Bull. Nov. Comp. Center, Comp. Science, 22 (2005), 41–49

NumGRID software for MPI-based applications*

D. Fougere, M. Gorodnichev, N. Malyshkin,
V. Malyshkin, A. Merkulov, B. Roux

1. Introduction

With evolution of mathematical modeling and creation of high-performance computer systems, many scientific applications have appeared, demanding an increase in computational performance higher than any of available supercomputers can provide.

In this connection, it seems necessary to integrate several supercomputers, i.e., to create a grid. Not every application can effectively be solved on grids because of low capacity of communications networks. However, such applications as the search for alien civilizations and decoding the human genome were successfully run on grids (http://setiathome.berkeley.edu).

In recent years, many large-scale problems have appeared in numerical modeling (evolution of a proto-planetary disk, evolution of galaxies, problems of plasma physics). Their solution demands integration of several multicomputers.

Another problem is the fast progress in microprocessor development which forces to use heterogeneous computer systems for solution of the large-scale problems. In particular, in 2005, the ICM&MG (Siberian Branch of the Russian Academy of Sciences) is planning to exploit the following multicomputers: a cluster of MVS-1000 type based on 3L Alpha EV68 microprocessors (833 MHz), another MVS-1000 cluster based on 128 Alpha EV67 microprocessors (633 MHz), and a multicomputer based on 32 Intel Itanium processors. Therefore, there is a necessity to create the software, which should provide the large-scale simulation in such heterogeneous environments.

In 2003, the co-operative project NumGRID intended for the creation of the necessary grid system software, started in Novosibirsk (ICM&MG) and Marseille (L3M CNRS).

*Partially supported by Dutch-Russian Project "High Performance Simulation on the Grid", contract NWO-RFBS 047.016.007; Dutch-Russian Project "Investigation of Plasma Induced Cluster Formation and Thin Film Deposition", contract NWO-RFBS 047.016.018; RFBR No. 04-01-00272.

2. Objectives

The main objective of the NumGRID project is to provide the use of multicomputers for the large-scale numerical modeling. In particular, in the near future, the proper computational resources for the simulation of protoplanetary disc evolution and galaxy formation should be provided. These models, developed within the frame of the project and dedicated to the investigation of the germ of life on the Earth, demand practically unlimited resources for simulations.

Another objective is to provide exploitation of a heterogeneous net of multicomputers for the numerical modeling.

Basic requirements:

- Multicomputers only are included into the NumGRID. Each node of a multicomputer can be an SMP system (two processors or more).
- MPI-programs should be executed on the NumGRID without changes. The global addressing of all the NumGRID resources should be provided.
- A language for program execution and optimization of communications should be developed. Our efforts are mostly concentrated on organization of a high-performance execution of application programs.
- The dynamic properties of a program under execution (tunability, dynamic load balancing, program execution monitoring, reliability) should be automatically provided.
- High security and safety of calculations on the Grid should be provided.
- Simple and powerful user tools should be developed to simplify processes of developing, debugging and running user programs.

3. The NumGRID approach

The description of the NumGRID approach can be found in [1]. Here a short description of the NumGRID key ideas is given. The problem of using the MPI on global grids is that all the multicomputers restrict access to their internal nodes from outside. Thus, the MPI-packets cannot reach their addressees. So, the main idea of the NumGRID is the following. A special program is loaded into master-nodes of all the multicomputers included into the NumGRID. This program is called the MPI_gateway. This program is actually a gateway service for the MPI-packets routing. This router provides receiving and sending packets to the internal (private) nodes of any NumGRID multicomputer. The router transfers the MPI-packets from the internal nodes of a multicomputer (Cluster 1) to the router installed on the

master of the other multicomputer (Cluster 2). Later the second router transfers the received MPI-messages to its internal nodes (inside Cluster 2).

Another part of the NumGRID software is a special NumGRID_MPI library with a possibility to send messages not only within a local network, but also through the MPI_gateway server to global networks. We proceed from the assumption that there are some MPI libraries pre-installed on each cluster/SMP-system. Therefore, it is desirable that application programs use this MPI libraries. Implementing this idea, the developed NumGRID_MPI library is actually a "wrapper" of the pre-installed MPI-functions. That is, inside a component of a grid (clusters and/or the SMP), all messages are transferred using the pre-installed MPI library, and all messages "outside" are transferred using the developed NumGRID_MPI library through the MPI_gateway.

