eChallenges e-2012 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2012
ISBN: 978-1-905824-35-9

UIMA-HPC – Application Support and Speed-up of Data Extraction Workflows through UNICORE

Sandra BERGMANN[1], Mathilde ROMBERG[1], Alexander KLENNER[2], Christian JANßEN[3], Thorsten BATHELT[4], Guy LONSDALE[4]
[1]*Forschungszentrum Jülich GmbH, Wilhelm-Johnen-Straße, Jülich, 52428, Germany*
Tel: +49 2461 61-[6753, 6656], Email: [s.bergmann, m.romberg]@fz-juelich.de
[2]*Fraunhofer-Institute for Algorithms and Scientific Computing, Schloss Birlinghoven, Sankt Augustin, 53754, Germany, Tel: +49 2241 14 [2736, 2276], Email: [alexander.garvin.klenner,marc.zimmermann]@scai.fraunhofer.de*
[3]*Taros Chemicals GmbH & Co. KG, Emil-Figge-Str. 76a, 44227, Dortmund*
Tel: +49 231 974272 11, Email: cjanssen@taros.de
[4]*scapos AG, Schloss Birlinghoven, Sankt Augustin, 53754, Germany*
Tel: +49 2241 14-2820, Email: [thorsten.bathelt, guy.lonsdale]@scapos.com

Abstract: The development of new chemicals or pharmaceuticals is critically dependent on a prior in-depth analysis of the published patents in this field. This is a cost- and time-consuming step when done by a human reader. One specific goal of the research project UIMA-HPC is to automate and hence speed-up the process of data extraction in patents. Multi-threaded analysis engines, developed following UIMA (Unstructured Information Management Architecture) standards, can process texts and images in thousands of documents concurrently. UNICORE (UNiform Interface to COmputing REsources) offers application support for the information extraction process via modular GridBeans, which provide graphical user interfaces for the comfortable creation of job descriptions. UNICORE workflow control structures make it possible to dynamically allocate resources for every given task to optimise cpu-time/real-time ratios in an HPC environment.

1. Introduction

1.1 Motivation

The UIMA-HPC (Multi-modal information extraction from unstructured data on HPC systems) project deals with data extraction from publications in the field of chemistry, in particular chemical patents, to enable knowledge mining [1]. The goal of the project is to automate and speed-up the process of data extraction. This process consists of several analysis tasks for identification, recognition and extraction of document constructs such as pictures, chemical names, chemical structures, disease and biological terms, co-references, patent claims and instructions.

Several automatic methods have been developed to support information extraction in the life sciences. A number of commercial organizations (e.g., TEMIS (www.temis.com), Linguamatics (www.linguamatics.com), Notiora (http://www.notiora.com), IBM (www.ibm.com), SureChem (https://surechem.com), and InfoChem (http://infochem.de)) have developed algorithms for chemical named-entity recognition [2], [3]. Systems for identification and extraction of chemicals from text are described in [4]-[6] and examples of the extraction of chemicals from structure depictions in [7]-[9].

The project targets both patent agents and research and development (R&D) experts. According to the requirements of the user group, a use-case specific workflow will be created containing the necessary analysis tasks.

The automation of the data extraction process reduces time and cost for the end-user. In addition the end-user is given flexibility regarding the choice of analysis tasks. For example, if the end-user wants to obtain both annotated references and claims, the workflow consists of two specific analysis tasks.

1.2 Technical Introduction

A key to automate the process is that a module exists for every sub-task and that these modules have a common I/O data format. It is then possible to combine the modules in workflows for information extraction. However, most of the existing annotation programs do not yet fulfill these requirements. In addition, the programs are not always suited for multi-core or cluster systems. We have identified three prior conditions for enabling data extraction on HPC systems:

- a program exists for each task,
- programs are system-independent and are capable of using all cores of a node,
- programs have a standardized I/O data format.

To implement this, the annotation programs need to be adapted or redeveloped; for creating standardized formats the existing framework UIMA (Unstructured Information Management Architecture) can be applied [10]. UIMA allows easy and comfortable integration of annotation tools by using standardized interfaces and XML meta descriptors.

