Application scenarios using serpens suite for Kepler scientific workflow system

Citation for the original published paper (version of record):

Plociennik, M., Owsiak, M., Żok, T. et al (2012). Application scenarios using serpens suite for Kepler scientific workflow system. Procedia Computer Science, 9: 1604-1613.
http://dx.doi.org/10.1016/j.procs.2012.04.176

(article starts on next page)

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 9 (2012) 1604 – 1613

Procedia Computer Science

International Conference on Computational Science, ICCS 2012

Application Scenarios Using Serpens Suite for Kepler Scientific Workflow System

Marcin Płóciennik[a]*, Michał Owsiak[a], Tomasz Zok[a], Bartek Palak[a], Antonio Gómez-Iglesias[b], Francisco Castejón[b], Marcos Lopez-Caniego[c], Isabel Campos Plasencia[c], Alessandro Costantini[d], Dimitriy Yadykin[e], Pär Strand[e]

a Poznan Supercomputing and Networking Center, Noskowskiego 12/14,61-704 Poznan, Poland
b Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, Av. Complutense 22. 28040,Madrid, Spain
c Department of Advanced Computation and e-Science, Instituto de Fisica de Cantabria, CSIC, 39005, Santander, Spain
d Dept. of Chemistry, University of Perugia Via Elce di Sotto, 8, 06123,Perugia, Italy
e Dept. of Radio and Space Science Chalmers University of Technology, SE-412 96 Goteborg, Sweden

Abstract

This paper presents the overview of exploitation scenarios making use of the Serpens suite for the Kepler workflow orchestration system. The proposed framework provides researchers with an easy-to-use, workflow-based environment for scientific computations. It allows execution of various applications coming from different disciplines, in various distributed computational environments using a user-friendly interface. This research has been driven initially by Nuclear Fusion applications' requirements, where the leading idea was to enhance the modeling capabilities for ITER sized plasma research by providing access to High Performance Computing resources. Several usage scenarios are being presented with an example of applications from the field of Nuclear Fusion, Astrophysics and Computational Chemistry.

Keywords: Kepler, workflows, Serpens suite, grid, distributed computing, application use cases,

1. Introduction

As computational capabilities are getting increasingly powerful, efficient use of the available infrastructure is becoming more and more complex from users' perspective. Clearly, scientists prefer to focus on their research and not on issues related to platforms and tools required to perform it. This is particularly important while developing the complex application scenarios using scientific workflow tools. The goal of the scientific workflows is to automate repetitive tasks (that are usually composed of several depended actions). They can greatly support researchers in such activities such data flow, simulations, modeling, analysis, etc. In many cases workflow tasks are computing-intensive. As an example the complex simulation use cases (as in the case of Nuclear Fusion

* Corresponding author. Tel.: +48 618582181.
E-mail address: marcinp@man.poznan.pl.

doi:10.1016/j.procs.2012.04.176

applications) are usually using a heterogeneous, distributed environment since the different codes' characteristics imply the usage of a different kind of computational resources. In this case the scientific workflows framework normally needs to jointly access both High Throughput Computing (HTC) driven infrastructures (EGEE, OSG, etc.) and High Performance Computing (HPC) driven ones, such as DEISA or TeraGrid. In this context, the difference between HPC and HTC is that HPC resources (e.g. supercomputers, large-scale clusters) provide a good interconnection of CPUs/cores while HTC resources (PC pools, smaller clusters, etc.) do not. From the practical point of view, in the current state of the art grid middlewares are used for implementation of such access. They support the sharing of geographically distributed, heterogeneous computing and storage resources. Grid middlewares provide a resource abstraction layer to the user. However, since application execution and data access should be done without exposing details of the middleware, usually the high level software and tools are required.

This paper presents the concept of the Serpens suite for the Kepler workflow orchestration system[1] and different scenarios based on Serpens. Using Kepler's graphical interface and components scientists can create and execute scientific workflows which allow accessing scientific data and executing complex analyses on them. The Serpens suite presented in this paper provides a set of Kepler actors and workflows enabling distributed execution. Its modules leverage and integrate various, already existing, middleware components and provide the user with a possibility to work with such grid middleware as UNICORE[2] and gLite[3] directly from Kepler. Serpens has been committed as a part of the main Kepler repository (BSD license). It is released with the newest Kepler 2.x version.

This research has been driven initially by Nuclear Fusion applications' requirements, where the leading idea was to enhance the modeling capabilities for ITER sized plasma research by providing access to High Performance Computing resources. Serpens development has been initiated as part of the EU FP7th EU Fusion for the ITER Applications (EUFORIA) project. The project was focused on the optimization and integration of a set of applications for edge and core transport modeling targeting different computing paradigms as needed (serial and parallel grid computing and HPC). The developed framework allowed the dynamic coupling and integration of codes and applications running on a set of heterogeneous platforms into a single framework through a workflow orchestration based on Kepler.

