



Porting Applications to Grids and Clouds

Wolfgang Gentsch, Duke University, USA

ABSTRACT

A Grid enables remote, secure access to a set of distributed, networked computing and data resources. Clouds are a natural next step of Grids towards the provisioning of computing as a service. To “Grid-enable” applications, users have to cope with: complexity of grid architectures; different compute and data nodes; wide spectrum of grid middleware tools and services; the e-science application architectures, algorithms and programs. Therefore, the aim of this chapter is to guide users through the important stages of implementing applications on Grid and Cloud infrastructures, together with a discussion of important challenges and their potential solutions. As a case study, we present the DEISA Distributed European Infrastructure for Supercomputing Applications and describe the DEISA Extreme Computing Initiative DECI for porting and running scientific grand challenge applications on the DEISA Grid. This chapter concludes with an outlook on Compute Clouds, and the top ten rules of building a sustainable Grid.

Keywords: cloud computing; data management; distributed systems; grid applications; grid computing; grid portals; grid resources

INTRODUCTION

Over the last 40 years, the history of computing is deeply marked by the application developers who continuously port and optimize their application codes to the latest and greatest computing architectures and environments. After the von-Neumann mainframe, there was the vector computer, then the shared-memory parallel computer, the distributed-memory parallel computer, the very-long-instruction word computer, the workstation cluster, the meta-computer, and the Grid (never mind, it continues with SOA, Cloud, Virtualization, Many-core, and so on). There is no easy solution to this,

and the real solution would be a separation of concerns between discipline-specific content, domain-independent software and hardware infrastructure. However, this often comes along with a loss of performance stemming from the overhead of the infrastructure layers. Recently, users and developers face another wave of complex computing infrastructures: the Grid.

Let's start with answering the question: What is a Grid? Foster and Kesselman (1998) attempted the following definition: A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. In a

subsequent article (Foster, 2002), this definition was changed to include social and policy issues, stating that Grid computing is concerned with coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. The key concept is the ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose. Furthermore, the sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization. This author's concern, from the beginning (Gentzsch, 2002), was that the new definition seemed very ambitious, and as history has proven, many of the Grid projects with a focus on these ambitious objectives did not lead to a sustainable grid production environment, so far. We can only repeat that the simpler the grid infrastructure, the easier it is to use with better chance for success. It is for this reason that currently the so-called Clouds are becoming more and more popular (Amazon, 2007).

Over the last ten years, hundreds of applications in science, industry and enterprises have been ported to Grid infrastructures, mostly prototypes in the early definition of Foster and Kesselman (1998). Each application is unique in that it solves a specific problem, based on modeling, for example, a specific phenomenon in nature (physics, chemistry, biology, etc.), presented as a mathematical formula together with appropriate initial and boundary conditions. Applications could be represented by its discrete analogue using sophisticated numerical methods, translated into a programming language computers can understand, adjusted to the

underlying computer architecture, embedded in a workflow, and accessible remotely by the user through a secure, transparent and application-specific portal. These examples summarize the wide spectrum and complexity we face in problem solving on grid infrastructures.

The user (and especially the developer) faces several layers of complexity when porting applications to a computing environment, especially to a compute or data grid of distributed networked nodes ranging from desktops to supercomputers. These nodes, usually, consist of several to many loosely or tightly coupled processors (frequently, these processors contain few to many cores). To run efficiently on such systems, applications have to be adjusted to the different layers, taking into account different levels of granularity, from fine-grain structures deploying multi-core architectures at processor level to the coarse granularity found in application workflows representing for example multi-physics applications. Even, the user has to take into account the specific requirements of the grid, coming from the different components of the grid services architecture, such as security, resource management, information services and data management.

Obviously, in this article, it seems impossible to present and discuss the complete spectrum of applications and their adaptation and implementation on Grids. Therefore, we restrict ourselves in the following to briefly describe the different application classes, present a checklist (or classification) with respect to grouping applications according to their appropriate grid-enabling strategy. Also, for lack of space here, we will not discuss the mental, social, or legal aspects which sometimes might be the knock-out criteria for running applications on a grid. Other show-stoppers such as sensitive data, security concerns, licensing issues, and intellectual property, were discussed in some detail in Gentzsch (2007a).

In the following, we will consider the three main areas of impact on porting applications to grids: infrastructure, data management and application architecture issues. These issues can have an impact on effort and success of

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/porting-applications-grids-clouds/2168

Related Content

Foundations of Rough Sets from Vagueness Perspective

Piotr Wasilewski and Dominik Slezak (2008). *Rough Computing: Theories, Technologies and Applications* (pp. 1-37).

www.irma-international.org/chapter/foundations-rough-sets-vagueness-perspective/28466

Tag-Based Classification for Software-Defined Networking

Hamid Farhady and Akihiro Nakao (2015). *International Journal of Grid and High Performance Computing* (pp. 1-14).

www.irma-international.org/article/tag-based-classification-for-software-defined-networking/128357

Competitive Processors Allocation in 2D Mesh Connected Multicomputer Networks: A Dynamic Game Approach

Ayoub Alsarhan, Emad Eddien Awad Abdallah and Ashraf H. Aljammal (2017). *International Journal of Grid and High Performance Computing* (pp. 53-69).

www.irma-international.org/article/competitive-processors-allocation-in-2d-mesh-connected-multicomputer-networks/182341

Keys for Administration of Reconfigurable NoC: Self-Adaptive Network Interface Case Study

Rachid Dafali and Jean-Philippe Diguët (2010). *Dynamic Reconfigurable Network-on-Chip Design: Innovations for Computational Processing and Communication* (pp. 67-83).

www.irma-international.org/chapter/keys-administration-reconfigurable-noc/44221

Toward a Quality-of-Service Framework for Peer-to-Peer Applications

Ankur Gupta and Lalit K. Awasthi (2010). *International Journal of Distributed Systems and Technologies* (pp. 1-23).

www.irma-international.org/article/toward-quality-service-framework-peer/46047