Research Plan

Preparing CERN's LHC Computing Grid for MPI-parallelization

LCG

Research Plan for RP2
University of Amsterdam
MSc in System and Network Engineering

Class of 2005-2006

Richard de Jong, Matthijs Koot
{rjong,mrkoot}@os3.nl

June 9, 2006

Contents

1 Introduction

As part of our Master of Science study in the field of System and Network Engineering at the University of Amsterdam, we will be doing research at CERN, Geneva [1], on enabling MPI-based parallel computing on the LHC Computing Grid (LCG). The research is performed on behalf of NIKHEF [2], Amsterdam, and is supervised by David Groep. The work will be done in close cooperation with Louis Poncet from CERN.

2 Research goals

Parallel programming is the art of using multiple processors to solve a single problem. The traditional paradigm of computer architecture is the exact opposite: to solve multiple problems with a single processor (serial computing). With millions of scalar computers now connected through the Internet, the perspective on computing is slowly changing to that of 'the network is the computer'; in stead of using a single processor to solve a problem, a global network of processors is now available to solve (the same or larger) problems. Historic methods of parallel computing included threading, IPC and Parallel Virtual Machines (*PVM*). None of them, however, are really suitable for a 'heavily distributed' environment, such as the globe-spanning LCG [3]. In response to this, a new protocol was designed to succeed PVM and has now, a decade later, become a *de facto* standard for massively parallel computing: the Message Passing Interface, or *MPI* [4]. MPI, which is a library specification, solves the problem of inter-process and job communication and allows data to be passed between processes in a distributed-memory environment. The prime quality attributes of MPI are source code portability, i.e. to support heterogeneous parallel architectures, and to allow efficient implementation, i.e. allow optimization for certain hardware platform [5]. Although MPI may be used in shared-memory architectures (SMP, NUMA), its original design was focussed on distributed-memory architectures. It is the latter type of environment in which MPI is used at the LCG; many scalar processors with their own memory, mostly grouped into scientific computing clusters, connected over the Internet.

So far for MPI and parallel computing. Where does the concept of a *grid* fit in? According to Ian Foster, one of the 'fathers of the grid', a grid is a system that [6]:

1. . . . coordinates resources that are not subject to centralized control;
2. . . . using standard, open, general-purpose protocols and interfaces;
3. . . . to deliver nontrivial qualities of service.

Thus, it is more than simply a bunch of interconnected computational resources (which might rather be called a *cluster*); it includes social and political aspects as well. The most prevalent vision on grid computing is that of 'service-oriented computing', i.e. considering computational resources to be like water and electricity facilities [7, 8]. Exploitation of these means for scientific purposes is also denoted with the term *e-Science*, as coined by John Taylor [7]. Building on knowledge and code from Globus Toolkit, the European DataGrid project, or *EDG*, and the Enabling Grids for E-sciencE project, or *EGEE*, the LCG project at CERN aims to deliver a production-quality world-wide grid for scientific purposes (and perhaps commercial usage at a later stage). It is believed that like every system needs an IP-address to be connected to the Internet, every system to be connected to 'the Grid' shall need to support the Open Grid Services Architecture, or *OGSA*. We will take OGSA into account where applicable.

During years of research on distributed computing, a general multilayer model has been developed for grid architectures, which is depicted in figure 1. This model is semantically comparable with the TCP/IP model used for Internet. The functions placed between the *User Applications* layer and the *Fabric* layer are provided by *Grid middleware*. In the case of the LCG, that middleware is called *gLite*. gLite, which was previously unpractically and ambiguously named *EDG* and *LCG*, is a piece of software delivered by the JRA-1 group of EGEE. The gLite middleware is typically deployed through Yet Another Installation Method, or *YAIM*, an installation method developed in the course of Grid deployment. The OGSA specifies interfaces for all layers.

That being said, these are the goals of our research:

- define the requirements for integrating MPI into the gLite middleware and the YAIM deployment scheme, so that, if enabled during gLite deployment, MPI jobs may be submitted through the LCG grid interface (this includes scalability and dependency issues);
- integrate (and demonstrate) MPI into the gLite middleware and the YAIM deployment scheme;
- evaluate YAIM by assessing it on various quality attributes, e.g. extensibility.

