Understanding User Behavior: from HPC to HTC

Stephan Schlagkamp[1,2], Rafael Ferreira da Silva[2],
Ewa Deelman[2], and Uwe Schwiegelshohn[1]

[1] Robotics Research Institute, TU Dortmund University, Dortmund, Germany
[2] University of Southern California, Information Sciences Institute, Marina Del Rey, CA, USA
{stephan.schlagkamp,uwe.schwiegelshohn}@udo.edu, {rafsilva,deelman}@isi.edu

Abstract
In this paper, we investigate the differences and similarities in user job submission behavior in High Performance Computing (HPC) and High Throughput Computing (HTC). We consider job submission behavior in terms of parallel batch-wise submissions, as well as delays and pauses in job submission. Our findings show that modeling user-based HTC job submission behavior requires knowledge of the underlying bags of tasks, which is often unavailable. Furthermore, we find evidence that subsequent job submission behavior is not influenced by the different complexities and requirements of HPC and HTC jobs.

Keywords: User behavior, user sessions, batch submissions.

1 Introduction

Understanding, interpreting, and modeling user submission behavior in parallel computing environments is crucial to build reliable and meaningful testing and benchmarking tools for parallel job schedulers [3]. However, there is a need to narrow the gap between scheduling techniques used in practice and suggested in theory [10]. Some studies have focused on understanding the feedback effects between system performance and the subsequent user behavior to improve the quality of performance evaluation methods [2, 8], and to support practical testing and application of scheduling algorithms. In the past few years, modeling and simulating user behavior (in a dynamic fashion), and adapting scheduling techniques have been studied for HPC systems [8, 11]. Nevertheless, most of that research focused on the development of methods and techniques to capture and evaluate the dynamic effects in HPC systems. In this paper, we investigate whether popular methods to analyze user behavior in HPC are also suitable for evaluating possible feedback effects in HTC systems. Although HPC jobs are mainly tightly-coupled and HTC jobs are mostly embarrassingly parallel (bags of tasks), they share common concepts inherent to parallel environments. Therefore, we aim to unveil similarities and differences in human job submission behavior in both systems. We focus on the two submission properties resulting from individual human user behavior: (1) the characterization of working in batches [12], and (2) the user behavior in terms of the so-called *think times* [2].

Related Work. The analysis of think times in HPC workloads was first introduced in [2], while concepts of analyzing batches and sessions describing user behavior were described in [6, 12]. In [5], an analysis of the accuracy of models for estimating bags of tasks in workflows highlight flaws, however the analysis is limited to aggregated CPU time, while this paper focuses on the subsequent job submission behavior. A characterization of the CMS workloads is presented in [4], however bags of tasks are not considered. Several papers use dynamically changing behavior, such as different think times or probabilities of job submissions simulating user behavior, e.g., [8]. The methodology of accessing aspects of user behavior through data-driven analyses is also applied to measure quality of user-provided runtime estimates of parallel computing jobs [7].

2 Workload Characterization

Studies presented in this paper are based on the workload of the HTCondor pool for the CMS experiment deployed at the San Diego Supercomputing Center [1], and on the workload from Mira, the IBM Blue Gene/Q system at the Argonne Leadership Computing Facility (ALCF). Table 1 shows the main characteristics for these workloads. Due to privacy issues, any user-specific data have been previously anonymized and not retained, and we do not perform analysis down to the job level, but to groups of jobs. The CMS workload is composed of single-core (embarrassingly parallel) jobs submitted as bag of tasks. Each bag of tasks belongs to a certain experiment, which is run by a unique user. A typical CMS analysis consists of the execution of collision readout events, which are stored in files logically grouped into datasets. In principle, all CMS experiments use the same software base, CMSSW, but users may define their own code, analyses, etc. CMS jobs are then distributed among several computing centers for execution. In Mira, the workload is composed of multicore (tightly-coupled) jobs.