The article is organised as follows. In Section 2 we analyze the state of the art, while in Section 3 we introduce the Serpens suite and its modules. In the subsequent section we describe generic workflow use cases implemented with the help of the Serpens suite. In the next part we demonstrate the application scenarios exploiting the generic workflows. Several usage scenarios are presented with an example of applications from the field of Nuclear Fusion, Astrophysics and Computational Chemistry. We also provide directions for further work.

2. Related Work

Presented later in this section are several examples of workflow frameworks that target at supporting distributed infrastructures using the Grid middleware. In respect of grid middleware initiatives, looking at the European scene, main middleware solutions are gLite, ARC [4], Globus Toolkit[5] and UNICORE. The European Grid Initiative project is now promoting Universal Middleware Distribution that includes packages from the European Middleware Initiative (gLite, ARC, UNICORE) and Initiative for Globus in Europe. Most grid resources currently available in Europe are accessible via the gLite middleware. gLite is a middleware developed and deployed by the EGEE project and it is focused on providing an HTC infrastructure for the support of distributed processing of very large data volumes with sequential batch jobs. Although access to HPC systems is typically accomplished using a direct SSH login, several HPC centers in Europe (DEISA, PRACE) use UNICORE for exposing their resources to external users. UNICORE is a middleware that implements some of the available standards for grid systems in order to provide secure and seamless access to the HPC resources. In the US it is worth mentioning OSG[6], TerraGrid[7] (and XSEDE). TeraGrid is an e-Science grid computing infrastructure. TeraGrid resources are integrated through a service-oriented architecture and computational resources run a set of software packages called "Coordinated TeraGrid Software and Services" (CTSS). CTSS includes the middleware like Globus Toolkit, Condor. Since TeraGrid project has finished, it is followed on by the Extreme Science and Engineering Discovery Environment (XSEDE). The Open Science Grid facilitates distributed computing for scientific research. The OSG infrastructure ships a common package of software solutions provided and supported by OSG that includes Condor and Globus technologies.

There are several systems that provide a convenient way of building and executing these workflows. We would like to mention the most popular systems like Pegasus[8], Taverna[9], Triana[10], P-GRADE[11], and Kepler. Pegasus is a framework for mapping scientific workflows onto distributed resources including grid and Cloud-based systems. Pegasus has been used for extensive grid experiments and is focusing on middlewares like Condor, Globus, or Amazon EC2. The P-GRADE Grid Portal is a web-based, service rich environment for the development, execution, and monitoring of workflows and workflow-based parameter studies on various grid platforms. P-GRADE is focused on interoperability with Globus Toolkit 2, Globus Toolkit 4, LCG and gLite grid middlewares. Triana is a workflow-based graphical problem-solving environment independent of any particular problem domain. Triana workflows have incorporated a range of distributed components such as grid jobs and Web services. Taverna is an open source and domain independent Workflow Management System. Taverna has targeted bioscience workflows composed of Web services. The supported grid middleware is KnowARC, and gLite. Kepler is an open source, scientific workflow application. Using Kepler's graphical interface and components scientists can create and execute scientific workflows which allow accessing scientific data and executing complex analyses on them. Besides the Serpens suite described in this paper Kepler have the basic actors that supports directly Globus Toolkit. Also there is Nimrod [12] system that allows to build parameter sweep and search applications using the grid middleware. So called Nimrod/K is built on Kepler using Kepler's runtime engine. Nimrod/K provides the grid execution features of Nimrod/G. In Nimrod/K only one actor is needed for the whole action called a GridJob actor. This actor pre-stage files to the machine, request a processing slot and transfer output files directly to the next machine. The idea of the Nimrod/G is very similar to the Serpens suite, however the main difference is that whole grid functionality in Nimrod/G is embedded in one actor while in Serpens Kepler was used to build whole workflow of grid operations. From user perspective Serpens also provides also one actor (composite actor). Nimrod/G supports Globus2, GT4, Unicore and Condor while Serpens focus on gLite and UNICORE (supporting Globus2 and GT4 via VineToolkit module). There is no general criteria to say that one of the systems is better than others. They usually target different aspects (discussed in details in [13]) and are built basing on specific applications requirements. As an example the fusion community a detailed evaluation of existing technologies resulted in choosing Kepler as the main platform, since Kepler as data-flow driven workflow system with its directors concepts was the most convenient solution for managing and integrating fusion codes and ontologies.

