

Programming High Performance Applications in Grid Environments

Domenico Laforenza
Advanced Computing Department
CNUCE-Institute of the Italian National Research Council
via Vittorio Alfieri, 1, I-56010 Ghezzano, Pisa, Italy
Domenico.Laforenza@cnuce.cnr.it

The need for realistic simulations of complex systems relevant to the modeling of several modern technologies and environmental phenomena increasingly stimulates the development of advanced computing approaches.

Nowadays it is possible to cluster or couple a wide variety of resources including supercomputers, storage systems, data sources, and special classes of devices distributed geographically and use them as a single unified resource, thus forming what is popularly known as a "computational grid" [1, 2].

Grid Computing enables the development of large scientific applications on an unprecedented scale. Grid-aware applications (meta-applications, multi-disciplinary applications) make use of coupled computational resources that cannot be replicated at a single site. In this light, grids let scientists solve larger or new problems by pooling together resources that could not be coupled easily before. Designing and implementing grid-aware applications often require interdisciplinary collaborations involving aspects of scientific computing, visualization, and data management [3].

Multi-disciplinary applications are typically composed of large and complex components, and some of them are characterized by huge high performance requirements [4, 5]. In order to get better performance, the challenge is to map each component onto the best candidate computational resource having a high degree of affinity with the software component. This kind of mapping is a non-trivial task. Moreover, it is well known that, in general, the programmer's productivity in designing and implementing efficient parallel applications on high performance computers remains a very time-consuming task.

Grid computing makes the situation worse as heterogeneous computing environments are combined so that the programmer must manage an enormous amount of details. Consequently, the development of grid programming environments that would enable programmers to efficiently exploit this technology is an important and hot research issue. A grid programming environment should include interfaces, APIs, utilities and tools so as to provide a rich development environment. Common scientific languages such as C, C++, Java and Fortran should be available, as should application-level interfaces like MPI and PVM. A range of programming paradigms should be supported, such as message passing and distributed shared memory. In addition, a suite of numerical and other commonly used libraries should be available.

Today, an interesting discussion is opened about the need to think at new abstract programming models and develop novel programming techniques addressing specifically the grid, which would deal with the heterogeneity and distributed computing aspects of grid programming.

In this talk, after an introduction on the main grid programming issues, an overview of the most important approaches/projects conducted in this field worldwide will be presented. In particular, the speaker's contribution in designing

some grid extension for a new programming environment will be shown. This work constitutes a joint effort conducted by some academic and industrial Italian partners, in particular the Department of Computer Science of the Pisa University and CNUCE-CNR, in the framework of the ASI-PQE2000 National Project aimed at building ASSIST (A Software development System based on Integrated Skeleton Technology) [6, 7, 8, 9].

The main target for the ASSIST Team is to build of a new programming environment for the development of high performance applications, based on the integration of the structured parallel programming model and the objects (components) model. In this way, ASSIST should be available for a wide range of hardware platforms from the homogeneous parallel computers (MPP, SMP, CoWs) to the heterogeneous ones (Grids) [10].

References

1. I. Foster, C. Kesselman (eds), *The Grid: Blueprint for a future computing infrastructure*, Morgan Kaufmann, 1999
2. M.A. Baker, R. Buyya, and D. Laforenza, *The Grid: International Efforts in Global Computing* Proceedings of the International Conference on Computer & eBusiness, Scuola Superiore Reiss Romoli, L'Aquila, Italy, July 31.2000 – August 6, 2000
3. K. Keahey, P. Beckman, and J. Ahrens, *Ligarure: Component Architecture for High Performance Applications* The International Journal of High Performance Computing Applications, Volume 14, No 4, Winter 2000, pp. 347-356
4. R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski, *Toward a common component architecture for high performance scientific computing*. In Proceedings of the 8th High Performance Distributed Computing (HPDC'99), 1999
5. C. René, T. Priol, MPI Code Encapsulation using Parallel CORBA Object., Proceedings of the Eighth IEEE International Symposium on High Performance Distributed Computing, IEEE, pages 3--10, August 1999
6. J. Darlington et al, *Parallel programming using skeleton functions*, LNCS 694, 1995, Springer-Verlag
7. S. Newhouse, A. Mayer, and John Darlington, *A Software Architecture for HPC Grid Applications*, A. Bode et al. (Eds.): Euro-Par 2000, LNCS 1900, pp. 686-689, 2000, Springer-Verlag
8. M. Vanneschi, *PQE2000: HPC tools for industrial applications* IEEE Concurrency, Oct.-Dec. 1998
9. B. Bacci, M. Danelutto, S. Pelagatti, and M. Vanneschi, *SkIE: a heterogeneous environment for HPC applications* Parallel Computing, Volume 25, pages 1827--1852, December 1999
10. Gruppo di Lavoro Progetto ASI-PQE2000, Sviluppo di applicazioni di Osservazioni della Terra mediante sistemi di calcolo ad Elevate Prestazioni, Settembre 1999 (Project Description, in Italian)