Characteristic		Aug 2014 (CMS08)	Oct 2014 (CMS10)
General workload	# jobs	1,435,280	1,638,803
	# users	392	408
	# execution sites	75	72
	# execution nodes	15,484	15,034
Jobs statistics	Completed jobs	792,603	816,678
	Preempted jobs	257,230	345,734
	Exit code (!= 0)	385,447	476,391
	Runtime (in seconds)	9,444.6	9967.1
	Disk (in MB)	55.3	32.9
	Memory (in MB)	217.1	2030.8

(a) CMS workload

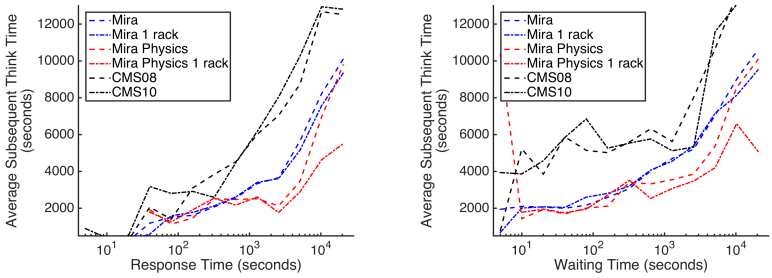
Science Field	#Users	#Jobs	CPU hours (millions)	Runtime (seconds)
Physics	73	24,429	2,256	7,147
Materials Sci.	77	12,546	895	5,820
Chemistry	51	10,286	810	6,131
Computer Sci.	75	9,261	96	917
Engineering	98	6,588	614	10,551
Earth Science	42	6,455	270	5,181
Biological Sci.	31	3,642	192	6,680
Other	40	5,575	565	6,017
Mira	487	78,782	5,698	6,093

(b) Mira workload

Table 1: Characteristics of the CMS (Aug and Oct 2014), and Mira (Jan–Dec 2014) workloads.

3 User and Job Submission Behavior

In this section, we analyze users' subsequent job submission behavior, in terms of *think time* in HPC and HTC. We characterize HTC workloads using common methods previously used to capture the dynamic effects in HPC. We then evaluate the quality of the analysis results by comparing them to a ground-truth knowledge recorded in the HTC traces. Finally, we extend the current methods by broadening their definition to accommodate the concept of bags of tasks.



(a) Think time / Response time (b) Think time / Waiting time

Figure 1: Average think times as a function of response and waiting times for CMS and Mira.

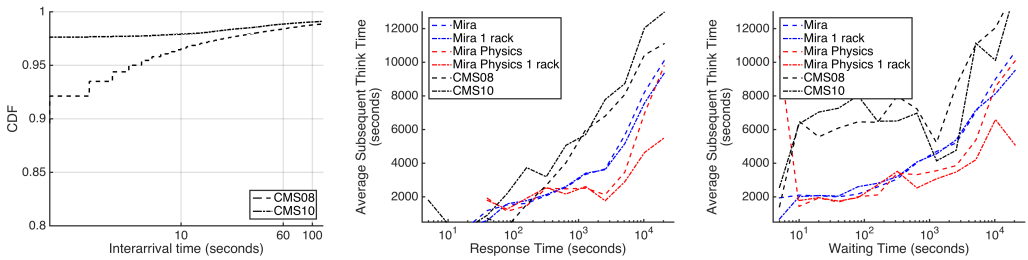
Think Time. The think time (TT) quantifies the time between the completion of a job j, and the submission of the next job j' by the same user [2]: $\mathrm{TT}(j, j') := s_{j'} - c_j$, where $s_{j'}$ is the submit time of job j', and c_j the completion time of job j. The completion time is the sum of the jobs submit time s_j and the response time $r_j = w_j + p_j$ (where w_j is the job waiting time, and p_j the processing time). Fig. 1a shows the average subsequent think times in terms of response time for Mira and CMS workloads. Both workloads follow the same linear trend. However, we observe higher subsequent think time values for the CMS workloads, while Mira-Physics (and its minimum allocation '*1 rack*') yield much lower think time values. The high values experienced by the CMS workloads may be due to long and nondeterministic waiting times experienced in HTC systems, which can influence the response time of jobs in a bag of tasks. Fig. 1b shows the average subsequent think times in terms of the waiting time. In [9], we showed that the waiting time has no prevalence on users' subsequent job submission behavior for the Mira workload, but the job complexity, measured as the number of allocated resources. For the CMS workloads, the waiting time has significant influence on jobs with very short queueing time. This nearly constant behavior is atypical and unexpected in such environment. However, as the waiting time increases, the think time follows a linear increase.

