Consecutive Job Submission Behavior at Mira Supercomputer

Stephan Schlagkamp[1,2], Rafael Ferreira da Silva[2], William Allcock[3]
Ewa Deelman[2], 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
[3]Argonne National Laboratory, Argonne, IL, USA
{stephan.schlagkamp,uwe.schwiegelshohn}@udo.edu, {rafsilva,deelman}@isi.edu, allcock@anl.gov

ABSTRACT

Understanding user behavior is crucial for the evaluation of scheduling and allocation performances in HPC environments. This paper aims to further understand the dynamic user reaction to different levels of system performance by performing a comprehensive analysis of user behavior in recorded data in the form of delays in the subsequent job submission behavior. Therefore, we characterize a workload trace covering one year of job submissions from the Mira supercomputer at ALCF (Argonne Leadership Computing Facility). We perform an in-depth analysis of correlations between job characteristics, system performance metrics, and the subsequent user behavior. Analysis results show that the user behavior is significantly influenced by long waiting times, and that complex jobs (number of nodes and CPU hours) lead to longer delays in subsequent job submissions.

Keywords

User behavior, workload analysis, performance modeling.

1. INTRODUCTION

High Performance Computing (HPC) is mainstream for performing large-scale scientific computing [8, 12]. As a result, computing centers are devoting significant effort to satisfy the requirements of scientific applications, and provide high QoS. Understanding user reactions to the system performance is a key factor for improving user satisfaction, while improving the performance of scheduling systems [3]. Job schedulers often model workloads based on job requirements or historical performance data. However, using workload traces without the understanding of the user behavioral mechanisms may produce misleading results [18].

In this work, we aim to improve performance evaluation processes and testing environments by investigating the feedback effects between parallel job characteristics. We evaluate how system performance and job characteristics impact users' subsequent job submission behavior in HPC. We then extend and evaluate the definition of users' think time [2] (the timespan between a job completion and the submission of the next job), to assess the influence of system delays, and job complexity (number of nodes and CPU time) on the user behavior. We analyze a 1-year scheduling trace (2014) from the Mira supercomputer at ALCF to characterize the subsequent think time as a function of the job response time, as well as the think time response to queueing and processing time. Furthermore, we also analyze the think time in response to the slowdown and the job complexity. Our findings show that these components are strongly correlated and have a significant influence on user behavior. The main contributions of this work include (1) the characterization of a leadership supercomputer scheduling workload; (2) an evaluation of the think time definition for measuring delays in users' subsequent job submission behavior in HPC systems; (3) an in-depth analysis of correlations between subsequent think times, job characteristics, and system performance metrics; and (4) an analysis of the correlation of multidimensional metrics on user behavior.

2. BACKGROUND AND RELATED WORK

Although there is a plethora of works that analyze and suggest improvements to schedulers in HPC [3], there is a gap between theoretical results and their practical application [17]. This issue can be addressed by: assessing user behavior through cognitive studies, e.g., in the form of questionnaires—to investigate user reactions to high system utilization or acceptance for long waiting times, and user satisfaction [13,16]; or analyzing workload traces gathered from these systems, which can reveal aspects of user behavior related to system performance metrics and job characteristics. In [2], aspects of dynamic correlations between system performance, utilization, and the subsequent behavior are observed from analyzes of user behavior using HPC traces. These analyses have enabled: the development of models emphasizing aspects of the user behavior [10]; scheduling algorithms that leverage the knowledge about the users [19]; the analysis of workloads to characterize the submission behavior in the form of batches of jobs and user sessions [20]; and workload models and simulations to mimic the dynamic nature of user and system interaction [5, 15]. Several pa-

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Science Field	#Users	#Jobs	CPU hours (millions)	#TT Jobs
Physics	73	24,429	2,256	2,675
Materials Science	77	12,546	895	1,530
Chemistry	51	10,286	810	1,959
Computer Science*	75	9,261	96	–
Engineering	98	6,588	614	1,870
Earth Science	42	6,455	270	1,397
Biological Sciences	31	3,642	192	–
Other	40	5,575	565	–
Mira	487	78,782	5,698	14,145

