PROCEEDINGS OF SCIENCE

InSilicoLab – A User Workspace Implementing e-Science Principles

Joanna Kocot[1]
ACC Cyfronet AGH
ul. Nawojki 11, 30-950 Krakow, Poland
E-mail: ymkocot@cyfronet.pl

Tomasz Szepieniec, Mariusz Sterzel, Daniel Harężlak, Klemens Noga
ACC Cyfronet AGH
ul. Nawojki 11, 30-950 Krakow, Poland
E-mail: t.szepieniec@cyfronet.pl, m.sterzel@cyfronet.pl, d.harezlak@cyfronet.pl, k.noga@cyfronet.pl

E-science is a powerful instrument of contemporary research. Among its principles, the collaboration between globally dispersed groups of scientists is usually pointed as most important. Typically, e-science involves also using large-scale computing resources and large data collections.

InSilicoLab is an application portal, that supports these aspects of research, by facilitating the access to computational software deployed on grids and the management of data and processes involved in such scientific computations. To meet the researchers' needs, the portal provides mechanisms to track the processes that lead to obtaining valuable data, as well as for sharing these data and processes with fellow scientists enabling their collaborative work. The record of a computation process may be used to repeat it, using different input parameters - which is a simple, yet, powerful approach towards workflow execution.

EGI Community Forum 2012 / EMI Second Technical Conference,
Munich, Germany
26-30 March, 2012

[1] Speaker

1. Introduction

In the era of rapid technological progress, when computer centres constantly improve and extend their offer of computer resources, the level of know-how required to use this power stays relatively high. The environment presented in this paper, called InSilicoLab, not only offers an easy access to computational and storage resources, but it is designed especially to assist researchers in solving computational problems of their specific domain of science.

The motivation behind the development of this tool is rooted in the observation that the language used by the researchers in their daily work and the language used for operating computing infrastructures are very different. We intend to bridge this semantic gap by introducing a so-called mediation layer, that translates the goals expressed in the researcher's domain language into tasks understood by computational infrastructure. Defining the functionality and providing implementation of this layer is the main achievement of our work [1].

With the proposed solution, the researchers, assisted by developers, define the computational processes that lead them to results needed to solve their research problem. Together, they form so-called experiments – a combination of computational operations that, from the input data provided by the researcher, produce valuable results. This includes running domain-specific applications on computational resources, as well as managing the input/output data, gathering and initial analysis of the results. Such experiments, being specific to a given domain of science, at the same time are generic enough to be used by various groups of researchers from that domain. What is more, the InSilicoLab portal creates a user workspace, where the scientists not only carry out their daily work, but also can access all the data they gathered in the course of experimentation.

2. InSilicoLab Architecture

Although the InSilicoLab concept was to serve in solving domain-specific problems, the architecture of the system remained generic – in this way, allowing it to be a base for different applications and experiments.

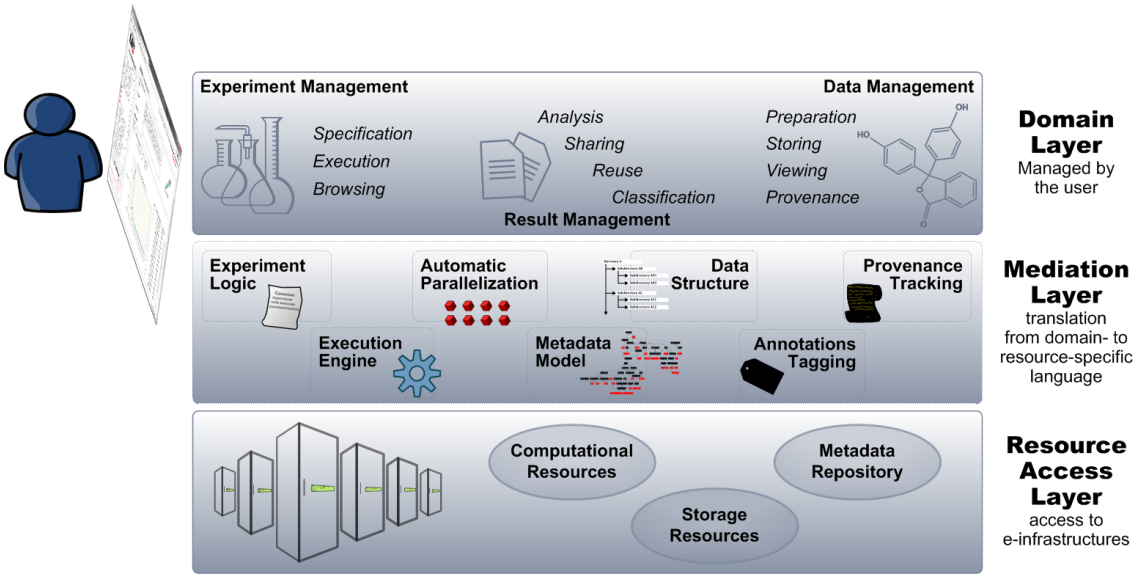

Fig. 1 Architecture of InSilicoLab, with three layers: Domain Layer, Mediation Layer and Resources Access Layer.