This paper focusses on the workflow construction, comprising the modules described above, for the automation of the document analysis processes and for exploiting HPC system capabilities. UNICORE (Uniform Interface to Computing Resources) is a client/server middleware providing a workflow system as well as application support and uniform access to compute and data resources [11]. UNICORE offers application support by server-side definitions and corresponding GridBeans in the UNICORE Rich Client. The UNICORE Rich Client is a graphical user client which provides the UNICORE functionality, such as job submission and monitoring.

GridBeans provide the comfortable creation of job descriptions. Each software module, an annotation engine embedded in UIMA, will be integrated in the HPC environment with a distinct GridBean. This makes it possible to easily combine several annotation programs in a predefined order into workflows. The UNICORE Rich Client provides a workflow editor for specifying the execution order of applications, represented by the respective GridBeans, and to define workflow control structures, which allow for parallel execution of jobs [12].

For dynamic parallel execution of jobs in the UIMA-HPC context, several application-specific factors must be considered: the number and size of input documents, and the application runtime corresponding to the size of the input data. A critical issue is to obtain results from such workflows as fast as possible while maintaining high quality of annotation results.

A software framework which handles the processing of huge data sets is Apache Hadoop. Apache Hadoop simplifies the creation of applications which process vast amounts of data in parallel on large clusters and it takes care of scheduling and monitoring of tasks and re-executing failed tasks. A UNICORE storage enhancement through the Hadoop distributed file system was implemented in 2009 [13]. For UIMA-HPC, the target is optimization of application runtime.

Therefore, no scheduling features of Hadoop are currently integrated. UNICORE offers a scheduling mechanism provided by the UNICORE workflow system, which can be extended by implementing new strategies to optimize the runtime of information extraction process.

2. Objectives

This paper describes the modular integration of annotation programs as applications in the UNICORE Grid environment. For each annotation program, one GridBean is developed, which provides the definition of application specific input and output parameters via graphical user interfaces. In order to be flexible in the choice of annotation tasks, the corresponding GridBeans are arranged in user-defined workflows. Two basic workflows demonstrate the usage of the current prototype, which is described in this paper.

The first workflow contains ProMiner applications for annotations of human genes, chemistry and disease terms (Figure 1, left hand side). The ProMiner annotations are based on dictionaries, thesauri and curated vocabularies derived from ontologies [14]. Important features of ProMiner are: context dependent disambiguation of biomedical termini and resolution of acronyms, specific handling of common English synonyms and recognition of spelling variants of expressions in the corresponding source dictionary.

For each ProMiner application one GridBean is used and, if needed, arranged in a workflow. This means for the first prototype three GridBeans for three ProMiner applications that will be executed.

The second prototype contains only one annotation program, the ProMiner Human GridBean (Figure 1, right hand side). This job is embedded in a workflow structure, the "For-Each loop", to demonstrate parallel job execution. Currently, the "For-Each Loop" allows for parallelization by specifying the number of files per iteration or number of bytes and number of threads. In combination with job brokering strategies we reduce the time to solution as much as possible.

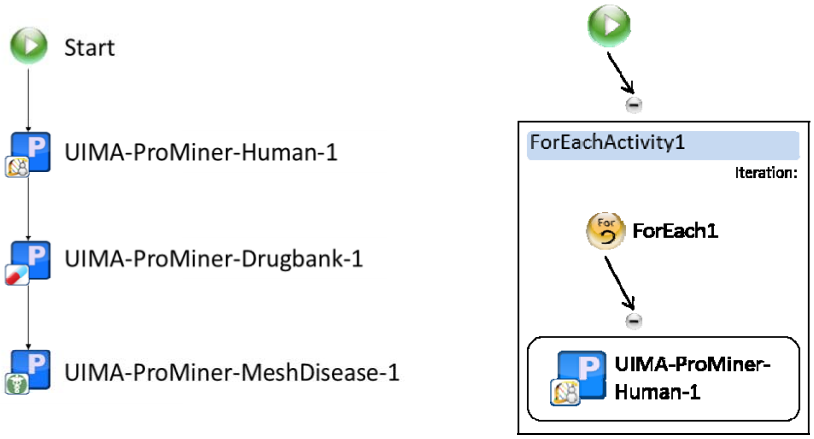

Figure 1: a) Workflow which contains ProMiner applications to be processed in sequence.
b) Workflow which contains the ProMiner Human application embedded in a workflow structure.

