Uncovering the Effect of System Performance on User Behavior from Traces of Parallel Systems

Edi Shmueli
School of Computer Science & Engineering
The Hebrew University, Jerusalem 91904, Israel
and IBM Haifa Research Lab
Mount Carmel, Haifa 31905, Israel
Email: edi@il.ibm.com

Dror G. Feitelson
School of Computer Science & Engineering
The Hebrew University
Jerusalem 91904, Israel
Email: feit@cs.huji.ac.il

***Abstract*—Intuitively, it seems that understanding how the performance of a system affects its users requires research in psychology and the conducting of live experiments. We demonstrate that it is possible to uncover the effect from traces of the system. In particular, we found that the behavior of users of parallel systems is correlated with the response time of their jobs, not the slowdown as was previously assumed. We show that response times affect the decision of users to continue or abort their interactive session with the system, and that this may relate to expectations the users develop. Although this research was conducted in the context of parallel systems, we believe our results are more general and may pertain to other types of systems as well.**

I. Introduction

Understanding how the performance of a system affects the behavior of its users helps improve system design. In the context of parallel systems, for example, it will allow the design of better job schedulers, with the goal of satisfying users by exploiting knowledge about user behavior to better plan future actions.

Intuitively, it seems that exploring user behavior requires research in psychology and the conducting of live experiments with real users. The problem is that live experiments are often impractical to conduct; few users have the time or patience to actually record the reasons for their behavior patterns.

We suggest a novel methodology to uncover the effect on users from *traces* of the system. The traces we use contain records of jobs that were actually submitted by the users, scheduled, and finally executed on real, production-use parallel systems. We show that using these traces we are even able to reason about user motivation, not just about the causal relationship between performance and behavior.

The performance of the system can be measured using different metrics, all which are assumed to be important to the users. In particular, the response time of jobs (the time from submission to termination), and the slowdown (the response time normalized by the actual execution time) are two metrics often used in performance evaluations. The first question is therefore, *which* of these metrics is most important to the users in the sense that it affects their subsequent behavior.

Intuitively, the slowdown of jobs is important to users because it reflects the degree to which the performance they actually observed from the system met their expectations. For example, it may be fine for a 10-minute job to wait 5 minutes in the queue (a slowdown of 1.5), but for a 1-minute job to be delayed 14 minutes (same response time of 15 minutes, but slowdown of 15), may be considered unacceptable. Somewhat surprisingly, we found that user behavior is strongly correlated with the response time of their jobs, not the slowdown. This finding calls for a reassessment of suggestions that jobs should be prioritized according to their slowdown [6].

The next question is *how* exactly response times affect the behavior. In reality, users tend to submit several jobs one after the other in periods of activity that are known as sessions. Previous work had already discovered how session data can be identified and extracted from the traces [17]. We found that the decision of the users to continue submitting jobs, or alternatively to abort their session, is affected by the response time of their jobs. Specifically, we show that the higher the response time, the higher the probability for the user to abort his interactive session with the system.

The third and final question we answer in this paper is *why* this happens. It is well known that user behavior is affected by expectations [13], but unfortunately, such informations does not appear explicitly in the traces. Instead, we show that it is possible to isolate specific scenarios in the traces. In particular, we examined the scenario where response times met the expectations of the users, and the scenario where they did not. We found that although the users' perception and motivation are different in the two cases, their actual behavior happens to be very similar.

The paper is organized as follows. Section II describes the traces we used for our analysis. Section III answers the question of *which* performance metric is most important to the users. Section IV answers the question of *how* that metric affect their behavior, and Section V answers the question of *why* this happens. Section VI discusses related work, and Section VII concludes the paper.

II. Trace Data

The data we used for our analysis come from traces that contain records of jobs that were submitted and executed on a variety of large-scale parallel machines over periods ranging

TABLE I
THE FIVE TRACES WE USED FOR OUR ANALYSIS: TOGETHER, THEY REPRESENT MANY YEARS OF ACTIVITY BY HUNDREDS OF USERS.

Trace	Duration	Users	Jobs
SDSC-Par-1995-2.1-cln	1/1995–12/1995	98	53,970
CTC-SP2-1996-2.1-cln	6/1996–5/1997	679	77,222
KTH-SP2-1996-2	9/1996–8/1997	214	28,489
SDSC-SP2-1998-3.1-cln	4/1998–4/2000	437	59,725
SDSC-BLUE-2000-3.1-cln	4/2000–1/2003	468	243,314

TABLE II
RESPONSE TIME BINS (RANGES ARE IN MINUTES).

