Exascale computing and data architectures for brownfield applications

Martin Bobák
Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9, SK-845 07 Bratislava, Slovakia
martin.bobak@savba.sk

Adam S. Z. Belloum
Informatics Institute
University of Amsterdam
P.O. Box 94323, 1090 GH, Amsterdam, Netherlands
a.s.z.belloum@uva.nl

Piotr Nowakowski, Jan Meizner, Marian Bubak
ACC Cyfronet AGH
AGH University of Science and Technology
ul. Nawojki 11, 30-950 Krakow, Poland
{p.nowakowski, j.meizner}@cyfronet.pl,
bubak@agh.edu.pl

Matti Heikkurinen
University of Applied Sciences Western Switzerland (HES-SO)
Sierre, Switzerland;
Ludwig-Maximilians Universitaet
Munich, Germany
matti.heikkurinen@hevs.ch

Ondrej Habala, and Ladislav Hluchý
Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9, SK-845 07 Bratislava, Slovakia
{ondrej.habala, ladislav.hluchy}@savba.sk

***Abstract*—Despite the recent dramatic advances in the computational and data processing capacities of the commodity solutions, a numerous scientific, socioeconomic and industrial "grand challenges" exists that could be solved only through capabilities that exceed the current solutions by orders of magnitude. To demonstrate the feasibility of addressing these problems necessitating processing of exascale data sets, novel architectural approaches are needed. These architectures need to support efficient service composition and balancing infrastructure- and user-centric points of view of exascale infrastructures and services. This combination of bottom-up and top-down approaches aims at narrowing the gap between infrastructure services and paving the way towards future high capacity generations e-infrastructure. The resulting architecture will help us provide computing solutions to exascale challenges within the H2020 project PROCESS[1].**

***Keywords*—exascale computing; exascale data management; architecture; functional design; brownfield applications**

I. Introduction

It is very important in some cases, that previous generation tools is still maintained (e.g. due to a client side). However, it causes many problems during development and deployment of new software systems, because they exist alongside existing (legacy) applications, or software systems. Redesign of an existing architecture is a complex task. Legacy applications increase a complexity of the environment, re-engineering could mean "waterfall changes", and so on which are the main source of project failures. Thus, the development is focused on integration and migration instead of new functionality creation. Brownfield approach offers an abstraction that simplifies system complexity which is often used during design of new software architecture. In the paper, a term brownfield applications refers to existing (legacy) applications.

The concept underpinning the paper is a modular exascale service platform which supports combining of data and computational services on top of existing European research e-infrastructures and other HPC computing platforms, and coordinating them through advanced mechanisms for efficient data processing and service provision. It is applicable in a wide range of approaches to exascale and big data services: from highly technical ones with unprecedented data challenges (e.g. Square Kilometre Array[2]) to approaches which address a highly heterogeneous, global user community (e.g. United Nations International Strategy for Disaster Reduction[3]) where the end user may not even be aware that he or she is using a "big data" solution. The technical solutions will aim at converging to a shared architectural framework and corresponding software package.

The key challenge will be addressed is the nature of modern applications and the ever-increasing volume and complexity of datasets, with the associated tough infrastructural demands in terms of efficient, scalable (capacity) and flexible data movement, sharing and access between distributed application tasks, users and storage and computing resources. In some cases multiple federated infrastructures need to be used either

[1]PROCESS project homepagehttps://www.process-project.eu/

[2]SKA homepage https://www.skatelescope.org/

[3]UNISDR homepagehttps://www.unisdr.org/

to provide the capacity and variety of services required by the user communities or due to the fact that resources (massive data stores, sensor networks) are tied to specific physical locations.

Our aim is to move the upcoming exascale data tools and technologies towards service-oriented computing and cloud computing, while retaining their capacity to answer unprecedented data challenges and fulfil the requirements and constraints of existing simulation and computational packages. Our approach is to define reliable and scalable techniques for service composition to enable application with exascale data sets, including both infrastructure-centric and user-centric points of view. Thorough the approach, the proposed architecture is able to optimize capacity, capability, throughput, response time, and so on. Performance improvement depends on a set of supported (brownfield) applications. The result is an architectural framework that can be tuned to address requirements that are orthogonal with each other (in a general sense). This combination of bottom-up and top-down approaches aims at narrowing the gap between infrastructure services while addressing specific requirements of the most demanding exascale applications, paving the way towards future high capacity generations e-infrastructure.

