IGI PUBLISHING ITJ 4536

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-global.com

This paper appears in the publication, International Journal of Grid and High Performance Computing, Volume 1, Issue 1
edited by **Emmanuel Udoh © 2008, IGI Global**

Speculative Scheduling of Parameter Sweep Applications Using Job Behavior Descriptions

Attila Ulbert, Eötvös Loránd University, Hungary

László Csaba Lőrincz, Eötvös Loránd University, Hungary

Tamás Kozsik, Eötvös Loránd University, Hungary

Zoltán Horváth, Eötvös Loránd University, Hungary

ABSTRACT

The execution of data intensive Grid applications still raises several questions regarding job scheduling, data migration and replication. This article presents new scheduling algorithms using complex job behavior descriptions that allow estimating job completion times more precisely thus improving scheduling decisions. Three approaches of using complex, re-fined job descriptions are discussed: a) single job description, b) multiple job descriptions, c) multiple job descriptions with mutation. The proposed Grid middleware components (1) monitor the execution of jobs and gather resource access information, (2) analyze the compiled information and generate a description of the behavior of the job, (3) refine the already existing job description, and (4) use the refined behavior description to schedule the submitted jobs.

Keywords: *clusters; data grids; execution time prediction; job scheduling; parameter-sweep applications*

INTRODUCTION

Resource management is one of the major tasks of Grid middleware. Resources include available computing power (i.e. CPUs), memory and secondary storage. The strategies implemented by the middleware fundamentally determine how early a job can finish its execution and provide the desired computing results. For data intensive parameter sweep applications the placement of data onto Storage Elements (SEs) and the selection of Computing Elements (CEs) have substantial impact on their completion time, therefore the combined efficiency of resource management and scheduling strategies significantly determine the performance of Grid.

The resource management and scheduling algorithms may take into account the current state of the Grid, or statistics collected on the performance of the Grid components and applications. Some of the resource management strategies make use of sophisticated economy-based decision algorithms (Bell, Cameron, Carvajal-Schiaffino, Millar, Stockinger and Zini, 2003), others focus chiefly on data replication, and present replica management Grid middleware (Laure, Stockinger and Stockinger, 2005). Scheduling algorithms may apply statistical prediction methods (Gao, Rong and Huang, 2005; Nabrizyski, Schopf and Weglarz, 2003), which can be used to rank the CEs by the estimated job completion time and select the target optimal CE.

Our resource management and scheduling approach is based on the realization that the completion time of a job on a CE can be determined exactly right after the given job is terminated. Furthermore, we could make perfect scheduling decisions if we were able to run the job on all possible CEs of the Grid one by one within the same circumstances, register the finishing times and run the job on the "best" CE. Obviously, such perfect decisions are not possible to be made, and we can only mimic the process of the selection of the best CE (Lőrincz, Kozsik, Ulbert and Horváth, 2005).

In order to predict the completion time of the job the proposed scheduling strategies need to know the state of the Grid, the characteristics of the CEs and the expected resource access patterns of the job. For each job, the proposed Grid middleware services (1) monitor the execution of the job and gather resource access information, (2) generate a compact description of the job's behavior, (3) use the job behavior description to calculate the job's possible termination times and schedule the job accordingly, and (4) re-fine the already existing behavior description using the behavior description reflecting its latest execution.

Our proposed scheduling strategies also take into consideration the effects of data replication and provide replication commands harmonizing with the actual scheduling decision. For example, if the job accesses large chunks of data, it is most likely a good idea to schedule it to the Computing Element (or to a location in its neighborhood) where the input files are available. However, if the job had to wait too long before it could be started on the chosen Computing Element, then it would be worth copying the input files to another Grid component where the job can be executed earlier. In the case of jobs that are less data intensive (use less and smaller input files), the nearness of the files is not so important since the cost of the replication is very low. Furthermore, knowing the resource access patterns of the job the files can be replicated parallel to the execution of the job by fetching the necessary file fragments just-in-time.

The rest of this article is organized as follows. Section 2 reviews the related work. Section 3 gives an overview of the proposed architecture. In Section 4 the description of a job's behavior is discussed in details. Section 5 presents our algorithms for automatically generating the job behavior descriptions. The proposed scheduling strategies are presented in Section 6. Section 7 gives an insight into the implementation of the proposed components. Simulation experiments are presented in Section 8. Finally, section 9 summarizes the contribution of this article and 10 hints of our future work.

RELATED WORK

Our approach focuses on the jobs' resource access, the scheduling decisions made based on the finishing time estimations that exploit the knowledge of the jobs' behavior.

Nabrizyski et al. (2003) gives an excellent overview of Grid resource management. Besides presenting a number of scheduling strategies (Ranganathan and Ian, 2003), introduces new statistical prediction techniques for the execution times for applications. The first technique uses historical information of previous similar runs to form predictions. The similarities of runs are determined by categorizing discrete characteristics of the submitted jobs. The second technique uses instance-based learning: a database of experiences is maintained and used to make predictions. Each experience consists

Related Content

Collaborative Web-Based System for Knowledge Transfer to Distributed Groups of Users Within Strategic Noise Mapping Domain

Marcin Dbrowski (2013). *International Journal of Distributed Systems and Technologies (pp. 39-49).*

www.irma-international.org/article/collaborative-web-based-system-for-knowledge-transfer-to-distributed-groups-of-users-within-strategic-noise-mapping-domain/104717

A Mathematical Analysis of a Disaster Management Data-Grid Push Service

Nik Bessis, Antony Brownand Eleana Asimakopoulou (2012). *Technology Integration Advancements in Distributed Systems and Computing (pp. 268-282).*

www.irma-international.org/chapter/mathematical-analysis-disaster-management-data/64453

Granular Computing Based Data Mining in the Views of Rough Set and Fuzzy Set

Guoyin Wang, Jun Hu, Qinghua Zhang, Xianquan Liuand Jiaqing Zhou (2010). *Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation (pp. 148-178).*

www.irma-international.org/chapter/granular-computing-based-data-mining/44703

Dynamic Reconfigurable NoC (DRNoC) Architecture: Application to Fast NoC Emulation

Yana E. Krasteva, Eduardo de la Torreand Teresa Riesgo (2010). *Dynamic Reconfigurable Network-on-Chip Design: Innovations for Computational Processing and Communication (pp. 220-255).*

www.irma-international.org/chapter/dynamic-reconfigurable-noc-drnoc-architecture/44227

G2G: A Meta-Grid Framework for the Convergence of P2P and Grids

Wu-Chun Chung, Chin-Jung Hsu, Yi-Hsiang Lin, Kuan-Chou Laiand Yeh-Ching Chung (2012). *Evolving Developments in Grid and Cloud Computing: Advancing Research (pp. 239-254).*

www.irma-international.org/chapter/g2g-meta-grid-framework-convergence/61995