HPC-as-a-Service via HEAppE Platform

Vaclav Svaton, Jan Martinovic, Jan Krenek, Thomas Esch, Pavel Tomancak

Abstract The HPC-as-a-Service concept is to provide users with simple and intuitive access to a supercomputing infrastructure without the need to buy and manage their own physical servers or data centers. This article presents the commonly used services and implementations of this concept and introduces our own in-house application framework called High-End Application Execution Middleware (HEAppE Middleware). HEAppE's universally designed software architecture enables unified access to different HPC systems through simple object-oriented web-based APIs, thus providing HPC capabilities to users without the necessity to manage the running jobs forms the command-line interface of the HPC scheduler directly on the cluster. This article also contains the list of several pilot use-cases from a number of thematic domains where the HEAppE Platform was successfully used. Two of those pilots, focusing on satellite image analysis and bioimage informatics, are presented in more detail.

1 Introduction

HPC-as-a-Service is a well-known term in the area of high performance computing. It enables users to access a High Performance Computing (HPC) infrastructure without the need to buy and manage their own infrastructure. Through this service,

Vaclav Svaton, Jan Martinovic, Jan Krenek
IT4Innovations, VSB Technical University of Ostrava, Ostrava, Czech Republic
e-mail: (vaclav.svaton, jan.martinovic, jan.krenek)@vsb.cz

Thomas Esch
German Aerospace Center, Earth Observation Center, Oberpfaffenhofen, Germany
e-mail: thomas.esch@dlr.de

Pavel Tomancak
Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany
e-mail: tomancak@mpi-cbg.de

academia and industry can take advantage of the technology without upfront investment in the hardware. This approach further lowers the entry barrier for users who are interested in utilizing massive parallel computers but often do not have the necessary level of expertise in the area of parallel computing.

Nowadays, very few companies or scientific research groups utilize the capabilities provided by Cloud [2, 9] or HPC [16] infrastructures. For this reason, there is a number of service providers who offer their application frameworks, middlewares, or APIs [14] to their users to improve their knowledge about this technology and mainly to provide them with an easy-to-use access to HPC computational resources or Cloud [17, 13].

The major representatives of the commercial service providers are Google LLC with *Google Cloud* [7] and Amazon Inc. with the *Amazon Web Services (AWS)* [6, 8]. For academic use there is *EMU-Cluster* [4] hosted by eResearch South Australia.

Google Cloud service provides complex brand of services ranging from HPC computations and big data analysis to machine learning use cases. *Google High Performance Computing* service provides a number of computation options to meet the standards of the users (multi-instance or single-instance virtual machines). *Google Compute Engine* is a product which provides customizable Virtual Machines (VMs) with a number of features and the option to deploy a user code or application directly to a VM or via a container management system.

Amazon Web Service are web-based cloud services that provide computing services for all types of data. This service enables the users to utilize the computational environment based on the virtual infrastructure or HPC infrastructure. AWS also provides intuitive workflow manager *AWS Step* for the ability to create computational pipelines consisting of a number of interconnected computational tasks. The actual resource allocation is selected automatically based on the chosen block of operations and the type of the input data. AWS Step also provides the information about the state of every running computation from each executed pipeline.

For academic purposes and eRSA researchers, *eRSA EMU-Cluster* provides service to access and use HPC computational resources. This service uses *Torque job management system* to submit and monitor the cluster's jobs. This service is able to allocate a maximum of 16 compute jobs (136 cpu cores) for a single user.

The common problem with the above mentioned services and approaches is that they are either commercial services which do not disclose their internal workflows or they are tailored for a specific type of infrastructure. As *IT4Innovations national supercomputing center (IT4Innovations)* is in the role of an HPC provider, there is also a problem with the internal control mechanisms based on *ISO standards* and other *certifications* and *security mechanisms*. It is almost impossible to adopt third-party software as a middleware to access HPC infrastructure directly instead of the users while retaining the same quality of service provided by a standard supercomputer support staff. Due to this reason we have developed our own open-source implementation of an HPC-as-a-Service concept called High-End Application Execution Middleware (HEAppE Middleware) [3]. IT4Innovations currently operates two HPC clusters, and thus also wanted to provide them as a service for users.

