numerical models in the Earth sciences. NetCDF data are portable, interoperable, allow for large file sizes, are suitable for long-term storage, self-describing through meta data (several standards exist, e.g., CF convention), and offer lossless compression. Using the lowest and fastest deflation level usually results in between 25% to 50% data volume reduction. As parallel I/O and compression exclude each other with shared-sfile netCDF, a trade-off is usually between I/O performance and data volume. For European model domain extends at non-convection permitting resolutions (e.g., 12 km), the I/O overhead is considered as less relevant than the benefits we gain via compression.

B. Input/Output

To adjust the I/O capabilities of the hydrological model ParFlow to the overall workflow and enhance its interoperability, a parallel netCDF API has been implemented, aided by the Jülich Benchmarking Environment tool (JUBE2) and the Darshan I/O characterization library [24]. ParFlow writes shared netCDF4 files using ROMIO hints for I/O optimization, and uses chunking to improve performance for typical later-on file access patterns, as well as an optional node-level collective I/O with a nearly linear strong scaling I/O behaviour, see Fig. 2.

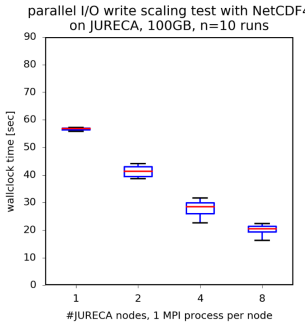


Fig. 2. Example of synthetic scaling study for fine-tuning netCDF4 parallel I/O before implementing the API into the ParFlow code.

C. Pre- and Post-Processing

Aside from various online diagnostics (e.g., variables on pressure levels, CAPE, CIN in COSMO), one of the most important post-processing steps is the transformation of raw model outputs into more user-friendly, lower volume, and better traceable data products. Based on the CMIP output requirements, as defined through the Climate Model Output Rewriter (CMOR) library[3], and its implementation[4] in the CORDEX project, a defined data structure, with a controlled vocabulary, a data reference syntax and unambiguous meta data standards are applied via a separate post-processing to the TSMP outputs. This "CMORization" complements the use of the standardized netCDF data format and usually reduces the data volume further, e.g., by a reduction of vertical levels and also provides products such as temporally aggregated data. CMORized model output of large simulations is often still well usable, even without big data-capable analysis tools.

D. Storage and Archival

The CMORization also allows for an efficient research data management as it helps to ensure that the FAIR (findable, accessible, interoperable, reusable) principles are met. In HPSC TerrSys, data is stored centrally on shared filesystems at Jülich Supercomputing Centre (JSC); data is collaboratively used and can remain without data movements. As part of our data handling and storage strategy, raw model outputs may be erased and only the restart files are retained for a selected number of, e.g., monthly restart points. This however requires a configuration management system to ensure reproducibility. Archiving of either post-processed and / or raw model outputs, to have, e.g., the full vertical resolution still available for subsequent data analysis, is done for some simulations through dedicated data projects of JSC.

E. Reproducibility

We try to meet reproducibility requirements [25] by combining git-based source code repositories and configuration management systems (i.e., compilation information, model configurations, etc.) and workflow engines [24], that allow for a reproduction of simulations and hence data. This system also relies on a succession of fully functional, stable software stages throughout the HPC system's life time. Yet still, an exact, bit-wise reproduction of data can seldom be achieved. A data provenance tracking capability is maintained by assigning universally unique identifiers (UUIDs) as data tracking IDs to individual files. Aside from identifiers that reflect experiment characteristics, UUIDs are also used as numerical model experiment identifiers and both are part of the netCDF file meta data and the configuration management system.

F. Dissemination

Open access research data management systems, that provide cataloging services, such as the Earth System Grid Federation[5] (ESGF) data nodes for global and regional climate projections, e.g., from CORDEX simulations, or the EUDAT[6] research data service, can efficiently be used to publish and share data for a later re-use. Through standardized APIs web-processing services can be connected for query and analysis. A more ad-hoc, long-term research data infrastructure is provided through a data publication repository infrastructure[7], which sits very close to the JSC filesystem.

IV. Necessary Next Steps

Based on existing technical solutions, a number of concrete further steps are ongoing in HPSC TerrSys to ensure fully big data-capable modeling chain and analytics frameworks:

1) HPSC TerrSys is participating in the development and testing of the Helmholtz Analytics Toolkit (HeAT), a distributed tensor framework for high performance data analytics; it is planned to use HeAT in future compute intensive data analysis tasks.
2) Where applicable, code modernization efforts are ongoing to successively parallelize the most relevant processing and analysis tools and include parallel I/O (e.g., Python netCDF4) throughout.
3) In-situ processing, analysis and visualization avoids cost-intensive and time-consuming I/O and storage operations, and reduces data volume and post-processing substantially. In-situ processing (staged vs. on-node, loosely vs. tightly coupled implementations) may be used for different tasks, e.g., solver run time analysis, 3D visualization, or large scale water cycle diagnostics, using, e.g., SENSEI, Catalyst, or Parallel Data Interface libraries.

V. Summary

We give an overview of relevant aspects on how we are dealing with increasing data volumes from simulations in HPSC TerrSys. It is worth noting that basically all necessary methods and software, to make our modeling chains

[3]https://cmor.llnl.gov/
[4]http://is-enes-data.github.io/cordex_archive_specifications.pdf
[5]https://esgf.llnl.gov/
[6]https://www.eudat.eu/
[7]https://www.re3data.org/repository/r3d100012923

big data capable, exist and are ready to be implemented. Given the current accelerated development towards exascale HPC systems and simulation software [26], further substantial efforts seem needed to not let the computational capacity outperform our data handling and analysis capabilities. From a practitioner's point of view, today's state-of-the-art numerical model experiment big data volumes already require a careful and prudent planning of the entire data life cycle.

ACKNOWLEDGMENTS

Work and developments described in this manuscript were supported by the Helmholtz Association Initiative and Networking Fund through the "Helmholtz Analytics Framework" (HAF)[8], the "Advanced Earth System Modelling Capacity" (ESM)[9] projects, and the Horizon 2020 "Energy oriented Centre of Excellence" (EoCoE)[10] project.