The following assumptions about future large-scale data processing systems have been made:

- Massive data rates (e.g. in the case of SKA) should not be considered "outliers" with relevance only to the specific research questions they address. Rather, they should be seen as precursors of the future Internet traffic, with data rates driven e.g. by explosion of Internet of Things (IoT) devices with more and more fine-grained and complex communication capabilities.
- Merging of IoT and machine learning (or Artificial Intelligence) applications will create new opportunities and challenges. Many of them will be based on emergent phenomena that are almost impossible to predict and model in advance.
- Dealing with the above phenomena necessitates development of big data services that are at the same time efficient, flexible and intuitive to use. Future data challenges will be inherently transdisciplinary and often intersectoral, which means that understanding the emergent behaviour of the system cannot rely on in-depth understanding of all specialities involved.
- Creating and deploying machine/deep learning scalable services on top of e-infrastructures is essential to provide support for clustering, classifying and prediction in individual use cases and will bring "smart solutions everywhere".

Exascale data presents several capability, capacity, compatibility and usability challenges. The size of the datasets involved mean that it is often impossible to move the data to the computing centre. Instead, computation needs to be performed in a distributed manner, with the processing happening close to the data. Thus, the infrastructure needed for exascale data processing is essentially distributed and federated in nature.

Data federation is a well-known approach to aggregating data located in independent sources and provisioning it to applications and users under a unified view [1]–[3]. As data is generated by different groups, institutes and disciplines it is often stored in varying formats, in geographically distributed locations which belong to different administrative and security domains. In addition, data owners often apply different technologies and platforms to store their data sets. Common storage platforms include data grids and storage clouds. In certain disciplines data grids are used to access, modify and transfer large amounts of data which is replicated in distributed and independent resources. Data grids rely on a variety of storage technologies, many of which are difficult to use. Storage clouds store data in logical containers and are replicated on several resources. The storage resources are typically owned and managed by one hosting company. As each cloud storage framework exposes custom and in many cases proprietary Application Programming Interfaces (APIs), users often find their data "locked" in a specific framework. Scientific applications often need to combine several datasets and make them available to tasks that are created from legacy applications capable of only reading data from their local disk.

In the case of data-intensive applications composed of multiple tasks that consume or produce large data sets, making data available through a single entry point greatly simplifies the development of these applications. The single entry point also ensures a consistent shared global namespace. However, this single entry point can be a significant bottleneck in terms of application performance. Distributing access is beneficial for performance reasons but contradicts the principles of data storage federation. Data storage federations have to address consistency, availability and partitioning (CAP) theorem where any networked shared-data system can only satisfy two of the three properties: consistency, availability and partition tolerance. The targeted reference architecture defines the basic blocks for data access and exchange and should satisfy the following requirements:

- **Minimize dependencies** (loosely coupled design): system components should be largely independent of each other. Loosely coupled systems are flexible, can adapt to changes and may take advantage of many different technologies.
- **Scalable**: systems should be able to gracefully handle increases in data volume and demand spikes.
- **Hide system complexity**: data access and exchange architectures should conceal their complexity from user applications and developers to provide easy-to-use solutions [4].
- **Brownfield oriented**: this term is borrowed from urban planning where development takes place in an area with existing structures. In computer science it refers to the development of systems that take into consideration legacy applications or systems [5].

Many exascale applications require complex data processing which includes requirement for advanced calculation environments supporting activities such as modelling, simulation, algorithms for data analysis (e.g. pattern recognition), visualisation etc. Therefore the main issues are as follows:

- **Support workflow processing combining heterogeneous sets of software and data.** Solution needs to support building advanced, ad-hoc data analytics applications from components of different origins. To support this requirement, there is a need to extend web-based notebooks (e.g. Jupiter) with the solutions developed in environments enabling researchers to seamlessly embed chunks of executable code, each processed by a different software packages.
- **Service-oriented application support that does not require any installation on the part of the end user.** This requirement includes access to HPC resources, where we combine web notebooks with ability to access HPC resources using standard REST protocols. This requirement is supported by leveraging cloud hosting capabilities and web-based access methods. Additionally, the feasibility of using lightweight containers (e.g. Docker) has to be investigated.
- **Bring exascale capacities and capabilities to interactive environments** for the creation and sharing of executable documents for advanced data analysis (such as Jupyter) that allows collaborating users to share not only input and output data, but also the calculations themselves. This requirement is addressed by investigating and extending the sharing capabilities of web-based notebooks.

