Grid-Workflows in Molecular Science *

Georg Birkenheuer [1], Sebastian Breuers [2], André Brinkmann [3], Dirk Blunk [4], Gregor Fels [5], Sandra Gesing [6], Sonja Herres-Pawlis [7], Oliver Kohlbacher [8], Jens Krüger [9], and Lars Packschies [10]

Paderborn Center for Parallel Computing, Universität Paderborn
[1]birke@uni-paderborn.de [3]andre.brinkmann@uni-paderborn.de

Department für Chemie, Universität zu Köln
[2]breuerss@uni-koeln.de [4]d.blunk@uni-koeln.de

Department Chemie, Universität Paderborn
[5]fels@uni-paderborn.de [9]mercutio@uni-paderborn.de

Zentrum für Bioinformatik, Eberhard-Karls-Universität Tübingen
[6]sandra.gesing@uni-tuebingen.de
[8]oliver.kohlbacher@uni-tuebingen.de

Fakultät Chemie, TU Dortmund
[7]sonja.herres-pawlis@tu-dortmund.de

Regionales Rechenzentrum, Universität zu Köln
[10]packschies@uni-koeln.de

Abstract: Computational Chemistry gathers information about properties of molecules based on compute intensive simulations. In this area, workflows are an essential instrument for managing complex simulations cascades. The aim of the MoSGrid project is an easy to use Grid integration of such workflows based on a portal that covers the complexity. This paper presents an initial general description of workflows for molecular science and details the result based on two examples for the integration of Gaussian and Gromacs.

1 Workflows and workflow management in MoSGrid

Computational Chemistry has evolved during the last two decades into an established discipline in natural sciences. Scientists from many disciplines are using molecular simulations to gather information about the properties of molecules, small and large alike. Computational procedures developed by theoretical chemists and physicists as well as mathematicians and bioinformaticians are now used on a daily bases as high performance computing infrastructures have become available to everyone. However, as computational chemistry allows users to answer increasingly complex scientific questions, more and more

*This work is supported by German Ministry of Education and Research under project grant #01IG09006.

scientists are encouraged to apply these tools, and consequently the demand for computational resources strongly increases.

There are, however, a number of limitations. The number of programs for molecular simulations (electronic structure methods and molecular mechanics/dynamics codes) that are well-established and accepted in the scientific community is still quite small. Furthermore, the usability of these tools is often limited. This is partly due to the design of the user interfaces, since a lot of sophisticated tools lack a graphical and easy to use interface. Altogether, many new users to this field have to be acquainted not only with the large number of methods and chemical theories that are implemented in these programs but also in operating the latter. To lower the hurdle of using these programs, there is the need for intuitive user interfaces.

MoSGrid anticipates to ease the handling of the large complexity of molecular simulation methods. The main goal of this BMBF funded joint research project of the German Grid (D-Grid) Initiative is to make molecular simulation codes available for German Grid resources. Molecular simulation codes and computational resources are intended to be accessible using the MoSGrid portal, which will offer various tools to import molecular information, to setup and submit calculations, and to extract relevant results easily.

The MoSGrid portal is going to offer a (graphical) workflow manager. Commonly used simple and complex workflows can be stored in recipe repositories and be made available for every user. All users can develop, improve, publish and use workflows for their everyday tasks. As a result of these efforts the variety of application cases increases, making the use of computational chemistry tools easier for less experienced users at the same time.

In this paper, we describe the nature of typical workflows and their management in the field of computational chemistry. The requirement analysis is based on a comprehensive survey within the computational chemistry community. Some preliminary results have already been presented in [NBB+09].

2 Computational Requirements in Molecular Structure Simulations

Molecular structure simulations can be demanding in various respects on several different levels. One level is the problem definition that depends on the chemist. In this process the experience and knowledge of the user about the particular simulation technique and its advantages and drawbacks is a critical element.

Another level is the resource requirements definition. This is more or less a computer science issue that depends on the size of the problem, on the algorithms applied and partly even on the hardware architecture to be used. This leads to a further level.

In *molecular mechanics* the problem size is determined by the number of atoms in the simulation and can be increased in a parallel tempering sampling approach [ED05] by the factor of the number of replicas.

A bulk simulation consisting of a vast amount of molecules and atoms can be reduced

to several smaller problems with interdependences, e. g. by domain decomposition or separated electrostatic field calculations [HKvdSL08].