3. Methodology

The exploitation of the characteristics of cluster systems is a key for minimizing time to solution. As explained in the introduction, the annotation programs are embedded in UIMA that has the advantage of providing standardized interfaces, XML meta descriptors and multithreading. The input/output data format of the modules is XML. This standardized format facilitates the flexible combination of modules.

For identifying the optimal number of nodes, jobs will be parallelized using UNICORE's brokering strategies. Each module gets as input a document set which will be annotated by the module's annotation program. If this module should be executed on several nodes in parallel, the input set will be split into multiple subsets and multiple instances of the same module will be executed, each with a different input subset. The optimal number of module instances is identified using the number of files and the number of input bytes as the most important factors. Preliminary tests with prototype modules show a linear correlation between runtime and the size of input files. In order to generate consistent distributed file sets we use "Bin Packing" heuristics [15].

For executing the job on distributed target systems, job brokering is mandatory. UNICORE makes it possible to define different brokering strategies taking memory attributes, the number of available cores and nodes into consideration.

4. Technology Description

The integration of annotation tools in a Grid environment is achieved by the following concept: the first step is to adapt the annotation programs to become system independent. Thereafter, the annotation programs must be wrapped as UIMA modules to have standardized interfaces and to ensure the multithreading capability. The modules have to be installed on the computing resources and integrated as applications in UNICORE. On the UNICORE server side, an XML descriptor file defines an application by an absolute path within the file system, name, version and arguments. To link this description with the client side, GridBeans contain application specific graphical components. Each job description for a specific module is defined by a specific GridBean. This concept provides modularity and flexibility for the process of workflow creation.

For each use case we can select several GridBeans and combine them via workflow control structures into one workflow. For example, if only human terms, chemistry and disease terms annotations are required, then only the three corresponding GridBeans are selected (see Figure 1, left hand side) whereas the choice of only human term annotations results in a workflow with only one GridBean (Figure 1, right hand side). The workflow on the right hand side in Figure 1 contains a workflow structure, the "For-Each" loop, which encloses the GridBean and allows for parallelization of the job. The first workflow contains no workflow structures, which allows only for an internal parallelization by UIMA.

The GridBeans for data extraction processes can be categorized into three types. The first type is a Reader-GridBean. The modules behind these GridBeans convert documents into the data format XCAS, which are Common Analysis System objects presented in XML [16]. Annotator-GridBeans are used to execute an annotation engine for an input document set. Finally, the Consumer-GridBeans execute a program which converts XCAS files to some specific output data format such as PDF.

UNICORE provides a GridBean service as well. GridBean services support the automatic download of all GridBeans associated with a defined Grid; once connected all available GridBeans are automatically download. This service also supports update mechanisms for previously installed GridBeans. We use a configured GridBean service to provide each user the new developed or updated GridBeans.

5. Developments

The Grid Programming Environment (GPE), which has been developed by Intel, supports a High-level API for programming Grid Clients and for developing GridBeans [17]. GridBeans are used to generate Job descriptions and to provide graphical user interfaces for defining input data and visualizing output data. They are constructed as presented in Figure 2 [18].

Each GridBean consists of application-specific (left hand side) and generic (right hand side) input and output panels, which provide graphical interfaces for input and output data. The application-specific panels are defined in the GridBean Plugin. The GridBean Model (middle) contains the internal data as key-value pairs and maintains the actual state of the job settings. It is only necessary to implement the GridBean Plugin, the input and output panels and the GridBean Model when implementing a new GridBean. The generic panels are predefined and added automatically in the UNICORE Rich Client. All GridBeans for data extraction processes consist of one GridBean Plugin, providing an application-specific input panel. The input panel provides graphical interfaces to define the job name, show the application name, and to set application-specific parameters. In case of the ProMiner-Human GridBean, the application-specific parameter is currently the number of threads to be executed pro node for the given Job.

It is possible to predefine input and output files in the generic file input panel. All data extraction GridBeans process files in a given input directory and mirror the input structure to an output directory containing the output files created. For all Reader-GridBeans, the output directory name is predefined, whereas for all Annotator- and Consumer-GridBeans the input directory name is also predefined. This allows the user of the GridBean a more comfortable workflow creation by using drag and drop of data control flows.

