Efficient Mass Spectra Prediction through Container Orchestration with a Scientific Workflow

Maximilian Hanussek[1,2,3], Felix Bartusch[1,2,3], Jens Krüger[1*]
[1] High-Performance and Cloud Computing Group
Zentrum für Datenverarbeitung, University of Tübingen
Tübingen, Germany
[*]jens.krueger@uni-tuebingen.de

Oliver Kohlbacher[2,3,4,5]
[2] Center for Bioinformatics, [3] Dept. of Computer Science,
[4] Quantitative Biology Center, University of Tübingen
Tübingen, Germany
[5] Biomolecular Interactions, Max Planck Institute for
Developmental Biology, Tübingen, Germany

***Abstract*—The mass spectroscopic fragmentation of small molecules such as metabolites can be simulated with QCEIMS. In this paper we present our work dealing with the containerization of the complex and interdependent software stack. The simulation protocol has been mapped to a UNICORE workflow enabling convenient access to powerful computing resources. To offer a maximum of convenience to the users a simple portal was deployed hiding the complexity of technical details.**

Keywords—containerization; workflows; reproducibilty; science gateway; mass spectrometry; quantum mechanics

I. INTRODUCTION

In the natural sciences, there are many software applications which are commonly used, but which are not easy to install or to apply. Installation problems can originate from special computing environments being required or the number of interdependent additional software packages that need to be installed. Furthermore, many programs require the usage of command line interaction by the user. This knowledge is not always present and should not be a prerequisite. But over time, technologies have emerged that allow an easy installation and operation of complex tools [1].

One particular technology that gained popularity in recent years is container virtualization. Representatives of container virtualization methods based on the Linux system are Linux-VServer [2], Docker [3], OpenVZ [4], Linux Container (LXC) [5] and Singularity [6]. Among all these representatives, Docker is the most prominent. Docker and its container-technology are a lightweight alternative to full virtual machines. Since the virtualization is running on the host OS, it is possible to run multiple applications in parallel without establishing a new kernel for each application, which makes the container-based technique more lightweight than a hypervisor-based approach [7], [8].

Most installation and computing environment problems can be solved by providing a container for any desired tool. Almost every required program can be wrapped into such a container, which saves time for installation and does not require special permissions. The container is a self-contained environment, no matter on which system it runs. This fact leads to a good reproducibility of already achieved results and is especially important in the natural sciences. Using Docker for distributing software stacks could be one approach to solve installation and computing environment problems. But the user-friendliness concerning the operation of a complex tool can not be increased through it.

Another technology that can be used to increase the user-friendliness are workflows representing specific scientific protocols. In the meantime, many workflow platforms are available such as KNIME [9], TAVERNA [10], Pipeline Pilot [11], [12], Galaxy [13], [14], and UNICORE [15]. The Uniform Interface to Computing Resources (UNICORE) is a mature so-called middleware solution to create workflows and in addition, get access to distributed computing resources. UNICORE is used in many research fields and settings from small projects up to large transnational projects like MoSGrid [16], [17], the European PRACE infrastructure [18], the US XSEDE Initiative [19] or the Human Brain Project [20]. An advantage of UNICORE is that it provides access to high-performance computing (HPC) clusters and file systems and offers the possibility to generate workflows suitable for HPC environments.

The interaction between Docker and UNICORE makes it possible to simplify both, the installation process and the use of complex tools. In the following chapters, Docker and UNICORE are explained in more detail.

II. Docker

There are two major concepts in the field of software virtualization, container-based virtualization and hypervisor-based virtualization. Both multilayered approaches are illustrated in Figure 1. Examples for hypervisor-based virtualization software are VMware [21] or Xen [22]. A major aspect is that a hypervisor-based virtualization establishes a full virtual machine on top of the host operating system. Such a virtual machine has its own operating system (Guest OS) and own kernel. This virtualization technology provides a virtualization on the hardware level.

