

PHASE and MICA: Application Specific Metacomputing*

Jörn Gehring, Alexander Reinefeld, Anke Weber

Paderborn Center for Parallel Computing (PC²)
Fürstenallee 11, D-33095 Paderborn, Germany
email: {joern, ar, weber}@uni-paderborn.de

Abstract. Metacomputing refers to the coordinated use of a pool of geographically distributed high-performance computers. The EU projects PHASE and MICA, presented in this paper, are settled in the field of application specific metacomputing combining high-performance computing expertise and industrial requirements.

1 Introduction

Distributed high-performance computing – so-called metacomputing – refers to the coordinated use of a pool of geographically distributed high-performance computers. The advantages of metacomputing are obvious: Metacomputers provide true supercomputing power at little extra cost, they allow better utilization of the available high-performance computers, and they can be easily upgraded to include the latest technology. It seems, however, that up to now no system has been built that rightfully deserves the name *metacomputer* in the above sense. The following two types of metacomputing can be distinguished: supercomputer metacomputing and workstation metacomputing

Supercomputer metacomputing is made up by a collection of parallel high-performance computers that are used in a coordinated way. The rationale is that it is easier to harness a few HPC systems than several hundreds or thousands of (unreliable) workstations. A first attempt in this direction is the Paderborn MOL project (Metacomputer Online) [3]. MOL is based on the design of versatile interfaces combining existing, well-proven MPP tools to a homogeneous environment.

In *workstation cluster metacomputing*, a collection of geographically distributed workstation clusters is used as a single computational resource. Examples of such initiatives can be found at the Dutch Polder [7] and the German Hypercomputing [8].

Many obstacles in building a metacomputer can be overcome, if we restrict its capabilities to what is needed by industrial end-users. These people want to use the computing power available in the Internet for getting problems solved with commercial applications. The projects MICA and PHASE both address the needs of these users by implementing an *Application Centered Metacomputer*. In

* This work has been supported by the EU, ESPRIT projects 23486 and 20966

application centered metacomputing, users may execute a number of predefined applications on a set of WAN-connected supercomputers. They specify job requirements (e.g., expected response time, data requirements, cost) and submit problems to an automated load-balancing scheme that distributes the parallel tasks to the best suited hardware platforms. Within the EU-funded project MICA, it was proved that the prototype of such a metacomputer can be built in less than two years. This metacomputer prototype is now being enhanced within the PHASE project.

Section 2 gives a more detailed view on MICA while PHASE will be discussed in Section 3. An overview of the current status of both projects is given in Section 4.

2 MICA – A Model for Industrial CFD Applications

The MICA project puts together existing technology in order to implement a powerful and easy-to-use CFD² -engine. By having the frontend of this engine running on low-cost personal computers and the backend on high-end supercomputers, MICA delivers the power of high-performance-computing onto the desks of small and medium sized industrial companies.

The system consists of three main components: The first component is the PHOENICS CFD code which runs on personal computers as well as on massively parallel supercomputers. The capabilities of this code have been improved significantly within the project. The second component is a graphical virtual environment editor which enables non-CFD-experts to set up complex problem descriptions and to examine the results within a powerful 3-D-viewer. These two components are connected by MICA-Net, which receives problem and description files from end-users and takes responsibility for providing the results as soon as possible and for the best achievable price.

Fig. 1 depicts the system architecture of MICA-Net. On the right there are the various users who may be distributed all over Europe. The users submit CFD-problems to the virtual entry point of MICA-Net. A virtual entry point is given by the collaboration of a set of MICA-servers. Each participating HPC center operates its server autonomously. Servers can be signed on or removed from the MICA-Net at any time. A user submits a problem to the nearest server which will then contact all other servers to find out where the best suited machine for this problem is located. For this decision the following parameters are taken into account: current load of all machines, network throughput, problem size and type, desired response time, and maximum cost.

In order to keep MICA open to new platforms, the hardware and its management system are encapsulated by *Load-Monitors* and *Job-Managers*. The Load-Monitors provide abstract load information to the MICA-servers while the Job-Managers supply routines for starting and controlling parallel jobs. Within the project, these modules have been developed for Unix, CCS [2], and CODINE [5].