The architecture is based on a three-layer concept, where the interaction with the end-user – the scientist is handled entirely by a layer called Domain Layer (see Fig. 1). As responsible for the contact with the users, it has to assure maximum usefulness to them. Therefore, the Domain Layer operates only on actions and objects from the user's domain – Experiment, Analysis, Results etc.

On the opposite side of the computation process, there are the computational and storage resources. The resources, too, have to be accessed in a specific way, using direct commands or APIs. As the objects from the Domain Layer are incompatible to the resource commands, there was a need for introducing a Mediation Layer – responsible for translation between the sphere of the researcher's domain and the layer of resource access. This requires performing all the actions the user would need to run their computation on the underlying resources, which is done by the components of the Mediation Layer:

- *Experiment logic* – an entity organising the user's computations into a workflow that constitutes the experiment (understood as described in Section 1). It provides a description of the operations that have to be performed to create computational tasks from the user's input data, to monitor these tasks and to analyse their results.
- *Automatic parallelisation* – a service capable of automatically dividing the computation into independent computational jobs running in parallel. It defines the jobs, distributing the parts of computation so that an optimum number of computational tasks is grouped in a job.
- *Execution engine* – a service responsible for conducting and monitoring of all the operations defined by the experiment logic. The engine is an executor of the – to this moment abstract – workflow, on the computational resources.
- *Storage structure* – a specific storage space organisation model, used for managing raw files created by the experiment and each of its jobs. It is applied to the storage resources

to organise the data files, allowing for easier access to the files produced by the research experiment.

- *Data model* – a model of the data storage – applied to all data (other than raw files) that is used for the computation or obtained in its course.
- *Metadata description* – a model of the metadata attached to the data and files produced or used by an experiment. Such metadata may be used to organise and allow searching within data or raw files and to create relations between the data and processes.
- *Provenance tracking* – a service responsible for recording all the transformations and usage of data relevant to the user and their experimentation process. In this way, every experiment may be retraced and the origin of every data entity stored in the system may be identified.

The components of the Mediation Layer interact with each other to conduct all the processes the Domain Layer requires (see Fig. 2). What is more, they are functionally joined by provenance tracking service, which takes part in all the actions performed on the data that might be relevant to the user.

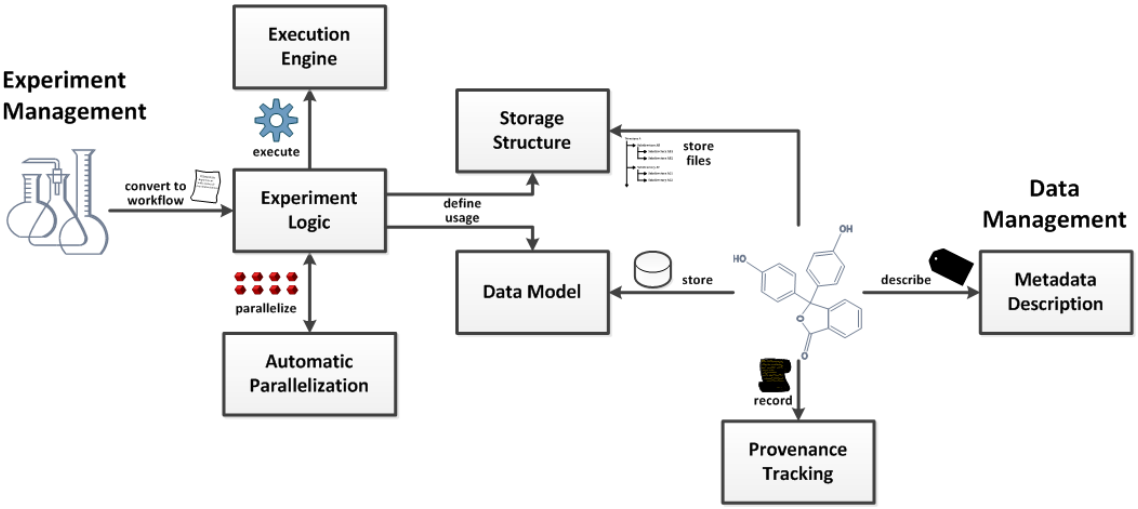

Fig. 2 The components of the InSilicoLab Mediation Layer, and interactions between them