Resp. time bin	R1	R2	R3	R4	R5
Resp. time range	0–5	5–15	15–45	45–135	135–∞

TABLE III
SLOWDOWN BINS.

Slowdown bin	S1	S2	S3	S4	S5
Slowdown range	1–1.41	1.41–2	2–4	4–16	16–∞

from one to three years. Each job record contains several fields, four of which are relevant for our study: the *user* who submitted the job, the *time of submission*, the job's *wait time* in the scheduler queue, and the job's actual *execution time*, once it got started. The first field allows us to analyze the data on a user basis. The other three fields allow us to find when each job terminated, and to calculate its *response time*, its *slowdown*, and the *think time* between the termination and the submission of the next job by the same user.

We used five traces to ensure that our results are not particular to a certain location and time: together, they represent many years of activity by hundreds of users. They are listed in Table I, and are available on-line from the Parallel Workloads Archive [4]. When available, we use the cleaned versions of the traces, where flurries and other extraordinary activity have been removed.

III. METRICS CORRELATION WITH USER BEHAVIOR

When a user submits a job for execution, this is typically not an isolated event. Rather, users tend to submit several jobs one after the other. In many cases there is a dependency between successive jobs: when a job terminates, the user examines its result, makes corrections and adjustments, and submits another job. The time between the completion of a job and the submission of the next job is known as the *think time*.

The system scheduler, in turn, accepts these jobs from the users and places them in its wait queue. When resources become available, it scans the queue and selects jobs for execution according to some prioritization criteria, and subject to possible reservation constraints.

Obviously, the scheduler's actions depend on the jobs submitted by the users, but user behavior is also dependent on feedback from the scheduler. An efficient scheduler that streams jobs through the system at a high rate encourages users to submit more jobs, while an inefficient scheduler that causes resources to be wasted and jobs to be delayed discourages the submittal of additional work [12].

While existence of such a feedback to the users is intuitively clear, the effect on their behavior is not. The performance of the system can be measured using different metrics, all which may be assumed to be important to the users. The challenge is to find the metric that is really important to the users in the sense that it affects their subsequent behavior.

We focus on the *response time* and *slowdown* of the jobs. The response time of a job is the time elapsed from its submission to termination; it is the sum of the time it spent waiting in the scheduler's queue and the time it actually executed. Intuitively, response time is important to users because they must wait for their jobs to terminate before they can examine the results and submit more jobs.

Slowdown is the response time normalized by the actual execution time. It is also intuitively important to users because it reflects the degree to which the performance they actually observed from the system matched their expectations: A slowdown value that is near 1 indicates that the job response time was close to its execution time, whereas a high slowdown value indicates that response time had lengthened far beyond what would have been expected based on the job's actual runtime.

Finding a metric that reflects the behavior of the users is more challenging. Metrics like the average job submission rate are not useful because averages necessarily mix multiple effects with multiple responses. Instead, we propose to use the *think times* that follow jobs in the traces. The rationale is that think times capture the user's immediate response to the job that has just terminated. A short think time indicates that the user was waiting for his job to complete, that he is satisfied with performance, and intends to continue the interaction. A long think time indicates that the user was probably not waiting for the job, possibly because he had given up on using the system.

Because different jobs receive different levels of service, and different users react differently to their jobs, the response times, slowdowns, and think times of the jobs in the traces exhibit large variance. Tabulating the interaction on an individual job basis in this case is not useful. Instead, we partition the jobs into classes that represent different levels of feedback to the users, and study the aggregate user reaction to the jobs in each class.

The classes are obtained by dividing the jobs into five bins, once according to the response time metric, and again according to the slowdown. The levels of feedback represented by the response time bins are "very fast response", "reasonably fast response", "medium response", etc. and the levels of feedback represented by the slowdown bins are "very low slowdown", "low slowdown", and so on. The boundary points between the bins are roughly logarithmic, as shown in Tables II and III. This has the advantage of resulting in classes of approximately the same sizes in terms of the number of jobs assigned to each.

Once the jobs are grouped in bins, we can analyze the think times that follow the jobs on a per-bin basis, and examine reaction of the users to the different classes of feedback. Figure 1 shows the *median* think times for the five response time bins, and the five slowdown bins.