3.1 Redefining Think Time Behavior Analysis in HTC

An HTC experiment is often defined as a bag of tasks problem. In Fig. 1, we observed that the CMS workloads tend to have similar job submission behavior, in terms of think time, to the HPC workload. However, handling HTC workloads at the job granularity may lead to misinterpretations of the data. Therefore, we extend the definition of think time to compute the time interval between subsequent *bags of tasks* (BoT) submissions, instead of jobs. Hence, the think time TT between two subsequent bags of tasks J and J' submitted by the same user is defined as $\mathrm{TT}(J, J') := s_{J'} - c_J$, where $s_{J'}$ is the submit time of the BoT J', and c_J the completion time of the BoT J. We define the submit time s_J, waiting time w_J, processing time p_J, and response time r_J of a bag of task J as follows:

$$s_J := \min\{s_j \mid j \in J\}, \tag{1}$$
$$w_J := \min\{s_j + w_j \mid j \in J\} - s_J, \tag{2}$$
$$p_J := \max\{s_j + w_j + p_j \mid j \in J\} - (s_J + w_J), \tag{3}$$
$$r_J := w_J + p_J. \tag{4}$$

Typically, two jobs submitted by a user are from the same batch if their interarrival time is within a threshold [6, 12]. However, this definition does not account for overlapping BoT submissions. The CMS workloads provide an additional field, which relates a job to an experiment.

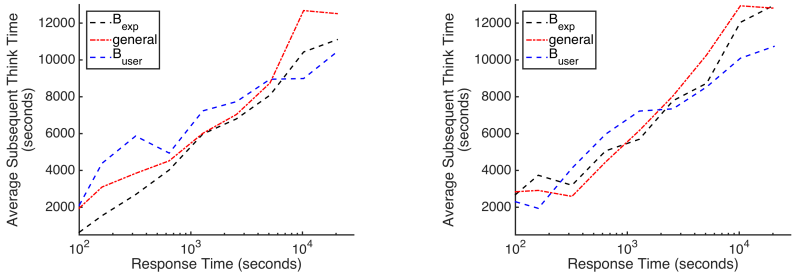


(a) Interarrival times CDF (b) Think time / Response time (c) Think time / Waiting time

Figure 2: Distribution of interarrival times and think times for approximated bag of tasks.

Fig. 2a shows the interarrival time distribution for both HTC traces. Most of the jobs belonging to the same experiment and user (97%) are submitted within one minute, which is the threshold used to distinguish between automated BoT submissions and human-triggered submissions (batches). Fig. 2b shows the average subsequent think times in terms of response time for approximated BoTs. Both HTC workloads follow the same linear trend. However, we observe lower TT values when compared to the standard analysis based on individual jobs (Fig. 1a). Note that the think time behavior of the CMS workloads are also closer to the HPC behavior. This result indicates that (1) methods commonly used to characterize HPC workloads produce better estimates when computational analyses are represented as BoTs, and (2) HTC BoTs are comparable to HPC jobs. Similarly to the analysis shown in Fig. 1b, the user behavior in CMS is not related to the waiting time (Fig. 2c). We acknowledge that the definition of waiting time for bags of tasks (Eq. 2) may not capture the actual think time behavior: we assume that the BoT waiting time is defined as the timespan between the first job submission and the earliest job start time of a BoT. The indeterministic waiting times experienced from the remaining jobs may significantly impact the user behavior. However, modeling this dynamic behavior would require a user-assisted analysis, which is out of the scope of this paper. As Mira's user behavior is strongly influenced by the job complexity (number of cores) [9], we argue that the HTC user behavior is mostly influenced by the number of jobs in a batch.