3. Serpens suite overview

In the 2.0 version of Kepler, a new framework schema was introduced. Functionality is divided between modules which are gathered in suites. A separate module consists of source codes for actors it publishes, external libraries, licensing information and other Kepler-related metadata. A suite is an ordered list of modules. A user may decide to install a single module or a full suite which will ensure that all intermodular dependencies are met. Each module is versioned, and resources created in Kepler are saved with these version numbers in metadata. This prevents users from using workflows or composite actors in a non-compatible Kepler installation. The Serpens suite is an extension to Kepler that employs the mentioned approach. The Serpens suite [14] was developed initially within the EUFORIA project. It provides standard activities related to remote job management: submission, monitoring, data management, and so on, as a set of actors, both basic and composite. By combining a meaningful sequence of such actors, client applications can create a complete workflow in a very flexible and convenient manner. Kepler is responsible for user interaction and managing multi-site, distributed and heterogeneous workflows. The heterogeneity here implies that a single workflow, spanning multi-step activities, contains HTC and HPC executions. Serpens currently contains three modules: gLite and UNICORE and VineToolkit.

3.1. gLite Module

This module contains actors and workflows integrated with the gLite middleware. It allows creating VOMS proxies, transferring data using the Globus GridFTP protocol, submitting and managing jobs. The accompanying template workflow is a full use case implementation ready to be modified according to specific needs of the grid applications. The main loop of the workflow which manages multiple grid jobs is presented in Figure 1. Access to gLite HTC resources is provided using a dedicated Roaming Access Server (RAS). It is a layer intermediating between Kepler actors and the grid infrastructure. It is designed to hide the complexity of grid environment configuration, grid security and operations execution by publishing a set of well-defined web services which operate on selected middleware on behalf of the user. The web services are common and uniform for all middlewares supported by RAS. A unique feature of the proposed solution is that RAS implements a number of failure detection and prevention mechanisms. This means that if a problem with particular resources is predicted to happen, the RAS and Serpens actor take action to avoid these specific grid resources.

Several static information sources are incorporated to avoid using faulty infrastructure elements. This means ignoring unresponsive storage, erroneous workload managers or computation elements that often lead to jobs failure. Besides these static tests, jobs are monitored according to their status or directly by the logging and bookkeeping events. All data transfers are checked, especially if they concern output retrieval. It may happen that despite the job's successful finish, its output files get lost due to various networking, security or infrastructure issues. Serpens actors handle all these kinds of situations. This strategy led to a significant improvement of the success ratio for jobs managed via Serpens actors and workflows comparing to jobs submitted without these features applied. Serpens allows submitting up to thousands of jobs, managing them in spite of infrastructure issues (both random and scheduled), and keeping track of the overall progress even between separate Kepler sessions. It means that once a workflow finishes job submission, it can be stopped and executed at a later time. Generally speaking, Serpens modules will resume the execution of the workflow in place where it was terminated.

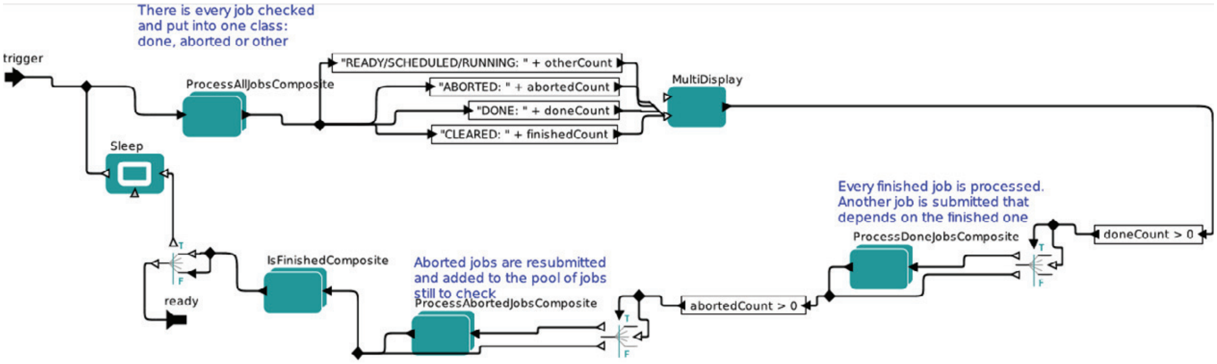

There is every job checked
and put into one class:
done, aborted or other
trigger
ProcessAllJobsComposite
"READY/SCHEDULED/RUNNING: " + otherCount
"ABORTED: " + abortedCount
"DONE: " + doneCount
"CLEARED: " + finishedCount
MultiDisplay
Sleep
Every finished job is processed.
Another job is submitted that
depends on the finished one
ProcessDoneJobsComposite
doneCount > 0
IsFinishedComposite
Aborted jobs are resubmitted
and added to the pool of jobs
still to check
ProcessAbortedJobsComposite
abortedCount > 0
ready
T
F

Figure 1 A main loop of gLite template workflow handling thousands of jobs.

