Are User Runtime Estimates Inherently Inaccurate?

Cynthia Bailey Lee[1,2], Yael Schwartzman[1], Jennifer Hardy[1], and Allan Snavely[1,2]

[1] San Diego Supercomputer Center
University of California, San Diego
9500 Gilman Drive
La Jolla CA 92093-0505, USA
{cl,yaels,jhardy,allans}@sdsc.edu
[2] Department of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive
La Jolla CA 92093-0114, USA

Abstract. Computer system batch schedulers typically require information from the user upon job submission, including a runtime estimate. Inaccuracy of these runtime estimates, relative to the actual runtime of the job, has been well documented and is a perennial problem mentioned in the job scheduling literature. Typically users provide these estimates under circumstances where their job will be killed after the provided amount of time elapses. Also, users may be unaware of the potential benefits of providing accurate estimates, such as increased likelihood of backfilling. This study examines user behavior when the threat of job killing is removed, and when a tangible reward for accuracy is provided. We show that under these conditions, about half of users provide an improved estimate, but there is not a substantial improvement in the overall average accuracy.

1 Introduction

It is a well-documented fact that user-provided runtime estimates are inaccurate. Characterizations of this error in various real workload traces can be found in several classic and recent papers. Cirne and Berman [1] showed that in four different traces, 50 to 60% of jobs use less than 20% of their requested time. Ward, Mahood and West [7] report that jobs on a Cray T3E used on average only 29% of their requested time. Chiang, Arpaci-Dusseau and Vernon [4] studied the workload of a system where there is a 1-hour grace period before jobs are killed, but found that users still grossly overestimate their jobs' runtime, with 35% of jobs using less than 10% of their requested time (includes only jobs requesting more than one minute). Similar patterns are seen in other workload analyses [2,3,5].

Many factors contribute to the inaccuracy of user estimates. All workloads show a significant portion of jobs that crash immediately upon loading. This

D. Feitelson, L. Rudolph, and U. Schwiegelshohn (Eds.): JSSPP 2004, LNCS 3277, pp. 253–263, 2005.

is likely more indicative of users' difficulties with configuring their job to run correctly, than difficulties with providing accurate runtime estimate [2]. However, a job's runtime may also vary from run to run due to load conditions on the system. In an extreme example, Jones and Nitzberg [9] found that on an Origin system where different jobs on the same node share memory resources, job runtime varied 30% on a lightly loaded system, to 300% on a heavily loaded system.

Mu'alem and Feitelson [2] note that because many systems kill jobs after the estimated time has elapsed, users may be influenced to "pad" their estimates, to avoid any possibility of having their job killed. Therefore, we believe that it is important to be precise about what users are typically asked to provide, which is a time after which they would be willing to have their jobs killed, and to distinguish this from the abstract notion of an estimate of their jobs' runtime. This leads us to prefer the term, *requested runtime* for the former, reserving the term *estimated runtime* for a best guess the user can make without any penalty (and possibly even with an incentive for accuracy).

This paper focuses on two specific causes of error in user provided runtime estimates:

1. Requested runtimes are used as a "kill time"—in other words, jobs are killed after the provided time has elapsed.
2. Users may be insufficiently motivated to provide accurate runtime estimates. Many users are likely unaware of the potential benefits of providing an accurate request, such as higher probability of receiving quicker turnaround (because of an increased likelihood of backfilling), or this motivation may not be strong enough to elicit maximum accuracy.

A significant unanswered question is, can and would users be accurate if these two barriers to accuracy were removed? This study addresses this question by asking users of the Blue Horizon system at the San Diego Supercomputer Center (SDSC) [8] for a non-kill-time estimate of their jobs' runtime, and offering rewards for accuracy.

The rest of the paper is organized as follows. In Section 2, we describe the experiment design. In Sections 3 and 4 we present the results of the accuracy of users' non-kill estimates, and their confidence in their estimates, respectively. Section 5 reviews related work on the impact of user inaccuracy on scheduler performance. Finally, Sections 6 and 7 present the conclusions and future work.

2 Survey Experiment Design

Users of the Blue Horizon system submit jobs by using the command *llsubmit*, passing as an argument the name of a file called the job *script*. The script contains vital job information such as the location and name of the *executable*, the number of *nodes* and *processors* required, and a *requested runtime*. An analysis of the requested runtimes from the period prior to the experiment shows that the error