Figure 3 shows the data flow of the first workflow example consisting of ProMiner-GridBeans. The output files from the previous GridBean are used as the input files for the current GridBean, combined per drag and drop.

Currently, we have developed three Reader-GridBeans, 14 Annotator-GridBeans and three Consumer-GridBeans. All developed GridBeans can be used in different combinations in workflows, depending on use case, and provide user-friendly annotation-specific input and output panels.

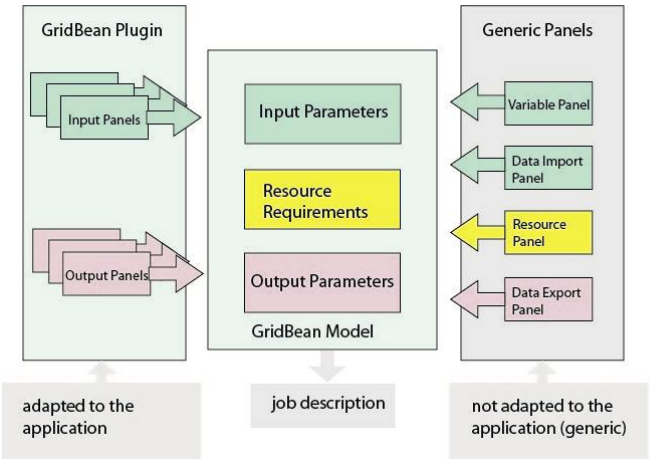

Figure 2: Structure of a GridBean (GridBean Plugin, GridBean Model and Generic Panels

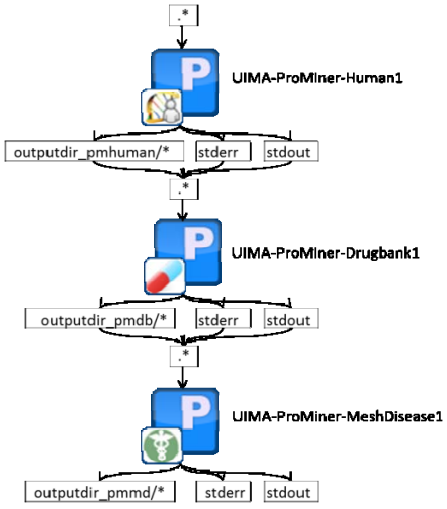

Figure 3: Data flow view of a workflow which contains ProMiner applications to be processed in sequence.

6. Results

This section presents experimental results concerning the runtime performance of the ProMiner Human application. 60 patents from EPO (European Patent Office, http://www.epo.org) were annotated by ProMiner Human. The patents have been chosen for their relevance and diverse scan quality.

In order to assess the relationship between the runtime and the size of input data, the ProMiner Human application was performed on each of the EPO patents, 10 identical runs were executed per patent. Each run was allocated 24 CPUs and 96GB RAM.

Figure 4 shows the results in a scatter diagram. It is apparent that runtime depends linearly on the size of the input data. The regression line provides an average runtime for a given input size.

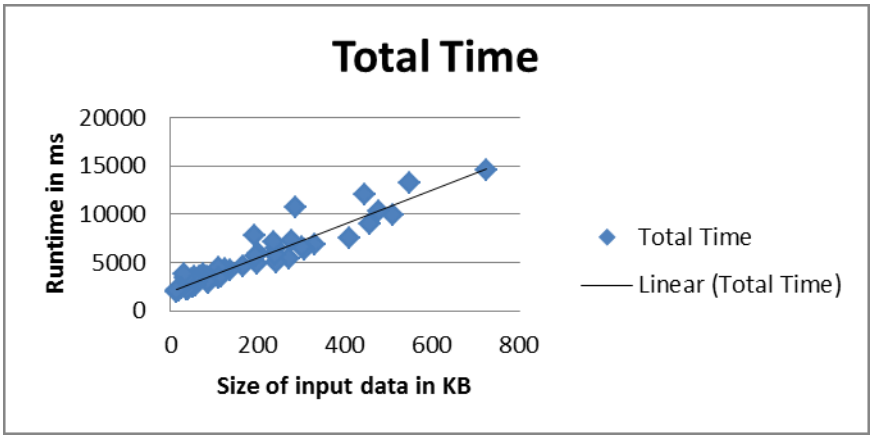

Figure 4: Correlation between runtime and input size data for the ProMiner Human application

In order to reduce the overall turn-around time, the input data is split and the number of jobs increased. The goal is to find the optimal number of jobs. The most important influencing factors are the number of resources, on which the application is installed, the number of available nodes and cores on a resource, and the size of the input data set.