3.2. UNICORE Module

This module is responsible for supporting the UNICORE middleware. Actors contained within are based on a UNICORE Commandline Client (UCC). A user can request information from the UNICORE registry, use available storages and sites, submit and manage jobs. This module also contains several example workflows presenting clearly the usage of each separate actor. There is also a full use case implementation which highlights the way to submit a job with input files, wait until it is successfully finished and retrieve its output. This is presented in Figure 2. The Kepler mechanism of composite actors enables to get this ready workflow as a single entity of a larger workflow, which makes the UNICORE module even more usable for a wider context.

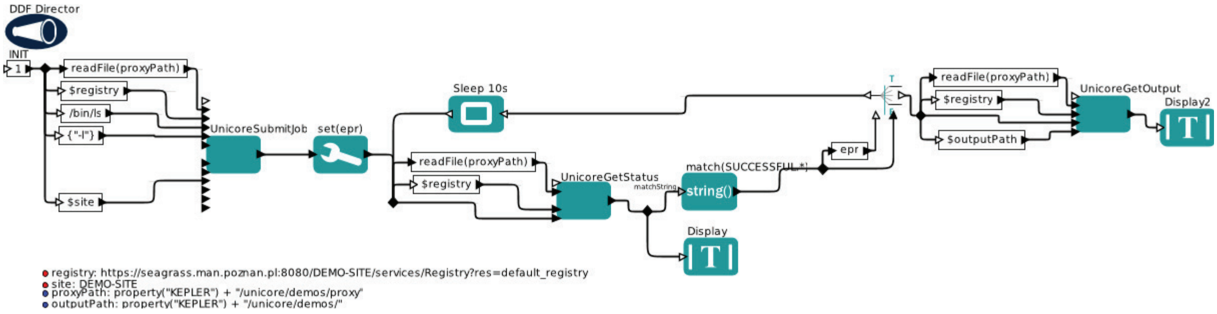


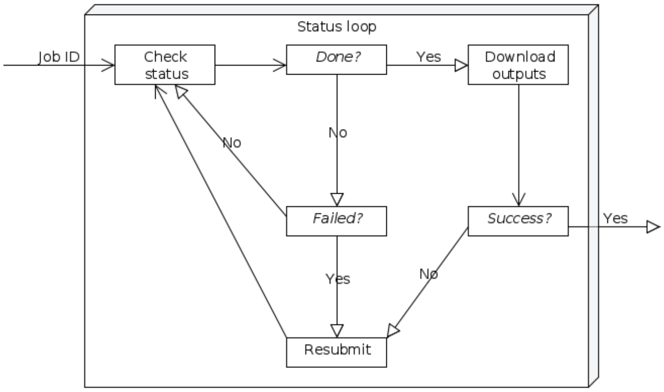
Figure 2 Full use case of the UNICORE composite actor.

3.3. Vine Toolkit Module

Vine Toolkit is a software package which unifies access to various grid middleware infrastructures. Using the same set of actions users can work with remote storages of different kinds and remote job management schemes in a simple and uniform way. Supported are particularly the UNICORE and Globus Toolkit 4, gLite middleware technologies. The Vine Toolkit module for Kepler contains actors which represent basic actions provided by this technology. There are also example workflows and, as with previous modules, a template workflow implementing a full job submission use case. The idea of implementing this module was to follow the concept of providing one generic case for job submission on the client side for all kinds of the grid middleware. The server side is using the OGF standards like BES (Basic Execution Service) to interact with different grid middleware (such approach is similar to the one described in [15]). This solution seems to be more scalable on the one hand, since from the client perspective it is grid middleware independent. However, it has the limited functionality as some of the specific functionalities of particular grid middleware's have not been exposed in the Vine Toolkit (that is the external software package). That is the reason why we have decided to keep and maintain 3 separate modules.

4. Generic Use Cases Description

In this section we have given a technical description of generic use case scenarios that are not domain-specific. These scenarios are implemented as generic Kepler template workflows that are later reused, configured and adapted by actual application use cases. Such workflows are also used as a components enabling a possibility of building larger complex application workflows scenarios.