This paper is organized as follows. Section 2 describes the overall general architecture of the developed HEAppE Middleware. It also describes the main internal processes and mechanisms regarding the user and cluster accounts, data management, job submission and application interfaces. Section 3 illustrates several use-cases where the HEAppE platform was successfully used, including domains of earth-observation image analysis in the *ESA's Urban Thematic Exploitation Platform* and biological image data analysis in the *SPIM image processing plugin for FIJI*. Finally, Section 4 presents the conclusion, lesson learned, and future work in the context of this project.

2 HEAppE Middleware

IT4Innovations supercomputing center as a HPC provider and DHI Group[1], a nation-wide company developing hydrologic software MIKE[2] powered by DHI, were a collaborators during the design and implementation phase of the HEAppE Middleware (formerly known as an HPC as a Service Middleware). At the beginning of the project, the HEAppE was used in the area of hydrological modelling in a decision support system for crisis management [15]. Since then the HEAppE was successfully used in a number of projects from a different thematic domains like satellite image analysis [10] and biological image analysis [11]. A more detailed description of these use-cases is located in Section 3.

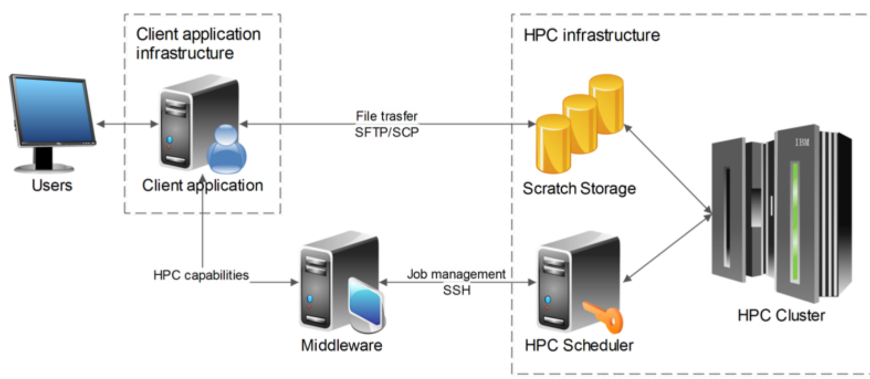

Fig. 1 HEAppE Middleware general architecture

[1] https://www.dhigroup.com/

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

2.1 Middleware General Architecture

There was the need for an architecture, which would allow users to run complex and computationally demanding calculations on a supercomputer directly from the user interface of a client application without the necessity to connect directly to the HPC cluster and manage the jobs from the command line interface of the HPC scheduler. Therefore, IT4Innovations[3] has developed an application framework called HEAppE Middleware. HEAppE manages and provides information about submitted and running jobs and their data between a client application and an HPC infrastructure. This middleware is able to submit the required computation or simulation to the HPC infrastructure, monitor the progress, and notify the user if needed. The Scheme of a general architecture is displayed in Figure 1.

Platform features:

- Providing HPC capabilities as a service to client applications and their users
- Unified middleware interface for different operating systems and schedulers
- Authentication and authorization for the provided functions
- Monitoring and reporting of executed jobs and their progress
- Current information about the state of the clusters
- Job accounting and job reporting for a single user or a user group
- Secure data migration between different jobs
- Prepared job templates for different computational tasks

The internal middleware architecture is separated into a number of interconnected layers. To simplify this architecture, there are three main layers called *ServiceTier*, *BussinesLogicTier* and *HpcConnectionFramework*.

ServiceTier represents high-level view of the middleware provided services. This layer is used for simple integration into different user interfaces or client applications. BussinesLogicTier contains concrete implementation of the provided functions and is therefore invoked from the ServiceTier layer. HpcConnectionFramework contains implementation of direct access to an HPC infrastructure. This could be considered as the lowest layer, which is responsible for the actual job submission and monitoring. The current implementation of the HpcConnectionFramework uses *PBS Manager* [1] as a main HPC job manager. If there is the need to support a different kind of HPC job manager, e.g. *Slurm* [12] which could be available at a specific HPC center the simple extension of this layer is all that is needed to use the HEAppE Middleware in a different HPC center.