*significant number of jobs run in *backfill* queue

Table 1: Characteristics of the Mira workload from Jan–Dec 2014, and number of subsequent jobs with positive think times: $0 < TT \leq 8$hrs.

pers have addressed computing workload characterization and modeling. In [6, 7, 9], analyses of grid, HPC, and HTC workload characteristics emphasized system usage, user population, and application characteristics. In [14], an analysis of a 5-years workloads from two Supercomputers at NERSC evaluates system performance metrics. The I/O behavior of the Intrepid Supercomputer at ALCF is shown in [1], while analyses of I/O workload traces from Intrepid and Mira are shown in [11]. Although these papers present a detailed analysis of system performance metrics, none of them have focused on the user behavior.

3. WORKLOAD CHARACTERIZATION

The analyses presented here are based on the workload from Mira, the IBM Blue Gene/Q system at ALCF. Mira is a 786,432-core production system with 768 TiB of RAM, and a peak performance of 10 PFlops. Each node is composed of 16 cores, and the minimum allocation per job is 512 nodes (8,192 cores). Mira's workload comprises 1-year computational jobs execution in 2014, which consists of 78,782 jobs, submitted by 487 users from 13 science domains. In total, these jobs consumed over 5.6 billion CPU hours. Table 1 shows the summary of the main characteristics of the dataset, and highlights the most important (by the number of jobs) science domain fields. Most of Computer Science jobs (∼65%) consume less than the minimum allocation (i.e., 512 nodes or 8,192 cores), and have very short runtimes (less than 15 min), thus we see the low CPU hours consumption regardless the high number of jobs. Furthermore, about 25% of the computer science jobs ran in the backfill queue, which may bias user behavior—the uncertainty of the job start time is elevated. Therefore, Computer Science jobs are not considered in this study.

4. CHARACTERIZING THINK TIME

The user's *think time* quantifies the timespan between a job completion and the submission of the next job (by the same user) [2]. This metric is seen as capturing the influence of the system performance on user behavior (e.g., dissatisfied users may tend to throttle job submission, long queueing times may deviate the user's focus from their experiments, etc. [4, 18]). In this paper, we analyze think time as a function of performance, i.e., response time (job walltime) and slowdown, and investigate whether waiting time or runtime have a more significant impact on the user behavior [2]. Additionally, we evaluate how job complexity (in terms of job size and total CPU time) may also affect the think time behavior.

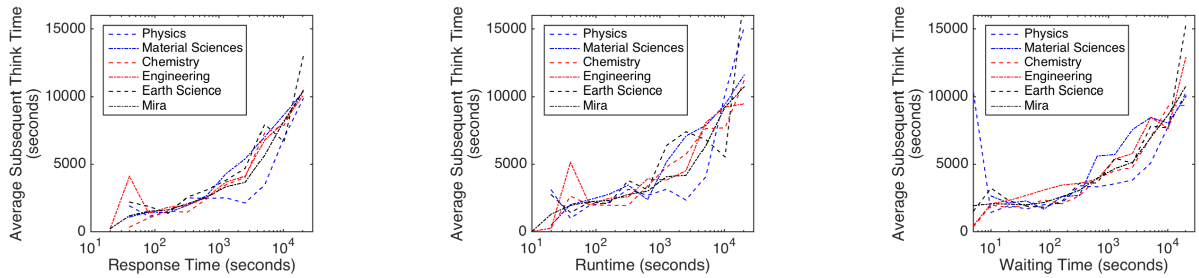
Think Time. Let s_j be the time when a job j is submitted, p_j the job processing time, and w_j the job waiting time. We define the job response time r_j as the sum of its waiting and processing times: $r_j = w_j + p_j$. Thus, we define the job completion time c_j as the sum of the job submission and response times: $c_j = s_j + r_j$. Job interarrival time is the timespan between two subsequent job submissions (j and j') by the same user. Two subsequent jobs are *overlapped* if job j has not finished before job j' is submitted ($c_j > s_{j'}$). Otherwise, they are considered *non-overlapped*, which are the set of jobs we focus on this paper. Therefore, we define think time TT as the timespan between the completion time of job j and the submission time of its successor j': $\mathrm{TT}(j, j') = s_{j'} - c_j$. For overlapping jobs, the think time is negative, thus we only consider subsequent job submissions of positive think time. Additionally, we only consider think times of less than eight hours, which is intended to represent subsequent job submissions belonging to the same working day. This threshold also eliminates biased user behaviors characterized by absent submissions for long periods of time followed by burst submissions for short periods (e.g., conference deadlines, allocation expiration, etc.). Table 1 also shows the number of subsequent jobs with positive think times for the studied science domains.