II. Data Centric Scientific experimentations

Many different approaches have been proposed to address the problem of delivering of large data sets across geographically distributed and heterogeneous storage systems. Most of these approaches aim to federate access to data distributed over independent data storage resources. The federation layer defines the rules and the protocols data storage, replication and migration.

An extensive work has been done in the area of federate storage trying to unify access to data files. Often the proposed solutions are too specific to the underlying storage technology. The proposed solutions are not flexible enough making them sensitive to hardware changes. Another important limitation of the existing solutions to federate storage is the ad-hoc interfaces provided to both users and applications. Globus online [6] is a service that federates a set of geographically distributed GridFTP servers allowing the movement of data among these services, but it is only limited to GridFTP and does not work with other technologies.

Another example of federate storage is RACS which is based on Cloud technology allowing users to avoid storage vendor locking, but works only with Amazon S3.

The VPH community implemented PhysiomSpace as a solution of federated storage, PhysiomSpace is designed specifically for managing medical data in various formats [7]. to achieve this goal PhysiomSpace use a custom developed client application which reduce its general use and require additional effort for maintaining the client application. As for the backend, PhysiomSpace works only with Storage Resource Broker (SRB). VeilFS works with different storage system hosted by multiple organization [8], it uses a custom access protocol limiting its usage to specific and dedicated VeilFS clients.

III. Federation Challenges

Many challenges are facing the implementation of a proper storage federation. Mainly these challenges are related to the infrastructure that manage and control computational and storage resources like grids and clouds. Data grids have proven to be able to handle large data movements as well as data replication across autonomous storage resources. Data grids supports various storage technologies like GridFTP, SRB, iRODS. The Cloud systems follow a different approach offering a unified access through the logical containers provided by many cloud storage data [9]. A more recent approach in the Cloud landscape implement a multi-cloud solution of federation challenges (e.g. UberCloud-on the service side, or SlipStream as a technical solution). Currently, the technical solutions and sustainable business models seems to be ways away. In practice, storage owners provide often APIs to promote usage of their storage resources. But often these APIs are inconsistent, subject to frequent changes, and not self-descriptive which complicate further the implementation of the federation layer. Because a number of these APIs are proprietary the possibility to get into vendor lock-in issues is likely to happen [10]–[12]. In such cases, the replacement of the entire storage stack is needed to maintain the data federation access, which is not desirable or even possible [13]. The multiplicity of proprietary or non-standardized APIs impose the use multiple client applications to perform simple operations of datasets. To complicate further the data management process, tasks composing scientific applications are legacy applications can only access local files. In many situation such a data management model is neither effective nor scalable [13].

IV. Data management services landscape

There are a number of data management services available for the researchers to user either for free as a community service like EUDAT (maintaining 1 PB online storage in the form of disks and 9 PB offline storage in the form of tapes)[4] and EGI (maintaining 300 PB online storage in the form of disks and 346 PB offline storage in the form of tapes)[5] services or as commercial services such as globus-online. The existing services are dealing mostly with data transfer, data sharing and synchronization, authorisation, security and privacy. The available data services are targeting end-users providing easy-to-use interface but do not provide APIs which allow to use them from third party systems. The technology used to develop

[4]EUDAT homepage https://www.eudat.eu
[5]EGI homepage https://www.egi.eu

the services is not always know to the public, which makes it difficult to envisage any re-use of these data services as sole building blocks of further services. Beyond the fact that the available services mention in their respective documentations and web sites that they are fit for large data management there is no study which can backup this claim. Thus these services would represent unacceptable risk when considering applications that deal with exascale data.

V. Data management tools and technology for distributed systems

OwnCloud is used in many projects and data management services, it is a self-hosted file sync and share server. It provides access to data through a web interface, sync clients or WebDAV while providing a platform to view, sync and share across devices easily. ownClouds open architecture is extensible via a simple API for applications and plugins and it works with any storage. OwnCloud is often used and promoted as an open source replacement for dropbox, however a study from CERN has shown a few scalability issues with Owncloud [14].

iRODS is a middleware layer that sits above the file systems that contain data, and below domain-specific applications. Because iRODS has a plugin framework and is technology-agnostic, it provides insulation from vendor lock-in. System administrators can slide iRODS on top of an existing heterogeneous data infrastructure and construct a flexible data grid. As middleware, iRODS allows administrators to track and control access to the data under their care; administrators can also monitor the status of iRODS deployment. iRODS is a interesting middleware technology but it comes a monolithic software with a lot of functionalities and will be quite hard to isolate or replace a subcomponents if it does not fit with the requirements of the exascale data applications.