Independently of these layers, there are five main categories into which the functions provided by the middleware are divided. These categories are as follows:

- *User and Limitation Management* provides several authorization and authentication services to the users and management of user limitation parameters.

[3] https://www.it4i.cz/

- *Cluster Information* provides information about the state of the connected clusters, available types of processing queues, and cluster node usage.
- *Job Management* ensures comprehensive job management functionality to create, submit, cancel, delete, and monitor the state of computational jobs.
- *File Transfer* provides the users with the necessary methods for data management between the client application and an HPC infrastructure.
- *Job Reporting* is used for reporting statistics. It provides a comprehensive reports for a specific user or a group of users in terms of submitted job and used resources.

2.1.1 User Accounts

HEAppE Middleware introduces an additional layer of abstraction in remote access to an HPC infrastructure from the security point of view. Middleware shields the users from direct access to HPC computing resources and all internal processes that are associated with direct access to a supercomputer and its storage. Therefore, the middleware does the mapping between *external user accounts* and *internal cluster accounts* (see Figure 2). Passwords stored within the middleware's internal database are encrypted using *PBKDF2 salted password hashing* [5].

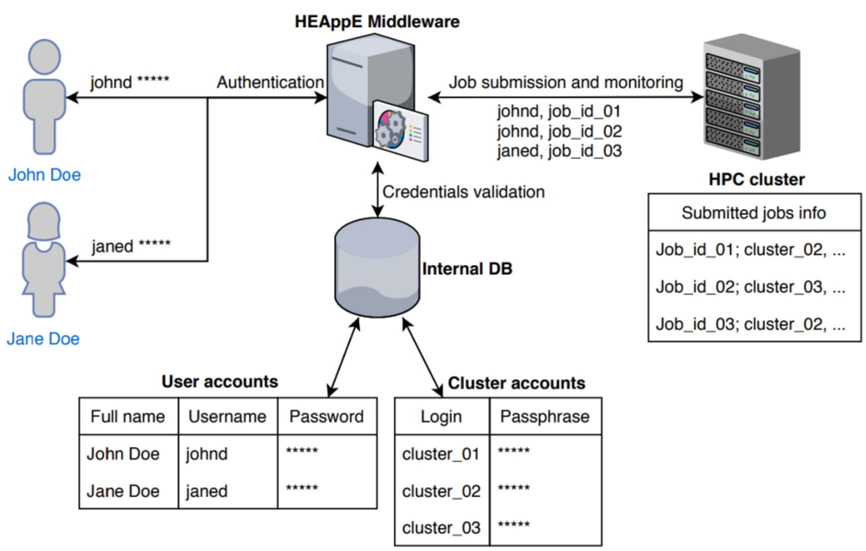

Fig. 2 Middleware's mapping mechanism

External user accounts are provided by HEAppE Middleware to users and is used by them to authenticate themselves via middleware to access its functionality.

Therefore, the user is only able to access the functions offered by the middleware and not the HPC infrastructure itself.

Internal cluster accounts are used by the middleware to access an HPC infrastructure and perform the job submission instead of the user. These accounts are basically the same as the one acquired by the user at the end of the successful standard authentication process within the HPC center.

Middleware keeps track of all submitted and running jobs, and via its monitoring and reporting services it is able to provide the users with an up-to-date information about running jobs and used resources.

2.1.2 Template Preparation

For security purposes, HEAppE Middleware enables the users to run only a prepared set of so-called *Command Templates*. Each template defines an arbitrary script or executable file that will be executed on the cluster, any dependencies or third-party software it might require, and the type queue that should be used for processing (type of computing nodes to be used on the cluster).

The template also contains the set of input parameters that will be passed to the executable script during run-time. This is achieved via *wildcard parameters "%%{}"*. Thus, the users are only able to execute pre-prepared command templates with the pre-defined set of input parameters. The actual value of each parameter (input from the user) can be changed by the user for each job submission.

Id	Name	Description	Code	Executable File	Command Parameters	Preparation Script	Cluster Node Type
1	TestTemplate	Desc	Code	/scratch/temp/ HaasTestScript/test.sh	"%%{inputParam}"	module load Python/2.7.9-intel-2015b;	7

Fig. 3 HEAppE Command Template

Example of command template with one custom user attribute *inputParam* is illustrated in the table in Figure 3. This Command Template allows the users to run only the prepared script *test.sh* which is already prepared on a cluster storage. The *Preparation Script* column defines any third party software or dependency that should be loaded before the actual script execution. The last parameter called *Cluster Node Type* denotes the type of the processing queue to be used on the cluster.