In the case of response time, the relationship between per-

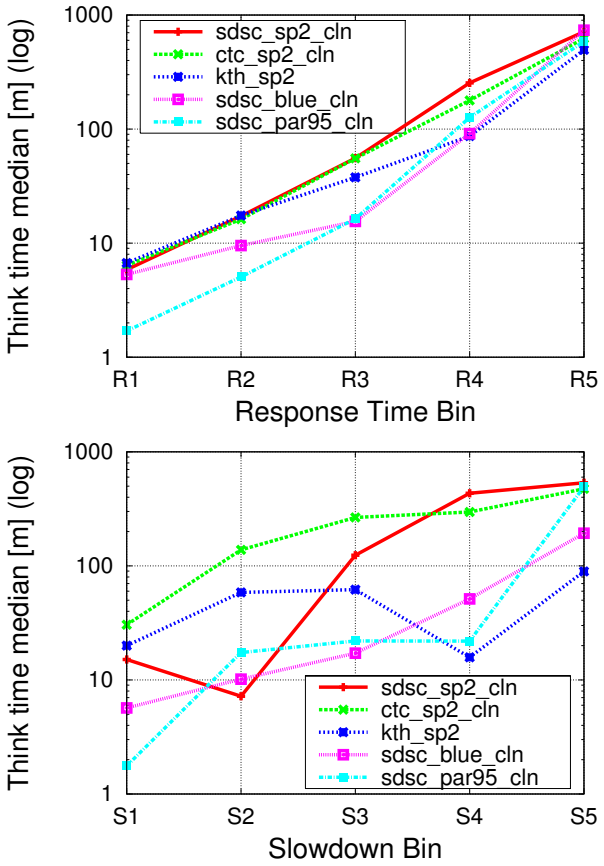


Fig. 1. Correlation of performance metrics with user behavior: Top: high response times correlate with longer think times. Bottom: for slowdown there is only a weak correlation.

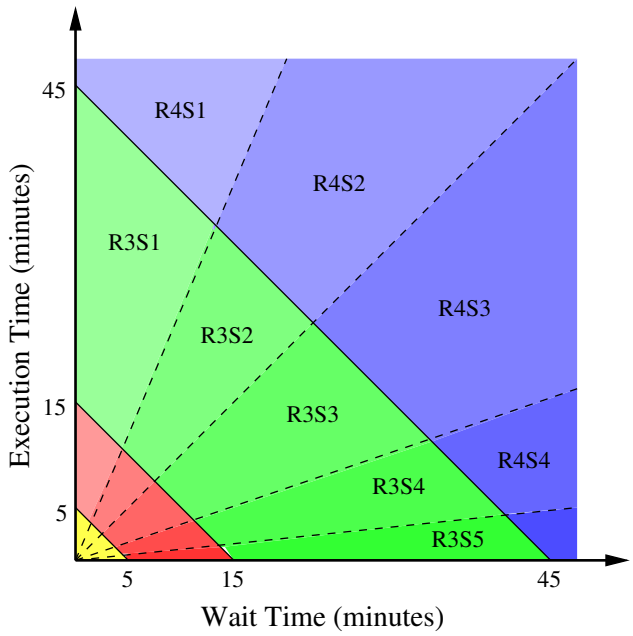


Fig. 2. Graphical illustration of the bins: The response bins are bounded between the solid lines, and the slowdown bins are bounded between the dashed lines.

TABLE IV
MEDIANS OF THINK TIMES FOR SUB-BINS.

Bin	S1	S2	S3	S4	S5
R1	3.6	6.6	8.2	8.8	8.9
R2	9.5	14	18	19	18
R3	17	49	50	41	40
R4	119	149	159	195	119
R5	524	600	652	712	702

formance and subsequent user behavior is clear and consistent for all five traces studied. Jobs in bin 1, which had response times of up to 5 minutes, have the smallest median think time — also about five minutes. The median think time is larger — up to 20 minutes — for bin 2, 20 to 60 minutes for bin 3, etc. This means that users' subsequent behavior is correlated with the response time of their jobs: the faster their jobs respond, the quicker users submit additional jobs.

In the case of slowdown, on the other hand, the relationship is much less clear. First, the results for the different traces are highly dispersed, with the highest median think time a factor of 20 or 30 larger than the lowest one for each bin. Second, the order of the traces varies too; for example, the SDSC SP2 trace exhibits the lowest think time median for bin 2, and the highest median for bin 4. Finally, some of the results are non-monotonic: in the SDSC Paragon trace, the think times following jobs in slowdown bin 4 are lower than those following jobs in bins 1, 2, and 3. Consequently, there is no easy and general way to characterize the relationship between the jobs' slowdown and the subsequent user behavior.