REFERENCES

[1] N. E. Davis, R. W. Robey, C. R. Ferenbaugh *et al.*, "Paradigmatic shifts for exascale supercomputing," *The Journal of Supercomputing*, vol. 62, no. 2, pp. 1023–1044, 2012. [Online]. Available: http://link.springer.com/10.1007/s11227-012-0789-3

[2] A. F. Prein, W. Langhans, G. Fosser *et al.*, "A review on regional convection-permitting climate modeling: Demonstrations, prospects, and challenges," *Reviews of Geophysics*, vol. 53, no. 2, pp. 323–361, 2015. [Online]. Available: http://doi.org/10.1002/2014RG000475

[3] P. Shrestha, M. Sulis, M. Masbou *et al.*, "A scale-consistent terrestrial systems modeling platform based on cosmo, clm and parflow," *Monthly Weather Review*, vol. 142, no. 9, pp. 3466–3483, 2014. [Online]. Available: http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-14-00029.1

[4] D. Leutwyler, O. Fuhrer, X. Lapillonne *et al.*, "Towards European-scale convection-resolving climate simulations with GPUs: a study with COSMO 4.19," *Geoscientific Model Development*, vol. 9, no. 9, pp. 3393–3412, 2016. [Online]. Available: http://www.geosci-model-dev.net/9/3393/2016/gmd-9-3393-2016-discussion.html

[5] V. Eyring, S. Bony, G. A. Meehl *et al.*, "Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization," *Geoscientific Model Development*, vol. 9, no. 5, pp. 1937–1958, 2016. [Online]. Available: http://www.geosci-model-dev.net/9/1937/2016/

[6] W. Kurtz, G. He, S. Kollet *et al.*, "TerrSysMP-PDAF (version 1.0): a modular high-performance data assimilation framework for an integrated land surface–subsurface model," *Geoscientific Model Development*, vol. 9, no. 4, pp. 1341–1360, 2016. [Online]. Available: http://doi.org/10.5194/gmd-9-1341-2016

[7] J. T. Overpeck, G. A. Meehl, S. Bony *et al.*, "Climate data challenges in the 21st century," *Science (New York, N.Y.)*, vol. 331, no. 6018, pp. 700–702, 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21311006

[8] T. N. Palmer, "A personal perspective on modelling the climate system," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, vol. 472, no. 2188, p. 20150772, 2016. [Online]. Available: http://rspa.royalsocietypublishing.org/lookup/doi/10.1098/rspa.2015.0772

[9] D. Heinzeller, M. G. Duda, and H. Kunstmann, "Towards convection-resolving, global atmospheric simulations with the Model for Prediction Across Scales (MPAS) v3.1: an extreme scaling experiment," *Geoscientific Model Development*, vol. 9, no. 1, pp. 77–110, jan 2016. [Online]. Available: http://www.geosci-model-dev.net/9/77/2016/

[10] M. F. P. Bierkens, V. A. Bell, P. Burek *et al.*, "Hyper-resolution global hydrological modelling: what is next?" *Hydrological Processes*, vol. 29, no. 2, pp. 310–320, 2015. [Online]. Available: http://doi.wiley.com/10.1002/hyp.10391

[11] R. M. Maxwell, L. E. Condon, and S. J. Kollet, "A high-resolution simulation of groundwater and surface water over most of the continental US with the integrated hydrologic model ParFlow v3," *Geoscientific Model Development*, vol. 8, no. 3, pp. 923–937, 2015. [Online]. Available: http://www.geosci-model-dev.net/8/923/2015/

[12] F. Gasper, K. Goergen, P. Shrestha *et al.*, "Implementation and scaling of the fully coupled Terrestrial Systems Modeling Platform (TerrSysMP v1.0) in a massively parallel supercomputing environment – a case study on JUQUEEN (IBM Blue Gene/Q)," *Geoscientific Model Development*, vol. 7, no. 5, pp. 2531–2543, 2014. [Online]. Available: http://doi.org/10.5194/gmd-7-2531-2014

[13] K. Oleson, D. Lawrence, G. Bonan *et al.*, "Technical description of version 4.0 of the community land model (clm)," National Center for Atmospheric Research, Boulder, CO, NCAR Technical Note NCAR/TN-478+STR, 2010. [Online]. Available: http://www.cesm.ucar.edu/models/cesm1.1/clm/CLM4_Tech_Note.pdf

[14] M. Baldauf, A. Seifert, J. Förstner *et al.*, "Operational Convective-Scale Numerical Weather Prediction with the COSMO Model: Description and Sensitivities," *Monthly Weather Review*, vol. 139, no. 12, pp. 3887–3905, apr 2011. [Online]. Available: http://dx.doi.org/10.1175/MWR-D-10-05013.1http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-10-05013.1

[15] S. Valcke, "The OASIS3 coupler: a European climate modelling community software," *Geoscientific Model Development*, vol. 6, no. 2, pp. 373–388, 2013. [Online]. Available: http://doi.org/10.5194/gmd-6-373-2013

[16] L. Nerger and W. Hiller, "Software for ensemble-based data assimilation systems – Implementation strategies and scalability," *Computers & Geosciences*, vol. 55, pp. 110–118, 2013. [Online]. Available: http://doi.org/10.1016/j.cageo.2012.03.026

[17] J. Keune, F. Gasper, K. Goergen *et al.*, "Studying the influence of groundwater representations on land surface-atmosphere feedbacks during the European heat wave in 2003," *Journal of Geophysical Research: Atmospheres*, vol. 121, no. 22, pp. 13 301–13 325, 2016. [Online]. Available: http://doi.org/10.1002/2016JD025426

[18] S. Kollet, F. Gasper, S. Brdar *et al.*, "Introduction of an experimental terrestrial forecasting/monitoring system at regional to continental scales based on the terrestrial systems modeling platform (v1.1.0)," *Water*, vol. 10, no. 11, 2018. [Online]. Available: http://www.mdpi.com/2073-4441/10/11/1697

[19] C. Furusho, K. Goergen, K. Kulkarni *et al.*, "Pan-european groundwater to atmosphere terrestrial systems climatology from physically consistent simulations," unpublished.

[20] E. Coppola, S. Sobolowski, E. Pichelli *et al.*, "A first-of-its-kind multi-model convection permitting ensemble for investigating convective phenomena over Europe and the Mediterranean," *Climate Dynamics*, 2018. [Online]. Available: https://doi.org/10.1007/s00382-018-4521-8

[21] S. Knist, K. Goergen, and C. Simmer, "Evaluation and projected changes of precipitation statistics in convection-permitting WRF climate simulations over Central Europe," *Climate Dynamics*, pp. 1–17, 2018. [Online]. Available: http://doi.org/10.1007/s00382-018-4147-x

[22] B. S. Naz, W. Kurtz, C. Montzka *et al.*, "Improving soil moisture and runoff simulations at 3 km over europe using land surface data assimilation," *Hydrology and Earth System Sciences*, vol. 23, no. 1, pp. 277–301, 2019. [Online]. Available: https://www.hydrol-earth-syst-sci.net/23/277/2019/

[23] S. Berthou, E. J. Kendon, S. C. Chan *et al.*, "Pan-european climate at convection-permitting scale: a model intercomparison study," *Climate Dynamics*, 2018. [Online]. Available: https://doi.org/10.1007/s00382-018-4114-6

[24] W. Sharples, I. Zhukov, M. Geimer *et al.*, "Best practice regarding the three P's: profiling, portability and provenance when running HPC geoscientific applications," *Geoscientific Model Development*, vol. 11, pp. 2875–2895, 2018. [Online]. Available: https://doi.org/10.5194/gmd-11-2875-2018

[25] V. Stodden, M. McNutt, D. H. Bailey *et al.*, "Enhancing reproducibility for computational methods," *Science*, vol. 354, no. 6317, pp. 1240–1241, dec 2016. [Online]. Available: http://www.sciencemag.org/cgi/doi/10.1126/science.aah6168