2.1.3 Data Management

The software architecture of the HEAppE Middleware requires the user to copy the input data to a cluster storage before processing and to download or remove the data from the cluster storage when the processing is finished. Each computation job is considered as an independent unit by the middleware.

However, this poses a problem if one computation job is just a part of the processing pipeline where the output of one job is used as an input of the following job. Therefore, there was the need for functionality that would provide the users with a secure way to migrate the data between different computation jobs. Based on this requirement, the middleware enables the users to copy the selected data from a finished computation job to a temporary users storage and to copy this data from the storage to an input of a newly created job thus mitigating the data transfer bottleneck. This methods utilize the secure token called *sessionCode*, which is extensively used by the middleware to authenticate the user as a key for *CopyToStorage* and *CopyFromStorage* functionality. Using the valid token, the user can store the data to a temporary storage and later use the same token to retrieve the data from the storage in a different computational job.

In the case of a standard computational job the user usually needs to upload the input data to a cluster storage for processing. For data management purposes, middleware offers the users the *File Transfer* methods as described in Section 2.1. The purpose of this functionality is to provide a secure way for the user to upload the data. Internally, the middleware creates a *temporary SSH key* for the user to access the job folder in the cluster storage to upload the input data. This data transfer is performed through a standard *Secure Copy Protocol (SCP)* without involvement of the middleware as a bottleneck. After successful data upload, this key is removed from a cluster and the job is prepared for submission to a job management system.

2.1.4 APIs

There is a number of ways to access the functionalities provided by the HEAppE Middleware. The most commonly used type of access is through standard *web services*. This approach includes an easy accessible web interface and programming language interoperability in case that these web services are invoked from a custom client interface, e.g. web page, desktop application, or a mobile application. Web services are simple to use and give the middleware user the freedom to develop his/her own application interface, i.e. wrapper build upon these services. The list of web services available to users via the middleware relates closely to the six main categories of the middleware's functions listed in Section 2.1.

One of the examples of a custom API wrapper is *Jupyter Notebook*[4]. The Jupyter Notebooks are live editable documents that allow mixing of *live* programming codes, equations, formulas, visualizations, and texts. Because of their flexibility and available kernels in many different programming languages they have become useful solution for demonstrations and education. Mixing the explanatory and instructional text with the programming code in Jupyter Notebook documents makes it generally easier for users without comprehensive knowledge about the topic. They