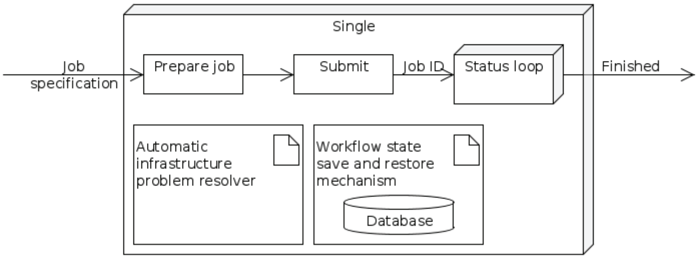


4.1. Status loop

This is the basic workflow used as a component by all other workflows. It performs the job state monitoring and takes care of downloading results and resubmitting if the application failed or the results are not accessible.

4.2. Single execution

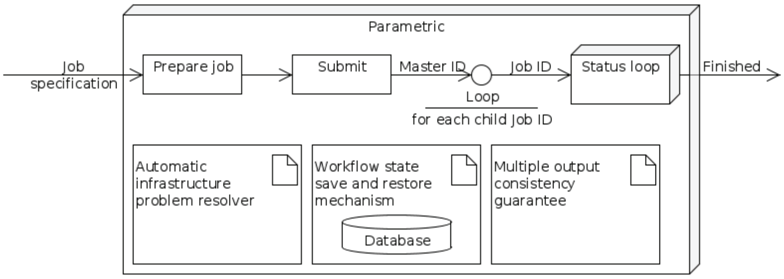
This is the simplest use case, where only one application is executed on the remote resources. This workflow is


Single
Job specification
Prepare job
Submit
Job ID
Status loop
Finished
Automatic infrastructure problem resolver
Workflow state save and restore mechanism
Database

built using several components. In the first step job specification is created basing on the user requirements. The inputs files are uploaded, and the job is submitted. In the next step the status loop workflow that is described in 4.1 takes care of job state monitoring, downloading results and resubmitting if needed. There are also components responsible for automatic infrastructure problem resolving and for saving and restoring job statuses.

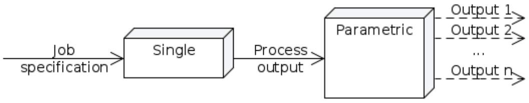
4.3. Parameter study

A parameter study is a systematic way to vary a number of parameters and have the system automatically run application for each combination of parameters. In the first step job specification is created basing on the user requirements. The inputs files are uploaded, and the parameter study job is submitted. In the next step the status loop workflow that is described in 4.1 takes care of job state monitoring, downloading results and resubmitting if needed for each of the sub-jobs of the main job. There are also components responsible for automatic infrastructure problem resolving and for saving and restoring job statuses and for guarantee multiply output consistency.


Parametric
Job specification
Prepare job
Submit
Master ID
Job ID
Status loop
Finished
Loop
for each child Job ID
Automatic infrastructure problem resolver
Workflow state save and restore mechanism
Database
Multiple output consistency guarantee

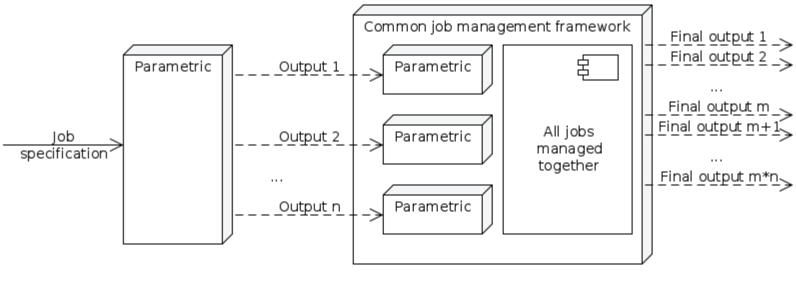
4.4. Single plus parametric

This use case is a simple combination of 4.2 and 4.3 The output of the first job is taken as an input to the parameter study job.



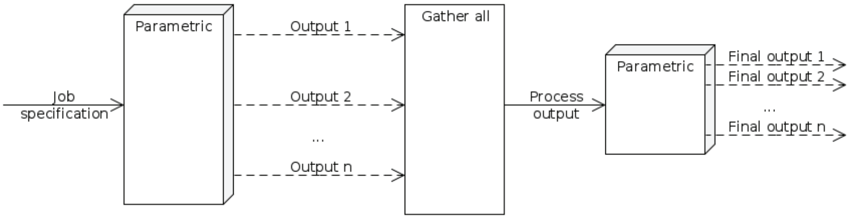
4.5. Parametric plus for each generated output, parametric

This use case is a combination of 4.3 Each of the output of the first parameter study sub-jobs? is taken as an input to the next parameter study job. Such workflows usually create a large number of sub-jobs.