The InterPlanetary File System (IPFS) is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. In some ways, IPFS is similar to the Web, but IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository. IPFS is becoming a new major subsystem of the internet. If built right, it could complement or replace the HTTP and some other protocols.

VI. Proposed approach

The state-of-the-art analysis in the previous chapters has led us to identifying the following three issues as most crucial ones to address:

- Separation of data transfer and launching of the computation
- Traceability of data evolution
- In-situ processing and pre-processing

The data-centric nature of scientific applications requires that research infrastructures (RI) support efficient, scalable and flexible data movement. Service orientation offers an appealing paradigm for developing scientific applications by facilitating interoperability and flexibility, as it allows developers to encapsulate the internal implementation of an application and provide a descriptive interface of its methods and data types. This approach makes it easy to develop services which aggregate other services in a workflow-like fashion, often following well-known coordination models, namely orchestration and choreography. In service orchestration all data is passed to a central coordinator before it is delivered to downstream services, while in the choreography approach data is delivered directly to the consuming services. In both coordination mechanisms significant performance issues occur when large datasets are exchanged via the invocation protocol. Using temporary intermediate storage when moving data between services can solve this performance issue but it considerably slows down the overall application execution. Moreover, this approach creates unnecessary load on storage resources. While investigating the state of the art in data-centric computing we have identified three main challenges to address in the paper:

- The invocation protocols used by services are not suitable for transferring significant volumes of data as they mix the invocation and actual data transfer. New data delivery models need to be researched where the invocation protocol is separated from data movement with the aim to reduce the execution time of workflows, especially in the case of streaming applications. The problem will become more challenging assuming data is distributed across RIs, loosely coupled, and stored in a variety of storage resources ranging from a simple file system to heterogeneous cloud storage.
- The processing of data adds information to data which increases knowledge about the data. A pronounced difference exists in the various data usage scenarios i.e. shared vs. private data. Simple batch systems assume the data processing tasks are independent of each other and thus do not preserve any order. This can be problematic with intra-dependent tasks (such as in scientific workflows). Within Scientific problem solving Systems, models of computations vary too a dataflow model of computation runs tasks only when all data is available for that task, while a Petri-net model runs tasks depending on token transmission as a means of flow control. The shift towards data-centric computing means that data processing needs to be managed alongside the management of computational tasks.
- Transferring exascale data sets through internet may remain impossible in the foreseeable future, therefore methods which will allow to preprocess such data in-situ (or close to where they are stored) need to be investigated, designed and evaluated.

The main result of the paper bases its developments on an architectural conceptual model, which is presented bellow in Figure 2, helping to align developments in each of the "service component clusters":

- Computing and storage infrastructure, responsible for optimizing the performance of the underlying infrastructure while providing consistent interfaces to physical

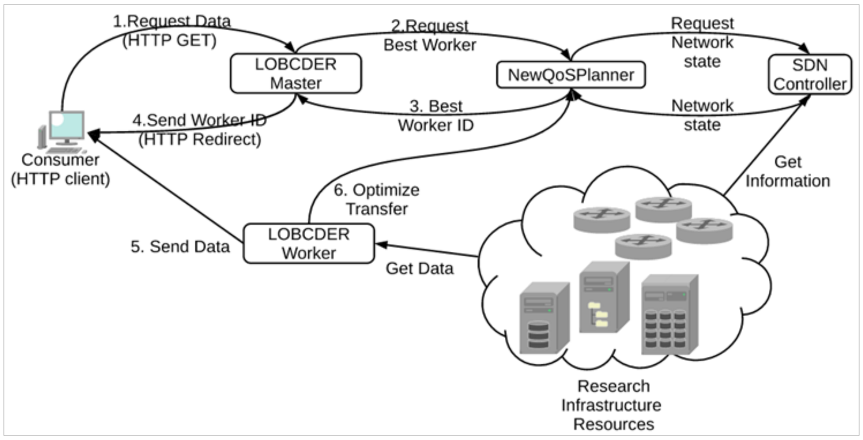

Fig. 1. Interaction of the main SDN components. When a consumer (e.g. HTTP client) requests data LOBCDER provides the data transfer from heterogeneous storage backends, while the the NEWQoSPlanner aggregates information collected from the network, reserves paths and controls the flows via SDN controllers [16]

computing and data resources to the higher level services.