[4] http://jupyter.org/

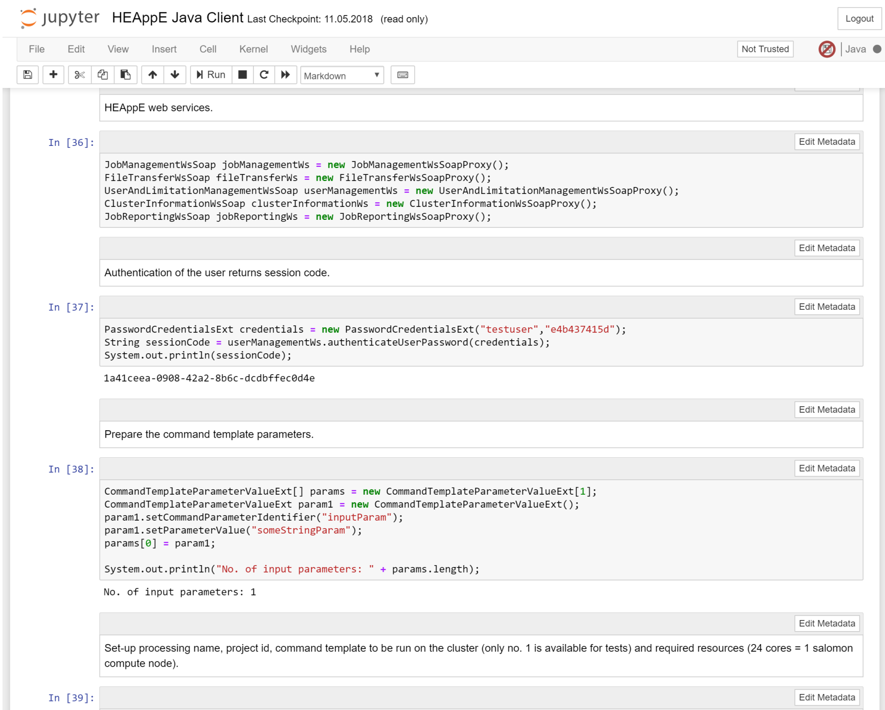

Fig. 4 HEAppE Java Client in Jupyter Notebook

have become popular in various scientific domains like Artificial Intelligence and Machine Learning in science area.

Two Jupyter Notebooks were created as an example for the users to explain the basic abilities of the HEAppE Middleware and to help these users learn how to use it. These notebooks (Java and Python based kernels) were implemented as wrappers on top of the standard web service API. An example of the HEAppE Java Client is shown in Figure 4.

2.1.5 Middleware Deployment

The deployment procedure can be divided into two separate tasks; deployment of the HEAppE Middleware application framework itself and preparation of the HPC infrastructure. Middleware deployment is pretty straightforward. As it is a standard web service application it just has to be set up in a web server environment. Upon a successful deployment to a web server, the middleware's services should be visible on a specific web address. Middleware also automatically creates its own internal database to store the user account information, information about submitted jobs, and much more.

For the middleware to be able to submit jobs to a cluster queue, there is the need for a set of cluster accounts that need to be bound to a specific *computation project*. Therefore, a set of cluster service accounts should be created for a specific computation project ID and project name. These accounts will be used by the middleware (have to be also included into a middleware's database) for the job submission on the cluster's side of the architecture.

The second task consists of cluster environment setup. This mainly includes preparation of so-called *key scripts* that contain various internal mechanisms e.g. job directory creation in a cluster storage, generation of a temporary ssh key for the file transfer methods, etc. These scripts will be invoked by the middleware during the execution of a specific computational job to simply prepare the cluster environment for the job submission.

It is a good practice to deploy independent instance of HEAppE Middleware for each computational project. Every instance contains its own physically separated database with a new set of user credentials used to authenticate the external users via the HEAppE Middleware, thus allowing them to remotely submit or manage their computation jobs and also a new set of internal cluster accounts to separate the computational project's jobs at an HPC infrastructure level.