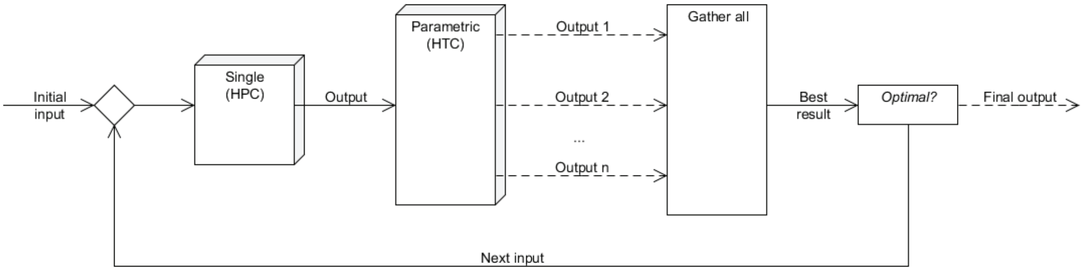
4.6. Parametric, gather all outputs, process them, submit parametric

This use case is a variation of 4.5. There is a collector that waits for the output of all subjobs of the first parameter study job. When this is accomplished only one new parameter study job is submitted.



4.7. Combination of the HPC+HTC Parametric jobs

This use case presents the simple heterogeneous workflow making use of the HPC and HTC resources at the same time. In the first step the parameter study is run on the HTC resources. Then the results are taken as an input for the HPC job. In some cases there is an output analysis in between and only the best results are processed in further steps.



5. Application scenarios

Several applications have already been deployed by means of the Serpens suite. These applications make use of the workflow template use cases described in the previous section. They are mainly coming from the Fusion community, but there are also some representing other disciplines like astrophysics or computational chemistry. The initial set of the workflows has been deployed in the EU EUFORIA project. Further workflows have been designed and implemented as part of the EU EGI_Inspire project.

5.1. Astrophysics workflow

This workflow implements the use case scenario 4.3. It controls the production of realistic simulations of the anisotropies of the Cosmic Microwave Background (CMB) radiation. These simulations will be used to test the performance of various algorithms designed to detect the presence or absence of non-Gaussian features in the simulated CMB maps, before they are applied to actual data as observed by Planck, a satellite from the European Space Agency. In order to test the algorithms we need to produce large numbers of simulations. Each one of them is made of a combination of a Gaussian and non-Gaussian component plus realistic instrumental noise that takes into account the observing strategy of the satellite. This workflow moves to the storage element the necessary information to generate the simulations, then the parameter study job is submitted and each subjob will generate in the actual simulation that will be copied back to the storage element once the job is finished.

5.2. GFIT3C+ABC

This workflow implements the use case scenario 4.4. It comes from the field of computational chemistry and the applications has been provided by COMPCHEM VO of the European Grid Initiative. The prototypical workflow presented in this paper involves the use of two applications: GFIT3C [16] and ABC [17]. The first is a routine performing the global fitting of the potential energy surfaces in triatomic systems; the latter is a quantum mechanical atom-diatom reactive scattering program. Both codes belong to the set of computational applications in which GEMS [18] is articulated. In a typical use case the ab initio points of the Potential Energy Surface (PES), provided by running ab initio electronic structure calculations or searched in web repositories, are fitted using GFIT3C. The output carried out by such global fitting routine ("fitting file") is then merged into the source code of the ABC application, and compiled to obtain the ABC executable which can be then distributed on the grid in a parameter study fashion where each input files will be processed by a separate task on a grid node.

5.3. CHEASE + MARS-F

This workflow implements the use case scenario 4.5. This workflow is a combination of the equilibrium solver CHEASE supplying input information to the MHD stability code MARS-F that calculates complex RWM eigenvalue (growth rate and real frequency). Stability and control of Resistive Wall Modes (RWM) are important issues for the fusion plasma research. Growing on a time scale of the magnetic field penetration through the resistive wall facing the plasma RWM is seen in different fusion devices. It sets the maximum pressure value in the steady-state tokamak operational scenarios and limits the discharge duration of the Reversed Field Pinch (RFP). RWM is predicted to be unstable in the ITER advanced scenario, and hence has an impact on the efficiency of future fusion reactors. This use case represents a kind of parametric + parametric execution model, where each application is executed following a parameter scan approach. Scientific results obtained using this workflow were presented at the 38th European Physical Society (EPS) conference on Plasma Physics [19].

