

Hungarian Supercomputing Grid¹

Péter Kacsuk

MTA SZTAKI

Victor Hugo u. 18-22, Budapest, HUNGARY

www.lpds.sztaki.hu

E-mail: kacsuk@sztaki.hu

Abstract. The main objective of the paper is to describe the main goals and activities within the newly formed Hungarian Supercomputing Grid (H-SuperGrid) which will be used as a high-performance and high-throughput Grid. In order to achieve these two features Condor will be used as the main Grid level job manager in the H-SuperGrid and will be combined with P-GRADE, a Hungarian produced high-performance program development environment. The generic Grid middleware will be based on Globus. The paper describes how P-GRADE, Condor and Globus will be integrated in order to form the H-SuperGrid.

1 Introduction

In the March of 2001 a 96-processor Sun Enterprise 10000 supercomputer (462nd position in the top500 list at that time), two 16-processor Compaq Alpha Server supercomputers, and several large PC-clusters were installed in Hungary as major supercomputing resources. All are placed at different institutes and are used by a growing user community from academy. However, even in this early stage of their use it turned out that there exist applications where the capacity of individual supercomputers and clusters are not sufficient to solve the problems at reasonable time. The solution for this problem is to connect these high-performance computing resources by the Hungarian academic network and to use them jointly in a newly formed supercomputing Grid. In order to achieve this goal a new 2-year project, called the Hungarian Supercomputing Grid (H-SuperGrid) project, started in January 2002.

One of the main goals of the project is to establish this Hungarian supercomputing Grid based on the current Hungarian and international results of cluster and Grid computing research. The project is strongly related with two already running Hungarian Grid projects (NI2000 [1], DemoGrid), two international projects (DataGrid [2], LHC Grid) and several national projects from other countries (Condor [3], INFN Grid [4], UK e-science [5]). The SuperGrid project is based on the experience learned on Globus [6] and Condor in the Hungarian NI2000 project and will strongly collaborate with the other running Hungarian Grid project, called DemoGrid (see Figure 1.).

¹ The research on the Hungarian Supercomputing Grid is supported by the OTKA T032226 and the OM IKTA-00075/2001 projects.