[26] B. N. Lawrence, M. Rezny, R. Budich *et al.*, "Crossing the chasm: how to develop weather and climate models for next generation computers?" *Geoscientific Model Development*, vol. 11, no. 5, pp. 1799–1821, 2018. [Online]. Available: http://doi.org/10.5194/gmd-11-1799-2018

[8] http://www.helmholtz-analytics.de

[9] https://www.esm-project.net

[10] https://www.eocoe.eu/

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

The Helmholtz Analytics Toolkit (HeAT) A scientific big data library for HPC

Kai Krajsek
Forschungszentrum Jülich GmbH
Institute for Advanced Simulation
Jülich Supercomputing Centre (JSC)
52425 Jülich, Germany
k.krajsek@fz-juelich.de

Claudia Comito
Forschungszentrum Jülich GmbH
Institute for Advanced Simulation
Jülich Supercomputing Centre (JSC)
52425 Jülich, Germany
c.comito@fz-juelich.de

Markus Götz
Karlsruhe Institute of Technology
Steinbuch Centre for Computing (SCC)
Scientific Data Management
76128 Karlsruhe, Germany
markus.goetz@kit.edu

Björn Hagemeier
Forschungszentrum Jülich GmbH
Institute for Advanced Simulation
Jülich Supercomputing Centre (JSC)
52425 Jülich, Germany
b.hagemeier@fz-juelich.de

Philipp Knechtges
German Aerospace Center
Simulation and Software Technology
High-Performance Computing
51147 Cologne, Germany
Philipp.Knechtges@dlr.de

Martin Siggel
German Aerospace Center
Simulation and Software Technology
High-Performance Computing
51147 Cologne, Germany
Martin.Siggel@dlr.de

***Abstract*—We present HeAT, a scientific big data library supporting transparent computation on HPC systems. HeAT builds on top of PyTorch, which already provides many required features like automatic differentiation, CPU and GPU support, linear algebra operations and basic MPI functionality as well as an imperative programming paradigm allowing fast prototyping essential in scientific research. These features are generalized to a distributed tensor with a NumPy-like interface allowing to port existing NumPy algorithms to HPC systems nearly effortlessly.**

***Index Terms*—Big Data Analytics, HPC, Machine Learning, Deep Learning, Data Mining**

I. Introduction

Scientific Big Data Analytics has become an important instrument for tackling scientific problems characterized by the greatest data and computational complexity. Scientific data, e.g. MRI images, satellite data, detectors or numerical simulations on high-performance computers, are growing exponentially in nearly all scientific fields [1]–[4] pushing storage, processing, and analysis of such data to its limits. Traditional techniques for handling scientific data need to be replaced by specific solutions taking structure, variability and size of todays data sets into account. This paper presents the Helmholtz Analytics Toolkit (HeAT), a scientific big data analytics library for HPC systems enabling scientists to take full advantage of parallel high-performance computing with minimal programming effort on their side.

The large progress in big data analytics in general and machine learning/deep learning (ML/DL) in particular, has been considerably spurred by well-designed open source libraries like Hadoop, Spark, Storm, Disco, scikit-learn, H2O.ai, Mahout, TensorFlow, PaddlePaddle, PyTorch, Caffe, Keras, MXNet, CNTK, BigDL, Theano, Neon, Chainer, DyNet, Dask and Intel DAAL, to mention some of them. Despite the large number of existing data analytics frameworks, a library taking the specific needs in scientific big data analytics under consideration is still missing. For instance, no pre-existing library operates on heterogeneous hardware like GPU/CPU systems while allowing transparent computation on distributed systems. Typical big data analytics frameworks like Spark are designed for distributed memory systems and consequently do not fully exploit the shared memory architecture as well as the network technology of HPC systems. ML/DL frameworks like Theano or Chainer focus on single node computations or, when providing mechanisms for distributed computation, as done by TensorFlow or PyTorch, they impose the details of the distributed computation to the programmer. Libraries designed for HPC like Dask and Intel DAAL do not provide any GPU support. In the following, we will describe the core concepts of HeAT in order to fill the gap of existing big data libraries, and demonstrate its usage on a k-means cluster algorithm.

II. Co-design development approach

The library is designed and will be implemented in close cooperation with domain scientists within a scientific project, the Helmholtz Analytics Framework[1]. Eight scientific use cases from five different scientific fields (see Figure 1), i.e. earth system modeling, structural biology, aeronautics and aerospace research, medical imaging and neuroscience, have been chosen to ensure consideration of actual challenges of the specific scientific aspects of big data analytics. The use cases are tackling current research questions in their respective

This work is supported by the Helmholtz Association Initiative and Networking Fund under project number ZT-I-0003.

[1] http://www.helmholtz-analytics.de/helmholtz_analytics/EN/Home/home_node.html

fields that come to their limits with traditional data analytics methods.

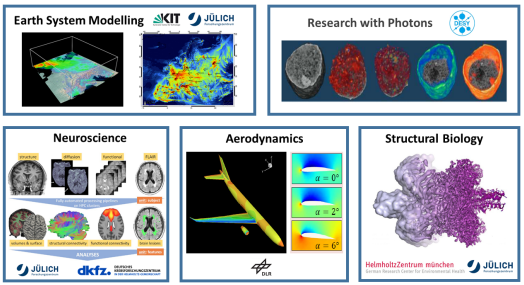


Fig. 1. Illustration of use cases from five scientific fields.

The techniques applied in the various use cases span over 20 different methods ranging from relatively light weight machine learning methods like k-means, or mean shift clustering, over frequent item set mining methods up to deep learning methods like convolutional neural networks for regression and classification tasks.

III. HeAT architecture

HeAT is based on a tensor data object on which basic scalar functions, linear algebra algorithms, slicing or broadcasting operations necessary for most data analytics algorithms can be performed. The tensor data objects reside either on the CPU or on the GPU and, if needed, are distributed over various nodes. Operations on tensor objects are transparent to the user, i.e. they remain the same irrespective of whether the tensor object resides on a single node or it is distributed over several nodes, allowing to conveniently port algorithms from single nodes to multiple nodes or from CPUs to GPUs. HeAT builds on top of PyTorch [5]. Development started in May 2018 and is, at the time of writing this paper, in an early pre-alpha phase. It is developed in the open, hosted on GitHub[2] and distributed under the MIT license. The basic design has been worked out and basic implementations have been carried out. A role model for

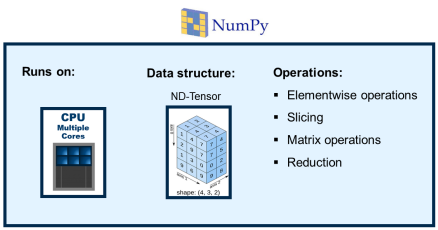


Fig. 2. The basic structure of the NumPy library: a tensor data structure and operations on top. The operations run transparently on multiple cores of one CPU.

