Costs and Benefits of Load Sharing in the Computational Grid

Darin England and Jon B. Weissman

Department of Computer Science and Engineering
University of Minnesota, Twin Cities
{england,jon}@cs.umn.edu

Abstract. We present an analysis of the costs and benefits of load sharing of parallel jobs in the computational grid. We begin with a workload generation model that captures the essential properties of parallel jobs and use it as input to a grid simulation model. Our experiments are performed for both homogeneous and heterogeneous grids. We measured average job slowdown with respect to both local and remote jobs and we show that, with some reasonable assumptions concerning the migration policy, load sharing proves to be beneficial when the grid is homogeneous, and that load sharing can adversely affect job slowdown for lightly-loaded machines in a heterogeneous grid. With respect to the number of sites in a grid, we find that the benefits obtained by load sharing do not scale well. Small to modest-size grids can employ load sharing as effectively as large-scale grids. We also present and evaluate an effective scheduling heuristic for migrating a job within the grid.

1 Introduction

An emerging trend in high-performance computing is to build interconnected networks of super-computing centers known as computational grids. Individually, these centers house computing resources and instruments needed for large-scale collaborative applications. As these applications place increasing demands on existing resources, increased efficiency in scheduling jobs onto the grid is becoming more important. Already proven in the LAN environment, load sharing is becoming feasible in WANs and grids. The emergence of test-beds like TeraGrid[15] promises remarkable network bandwidth between distant sites, enabling load sharing with minimal network penalties.

In this work we investigated the costs and benefits of load sharing of parallel jobs in a simulated computational grid. First we present a detailed model of a supercomputer workload. The nature of our workload model makes it easy to use as input to the grid simulation experiments. We performed experiments for both homogeneous and heterogeneous grids. The results indicate that load sharing among sites is indeed worthwhile. We find that in a homogeneous grid any amount of load sharing results in decreased wait times for users. In a heterogeneous grid with differing workloads and machine capacities, we find that the processing of remote jobs on a (previously) lightly-loaded machine can cause

D. Feitelson, L. Rudolph, and U. Schwiegelshohn (Eds.): JSSPP 2004, LNCS 3277, pp. 160–175, 2005.

delay to local jobs. We also investigate how well the benefits of load sharing scale as the number of sites in a grid increases. The benefits are limited in that most of the opportunities for load sharing are exploited in small and medium-sized grids. Finally we present a simple heuristic for determining the target machine of a migrated job. This heuristic, which we call *Weighted Queue*, is easy to compute and does not require estimates of job run time. The paper is organized as follows: In Section 2 we discuss related research. We present our workload model in Section 3. Section 4 represents the bulk of the paper. It includes a description of the simulation model, the homogeneous and heterogeneous grid results, the scaling results, and the evaluation of the scheduling heuristic. Section 5 concludes the work.

2 Related Work

The quality of the input data is paramount to any simulation model. Cirne and Berman [1] developed a comprehensive model of workloads for space-shared parallel supercomputers. They modeled the variation in job arrival rates throughout the work day and they examined the differences between estimated and actual job run times. Our workload model differs from theirs in that we provide an alternative method for generating the job arrivals and for modeling the job run times and job run lengths. We describe our workload model in the next section. In considering the importance of workload traces in simulation experiments, Lo et. al. [10] investigated the effects on job scheduling algorithms due to the use of real workload traces vs. synthetic workload models. They found that the use of either real or synthetic workloads did not affect the overall performance of job scheduling algorithms. However, we note that the use of a real workload trace necessarily limits the simulation to a single run. By using a workload model and its generated job traces, a large number of simulation runs may be conducted, thereby producing enough data to make statistically significant comparisons among alternative scenarios. Lo et. al. did find that other workload characteristics such as the proportion of *power-of-two* job sizes and the correlation between job size and job run time did affect scheduler performance. In the next section we discuss these two characteristics as they relate to our experiments.

Hollingsworth and Maneewongvatana [7] propose a novel approach to scheduling parallel jobs in a computational grid. They present the idea of an imprecise calendar where jobs are scheduled into time slots by a hierarchical system of manager nodes. Time slots that are further into the future are scheduled at a coarse level. As the time for a slot nears, it is scheduled at a finer level. Like the imprecise calendar approach, we wish to efficiently distribute parallel jobs in a grid. However, we employed simple scheduling methods that do not require job information such as run length[1]. Eager et. al. [2] examined the relative benefits of

[1] That is, no estimate of run length is required for job migration to another machine in the grid. However, we employ backfilling at the local machine level which requires run time estimates.