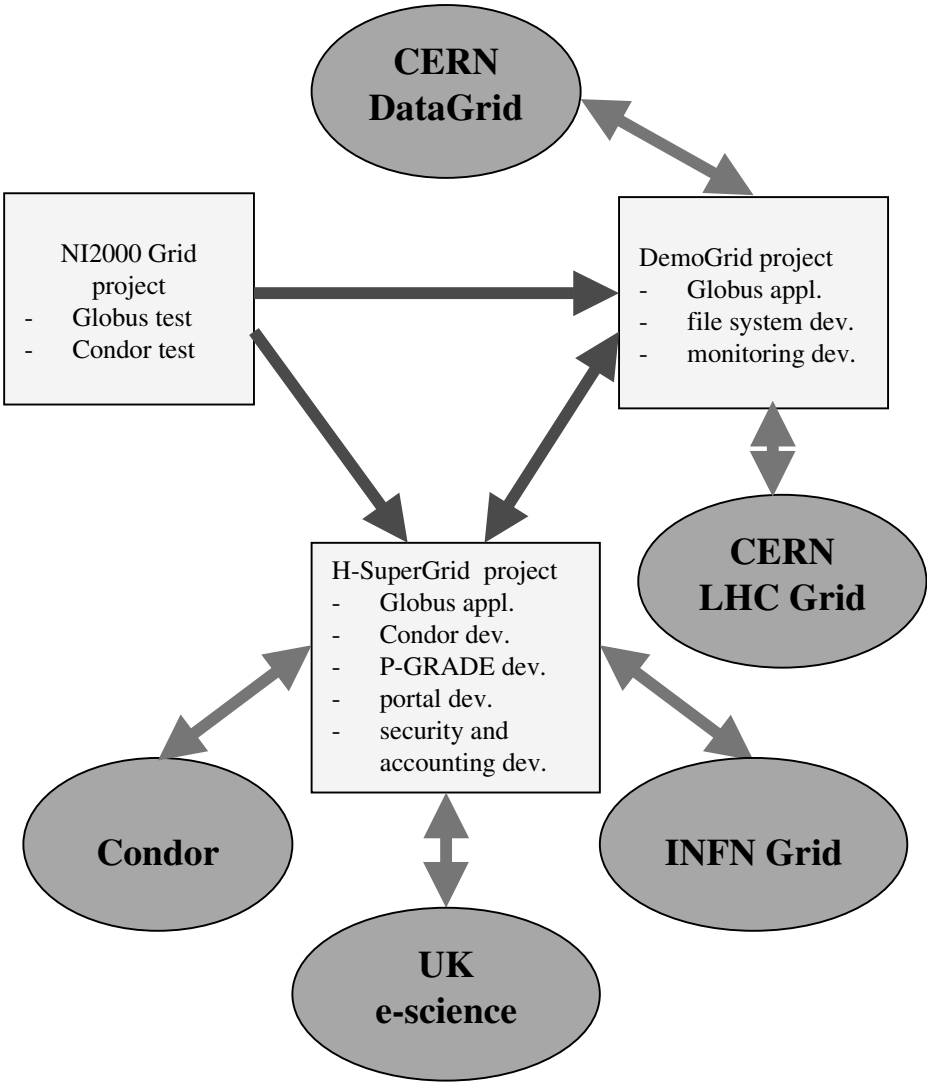


Fig. 1. Links between the Hungarian and International GRID projects

The H-SuperGrid will be used as a high-performance and high-throughput Grid. In order to achieve these two features Condor will be used as the main Grid level job manager in the H-SuperGrid and will be combined with P-GRADE [7], a Hungarian produced high-performance program development environment.

The heart of the H-SuperGrid is P-GRADE, a graphical parallel programming environment that supports the whole life-cycle of parallel program development and makes parallel programming reasonable easy for scientists and other non-professional programmer end-users. The major goal of P-GRADE is to provide an easy-to-use, integrated set of programming tools for the development of general message-passing

applications to be run in heterogeneous computing environments. Such systems currently include supercomputers and clusters. As a result of the H-SuperGrid project P-GRADE will be usable for the Grid and in this way the same programming environment will cover the whole range of parallel and distributed systems.

2 Structure of the H-SuperGrid project

The structure of the H-SuperGrid project is shown in Figure 2. The top layer is the application layer where currently a Monte-Carlo method based application is investigated in the field of simulating nuclear techniques. The user will access the H-SuperGrid by the Grid portal to be developed in the project. The application will be developed in the P-GRADE parallel program development environment which will be connected to Condor in the project. It means that the user can generate directly Condor jobs (containing parallel PVM [8] or MPI [9] program) from the P-GRADE environment. Condor will be used as the Grid level resource manager in the H-SuperGrid. The basic middleware services will come from Globus. The fabric will contain the Hungarian supercomputers and clusters connected by the Hungarian academic network. On the supercomputers and clusters local job schedulers like LSF, PBS, Condor, Sun Grid Engine will be used.

3 P-GRADE

The main advantages of P-GRADE can be summarised as follows. Its visual interface helps to define all parallel activities without knowing the syntax of the underlying message-passing system. P-GRADE generates all message-passing library calls automatically on the basis of graphics. Compilation and distribution of the executables are performed automatically in the heterogeneous environment. Debugging and monitoring information is related directly back to the user's graphical code during on-line debugging and performance visualisation.

P-GRADE currently consists of the following tools as main components: GRAPNEL (GRaphical Parallel NEt Language [7]), GRED (GRaphical EDitor [10]), GRP2C (pre-compiler to produce C/C++ code with PVM/MPI function calls from the graphical program [11]), GRM Monitor (to generate trace file during execution), DIWIDE (distributed debugger [12]), PROVE (performance visualisation tool [14]). P-GRADE is currently used at several universities of three continents for teaching parallel processing. Recently, the SYMBEX COST D23 project accepted P-GRADE to use for parallelising chemistry code and to make it part of a chemistry-oriented metacomputing system to be developed in the SYMBEX project.