- SOA/Cloud solutions, supporting mapping of the "business processes" to mature, well-documented and supported software and service components.

Unlike existing proposals, the approach followed in the paper is based on the LOBCDER technology which is agnostic from the backend storage available technologies to offer a large-scale collaborative storage environment. It does not assume any central control of the back end storage an thus by design it fit the federated architecture principles, and it does not require to use a specific client, any client which use WebDav protocol can be used on top on the LOBCDER backend. LOBCDER offers a Virtual File System like abstraction, which scales and operates over federated heterogeneous distributed systems (no central control). LOCDER is able to federate multiple, heterogeneous and independent storage resources and present them as a unified storage space. Some of the basic features provided by LOBCDER like the access transparency, file management policy, actual file transfers and the location transparency can be found across multiple services like EUDAT set of services B2ACCESS, B2DROP, B2SAFE, B2STAGE, and B2FIND, or the EGI online storage and data transfer services. On top on these basic services LOBCDER offer more advanced services like Replicas management, Migration management and support in case of concurrent access to files coherency and partitioning [15]. The LOBCDER system has also network services which allow in case of programmable networks (SDN) to optimize the speed of large data transfers (see fig. 1), by either selecting the best replica or rerouting the traffic in case of network congestion [16]. These advanced services can be easily ported to the EUDAT service eco-systems as the LOBCDER has a flexible Plug-In Architecture which allows to substitute any of the basic services.

According to current situation of shared computing notebooks, we propose to integrate them with a robust remote process controller which simplifies user interaction with remote HPC servers. It allows to execute application in a batch or interactive mode where application output can be fetched online and new input can be sent using a simple REST interface. This solution would support efficient creation and sharing of executable documents for analysis of heterogenous research datasets.

The main challenges in processing of exascale data which we want to tackle in the paper are the extreme size of the data, often preventing its movement to a remote computation location, and the complexity of the structure of the data, requiring extensive domain knowledge in order to be able to use it effectively [17]. This forces users of exascale data to perform computations in situ, and only the biggest research institutions and the best funded projects can afford to acquire access to an infrastructure able to house both the data and the extreme computational resources that can process it. Additionally, the expert domain knowledge required to be able to make sense of the data is often hard to find. These requirements make exascale processing currently prohibitively expensive for smaller players.

There is a need of an architecture which alleviates some of these problems and brings exascale data closer to smaller research organizations and even to individual researchers. The architecture must be able to decouple access to the data from their use in computation, and to provide a simplified access to the separate components of the data.

In future, we will also design a reference architecture that not only addresses exascale data management challenges related to infrastructure owned by the consortium but also data management challenges across existing Research Infrastructures (RI). RIs plays an important role in establishment or the European research landscape, they offer services distributed across Europe, hosted on a diversity of infrastructure including grids, clouds and HPC clusters. To support the usage of RI and make it easy to use by scientists with various background a number of high level services are used to manage CPU and data intensive computations over this highly distributed infrastructure. One of the main forms of scientific application that are currently deployed over RIs are the so-called scientific workflows. These workflows can be formulated as DAG where the nodes are computation tasks and the edges represent the data dependencies. The data dependencies are the result of file exchange [18], [19] which is enabled by Distributed File Access Services (DFASs) which abstract the low-level details providing single unified view for the management of data files which facilitate the implementation of a number of file managements services such as file replication, caching, and discovery. In some case, data management is further complicated by privacy and security constraints limiting the possibility to move the data across remote sites where significant and specialized computing resources are available [20], [21].

Often scientific workflow transforms datasets according to a well-defined series of data processing tasks which can be geographically distributed until the final and targeted results is reached [22]. In such processing model consistency is of crucial importance and have to be support by the used DFAS. In general, commonly used DFAS implement standardized protocols allowing to on one hand decouples the development of client software [23], and on the other hand hides a network detail. DFAS allow to perform remote Data operations in a similar way local data operation are performed. These feature of the DFAS are the bases for the development of Distributed File Access Services (DFASs) which help to scale the computing infrastructure when the load increases. In practice, DFAS are not taking advantage of the capabilities offered by the RIs. The lack of interaction between the DFAS layers and the network devices (switches, routers, etc.) prevent the optimization of data transfers to reach the required QoS. One reason for this lack interaction is the existence of a multitude of protocols and interfaces which are used to configures the network devices [24]–[26].

