Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 9 (2012) 1713 – 1722

Procedia Computer Science

International Conference on Computational Science, ICCS 2012

Impact of urgent computing on resource management policies, schedules and resources utilization

Krzysztof Kurowski[a,*], Ariel Oleksiak[a], Wojciech Piatek[a], Jan Węglarz[a,b]

[a] *Poznan Supercomputing and Networking Center, Poznan, Poland*
[b] *Institute of Computer Science, Poznan University of Technology, Poland*

Abstract

In general resource management is often defined as a process of identifying application requirements, matching compute resources to jobs, allocating those resources, scheduling and monitoring them over time to run jobs as efficiently as possible. In case of urgent computing, additionally, in this process we have to deal with a set of specific event-driven or time-critical application requirements that must be satisfied by a resource provider to guarantee the immediate access to compute resources. There is no question that capabilities supporting urgent computing are required in distributed high performance and high throughput computing infrastructures like Grids. Nevertheless, it is difficult to predict if and how urgent jobs will disturb scheduling policies as well as coexist with less critical waiting and running jobs submitted by other end-users. To date many approaches have been proposed to deal with urgent computing, for instance based on preemption or high job priorities and recently based on advance reservations. However, there is a lack of comprehensive analysis of the impact of urgent computing on existing schedulers and their efficiency. Moreover, many resource providers use a single evaluation criterion - utilization or average load of compute resources. In fact the utilization of compute resources as a metric is in contrast to other evaluation criteria relevant for urgent jobs, e.g. waiting time, response time, or mean flow time. These additional criteria in our opinion can be treated as good metrics for urgent computing taking into account the end-users perspective rather than the resource provider centric view. As we demonstrate in this paper, the problem is not trivial and it requires workload analyses. Finally, in this paper we successfully used GSSIM simulator to run various experiments and benchmarks for different scheduling strategies where urgent jobs were additionally considered.

Keywords: urgent computing, on-demand computing, parallel job scheduling, hierarchical scheduling, HPC, Grids

1. Introduction

Large-scale, time-critical and event-driven application scenarios have been pushing demands for new capabilities available in distributed computing infrastructures to deal with urgent computing. In a nutshell, end-users submitting urgent jobs to remote compute resources do not expect that their jobs will waste time waiting in batch queues until compute resources become available. On the other hand, batch queues defined within local schedulers are core mechanisms for resource providers to manage and share compute resources among multiple end-users and their jobs. The

*Corresponding author
Email address: krzysztof.kurowski@man.poznan.pl (Krzysztof Kurowski)

1877-0509
doi:10.1016/j.procs.2012.04.189

problem is even more challenging in Grids linking together large collections of geographically distributed resource providers that usually span several organizations to share a variety of resources dynamically, depending on their availability, capability, end-users requirements, and any other predefined rules set by local resources owners. Thus, such computing environments could consist of independent resource providers, where each of them may imply different resource management policies and rules for responding to urgent computation. One should also note that most of Grids are organized hierarchically - that is, there is at least one grid scheduler or broker assigning jobs to a local schedulers without being able to fully control what happens next at the local level within a local scheduler. Taking into account the complexity of a two-level hierarchical structure of schedulers together with variety of grid middleware as well as constraints based on resources utilization, different levels of requested urgencies, end-users behavior and acceptance it is difficult or even impossible to come up with one unified resource management strategy for urgent computing.

Additionally, according to analyses of different workloads collected in production HPC systems or clusters, local schedulers are often configured to use a single evaluation criterion forced by a local resource provider - utilization of compute resources. As the utilization of compute resource criterion is in fact contradictory to other evaluation criteria relevant for end-users submitting urgent jobs, e.g. waiting time or a number of canceled jobs, the problem is even more difficult as it is multi-criteria by nature. On the other hand, in many production setups some end-users may have more influence on what compute resources are provided locally and how they are used than more objective measures of performance or hardware costs. We should also emphasize dynamic characteristics of end-users behaviors and job patterns. Preferences of end-users may vary in time and are correlated with resource requirements of submitted jobs. For example, small interactive or parameter sweep type jobs require often the immediate access to resources, whereas end-users submitting large-scale and long calculations can tolerate delays. Consequently, in our opinion both end-users and resource providers should be treated as stakeholders within the resource management process, in particular in case of urgent computing. Then, the resource management process should led to the improvement of all stakeholders satisfaction that can be measured and aggregated by multiple criteria.

Therefore, in this paper we show how multi-criteria approaches can be applied to validate the impact of different scheduling policies and their hierarchical configurations on urgent computing. We show how to model different and hierarchical scheduling setups in Grids where urgent jobs may appear irregularly. Then, we discuss achieved results in the light of key evaluation criteria relevant for urgent jobs and their users, in particular resources utilization vs. waiting time of both typical and urgent jobs. Moreover, based on analysis and performed experiments using real workloads we compare commonly used scheduling strategies and show if and how urgent jobs will disturb scheduling policies as well as coexist with less critical waiting and running jobs submitted by other end-users.

The rest of this paper is organized as follows. In Section 2 we introduce leading grid initiatives together with references to main capabilities introduced so far to deal with urgent computing. Section 3 presents briefly selected real workloads for basic analyses of job patterns as well as end-users preferences. The considered resource management problem is formulated in Section 4 together with the main assumptions. Section 5 describes a set of experiments we have performed to measure the impact of urgent jobs on resource management using high job priorities and advance reservations in local schedulers. Conclusions and future work are presented in Section 6.

2. Related work

Grid computing has become a popular way of providing distributed high performance, high throughput and parallel computing for advanced science and engineering based on interconnected networks of supercomputing centers, campuses and emerging service-oriented, Internet-based technologies. In other words, the Internet provides uniform access to World Wide Web resources, and Grids or recently Clouds provide their users with an access to all kinds of compute resources in a dynamic, geographically distributed fashion. Existing grid environments, such as PL-Grid in Poland [4], TeraGrid [6] in the United States, LHC/CERN in Switzerland [1], NorduGrid in Sweden [12], Grid5000 in France [5] to cite a few of them, provide thousands of compute resources, and offer similar or even better facilities for end-users when compared to supercomputers. Unfortunately, many of existing Grids offer relatively simple remote job submission, brokering and monitoring capabilities, whereas others support advanced capabilities that can be used for urgent computing, such as high job priorities or advance reservation. A set of interesting application scenarios and new requirements for capabilities that should be available for end-users have been collected in [26].

Probably the first complete high job priorities strategy for urgent computing has been proposed and prototyped in [2]. Then, a new system known as Special PRiority and Urgent Computing Environment (SPRUCE) framework was