5.4. FAFNER2-ISDEP

This workflow implements the template use case presented in 4.6. The numerical simulation of NBI (Neutral Beam Injection) fast ion dynamics is an important point in fusion devices, since these particles are used to heat and fuel the plasma [20]. In particular, the high confinement mode (Hmode) is usually driven by the NBI heating system. Finally, the fast ion uncontrolled losses can damage physically the walls of the reactor. The usefulness of the simulation of NBI ion dynamics in fusion plasmas is then clear. This workflow is devoted to show a new scenario that simulates these dynamics using two different nuclear fusion codes on the grid: FAFNER2 [21] and ISDEP [22]. FAFNER2 provides the fast ion initial state in the fusion device being simulated. With these initial states, ISDEP calculates their evolution in time.

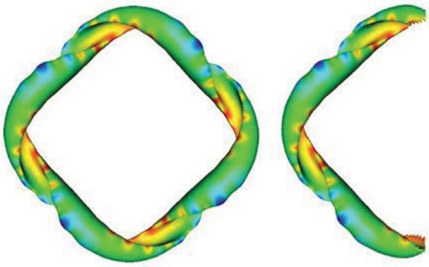
Each instance of ISDEP considers a subset of the particles initially obtained with FAFNER2. The user can define the number of instances of ISDEP that will be submitted to the grid. Based on this number, the number of particles considered by each ISDEP task is automatically calculated. After all the partial results calculated by ISDEP are retrieved, a post-processing application is used to analyze those partial results and achieve the final overall result.

5.5. VMEC-Mercier-COBRA-Visualisation

This workflow implements the use case described in section 4.7. This use case is focused on the characterisation of the transport of particles in confined plasmas. VMEC[23] calculates the MHD equilibrium of a given configuration. Once we have achieved this equilibrium, we can proceed with further characteristics of plasmas. The so called Mercier[24] modes are local instabilities characterised by being radially localised but toroidally and poloidally extended. The Mercier stability criterion is presently implemented in the VMEC code (refer to the following section). The last versions of VMEC code estimates the Mercier stability criterion for the effective radius 0 to 1 (the radius of the plasma is reduced to that format). Since the area of interest is in the range [0.8,1], only those values are taken into account. Stable Mercier configurations are those satisfying the criterion in that range. Besides Mercier stability criterion, the Ballooning mode stability is also imposed. Ballooning modes are characterised by being radially, toroidally, and poloidally localised. Ballooning modes appear in zones of unfavourable curvature. The COBRA (Code for Ballooning Rapid Analysis) code[25] performs this stability analysis.

These are serial applications which are executed in the grid. On the other hand, the generation of the 3D visualization file of the results obtained with those applications is executed on HPC resources. This is a shared-memory parallel application using OpenMP. Thus, the output of VMEC and COBRA is sent from the grid resources to HPC resources, where it will be used by the parallel application to create a visualization. Figure 3 shows the representation of this visualization file. This visualization plays an important role in the understanding of the confinement capabilities of the fusion reactor.

A single execution of this use case implies of the evaluation of a configuration of a given fusion device. But this configuration is characterized by a large number of configuration parameters. Different values for these parameters will provide different configurations which may present severe differences among each other. A parameter scan is performed by changing the values of several of these parameters. Thus, the use case performs a study over a solution space formed by all the possible stellarators which can be defined by those parameters.



(a) TJ-II stellarator configuration

(b) Cut of the confined plasma. Six magnetic surfaces are displayed.

Figure 3. Top view of the plasma confined in a fusion device.

6. Future Plans

Maintenance of the Serpens suite will require following the new Kepler releases and keeping track of compatibility with the underlying grid middleware that is under constant changes. Also since new emerging infrastructures are appearing, Serpens is planned to be extended in order to support new scenarios. In particular, the usage of cloud technologies is foreseen. In this area we have already made proof of concept examples making use of cloud using the Amazon EC2 and OpenNebula. Also, new scenarios coming from the different application domain will be introduced.

7. Summary

The proposed framework provides researchers with the easy-to-use, workflow-based environment where multiple tasks can be run in different resources (both HTC and HPC) The workflows that the researchers may execute using the proposed framework range from the simplest tasks that require the use of a single CPU for computations to the most demanding ones using as many resources as they can gain in order to complete the calculations in a reasonable amount of time. We have presented the developed software, namely the Serpens suite for Kepler, and presented generic use case scenarios. Demonstration of the exploitation of generic use cases with an example of the scenarios coming from the Nuclear Fusion, Astrophysics and Computational Chemistry has been described in detail. This scenarios show that the proposed solution is generic enough to be used by different applications from various scientific domains. We have also presented future work direction with respect to new applications as well as other computational infrastructures, and their capabilities.

Acknowledgements

The research leading to these results has received funding from the FP7 (FP7/2007-2013) under grant agreement no211804 (EUFORIA). Support for user communities using Kepler scenarios is provided under the EGI_InSPIRE project, under grant agreement no261323. We would like to thank Ilkay Altintas, Jianwu Wang from San Diego Supercomputer Center for help and support during the process of official releasing of the Serpens suite.