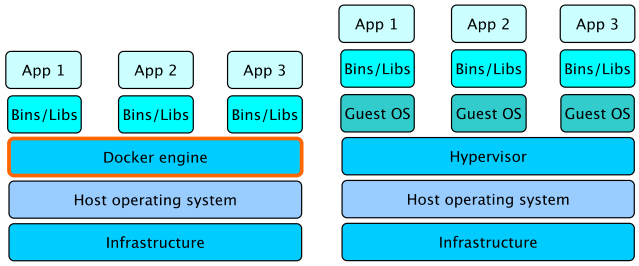

Fig. 1. Container-based approach including applications and the necessary binaries and libraries building up on the Docker engine (left). Hypervisor based approach with an additional guest OS on top of the hypervisor layer (right).

In contrast to the hypervisor-based virtualization, Docker establishes a virtualization based on the host OS. The virtual environments are directly run on the host kernel, which are usually named containers [8]. It is possible to create own Docker images, which serve as template for the Docker containers. The images are created via the so-called Dockerfile, which is a plain text file that specifies how the containers are created and run. Docker images are built upon a base image which can be any operating system that fits to the host OS, on a Linux system for example Ubuntu or CentOS. Images consist of a series of data layers on top of the base image. Worth mentioning is that a variety of containers can be started from only one image, each container does not need its own image. The already available images can be used as a new base image and can be extended further [7]. This is simply done by adding a new data layer which is more efficient than building the whole image from scratch. To work with these multiple layers, Docker uses the Union File System to merge the different layers into a single and consistent file system which is one of the underlying techniques. A Docker container provides a virtual environment for its contained applications by leveraging the Linux kernel features control groups (cgroups) for accounting processes of the container and namespaces for providing isolated instances of host resources [8].

III. UNICORE

The UNICORE software package is developed at the research center in Jülich and by further partners [15], [23]. It provides different components for handling HPC environments as well as a workflow editor, a web interface (UNICORE-portal) and a more advanced graphical user interface, the UNICORE rich client (URC). One advantage that UNICORE offers is that it is designed to abstract computing resource specific details and through that simplifies the user experience. Furthermore, it is extensible due to the use of standardized APIs, which for example makes it possible to run KNIME nodes on a HPC via UNICORE [24]. No special operating system is required as UNICORE is completely written in the platform independent programming languages Java and Python. Another not negligible aspect is the need to use a certain safety standard to prevent the loss of sensible data. Due to different security and authorization methods offered by UNICORE the connection between client and server is considered to be save [15].

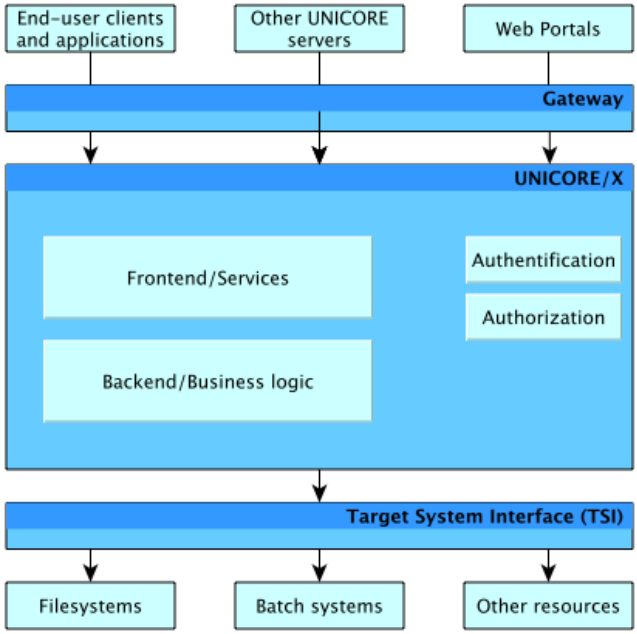

Fig. 2. Overview of the different UNICORE components and their interaction with each other [15].

The UNICORE architecture consists of five layers (user-layer, gateway-layer, UNICORE/X-layer, TSI-layer, resource-layer). The user-layer provides end-user clients and applications but also other UNICORE servers and web portals. The gateway-layer serves mostly as a firewall transversal point and forwards information such as IP addresses or SSL certificates via the connecting client to the following servers. The UNICORE/X is the central component of UNICORE. It receives the client requests, which has been submitted via the gateway, authenticates the request, checks the authorization and in the end, invokes the appropriate service. The Target System Interface (TSI) is connected with the local operating system, file system and usually a batch system for the resource management. The tasks of the TSI are for example to submit the sent jobs from the client, check the status of the jobs, or perform the I/O operations. A schematic illustration of the different layers is shown in Figure 2.

