Future Generation Computer Systems 26 (2010) 217–227

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Grids with multiple batch systems for performance enhancement of multi-component and parameter sweep parallel applications☆

Sivagama Sundari M. [a], Sathish S. Vadhiyar [a,*], Ravi S. Nanjundiah [b]

[a] Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India
[b] Centre for Atmospheric & Oceanic Sciences, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Article history:
Received 15 August 2008
Received in revised form
6 August 2009
Accepted 7 August 2009
Available online 15 August 2009

Keywords:
Grids
Batch systems
Multi-component applications
Parameter sweep applications
Queue waiting times

ABSTRACT

In this work, we evaluate the benefits of using Grids with multiple batch systems to improve the performance of multi-component and parameter sweep parallel applications by reduction in queue waiting times. Using different job traces of different loads, job distributions and queue waiting times corresponding to three different queuing policies (FCFS, conservative and EASY backfilling), we conducted a large number of experiments using simulators of two important classes of applications. The first simulator models Community Climate System Model (CCSM), a prominent multi-component application and the second simulator models parameter sweep applications. We compare the performance of the applications when executed on multiple batch systems and on a single batch system for different system and application configurations. We show that there are a large number of configurations for which application execution using multiple batch systems can give improved performance over execution on a single system.

1. Introduction

Computational grids have been increasingly used for executing large scale scientific applications [1–11]. Most of the benefits of using grids are primarily due to the increase in the number of processors used for execution. In this work, we focus on yet another potential use of grids where the total processor space used for application execution is not increased. Specifically, we deal with grids with multiple batch systems (for brevity, we will refer to such grids as *batch grids*) and show that employing multiple batch systems can improve the response times of parallel applications than when using a single batch system.

Parallel batch systems provide space sharing of available processors among multiple parallel applications or jobs. These batch systems employ queues in which the incoming parallel applications are queued before allocation by a batch scheduler to a set of processors for execution. Thus a batch system is associated with a set of queues and a scheduling policy that selects a job from the queue and maps it to a set of processors. An application submitted to a batch system incurs additional time for waiting in a queue before actual execution. The overall response time of an application is the sum of its queue waiting time and execution time. Application with small processor requirements can be backfilled to the available processors and hence incur smaller queue waiting times than applications with large processor requirements. Thus the queue waiting times for applications are proportional to their processor requirements as illustrated in Fig. 1. The figure shows the average queue wait times for jobs with different processor requirements on an IBM SP2 system in SDSC (San Diego Supercomputer Center). The job traces were obtained from the logs maintained by Feitelson[12].

Consider a parallel application J needing P processors that consists of or can be decomposed into multiple sub-parallel applications or components, $J_1, J_2, \ldots, J_n$ with processor requirements, $P_1, P_2, \ldots, P_n$, respectively, such that $P = P_1 + P_2 + \cdots + P_n$. In this case, simultaneous submission of the multiple sub-parallel applications, $J_1, J_2, \ldots, J_n$, with small processor requirements to multiple batch systems of a batch grid can result in improved response times of the application over submitting the entire parallel application, J, with a large processor requirement to a single batch system. This is because the maximum of the queue waiting times in the former case can be lesser than the single queue waiting time in the latter case.

While this advantage of simultaneous submissions of the sub-applications to multiple batch systems is generally well understood [13], the actual improvements in response times of the applications depend on various factors, including decomposability

☆ This work is supported partly by Department of Science and Technology, India, project ref no. SR/S3/EECE/59/2005/8.6.06 and partly by Ministry of Information Technology, India, project ref no. DIT/R&D/C-DAC/2(10)/2006 DT.30/04/07.

* Corresponding author.
E-mail addresses: sundari@rishi.serc.iisc.ernet.in (Sivagama Sundari M.), vss@serc.iisc.ernet.in (S.S. Vadhiyar), ravi@caos.iisc.ernet.in (R.S. Nanjundiah).

0167-739X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2009.08.009

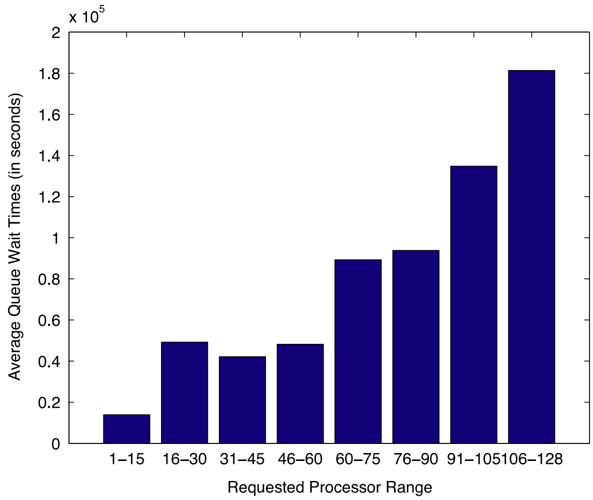

Fig. 1. Average queue wait times for jobs on a SDSC system.

