Proposal for cooperation in research on Grid application and infrastucture

Zoltán Horváth,
PhD in Informatics, associate prof.[*]
and
Ildikó László, PhD in Physics, assistant professor

Institute of Informatics
Eötvös L. University (ELTE), Budapest, Hungary
E-mail: hz@inf.elte.hu

1 Introduction

Our institute is involved in several Grid research projects (DemoGrid, JiniGrid) in Hungary with international cooperation with CERN and other European partners. We are open for cooperation with further institutes. A full professor, 2 associate professors, sevearal assistant professors and PhD and MSc. students are involved. The group has 15 years research background in design and implementation of parallel and distributed programs.

2 Main topics of our research

One of our primary concern of research activity is design methodology of parallel and distributed programs. We developed a formal model for specification and verification of parallel an distributed programs in a Unity-like way [2, 4]. Our main concern of recent research is how to design and implement effeciently algorithms for PC clusters and for Grid. We have investigated the applicability of PC clusters with terabyte disk-servers for data-intensive parallel computing. For example we used parallel elementwise processing as our testcase [3].We have searched for the optimal value for parameters of the algorithm running on our hardware environment.

We investigate the suitability of programming language elements (especially elements of functional programming languages) and communication libraries. The performance of several communication frameworks has been tested by us, such as C/PVM, C/MPI, Distributed Haskell and socket interface in C. On large inputs with heavy operations our implementations showed considerable speedups.

We investigated the applicability of the Object Request Broker (ORB) midllewares in Grid environment. We develooped a flexible approach of extensible

[*] Project leader of WP in DemoGrid (2001-2002) and JiniGRid (2003-2004)

ORB. Our model of Pluggable Semantic Elements (PSE) allows the user to implement and arbritarily combine the well-defined functional components of the invocation semantics.

We are interested in the structure of the Grid middleware (EDG, Globus, etc.). Applications on the Grid need different types of resources. Such resources are CPU, memory, network bandwidth and secondary storage. The grid middleware tries to allocate resources to applications in a way that results can be computed as fast and as efficiently as possible.

Our main focus is on the secondary storage, on data access. Many applications read and write large files[6]. Often these files are not available on the computing element where the application runs. An optimal strategy for data access may depend on the kind of the application.

We introduced the concept of data access patterns. The description of the data access pattern of an application should provide sufficient amount and quality of information for the resource broker and the replica manager to determine an optimal data access strategy for the application.

We have examined those kinds of information that can make up data access patterns and proposed a resource consumption specification language that can form the basis of an extension to EDG JDL. We have also proposed a framework for separating pure computational and distributed resource access aspects of an application. In this framework an extensible library of grid interaction patterns can be developed.

We suggest data access patterns be either specified by a programmer or, in case of parameter-sweep applications, determined by a monitoring grid middleware tool. We have investigated several ways to add monitoring of data access to an application.

We are interested in the application of mobile code in Grid environment. We proposed the Certified Proven Property Carrying Code architecture [5], which is related to PCC (Proof Carrying Code). When an application or the middleware allows to run a piece of code received via the net, the application or the middleware may want to get guarantees that this piece of code does not harm: does not use too much resources, does not read or modify data unauthorised, etc. PCC and CPPCC [7] are techniques for providing such assurances. With PCC and CPPCC, the code consumer specifies safety requirements, which specifies the access conditions of parts of memory and/or how much of a resource is allowed to use. The code producer must also provide a set of program properties and an encoded proof packed together with the code transmitted, to certify that the code really satisfies these conditions.

References

[1] Attila Ulbert, Custom Invocation semantics and Corpuses in Data Multicasting, ECOOP 2002, PhD Workshop, Málaga, Spain, June 2002.

[Ulb02a] Ulbert A.: Semantic Elements for Parallel Computing in ORB(M). In: Distributed and Parallel Systems, Cluster and Grid Computing, Kluwer, 2002. pp. 133-140.

[2] Chandy, K.M., Misra, J.: *Parallel program design: a foundation*, Addison-Wesley, 1989.

[3] Horváth Z., Hernyák Z., Kozsik T., Tejfel M., Ulbert A.: A Data Intensive Computation on a Cluster. In: Distributed and Parallel Systems, Cluster and Grid Computing, Kluwer, 2002. pp. 46-53.

[4] Horváth Z.: A Relational Model of Parallel Programs (in Hungarian). PhD Thesis. Phd School in Informatics, University Eötvös Loránd, Budapest, Hungary, 1996. (120 pages)

[5] Daxkobler K.-Horváth Z.-Kozsik T.: Prototype of CPPCC - Safe Functional Mobile Code in Clean. Proceedings of *Implementation of Functional Languages'02*, Madrid, Spain, Sept. 15-19, 2002. pp. 320-329.

[6] Horváth Z., Kozsik T., Ulbert A.: Towards a better optimisation of resource allocation in Grid. Technical Report. 2002.

[7] Necula, G.: Proof-carrying code. In *Proceedings of the ACM Symposium on Principles of Programming Languages*, 1997.