² Computational Fluid Dynamics

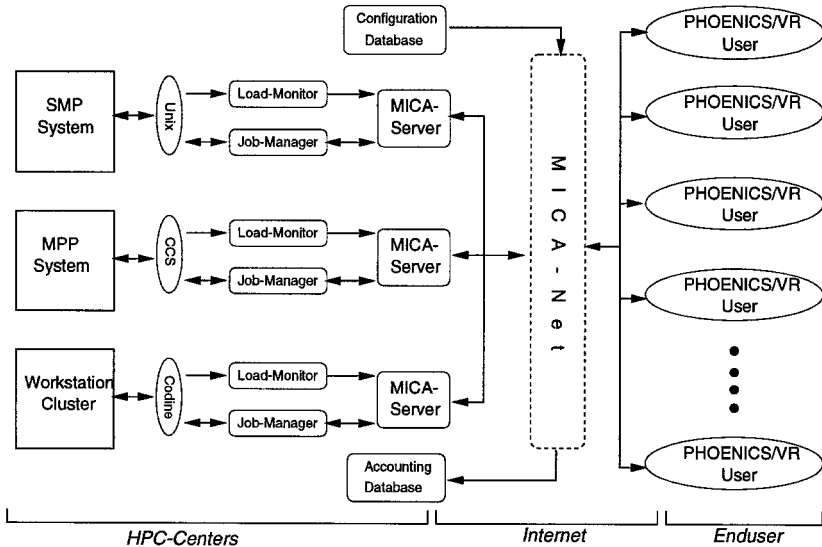


Fig. 1. System architecture of the MICA network

The system is currently limited to one application only and it does not yet support applications that are distributed among several centers. We consider it as a first step towards a metacomputer which brings the huge computing power of the Internet onto the desks of industrial end-users. This is especially true as the MICA-Net services can be seamlessly integrated into the user interfaces of the applications. Thus, it was decided to extend the MICA system within the PHASE project towards a wider range of applications and towards jobs of interacting applications.

3 PHASE – A Distributed Pharmaceutical Applications Server

Drug development in the pharmaceutical industry is a long drawn-out process that starts with the discovery of drug targets and ends in years of clinical testing. The total process can take more than a decade and costs hundreds of millions of ECUs. High-performance computing can make a major contribution to improve both speed and scope of the drug target discovery process. Work that, due to the lack of adequate computing facilities, would have been submitted in batch mode (e.g., profile searches in genome sequence databases) now becomes nearly interactive. This allows the researcher to approach a problem from many different angles. It improves iteratively the detection of similarities leading to the specification of new drug targets and the automatic analysis of large data volumes, e.g. entire genomes. In addition, simulation jobs now become realistic for the first time, e.g., the simulation of protein molecular complexes using distance geometry approaches ([4], [1]).

3.1 Bio-Informatic Tools

The ongoing genome projects identify a large number of partial and complete genes every day. The knowledge of the sequence alone does not give any clue on its function and is useless for drug discovery. Only when a function has been assigned to a sequence it becomes a potential drug target, and once the 3D structure is known it also allows for small molecule drug design, the ultimate target of the process chain. GeneQuiz, MaxHom, DRAGON and MSAP [6] all help in the identification of function, though at different levels. All four also contribute to the generation of meaningful 3D models of potential drug targets.

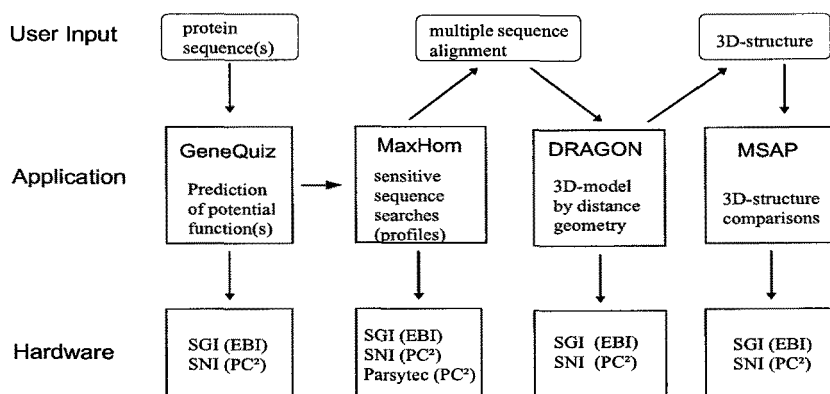


Fig. 2. Computational contribution to the drug target discovery process

Fig. 2 illustrates how the four PHASE applications build on each other to cover the complete drug target discovery process. All applications have been or will be efficiently parallelized using the portable PVM message passing library as a hardware independent programming model. MaxHom, the code with the highest computing demand has been benchmarked on the largest MPP system, a 64 processor system with more than 90% efficiency. The other codes run efficiently on workstation clusters with a moderate number of powerful computing nodes, e.g. 8 to 16 processors.

Currently, the four applications are driven by either a command-line interface or by a graphical user interface, both facilitating the integration into the server interface.

