

# Grid Workflow Approach using the CELLmicrocosmos 2.2 MembraneEditor and UNICORE to commit and monitor GROMACS Jobs

Sebastian Rubert<sup>1</sup>, Christian Gamroth<sup>1</sup>, Jens Krüger<sup>2</sup> and Björn Sommer<sup>1,\*</sup>

<sup>1</sup> Bio-/Medical Informatics Department, Bielefeld University, Universitätsstraße. 25, D-33615 Bielefeld

<sup>2</sup> Organic Chemistry, University of Paderborn, D-33098 Paderborn

---

## ABSTRACT

**Motivation:** Molecular dynamic simulations of membrane systems are an important method for the prediction and analysis of physicochemical properties. The CELLmicrocosmos 2.2 MembraneEditor (CmME) provides a comfortable workflow to generate lipid membranes with different conformations. While CmME is intended to generate molecular structures on desktop and mobile computers in a very short time, the atomic simulation of exported membranes needs external high performance computer resources. In this work, a first approach of a direct connection between CmME and a cluster running GROMACS using the Grid-middleware UNICORE-6 is discussed.

## 1 INTRODUCTION

The generation and simulation of membranes is a very important topic of chemical and physical biology. Model membranes are used to analyze physicochemical properties at a level still beyond the reach of present experimental and microscopic techniques.

The CELLmicrocosmos 2.2 MembraneEditor (CmME) (Sommer et al. 2011) provides a modular interactive shape-based software approach to model membranes based on the Protein Databank format (PDB) (Berman et al. 2000). Different PDB lipid types can be imported into CmME and used to generate membranes featuring different percentages of lipid distributions. The computation of an average-size membrane system (~500x500 Å<sup>2</sup>) on an actual desktop or mobile computer takes a few seconds up to several minutes, because the lipid placement algorithm does not change the atomic structure of single (rigid) molecules.

To simulate the atomic structure of the exported PDB membranes, molecular dynamics programs like GROMACS (Hess et al. 2008) are deployed. This computationally very space- and time-consuming task needs adequate computer resources. Many research groups have the opportunity to use internal or external cluster facilities for molecular simulations. But often this possibility does not exist for multiple reasons.

This fact is one reason why the public and scientific attention for Grid computing is raising rapidly. The advantages of Grid computing lies in the possibility to use a Grid-middleware, which enables the user to use the power of large computer resources without the need to know and configure the underlying hardware and software. Especially companies like Google and Amazon raised attention for this field by giving external users the

possibility to temporarily access idle computer resources. And like many other scientific associations, this technology came to the attention of the membrane and protein modeling community as well (Birkenheuer et al. 2010).

In this work we present a first approach to connect CmME to GROMACS-running cluster resources using the Grid-middleware UNICORE (Schuller and Schumacher 2009), providing a fast and easy approach to model and simulate membranes at the molecular and atomic level. Computational resources were provided by the Paderborn Center of Parallel Computing (PC<sup>2</sup>), which is involved in the MoSGrid initiative (Wewior et al. 2010). MoSGrid is part of the D-Grid initiative, which is promoted by the ministry of education and science.

## 2 METHODS

### 2.1 Technical Details

For the development of the GROMACS plug-in, CmME 2.2 and GROMACS 4.5 are used. Java 1.6 and Java3D 1.5 are the current programming APIs included in the CmME. The computation takes place at the local grid resource BisGrid at the PC<sup>2</sup> in Paderborn. To gain access, a BisGrid account and a D-Grid certificate are needed. This grid resource can be accessed using UNICORE at version 6.0 and presently has a local installation of GROMACS version 4.5. It consists of 64 cores, 512 gigabyte main memory and 10 terabyte harddisk.

### 2.2 UNICORE

The UNICORE middleware is a comfortable option to gain direct access to grid resources (Gesing et al. 2010). To connect to a server resource a key-store is used to provide the security needed. A job can be started by providing a job file containing information about the application, imports, exports and some server settings like the number of cores to be used for this calculation. This procedure can be accomplished through a rich client, a command line client or the high level API directly from a Java program. To connect to the grid server a property file containing information about a keystore and the computation resource is needed. Every single command has to be included in a job file, which needs to be uploaded. The application output stdout, stderr and created files can be download after completion of the job.