From the viewpoint of computations, the focus of the architecture facilitates simulation, modelling and development of mathematical methods and tools for exascale data processing. This, in particular includes:

- services that enable execution of simulations models as interactive workflows using heterogeneous software, including modern languages for scientific computing.
- services that facilitate access to HPC and cloud resources by using well-defined standards.
- services that facilitate programming for multicore architectures.
- benchmark and monitoring services.

A. Leveraging locally-deployed services to access exascale data

Data in scientific applications often come in bulky packages - files often containing not only the information the application seeks, but additional data as well. For example, when working with geographically-located data, these are typically divided into a grid, with each part of the grid stored in a separate file. Sometimes the files may be unnecessarily big for the application, encompassing a much larger area than is needed - the application needs to access and transfer more data than it will actually use. This same problem applies to layers in data - a dataset may contain many layers, with each layer containing different aspect of the data. The application will again need to download the whole dataset even if it requires only one layer (aspect) of it. This necessity to transfer data over the network in whole packages and only then extract the actually needed items produces excess network traffic and consumes resources - network bandwidth and processing time. One of the solutions to this problem is to preprocess the data locally, near to where they reside, and transfer over the network only the actually needed parts. The three main ways of placing the preprocessing stage are either (a) (as is usually done) to co-locate it with the actual computation, (b) to co-locate it with the data - the most compelling option from the point of view of conserving network bandwidth, or (c) to use an intermediate place, close to the data storage facility, and transfer the preprocessed data from there. While option (b) is usually superior to option (c), often our ability to deploy the pre-processing program to the actual storage facility is often very limited by administrative boundaries, lacking computational capacity of the storage facility or security concerns. In these instances, locating a computational facility able to run the preprocessing program near the data is an option much better than the default (a) - downloading data en bulk and preprocessing it at home.

To handle the deployment of workflows to remote places, a system of service containers, data processing services (both generic and specific to our pilot applications) and services for data access and integration are necessary. They allow to use the deployed preprocessing services, connect them into a single process, and also reuse the logic of the preprocessing workflow over many instances of computations.

B. Description of the proposed architecture

According to the requirements analysis, the main demands placed on the architecture are: an exascale data-capable, service-oriented, and cloud-based architecture allowing exascale computational research even for organizations which do not possess extreme resources. Considering the demands, a modular open-source architecture is proposed (Fig. 2). It is divided into three main modules: (1) exascale data module (green boxes), (2) exascale computing module (red boxes), and (3) service orchestration module (blue boxes, it includes a user interface).

Users access the architecture through an application-oriented scientific gateway which provides secure access in the form of an interactive environment. The environment is represented as a script application programming interface (API), command-line interface (CLI), and graphical user interface (GUI). Authentication and authorisation of the user is performed by a dedicated service (e.g. EGI AAI Check-in). Federated authentication and authorisation are issues that need to be solved, but need additional analysis in terms of emerging trust network structures and growing privacy and security concerns.