3.2 Distributed Applications Server

The distributed application server provides a central user access point to the described bio-informatic applications. The server accepts requests from the end-users via a graphical user interface and submits them to the best suited hardware system. It builds upon the MICA-Net architecture (see Fig. 1) and consists of the following main components:

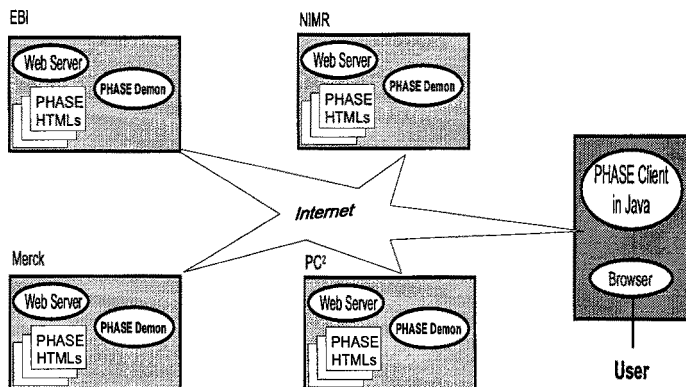


Fig. 3. Integration of PHASE-Net into the WWW

- A *Graphical User Interface* based on HTML and Java provides a user-friendly, portable entry point for incoming requests. HTML and Java are standardized WWW languages that run on virtually any hardware platform with standard web browsers.
- The *PHASE Access Layer* collects all incoming requests, checks authorization and registers them in the Accounting Database.
- A *Configuration Database* maintains a list of resources (programs and HPC systems) that are currently accessible by the PHASE server.
- A *Request Analyzer* scans incoming requests and determines their hard and software requirements.
- A *Dynamic Job Distributor* forwards the requests to the best suited platform. Its decisions are based on the system load factors and the current network throughput.
- *Distributed Load Monitor Daemons* provide information on the current system and network performance.
- *Cluster management software* like CODINE or the Computing Center Software CCS [2] will be used as an access point to the distributed resources.

Fig. 3 illustrates the planned Internet architecture for PHASE. The WWW user interface builds on GeneQuiz and Predict Protein access points, which are well-established WWW interfaces in the bio-informatics community.

3.3 Security and Confidentiality Aspects

Security issues are concerned with respect to industrial requirements and the repeated misuse of the Internet by intruders. MICA concentrates on data encryption techniques. Extending these concepts, the Internet version of PHASE is based on Java applets [9]. Applets loaded over the network are untrusted by the Web browser. Untrusted applets are run in a restricted environment and the Web browser carefully controls, what an applet is allowed to do, e.g., deleting files

on the local host, sending out emails, list directory contents, open socket connections to remote hosts, and many more. Additionally, a two-stage security system will be provided: a normal security level with standard Kerberos/Netscape security mechanisms, and a high-security level with email access to trusted servers guarded by UNIX fire walls.

4 Current Status

The distributed application server developed in MICA is ready for use. It has been presented to the consortium in a review meeting in March 1997. Current work concentrates on the seamless integration of human experts as an additional resource. The distributed application server will be adopted and extended to the specific needs of PHASE within the next months. As PHASE has started in February 1997, conceptual work regarding the WWW based user interface and security issues is going on.

References

1. Casari, G., Andrade, M.A., Bork, P., Daruvar, A., Ouzounis, C., Schneider, R., Tamanes, J., Valencia, A., Sander, C.: Challenging times for bio-informatics. *Nature* (1995), Vol 376, 24 August 1995.
2. Ramme, F., Römke, T., Kremer, K: A Distributed Computing Center Software for the Efficient Use of Parallel Computer Systems. Springer-LNCS 797, 1994, 129–136.
3. Reinefeld, A., Baraglia, R., Decker, T., Gehring, J., Laforenza, D., Ramme, F., Römke, T., Simon, J.: The MOL Project: An Open, Extensible Metacomputer. Heterogenous computing workshop HCW'97 at IPPS'97, April 1-5, Geneva.
4. Sander, C., Schneider, R., Stouten, P.: The Human Genome and High Performance Computing in Molecular Biology. Supercomputer 1992.
5. GENIAS Homepage: <http://www.genias.de/>
6. PC² project page: <http://www.uni-paderborn.de/pc2/projects/>
7. POLDER Homepage: <http://www.wins.uva.nl/projects/polder>
8. Hypercomputing Homepage:
<http://www.tec.informatik.uni-rostock.de/hypercomp>
9. Flanagan, D.: Java in a Nutshell. O'Reilly & Associates, 1996.

Acknowledgements

This work has been done in a stimulating environment. Many thanks to our MICA and PHASE project partners, the EU project officer Massimo Lucioli and the EU reviewers.