P-GRADE has been successfully used by the Hungarian Meteorology Service to parallelise their nowcast program package, called MEANDER. A typical P-GRADE snapshot used in the parallelisation of the CANARI algorithm of MEANDER is illustrated in Figure 3. The Application window of the figure shows how individual processes (master) and the process farm template (Slaves) are graphically represented in GRAPNEL. The overall speed-up achieved with this program was 7.5 using 11 processors on an Origin 2000 supercomputer.

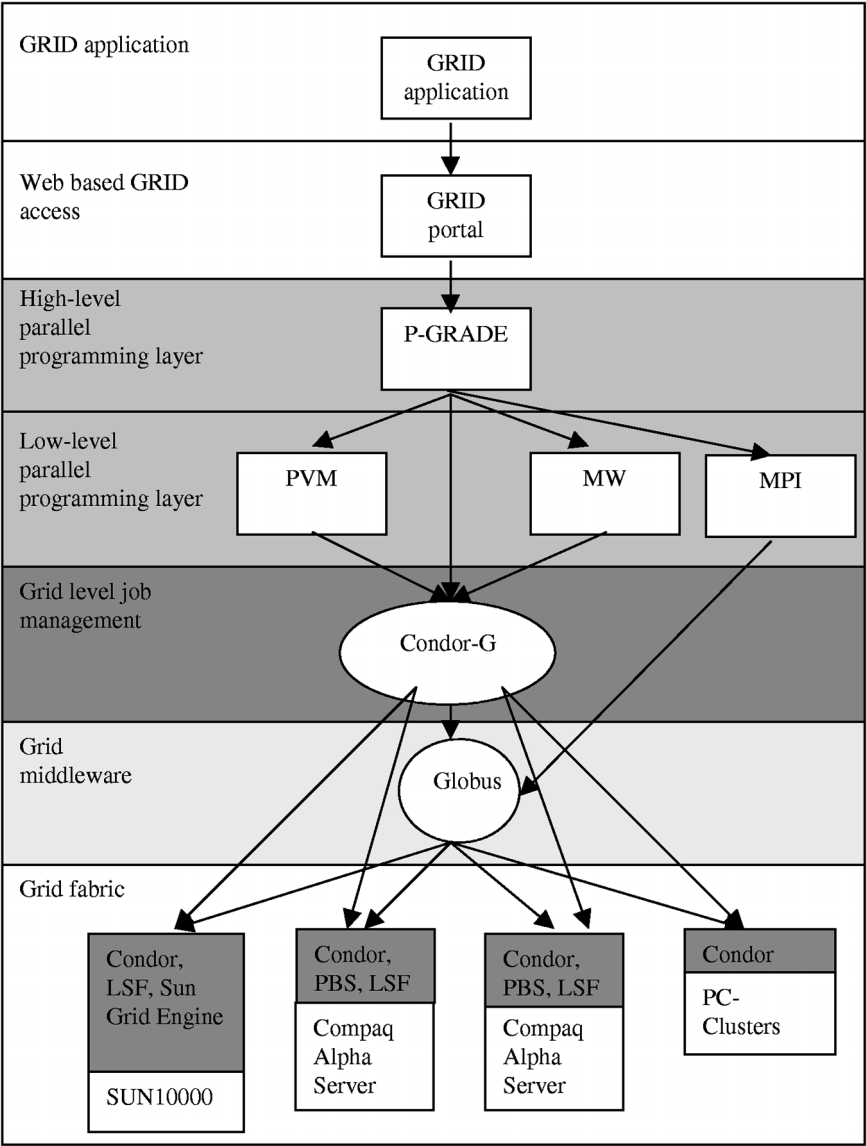


Fig. 2. Components of the Hungarian Supercomputing Grid and their relations

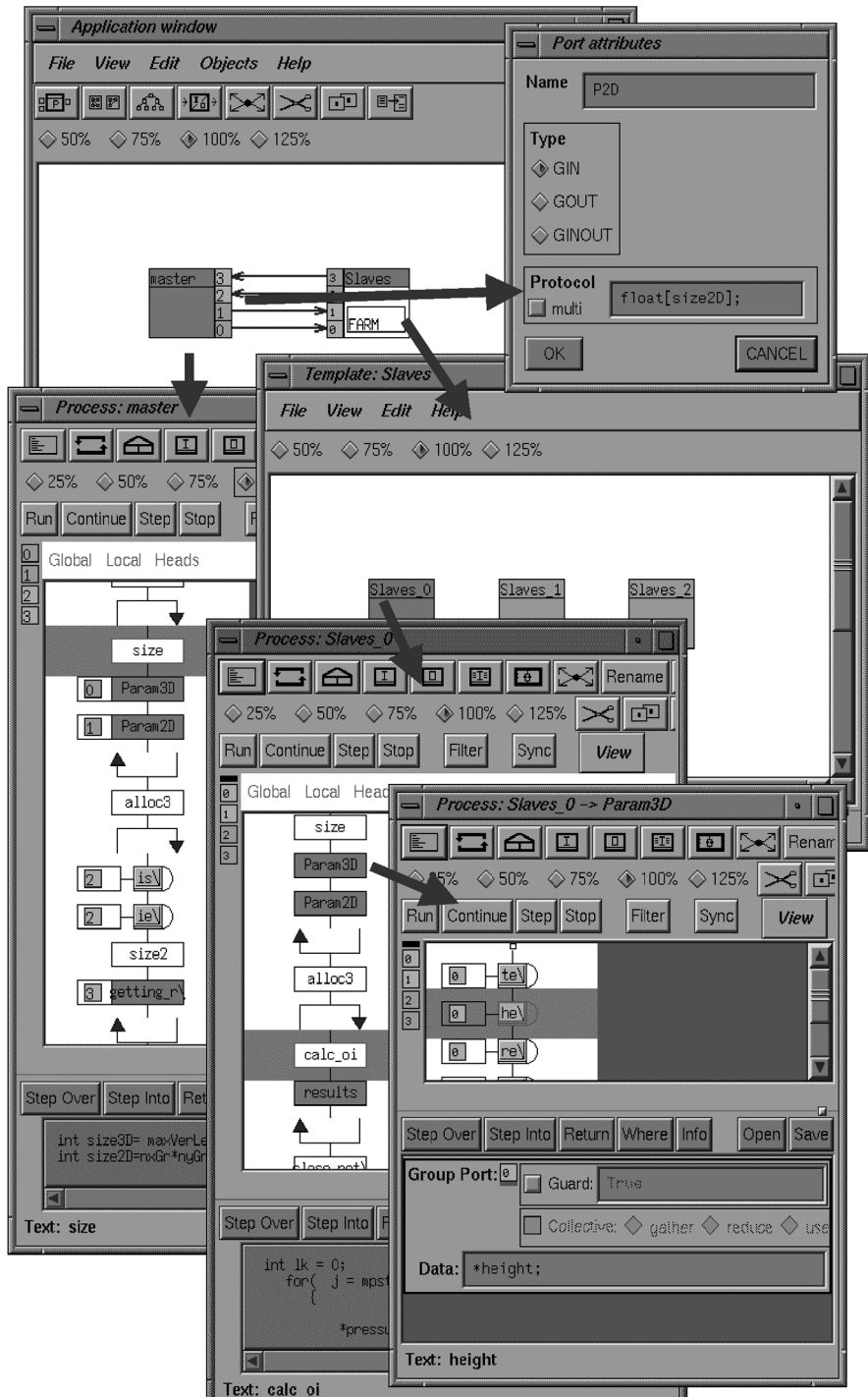


Fig. 3. Graphical parallel code of the CANARI algorithm

4 Activities in the H-SuperGrid project

The main idea of the H-SuperGrid is that the user can start parallel programs that can run in parallel on several supercomputers and/or clusters of the H-SuperGrid. The parallel programs (either PVM or MPI ones) will be developed in P-GRADE and started from P-GRADE either as a Condor job or as an immediately running interactive parallel program if the user gained direct access to several resources of the Grid. In the latter case resources can be reserved by the Grid portal.

The possible scenarios of using the H-SuperGrid are as follows:

1. Development phase: Interactive usage of reserved H-SuperGrid resources for an MPI program without using P-GRADE and Condor.
2. Development phase: Interactive usage of reserved H-SuperGrid resources for a PVM or MPI program using P-GRADE but without using Condor.
3. Production phase: Running a PVM or MPI program as a Condor job in the H-SuperGrid.