IV. QCEIMS

The fragmentation of small molecules as it occurs within a mass spectrometry experiment can be simulated with quantum chemical simulations. The method called QCEIMS (Quantum Chemistry Electron Ionization Mass Spectrometry) developed by Grimme et al. [25]–[27] creates initially a trajectory for the molecule of interest and extracts a set of starting conformers for further calculations. Each ionized conformer gets fragmented at high temperature resembling the conditions within a mass spectrometer. The resulting fragmentation distribution over several hundred individual fragmentation runs resembles a mass spectrum and can be compared to experimental data. Such simulated spectra may be used in metabolomics to facilitate the identification of compounds.

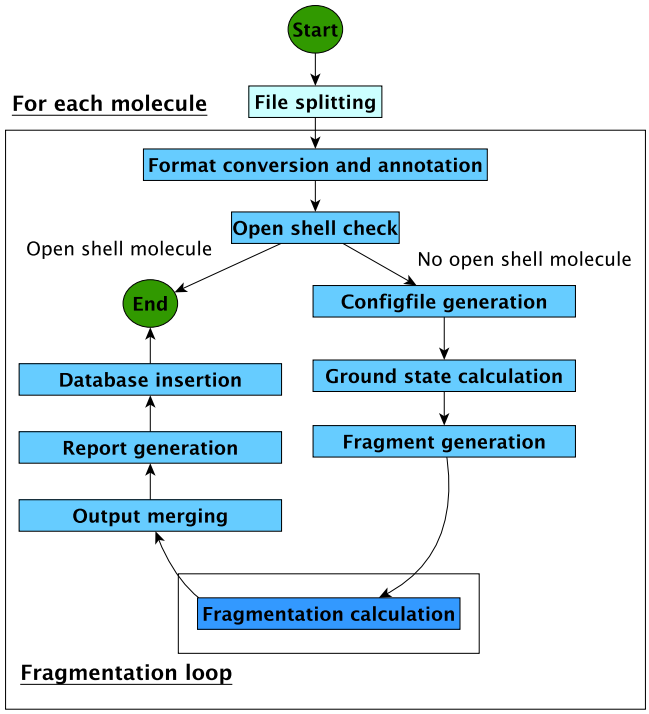

Fig. 3. The workflow for the prediction of mass spectra based on quantum chemical fragmentation calculations is shown. It takes advantage of multiple control structures to efficiently process even larger numbers of molecules. The different nesting levels are highlighted by the distinct colors.

V. WORKFLOW AND SCIENCE GATEWAY

A. Overview

The first application using both Docker and UNICORE together is the UNICORE QCEIMS workflow for mass spectra prediction. The implemented UNICORE workflow embeds the QCEIMS tool [25]. QCEIMS is very well suited for execution on HPC clusters, as it makes strong use of quantum chemistry programs that require high computing power, and the fact that the QCEIMS calculations are easy to parallelize. Furthermore, installation and handling are quite complex. An overview of the workflow is shown in Figure 3. UNICORE is used to encapsulate the whole mass spectra prediction process into a UNICORE workflow. Further Docker is used to encapsulate and execute all QCEIMS calculations in Docker containers. The created Docker image contains all necessary software to execute the QCEIMS calculations. These tools are listed in Table 1. Only the UNICORE specific software components are not included and also MOPAC due to the required license.

Tab. 1. Software included in QCEIMS Docker image.

Program	**Version**
Python	2.7.10
R (with Sweave)	3.2.3
QCEIMS	2.26I
MNDO99	7.0
DFTB+	1.2.2
ORCA	3.0.3
InChI version 1	1.04
PubChemPy	1.0.3
Tex Live	2016