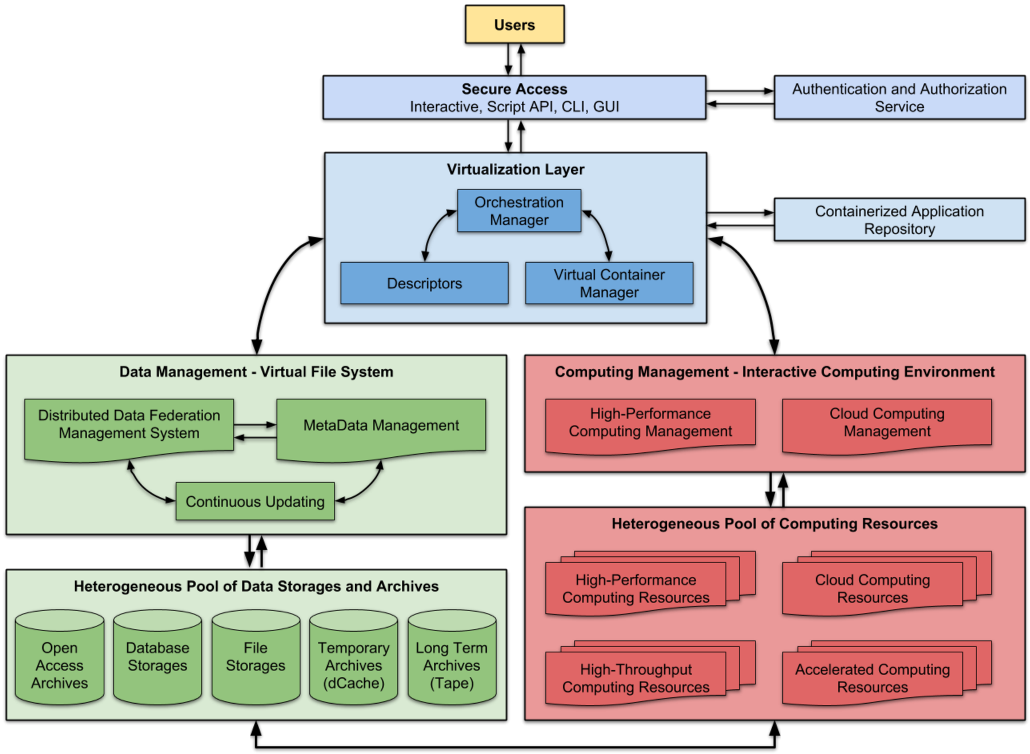

Fig. 2. Core elements of the functional design.

In order to alleviate the problems with processing exascale data, each of use case scenarios will be expressed as a set of containers or virtual machines. Commonly, workflow/scenario steps consist of applications that have enormous dependencies. This approach will unlock the full potential of the infrastructure, and also provide a secure and isolated environment. However, it implies virtualization/containerisation of an application environment as well as its execution environment. The architecture will provide a containerized application repository, which will offer a container or virtual machine per workflow/scenario step.

The core of the platform services is the virtualization layer. This component will be responsible for deployment of application services. Once the services are deployed, they will be orchestrated by high-level descriptions automatically. This element will be responsible for resource allocation, configuration of all services and management of their lifecycle.

To reach an exascale-capable infrastructure, interoperability across multiple computing centres is essential. The whole infrastructure is divided into two parts: (1) an extreme large data service-oriented infrastructure, and (2) an extreme large computing service-oriented infrastructure. The first part is modelled as a distributed virtual file system (DVFS), an abstraction layer on top of a more concrete file system. It allows applications to access different types of file systems (within a data storage federation) in a uniform way. It also supports access to files from multiple hosts via a computer network. The main requirements are to fit the federated architecture principles andto avoid forcing users to rely on any specific client. DVFS also includes a manager for metadata and supporting data services (such as monitoring).

The second part is responsible for management of computing resources. It is modelled as an interactive computing environment, allowing to access different computing resources in a uniform manner. It supports extreme large computations that require heterogeneous infrastructures. It offers HPC computations, HTC computations as well as cloud computations, and supports accelerators.

VII. Conclusion

In the paper current challenges related to processing of exascale data have been presented. According to them a solution in form of functional design of the exascale-capable architecture has been proposed.

The architecture is based on modular exascale service platform which supports combining of data and computational services, and service-oriented applications that do not require any installation on the part of the end user. Its main features are effectiveness, scalability (capacity) and flexible data movement, sharing and access between distributed application tasks, users and storage and computing resources. The design also takes into account minimisation of dependencies and hiding of system complexity. Finally, it paves the way to next-generation workflow processing solutions for heterogeneous, exascale data processing.

ACKNOWLEDGMENT

This work is supported by projects EU H2020-777533 PROCESS, and VEGA 2/0167/16.

REFERENCES

[1] P. Kacsuk, G. Kecskemeti, A. Kertesz, Z. Nemeth, A. Visegradi, and M. Gergely, "Infrastructure aware scientific workflows and their support by a science gateway," in *Proceedings - 7th International Workshop on Science Gateways, IWSG 2015*. Institute of Electrical and Electronics Engineers Inc., August 2015, pp. 22–27.

[2] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and Architectures for Parallel Processing*, C.-H. Hsu, L. T. Yang, J. H. Park, and S.-S. Yeo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 13–31.

[3] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," in *2010 IEEE 3rd International Conference on Cloud Computing*, July 2010, pp. 337–345.

[4] J. Gray, D. T. Liu, M. Nieto-Santisteban, A. Szalay, D. J. DeWitt, and G. Heber, "Scientific data management in the coming decade," *Acm Sigmod Record*, vol. 34, no. 4, pp. 34–41, 2005.

[5] B. Boehm *et al.*, "Applying the incremental commitment model to brownfield systems development," in *7th Annual Conference on Systems Engineering Research, CSER 2009*, April 2009.