In the case of the experiment presented, the size of the complete input data set is 9507KB. Assuming that we have 4 nodes on a cluster system, optimally we can create 4 jobs and each job gets roughly 2377KB as input.

The implementation for splitting the input data set currently uses a “Bin Packing” algorithm, which works as follows: Each data set is filled with input files until a predefined limit is reached, in this case 2377KB. If the limit is reached, the next data set will be created and filled. The order of file distribution depends on the order of files successfully transferred to the server. Different runs with the current “Bin Packing” algorithm implementation show that each time five jobs will be created instead of four. Instead of using four nodes which shows an optimal use of the available computing resources, we need more resources for the automatic mechanism. The automatic mechanism must be optimized by comparing different Bin Packing strategies and to find the optimal balance between computing time for the file distribution and the benefit for the runtime of the application.

In addition, the runtime of some applications have been evaluated concerning the usage of several numbers of cores. 10 identical runs for 60 patents were executed per ProMiner application. Each run was allocated 24 CPUs and 96GB RAM. The result of the ProMiner application runs yield the optimal usage of 11 cores for ProMiner Drugbank and MeshDisease, and 20 cores for ProMiner Human. In future the inner parallelization of all applications must be analyzed in addition to the outer parallelization with UNICORE scheduling algorithms.

7. Business Benefits

The UIMA-HPC project developments target the delivery of multi-modal document analysis services for a wide-range of knowledge-discovery applications, but using the analysis of patents in the pharmaco-chemical sector as a demonstrator and initial focal

point. The project developments focus on two specific end-user groups: patent agents and R&D experts. R&D experts request the highlighting of annotations, links to external data bases, jump labels or extraction of references. Each use case can be implemented by an individually constructed workflow. However patent agents focus on annotation of claims and referenced patents.

The delivery mechanism will be a Web-portal behind which a flexible computational infrastructure orchestrates the required combination of (sets of) analysis engines deployed on the HPC systems. The end-user is expected to be a technical expert from the application sector, e.g. a chemist, who has already assembled a set of documents for analysis and who may wish to provide proprietary additional information (such as vocabularies or business process-relationships) as a basis for the document analysis and annotation.

The web-portal will allow for user-selection, and customisation, from a variety of analysis types or alternatives. The Grid-Beans architecture used within the UNICORE-based "back-end" computational infrastructure facilitates the flexible construction (and orchestration) of different workflows for the realisation of the analysis needs prescribed by the end-user's choices. The architecture also allows for the simplified integration of new analysis engines, which allows for the service offering to end-users to be maintained, expanded and adapted much more rapidly and at reduced cost.

8. Conclusions

The concept of embedding annotation programs in UIMA and integrating these modules, each one by a distinct GridBean, in UNICORE has proven to be effective for the first workflows. The standardized interface provided through UIMA defines the data format between modules, XML, which allows for an easy combination of the modules in workflows. First annotation programs such as recognition of chemical and human terms are integrated and successfully tested via newly developed Annotator-GridBeans. Run time analyses verify the correlation between the runtime of the ProMiner application and the size of the input data. First scaling tests prove a runtime decrease up to 15,5% by using the optimal number of cores for the selected application. The first version of the "Bin Packing" algorithm was compared with the optimal distribution of files regarding the benefit for the runtime. This shows that the current version does not distribute the file in a perfect manner and must be improved.

9. Outlook

The next steps are the development of new annotation programs, their integration in UIMA and the UNICORE environment and the development of corresponding GridBeans. Some of the application tools developed are already scaling well. This must be improved further and is part of the optimization step in the UIMA-HPC project. Another issue is the optimization of the distribution into file sub sets. By the comparison of different brokering strategies we will analyze the important factors and improve the runtime.

UIMA-HPC is partly funded by the German Ministry of Education and Research (BMBF) under grant id 01IH11012A-D.