Replica exchange or parallel tempering simulates different replicas with defined order parameters like temperature or pressure, eventually exchanges them and subsequently continues the simulation on the exchanged replica. Dependent on the number of atoms many different replicas can be necessary to achieve a good sampling. Both approaches require at least several processors and a – shared or distributed – common memory. This can be reasonably achieved by a MPI (Message Passing Interface) implementation that is realised in the Gromacs molecular dynamics package [HKvdSL08].

In *quantum mechanics* the system size is dependent on the number of basis functions and the calculation complexity on the method employed. The solution of such systems can be enhanced by, e. g. the parallel solving of the self consistent field equations.

A screening setup that employs a quantum mechanical approach to calculate certain properties of several hundred up to millions of molecules where each of them in their own simulation can increase the problem size. This screening consists per se of several tasks without internal dependences which means that every task can be solved on a different ressource. Depending on the problem, the calculation can be enhanced and accelerated by multiple processors [FTSea09].

3 General Workflow Description

One of MoSGrid's aims is to simplify the access to molecular simulation tools. This will be accomplished by providing a collection of recipes to the user of the MoSGrid portal. Chemists are familiar with recipes from their daily work and in this context working with a recipe defines an ontology.

A recipe is a concept that can be implemented using a workflow. The workflow that will be applied to the simulation codes in the MoSGrid project can be depicted as a multi-step process consisting of the following tasks:

Job definition that describes the problem the user wants to solve. For this description the user has to provide information with intelligent assistance of the portal. Every chemical question concerns a structure of interest and one or more of its properties. The user starts a workflow by uploading a structure (which is possible in several different formats) or by usage of a data repository that is also part of the MoSGrid developmental efforts. This repository will contain descriptions of molecular structures and results of structure calculations in a meta molecule description language. The input of this information is supported by a continuous

Metaprocessing that checks the user input for consistency. This processing will assist the user to provide a complete set of meta information describing the job, e. g. structure, application, method, temperature or basis set. To avoid annoyance of the user by (over-)demanding input masks sensible default values will be provided by the system as preset values which can be overwritten by the user.

Afterwards this molecular simulation meta description (MSMD) will be translated to the user in a totally transparent manner by a

Preprocessing step resulting in an application specific input format. For this process adapters will be developed for several quantum chemistry and molecular dynamic codes, e. g. Gaussian and Gromacs. As far as possible the estimation of resource requirements as memory, storage or number of nodes and processors shall also be done by these adapters. Concerning partial platform specificity of input files a differentiation between a portal based and a resource based preprocessing has to be done.

After conversion of the MSMD and generation of the input files, the

Job Submission is initiated. This can be understood as queuing a middleware specific container which is an execution environment on a grid resource. All the described information is than transferred to this middleware environment, i. e. the staging in of the generated input files and additional input, like structure format files, as well as the translation of the system requirements into a batch system call.

Steering is a feature that in a first implementation will rely on a simple monitoring and abortion mechanism that allows the user to interfere and stop the simulation, if it develops not as expected. For this purpose the user is granted access to the result files of the ongoing simulation and is assisted by tools that generate graphical representations of the current simulation state. The steering option is intended to avoid waste of time and resources. After successful execution of the job a

Postprocessing will be performed to extract application independent information from the result files. This information will be reported in an adequate form, e. g. as charts or graphs, on the portal and can be optionally stored in the data repository provided by MoSGrid.

The huge benefit of this approach is the fact that the process of designing a molecular simulation is separated from the application. Thus, the user can concentrate on the definition of the scientific part of the simulation while MoSGrid cares about the tedious steps of preprocessing, job submission, and postprocessing in a totally transparent way.

4 Examples for Grid-Workflows in Molecular Science

The following sections describe two selected applications of workflows in molecular science. We focus on the analysis of a typical simulation process in quantum chemistry (Gaussian) and molecular dynamics (Gromacs).

4.1 Gaussian

Gaussian is a quantum mechanics package that has been widely used for over three decades by a large scientific community [FTSea09]. The first step in a Gaussian workflow (see Fig. 1) is the definition of a molecular structure. The molecule is typically created by a graphical molecule editor like GaussView. At a later stage of the MoSGrid project, the basic molecule-definitions will be available from a molecular database and the ability of editing the molecule will be added to the portal itself.