### 2.3 CELLmicrocosmos MembraneEditor

CmME currently implements four algorithms for the arrangement of lipids. After defining the percental lipid distribution, an algorithm can be started to create the initial membrane according to the given percentages. In addition, CmME has an algorithm plug-in interface, which was originally integrated to create external user-specific algorithms. The Gromacs plug-in discussed later in this work, is based on this plug-in interface. The advantage of this approach is that the plug-in can be excluded from the Standard Edition of CmME, because the regular user does not need GROMACS support. Another feature of CmME is that the exported PDB format is adjustable to the requirements of different programs. This feature is used for exporting the membrane to GROMACS.

## 2.4 GROMACS

GROMACS is an open source molecular dynamics software, which is very popular in the membrane and protein modeling community (Krüger and Fels 2010). A standard GROMACS workflow consists of eight steps. At first the `editconf` application is called to create a box around the virtual membrane. After that, `genbox` is used to fill this box with water (needed for the molecular movement of the atoms). Lastly `grompp` creates the input file for the energy minimization step, which is computed in `mdrun`. These two steps are repeated three times for the minimization, equilibration and finally for the molecular simulation of the membrane.

## 3 RESULTS

### 3.1 Plug-in Workflow

To control the GROMACS workflow (see Fig. 1) discussed above with CmME, a plug-in has been developed. Before adjusting the parameters of the application the membrane has to be created. Afterwards the GROMACS plug-in is used to set up the parameters of the needed commands and put them into the job list (see Fig. 2). Each job from the job list can be launched as often as required; the sequence and the configuration can be changed again prior to that.

The CmME GROMACS plug-in uses the high level API from UNICORE to send the computed membrane to the grid server. Calculation of membranes using GROMACS can be completed in a short time using the strength of a grid system compared to a local machine. UNICORE connections can be established when the job list is generated by providing a properties file. The needed files for the simulation can be uploaded via the graphical user interface of the GROMACS plug-in. Furthermore, the configuration of each GROMACS application can be saved in an XML file and loaded again if a similar simulation needs to be done. At the time, a job is configured and finally started; a job file is created and committed via UNICORE to the server. While the GROMACS application runs on the grid resource it is possible to check the status of the job using the CmME GROMACS plug-in. When the job is finished, the standard output and the standard error channel from the invoked GROMACS tools are presented. If desired, the produced file is downloaded to the local machine. This procedure is repeated for every single job until the molecularly simulated membrane has

been generated with the final MD run and loaded back into the CmME.

### 3.2 Plug-in Design

The Plugin is developed around a GUI `Toolbar`, which is inherited from the `MembraneAlgorithm-Interface`. The GUI `Toolbar` offers the ability to open different windows to configure the chosen algorithm (in this case the GROMACS plug-in). Furthermore, the assembling of the settings window is adjusted by the `GromacsData Class`, where all GROMACS application values are stored. By changing values in the settings window the data is modified through a controller class (`GromacsController`, see Fig. 1). In addition, the job list is saved in the data class and can be transferred to the calculation window at the time the corresponding button is clicked. The calculation window is assigned to start and stop the configured jobs. Changes in the job list are monitored and executed by an additional controller class, the `JLDController`. On submitting a job, a `ProcessRunner` is created and a GROMACS process is started. To show the output of the started process the `StreamGobbler` listens to the given input stream.

To gain access to a network resource like the BisGrid cluster, an additional `UNICOREConnector` class is implemented (see Figure 2). This thread is started in the settings window and can be invoked by the calculation window. To simplify the usage of UNICORE connections, the GUI classes merely need to use this connector class to run jobs on the grid server. All necessary UNICORE operations are implemented.

### 3.3 The GUI

The GUI consists of two main parts. In the settings window, the user can set up parameters of the different applications, which are shown in different tabs (see Fig. 3), connect to a UNICORE resource, load and save application configuration to/from XML file and create a job list. The connection state is indicated by a text line, which is lighted in green if a connection is established successfully. To give the user additional feedback about the file operation, every upload is illustrated by a progress bar and a dialog window to show an error or the success of the process. Each Application can be configured by checkboxes, spinnerfields or by just providing a file-choose dialog. A resulting command is generated automatically in a text field for each application. Furthermore each application-tab could be saved to and loaded from XML files. The job list (see Fig. 2) is sent to the calculation window which has the ability to reconfigure some jobs, start and stop them. To give the user some feedback, the standard output and error channels are displayed.

This kind of invoking a GROMACS job is very comfortable and easy to use. No local GROMACS installation is needed and a created workflow can be loaded into the GUI via XML.