In the above, we considered the response time bins to be homogeneous but in reality, different jobs in the same bin may have far different wait and execution times. This means that the slowdown of the jobs within the same response bin exhibit a large variance. For example, a 15-minutes job that waited 5 minutes in the queue and a 1-minute job that waited 19 minutes both belong to the same response bin R3, since both responded in 20 minutes. The question is whether these two jobs indeed trigger a similar user behavior, even though the first has a slowdown of 1.33 and the second has a slowdown of 20.

To answer this question, we need to examine our response bins more closely. The natural way to do this is to simply divide each bin into sub-bins based on the slowdown metric. We used the same ranges as for the original slowdown bins.

Figure 2 illustrates these bins graphically. The horizontal axis represents the jobs' wait time, and the vertical axis represents their execution time. The response bins are bounded between the solid lines, and create diagonal regions going from top-left to bottom-right. The slowdown bins are bounded between the dashed lines, and create radial regions emanating from the origin. The intersections of the two types of regions represent the sub-bins. We named these sub-bins according to response and slowdown bins their jobs belong to, so for example sub-bin R3S1 holds jobs whose response time belongs to response bin R3, and whose slowdown belongs to slowdown bin S1, etc.

Table IV shows the median think time following the jobs in each of the sub-bins, using data combined from all five traces. As expected, this exhibits a strong dominant effect of the response times. In each column, we see that the median think

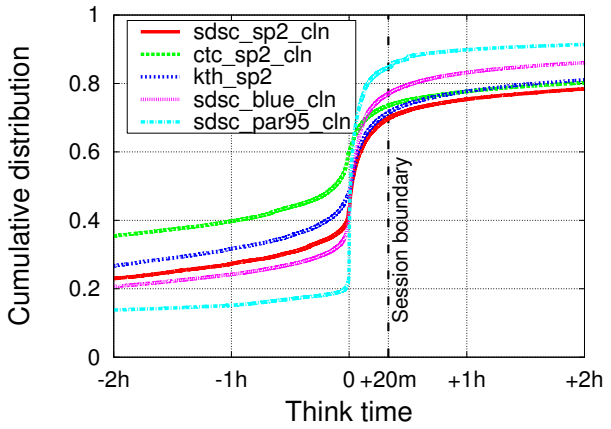


Fig. 3. CDF of think times for the five traces: Negative values indicate that one job started before the previous one completed. Session boundary is defined to be twenty minutes.

time grows dramatically and monotonically with the response-time bin. Looking at rows does not reveal any such pattern for slowdown. However, jobs with the very lowest slowdowns do consistently tend to lead to lower think times than the other jobs with the same response time.

The conclusion of the above discussion is that the response time metric is the one most important to the users since it has the strongest correlation with their subsequent behavior. The question that immediately follows is therefore *how* exactly response times affect the behavior. We show in the next section that it affects users' decision to continue or abort their interactive session with the system.

IV. Jobs Response Time and User Sessions

Sessions are periods of continuous activity by the users. This does *not* mean that their jobs must be continuously active throughout the session. A job may complete, and the user may think for a while before submitting the next job. If the think time is too long, on the other hand, it may well indicate that the user took a break. In this case, subsequent jobs will belong to a different session. The question is, therefore, how to distinguish between jobs that are separated by actual think times and belong to the same session, from jobs that belong to different sessions.

Zilber et al. answered this question by examining the distribution of think times in the traces [17]. Figure 3 reproduces part of their data (they used two additional traces in addition to our five), showing the CDF of the think times in the traces. We first observe that think times can be negative. This stems from the definition that think time is the time between the completion of a job and the submission of the next job, and indicates that sometimes users submit jobs without waiting for their previous jobs to complete. Such jobs are often submitted in *batches* — one after the other with very short gaps between the successive submissions, and without being affected by feedback from previous jobs [12]. Consequently, the negative think times are not useful for our study of feedback to the users, and we therefore ignore them for the rest of this paper.

Focusing on the positive think times, we see a steep climb in the CDF curve of all traces for think times of a few minutes, which levels off at about twenty minutes. This means that a large portion of the jobs are submitted within twenty minutes of the completion of a previous job, which is an indication for continuous activity periods by the users. Furthermore, beyond twenty minutes the think times are evenly distributed, without any features indicating a natural threshold. Zilber et al. therefore defined the sessions' think time boundary to be *twenty minutes*; jobs submitted after a think time that is longer than twenty minutes are considered to start a new session. In our work we adopt this definition.