Analysis of distinct think time definitions. Most workload traces are devoid of BoTs or experiment identifier information. Therefore, we investigate how different definitions of think time, in terms of how the job granularity is defined, may lead to misinterpretations of the data. We define B_{exp} as the aggregation of BoTs based on jobs submitted by the same user and belonging to the same experiment (Fig. 2). B_{exp} represents the ground-truth knowledge and is



(a) `CMS08` (Aug 2014) (b) `CMS10` (Oct 2014)

Figure 3: Different data interpretations for TT computation in the CMS workloads.

used to evaluate the accuracy of the following definitions: (1) B_{user}, BoTs are defined based on jobs submitted by the same user (the most common method, but it does not respect overlapping jobs or BoTs); and (2) `general`, jobs are treated individually (Fig. 1). Fig. 3 shows the average subsequent think times in terms of response time. All data sources have comparable correlation between response and subsequent think times. However, the think time behavior for `general` is closer to B_{exp} than the B_{user}. The root-mean-square error (RMSE) between `general` and B_{exp} is 1,021.52s for `CMS08` and 1,215.28s for `CMS10`, while RMSE between B_{user} and B_{exp} is 1,287.87s for `CMS08` and 2,451.69s for `CMS10`. The poor prediction produced by B_{user} is due to the overestimation of the BoT sizes. Similar behavior is also verified in a previous work [5].

4 Conclusion

In this paper, we have shown that user submission behavior extracted from HTC workloads is not inherently different from submission behaviors in HPC environments. Nevertheless, the analysis of HTC workloads should target bags of tasks, instead of individual embarrassingly parallel jobs. We extended the characteristics of parallel jobs to BoTs, and demonstrated that the think time behavior follows a similar linear trend for increasing response times of jobs, but is slightly greater in the HTC environment. The findings of this paper support the use of evaluation methods respecting individual user behavior to simulate HTC behavior. Future work includes the in-depth characterization of waiting times in bags of tasks to improve the correlation analysis between queueing times and the subsequent user job submission behavior.

Acknowledgements. This work was partly funded by DOE under the contract number ER26110, "dV/dt - Accelerating the Rate of Progress Towards Extreme Scale Collaborative Science", and contract #DESC0012636, "Panorama - Predictive Modeling and Diagnostic Monitoring of Extreme Science Workflows". We also thank William Allcock, Frank Würthwein, James Letts, OSG, the CMS collaboration, and ALCF.

References

[1] D Bradley et al. Use of glide-ins in CMS for production and analysis. *Journal of Physics: Conference Series*, 219(7), 2010.

[2] D. G. Feitelson. Looking at data. In *IPDPS*, 2008.

[3] D. G. Feitelson. *Workload modeling for computer systems performance evaluation.* 2015.

[4] R. Ferreira da Silva et al. Characterizing a high throughput computing workload: The compact muon solenoid (CMS) experiment at LHC. *Procedia Computer Science*, 51, 2015. ICCS.

[5] R. Ferreira da Silva and T. Glatard. A science-gateway workload archive to study pilot jobs, user activity, bag of tasks, task sub-steps, and workflow executions. In *Euro-Par*, 2013.

[6] A. Iosup et al. The characteristics and performance of groups of jobs in grids. In *Euro-Par*. 2007.

[7] C.B. Lee and A. Snavely. On the user–scheduler dialogue: studies of user-provided runtime estimates and utility functions. *IJHPCA*, 20(4), 2006.

[8] S. Schlagkamp. Influence of dynamic think times on parallel job scheduler performances in generative simulations. In *JSSPP*, Hyderabad, India, 2015.

[9] S. Schlagkamp et al. Consecutive Job Submission Behavior at Mira Supercomputer. In *HPDC*, 2016.

[10] U. Schwiegelshohn. How to design a job scheduling algorithm. In *JSSPP*, 2014.

[11] E. Shmueli and D. G. Feitelson. On simulation and design of parallel-systems schedulers: Are we doing the right thing? *IEEE TPDS*, 20(7), 2009.

[12] N. Zakay and D. G. Feitelson. On identifying user session boundaries in parallel workload logs. In *JSSPP*, 2012.