of parallel applications, communication characteristics of the applications, speed of interconnections between the batch systems and batch queue characteristics. Multi-component MPMD applications [14,15] consists of component applications which are parallel application themselves. In these applications, the components are loosely synchronized and communications between components are lighter and less periodic than within components. These multi-component applications and parameter sweep applications, where the parallel tasks are independent, are amenable to decomposition into multiple parallel sub-applications and can potentially benefit due to submissions of the sub-applications to multiple batch systems. Batch queue characteristics including queuing and scheduling policies followed in the batch systems, the loads in the queues, the job distribution in terms of their processor requirements and maximum execution times associated with the queues impact the queue waiting times of the jobs in the queues.

In this paper, we study the effects of these different factors on the potential improvement in performance of a parallel application due to simultaneous submissions of the parallel sub-applications to multiple batch systems of a grid over submission of the entire parallel application to a single batch system. At the application level, we used two simulators, one that models the most prominent multi-component application, CCSM [14,15], and the other that models parameter sweep applications. At the network level, we simulated different interconnection speeds between the batch systems. At the batch level, we used different job traces produced from Feitelson's job models [16] and containing different distributions of jobs with different processor, execution time requirements and execution time limits. We then generated the queue waiting times of the jobs by using three different queuing policies, namely, FCFS, conservative and easy backfilling. We performed large number of simulations with different distributions of processors to components and system configurations with 24 different queues. We show that there are a large number of configurations for which performance improvements are obtained for the applications on batch grids. We further performed real experiments with CCSM by executing the components of CCSM across two AMD Opteron clusters and show similar benefits.

We assume that sub-components executed on different batch systems can communicate with each other. For simplicity, we evaluate the advantages of batch grids using two queues. Thus, our results show the comparison between executing the entire parallel application with P processor requirements on a single batch system and executing two sets of sub-components of the parallel application with processor requirements, $P1$ and $P2$ ($P = P1 + P2$), respectively, on two batch systems of a batch grid.

Section 2 motivates the use of batch grids. In Section 3, we explain in detail our simulation framework and calculations of probabilities of benefits of multiple batch executions for our two applications. Section 4 gives the simulation setup we used for our experiments. In Section 5, we present the results corresponding to CCSM simulations and real executions and simulations of parameter sweep applications. Section 6 describes related work on queue wait times of batch systems and executing applications on multiple batch grids. Conclusions are presented in Section 7 and future work is outlined in Section 8.

2. Batch grids — motivations and contributions

Application jobs submitted to batch systems incur overheads associated with the times spent in the batch queues waiting for resources to become available for execution. These queue waiting times tend to be higher when greater number of processors are requested. The central idea of this paper is to split the application into components and execute them on different batch systems. The smaller requests for the components on the different batch systems are expected to have lower wait times than the single complete request, leading to potential gains in the overall application execution rates. The focus of this paper is to study the incidence of this gain and its variation with different application and system factors. We have developed a simulator framework consisting of multiple components for studying the potential gains due to multiple batch executions for different application and system configurations.

The primary contributions of our work are:

(1) investigation of the benefits of execution of applications on batch grids over execution on single batch systems,
(2) development of a simulation framework, including development of an event-based batch system simulator, inter-site application execution model and a trace-based statistics calculator,
(3) definition of new probability metrics for comparison of batch grid vs. single batch system executions of long-running applications, and
(4) large scale analysis of incidence of benefits due to execution on batch grids for various system and application configurations.

The following section describes in detail our simulation framework.

3. Simulation methodology

We have developed a simulator framework consisting of multiple components for studying the potential gains due to multiple batch executions for different application and system configurations. Our simulation framework, shown in Fig. 2, consists of four components: (i) workload generator, (ii) application simulator, (iii) batch system simulator and (iv) statistics calculator. The workload generator is used to generate job traces with job arrival times, execution times and processor request sizes. The application simulator estimates the application execution rates for various intra-site and inter-site distributions of components. This is used in the calculation of our comparison metrics. The batch system simulator is our event-based simulator that processes the job traces produced by the workload generator and outputs the queue waiting time for each job. The new traces with the queue waiting times are used by the statistics calculator along with the application simulator, to estimate the probabilities of multiple-site executions outperforming single-site executions.