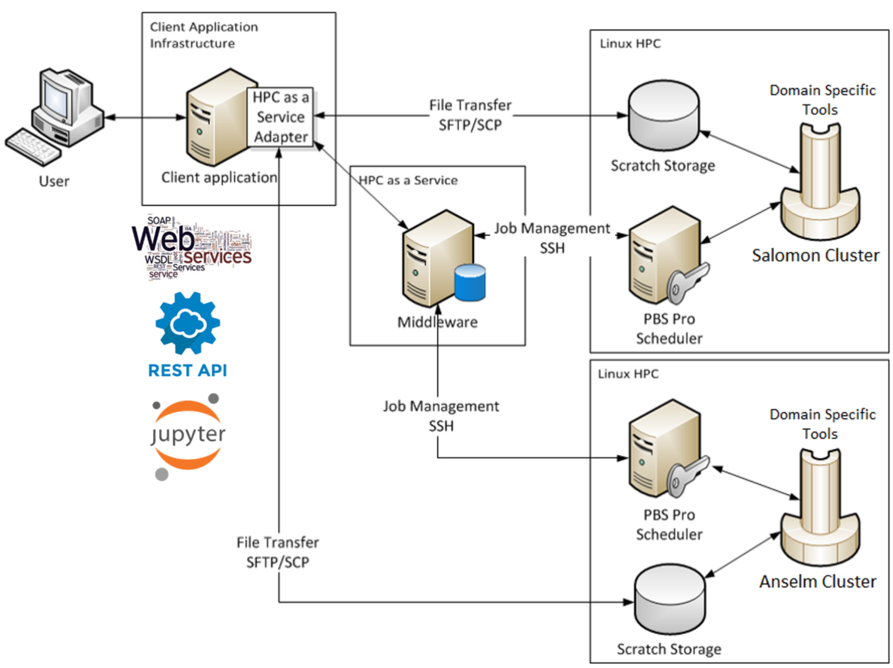

Fig. 5 Deployment of HEAppE Middleware at IT4Innovations

2.2 HEAppE at IT4Innovations

IT4Innovations national supercomputing center operates supercomputers Salomon (2 PFLOP/s) and Anselm (94 TFLOP/s). The supercomputers are available to academic community within the Czech Republic and Europe and industrial community worldwide. Both supercomputers are available to users via HEAppE Middleware. The scheme of the HPC architecture with the Salomon and Anselm clusters is displayed in Figure 5. The detailed description of the IT4Innovations' HPC infrastructure and the hardware specification of both clusters can be found in the official documentation[5].

HEAppE's universally designed software architecture enables unified access to different HPC systems through a simple object-oriented client-server interface using standard web services, REST API, and Jupyter Notebooks. Thus providing HPC capabilities to the users yet without the necessity to manage the running jobs from the command-line interface of the HPC scheduler directly on the cluster.

3 Integration Use Cases

HEAppE Middleware has already been successfully used in a number of public and commercial projects where there is a need to remotely manage computational jobs without directly accessing the HPC infrastructure itself.

HEAppE was used in the crisis decision support system *Floreon+* [15] for What-If analysis workflow utilizing HPC clusters; in the *Urban Thematic Exploitation Platform* (Urban-TEP) [10] financed by the ESA as a middleware enabling sandbox execution of user-defined docker images on the cluster; in the *H2020 project ExCaPE*[6] as a part of a Drug Discovery Platform enabling the execution of drug discovery scientific pipelines on a supercomputer; in the area of molecular diagnostics and personalized medicine in the scope of the *Moldimed project*[7] as part of the Massive Parallel Sequencing Platform for analysis of NGS data; and in the area of bioimageinformatics as a integral part of the *FIJI plugin* [11] providing unified access to HPC clusters for image data processing.

Two of these use-cases are introduced in the following subsections in more detail to illustrate that the user usually sees only the dedicated graphical interface and is completely shielded from the functionality provided by the HEAppE Middleware. *Submitting a job to a supercomputer is as easy as clicking the button on a website.*

[5] https://docs.it4i.cz
[6] http://excape-h2020.eu
[7] https://www.imtm.cz/moldimed

3.1 Urban Thematic Exploitation Platform

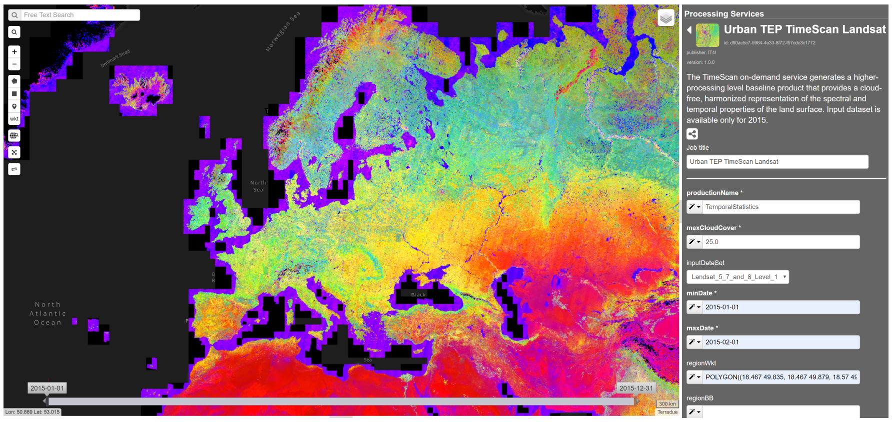

Fig. 6 Urban TEP Platform: Landsat TimeScan product and on-demand processing service

The main goal of the *Urban Thematic Exploitation Platform (Urban TEP)* project is implementation of an instrument that helps address key research questions and societal challenges arising from the phenomenon of global urbanization. The created *Urban TEP* platform[8] is a workplace for a scientific community and commercial subjects alike to access a number of thematically oriented processing services and data products. These products and services are created with a focus on the domain of global urbanization and provide the users with an easy and intuitive way of how to access and analyze satellite image data from a number of space missions.