4.1 Analysis of Job Characteristics and Performance Parameters on Think Time

The analysis of think time behavior is often limited to the study of the impact of response time on user behavior. As response time is defined as a function of waiting and processing times, we evaluate how these components correlate with users' think times. Fig. 1a shows the average think times for subsequent jobs of Mira. All science fields follow the same linear trend, with slight differences for Engineering (for short response times) and Physics (for response times ∼5,000s). This difference is due to a few points that deviate from the averages. For Engineering, the peak is due solely to a pair of jobs that present a very high think time value of ∼8h. For Physics, a few points yield very low values (nearly instantaneous subsequent submissions). This behavior is typically due to the use of automated scripts or jobs that failed within a few seconds after submission. The analysis of think times in terms of processing time (a.k.a. runtime, Fig. 1b) and waiting time (Fig. 1c) shows that on average, the parameters have an equal influence on user behavior. This result lead to the conclusion that reducing queueing times would not significantly improve think times for long running jobs.

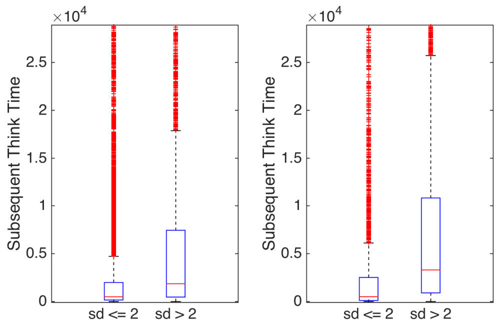
4.2 Analysis of Job Characteristics in Terms of Runtime and Waiting Time

The analysis of think times for subsequent job submissions of the Mira's trace showed that system performance metrics such as runtime and waiting time have a significant impact on user behavior. Hence, we investigate how job characteristics, in particular the job size and workload, combined with performance parameters impact think times. To this end, we conduct analyses using multidimensional metrics, i.e., we analyze the subsequent think time in response to, slowdown and job size. Note that the slowdown is itself another multidimensional metric defined as the factor between a job actual response time and its runtime: $\mathrm{sd}(j) = \frac{r_j}{p_j} = \frac{w_j + p_j}{p_j}$. The analyses conducted here use the



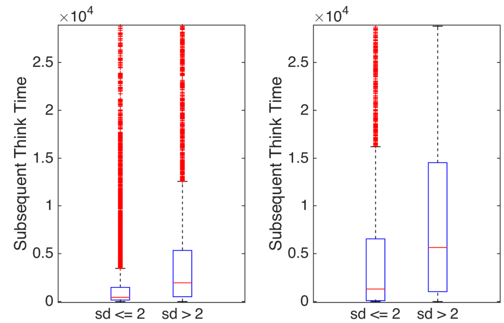
(a) (b) (c)

Figure 1: Average think times as a function of (a) response time, (b) runtime, and (c) waiting time.



(a) $\sigma \leq 512$ (b) $\sigma > 512$

Figure 2: Influence of prevalent (sd ≤ 2) and non-prevalent (sd > 2) runtimes on the users think times for (a) small and (b) large jobs in terms of job size (number of nodes). Note that sd denotes the slowdown, and whiskers are defined as 1.5 IQR.