References

1. I. Altintas, et al. Kepler: an extensible system for design and execution of scientific workflows. 16th International Conference on Scientific and Statistical Database Management, pp. 423–424 (2004).
2. A. Streit, et al. UNICORE 6 - Recent and Future Advancements, JUEL-4319, February 2010, ISSN 0944-2952
3. E. Laure, et al. Programming the Grid with gLite Computational Methods in Science and Technology 12(1), pp. 33-45 (2006).
4. Kleist, Josva , et al., Roadmap for the ARC Grid Middleware..In: Ikke angivet. Vol. 4699 Springer Berlin / Heidelberg, 2006. p. 471-479 (Lecture Notes in Computer Science; 4699).
5. I. Foster, Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2006.
6. OSG Executive Board. 2007. The Open Science Grid. Journal of Physics: Conference Series 78:012057, Boston, 2007
7. Charlie Catlett "The Philosophy of TeraGrid: Building an Open, Extensible, Distributed TeraScale Facility". 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid: 8. doi:10.1109/CCGRID.2002.1017101. ISBN 0-7695-1582-7.
8. Kevin Lee, et al. Adaptive Workflow Processing and Execution in Pegasus. In Concurrency and Computation: Practice and Experience,. Volume 21, issue 16, 2009, pages 1965-1981.
9. D. Hull, et. al.Taverna: a tool for building and running workflows of services. Nucleic Acids Research, 34, iss. Web Server issue, pp. 729-732 (2006).
10. I. Taylor, et al. The Triana Workflow Environment: Architecture and Applications Workflows for e-Science, Springer 07, pp. 320-339.
11. Z. Farkas, P. Kacsuk P-GRADE Portal: A generic workflow system to support user communities Future Generation Computer Systems 27(5), pp. 454-465 (2011).
12. B. Bethwaite, et al *Mixing Grids and Clouds: High-Throughput Science Using the Nimrod Tool Family*, Computer Communications and Networks, 2010, Volume 0, Part 2, 219-237
13. I. Altintas, “Collaborative provenance for workflow-driven science and engineering”, 2011.
14. Luis Cabellos, et al. Scientific workflow orchestration interoperating HTC and HPC resources Computer Physics Communications, 182 (4), pp. 890–897(2011).
15. Chin G, et al. Scientist-Centered Workflow Abstractions via Generic Actors, Workflow Templates and Context-Awareness for Groundwater Modelling and Analysis, IEEE 2011 Fifth International Workshop on Scientific Workflows, Washington D.C., July, 2011
16. A. Aguado, et al., “Global fit of ab initio potential energy surfaces I. Triatomic systems", Computer Physics Communications, vol. 108, no. 2-3, pp. 259-266, 1998.
17. D. Skouteris,et al. , “Abc: a quantum reactive scattering program", Comp. Phys. Comm., vol. 133, pp. 128-135, 2000.
18. A. Costantini, et al, “COMPCHEM:progress towards GEMS a Grid Empowered Molecular Simulator and beyond”, Journal of Grid Computing, vol. 8(4), pp. 571-586, 2010
19. D. Yadykin et al, Studies of Resistive Wall Mode stability in the multi-parametric space using Grid infrastructure, 38th EPS Conference on Plasma Physics, Strasbourg, France, P5.090
20. T. H. Stix, “Heating of toroidal plasmas by neutral injection,” Plasma Physics, vol. 14, pp. 367–384, 1972.
21. G. Lister, FAFNER: A Fully 3-D Neutral Beam Injection Code Using Monte Carlo Methods, 1985.
22. A. Bustos, F. Castejon, M. Osakabe, L. Fernandez, V. Martin-Mayor, J. Guasp, and J. Fontdecaba, “Kinetic simulations of fast ions in stellarators,” Nuclear Fusion, vol. 51, no. 8, p. 083040, 2011. [Online]. Available: http://stacks.iop.org/0029-5515/51/i=8/a=083040
23. S.P. Hirshman and G.H. Neilson External inductance of an axisymmetric plasma. Physics of Fluids 29(3), pp. 790–793 (1986).
24. C. C. Hegna and N. Nakajima. On the stability of Mercier and ballooning modes in stellarator configurations. Physics of Plasmas, 5(1336), pp. 1336–1344 (1998).
25. R. Sanchez, S. P. Hirshman, J. C. Whitson, and A. S. Ware. COBRA: an optimized code for fast analysis of ideal ballooning stability of three-dimensional magnetic equilibria. Journal of Computational Physics, 161(2), pp. 576–588 (2000).