Additionally, the NumGRID offers a lot of other useful features such as a friendly user environment, a high security, debugging and monitoring systems, an executive system, etc.

4. Implementation

The NumGRID software is developed as a cross-platform environment that can be freely used on many hardware architectures and operating systems. As the first NumGRID software implementation is done for the GRID of clusters, the terms "multicomputer" and "cluster" are equally used.

By now the software has been tested on:

Processors: Intel Celeron/P3, AMD Athlon, Alpha;

OS: Windows 2000/XP, Red Hat LINUX;

Compilers: Visual C++ 6.0, GNU C++.

The NumGRID software consists of seven modules:

- NumGRID_MPI library;
- NumGRID_gateway;
- Executive subsystem;
- Cluster management module;
- Client module (console tools & IDE);
- Security subsystem;
- Debugging and monitoring system.

NumGRID_MPI library is used for the MPI calls in the user's programs. It is able to transfer the MPI-messages not only inside clusters but also between clusters through the NumGRID_gateway. The NumGRID_MPI library actually represents not a special MPI standard implementation but

a lot of "wrappers" for common MPI-functions. In order to use the NumGRID_MPI library one should have any MPI implementation on all the multicomputers that are planned to be included into the NumGRID. The NumGRID_MPI library is able to support many common MPI implementations such as MPICH, LAN-MPI, SCALI MPI and some others.

NumGRID_gateway is a service to be installed on all the master nodes of all the clusters of the NumGRID. This service transfers the MPI-packages inside and outside the cluster it controls.

Executive subsystem Programming tools. This includes a number of tools that facilitate the development of programs for heterogeneous computational systems. First of all, it includes a library of routines that allows one to reveal properties of the system components, such as performance of computational nodes and bandwidth of network links.

Cluster management module allows the NumGRID software to use all the cluster facilities to run users' jobs, to transfer data to/from clients, to allocate resources, and so on. The module can work with common system software that manages the cluster's queue. The first queue management systems that are planned to be supported are PBS, SGE and MVS1000. This module is very close to the Globus job-managers modules.

Client module (console tools & IDE) allows the user to submit a job for execution on the NumGRID, to put/receive data into/from the NumGRID, to control and to manage resources and so on. Console tools are close to the Globus' globus-job-run, globus-job-submit. The IDE allows running all the above operations in a friendly graphical environment.

Security subsystem is used in different NumGRID components. It provides a high safety and is based on Kerberos.

Debugging and monitoring system can be integrated in many NumGRID applications. It is used for debugging parallel applications on the NumGRID, viewing and analyzing the results of numerical experiments.

5. Experiments

A pilot version of the NumGRID software has been developed. A number of experiments have been carried out with the current implementations of NumGRID_MPI and NumGRID_Gateway systems.

5.1. Objectives. The experiments were supposed to:

- demonstrate the ability to run the MPI-programs with the MPI-processes spread over several clusters on the basis of the NumGRID approach;
- reveal the problems that might occur.

5.2. Equipment. Three remote clusters were involved in testing the NumGRID software. Two of them, named SCALI and CHOEUR, are located in the L3M laboratory of CNRS-d'Aix-Marseille II University and interconnected with the Gigabit Ethernet network. The third one, p2chpd-cluster, is located at the UFR de Mecanique of the Lyon University.

Cluster name	Intranetwork	Hosts	Computational nodes	Nodes number
Choeur	Fast Ethernet	2 processors AMD Athlon MP 2000+, 1.67 GHz, 2 Gb RAM	2 processors AMD Athlon MP, 1.53 GHz, 1 Gb RAM	20
Scali	SCI Dolphin	2 processors AMD Athlon MP 1800+, 1.5 GHz, 1 Gb RAM	2 processors AMD Athlon MP 1800+, 1.5 GHz, 1 Gb RAM	5
P2chpd-cluster	SCI Dolphin	2 processors AMD Athlon MP 2000+, 1.67 GHz, 2 Gb RAM	2 processors AMD Athlon MP 2000+, 1.67 GHz, 1 Gb RAM	40

5.3. Test 1. Shift of data along the virtual circle of MPI-processes

Description. Each process of rank R transfers some double precision number to a process of rank $(R + 1) \bmod P$, where P is the number of processes.