3 Project scope

3.1 MPI

In Grid terminology, a Computing Element, or *CE*, is an abstraction of Worker Nodes, or *WNs*. CEs represent an interface to computing resources. When an end-user submits a *job* described in the Job Description Language, or *JDL*, through an User Interface, or *UI*, the job is sent to a Resource Broker, or *RB*. The RB will notify the Logging & Beekkeeping component, or *LB*, and then query the Grid Index Information Service, or *GIIS*, to find CEs that are capable of executing the job. If such a CE, or group of CEs, is found, the RB delegates the job to it by accessing it through its *Gatekeeper*. Our research on MPI will be

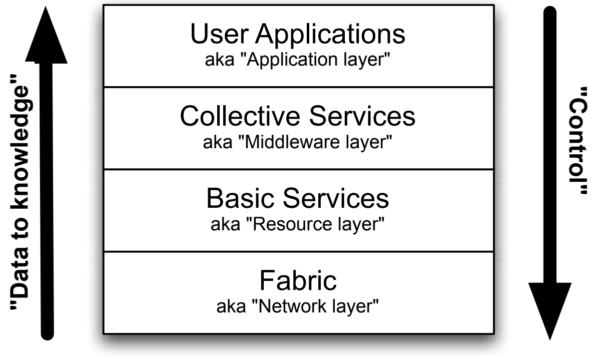

Figure 1: The layered grid model

focussed on MPI within a single CE. Only if time allows it, we will investigate the possibilities of spanning MPI jobs across a group of CEs.

3.2 YAIM

We will assess YAIM on elementary software quality attributes, as well as on adherence to best practices and standards that may have been defined by the Global Grid Forum, or *GGF*[1]. If, however, any such GGF publication exceeds the level of detail that we can handle in one month, we won't perform an exhaustive evaluation; rather, we'll pick the 'most important' elements from the GGF specs (with 'most important' being determined in cooperation with the IT/Grid Deployment team).

4 Project organization

The project is organized in an informal way and consists of the people and roles defined below. As a means of minimum quality assurance, we will have a weekly feedback meeting at CERNs site in Meyrin. The meetings will be scheduled as time goes by.

Name	**Role(s)**	**Contact**
Richard de Jong	project member	rjong at os3.nl
Matthijs Koot	project member	mrkoot at os3.nl
David Groep	supervisor NIKHEF	david.groep at nikhef.nl
Jeff Templon	contact at NIKHEF	jeff.templon at nikhef.nl
Louis Poncet	contact at CERN	louis.poncet at cern.ch
Cees de Laat	university supervisor	delaat at science.uva.nl

[1] GGF is often called 'the IETF of Grid computing'.

5 Project resources

Project resources may be roughly divided in two categories: human (H) and non-human (NH) resources. The following resources are anticipated:

MoSCoW	**Type**	**Resource**
MUST	NH	Access to the information about MPI.
MUST	NH	Access to the information about Grid computing.
MUST	NH	Access to the LCG production-bed.
MUST	NH	Access to gLite architecture documents.
MUST	NH	Access to gLite sources.
MUST	NH	2+ systems for testing RB/CE.
SHOULD	H	Vizzavi access to gLite developers for support.
WOULD	NH	Access to parallel debugger tools.

The first two resources are readily available on Internet (i.e. academic papers, e-books). Through our CERN Access Card and computer account, we already have access to the gLite documents and sources. Access to the LCG production-bed has been granted through NIKHEF, which has assigned us grid certificates. As of June 7th, Louis Poncet at CERN has provided us with four systems (running Scientific Linux 3.0.6) which we may use freely to experiment with grid components. We are physically located at the IT/Grid Deployment group and are amongst several people with in-depth knowledge of grid middleware, but (as far as we currently know) none of them are gLite software engineers. However, based on earlier conversations, we expect their level of knowledge of the gLite internals will suffice for our purposes. Access to parallel debugger tools, e.g. TotalView or the Distributed Debugging Tool is not critical for our purposes; it aids in troubleshooting running MPI jobs, but our research mainly focusses on getting MPI jobs to be executed in the first place.

6 Project deliverables

The deliverables for this project are as follows:

- research plan (this document);
- research report;
- production-ready integration of MPI within gLite and YAIM.

The research plan is the document you're currently reading; it describes the goals and requirements for the project, as well as a roadmap. We will describe our findings in the research report, for which a preliminary template is listed in Appendix A - Structure of the Research Report. As a minimum, the report will include architectural considerations for using MPI on grids, an advise to the LCG group, and an assessment of YAIM. If all goes as planned, the last deliverable will be a set of files and accompanying documentation for actually integrating MPI within the gLite deployment process.