3. Application to Specific Domain Problems

The InSilicoLab portal was already used by experts from two scientific domains: computational chemistry and astrophysics. In each of these domains, there are some typical problems, the researchers solve with the use of (usually large) computational power. When performed individually, their tasks usually require much attention and manual work to analyse the results. By close cooperation of the portal developers and the domain-experts, we were able to identify such tasks, and provide a comprehensive solution to a larger problem. In this way, the portal now features several applications – each one realising the whole process leading from the input data the researcher wants to use, to pre-analysed complex results, being often a compilation of several tasks.

In this section we present our approach to solving several computational problems in the domain of computational chemistry and astrophysics.

3.1 Computational Chemistry

The domain of computational chemistry is an interesting study, not only as a field of computing-intensive research, but also due to its specific nature. The computations in this domain often need several iterations using various (sets of) tools. This introduces a high level of complexity in managing the computation process and the data it consumes and produces.

One of the typical computational problem in chemistry is performing a so-called conformation scan – a process of searching for an optimum configuration of a molecule. For this purpose, a set of similar molecules is generated from an initial input – by transforming the input molecule by rotation at a given bond, angle, or dihedral. Each generated molecule is then used as an input to one of the programs that the chemists usually use to compute its energy. By analysing the results of each of such program runs, the scientist is able to identify the molecule which is the best conformation – usually the one with the lowest energy. Such molecule can be then used for more profound analysis – done with another chemistry program.

By analysing this scenario from the point of view of distributed computation, we can observe the following characteristics:

- Both input and output of the computation are molecules, which are generated or/and transformed in the process.
- There can be large number of molecules that need to be managed properly.
- The processing of one molecule is independent from the others, therefore the computation can be performed in parallel without introducing communication overheads.
- Molecules can serve as input to various applications, which might require different format of molecule representation. Therefore, conversion between formats is an important feature enabling integration between different applications.

What is more, the results of the aforementioned chemistry programs are usually large text files, what makes analysing them extremely tedious work. The computations are also time- and resource-consuming, therefore, performing them on a desktop computer or a laptop, may take substantial amount of time.

In the InSilicoLab portal, all this operations may be performed by executing a single "experiment" through a Web interface (see Fig. 3). The portal allows the scientist to specify the input molecule and its parameters, as well as input data to a chosen chemistry program (all these parameters are well-known to the researcher). It then automatically parallelises the computation to as many computational jobs, as needed, and runs them on the Grid. The jobs are continuously monitored, and their results are analysed on-line and appear in the portal. Specialised diagrams offer a summary of the results, allowing the user to find the optimum molecule, without arduous analysis of each of the task's output. The chosen molecule can be then easily reused as an input to another experiment – e.g., a one that uses another chemistry program.

The input geometries can be specified in different formats, and transformed to others, thanks to a built-in mechanism of molecule translation (based on OpenBabel [2]).

