Hierarchical Dynamics, Interarrival Times, and Performance

Stephen D. Kleban
Sandia National Laboratories
Albuquerque, NM 87185 USA
1 505 844 0553
sdkleba@sandia.gov

Scott H. Clearwater
P.O. Box 620513
Woodside, CA 94062 USA
1 650 851 9510
clearway@ix.netcom.com

ABSTRACT

We report on a model of the distribution of job submission interarrival times in supercomputers. Interarrival times are modeled as a consequence of a complicated set of decisions between users, the queuing algorithm, and other policies. This cascading hierarchy of decision-making processes leads to a particular kind of heavy-tailed distribution. Specifically, hierarchically constrained systems suggest that fatter tails are due to more levels coming into play in the overall decision-making process. The key contribution of this paper is that heavier tails resulting from more complex decision-making processes, that is more hierarchical levels, will lead to overall worse performance, even when the average interarrival time is the same. Finally, we offer some suggestions for how to overcome these issues and the tradeoffs involved.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: *modeling techniques, performance attributes.*

General Terms

Your general terms must be any of the following 16 designated terms: Algorithms, Measurement, Performance, Experimentation, Theory.

Keywords

hierarchy, relaxation process, interarrival, ASCI queueing, dynamics.

1. INTRODUCTION

Supercomputers represent the largest single computing resources in the world and they must perform over a staggering range of conditions spanning small interactive jobs to very large jobs, both in terms of the number of processors involved (in the thousands) and for long time periods (on the order of a day for a single run). Like many phenomena found in nature, the interarrival time distribution is found to be heavy-tailed. Heavy-tailed distributions, defined as those that drop off more slowly than an exponential, have also been reported in a number of manmade phenomena, specifically computer systems. For example, the simplistic model of exponential arrival times has been shown to be inadequate for describing wide area network traffic[18]. Some examples of heavy tail distributions in computer systems include: computer networks both in terms of their connectivity[21] and their traffic patterns[22], file systems[7], video traffic[2], and software caches[20], and the job size distributions on a single processor[8] as well as supercomputers[5].

Heavy-tail distributions have important implications for both physical and manmade systems. In particular, heavy tails indicate a significant probability of very large deviations from average or "typical" events. In the case of earthquakes it means a meaningful chance for very large and damaging events. In the case of supercomputers it means the possibility that the machine may become overloaded for significant periods of time even if the average job submission rate is moderate. Significantly, the confluence of many large jobs impinging on a supercomputer and the droughts in between that occur as a consequence of heavy-tailed distributions in interarrival time, can have serious consequences on the timeliness of the important work done at these facilities.

Supercomputers typically have thousands of processors and perform the most sophisticated simulations on a wide variety of problems in material science, structural, and thermal dynamics. Among the most powerful supercomputers are those from the Advanced Simulation and Computing Initiative (ASCI)[1]. These machines were built for specific purposes to primarily serve a small group of users who end up dominating the cycles on the machine.

Despite the extensive work in networks and single processor systems, less is known quantitatively about the scaling behavior in the largest supercomputers[5, 8]. Do they in fact follow a power law, or is there a characteristic size to the distribution, meaning that the fraction of large events is influenced by the limits of the

SC'03, November 15-21, 2003, Phoenix, AZ, USA.

machines themselves as well as the administrative constraints? Given log data from thousands of jobs over a period of several months we can examine these issues quantitatively. We shall see that the complex and ongoing dynamics of the interplay between users, the facility managers who run the queues, and the machine itself come together in a way that qualifies it with the observed heavy-tail distributions.

2. SUPERCOMPUTER WORKFLOW

The workflow of jobs through a supercomputer system is an ongoing and complicated cycle of phases involving submission, dispatch, running, and analysis(Fig. 1). The “Big Iron” is the supercomputer. Note the closed-loop that indicates how the discharge of completed jobs from the machine influences the input to the queues much later. Not shown are the levels of management, both local and national, that also come into play on much longer time scales.

The “output” of each of these phases depends critically on the other phases. For example, the submission of a particular job at a particular time by a particular users depends on the time the user has to spend setting up the next run, the previous runs the user has to analyze, which in turn depended upon when it finished running on the machine, which depended upon when it was dispatched which depended upon the prioritization constraints imposed by the facility managers via the queuing system.

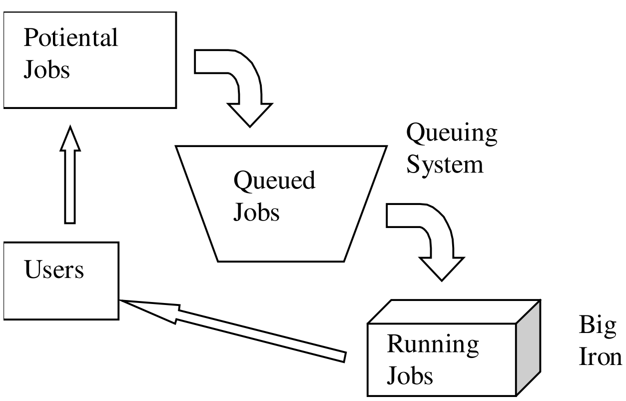

Fig. 1. Influx and discharge of jobs through a queuing system.