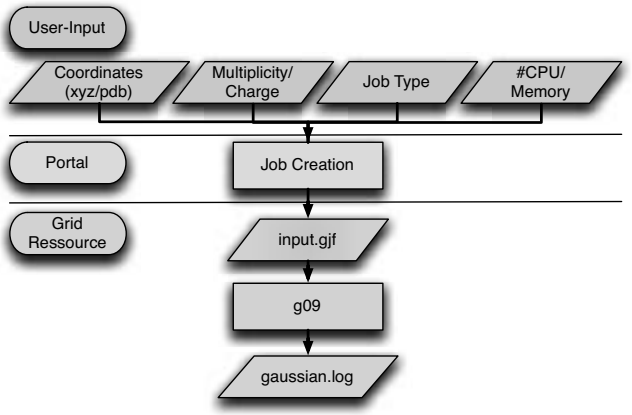

Figure 1: Basic Gaussian workflow showing default input and output.

Furthermore, technical information like the memory and the number of CPU's have to be inserted. This could be added automatically by the preprocessor upon resource selection. Following our recent analysis, all required information can be collected in a Grid portal. Even the preprocessor is likely to be provided by the portal itself, as these steps are not compute-intensive and a direct user input is possible. After molecule creation, job definition, and preprocessor check, a Gaussian job file is submitted into the Grid, and computed with Gaussian *g09*. After the termination of the job, the user analyses the results and mostly submits subsequent calculations based on this first calculation. In a later stage of MoSGrid these subsequent calculations can be added to the workflow as well.

This workflow can be extended by using a huge array of molecules as input which will be processed in parallel. For this purpose, the portal will offer the possibility to select an array of molecule coordinates, convert them into the corresponding gjf-files with predefined information and to subsequently submit the bulk to the Grid.

4.2 Gromacs

Gromacs 4 is a software package for molecular dynamics (MD) simulations. It is considered one of the fastest MD codes available [HKvdSL08]. Examples of MD applications, covering only a small portion of possible applications, are the study of drug-receptor interaction, shear viscosities of technical fluids, folding of pharmaceutically interesting proteins, or ion permeation through ion channels [KF09]. Gromacs is an open-source project originated from the University of Groningen, now maintained by an active community of users and developers from all over the world [HKvdSL08].

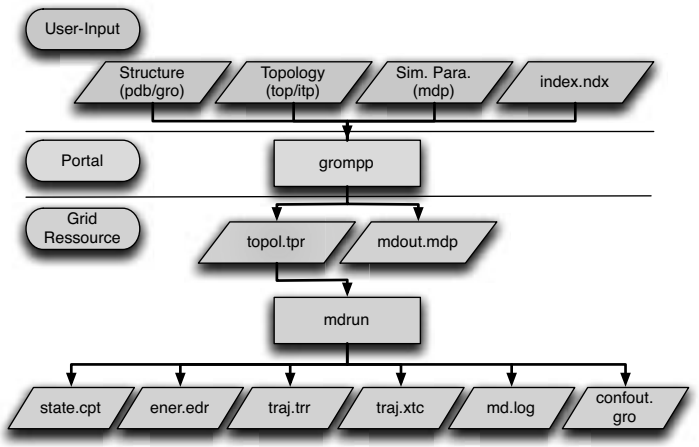

Figure 2: Basic Gromacs workflow showing default input and output.

The most simple Gromacs workflow (2) consists of the call of the preprocessor *grompp*, which joins input data such as the atom coordinates and system topology with the calculation specification into a single input file (*topol.tpr*). The actual calculation is performed by the (parallel) MD application (*mdrun*) taking great advantage of high performance compute resources with low latency network interconnects. Real life examples for anticipated Gromacs workflows for the MoSGrid projects are (i) full equilibrations of globular proteins in explicit water, (ii) semi-automatic free energy perturbation simulations to determine the free energy of solvation of drugs and other small molecules, (iii) semi-automatic linear interaction-energy based calculations of the binding energy of drugs to their receptors, (iv) semi-automatic property studies of bulk or water solute liquid crystal materials.

5 Selected Workflow Engines

In accordance to MoSGrids workflow requirements, several workflow engines could be chosen for implementation. The following will give a brief introduction to some potential workflow engines for MoSGrid. However, which engine to take is not decided yet.

UNICORE 6 The UNICORE 6 embedded workflow engine follows a data driven workflow concept. The different steps of a workflow are described in an XML format. At execution the workflow service decides which step can be executed and forwards this step to a service orchestrator that assigns the subjob to a compute resource. When a subjob finishes, the workflow manager checks if a following subjob can be started due to predefined and now fulfilled dependencies.