The main role of the *IT4Innovations* center in the project is to provide state-of-the-art technology and expertise in high performance computing. Thus the IT4-Innovations center provides the platform with the processing services and data storage services needed to access, analyse, and visualize geospatial data and derived products (see Figure 6). Via the platform's processing services the users are able to submit their analysis into the IT4Innovations processing center.

This access to the IT4Innovations' HPC infrastructure is achieved via a simple integration of the HEAppE Middleware into the job management processes of the platform. The results of these remotely invoked computations are automatically visible directly in the geobrowser of the platform immediately after the computation is done, thus completely shielding the users of the web platform from a somehow complicated direct HPC infrastructure access.

[8] https://urban-tep.eu

3.2 SPIM Image Processing Pipeline

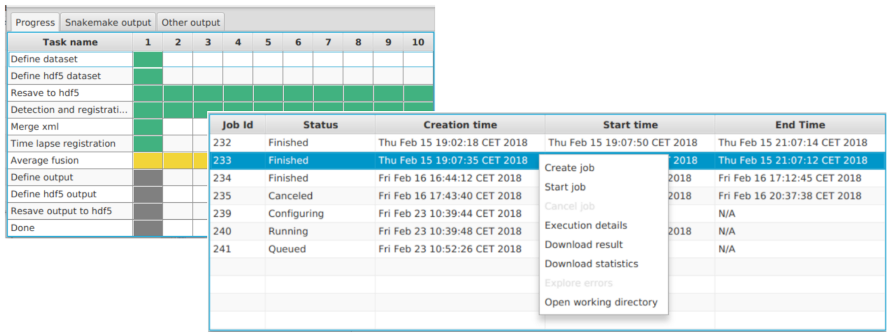

Fig. 7 Fiji plugin: SPIM image processing pipeline interface

State-of-the-art imaging devices, such as light sheet microscopes, produce datasets so large that they can only be effectively analyzed by employing methods of image processing on high-performance computing clusters. To address this issue, an HPC plugin for Fiji[9], one of the most popular open-source software tools for image processing, has been developed. The plugin enables end users to make use of HPC clusters to analyze large scale image data remotely and via the standard Fiji user interface (see Figure 7).

This Fiji plugin utilizes the HEAppE Middleware for remote execution of *SPIM image processing pipeline* on selected HPC infrastructures. The created framework will form the foundation for parallel deployment of any *Fiji/ImageJ2* command on a remote HPC resource, greatly facilitating big data analysis.

4 Conclusion and Future Work

The HEAppE Middleware is a powerful tool for the users who wish to utilize the *HPC-as-a-Service* concept. It provides a way how to easily access an HPC infrastructure without upfront investment in the hardware itself, necessary services associated with the maintenance of a data processing and data storage infrastructure, or training of specialized personnel.

The first version of the middleware has been extensively used within the IT4-Innovations center in a number of internal or public projects for the scientific community or private sector. Whenever there is a need for an easy-to-use access or integration of HPC capabilities, the HEAppE Middleware is a go-to alternative for

[9] http://fiji.sc

the users who wish to use the supercomputer infrastructure yet unwilling to spend too much time learning how to both access it and work with it.

Since 2018, HEAppE Middleware has been released as an *open source under the GNU General Public License v3.0*. The source codes are available for everyone via the project's git repository at *http://heappe.eu*.

In terms of future work this project is under constant development. As the current implementation of this application framework is based on .NET Framework and uses Microsoft-SQL database it can only be deployed on the windows operated machines. We are currently preparing the .NET Core multi-platform version of the middleware with a simplified middleware deployment and versioning functionality by utilizing the Docker packages.

The second main task is to provide the users with a simple and easy-to-use graphical interface in the form of a workflow manager with the ability to create custom analytical pipelines (a sequence of interconnected computational tasks) to run on the cluster as a single computational job while retaining the ability to monitor the running tasks in near real time.