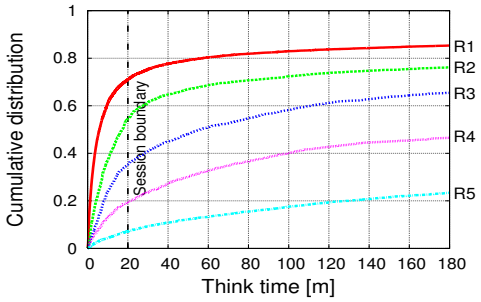
We remain focused on the response time bins of the previous section, but this time we expand our analysis, and consider not only the median think times, but the entire distribution of think times following the jobs in each bin

The CDF of think times for the different bins is shown in Figure 4. The immediate impression is that the five sub-figures that represent the different traces are very similar. In all traces, there is a noticeable and a similar gap between the CDF curves of the different bins. Furthermore, for all traces, the curves follow the same vertical order: response bin R1 has the highest CDF, bin R2 has the second-highest CDF, etc.

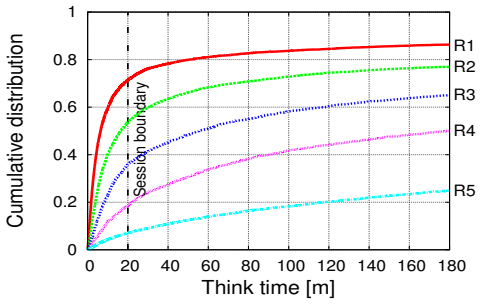
A closer examination of the figure also reveals that in all traces, all bins exhibit the same steep climb in the CDF up-to the session's twenty-minutes think time boundary, and beyond that point, all curves level off. In fact, the major difference between the curves is in the *percentage of the jobs that were submitted below the session boundary*, and it is this difference that determines the vertical order of the curves. For the SDSC-SP2 trace, for example, 72% of the subsequent jobs for response-bin R1 were submitted below the session boundary. For response bin R2, only 55% of the jobs were submitted below this boundary, and so on. The higher the bin number, the lower the percentage of jobs that were submitted below the session boundary.

Figure 5 summarizes these results for all five traces. For each bin we extracted the percentage of subsequent jobs that were submitted below the session boundary, using the CDF of think times of the bin. We also calculated the median of *response time* of the jobs in the bin, and used the median to represent the bin. We then plotted one against the other. The result is a *mapping* between the jobs response times, and the probability for the users to continue their sessions. We see that for all traces, the higher the response time of the jobs, the lower the probability for users to continue submitting jobs within the same session. The mapping itself is *non*-linear; the probability to continue the session initially drops rapidly as response-time increases, and continues to drop more slowly for higher response times.

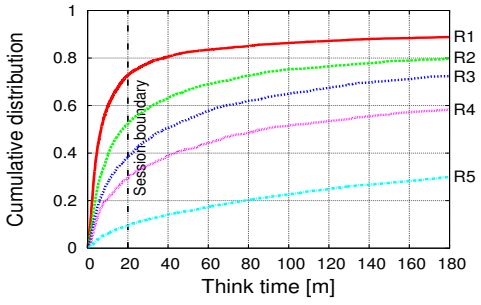
The conclusion is that the jobs' response times affect the users decision to continue or abort their interactive session with the system. In the next section we show that this decision may stem from expectations the users develop regarding the response time of their jobs.



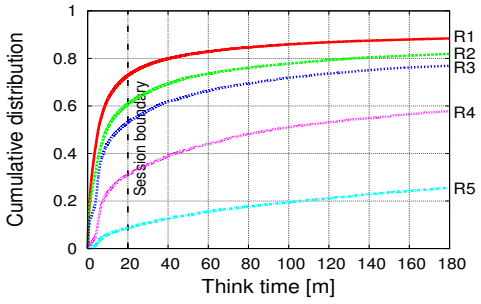
(a) SDSC-SP2



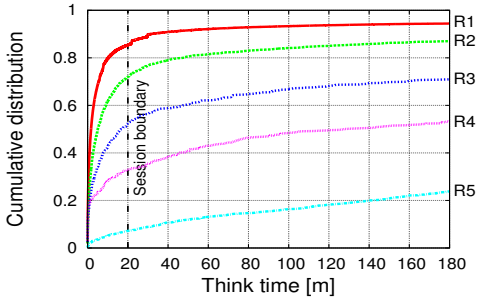
(b) CTC-SP2



(c) KTH-SP2



(d) SDSC-BLUE



(e) SDSC-PAR95

Fig. 4. CDF of think-times for the five response-bins: The five sub-figures that represent a different trace each are very similar. In all, the higher the bin number, the lower the percentage of jobs that were submitted below the twenty-minutes session boundary.