Purpose. This is a simple example of the MPI-program, which should demonstrate the capability of the NumGRID software to support the basic MPI communications.

Algorithm. For the sake of simplicity, P is assumed to be even. First, all the processes with even ranks send their data to their "right" neighbors, then the odd-ranked processes do the same.

Result. This was the first test. It has been passed successfully for $P = 2, 4, 6, 8$ with different distributions of processes between CHOEUR and SCALI, between CHOEUR and p2chpd-cluster. There were no problems at all with the program execution on CHOEUR and SCALI. The same test on the CHOEUR and p2chpd-cluster took a lot of efforts and time to agree upon firewall policies in order that the data transfer between these two clusters be provided.

5.4. Test 2. Each process receives data from all the other processes

Description. In turn, the processes receive messages from all the other processes. The messages are ordered according to the rank of sending processes.

Purpose. This test demonstrates that the NumGRID_MPI keeps an assumed order of incoming messages.

Algorithm.

```
For i from 0 to P-1 {
  if R equals I  {
    For j from 0 to P-1:
      if not j equals R,
        receive a message from the process with rank j
  }
  else,
    send a message to the process with rank i
}
```

Result. The test has passed for 2, 4, 6 and 8 processes distributed (in different combinations) between CHOEUR and SCALI and between CHOEUR and p2chpd-cluster. All the messages have been received in the expected order.

5.5. Test 3. Performance of communications

Description. The measurements of performance of data transfer between two communicating processes located:

a) on the same computational (SMP) node;

b) on different nodes;

c) on different clusters

Purpose. The purpose of this test was to compare the performance of communications as depending upon different mutual dispositions of communicating processes. It is important for further development of the project to take into account the great heterogeneity of the target systems.

Algorithm. The two processes simply send 1Mb messages to each other alternatively. This is done several times in order to reduce the error.

Result. The performance of data exchanges is presented in the table bellow. The experiments demonstrate a difference in the times of data sending and receiving. The results are shown for both types of activities. In experiment number 9, the sending performances were different for two communicating processes.

5.6. Test 4. Performance of communication (O. Bessonov's test)

Description. The test is composed of several series of data exchanges between MPI-processes. A difference in times between the data transfer series lies in the number of exchange operations and the volume of messages.

Experiment number	Disposition of the communicating processes	Speed, Mb/s
1	2 processors of the SMP node at CHOEUR	send: 147.96 receive: 55.04
2	2 processors of the SMP node at SCALI	send: 200.37 receive: 151.39
3	2 processors of the SMP node at p2chpd-cluster	send: 187.87 receive: 123.21
4	2 nodes at CHOEUR FastEthernet connection	send: 11.27 receive: 9.96
5	2 nodes at CHOEUR Gigabit Ethernet	send: 76.1 receive: 43.7
6	2 nodes at SCALI, SCI	send: 140.58 receive: 101.72
7	2 nodes at p2chpd-cluster, SCI	send: 220.85 receive: 158.36
8	The host machines of CHOEUR and SCALI	send: 2.75 receive: 0.46
9	The node at CHOEUR and the node at SCALI	send from CHOEUR: 2.39 send from SCALI: 1.87, receive: 0.35
10	The host machines of CHOEUR and p2chpd-cluster	send from p2chpd-cluster: 1.28 receive: 0.19
11	The node at CHOEUR and the node at SCALI	send: 1.28 receive: 0.17

The results are given as average performance observed during the series, i.e., the relation of the data volume (Mbytes) transferred during the series to the time the series took. To gain the correct understanding of the results, one has to remember that the time of a series comprises not only the time of data transfer but also latencies, the time when the processes wait for each other and other overheads. O.Bessonov published the results of his test on SCALI and CHOEUR in [2].