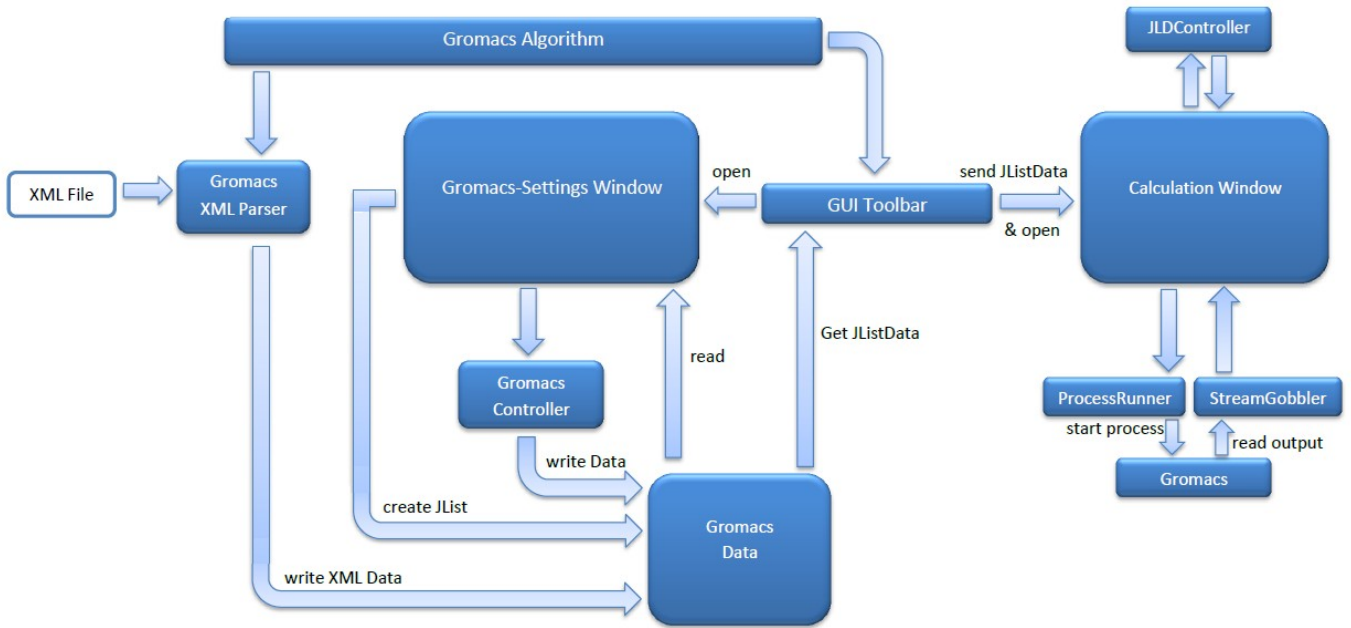
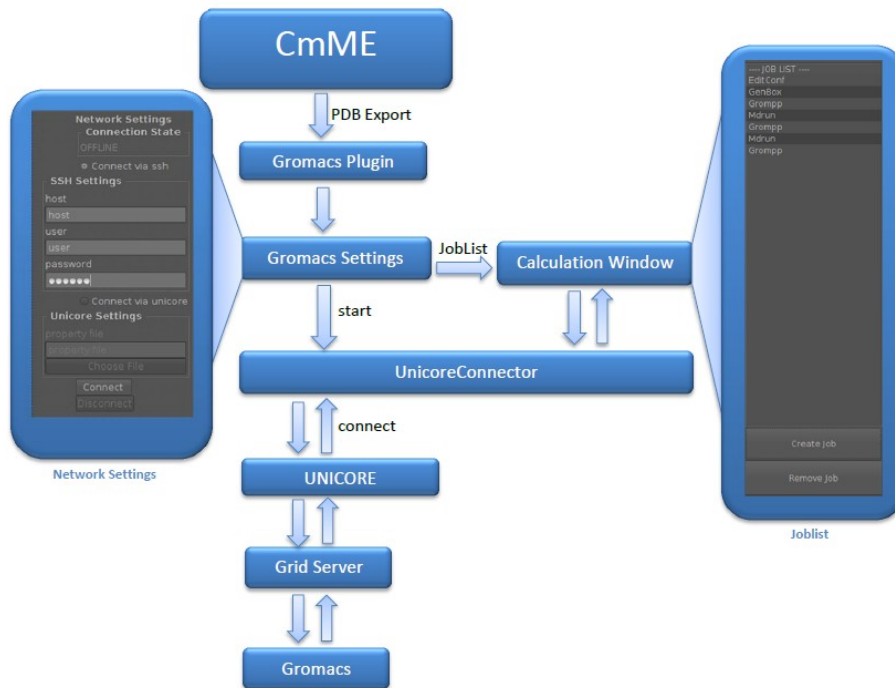


Fig. 1. CmME GROMACS Plug-in: Workflow Graphic

Fig. 2. CmME GROMACS Plug-in: Communication via UNICORE



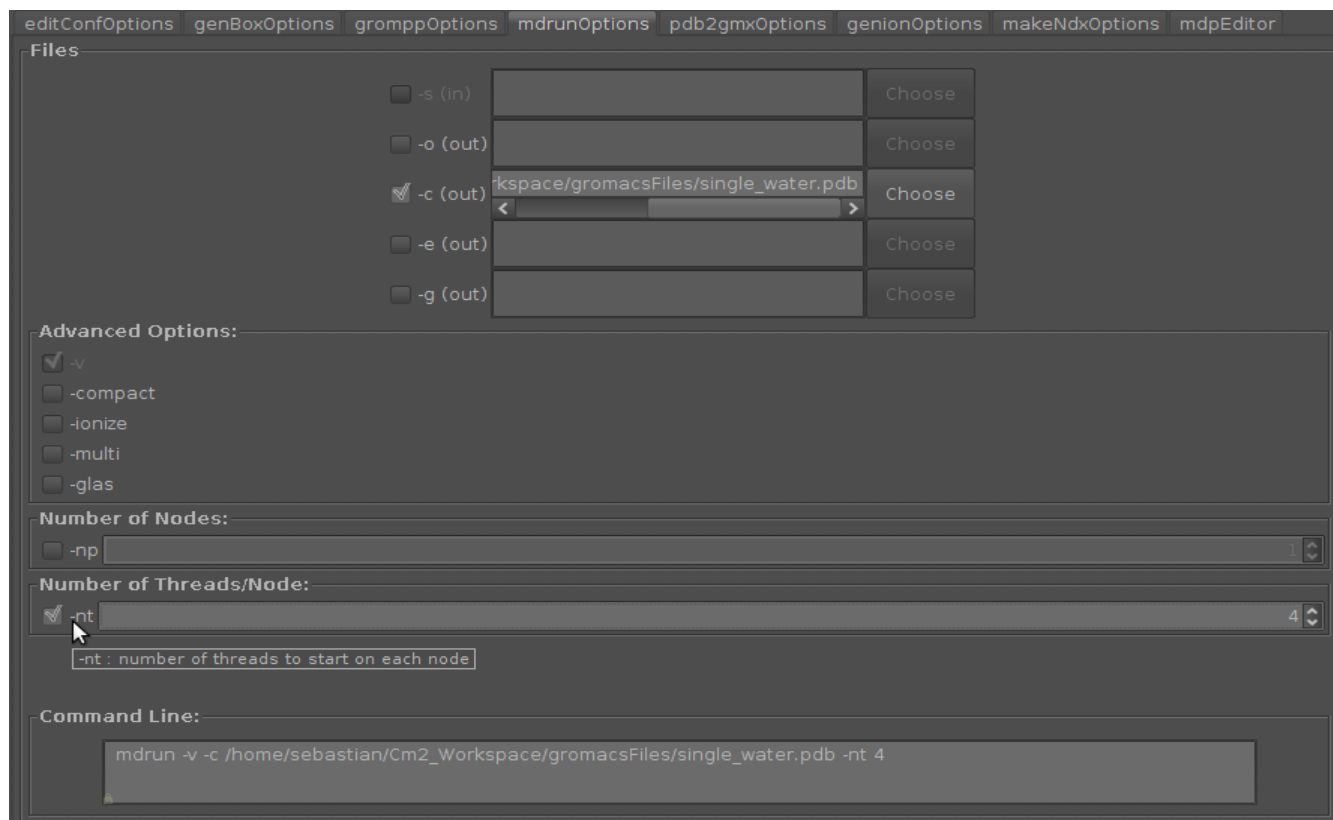


Fig. 3. CmME GROMACS Plug-in: The GUI of `mdrun`

### 3.4 Potential Problems

A Problem in the GROMACS workflow is the potential requirement for changes in the configuration files by the user during the job execution. Topology files, which are required by the `genbox` command, can be created based on information given by CmME. But the balancing of charge in the membrane needs to be done manually or in the calculation window.

### 3.5 Performance

The Performance of the whole calculation depends on the used grid resource and the size of the given membrane. Using a local GROMACS installation on a single machine can take weeks to complete, even the simulation for only one lipid may take half an hour on a local machine. Because of this, a grid resource like BisGrid is indispensable to get a result within a reasonable time. The calculation itself, and the connection as well as some file uploading is done in different threads to keep the main GUI running.

## 4 CONCLUSION

Here we present a fast and comfortable way to generate and simulate membranes at the molecular and atomic level by combining CmME and GROMACS using UNICORE. To make the configuration even more simple an additional GUI mode is planned. The creation of the job list should be done by one click and the user will be guided directly to the calculation window to start the GROMACS workflow. This approach is a first part of the

ongoing CmME MD Edition initiative, which develops a CmME plug-in

specially adjusted to the needs of molecular dynamic simulations using GROMACS. In addition, a direct and a SSH connection to GROMACS is in preparation.

The great advantage of the GROMACS plug-in is the link to CmME. In contradiction to other projects like GUIMacs (Kota 2007), the membrane, which has to be simulated, can directly be parsed into the GROMACS tool. In addition, the connection to grid resources makes the plug-in a powerful appliance for molecular membrane simulation.

The Webstart version of CmME is available free-of-charge from <http://Cm2.CELLmicrocosmos.org>

## ACKNOWLEDGEMENTS

We thank the Paderborn Center for Parallel Computing (PC<sup>2</sup>) for providing access to BisGrid. Special thanks go to Georg Birkenheuer and Johannes Schuster for valuable discussion about grid middleware. Our thanks go to Kai Löwenthal, whose diploma thesis (Löwenthal 2005) provided a good basis for discussion, and his thesis supervisor Dr. Dieter Lorenz.

*Funding:* This work was partially funded within the Graduate College Bioinformatics (GK635) of the DFG (German Research Foundation). We are grateful to Prof. Ralf Hofestädt for continuous

---

support, to the Bio-/Medical Informatics Group of Bielefeld University, where this work has been realized, and all other people supporting or participating in this project: <http://team.CELLMicrocosmos.org>

## REFERENCES

- Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.N.; Weissig, H.; Shindyalov, I.N.; Bourne, P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**(1):235-242.
- Birkenheuer, G.; Breuers, S.; Brinkmann, A.; Blunk, D.; Gesing, S.; Herres-Pawlis, S.; Krüger, J.; Packschies, L.; Fels, G. (2010) Grid-Workflows in Molecular Science Software Engineering 2010. *GI-Edition - Lecture Notes in Informatics (LNI)*, **P-160**:177-184.
- Gesing, S.; Marton, I.; Birkenheuer, G.; Schuller, B.; Grunzke, R.; Krüger, J.; Breuers, S.; Blunk, D.; Fels, G.; Packschies, L.; Brinkmann, A.; Kohlbacher, O.; Kozłowski, M. (in print 2010) Workflow Interoperability in a Grid Portal for Molecular Simulations. *IWSG2010 (International Workshop on Science Gateways)*, Catania, Italy.
- Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. (2008) GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput.*, **4**(3):435-447.
- Kota, P. (2007) GUIIMACS - a Java based front end for GROMACS. *In Silico Biol.*, **7**(1):95-99.
- Krüger, J.; Fels, G. (2010) Ion Permeation Simulations by Gromacs - An Example of High Performance Molecular Dynamics. In *Concurrency and Computation: Practice and Experience*; Fox, G. C.; Moreau, L., Eds., ISSN: 1532-0634.
- Löwenthal, K. (2005) Molekulardynamische Berechnungen auf Basis von GROMACS in einem hybriden Netzwerk. Diploma thesis, Bielefeld University.
- Schuller, B.; Schumacher, M. (2009) Space-Based Approach to High-Throughput Computations in UNICORE 6 Grids. *Lect. Notes Comput. Sc.*, 5415, 75-83.
- Sommer, B.; Dingersen, T.; Gamroth, C.; Schneider, S. E.; Rubert, S.; Krüger, J.; Dietz, K.-J. (forthcoming 2011) )CELLmicrocosmos 2.2: A modular interactive shape-based software approach to model scalable PDB membranes with definable lipid compositions and semi-automatic protein placement. *J. Chem. Inf. Model.*
- Wewior, M.; Packschies, L.; Blunk, D.; Wickerth, D.; Warzecha, K.; Herres-Pawlis, S.; Gesing, S.; Breuers, S.; Krüger, J.; Birkenheuer, G.; Lang, U. (in print 2010) The MoSGrid Gaussian portlet - Technologies for Implementation of Portlets for Molecular Simulations. *IWSG2010 (International Workshop on Science Gateways)*, Catania, Italy.