HeAT is NumPy [6], a popular scientific Python library widely used for data analytics (see Figure 2). NumPy transparently makes use of all available CPU cores on one processor such that the user can focus on the algorithmic development without struggling with parallel programming issues. But NumPy has no further parallel programming features nor any GPU capabilities. In order to account for GPU computing and automatic differentiation we decided to rely on a modern tensor library. Overall, we examined 16 deep learning and big data libraries with respect to their properties and selected four of them for a benchmark with respect to memory consumption, CPU as well as GPU runtime: PyTorch, MXNet [7], TensorFlow [8] and ArrayFire [9]. As a result of the benchmark, we chose PyTorch as the backend for our HeAT library. Detailed results of the benchmark will be published separately. PyTorch is a

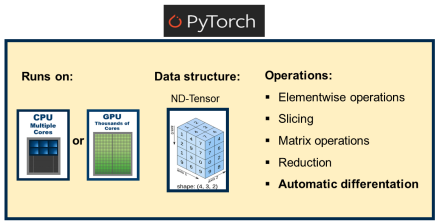


Fig. 3. The basic structure of the PyTorch library: A tensor data structure and operations as well as automatic differentation on top. The operations run transparently on multiple cores of one CPU or on one GPU.

deep learning library originally developed for neural network training and inference (see Figure 3). Its core module can be considered as an extension to NumPy with respect to automatic differentiation and GPU computation. It supports a subset of the NumPy operations and provides own operations required for artificial neural networks. A PyTorch tensor can be labeled to be differentiable and all subsequent operations are traced within a dynamical computational graph. The derivative of any transformed tensor with respect to the differentiable tensor can then be obtained with just one command due to the involved automatic differentiation mechanism. Computations on the GPU are automated, too. The PyTorch tensor is transferred onto the GPU by a single command or constructed directly on the GPU. PyTorch operation commands remain the same as for the CPU. When it comes to distributed computation, PyTorch supports several frameworks, i.e. TCP, GLOO, MPI and NCCL. However, details of the distribution of tensors on different nodes as well as the communication between the nodes need to be managed by the user.

HeAT builds upon PyTorch, providing an additional layer for distributed computation on GPUs as well as CPUs based on MPI (see Figure 4). Operations on tensor objects are transparent to the user, i.e. they remain the same irrespective of whether the tensor object resides on a single node or it is distributed over several nodes, allowing to conveniently port algorithms from single nodes to multiple nodes or from CPUs to GPUs. The basis of HeAT is a tensor object, an ND

[2]https://github.com/helmholtz-analytics/heat

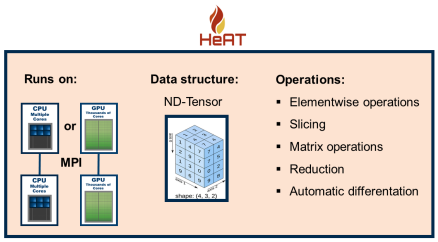


Fig. 4. The basic structure of the HeAT library: A tensor data structure and operations as well as automatic differentation on top. The operations run transparently on multiple cores of multiple CPUs or on mulptiple GPUs.

array structure of homogeneous numerical values. The tensor object is, if requested, split into several subsets along one selected dimension, whereby each subset belongs to one MPI rank (see Figure 5). The tensor object is directly created on different MPI ranks and filled with predefined values, e.g. equal values or random numbers. Alternatively, values are loaded from disc by parallel I/O via parallel HDF5 or parallel NetCDF. Operations on the HeAT tensor object can then be applied transparently, i.e. the user does not need take care about data transfer between the MPI ranks. The design of the HeAT operations follows the NumPy convention as far as possible, i.e. in the ideal case an algorithm implemented in NumPy can be ported to HeAT by simply exchanging NumPy operations with their HeAT counterparts. To this end, NumPy functions and methods are re-implemented using PyTorch and MPI4Py [10]. As an example, consider the creation of a one dimensional tensor filled with evenly spaced float values within a given interval running on three MPI ranks:

```
import heat as ht
range_data = ht.arange(6, split = 0)
```

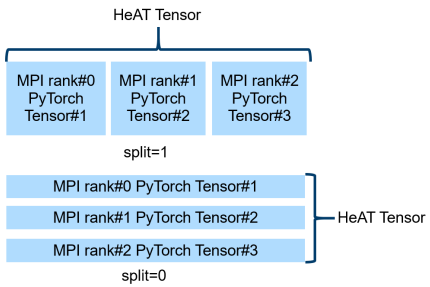


Fig. 5. Illustration of the splitting mechanism of the HeAT library on a two dimensional tensor. The tensor is equally distributed among the three requested MPI ranks. The HeAT tensor subset and each rank is realized by a PyTorch tensor. Splitting is supported in one of the two dimensions.

After importing the HeAT module, a tensor containing the numbers from 0. up to 5. is created. Internally, a subset containing values 0. and 1. is attached to rank number zero, the values 2. and 3. are attached to rank number one and the last two numbers are attached to rank number three. Subsequent operations can then be applied to the tensor object without caring about its distributed nature. For instance, the maximum of the tensor object can be obtained by the `argmax` method:

```
range_data.argmax()
>>>5
```

Also, computing the sum over all elements correspond to its NumPy counterpart:

```
range_data.sum()
>>>15
```

In order to support deep learning approaches and other ML methods requiring gradient based optimization, the automatic differentiation mechanism proposed in [11] will be extended to distributed computation. In a first step, a corresponding distributed adjoint operation is implemented for each HeAT tensor. Note that the PyTorch automatic differentiation mechanism can be re-used for all pointwise operations. If an operation is performed on a HeAT tensor being marked as differentiable, references to the operation, to its results as well as to the operation's arguments are stored in an object constituting a node in a dynamical computational graph. The references to the operation arguments are the edges to the parents of the dynamical graph. In order to perform back-propagation, we need the topological order of the graph. This order is obtained by storing a list tracking the order of the transformations applied to each differentiable tensor, i.e. we store a history of transformation for each differentiable tensor. In order to obtain the derivative of any node with respect to a differentiable tensor, the corresponding lists are traversed in reverse order. At each position in the list, the derivative of the output with respect to the input is computed using the corresponding stored node object.

IV. Example: k-means

As a demonstration of the library we describe how to port a k-means [12] NumPy implementation to its HeAT counterpart. We sketch the important steps of the algorithm by comparing NumPy and HeAT code snippets. The full HeAT k-means implementation can be found at https://github.com/helmholtzanalytics/heat/tree/master/heat/ml/cluster. K-means is a clustering algorithm that groups a set of data points with a predefined number of clusters according to the minimization problem

$$\arg\min_{\mathcal{C}} \sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} \|x - \mu_i\|^2 \qquad (1)$$

where μ_i denotes centroid i, $\mathcal{C}_i$ denotes cluster i and k denotes the number of clusters. A local minimum of the optimization problem (1) can be obtained by the algorithm:

1) Choose k centroids
2) For each data point calculate the distance to all centroids
3) Assign each data point to the cluster with the closest centroid
4) Estimate new centroids as the mean of their corresponding cluster points
5) Go to 2 until convergence