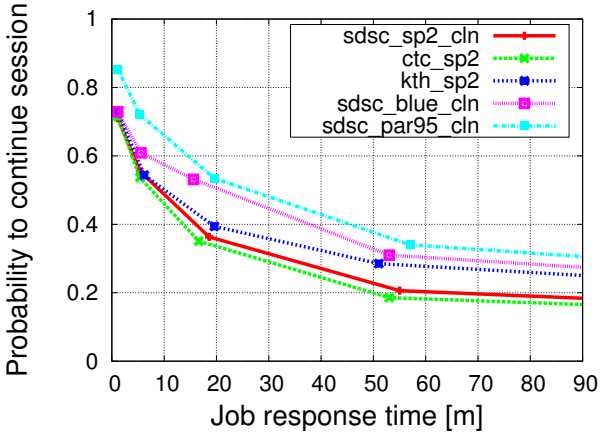


Fig. 5. Jobs' response-time effect on users' behavior: The higher the response time of the jobs, the lower the probability that users will continue submitting jobs within the same session. The mapping is *non-linear* and is highly similar for all traces.

V. User Performance Expectations

It is well known that user behavior is affected by expectations [13]. Live experiments conducted in the context of the web have found that users' tolerance to server delays is strongly affected by their expectations regarding the duration of the delay [1]. Accordingly, one would expect users of parallel systems to also develop expectations regarding the time frame by which their jobs should respond. The question is how these expectations affect their behavior, and what happens when response times lengthen beyond their expectations.

To answer this question we focus on a subset of our previous results. Specifically, we define two bins, so that one holds jobs that had a short wait, and the other holds jobs that had a short execution. Assuming that users expectations would be related to the *execution time* of their jobs, and not the wait times which are an artifact of certain conditions that existed in the system, these two bins actually represent two different scenarios: the one with the short waits represents the scenario where response times met their expectations, and the other represents the case where they expected a quick response, but it got lengthened because of long waits in the scheduler's queue. The threshold we chose for the bins is five-minutes of wait, and five-minutes of execution, respectively.

Due to the nature of the bins, the response times of the jobs they hold exhibit a large variance. The bin with the short waits for example, may hold two jobs that waited only a minute for execution, but the first executed for a few seconds, and the other for several hours. We therefore divided our bins into sub-bins, based on the response time metric, and using the same ranges we used before as indicated in Table II. This enabled us to examine the user behavior under the two scenarios, while also considering the effect of the response time of their jobs.

Figure 6 illustrates these bins graphically. Again, the horizontal axis represents the jobs' wait time, and the vertical axis represents their execution time. Our two bins are bounded between the two solid lines: the vertical bin holds the jobs with the short waits, and the horizontal bin holds the jobs with the short execution. The dashed diagonal lines represent

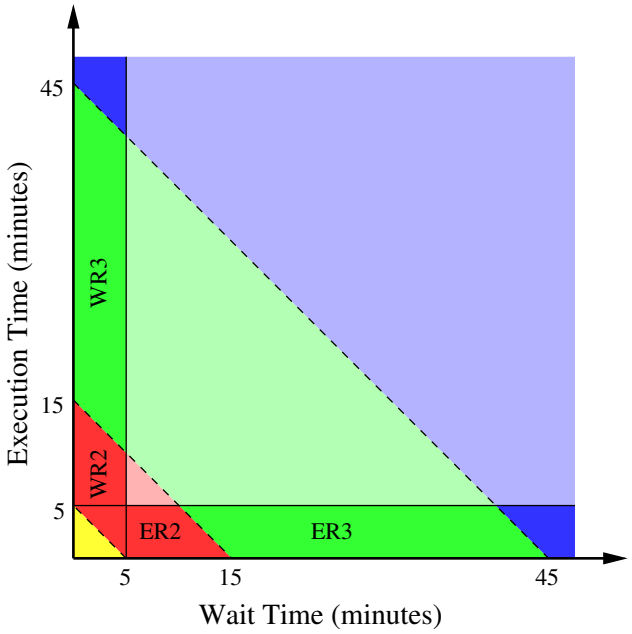


Fig. 6. Graphical illustration of the bins: The vertical bin holds the jobs with the short waits, and the horizontal bin holds the jobs with the short execution.

the boundaries of the response bins, which intersect with our main bins and define the sub-bins. The sub-bins with the short waits carry the prefix 'W', and the ones with the short execution carry the prefix 'E'. All carry a suffix that identifies the response bin to which their jobs belong to, e.g., R1, R2, etc.