References

[1] UIMA-HPC project Web site: http://uima-hpc.de/en
[2] Warr W. A.:. Chemoinformatics and Computational Chemical Biology, volume 672 of Methods in Molecular Biology. Humana Press, Totowa, NJ, 2011.
[3] Zimmermann M., Fluck J., Thi L. T. B., Kol´arik C., Kumpf K., Hofmann M.:.Information extraction in the life sciences: perspectives for medicinal chemistry, pharmacology and toxicology. Current Topics in Medicinal Chemistry, 5(8):785–796, 2005.

[4] Hettne K. M., Stierum R. H., Schuemie M. J., Hendriksen P. J. M., Schijvenaars B. J. a., Mulligen E. M. V., Kleinjans J., Kors J. a.:.A dictionary to identify small molecules and drugs in free text. Bioinformatics (Oxford, England), 25(22):2983–91, November 2009.
[5] J. Yuan M.:. Watson and healthcare: How natural language processing and semantic search could revolutionize clinical decision support. developerWorks, April 2011.
[6] Jessop D. M., Adams S. E., Murray-Rust P.:. Mining chemical information from Open patents. Journal of cheminformatics, 3(1):40, October 2011.
[7] Algorri M.-E., Zimmermann M., Friedrich C. M., Akle S., Hofmann-Apitius M.:.Reconstrucition of chemical molecules from images. In Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society, volume 2007 of Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 4609–4612. Department of Digital Systems, Instituto Tecnologico Autonomo de Mexico, Mexico City. algorri@itam.mx, 2007.
[8] Filippov I. V., Nicklaus M. C.:. Optical structure recognition software to recover chemical information: OSRA, an open source solution. Journal of chemical information and modeling, 49(3):740–3, March 2009.
[9] Park J., Rosania G. R., Shedden K. a., Nguyen M., Lyu N., Saitou K.:. Automated extraction of chemical structure information from digital raster images. Chemistry Central journal, 3:4, January 2009.
[10] Ogren P., Bethard S.:. Building test suites for UIMA components. In Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009), pages 1–4, Boulder, Colorado, June 2009. Association for Computational Linguistics.
[11] Streit A., Bergmann S., Breu R., Daivandy J., Demuth B., Giesler A., Hagemeier B., Holl S., Huber V., Mallmann D., Memon A. S., Memon M. S., Menday R., Rambadt M., Riedel M., Romberg M., Schuller B., Lippert T.:. UNICORE 6 A European Grid Technology, volume 18. 2009.
[12] Demuth, B.; Schuller, B.; Holl, S.; Daivandy, J.; Giesler, A.; Huber, V.; Sild, S.; , "The UNICORE Rich Client: Facilitating the Automated Execution of Scientific Workflows," e-Science (e-Science), 2010 IEEE Sixth International Conference on , vol., no., pp.238-245, 7-10 Dec. 2010, doi: 10.1109/eScience.2010.42, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5693923&isnumber=5693891
[13] Bari, W., Memon, A. S., & Schuller, B. (2010). Enhancing UNICORE Storage Management Using Hadoop Distributed File System. Nuclear Physics, 6043 LNCS, 345-352. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-77954619523&partnerID=40&md5=6fed479f78d3f8b5ba0ba4c7486038a6
[14] Lynette Hirschmann, Martin Krallinger, and Alfonso Valencia, editors. Proceedings of the Second BioCreative Challenge Evaluation Workshop. Centro Nacional de Investigaciones Oncologicas, CNIO, Madrid, Spain, 2007. pp. 149–151: ProMiner: Recognition of Human Gene and Protein Names using regularly updated Dictionaries
[15] David S. Johnson, Fast algorithms for bin packing, Journal of Computer and System Sciences, Volume 8, Issue 3, June 1974, Pages 272-314, ISSN 0022-0000, 10.1016/S0022-0000(74)80026-7. (http://www.sciencedirect.com/science/article/pii/S0022000074800267)
[16] T. Gotz and O. Suhre. Design and implementation of the UIMA common analysis system. IBM Systems Journal, 43(3), 2004.
[17] Intel: Intel's Grid Programming Environment, An Overview, White Paper, http://gpe4gtk.sourceforge.net/GPE-Whitepaper.pdf
[18] Sandra Bergmann, May 2009, GridBean Developer's Guide, http://unicore.eu/documentation/manuals/unicore6/files/GridbeanDevelopersGuide.pdf