Before one can apply the first step of the k-means algorithm, the data points to be clustered need to be loaded in the corresponding NumPy arrays as well as HeAT tensors. Whereas in the NumPy implementation the data are loaded into the NumPy arrays as a whole data block, in the HeAT implementation, if HeAT is running in distributed mode, only the data needed by the corresponding rank are loaded by the parallel I/O mechanism. All consecutive operations on the constructed arrays/tensors are equal or differ only with respect to small details. After choosing k initial centroids we need to compute the distance (step 2) of each point to the centroids and determine the index of the smallest distance. With NumPy, the second step can be realized by

```
distances = ((data - centroids) **
    2).sum(axis=1, keepdims=True)
matching_centroids =
    np.expand_dims(distances.argmin(axis=2),
    axis=2)
```

where `data` is a NumPy array of size $(n, m, 1)$ containing n m-dimensional data points and `centroids` is a NumPy array of size $(1, m, k)$ containing the initially chosen centroids. The corresponding HeAT implementation reads

```
distances = ((data - centroids) **
    2).sum(axis=1)
matching_centroids = distances.argmin(axis=2)
```

where the only differences stem from the fact that HeAT keeps dimensions after `sum` and `argmin` operations. Assigning the data points to their closest centroids (step 3) differs in NumPy

```
selection = (matching_centroids ==
    i).astype(np.int64)
```

from HeAT

```
selection = (matching_centroids ==
    i).astype(ht.int64)
```

by the build-in data types. The estimate of the new centroids (step 4) in NumPy is

```
new_centroids[:, :, i:i + 1] = ((data *
    selection).sum(axis=0. keepdims=True)
selection.sum(axis=0).clip(1.0, sys.maxsize))
```

and in HeAT

```
new_centroids[:, :, i:i + 1] = ((data *
    selection).sum(axis=0)
selection.sum(axis=0).clip(1.0, sys.maxsize))
```

The only difference is given by the way dimensions are kept after the `sum` operation.

V. Summary

We presented HeAT, a scientific big data library. After motivating the need for an additional big data analytics library in the scientific context, we described its core design principles, i.e. a distributed tensor object with transparent operations on top, as well as the design of the automatic differentiation mechanism. We finally illustrated the usage of the HeAT library by porting the k-means cluster algorithm from NumPy to HeAT demonstrating the close similarity of their user interfaces.

References

[1] R. N. Boubela, K. Kalcher, W. Huf, C. Nasel, and E. Moser, "Big data approaches for the analysis of large-scale fMRI data using Apache Spark and GPU processing: A demonstration on resting-state fMRI data from the human connectome project," *Frontiers in Neuroscience*, vol. 9, p. 492, 2015.

[2] F. Bamberg, H.-U. Kauczor, S. Weckbach, C. L. Schlett, M. Forsting, S. C. Ladd, K. H. Greiser, M.-A. Weber, J. Schulz-Menger, T. Niendorf, T. Pischon, S. Caspers, K. Amunts, K. Berger, R. Blow, N. Hosten, K. Hegenscheid, T. Krncke, J. Linseisen, M. Gnther, J. G. Hirsch, A. Khn, T. Hendel, H.-E. Wichmann, B. Schmidt, K.-H. Jckel, W. Hoffmann, R. Kaaks, M. F. Reiser, and H. a. Vlzke, "Whole-body MR imaging in the german national cohort: Rationale, design, and technical background," *Radiology*, vol. 277, no. 1, pp. 206–220, 2015. PMID: 25989618.

[3] J.-G. Lee and M. Kang, "Geospatial big data: Challenges and opportunities," *Big Data Research*, vol. 2, pp. 74–81, 2015.

[4] T. C. P.-G. Consortium, "Computational pan-genomics: Status, promises and challenges," *Briefings in Bioinformatics*, vol. 19, no. 1, pp. 118–135, 2018.

[5] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[6] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online; accessed 15.11.2018].

[7] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems.," *CoRR*, vol. abs/1512.01274, 2015.

[8] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[9] P. Yalamanchili, U. Arshad, Z. Mohammed, P. Garigipati, P. Entschev, B. Kloppenborg, J. Malcolm, and J. Melonakos, "ArrayFire - A high performance software library for parallel computing with an easy-to-use API," 2015.

[10] L. Dalcín, R. Paz, and M. Storti, "MPI for Python," *J. Parallel Distrib. Comput.*, vol. 65, pp. 1108–1115, Sept. 2005.

[11] D. Maclaurin, *Inference and Optimization with Composable Differentiable Procedures*. Dissertation, Harvard University, 2016.

[12] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, (Berkeley, Calif.), pp. 281–297, University of California Press, 1967.

Extreme Data: Demands, Technologies, and Services
Workshop at Forschungszentrum Jülich, Germany, 18-19 September 2018
Proceedings, IAS series, volume 40, March 2019
ISBN: 978-3-95806-392-1, urn:nbn:de:0001-2019032102

Personalized medicine: the need for exascale data handling

Matthias Becker

PRECISE, Platform for Single Cell Genomics and Epigenomics at the German Center for Neurodegenerative Diseases (DZNE) and the University of Bonn, Bonn, Germany

Matthias.Becker@dzne.de

Hartmut Schultze

Hewlett Packard Enterprise

Ratingen, Germany

Hartmut.Schultze@hpe.com

Joachim L. Schultze

PRECISE, Platform for Single Cell Genomics and Epigenomics at the German Center for Neurodegenerative Diseases (DZNE) and the University of Bonn, Bonn, Germany

Joachim.Schultze@dzne.de

***Abstract*—The intersection of life and computer sciences is of growing importance for the future of personalized medicine. To enable such data-driven treatments, large collections of data need to be gathered, ranging from genomics to high-resolution image data. These collections, initially in cohorts, will be performed at local physicians, hospitals and specialized centers. This results in large amounts of data that require near-real-time processing, to be usable as input for medical decisions. Handling such data collections and providing the required computational resources, either at the edge or in cloud solutions is challenging, since population-wide applications of precision medicine will reach exascale levels. Memory-driven computing with a flexible fabric is one approach to face these challenges.**

Keywords—exascale data, exascale computing, precision medicine, memory-driven computing, population studies

I. Introduction

Neurodegenerative diseases, like Alzheimer's, are not only a burden for those affected but also a growing challenge for society. While many of these diseases are not fully understood yet, the search for biomarkers, which allow early detection, is ongoing. Such research efforts require a paradigm shift, where a data-driven approach is used to recognize pattern, e.g. using AI, in exascale data collections. Besides research questions, insights gained from large data sets are being translated to clinical practice, fueling the personalized medicine domain by including many data sources such as genomics, imaging, laboratory and clinical testing data.

II. Medical Data Sources

There are two scenarios where large-scale data collection is performed in medicine. First, population studies follow large cohorts and collect data for research purposes (Fig 1). Second, acquisitions in clinical practice to apply personalized medicine. These acquisitions can be single modalities or consist of multiple sources, especially population studies aim to collect a large body of measurements for analysis. These collections are often made available to other researchers or provided in open repositories.