For each sub-bin we extracted the percentage of jobs that were submitted below the session twenty-minutes boundary, and plotted this percentage against the median of response time of the jobs in the sub-bin. The result is shown in Figure 7. The two curves represent the probability for the users to continue their session as a function of the response time of their jobs. One curve represents the scenario where response times met their expectations, and the second represents the case where they did not.

In accordance with our previous results, we see that in both scenarios the probability for users to continue their session decreases as response time increases. What is surprising though is the high level of similarity between the curves, despite of the fact that they represent two essentially different scenarios.

Our proposed explanation to this difficulty is that the users' perception and motivation are indeed different in the two cases, but that their actual behavior just happens to be very similar. In the first scenario, response times meet users expectation. The fact that the probability to continue the sessions decreases as response times increase is then a straightforward result of the fact that users expect long response times, and therefore tend not to wait for their jobs. The longer they expect the response time to be, the higher the probability for them to discontinue their sessions.

In the second scenario execution times are short and users expect a quick response. They start to wait for their jobs, but when response times lengthens beyond their expectation

Fig. 7. Performance expectations and user behavior: The probability for users to continue their sessions decreases as the response time of their jobs increases, regardless of whether they expected that response or not.

because the jobs wait for long time in the scheduler's queue, they tend to lose their patience and abort their sessions. In this case the user behavior is indeed affected by the performance of the scheduler, but it happens in such a way that makes the end result appear similar to the behavior had they anticipated the long response time in advance.

We can also examine the bins with respect to the slowdown metric. In the bin with the short waits, slowdown *decreases* as response time increases. In the bin with the short execution times on the other hand, slowdown *increases* with the response time of the jobs. Still, the behavior of the users appears to be similar in both cases. This corroborates the results from Section III, and indicates that response time is a much more reliable predictor of user behavior than slowdown.

VI. Related Work

In the context of parallel system scheduling, we could not find any references that discuss the direct effect of the scheduler performance on the users' behavior. In fact, virtually in all the publications, the schedulers are evaluated in simulation using an open model, which means that the arrival rate of the jobs is already given, and is not affected by feedback from the scheduler [15], [10], [14], [16], [8], [11]. Works on modeling parallel workloads either try to mimic the arrival process found in workload traces [7], [2], [9], or simply use a Poisson model [5], [3].

There is no single consensus regarding the performance metric that is most important to users. More than 35 years ago, Brinch Hansen suggested to prioritize jobs using the slowdown metric [6], but since then many scheduling policies have been proposed and evaluated differently, as shown in Table V. All believe their metric is the most important, but none had ever investigated why.

In previous work, we investigated the importance of feedback from the scheduler to the generated workload for the accuracy of the evaluation, and suggested that the workload should be generated by user models that react to the response of their previous jobs [12]. In this model, think times for the jobs were drawn from the same distribution, irrespective

TABLE V
SCHEDULING POLICIES AND METRICS USED FOR THEIR EVALUATION: ALL BELIEVE THEIR METRIC IS THE ONE THAT IS MOST IMPORTANT TO USERS, BUT NONE HAD EVER INVESTIGATED WHY.

Year	Scheduling policy	Metrics
1999	Slack based backfilling [15]	Wait time
2001	EASY and conservative backfilling [10]	Response, Slowdown
2002	Selective reservation backfilling [14]	Slowdown, Turnaround
2002	Relaxed backfilling [16]	Total wait time
2002	Multiple queue backfilling [8]	Slowdown
2005	Lookahead based backfilling [11]	Response, Slowdown

of how previous jobs were handled by the scheduler. Consequently, the model didn't include any notion of aborting a user session as a result of poor performance.

Bouch et al. investigated users tolerance to web server delays and found that it is influenced by various factors, including the type of task they perform, and the cumulative time they interact with the server [1]. They used live experiments to extract the mapping between the duration of the delay, and the users perception of that delay.

Zilber et al. [17] examined the distribution of think times in traces of parallel system schedulers, and found that a large portion of the jobs were submitted within twenty-minutes of the completion of a previous job. Consequently, they defined sessions to be sets of jobs submitted within twenty minutes from the completion of previous jobs, a definition that we adopt in this paper.

VII. CONCLUSIONS

A good scheduler should increase the throughput of its users, but this requires an understanding of the users' behavior. Surprisingly, in virtually all performance evaluations, the effect of the scheduler on the users is ignored. The common methodology is to simply use a trace to generate the workload, but in the trace the arrival rate of the jobs is already given, and is not affected by feedback from the scheduler. Furthermore, the metrics by which schedulers are compared vary from one evaluation to the other. Each analyst believes their metric is the one that is most important to users, but this is not justified.