[6] I. Foster, "Globus online: Accelerating and democratizing science through cloud-based services," *IEEE Internet Computing*, vol. 15, no. 3, pp. 70–73, 2011.

[7] D. Testi, P. Quadrani, and M. Viceconti, "Physiomespace: digital library service for biomedical data," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1921, pp. 2853–2861, 2010.

[8] Ł. Dutka, R. Słota, M. Wrzeszcz, D. Król, and J. Kitowski, "Uniform and efficient access to data in organizationally distributed environments," in *eScience on Distributed Computing Infrastructure*. Springer, 2014, pp. 178–194.

[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[10] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," in *Proceedings of 2nd International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2011. IARIA*, vol. 2011, September 2011, pp. 32–38.

[11] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. IEEE, April 2010, pp. 27–33.

[12] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*. IEEE, August 2009, pp. 44–51.

[13] W. Kuchinke, C. Ohmann, Q. Yang, N. Salas, J. Lauritsen, F. Gueyffier, A. Leizorovicz, C. Schade-Brittinger, M. Wittenberg, Z. Voko *et al.*, "Heterogeneity prevails: the state of clinical trial data management in europe-results of a survey of ecrin centres," *Trials*, vol. 11, no. 1, p. 79, 2010.

[14] J. T. Mościcki and M. Lamanna, "Prototyping a file sharing and synchronization service with owncloud," in *20th International Conference on Computing in High Energy and Nuclear Physics (CHEP2013) IOP Publishing, Journal of Physics: Conference Series*, vol. 513, no. 4. IOP Publishing, 2014, pp. 34–42.

[15] S. Koulouzis, A. Belloum, M. Bubak, P. Lamata, D. Nolte, D. Vasyunin, and C. de Laat, "Distributed data management service for vph applications," *IEEE Internet Computing*, vol. 20, no. 2, pp. 34–41, 2016.

[16] S. Koulouzis, A. S. Belloum, M. T. Bubak, Z. Zhao, M. Živković, and C. T. de Laat, "Sdn-aware federation of distributed data," *Future Generation Computer Systems*, vol. 56, pp. 64–76, 2016.

[17] S. Ashby, P. Beckman, J. Chen, P. Colella, B. Collins, D. Crawford, J. Dongarra, D. Kothe, R. Lusk, P. Messina *et al.*, "The opportunities and challenges of exascale computing–summary report of the advanced scientific computing advisory committee (ascac) subcommittee," *US Department of Energy Office of Science*, pp. 1–77, 2010.

[18] R. Agarwal, G. Juve, and E. Deelman, "Peer-to-peer data sharing for scientific workflows on amazon ec2," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*. IEEE, 2012, pp. 82–89.

[19] L. Wang, J. Tao, R. Ranjan, H. Marten, A. Streit, J. Chen, and D. Chen, "G-hadoop: Mapreduce across distributed data centers for data-intensive computing," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 739–750, 2013.

[20] M. Vairavanathan, S. Al-Kiswany, A. Barros, L. B. Costa, H. Yang, G. Fedak, Z. Zhang, D. S. Katz, and M. Wilde, "A case for workflow-aware storage: An opportunity study using mosastore,," *Submitted to FGCS Journal*.

[21] R. Tudoran, A. Costan, O. Nano, I. Santos, H. Soncu, and G. Antoniu, "Jetstream: Enabling high throughput live event streaming on multi-site clouds," *Future Generation Computer Systems*, vol. 54, pp. 274–291, 2016.

[22] A. Simonet, G. Fedak, and M. Ripeanu, "Active data: A programming model to manage data life cycle across heterogeneous systems and infrastructures," *Future Generation Computer Systems*, vol. 53, pp. 25–42, 2015.

[23] Storage Networking Industry Association, "Cloud data management interface," *Tech. Rep*, 2010.

[24] J. Chen, A. Choudhary, S. Feldman, B. Hendrickson, C. Johnson, R. Mount, V. Sarkar, V. White, and D. Williams, "Synergistic challenges in data-intensive science and exascale computing: Doe ascac data subcommittee report," 2013.

[25] M. Kluge, S. Simms, T. William, R. Henschel, A. Georgi, C. Meyer, M. S. Mueller, C. A. Stewart, W. Wünsch, and W. E. Nagel, "Performance and quality of service of data and video movement over a 100 gbps testbed," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 230–240, 2013.

[26] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.