5 Aknowledgments

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project 'IT4Innovations excellence in science - LQ1602', by the European Regional Development Fund in the IT4Innovations national supercomputing center - path to exascale project, project number CZ.02.1.01/0.0/0.0/16_013/0001791 within the Operational Programme Research, Development and Education, partially supported by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project 'IT4Innovations National Supercomputing Center LM2015070' and by the SGC grant No. SP2019/108 'Extension of HPC platforms for executing scientific pipelines', VSB - Technical University of Ostrava, Czech Republic. The project 'Urban Thematic Exploitation Platform' was funded by European Space Agency (ESA) of ESA Contract No. 4000113707/15/I-NB.

References

1. Altair Engineering, Inc. PBS Professional. `https://www.pbsworks.com/PBSProduct.aspx?n=Altair-PBS-Professional&c=Overview-and-Capabilities`. Cited 28 February 2019.
2. Baun Ch., Kunze M., Nimis J., Tai S. *Cloud Computing Web-Based Dynamic IT Services*. Springer-Verlag Berlin Heidelberg, 2011. ISBN: 978-3-642-20917-8.
3. IT4Innovations National Supercomputing Center. High-End Application Execution Middleware. `http://www.heappe.eu`. Cited 28 February 2019.

4. eResearch South Australia. eResearch SA EMU-Cluster. `https://www.ersa.edu.au/service/cloud/clusters-in-the-cloud`. Cited 9 February 2019.
5. Ertaul L., Kaur M., Gudise V. A. K. R. Implementation and performance analysis of PBKDF2, bcrypt, scrypt algorithms. *Proceedings of the International Conference on Wireless Networks (ICWN); Athens*, pages 66–72, 2016.
6. Fusaro V. A., Patil P., Gafni E., Wall D. P., Tonellato P. J. Biomedical Cloud Computing With Amazon Web Services. *PLOS Computational Biology*, 2011. doi: 10.1371/journal.pcbi.1002147.
7. Google LLC. Google Cloud HPC Computing. `https://cloud.google.com/solutions/hpc`. Cited 28 February 2019.
8. Halligan B. D., Geiger J. F., Vallejos A. K., Greene A. S., Twigger S. N. Low cost, scalable proteomics data analysis using Amazon's cloud computing services and open source search algorithms. *Journal of Proteome Research*, pages 3148–3153, 2009. doi: 10.1021/pr800970z.
9. Armbrust M., Fox A., Griffith R., Joseph D. A., Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stoica I. and Zaharia M. A View of Cloud Computing. *Communications of the ACM*, 53(4), 2010.
10. Esch T., Asamer H., Bachofer F., Balhar J., Böttcher M., Boissier E., Hirner A. , Mathot E., Marconcini M., Metz-Marconcini A., Permana H., Soukup T., Svaton V., Üreyen S., Zeidler J. New Prospects in Analysing Big Data from Space - The Urban Thematic Exploitation Platform. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 8193–8196, 2018.
11. Krumnikl M., Moravec P., Kozusznik J., Klimova J., Bainar P., Svaton V., Tomancak P. SPIM Workflow Manager for HPC. *Bioinformatics*, 02 2019. doi: 10.1093/bioinformatics/btz140.
12. Yoo A.B., Jette M.A., Grondona M. SLURM: Simple Linux Utility for Resource Management. *Job Scheduling Strategies for Parallel Processing*, pages 44–60, 2003. doi: 10.1007/10968987_3.
13. Schadt E. E., Linderman M. D., Sorenson J., Lee L., Nolan G. P. Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics*, pages 647–657, 2010. doi: 10.1038/nrg2857.
14. Richardson L., Ruby S. *RESTful web services*. O'Reilly Media; First Edition edition, 2007. ISBN: 978-0596529260.
15. Svaton V., Podhoranyi M., Vavrik R., Veteska P., Szturcova D., Vojtek D., Martinovic J., Vondrak V. Floreon+: A Web-Based Platform for Flood Prediction, Hydrologic Modelling and Dynamic Data Analysis. *GIS OSTRAVA 2017*, pages 409–422, 2018. doi: 10.1007/978-3-319-61297-3_30.
16. Vecchiola Ch., Pandey S., Buyya R. High-Performance Cloud Computing: A View of Scientific Applications. *10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pages 4–16, 2009. doi: 10.1109/I-SPAN.2009.150.
17. Youseff L., Butrico M., Silva D. D. Toward a Unified Ontology of Cloud Computing. *Grid Computing Environments Workshop*, pages 1–10, Nov 2008.