Scenario 1 is supported by MPICH-G2 [14] and Globus which are available in other Grid systems, too. The realisation of scenario 2 is one of the novel possibilities in the H-SuperGrid. It enables the developing and running parallel/distributed programs for the Grid using all the tools and facilities of P-GRADE. It will be implemented in a different way for MPI and PVM programs. In the case of MPI the solution is straightforward, it requires the generation of MPICH-G2 code from GRAPNEL programs. In the case of PVM we have to solve the problem how to distribute a parallel GRAPNEL program as a collection of several Parallel Virtual Machines running on different supercomputers and clusters of the H-SuperGrid. For such case we also have to solve the problem of dynamic process migration

a/ among the nodes of a supercomputer or cluster

b/ among the nodes of different supercomputers and clusters.

In a currently running cluster programming project we have already solved the problem of checkpointing PVM programs and their migration among the nodes of a cluster. This checkpointing and migration scheme will be extended for case b/ as well.

Scenario 3 requires the connection of P-GRADE with Condor. It means that P-GRADE will be able to directly generate Condor jobs that internally contain either MPI or PVM programs. Again the MPI case will be simpler since Condor supports the usage of MPI on top of Globus. However, this solution will not support the checkpointing and migration of GRAPNEL programs in the Grid, only their parallel execution on the supercomputers and clusters of the H-SuperGrid. By extending our PVM checkpointing and migration mechanism towards the Grid, the Condor jobs containing PVM programs will be checkpointed and migrated in the H-SuperGrid for improving fault-tolerance and load-balancing capabilities. Notice that the current Condor system cannot support the checkpoint and migration functionalities for general PVM programs. The Condor group and the H-SuperGrid project will strongly collaborate to solve these problems.

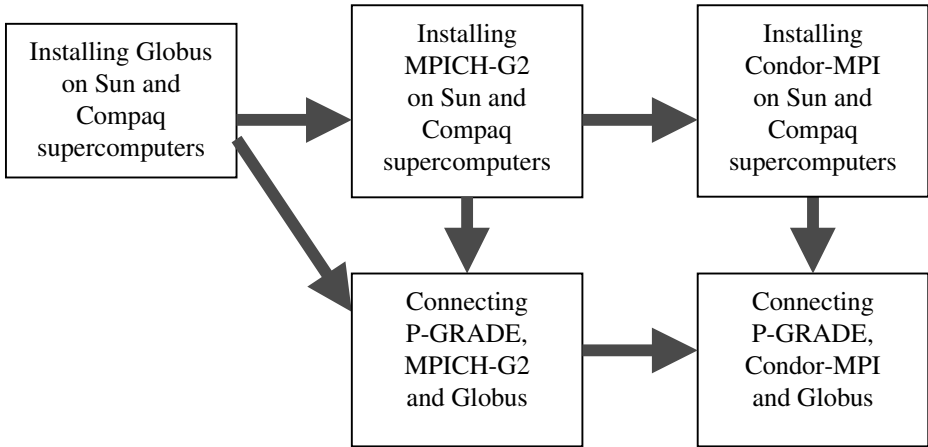


Fig. 4. Workflow diagram to support scenarios 1-3 for MPI programs

All these scenarios will be investigated and implemented in the H-SuperGrid. The workflow diagram of Figure 4 shows the steps of connecting Globus, MPICH-G2, Condor and P-GRADE together to realise scenarios 1-3 for MPI programs generated from GRAPNEL under P-GRADE. The activities to implement scenarios 2 and 3 for PVM programs generated from GRAPNEL are shown in Figure 5.

Besides combining and extending Condor and P-GRADE the other two main tasks of the project are to solve the security problems and to develop an accounting system. These issues will be addressed in a forthcoming paper.