Population studies can be very specialized or rather broad. The Human Functional Genomics Project [1] has acquired data from over 500 participants to study the effect of genetic variation in human DNA, but also epigenetic and environmental influences. The UK Biobank [2] on the other side has the broad approach to improve the prevention, diagnosis and treatment of serious illnesses by studying 500,00 volunteers. The Rhineland Study [3] aims to study neurodegenerative diseases by following 30,000 participants, but repeatedly over the next 30 years. These studies generate large data sets that already play a significant role in the study design, since data handling and processing remain a major challenge. Results obtained from such large studies, but also clinical studies build the foundation of future data-driven medicine in daily practice.

However, once personalized medicine becomes more prevalent, medical centers, hospitals and physicians will generate an avalanche of data that requires near-real-time processing for medical decision support.

A. Data types

Data falls into two categories, user-generated and clinically collected data. Data curation is needed to ensure a consistent level of quality. In clinical settings, this can be achieved through SOPs (standard operating procedures), for user-generated input careful checking and filtering is needed.

Typical data points collected are body measurements (including height, weight, body mass index, body circumference at different positions, body fat), medical history and family background, cognitive tests, psychological evaluations, cardiovascular data (blood pressure, heart rate), ECGs (electrocardiogram) in rest and exercise and data derived from a collection of biopsy samples. Blood samples can be used to extract genomic data, a source of large data sets. Another modality generating large data, is medical imaging. Two-dimensional imaging is used for retinal scans and skin measurements. Video data is acquired in colonoscopies or bronchoscopy. Finally, a large contributor of data is volumetric imaging. The two most common modalities are CT (computed tomography) and MRI (magnetic resonance imaging). They can be used for whole-body imaging or more targeted acquisitions of individual organs, e.g. kidneys or the brain to understand the morphology or neuro-degenerative processes.

B. Data sizes

The data points acquired in population studies and clinical routine ranges over a large scale of sizes. Clinical reports and many measurements are either textual or small vectors

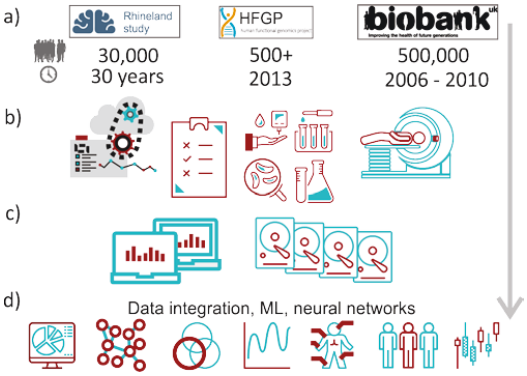


Figure 1 Data sources like population studies (a) acquire different data types (b) that require processing (c) to generate insights (d).

of measurements. They can be easily stored in electronic health records, solutions which are typically built on top of databases. However, the image data of single acquisitions is typically in the range of several GB. In clinical environments, PACSs (Picture Archiving and Communication System) are commonly used to store and share this data. Specialized software like Osirix [4] can be used to browse, view and annotate the image data with standardized data transfer protocols [5].

Genomic data is another reason for growing storage needs. The human genome consists of 3 billion bases and 30 copies are required for reliable analysis. Together with necessary quality information, this leads to an uncompressed size of 180 GB per genome. After processing, 100 GB of compressed information remain for long term storage.

With clinical application of imaging and genomics becoming more prevalent, data management becomes increasingly important for the health sector. Medical research with population studies faces a similar challenge. Just for the Rhineland study with its 30,000 participants, the expected amount of generated data is estimated to be between 500 PB and up to 2 EB.

III. Computational Challenges

Exascale amounts of data from such studies and growing clinical acquisitions for personalized medicine create manifold computational challenges. The data needs to be stored and organized, including ensuring the security of this sensitive data. Collaboration is a core component of scientific progress; therefore, data sharing needs to be organized. Besides the technical challenges, a legal framework is required to ensure privacy and compliance with national and international regulations. Finally, the actual processing and analysis of such large data collections requires a combination of heterogenous computing architectures and approaches.

A. Data Management

Collecting exascale amounts of data requires proper data management and tracking, especially when data is distributed across multiple sites (labs, institutes, hospitals) and needs to be made available across the organization. Discoverability is a core issue since data can only be used if users are aware of its existence. This can be realized through databases, data management frameworks and searchable interfaces [6]. Besides providing storage systems, a proper distributed backup is needed, whilst trying to minimize data duplication. Archival storage, e.g. on tapes, can be considered, however, researchers often need access to also older data as well, it should be available in near real-time. Waiting times of hours or even days can seriously hinder scientific progress.

In long term storage, data can (and should) be compressed, however, only loss-less approaches can be used, since compression artifacts in, e.g. MRI scans, could be mistaken for biomarkers. When selecting the compression approach, its properties like compression ratio and (de-)compression speed need to be carefully evaluated to fit for the use case. If decompression takes more time than processing the data, a re-evaluation might be needed. A multitude of compression approaches exist, ranging from generic zip compression to specialized genomics formats like MPEG-G [7]. Medical data is highly personal, resting data should always be encrypted using industry standards and regular evaluations of the approach should be performed.

B. Data sharing and privacy

Sharing data is a necessity for research and is encouraged in all research institution. However, medical data requires certain precautions. An ethical evaluation of research projects in advance is common practice and the same committee also checks sharing data. Patient consent forms often limit the use of data, since they specify (and therefore limit) the research that is done with data. In addition, privacy plays an important role that is enforced through a strict legal framework like the European GDPR [8] or the HIPAA [9] rules. Sharing is often limited to anonymized or pseudonymized data, with as little unnecessary meta-information as possible. Nonetheless methods have been proven that it is possible to re-identify individuals from epigenetics data [10].

The actual process of sharing, hence transferring, the data will remain the bottleneck with growing data sizes. Strategies for efficient transfer and sub-setting of data points are needed. The network architecture between the collaborating institutions need be designed to handle the amount of data. Another approach is sharing the processing tools in a container and to process the data locally and only share the (smaller) results. This approach minimizes the data volumes that need to be transferred and sharing processing tools is feasible, since most research tools are now openly available.

C. Processing

Processing the exascale amount of heterogenous data requires different processing approaches. Small data points can be processed using standard systems, e.g. gait data can be easily analyzed in real-time on a laptop. For larger data types server systems or clusters are used. These resources can be either local or federated. Cloud resources can be used where data security and privacy rules permit it. In such environment, approaches like homomorphic encryption [11] need to be further evaluated. Specialized hardware like GPUs have been found to be ideal tools for processing volumetric image data since they provide massively-parallel