Purpose. The time costs of the series of data exchanges are studied as depending on the number of exchanges and the volume of messages.

Algorithm. Exchanges are carried out in series. Within a series, two processes send messages to and receive from each other alternatively. The parameters of the series are presented in the table in the Results section.

Results again demonstrate a considerable difference between intra-cluster and inter-cluster communication speeds and show the role of latencies. The communication scheme is typical of numerical iterative computations.

Measure	Series							
	1	2	3	4	5	6	7	8
Number of send-receive pairs	8192	4096	2048	1024	5120	2560	1280	640
Message size, 8 byte numbers	64	128	256	512	1024	2048	4096	8192
Amount of transfers for the series, Mb	8	8	8	8	80	80	80	80
Speed between CHOEUR nodes (Gigabit), Mb/s	5.66	8.48	13.71	22.92	33.74	44.27	51.33	54.54
CHOUR–SCALI speed, Mb/s	0.010	0.019	0.032	0.064	0.12	0.85	1.41	0.93
CHOEUR–p2chpd speed, Mb/s	0.010	0.014	0.029	0.058	0.11	0.76	0.62	0.71

5.7. Test 5. Parallel implementation of the explicit Poisson solver

Description. Solution to the 2D Poisson equation with the finite differences explicit 5-point stencil scheme. It is one of simple examples of a somewhat real computational problem.

Purpose. This test demonstrates how the NumGRID software copes with the computational MPI-program, thus approaching useful applications.

Algorithm. A simple data-partitioning parallelization of the explicit scheme on a virtual line of processors.

Results. The test has successfully passed for 2, 4, 6, and 8 processes. The correctness of data transfers has been demonstrated. When the size of a mesh (that is actually equal to the size of a problem) is large, a good speed up can be observed, because calculations dominate over communications.

6. State-of-the-Art

The approach to solving the large-scale numerical problems that demand intensive communications between subparts of the NumGRID was elaborated. The basic architecture of the NumGRID system was designed. The states of all modules are listed below:

- NumGRID_MPI library — a pilot version designed and implemented (see tests above);
- NumGRID_gateway — a pilot version designed and implemented (see tests above);
- Executive subsystem — is being designed;
- Cluster management module — partially implemented;
- Client module (console tools & IDE) — under construction;
- Security subsystem — partially designed and implemented;
- Debugging and monitoring system — under construction.

By now an experimental prototype of the NumGRID system that allows the MPI calls between different clusters has been developed and tested.

7. Conclusions

A lot of firewalls between clusters make great administrative problems. This is because the NumGRID uses the TCP protocol for communications between the host machines and most of the TCP ports are closed with the firewalls. The solution is to use some standard communication services that are always open. The most widespread is the SSH protocol, and the suggestion is to have two transport subsystems in the NumGRID. The TCP protocol should be preferably used, because the SSH has an overhead of encryption and the SSH should be used in the case, when the TCP eligible ports are closed.

The experiments demonstrated the effectiveness of the general design of the NumGRID software.

The measurements of communication performance between differently disposed processes reveal the ultimate heterogeneity of the target computer system. It is important to note that the computational power of the nodes differs as well. Moreover, in a general setting, nodes might be of different architectures. The latter should be taken into account when the NumGRID_MPI is designed and implemented. Also, the heterogeneity of the network and processor elements should be taken into account when application programs to be run on such a computer system are developed. It is necessary to develop programming tools to facilitate the work of application programmers.

8. Future work

The next step is to produce a stable release of the NumGRID software, test and install it on ICM&MG + NSU, L3M Marseille + UFR Lyon clusters. The release should contain stable versions of the NumGRID_MPI library and the NumGRID_Gateway and the other components.

References

[1] Fougere D., Malyshkin N.V., Malyhskin V.E., Roux B. The NumGRID metacomputing system // NCC Bulletin. Series Computer Science. — Novosibirsk: NCC Publisher, 2004.

[2] Bessonov O., Fougere D., Roux B. Using a Parallel CFD Code for Evaluation of Clusters and MPPs // Proc. Int. Parallel and Distributed Processing Symposium — IPDPS 2003. — Nice (France), 2003. — (IEEE Computer Society; PR01926).