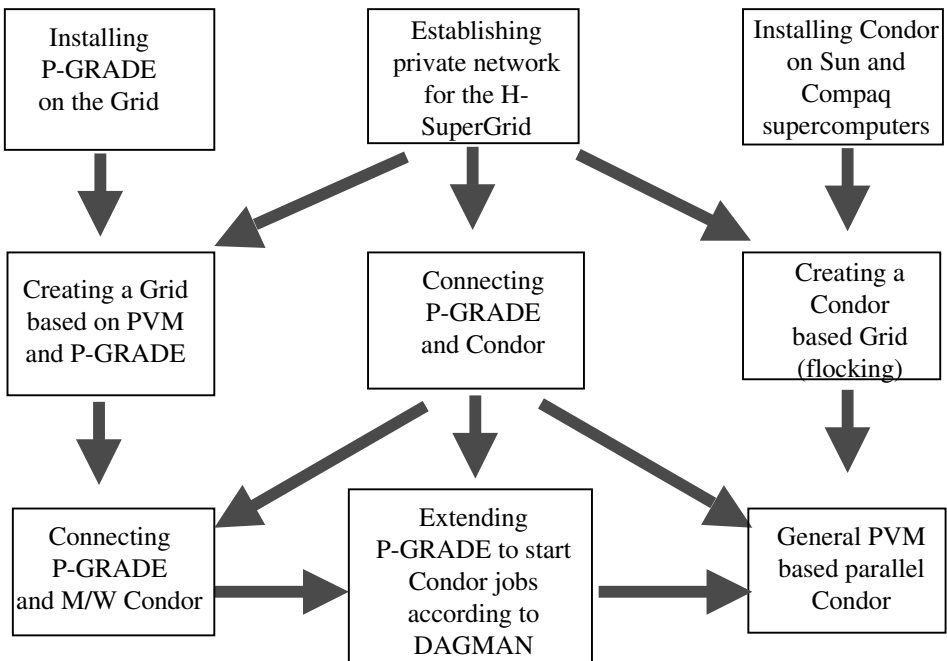


Fig. 5. Workflow diagram to support scenario 2 and 3 for PVM programs

5 Conclusions

The H-SuperGrid project started in 2002 January and will last two years. It will connect all the major supercomputers and clusters of Hungary and will enable to execute grand-challenge parallel applications in a dynamic way on all the high-performance resources of Hungary by integrating Globus, Condor, PVM, MPICH-G2 and P-GRADE technologies. The main advantages of the H-SuperGrid compared with other grid systems are as follows:

- The same parallel program development environment (P-GRADE) will be usable for supercomputers, clusters and the Grid.
- The PVM/MPI parallel programs developed under P-GRADE can run interactively both on individual supercomputers, clusters or any combination of supercomputers and clusters of the H-SuperGrid.
- The parallel PVM programs developed by P-GRADE can run as checkpointable parallel Condor jobs both on individual supercomputers, clusters or any combination of supercomputers and clusters.
- Condor jobs containing parallel PVM programs will be able to migrate among supercomputers and clusters of the H-SuperGrid.

References

1. <http://www.lpds.sztaki.hu/>
2. <http://www.datagrid.cnr.it/>
3. <http://www.cs.wisc.edu/condor/>
4. B. Codenotti and D. Laforenza, Italian Initiatives in Grid Computing, ERCIM News, No. 45, April 2001, pp. 12-13
5. D. Boyd, et al., UK e-Science Grid Programme, ERCIM News, No. 45, April 2001, pp.11-12
6. <http://www.globus.org/>
7. <http://www.lpds.sztaki.hu/>
8. <http://www.epm.ornl.gov/pvm/>
9. <http://www-unix.mcs.anl.gov/mpi/>
10. P. Kacsuk, G. Dózsa, T. Fadgyas and R. Lovas: The GRED Graphical Editor for the GRADE Parallel Program Development Environment, Journal of Future Generation Computer Systems, Vol. 15(1999), No. 3, pp. 443-452.
11. D. Drótos, G. Dózsa and P. Kacsuk: GRAPNEL to C Translation in the GRADE Environment, in P. Kacsuk, J.C.Cunha and S. Winter (eds): "Parallel Program Development: Methodology, Tools and Case Studies", Nova Science Publishers, Inc. pp. 249-263, 2001.
12. P.Kacsuk: Systematic Macrostep Debugging of Message Passing Parallel Programs, Journal of Future Generation Computer Systems, Vol. 16, No. 6, pp. 609-624, 2000.
13. P. Kacsuk: Performance Visualisation in the GRADE Parallel Programming Environment, Proc. of the HPC'Asia 2000, Peking. pp. 446-450, 2000.
14. <http://www.niu.edu/mpi/>