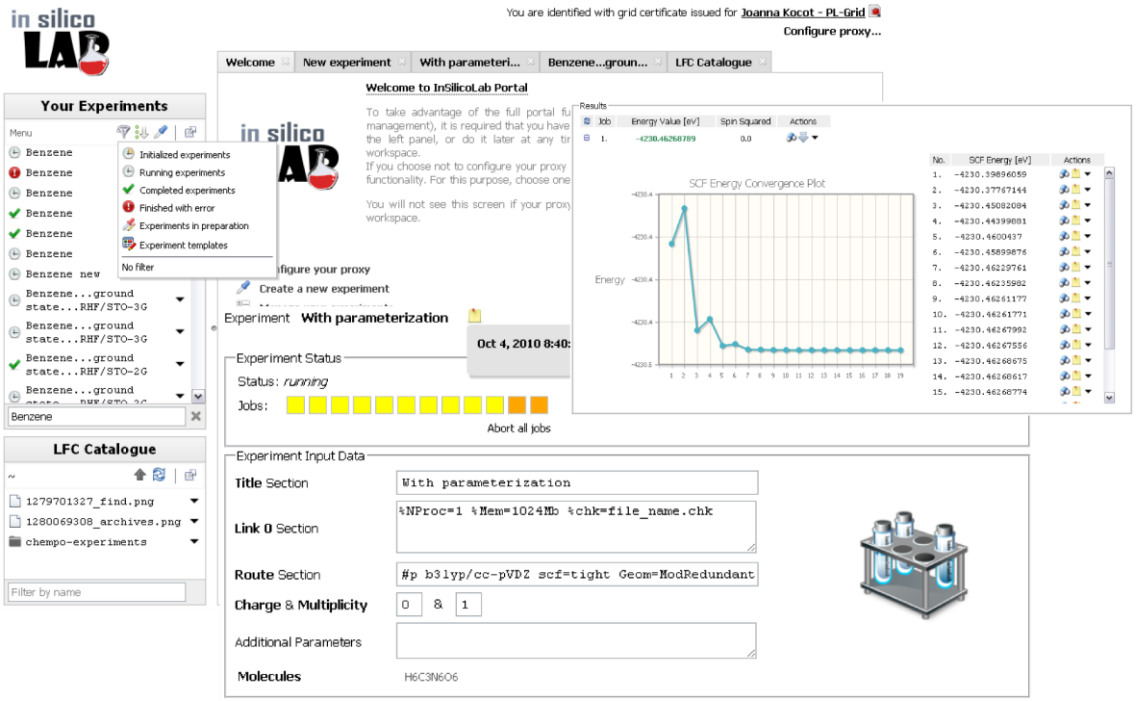

Fig. 3 A view on the input specification, job monitoring, graphical representation of the results obtained for a conformation scan in chemistry experiment.

The chemistry programs currently supported by the InSilicoLab portal are: Gaussian [3], GAMESS [4] and TURBOMOLE [5].

3.2 Astrophysics – the Cherenkov Telescope Array Project

The Cherenkov Telescope Array (CTA) [6] project is an international initiative to build an array of telescopes for observing very high energy gamma rays. It will be a ground-based instrument, more powerful than all the current installations of this kind.

As the project is now in its design phase, the astrophysicists engaged in it, are now analysing different configurations of the telescope array, in order to find an optimum one. The configuration of the telescopes includes large number parameters, therefore, the space of the search and the computational potential required is huge. Using a simulated atmospheric showers (the images of the Cherenkov radiation), they simulate the telescope array with requested parameters, compute several result values, and compare the results of different simulations. For a simulation of the telescope array, a dedicated program – *sim_telarray* is used.

Each of the simulations has to be tested on a number of different atmospheric showers. The showers are very large and so are the resulting files generated by the sim_telarray programs. Therefore, scanning all the parameter space and testing with large number of input files, together with gathering the scan results from all the program runs is a very time-consuming and laborious work.

The "experiments" for CTA in the InSilicoLab portal (see Fig. 4) allow to perform all the above mentioned work from a single location, only requiring the initial configuration of the telescope array, the test shower files and the range of configuration parameters to be scanned. Many of the simulated shower files were stored in the LFC (LCG File Catalogue) by the CTA consortium, and can be automatically picked from there to be included in the computations.

InSilicoLab assures that the simulation program will be run on the Grid infrastructure with requested parameters – creating as many parallel jobs, as required to scan through the user-defined range of parameters, for all the shower files. The job results are then automatically gathered and presented in form of tables and diagrams, to facilitate their analysis. All of them can be exported to a user-chosen format, to enable further analysis.