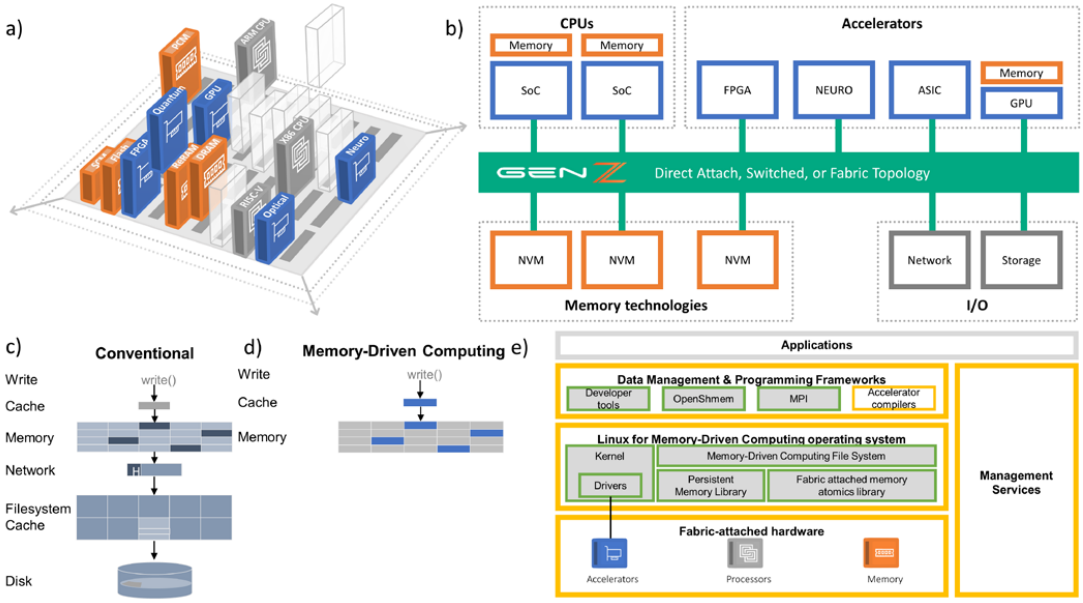


Figure 2 (a) Memory-driven computing (MDC) connects a pool of devices into a single environment. (b) The Gen-Z fabric connects different types of components. (c) Conventional data access and (d) MDC. (e) Software environment for MDC.

capabilities required for image analysis. Furthermore, (deep) neural networks are often used and optimized for GPUs. Large genomics data is processed by a series of bioinformatics tools in specialized pipelines. These tools exchange data through large intermediate files, generating plenty of unnecessary I/O load. This phenomenon is not limited to genomics, e.g. radio astronomy needs to process ever larger amounts of data. Novel approaches and architectures are needed to overcome this bottleneck in data processing.

IV. Memory-Driven Computing

Traditionally, computers are based on the von Neumann architecture. Scaling such systems is achieved by adding more resources, like in clusters or super computers. Scaling is limited by the end of Moore's law [12]. At the same time, data sets grow in size and distributing workloads and partitioning data becomes more complex. The answer to certain questions can simply not be computed fast enough today. One of the approaches to overcome these problems is memory-driven computing where a pool of devices is connected into a single environment (Fig. 2a) using the optical Gen-Z fabric.

Memory-driven computing is a paradigm shift, that puts memory at the center of the compute infrastructure to support todays data-driven applications. System components are connected through a fabric, Gen-Z. Applications have access to a shared, persistent pool of fabric attached memory (FAM). This eliminates the need for dividing or partitioning data for processing reasons. Furthermore, access to remote memory, which typically involves system calls and network operations, is removed. Remote data access can use up to 25.000 operations, with MDC and FAM, the same data can be accessed in just three operations. The underlying fabric allows to set up environments where data can be processed at the edge and large data pools can be made available to concurrent applications without data movements. Existing applications require only few modifications to see first benefits from MDC.

A. Gen-Z Fabric

The Gen-Z consortium was started by 12 core members and by now has grown to a large industry consortium with currently 64 members. In 2018, the version 1.0, of the Gen-Z core specification was published [13]. It standardizes the protocol, components, connectors and component dimensions. Security and authentication have played an important role in the specification, making the fabric ideal for sensitive data such as medical information.

In its current version, Gen-Z connects up to 2^24 devices (16 million), has a byte-addressable space of 4096 Yottabytes (allowing to access up to 250,000 times the size of all data currently existing) and can connect a processing power of up to 2^{70} FLOPs (equaling 1,600 exascale computers).

Components connected by Gen-Z, or Fabric Attached Hardware, fall into three categories: processors, memory and accelerators. This allows the selection of application-specific hardware to compose a system. Furthermore, it is possible to have gateway components, to connect external networks as well as traditional I/O-based system for long-term and archival storage (Fig. 2). The Gen-Z architecture is designed to accommodate even quantum and neuromorphic processors, so novel processor designs can be integrated into the ecosystem. Novel memory types and specialized accelerators can be connected via the fabric as well. This allows existing systems to grow and constantly adapt to changing requirements. Components connected by the Gen-

Z fabric can be combined to flexibly create virtual systems optimized for specific workloads.

B. Transition path to MDC

From an application perspective, MDC can be used through a layered architecture (Fig. 2e). Applications have access to data management and programming frameworks, like MPI and OpenShmem, as well as general developer tools and accelerator-specific compiler. These are on top of an MDC optimized version of linux [14]. The OS runs the drivers for the accelerator. The fabric attached memory is exposed as memory-driven computing file system, based on the persistent memory library and the FAM atomics library. The final layer is the fabric attached hardware. In parallel, management services exist across the layers.

Memory-driven computing systems are not yet commercially available. However, a FAM-Emulation environment has been made available [15] which runs on existing systems with sufficiently large memory. It generates several virtual machines, that share a memory pool (FAM). Development tools and libraries are also available online. First promising results were shown for different domains, including graph processing [16].

There is a transition path for applications to benefit from MDC. The first step is simple: all I/O-operations should be removed in favor of memory-mapping data input and output. Having data in memory enables data loading to be performed in parallel in many cases since performance penalties from random data access are no concern.

In a second step, the overall memory usage of the application can be examined. Often internal data structures, like a reference genome, are optimized to fit into the memory of smaller systems. However, with MDC, these structures can be persisted in FAM and be shared between multiple instances of a tool.

Processing medical and in particular genomic data, often is a task of serially running tasks that load, modify and finally store data. A significant part of the overall processing time is devoted to data loading and sorting. With FAM, data can be exchanged through memory, removing a major bottleneck. Such architectures will be important for effective, energy and cost-efficient processing the data sizes acquired in research and finally in clinical practice.

V. Conclusion

Personalized medicine is the next major change in medicine and default inclusion of genomic and other large data will play a major role. Large population studies will collect the foundational data for the research enabling personalization. The resulting data sizes in the exascale dimension do not only pose a challenge for data handling, management and storage, but also require exascale computing capabilities. Traditional architecture struggle to scale with data sizes, therefore novel architectures are needed. One such example is memory-driven computing, where an abundance of persistent memory is combined with a flexibly configurable system that supports novel accelerators.

Transitioning to such a system requires interfacing computer science with biological research. This could be realized through interdisciplinary researchers that translate between the domains and establish a common language. Using memory-driven computing is a step towards handling the avalanche of data from personalized medicine and provides a possibility to shape the future of the computational needs in clinical and research environments.

Acknowledgment

This work was funded in part by the HGF grant sparse2big and the FASTGenomics grant of the German Federal Ministry for Economic Affairs and Energy. Joachim L. Schultze is member of the excellence cluster ImmunoSensation. We thank Sharad Singhal, Milind Chabbi, Bill Hayes, Keith Packard, Patrick Demichel, Binoy Arnold, Robert Peter Haddad, Eric Wu, Chris Kirby, Rocky Craig from Hewlett Packard Enterprise and the Hewlett Packard Labs and the bioinformatics group at the AG Schultze.