B. Workflow description

The implemented workflow accepts structure data files (.sdf) as input, which can contain the structure of one molecule or more. Due to the UNICORE characteristic that every job is executed in a single directory, with no subdirectories, it is necessary to encapsulate each molecular structure in its own job directory. This is achieved by using the for-each loop concept of the UNICORE workflow editor and represented as the outer for loop in Figure 3. After the necessary format conversion from the .sdf file format into the .tmol format an open shell check is performed with MOPAC. If the molecule is not an open shell molecule the configuration file containing the QCEIMS parameters is automatically generated. After these preparation steps, the quantum chemical calculations are started for the first time. All necessary programs for this step are already installed in a Docker container. After the calculations have finished the second encapsulation with a second for-each loop for the fragmentation calculations is performed. This is necessary due to the structure of the QCEIMS tool. QCEIMS assumes that the files, required for the subsequent calculations, are available in separate directories and can be processed within these directories. But this characteristic does not fit the workflow implementation of UNICORE. Further, QCEIMS uses the same file names in the subdirectories, which is not a problem if the files remain separate but this is not the case for UNICORE. If the different files would not be encapsulated into single jobs with the for-each loop construct, it would not be possible to distinguish them later. After the fragmentation calculation, it is necessary to merge the single spectra of each fragmentation into the final

simulated spectrum. In the end the generated results are automatically integrated in a report and stored in a database.

C. *Workflow execution through webinterface*

The implemented QCEIMS workflow can be used by importing it into the UNICORE rich client (URC) which is recommended for users who are already familiar with the UNICORE environment. With the URC it is also possible to export the implemented workflow as an .xml file and use it in the UNICORE portal if the user wants to modify the workflow. Another option to execute the workflow is to use the UNICORE portal. The UNICORE portal is a further component of the UNICORE tool set and works seamlessly with the UNICORE server and the UNICORE workflow engine. The portal offers a straightforward way to make high level computing resources and complicated workflows available to a wide range of users, who are not familiar with these techniques, and hide the complex underlying structures.

To use the portal to its full extent it is necessary to have a valid User-Grid certificate imported in the browser that has to be registered once.

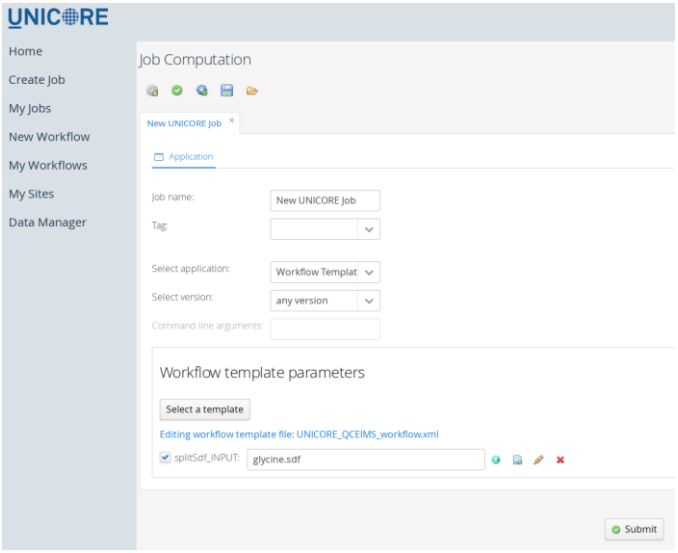

Fig. 4. UNICORE portal job submission screen, allowing the selection of a workflow, its configuration and specification of input data.

In order to run the UNICORE QCEIMS workflow it is necessary to create a new job in the "Create job" screen (Figure 4). The "Select application field" parameter has to be set to "Workflow Template". Now it is possible to "Select a template" from the file system, in our case the QCEIMS workflow XML file. After the template has been uploaded to a HPC instance, the input data can be set and uploaded too.

D. *Evaluation of QCEIMS*

Overall the results of the QCEIMS tool, the simulated mass spectra, show a sufficient similarity with the experimentally generated spectra. We used the absolute value distance as quality measure to compare the simulated spectra with their experimental spectra [28]. Every molecule of the small test set we have used, showed a score close to 0.6 or higher. The similarity represented by this score is high enough to find the simulated compounds under the top 3 hits if matched against a mass spectral database. In most of the cases it will be the first hit. When comparing Docker containers with bare-metal execution no differences in run time were observed. Due to the use of the for-each loop construct, provided by the UNICORE workflow, it is possible to sequentially parallelize the computation of each fragmentation step. Furthermore, each quantum chemical simulation can be distributed over more than a single CPU but this would not be efficient.

VI. Future work

Due to the special licenses required for the quantum chemistry tools it is not possible to distribute the created Docker image on Docker Hub, which motivated us to create a portal solution instead.

An important future task is the improvement of the login procedure into the UNICORE portal. It is quite a considerable effort to appear personally at an authentication center and apply for a personal grid certificate. With the software Unity, which is already integrated in UNICORE, it would be possible to simplify the registration process. Unity would enable the authentication via user credentials provided by the user's home organization and corresponding Shibboleth entitlements.

Another area where enhancements are anticipated is the provision of input options for the user to modify the default parameters of the QCEIMS tool.

The automatically generated report at the end of the calculations includes the basic results and therefore could be extended with further graphics and more detailed statistics.

Currently the containerized workflow is a stable prototype and is already being used by first experimental groups.

VII. Conclusion

The use of Docker in combination with UNICORE made it possible to simplify a complex tool such as QCEIMS with regards to its installation process as well as to its handling. The presented QCEIMS Docker image is the first of its kind to cover the topic of mass spectra prediction and also the first publication using Docker with UNICORE workflows. Furthermore, providing Docker images as a complete execution environment results in a good reproducibility of results for other users. In the Docker image is clearly stated which parameters of various additional tools were used, which cannot change as long as no update of the image is carried out.

The use of Docker also revealed weaknesses concerning security and a missing garbage collection, in particular for the application of Docker container in a massive parallel way on HPC clusters. In its present version, Docker can only be used with drawbacks in parallel environments on HPCs. This is valid until the problem of the accumulation of data and metadata files is solved. A valid approach to clean up the

exited and sometimes dead containers would be the implementation of a Cron job that searches for containers in the described states and deletes them. A housekeeping strategy to solve the metadata accumulation could be a central Docker image repository. A further Cron job, that deletes the directory of the metadata files and loads all images from the repository back into the system would be a possible workaround.

Despite these developed workarounds, it would be desirable if Docker provides a garbage collection tool once for the exited or dead containers and even more important, for the metadata accumulation due to the container-snapshots. If these problems can be solved, the popularity of Docker would rise even further.

Another outcome is that UNICORE is a very useful piece of software. The installation is not that difficult if considered what the components do. By installing only three packages you easily get access to a HPC, a graphical user interface with a versatile workflow environment and a web interface. The wrapping of the QCEIMS tool into a UNICORE workflow has been successfully achieved which shows that UNICORE is generally applicable to such kind of problems and could be used for future projects. To the present day UNICORE does not support Docker directly but the developers are aware of this virtualization technique.

Consequently, the combination of Docker and UNICORE represents an excellent setup to carry out fragmentation calculations using QCEIMS. Large libraries of small molecules can be processed conveniently and their simulated spectra can be used to help with the identification of metabolites.

ACKNOWLEDGMENT

The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 37/935-1 FUGG. Part of the work presented here was also supported through BMBF funded project de.NBI (031 A 534A) and MWK Baden-Württemberg funded project CiTAR ("Zitierbare wissenschaftliche Methoden"). We thank Bernd Schuller for invaluable support with UNICORE, and especially Christoph Bauer and Stefan Grimme for the help with QCEIMS.

REFERENCES

[1] J. Krüger and O. Kohlbacher, "Containerization and Wrapping of a Mass Spectra Prediction Workflow," *PeerJ Preprints*, pp. 8–10, 2016.

[2] S. Soltesz *et al.*, "Container-based operating system virtualization," in *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007 - EuroSys '07*, 2007, vol. 41, no. 3, p. 275.

[3] "Docker." [Online]. Available: https://www.docker.com/. [Accessed: 30-Mar-2017].

[4] "OpenVZ." [Online]. Available: http://openvz.org/. [Accessed: 30-Mar-2017].

[5] "LXC." [Online]. Available: https://linuxcontainers.org/. [Accessed: 30-Mar-2017].

[6] G. M. Kurtzer, "Singularity 2.1.2 - Linux application and environment containers for science," 01-Jan-2016. [Online]. Available: https://zenodo.org/record/60736#.WOErqqJBrZs. [Accessed: 02-Apr-2017].

[7] C. Boettiger and Carl, "An introduction to Docker for reproducible research," *ACM SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 71–79, Jan. 2015.

[8] T. Bui, "Analysis of Docker Security," *arXiv.org*, p. http://arxiv.org/abs/1501.02967, Jan. 2015.

[9] M. R. Berthold *et al.*, "KNIME - The Konstanz Information Miner," *SIGKDD Explor.*, vol. 11, no. 1, pp. 26–31, Nov. 2009.

[10] T. Oinn *et al.*, "Taverna/myGrid: Aligning a Workflow System with the Life Sciences Community," in *Workflows for e-Science*, I. J. Taylor, E. Deelman, D. B. Gannon, and M. Shields, Eds. Springer London, 2007, pp. 300–319.

[11] "Pipeline Pilot." [Online]. Available: http://accelrys.com/products/collaborative-science/biovia-pipeline-pilot/. [Accessed: 30-Mar-2017].

[12] W. A. Warr, "Scientific workflow systems: Pipeline Pilot and KNIME.," *J. Comput. Aided. Mol. Des.*, vol. 26, no. 7, pp. 801–4, Jul. 2012.

[13] E. Afgan *et al.*, "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update.," *Nucleic Acids Res.*, p. gkw343, May 2016.

[14] A. K. Hildebrandt *et al.*, "ballaxy: web services for structural bioinformatics.," *Bioinformatics*, vol. 31, no. 1, pp. 121–122, Sep. 2014.

[15] K. Benedyczak, B. Schuller, M. Petrova-El Sayed, J. Rybicki, and R. Grunzke, "UNICORE 7 — Middleware services for distributed and federated computing," in *2016 International Conference on High Performance Computing & Simulation (HPCS)*, 2016, pp. 613–620.

[16] J. Krüger *et al.*, "The MoSGrid Science Gateway – A Complete Solution for Molecular Simulations," *J. Chem. Theory Comput.*, vol. 10, no. 6, pp. 2232–2245, Jun. 2014.

[17] L. Zimmermann, R. Grunzke, and J. Krüger, "Maintaining a Science Gateway – Lessons Learned from MoSGrid," in *Hawaii International Conference on System Sciences (HICSS)*, 2017, p. http://hdl.handle.net/10125/41918.

[18] "PRACE." [Online]. Available: http://www.prace-ri.eu/. [Accessed: 30-Mar-2017].

[19] "UNICORE in the XSEDE infrastructure." [Online]. Available: https://portal.xsede.org/software/unicore/. [Accessed: 30-Mar-2017].

[20] "Human Brain Project." [Online]. Available: https://www.humanbrainproject.eu/. [Accessed: 30-Mar-2017].

[21] "VMware." [Online]. Available: http://www.vmware.com. [Accessed: 30-Mar-2017].

[22] "Xen." [Online]. Available: https://www.xenproject.org. [Accessed: 30-Mar-2017].

[23] A. Streit *et al.*, "UNICORE 6 - Recent and Future Advancements," *JUEL-4319*, 2010.

[24] R. Grunzke, F. Jug, B. Schuller, R. Jäkel, G. Myers, and W. E. Nagel, "Seamless HPC Integration of Data-intensive KNIME Workflows via UNICORE," in *4th International Workshop on Parallelism in Bioinformatics (PBio 2016)*, 2016, p. (accepted).

[25] S. Grimme, "Towards First Principles Calculation of Electron Impact Mass Spectra of Molecules," *Angew. Chemie Int. Ed.*, vol. 52, no. 24, pp. 6306–6312, Jun. 2013.

[26] C. A. Bauer and S. Grimme, "How to Compute Electron Ionization Mass Spectra from First Principles," *J. Phys. Chem. A*, vol. 120, no. 21, pp. 3755–3766, Jun. 2016.

[27] V. Ásgeirsson *et al.*, "Unimolecular decomposition pathways of negatively charged nitriles by ab initio molecular dynamics," *Phys. Chem. Chem. Phys.*, vol. 18, no. 45, pp. 31017–31026, 2016.

[28] S. Stein and D. Scott, "Optimization and testing of mass spectral library search algorithms for compound identification", *Journal of the American Society for Mass Spectrometry*, 5(9):859–866, 1994.