7 Project planning

Aligning with the requirements of our university, these dates are considered the deadlines for this project:

Deadline	**Deliverable**	**Comments**
June 9th, 2006	Research plan	Will be sent to David Groep, Louis Poncet and Cees de Laat for approval.
June 30th, 2006	gLite/YAIM-pkg for MPI	Will be sent to Louis Poncet for approval.
June 30th, 2006	Research report	Will be sent to David Groep, Louis Poncet and Cees de Laat for approval.
July 8th, 2006	Presentation	A public event in the Turingzaal at SARA, Amsterdam.

During the period of June 5th to June 9th, we will be gaining more in-depth knowledge on grid computing and MPI by reading related work, tutorials and performing some simple non-MPI jobs on the LCG and simple MPI-jobs on a MPI-enabled cluster. From there, we should be able to understand the issues for which requirements will be defined. Between June 12th and June 16th, we will be working on a gLite-based package for MPI (albeit OpenMPI, MPICH-G2, PACX-MPI or another implementation), as well as its integration into YAIM. In the week of June 19th to June 23rd, we will investigate the quality of YAIM and advise on how it might be improved. The latter may include best practices defined by the Global Grid Forum, as well as general software quality attributes. In the last week of June 26th to June 30th we will deliver our research report to the relevant persons (as listed in the above table). We anticipate a buffer of three days in the last week for delays in our research.

8 Copyrights and ownership

All documents which are created as a part of this project will be licensed under the Creative Commons 2.5 Attribute license [10]. All source and object code which is produced as a part of this project will be licensed under the revised BSD license [11].

Appendix A - Structure of the Research Report

The research report, our main deliverable, will consist of the elements summarized in the next subsections. The sections will be written in the course of our research.

Introduction

This section will contain a general introduction into grid computing and parallel programming with MPI.

Scope

This section will contain the definite specification of the scope of our research. The scope may be subject to slight changes at the beginning of our project due to knowledge obtained during that period. Any such changes will be definite *only* after approval of the supervisor(s).

Related work

This section will summarize related work / previous art.

MPI-parallelization on the LCG

This section will describe our findings on how to use MPI on grids, as well as an advice to the LCG group on how to proceed with enabling MPI. Best case scenario, this advice will also include an actual implementation (i.e., architectural modification to gLite or other components requiring modification to support MPI).

Evaluation of YAIM

This section will describe our evaluation of YAIM, which will preferable be performed within a GGF framework (i.e. best practices for grid deployment).

Future work

This section will contain some suggestions for future work.

Conclusion

The report will be finalized with a conclusion summarizing our results, and lastly a reflection on the goals defined in the research plan.

Appendix B - The BSD license for this project

Copyright (c) 2006, Richard de Jong and Matthijs Koot
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the University of Amsterdam nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

References

[1] CERN: European Organization for Nuclear Research, www.cern.ch

[2] NIKHEF: Dutch Institute for High Energy and Nuclear Physics, www.nikhef.nl

[3] Gropp, William and Lusk, Ewing: *"PVM and MPI are completely different"*, 1997, http://citeseer.ist.psu.edu/573977.html

[4] MPI Forum: Message Passing Interface (MPI) Forum homepage, 1998, http://www.mpi-forum.org/docs/docs.html

[5] Fagg, Graham and London, Kevin: *"MPI Inter-connection and Control"*, 1998, http://citeseer.ist.psu.edu/400213.html

[6] Foster, Ian: *"What is the Grid? A Three Point Checklist."*, 2002, http://www-fp.mcs.anl.gov/ foster/Articles/WhatIsTheGrid.pdf

[7] Foster, Ian: *"Service-Oriented Science."*, 2005, http://www.sciencemag.org/cgi/content/short/308/5723/814

[8] Papazoglou, M. and Georgakopoulos, D.: *"Service-oriented computing: Introduction"*, 2003, http://portal.acm.org/citation.cfm?doid=944217.944233

[9] Gropp, William and Lusk, Ewing: *"Goals Guiding Design: PVM and MPI"*, 2002, http://citeseer.ist.psu.edu/568858.html

[10] Creative Commons: Creative Commons Attribution 2.5 license, www.creativecommons.org

[11] Open Source Initiative, The BSD License, www.opensource.org