References

[1] Human Functional Genomics Project. Retrieved from http://www.humanfunctionalgenomics.org/site/. [Accessed: 16-Dec-2018].

[2] L. J. Palmer, “UK Biobank: bank on it,” *Lancet*, vol. 369, no. 9578, pp. 1980–1982, Jun. 2007.

[3] German Center for Neurodegenerative Diseases, “Rhineland study.” https://www.rheinland-studie.de/. [Accessed: 16-Dec-2018].

[4] A. Rosset, L. Spadola, and O. Ratib, “OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images,” *J. Digit. Imaging*, vol. 17, no. 3, pp. 205–216, Sep. 2004.

[5] J. J. P. C. Rodrigues, Health Information Systems: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications, no. Bd. 1. IGI Global, 2009.

[6] P. B. Jensen, L. J. Jensen, and S. Brunak, “Mining electronic health records: towards better research applications and clinical care,” *Nat. Rev. Genet.*, vol. 13, no. 6, pp. 395–405, Jun. 2012.

[7] C. Alberti, T. Paridaens, J. Voges, D. Naro, J. J. Ahmad, M. Ravasi, D. Renzi, G. Zoia, I. Ochoa, M. Mattavelli, J. Delgado, and M. Hernaez, “An introduction to MPEG-G , the new ISO standard for genomic information representation,” *bioRxiv*, pp. 1–17, 2018.

[8] B. McCall, “What does the GDPR mean for the medical community?,” *Lancet*, vol. 391, no. 10127, pp. 1249–1250, Mar. 2018.

[9] P. P. Gunn, A. M. Fremont, M. Bottrell, L. R. Shugarman, J. Galegher, and T. Bikson, “The health insurance portability and accountability act privacy rule a practical guide For researchers,” *Med. Care*, 2004.

[10] F. Chen, Q. Gonzalez, Maria Teresa Viswanathan, Krishnamurthy Cai, H. Laffite, J. Rivera, A. Mitchell, and S. Singhal, “Billion node graph inference: iterative processing on The Machine,” 2016.

[11] F. Chen, Q. Gonzalez, Maria Teresa Viswanathan, Krishnamurthy Cai, H. Laffite, J. Rivera, A. Mitchell, and S. Singhal, “Billion node graph inference: iterative processing on The Machine,” 2016.

[12] T. N. Theis and H. S. Philip Wong, “The End of Moore’s Law: A New Beginning for Information Technology,” *Comput. Sci. Eng.*, vol. 19, no. 2, pp. 41–50, 2017.

[13] Gen-Z consortium, “Gen-Z core specification 1.0,” 2018. [Online]. Available: https://genzconsortium.org/specification/core-specification-1-0/. [Accessed: 16-Dec-2018].

[14] Hewlett Packard Labs, “Linux for Memory-Driven Computing.” Available: https://github.com/FabricAttachedMemory. [Accessed: 16-Dec-2018].

[15] Hewlett Packard Labs, “Fabric Attached Memory Emulation.” [Online]. Available: https://github.com/FabricAttachedMemory/Emulation. [Accessed: 16-Dec-2018].

[16] F. Chen, Q. Gonzalez, Maria Teresa Viswanathan, Krishnamurthy Cai, H. Laffite, J. Rivera, A. Mitchell, and S. Singhal, “Billion node graph inference: iterative processing on The Machine,” 2016.

Schriften des Forschungszentrums Jülich
IAS Series

Band / Volume 28
Computational Trends in Solvation and Transport in Liquids
edited by G. Sutmann, J. Grotendorst, G. Gompper, D. Marx (2015)
ISBN: 978-3-95806-030-2
URN: urn:nbn:de:0001-2015020300

Band / Volume 29
Computer simulation of pedestrian dynamics at high densities
C. Eilhardt (2015), viii, 142 pp
ISBN: 978-3-95806-032-6
URN: urn:nbn:de:0001-2015020502

Band / Volume 30
Efficient Task-Local I/O Operations of Massively Parallel Applications
W. Frings (2016), xiv, 140 pp
ISBN: 978-3-95806-152-1
URN: urn:nbn:de:0001-2016062000

Band / Volume 31
A study on buoyancy-driven flows: Using particle image velocimetry for validating the Fire Dynamics Simulator
by A. Meunders (2016), xxi, 150 pp
ISBN: 978-3-95806-173-6
URN: urn:nbn:de:0001-2016091517

Band / Volume 32
Methoden für die Bemessung der Leistungsfähigkeit multidirektional genutzter Fußverkehrsanlagen
S. Holl (2016), xii, 170 pp
ISBN: 978-3-95806-191-0
URN: urn:nbn:de:0001-2016120103

Band / Volume 33
JSC Guest Student Programme Proceedings 2016
edited by I. Kabadshow (2017), iii, 191 pp
ISBN: 978-3-95806-225-2
URN: urn:nbn:de:0001-2017032106

Band / Volume 34
Multivariate Methods for Life Safety Analysis in Case of Fire
B. Schröder (2017), x, 222 pp
ISBN: 978-3-95806-254-2
URN: urn:nbn:de:0001-2017081810

Schriften des Forschungszentrums Jülich
IAS Series

Band / Volume 35
Understanding the formation of wait states in one-sided communication
M.-A. Hermanns (2018), xiv, 144 pp
ISBN: 978-3-95806-297-9
URN: urn:nbn:de:0001-2018012504

Band / Volume 36
A multigrid perspective on the parallel full approximation scheme in space and time
D. Moser (2018), vi, 131 pp
ISBN: 978-3-95806-315-0
URN: urn:nbn:de:0001-2018031401

Band / Volume 37
Analysis of I/O Requirements of Scientific Applications
S. El Sayed Mohamed (2018), XV, 199 pp
ISBN: 978-3-95806-344-0
URN: urn:nbn:de:0001-2018071801

Band / Volume 38
Wayfinding and Perception Abilities for Pedestrian Simulations
E. Andresen (2018), 4, x, 162 pp
ISBN: 978-3-95806-375-4
URN: urn:nbn:de:0001-2018121810

Band / Volume 39
Real-Time Simulation and Prognosis of Smoke Propagation in Compartments Using a GPU
A. Küsters (2018), xvii, 162, LIX pp
ISBN: 978-3-95806-379-2
URN: urn:nbn:de:0001-2018121902

Band / Volume 40
Extreme Data Workshop 2018
Forschungszentrum Jülich, 18-19 September 2018
Proceedings
M. Schultz, D. Pleiter, P. Bauer (Eds.) (2019), 64 pp
ISBN: 978-3-95806-392-1
URN: urn:nbn:de:0001-2019032102

Weitere ***Schriften des Verlags im Forschungszentrum Jülich*** unter
http://wwwzb1.fz-juelich.de/verlagextern1/index.asp

IAS Series
Band / Volume 40
ISBN 978-3-95806-392-1

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum