

International Conference on Computational Science, ICCS 2013

Support for Multiscale Simulations with Molecular Dynamics

Katarzyna Rycerz^{a,b}, Eryk Ciepiela^b, Grzegorz Dyk^b, Derek Groen^c, Tomasz Gubala^b,
Daniel Harezlak^b, Maciej Pawlik^b, James Suter^c, Stefan Zasada^c, Peter Coveney^c,
Marian Bubak^{a,b}

^aAGH University of Science and Technology, Department of Computer Science, Al. Mickiewicza 30 30-059 Krakow, Poland

^bAcademic Computer Centre CYFRONET, University of Science and Technology, ul. Nawajki 11, 30-950 Krakow, Poland

^cCentre for Computational Science, University College London, London, United Kingdom

Abstract

We present a reusable solution that supports users in combining single-scale models to create a multiscale application. Our approach applies several multiscale programming tools to allow users to compose multiscale applications using a graphical interface, and provides an easy way to execute these multiscale applications on international production infrastructures. Our solution extends the general purpose scripting approach of the GridSpace platform with simple mechanisms for accessing production resources, provided by the Application Hosting Environment (AHE). We apply our support solution to construct and execute a multiscale simulation of clay-polymer nanocomposite materials, and showcase its benefit in reducing the effort required to do a number of time-intensive user tasks.

Keywords: distributed multiscale simulations; tools; e-infrastructures

1. Introduction

Simulating phenomena using a multiscale approach is of great importance in many fields of science, and has resulted in a large number of new insights [1, 2, 3]. In this paper we focus on simulations that combine single scale models of physical phenomena, each of which is resolved using a well-known simulation code. Such simulations require a range of tasks, including selecting appropriate single-scale models (and codes which support them), converting output data from one model to the input data of another, executing the single-scale models on appropriate production resources, and ensuring that the necessary data is transferred between the user interface and the remote resources at the various stages of the simulation.

Computational chemists commonly rely on widely used simulation codes such as the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)¹, a classical parallel molecular dynamics code, and the Car-Parrinello Molecular Dynamics (CPMD)², a parallelized ab-initio molecular dynamics code used for quantum-mechanical simulations [4]. These solvers are commonly run in parallel using a cluster resource or supercomputer (in the case of LAMMPS). The work required to convert the output of one model to the input of another is done

*Corresponding author. Tel.: +48 12-328-33-58 ; fax: +48 12 617-51-72.

E-mail address: kzajac@agh.edu.pl.

¹<http://lammps.sandia.gov/>

²<http://www.cpmc.org/>

by coupling scripts, often custom-written by domain scientists using a language such as Python or Perl. Other tasks such as remotely executing a single-scale model or transferring data between locations are frequently done manually, using basic tools such as SSH or GridFTP.

In this paper we propose a solution which supports the creation of (distributed) multiscale applications and their execution on production infrastructures such as PRACE³ and EGI⁴, and which allows convenient reuse of previously defined simulation workflows. The solution extends the general purpose scripting approach of the GridSpace platform [5] with new features for e-Infrastructure access, which are provided by the Application Hosting Environment (AHE) [6]. It also provides support for reusability and visual composition of created applications, relying on the Mapper Memory (MaMe) and Multiscale Application Designer (MAD) tools. The work presented in this paper is a part of an advanced environment for building and running multiscale applications, which has been developed within the MAPPER project⁵.

This paper is organized as follows: we present related work in Section 2, and explain the process of creating multiscale applications and the needs for support in Section 3. We present our combined GridSpace-AHE approach in Section 4 and describe our software infrastructure to allow application reuse in Section 5. In Section 6 we present a case study of clay-polymer interaction simulation and explain how it can benefit from our proposed solution. We conclude the paper in Section 7.

2. Related Work

There are variety of solvers supporting computational science that require inputs as scripts in a Domain Specific Language (DSL), such as the aforementioned LAMMPS [7] and CPMD [8] codes. Simulations built in LAMMPS represent molecular systems, resolving the atoms either individually as an all-atom (AA) simulation or by representing groups of atoms as single particles in a coarse-grained (CG) simulation. CPMD uses a parallelized plane wave and pseudopotential implementation of Density Functional Theory, particularly designed for ab-initio molecular dynamics. Another example solver that supports input as DSL scripts is Gaussian [9], which is a widely-used code for electronic structure modeling.

Several tools have been developed to assist in composing applications from existing software modules, the approach that can also be applied to build simulations that rely on multiple single scale models. These include workflow engines such as Kepler [10], Taverna [11] or WS-VLAM [12], each of which offers a visual interface that aids in constructing workflow-based application. Another example is a component-based approach [13] and its application to HLA-based multiscale simulations [14]. However, the single-scale models we use here rely on DSL-based inputs, which matches intuitively to a more scripting-based approach. We have chosen to use the GridSpace platform [5], which can combine scripts expressed in a number of popular languages to facilitate the exploratory development of so-called computational "in-silico" experiments.

Multiscale molecular dynamics simulations often require large computational resources. To fulfill this requirement, users need to have easy access to production e-Infrastructures such as EGI and PRACE. The application Hosting Environment (AHE) [6, 15] is designed to allow scientists to quickly and easily run unmodified legacy applications on grid resources, manage the transfer of files to and from the grid resource and monitor the status of the application. AHE is able to interface to HPC resources, via the back-end middleware interfaces they present, including UNICORE [16], QCG-Computing [17] and Globus [18], meaning that a user can use a single AHE installation to access resources from a variety of different resource providers, with AHE acting as an integration layer.

3. Support for multiscale application creation and execution process

We present the process of creating applications, and the tools required to support this, in Fig. 1. Here, users create appropriate input scripts using their DSL for each single scale model. User can run a partially created

³<http://www.prace-ri.eu/>

⁴<http://www.egi.eu/>

⁵<http://www.mapper-project.eu>

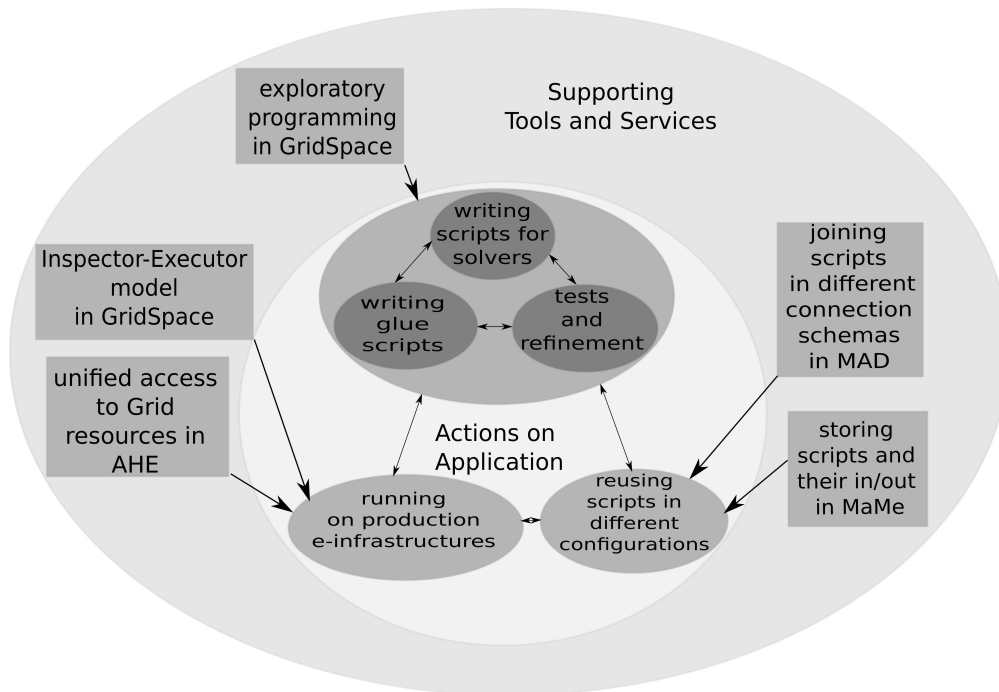


Fig. 1. A diagram describing support activities for the process of creating, testing and running DSL-based multiscale applications. Actions performed by the user are showed in the inner circle, tools supporting these actions in the outer circle.

application, test and create coupling codes using well-known scripting languages such as Python or Perl. Once the application is ready, users can immediately run it on production e-Infrastructures resources they have access to, provided the appropriate single-scale solver binary is available on-site. In addition, they can save their created scripts and their connection scheme for further reuse.

In the context of these activities, GridSpace aids the user in creation and execution of scripts that are combined into a full multiscale application. The process of creation also includes testing and improving the scripts. Once an application is prepared, the user can use AHE to execute the application on different infrastructures by selecting an appropriate resource in the interface and providing a few resource-specific details. To make switching resources easier for the user, we designed and implemented the interpreter-executor model, where we separate the software package from its execution environment (see Section 4). The user can save created scripts together with the required input and output names in the MAPPER Memory registry (MaMe) which can be reused later on to compose different applications in the Multiscale Application Designer (MAD) graphical web interface (see Section 5).

4. GridSpace and AHE working in Inspector-Executor Model

GridSpace [5] introduces a model for scientific computing in a distributed production environment and provides a common integration platform to access and exploit such environments [19]. The platform comes with the Experiment Workbench web application⁶ which allows users to write, run, share and publish [20] computational "in-silico" experiments. This application relies on the Experiment Execution Engine, which dispatches and orchestrates executions of these experiments.

The GridSpace model introduces code snippets that are artifacts such as programs, algorithms, problem definitions or specification of computations. Code snippets can be written in any general-purpose or domain-specific,

⁶<https://gs2.mapper-project.eu/ew>

```

<gridSpace configurationVersion="1.0">
  <!-- interpreters -->
  <interpreter id="Perl-5.8.8" name="Perl 5.8.8" arguments=""
manualUrl="http://www.perl.org/learn.html" executionType="BATCH">
  </interpreter>
  <interpreter id="LAMMPS-30Sep11" name="LAMMPS 30Sep11"
arguments="" manualUrl="http://lammeps.sandia.gov/" executionType="BATCH">
  </interpreter>
  ...
  <!-- AHE executor -->
  <executorFactory id="ahe-executor-0.1.2" name="AHE Executor"
loginType="PROXY" accountRequestUrl="..."
executorClass="ahe-executor-0.1.2:cyfronet.gs2.executors.ahe.AHEExecutor">
  <description>Application Hosting Environment executor</description>
  <property name="..." value="..." />
    <executable interpreter="Perl-5.8.8"
cmd="/usr/bin/perl@[VAR:machine]" />
  <executable interpreter="LAMMPS-30Sep11"
cmd="/opt/mpich2/gnu/bin/mpirun -np [VAR:nodes]
/bin/lmp_openmpi_Aug11 -in {SNIPPET}@{SNIPPET}@[VAR:machine]" />
  </executorFactory>
</gridSpace>

```

Fig. 2. GridSpace configuration file (irrelevant parts omitted) that specifies the AHE Executor along with example LAMMPS and PERL interpreters installed on underlying infrastructure. First section contains general information about interpreters, second section describes how to actually invoke them using the AHE executor.

4th or 5th generation programming language, which we refer to as the interpreting language, or *interpreter*. Code snippets are executed by so-called *executors* which manage computations on remote resources and the required data staging operations.

Executors provide a level of abstraction when dealing with the management of computational facilities in several ways: e.g. through brokering software such as QCG-Broker, through a SSH connection to a user interface node of a computational site, or through interfaces to a custom user environment such as AHE. Once resource access has been established, the respective software packages, implementing the interpreters, get executed in the infrastructure and the codes written in given interpreter languages get evaluated. Multiple code snippets and data files can be combined and form complex applications.

The GridSpace model introduces flexibility in terms of adapting to the existing work environment of computational scientists. It allows users to easily incorporate a variety of interpreters that are in use in research, and it gives users a choice of executors enabling access to a wide range of computational resources, provided by private, public or commercial parties. Moreover, all entities such as code snippets, data files, computational experiment plan files (for simplicity called GridSpace experiments), interpreters, executors and infrastructure elements can be stored for reuse, promoting effective usage and collaborative sharing of resources, methods and results.

The architecture of GridSpace supports this flexible model by allowing customization of and extensions to the platform. Executors and interpreters, the latter if installed in the infrastructure, are easily configured in the platform. We present an example GridSpace configuration file, which specifies the environment consisting of the AHE executor along with sample LAMMPS and Perl interpreters installed in the underlying infrastructure, in Fig. 2.

Each executor implements the GridSpace Executors API, which is a Java API for setting up, managing and securing the connections with underlying infrastructure, in accordance with respective policies. The AHE executor, which was developed within the scope of this work, is shown in Fig. 3 and implements this API as follows:

- Sessions are established with user-provided credentials (either GSI proxy certificate or login–password pair)

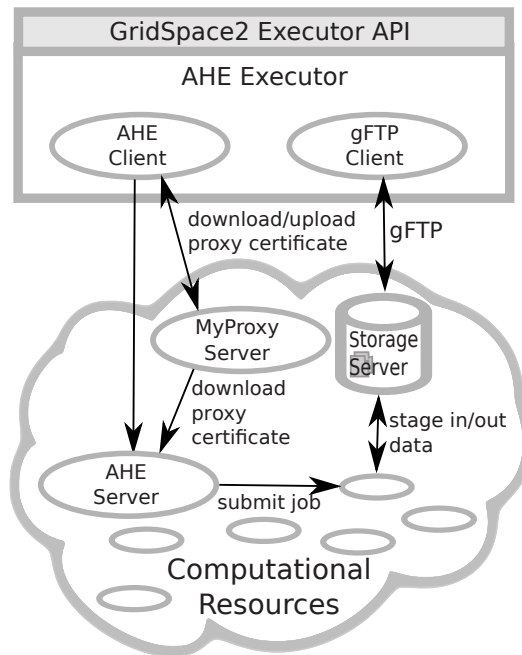


Fig. 3. AHE Executor implementing GridSpace Executor API using AHE and GridFTP clients in order to manage computations and data.

that are used to set up connection to indicated GridFTP storage server, and initialize AHE Client. When initializing, the AHE Client contacts a MyProxy server and, depending on user's choice, it either uploads proxy certificate and secures it with a newly generated login–password pair, or it downloads a previously uploaded proxy certificate after authentication through a user supplied login–password pair. At this stage the users get authenticated and authorized. Shutting down sessions is done by closing all open connections to the aforementioned external systems.

- Executions are managed with the AHE Client which contacts an indicated AHE Server and provides both a MyProxy login–password pair and a job description. The job description specifies the location where input and output data stored, including the address of a GridFTP server. The MyProxy login–password pair is used by the AHE Server to retrieve a proxy certificate and to submit jobs.
- Data is stored on the dedicated GridFTP server instance where it can be managed by the user and accessed from the computing infrastructure during execution.

5. Reusability Support

As shown in Fig. 1, part of the application lifecycle is its reusability. We propose tools that allow the user to store elements that comprise multiscale applications and compose them in different configurations.

MAPPER Memory (MaMe) is mainly responsible for providing a rich, semantics-aware persistence store which records information based on the semantic integration technology [21]. MaMe is a successor of Grid Resource Registry[22]. Application designers can use the web-based interface of the registry⁷ to record scripts they created, and can use these to compose complex applications with the Multiscale Application Designer (MAD). The registry also stores metadata about inputs and outputs needed by such scripts which are indicated as ports to be connected. If a script simulates phenomena that have a notion of a scale, the detailed information about this can be stored as well. The metadata model of MaMe is based on the Multiscale Modeling Language (MML) [23]. MaMe also provides a REST interface for other software tools such as MAD to store, publish and share common information.

⁷<http://mapper-project.eu/mame>

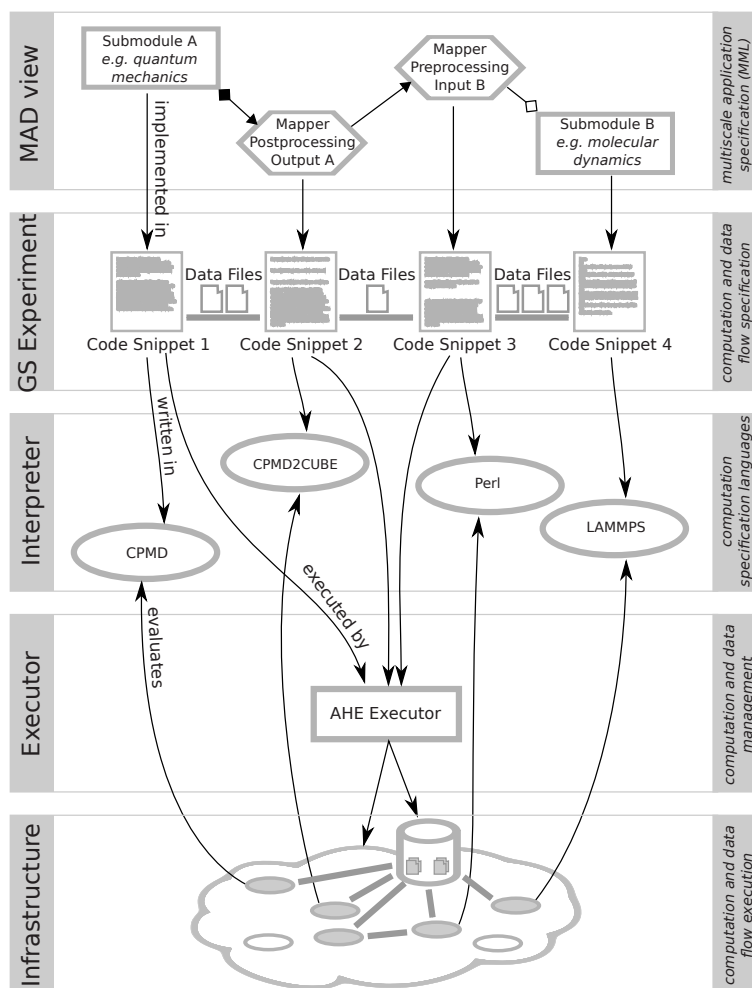


Fig. 4. Design of the clay-polymer interactions simulation using Multiscale Application Designer, GridSpace and AHE. A Visual connection scheme of the application is transformed into a GridSpace experiment definition. This experiment definition can then be executed on production resources using SSH and/or the AHE executor.

Multiscale Application Designer fetches data from MaMe and allows for a visual composition of applications through a web page⁸. The corresponding connection scheme is generated from its visual view. This scheme is then translated into a portable executable experiment form, which is used by GridSpace to run the application on resources chosen by a user and fetch the results. The application assembly in MAD supports saving the application at any state of composition together with parameters of individual models. The default values for these parameters are obtained from the Mapper Memory registry. It is also possible to share a composed application with other users through the application repository for collaborative work.

6. Case study

6.1. Multiscale Modelling of Clay-Polymer Nanocomposite Materials

We have applied our support approach to facilitate multiscale simulations of clay-polymer interactions [4]. Our simulations allow us to study and design of layered mineral composites in areas that have a substantial potential

⁸<https://gs2.mapper-project.eu/mad>

impact such as energy applications (oil industry additives), materials applications (nano composite materials) and biomedical applications (e.g. drug delivery).

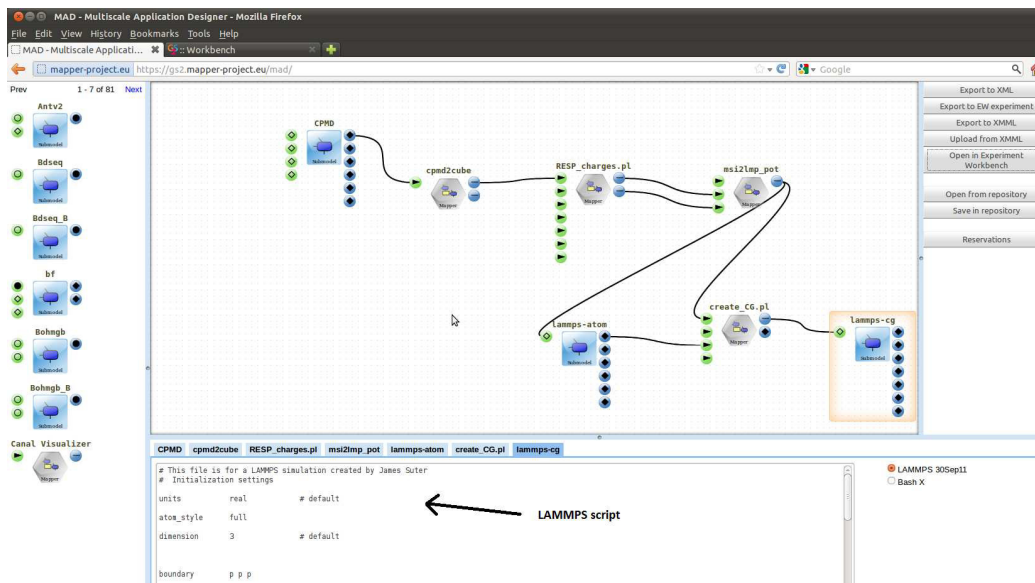


Fig. 5. Visual view of the simulation of clay-polymer interaction in Multiscale Application Designer; possible modules registered in MaMe are seen on the left; the actual script code can be seen and changed in the lower part of the tool;

Due to the hierarchical microstructure of these composites, which exist on many different length and timescales, these processes involve both microscopic and macroscopic phenomena. No one simulation code can capture all these phenomena; for example, conventional molecular dynamics (MD) may accurately simulate the interface between the nanoscale particles and the polymer, but is unable to fully resolve inter-particle behavior [24], even with the promise of exascale resources. We therefore need to use multiscale modelling, combining quantum-

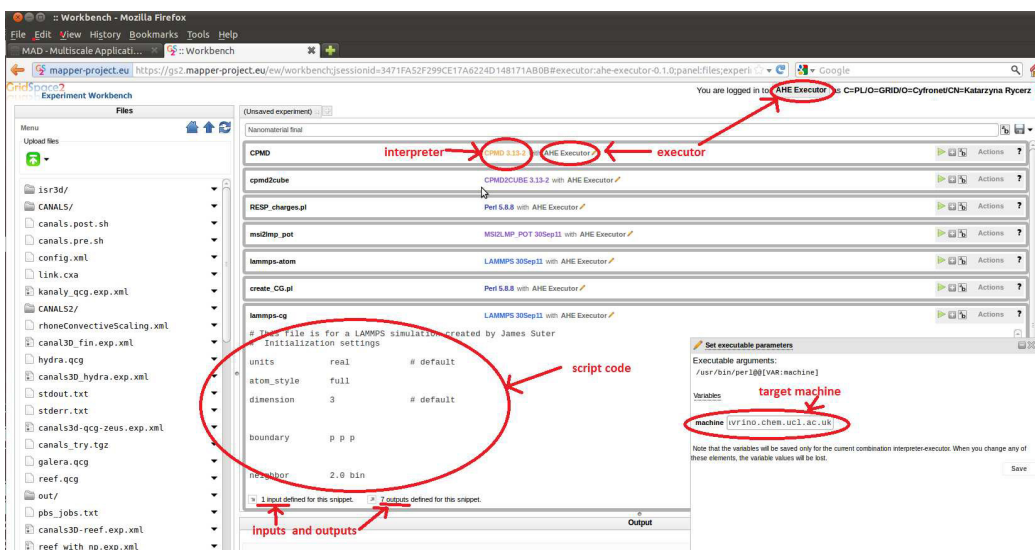


Fig. 6. Simulation of clay-polymer interactions in GridSpace. The experiment consists of seven snippets, each having their interpreter and executor. Files accessible through GridFTP on the storage server are visible on the left.

mechanical simulations with atomistic and coarse-grained molecular dynamics simulations [25, 26].

In this paper we apply our support software tools to facilitate this simulation, includes three submodels. The calculation of sheet edge potentials is done using a quantum mechanical solver (CPMD [8]), which operates on a spacial scale of less than a nanometer. The hierarchical modelling of clay sheets in a polymer matrix over short time periods of high importance using an atomistic MD approach (LAMMPS [7]) for the smaller scales, and a coarse-grained approach (LAMMPS in coarse-grained mode) for the larger scales. The molecular system is modelled on a spatial scale ranging from roughly a nanometer to tens of microns. The three submodels are coupled in an acyclic, or loosely coupled, multiscale modelling scheme, and are run on a distributed computing infrastructure.

6.2. Tools support for clay-polymer interaction

Fig. 4 shows building and executing of the simulation using MAD, GridSpace and AHE. First, a user builds the application from blocks by connecting their inputs and outputs. Next, the visual connection scheme is transformed into a GridSpace experiment definition, which consists of a chain of scripts processed by different interpreters. There are two particular executors available from interpreters needed by the application: the SSH executor which can be used for tests and the AHE executor which is used in a production phase. In Fig. 5 we show the

Table 1. Multiscale application creation steps. For every step, we describe if and how it can be supported by the tools, automating various tasks for the user.

| Action | Tools support |
|--|--|
| design and implementation of single scale modules | done by a user usually using scientific software like LAMMPS or CPMD |
| conceptual modeling multiscale phenomena | done by a user |
| design and implementation of glue modules | done by a user using Perl, CPMD2CUBE etc. |
| description of single scale models | the description is registered to MaMe by using an interactive user interface; |
| design of connection scheme between single scale modules | interactive visual design in MAD; the previously designed connection schemes can also be loaded from MaMe (a few milliseconds); |
| preparation of executable application from connection scheme | automatic generation of executable GridSpace Experiment by MAD (a few milliseconds) – this assumes that implementation of single scale modules are already available |
| mapping modules to possibly different services that access e-Infrastructures; setting parameters of these services | done by a user from the single web interface (GridSpace)– interactive process that usually takes from a few seconds to a few minutes; |
| execution of modules - exploratory programming | goal is to facilitate an application run, not improve performance - initialization of execution is done from a single web interface by pressing the run button; there is a possibility of displaying standard error and output from GridSpace EW that enables quick result inspection (useful for tests and refinement); |
| switching services (executors) assigned to modules e.g. from ssh to AHE | done by a user from the single web interface (GridSpace) – interactive process that usually takes a few seconds and does not require any changes in an application; |
| execution of modules - production run | using AHE allows for transparent access to different types of grid middleware with single sign-on. |
| fetching results | automatically fetched and visible in GridSpace by means of standard protocol ssh and GridFTP (time depends on data size and protocol performance) |
| viewing results | visible in GridSpace |
| redesign of the connection scheme setting single scale modules in different configurations | interactive visual design in MAD; |

graphical structure of the described simulation in MAD. It consists of three computationally intensive modules: quantum mechanics (simulated by CPMD), classical molecular dynamics (simulated by LAMMPS), and coarse grained molecular dynamics (simulated again by LAMMPS). The output from one simulation module is transformed into the input of another simulation using additional tools (CPMD2CUBE, MSI3LMP_POT) and general purpose scripts (Perl). In MAD, it is also possible to modify the actual script implementing the model, and to switch between different implementations of the model. The presented graphical scheme can be translated into the GridSpace experiment definition which consists of different snippets (each corresponding to one graphical

module) as shown in the Fig. 6. Each element of the application is, in fact, the snippet with the script code and defined input and output files. A user can switch between different executors (SSH, AHE) for the same snippets. The user also choose the machine on which the calculations are performed. On the left side of Fig. 6 one can see files available through GridFTP on a storage server from where they can be staged by AHE to the machine where the job is executed (see also Fig. 3). For our use case, we used NGS resources (mavrino.chem.ucl.ac.uk), PL-Grid (zeus.cyfronet.pl) and PRACE (huygens.sara.nl) infrastructures.

The goal of the tools we presented here is mainly to facilitate composability, execution on various e-Infrastructures and reusability of multiscale applications. However, the usage of the tools themselves does not hinder the actual execution performance. In Table 1 we summarize the actions required to create and execute the nanomaterials application. For each of these actions, we describe if and how it was supported by the tools. For case of Multiscale Application Designer (MAD), MAPPER Memory (MaMe) and GridSpace actions are facilitated by providing interactive interface to the user. If the action can be fully automated, we also provide the estimated time of the action.

7. Summary

In this paper we described a support solution for multiscale applications that are created from modules which rely on scientific software packages. We focus on applications that are constructed using scripts of different languages: either general purpose languages (e.g. Python, Perl, Ruby) or domain specific languages for scientific software (e.g. LAMMPS, CPMD, Gaussian, MATLAB). By joining this script-based approach with the application composition methodology offered by GridSpace and the transparent access to various infrastructures offered by AHE, we offer users a convenient and transparent way of composing and running applications in both test and production mode on production resources (e.g. as offered by EGI and PRACE).

Additionally, we provide support for reusability by offering the MAPPER MeMory tool which stores information about application elements and the way they can be connected. Different versions of the application can be visually composed in the Multiscale Application Designer.

As a case study, we have presented a simulation of clay-polymer interactions, which relies on CPMD for quantum mechanics and LAMMPS for classical and coarse-grained molecular dynamics. Here we show that our support tools allow users to automate part of the work required to run and manage their multiscale simulations. The tools usage does not influence application performance, but facilitates the set up and execution on various e-Infrastructures. In the future we plan to optimize data transfer between staging GridFTP Storage server and the chosen e-Infrastructure by enabling to run multiple parts of the experiments (multiple snippets) in a single job of an executor.

Acknowledgements

The authors wish to thank Alfons Hoekstra, Joris Borgdorff, Bastien Chopart and Mohamed Ben Belgacem for discussions on MML. The authors wish also to thank Mariusz Mamonski, Krzysztof Kurowski and Ilya Saverchenko for valuable discussions on using e-Infrastructures. The research presented in this paper was partially supported by the MAPPER project – grant agreement no. 261507 and the AGH grant 15.11.120.090.

References

- [1] A. Hoekstra, J. Kroc, P. Sloot (Eds.), *Simulating Complex Systems by Cellular Automata*, Understanding Complex Systems, Springer, 2010.
URL <http://springer.com/978-3-642-12202-6>
- [2] Weinan, *Principles of Multiscale Modeling*, 1st Edition, Cambridge University Press, 2011.
URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1107096545>
- [3] J. Fish, *Multiscale Methods: Bridging the Scales in Science and Engineering*, Oxford University Press, Inc., New York, NY, USA, 2009.
- [4] J. L. Suter, R. L. Anderson, H. Christopher Greenwell, et al., Recent advances in large-scale atomistic and coarse-grained molecular dynamics simulation of clay minerals, *J. Mater. Chem.* 19 (2009) 2482–2493. doi:10.1039/B820445D.
URL <http://dx.doi.org/10.1039/B820445D>
- [5] E. Ciepiela, P. Nowakowski, J. Kocot, D. Harężlak, T. Gubała, J. Meizner, M. Kasztelnik, T. Bartyński, M. Malawski, M. Bubak, Managing entire lifecycles of e-science applications in the gridspace2 virtual laboratory—from motivation through idea to operable web-accessible environment built on top of pl-grid e-infrastructure, *Building a National Distributed e-Infrastructure—PL-Grid* (2012) 228–239.

- [6] S. Zasada, P. Coveney, Virtualizing access to scientific applications with the application hosting environment, *Computer Physics Communications* 180 (12) (2009) 2513 – 2525, [jce:title;40 YEARS OF CPC: A celebratory issue focused on quality software for high performance, grid and novel computing architectures;jce:title; doi:10.1016/j.cpc.2009.06.008](http://dx.doi.org/10.1016/j.cpc.2009.06.008).
URL <http://www.sciencedirect.com/science/article/pii/S0010465509001830>
- [7] LAMMPS, Large-scale atomic/molecular massively parallel simulator, <http://lammps.sandia.gov/> (2011).
- [8] CPMD, Parallelized plane wave / pseudopotential implementation of density functional theory, <http://www.cpmc.org/> (2011).
- [9] GAUSSIAN, <http://www.gaussian.com/>.
- [10] D. Barseghian, I. Altintas, M. B. Jones, D. Crawl, N. Potter, J. Gallagher, P. Cornillon, M. Schildhauer, E. T. Borer, E. W. Seabloom, P. R. Hosseini, "workflows and extensions to the kepler scientific workflow system to support environmental sensor data access and analysis", *Ecological Informatics* 5 (2010) 42 – 50.
- [11] P. Missier, S. Soiland-Reyes, S. Owen, W. Tan, A. Nenadic, I. Dunlop, A. Williams, T. Oinn, C. Goble, Taverna, reloaded, in: M. Gertz, T. Hey, B. Ludascher (Eds.), *SSDBM 2010*, Heidelberg, Germany, 2010.
URL <http://www.taverna.org.uk/pages/wp-content/uploads/2010/04/T2Architecture.pdf>
- [12] S. Koulouzis, R. Cushing, K. A. Karasavvas, A. Belloum, M. Bubak, Enabling Web Services to Consume and Produce Large Datasets, *IEEE Internet Computing* 16 (1) (2012) 52–60.
- [13] M. Malawski, T. Gubala, M. Bubak, Component-based Approach for Programming and Running Scientific Applications on Grids and Clouds, *IJHPCA* 26 (3) (2012) 275–295.
- [14] K. Rycerz, M. Bubak, Building and Running Collaborative Distributed Multiscale Applications, in: W. Dubitzky, K. Kurowski, B. Schott (Eds.), *Large-Scale Computing Techniques for Complex System Simulations Wiley Series on Parallel and Distributed Computing*, Vol. 1, John Wiley & Sons, 2012, Ch. 6, pp. 111–130.
- [15] S. J. Zasada, M. Mamonski, D. Groen, J. Borgdorff, I. Saverchenko, T. Piontek, K. Kurowski, P. V. Coveney, Distributed infrastructure for multiscale computing, in: *Distributed Simulation and Real Time Applications (DS-RT)*, 2012 IEEE/ACM 16th International Symposium on, 2012, pp. 65 –74. doi:10.1109/DS-RT.2012.17.
- [16] Uniform Interface to Computing Resources, <http://www.unicore.eu/>.
- [17] K. Kurowski, B. Bosak, P. Grabowski, et al., Building and Running Collaborative Distributed Multiscale Applications, in: W. Dubitzky, K. Kurowski, B. Schott (Eds.), *QosCosGrid e-Science Infrastructure for Large-Scale Complex System Simulations*, Vol. 1, John Wiley & Sons, 2011, Ch. 9, pp. 163–183.
- [18] Lightweight Middleware for Grid Computing, <http://glite.cern.ch>.
- [19] E. Ciepiela, D. Harezlak, J. Kocot, T. Bartyński, M. Kasztelnik, P. Nowakowski, T. Gubała, M. Malawski, M. Bubak, Exploratory programming in the virtual laboratory, in: *Proceedings of IMCSIT 2010*, Wisla, Poland, 2010.
- [20] P. Nowakowski, E. Ciepiela, D. Harezlak, J. Kocot, M. Kasztelnik, T. Bartyński, J. Meizner, G. Dyk, M. Malawski, The collage authoring environment, *Procedia CS* 4 (2011) 608–617.
URL <http://dx.doi.org/10.1016/j.procs.2011.04.064>
- [21] T. Gubala, M. Bubak, P. M. Sloot, Semantic Integration of Collaborative Research Environments, *Information Science Reference IGI Global*, 2009, Ch. XXVI, pp. 514–530.
- [22] M. Kasztelnik, M. Bubak, Grid Resource Registry - Abstract Layer for Computational Resources, *Computer Science* 12 (2011) 33–44.
URL <http://csci.iisg.agh.edu.pl/114/>
- [23] J.-L. Falcone, B. Chopard, A. Hoekstra, MML: towards a Multiscale Modeling Language, *Procedia Computer Science* 1 (1) (2010) 819 – 826, ICCS 2010. doi:DOI: 10.1016/j.procs.2010.04.089.
URL <http://www.sciencedirect.com/science/article/pii/S1877050910000906>
- [24] F. Bergaya, B. Theng, G. Lagaly, *Handbook of clay science*, Vol. 1, Elsevier Science, 2006.
- [25] D. Groen, J. Suter, P. V. Coveney, Modelling distributed multiscale simulation performance: An application to nanocomposites, in: *Seventh IEEE international conference on e-Science and Grid computing*: Stockholm, Sweden, IEEE Computer Society, Piscataway, NJ, 2011, pp. 105–111.
- [26] J. Suter, D. Groen, L. Kaban, P. V. Coveney, Distributed multiscale simulations of clay-polymer nanocomposites, in: *Materials Research Society Spring Meeting*, Vol. 1470, MRS Online Proceedings Library, San Francisco, CA, 2012. doi:10.1557/opl.2012.1009.