(a) $\omega \leq 10^6$s (b) $\omega > 10^6$s

Figure 3: Influence of prevalent (sd ≤ 2) and non-prevalent (sd > 2) runtimes on the users think times for (a) small and (b) large jobs in terms of workload.

job slowdown sd as a metric to separate jobs into two subsets: (1) *runtime-dominant*—the job runtime prevails the waiting time (sd $\leq$ 2); and (2) *wait-time-dominant*—jobs spend more time in queue than running (sd $>$ 2).

Fig. 2 shows the think time distribution according to job sizes. We divide the dataset into groups of small jobs that require the minimum amount of allocated nodes ($\sigma \leq 512$, Fig. 2a), which represent 49.2% of the total number of subsequent jobs, and large jobs requiring up to all available nodes (Fig. 2b). This threshold identifies the subset of jobs with low think time values ($<$ 1.5 hours). Several outliers characterize the datasets as heavy-tailed distributed, which is expected due to the natural variation of the user behavior and the large number of sampling data. Therefore, our analyses use the median as a robust metric to cope with outliers. In both scenarios, think times are relatively small when runtime prevails. The median think time is 507s for small jobs, and for large jobs 439s. The third quartile also yields low values (2,083s for small, and 2,361s for large jobs). Additionally, user behavior does not seem to be impacted by the job complexity (job size)—the average think times for both small and large jobs are of similar magnitude. Note that the third quartile values for *runtime-dominant* are below median values of *wait-time-dominant*. Prevailing waiting times may significantly affect user behavior, and the job size seems to influence the queueing time. For small jobs, the median think time is 2,478s, and for large jobs 4,276s. This result suggests that think time is not directly bound to job size, but the uncertainty produced by large waiting times. The analysis of the job size parameter is limited to the number of nodes. On the other hand, the job workload ω (defined as the total CPU time of the job) also includes the time dimension. Fig. 3 shows the think time distribution in terms of workload. Small jobs are characterized by usage of less than ~277 CPU hours (10^6s), which identifies subsets with low think time values (under 1.5 hours). In contrast to the previous analysis, more complex jobs do yield higher think times. However, similar behavior is observed when the runtime or waiting time prevail. For *runtime-dominant*, small jobs have a median think time of 437s, and large jobs 1,305s. However, the third quartiles present a larger difference—1,478s for small jobs, and 6,544s for large ones. Waiting times have equivalent influence on the job size analysis. For small jobs, the median think time is 1,954s, and for large jobs 5,645s. These results indicate that (1) complex jobs require more think time to plan and release a new experiment (e.g., visualization and analysis on other systems); or (2) users do not have a full understanding of the expected behavior of their jobs, thus they lack of accurate estimate of the processing time. To validate the first assumption, an assessment of user behavior in the form of direct interview or questionnaires would be required [13], while the second assumption could be validated by investigating jobs that used a notification mechanism to alert the user of job completion.

4.3 Summary and Discussion

Our analysis of the user behavior has advanced the understanding of the think times between subsequent job submissions. Our findings sustain the premise that the job response time is the most significant factor influencing think time. However, not all elements constituting the response time have equivalent influence. Job characteristics (job size and workload) have a substantial impact on the queueing time a job will experience. Therefore, we argue that the think time definition should also consider job complexity. For large workloads, the job runtime also negatively influences the user behavior, despite short queuing times. This result suggests that users need more time to *think* about their experiment results and next steps, in particular for complex experiments. The analysis results contradict the assumptions made for the development of user-aware algorithms based on batches and sessions (e.g., CREASY scheduler [19]). For instance, CREASY considers response time as the main factor to increase steadiness within user sessions. However, further analysis suggests that other characteristics also impact the delays in subsequent job submission behavior. that other characteristics correlate Therefore, we argue that user-aware scheduling should not only consider response time, but also job characteristics such as the job complexity.