In this paper we investigated the effect of the performance of the scheduler on the behavior of the users, and found that their behavior correlates best with the response time of their jobs, not the slowdown as was previously sometimes assumed. We continued to investigate the actual type of the effect, and found that response times affect the users' decision to continue of abort their sessions, and that the higher the response times of the jobs, the lower the probability for users to continue submitting jobs within the same session. Finally, we've shown that the decision to abort the session may stem from certain performance expectation that the users develop, regarding the time frame by which their jobs should respond.

We did *not* reach these findings using live experiments. Instead, all we did was to examine traces that contain raw data on jobs that were submitted to real, production-use parallel systems. We are not the first to examine these traces. In fact, some of the older traces were first analyzed more than ten years ago. We are though the first to take a different, slightly less obvious look at things. An important conclusion of this work is that a lot of interesting observations are still out there in the traces. All it takes is different angle and a fresh way of thinking to extract them.

ACKNOWLEDGMENTS

This work was supported in part by the Israel Science Foundation (grant no. 167/03). Many thanks to all those who provided workload data to the Parallel Workloads Archive.

REFERENCES

[1] A. Bouch, A. Kuchinsky, and N. Bhatti. Quality is in the eye of the beholder: meeting users' requirements for internet quality of service. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 297–304, New York, NY, USA, 2000. ACM Press.

[2] W. Cirne and F. Berman. A comprehensive model of the supercomputer workload. In *4th Workshop Workload Characterization*, Dec 2001.

[3] A. B. Downey. A parallel workload model and its implications for processor allocation. *Cluster Computing*, 1(1):133–145, 1998.

[4] D. Feitelson. Parallel workload archive. http://www.cs.huji.ac.il/labs/parallel/workload/.

[5] D. G. Feitelson. Packing schemes for gang scheduling. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 89–110. Springer-Verlag, 1996. Lect. Notes Comput. Sci. vol. 1162.

[6] P. B. Hansen. An analysis of response ratio scheduling. In *IFIP Congress (1)*, pages 479–484, 1971.

[7] J. Jann, P. Pattnaik, H. Franke, F. Wang, J. Skovira, and J. Riodan. Modeling of workload in MPPs. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 95–116. Springer Verlag, 1997. Lect. Notes Comput. Sci. vol. 1291.

[8] B. G. Lawson and E. Smirni. Multiple-queue backfilling scheduling with priorities and reservations for parallel systems. *SIGMETRICS Perform. Eval. Rev.*, 29(4):40–47, 2002.

[9] U. Lublin and D. G. Feitelson. The workload on parallel supercomputers: Modeling the characteristics of rigid jobs. *J. Parallel & Distrib. Comput.*, 63(11):1105–1122, Nov 2003.

[10] A. W. Mu'alem and D. G. Feitelson. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. *IEEE Trans. Parallel Distrib. Syst.*, 12(6):529–543, 2001.

[11] E. Shmueli and D. G. Feitelson. Backfilling with lookahead to optimize the packing of parallel jobs. *J. Parallel Distrib. Comput.*, 65(9):1090–1107, 2005.

[12] E. Shmueli and D. G. Feitelson. Using site-level modeling to evaluate the performance of parallel system schedulers. In *MASCOTS '06: Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation*, pages 167–178, Washington, DC, USA, 2006. IEEE Computer Society.

[13] B. Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.

[14] S. Srinivasan, R. Kettimuthu, V. Subramani, and P. Sadayappan. Selective reservation strategies for backfill job scheduling. In *JSSPP '02: Revised Papers from the 8th International Workshop on Job Scheduling Strategies for Parallel Processing*, pages 55–71, London, UK, 2002. Springer-Verlag.

[15] D. Talby and D. G. Feitelson. Supporting priorities and improving utilization of the IBM SP scheduler using slack-based backfilling. In *IPPS '99/SPDP '99: Proceedings of the 13th International Symposium on Parallel Processing and the 10th Symposium on Parallel and Distributed Processing*, page 513, Washington, DC, USA, 1999. IEEE Computer Society.

[16] J. William A. Ward, C. L. Mahood, and J. E. West. Scheduling jobs on parallel systems using a relaxed backfill strategy. In *JSSPP '02: Revised Papers from the 8th International Workshop on Job Scheduling Strategies for Parallel Processing*, pages 88–102, London, UK, 2002. Springer-Verlag.

[17] J. Zilber, O. Amit, and D. Talby. What is worth learning from parallel workloads? a user and session based analysis. In *Proc. 19th intl. conf. Supercomputing*, pages 377–386, Jun 2005.