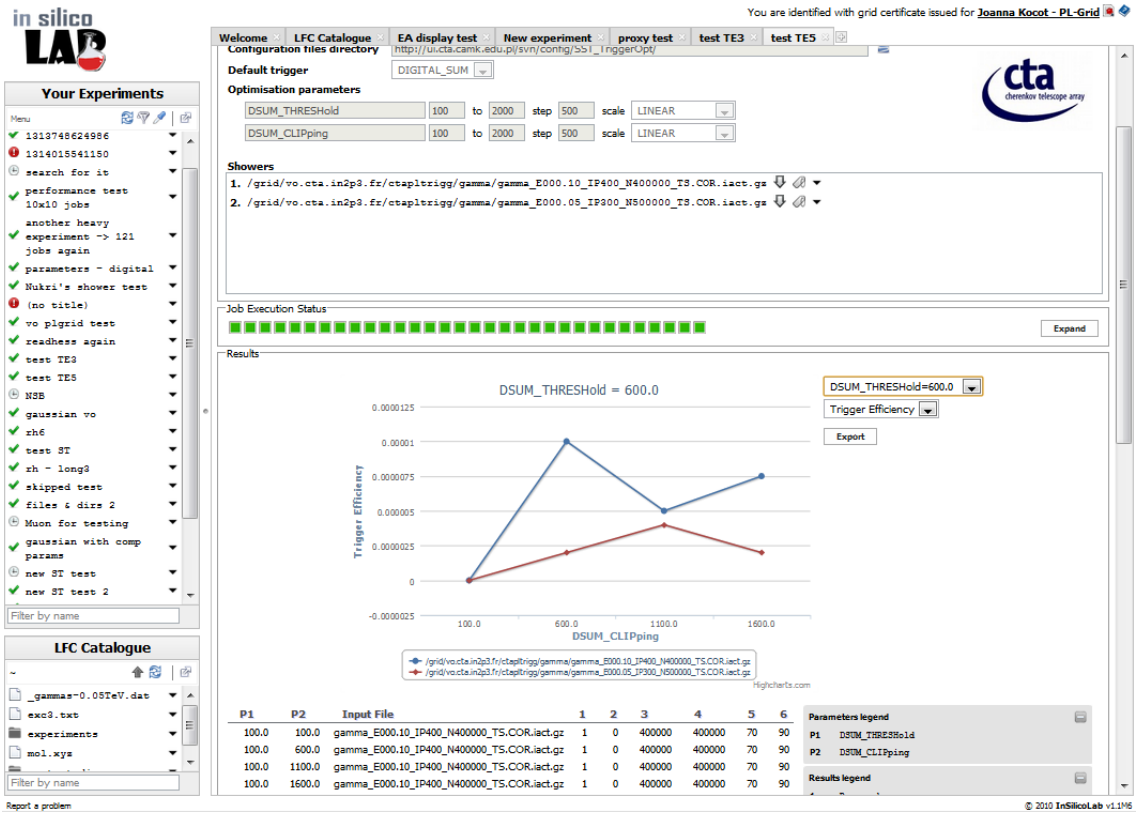

Fig. 4 A view on the analysed and gathered results for different configurations and two shower files in telescope array simulation experiment.

The portal enables also gathering and analysis of the results obtained by the researchers by manually running the simulations (outside the portal).

4. Conclusions

The implementation of e-science principles in contemporary research, especially in the case of computationally intensive and conceptually complex *in silico* experiments, seems an important and natural path for the development of tools for science.

An approach that is presented in this paper, in the shape of the InSilicoLab portal, allows separation of the domain-specific concepts that are close to the researcher from the issues related to the infrastructure they want to use for their computations. This makes the scientists' work much more efficient, as they may focus only on the information and activities relevant to their research, instead of learning the technical details of the underlying resources. Another obstruction in a researcher's daily work is looking for their data in many different locations – local files, cluster storage, LFC, etc. The InSilicoLab portal aims also at eliminating this problem – becoming a workspace, where all the data required for the experimentation process is available along with all the results obtained in its course.

The development of such tools as InSilicoLab cannot be successful without collaboration with the scientist who would use them, and their validation and feedback afterwards. Therefore, we stay in close contact with the domain-experts, and continuously offer them improved versions of our product.

The InSilicoLab portal is now accessible to all the members of Gaussian VO [7], vo.plgrid.pl [8] and vo.cta.in2p3.fr [9] at http://insilicolab.grid.cyfronet.pl/ (stable release for chemistry applications) and http://ctaportal.grid.cyfronet.pl/ (beta-stage release for the CTA project). The access to the portal is granted on the base of user certificate installed in the browser and respective Virtual Organisation membership.

References

[1] J. Kocot, T.Szepieniec, D. Harezlak, K. Noga, M. Sterzel: *InSilicoLab – Managing Complexity of Chemistry Computations*; in PL-Grid: Building a National Distributed e-Infrastructure.

[2] *Open Babel: The Open Source Chemistry Toolbox*. http://openbabel.org

[3] M. J. Frisch et al., *Gaussian 09*; Gaussian, Inc., Wallingford CT, 2009.

[4] M.W.Schmidt, K.K.Baldridge, J.A.Boatz, S.T.Elbert, M.S.Gordon, J.H.Jensen, S.Koseki, N.Matsunaga, K.A.Nguyen, S.Su, T.L.Windus, M.Dupuis, J.A.Montgomery: *General Atomic and Molecular Electronic Structure System*; in: J. Comput. Chem., 14, 1347-1363(1993).

[5] *TURBOMOLE - Program Package for ab initio Electronic Structure Calculations*. http://www.turbomole.com/

[6] *The Cherenkov Telescope Array project*: https://www.cta-observatory.org/

[7] *Gaussian VO*. http://egee.grid.cyfronet.pl/Applications/gaussian-vo/

[8] *PL-Grid - Polish Infrastructure for Supporting Computational Science in the European Research Space*. http://www.plgrid.pl/

[9] *vo.cta.in2p3.fr*: https://cclcgvomsli01.in2p3.fr:8443/voms/vo.cta.in2p3.fr/