BPEL BPEL defines atomic operations as WSDL described web services. This allows the usage of an external web service inside a workflow instead of executing it as a subjob on the resources. The BIS-Grid project incorporated the ActiveBPEL workflow engine into UNICORE 6. This added functionality allows to execute a frequently used parser as a web service either in the portal or as a subjob in the UNICORE workflow.

jBPM jBPM from JBoss is a Java-based framework for orchestrating and invoking workflows by using the proprietary process language jPDL (jBPM Process Definition Language) and the workflow engine PVM (Process Virtual Machine). PVM supports multiple process languages and offers to combine computing tasks with human interactions. It is easy to extend with Java expertise and has an active community. One of its drawbacks is that it also requires Java expertise for a basic action like orchestrating workflows, e.g. to integrate web services.

MSS MSS is designed for co-allocation of computing, software licenses and networking resources. Important is the reservation of network bandwidth for workflows that require QoS during stage-in, compute, and stage-out. MSS uses WS-Agreement as interface to communicate with external components like the middleware or the user client.

WSS The Workflow Scheduling Service (WSS) is a workflow management and scheduling component. WSS deals with workflow management and execution including data extraction, transport, job execution and fully integrates into the provided Globus WSRF framework. Further, WSS enables users to describe workflows with a simple XML-based Workflow Specification Language (WSL) that encapsulates JSDL.

GWES The Grid Workflow Execution Service (GWES) is a workflow enactment engine with Meta scheduling capabilities. GWES coordinates the composition and execution process of workflows. It implements a workflow concept by means of the Grid Workflow Description Language (GWorkflowDL), which is based upon the theory of Petri nets. It provides interfaces to the Web Portal, to a command line client, and to Globus Toolkit 4.

GridWay The GridWay meta-scheduler describes interdependencies between tasks of a workflow in a "Job Description Document" (JDD) or JSDL. GridWay is part of the Globus Toolkit and its Job Management assigns every job, to a host for execution due to job dependencies, resource requirements and resource limits. A Dispatch Manager

performs the submission and watches over the execution of a job, the Execution Manager is responsible for job execution and management, and a Transfer Manager is responsible for file staging, handling the remote working directory and remote host clean-up.

All workflow engines support the basic workflow functionalities required by MoSGrid. However, during the definition of MoSGrid, it was decided to use UNICORE 6 as Grid middleware as it is widely distributed and already offers graphical user interfaces for an intuitive access to Grid resources even for non computer scientists. Therefore, the further analysis of workflow engines will be restricted to those usable with UNICORE 6 and compare them to MoSGrid's requirements. In focus of this process will be the UNICORE 6 embedded workflow engine and BIS-Grid's ActiveBPEL workflow engine.

6 Conclusion and Future Work

MoSGrid is at an early stage of the project. Therefore, this paper describes the work done for the conceptual design of typical workflows. Based on the workflow concept an evaluation of workflow engines will follow. MoSGrid will use UNICORE 6 as Grid middleware restricting the analysis of workflow engines. We will focus on the UNICORE 6 embedded workflow engine and the BPEL integrated workflow engine originating from the BIS-Grid project. A detailed evaluation of the engines will affect the deployment of the MoSGrid portal.

References

[ED05] David J. Earl and Michael W. Deem. Parallel Tempering: Theory, Applications, and New Perspectives, 2005.

[FTSea09] M. J. Frisch, G. W. Trucks, H. B. Schlegel, and et. al. Gaussian 09 Revision A.1, 2009. Gaussian Inc. Wallingford CT.

[HKvdSL08] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *Journal of Chemical Theory and Computation*, 2008.

[KF09] J. Krüger and G. Fels. Potential of Mean Force of Ion Permeation through alpha7 nAChR Ion Channel. In *Proceedings of IWPLS'09, Edinburgh, UK, September 14-15, 2009, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-513/paper06.pdf*, 2009.

[NBB+09] O. Niehörster, G. Birkenheuer, A. Brinkmann, B. Elsässer, D. Blunk, S. Herres-Pawlis, J. Krüger, J.Niehörster, L.Packschies, and G. Fels. Providing Scientific Software as a Service in Consideration of Service Level Agreements. In *Proceedings of the Cracow Grid Workshop (CGW)*, 2009.