Although similar think time behaviors can still be identified in today's systems, the assumptions taken by this definition are restrictive and may lead to misleading conclusions. For instance, the 8hs threshold between subsequent job submissions limits the analysis for a small subset of the dataset (∼19% in this work), which may not capture all consecutive job submission behaviors of the system. The analyzed subset is mostly composed of jobs that require up to 512 nodes, which represents less than 10% of the total dataset. Thus, we argue that this definition does not scale to the complexity of today's applications and systems. Additionally, think time may also include the time that the user spends on other steps of the experiment—it is common to perform further computational analysis and visualization within an experiment using other systems. In this case, the time spent on these system should also be accounted for the think time. Therefore, we argue that a user-assisted analysis would significantly contribute to the understanding of this process. Finally, when simulating submission behavior one has to consider other job characteristics and system performance components beside the response time. For instance, probability based models [18], or linear models [15], which consider response time of jobs or batches to model inter-arrival times, would not produce accurate predictions.

5. CONCLUSION

In this paper, we have investigated the main factors influencing the users' response to system performance (think time). We analyzed over 78K jobs submitted by 450+ users to the Mira HPC system at ALCF. Analysis results show that job response times are linearly correlated to think times. Additionally, the analysis of the job complexity (number of nodes or workload) combined with slowdown unveil strong correlations between waiting time and the subsequent think time. Moreover, large workloads negatively influence user behavior. We acknowledge that the definition of think time may be restrictive and does not cover all edge cases. In the future, we intend to extend the current definition and explore new ones (based on concurrent activities) to evaluate how different assumptions of user behavior are influenced by performance metrics and job characteristics. We also intend to model think time as a function of job complexity from past job submissions. Future work will also include cognitive studies to unravel the real causes driving the user decisions, which cannot be obtained from statistical analysis.

Acknowledgements. This work was partly funded by DOE contract number ER26110, "dV/dt - Accelerating the Rate of Progress Towards Extreme Scale Collaborative Science", and #DESC0012636, "Panorama - Predictive Modeling and Diagnostic Monitoring of Extreme Science Workflows", and by the German Research Foundation, RTG 1855. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

6. REFERENCES

[1] P. Carns et al. Understanding and improving computational science storage access through continuous characterization. *ACM TOS*, 7(3), 2011.

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

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

[4] D. G. Feitelson and A. W. Mu'alem. On the definition of on-line in job scheduling problems. *ACM SIGACT News*, 36(1), 2005.

[5] D. G. Feitelson and E. Shmueli. A case for conservative workload modeling: Parallel job scheduling with daily cycles of activity. In *IEEE MASCOTS*, 2009.

[6] 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.

[7] 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*. Springer, 2013.

[8] A. Geist and D. A. Reed. A survey of high-performance computing scaling challenges. *IJHPCA*, 2015.

[9] D. L. Hart. Measuring TeraGrid: workload characterization for a high-performance computing federation. *IJHPCA*, 25(4), 2011.

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

[11] H. Luu et al. A multiplatform study of I/O behavior on petascale supercomputers. In *ACM HPDC*, 2015.

[12] D. A. Reed and J. Dongarra. Exascale computing and big data. *Communications of the ACM*, 58(7), 2015.

[13] J. Renker et al. Questionnaire for user habits of compute clusters (QUHCC). In *HCI International*. Springer, 2015.

[14] G. P. Rodrigo Álvarez et al. HPC system lifetime story: Workload characterization and evolutionary analyses on NERSC systems. In *ACM HPDC*, 2015.

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

[16] S. Schlagkamp and J. Renker. Acceptance of waiting times in high performance computing. In *HCI International*. Springer, 2015.

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

[18] E. Shmueli and D. G. Feitelson. Using site-level modeling to evaluate the performance of parallel system schedulers. In *IEEE MASCOTS*, 2006.

[19] 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.

[20] N. Zakay and D. G. Feitelson. On identifying user session boundaries in parallel workload logs. In *JSSPP*. Springer, 2013.