On top of all that, the large users of supercomputers often meet informally to decide which large jobs will run. Specifically, the user community itself acting sometimes independently, and sometimes in a coordinated fashion, submits jobs to the queueing system. For example, the individual user may be doing a parameter study and each job will be a variation of the other. Thus, the users themselves, with their requirements for computational resources represent a source of computational needs ultimately driven by the funding agencies. So, ultimately, the act of submitting a particular job at a particular time is seen as being the result of a conjunction of a large number of decisions of involving many different agents arranged in a hierarchy where the decisions at each level in the hierarchy are made at different rates. For example, the queuing algorithm may be making decisions on the order of tens of seconds while budget and program-level decisions are made on the time scale of a year or more. These decisions and conditions, in turn, are influenced by various constraints, such as, a finite number of processors and their mean time between failure, maximum run time limits(typically 24 hours/job), and various administrative limits imposed on the system to provide some sense of “fair use” to the users.

3. HIERARCHICAL DYNAMICS

To motivate our model for hierarchical decision-making in the submission of supercomputer jobs, we now discuss analogies to dynamical and statistical models developed to understand relaxation processes. While not necessarily exact, the models proposed below are meant to capture the essential dynamics that occur in the job submission process.

In the first model[10, 17], the role of the decision-making agents at each level is to ultimately make a yes/no decision. In physics, binary systems like these are modeled with up/down spin systems. The idea here is that the highest level of spins(higher-level management), work at the lowest rate and in turn drive what happens at the lower levels. As one goes down the hierarchy to the end user who actually submits the jobs, the decision-making process becomes faster and faster. When a certain number of spins at a given level are aligned in the same direction, say up, then a decision has been made and the next level in the hierarchy can then begin its decision-making process.

In the second model, the decision to submit a job is seen as the result of climbing a number of higher and higher energy barriers[6]. In this model the highest level of decision maker is at the bottom of the energy scale and takes the longest to make the decisions. As one climbs higher in energy, the steps between levels become smaller and therefore the rate at which decisions are made become quicker. The idea behind both these models is shown in Fig. 2.

The complicated set of conditions for determining which job gets submitted and when is exactly the sort of conjunctive, i.e., multiplicative, process that is described by the stretched exponential distribution[6]. The stretched exponential has been used to account for a number of natural phenomena such as radio emission from galaxies, oilfield reserved sizes as well as man-made phenomena such as certain market price variations and numbers of citations of published work[16]. It was originally developed by Kohlrausch[13] to explain the discharge in Leiden jars.

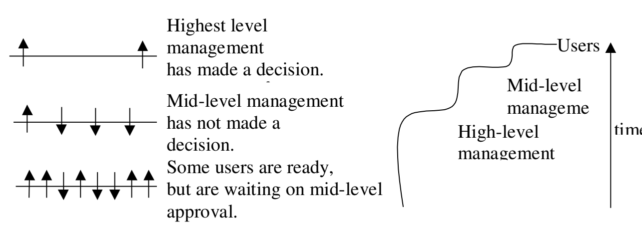

Fig. 2. (left)Hierarchical spin system of decision-making agents. (right)Series of energy barriers that must be climbed before submitting a job.

The probability distribution function for a stretched exponential, or Weibull, is given by

$$PDF(x) = \frac{\beta}{x_0}\left(\frac{x}{x_0}\right)^{\beta-1} \exp\left(-(x / x_0)^{\beta}\right)$$

where x_0 defines a characteristic scale to the distribution (contrast this with a power law's scale free behavior) and β is a measure of the heaviness of the tail. The smaller the value of β, the heavier the tail. For β→0 we recover a power law and for β→1 we recover the exponential distribution. The form of the cumulative distribution is particularly clean,

$$CDF(x>X) = \exp(-(x/x_0)^{\beta}).$$

One intuitive derivation of the stretched exponential can be found by considering a product of i.i.d. positive random variables,

$$PDF = \prod_{i=1}^{n} pdf_i \ .$$

If pdf_i is $\exp(-ax^{\alpha})$ with $\alpha > 0$ then *PDF* will be a stretched exponential[6]. Note that a product of i.i.d. distributions can also lead to a lognormal distribution. However, as we are concerned with the tails, the assumptions that lead to a lognormal distribution from a product of distributions breaks down because the central limit theorem does not apply far from the peak of the distribution[6].

The interpretation of the significance of x_0 in terms of more familiar quantities is given in terms of the average value of x, $\bar{x}$,

$$\bar{x} = \frac{x_0}{\beta}\Gamma(1/\beta)$$

where $\Gamma(z)$ is the usual gamma function. When $\beta \to 1$, then $\bar{x} \to x_0$. When $\beta \to 0$, then $\bar{x} >> x_0$ because the distribution becomes heavy-tailed. The interpretation of β is more difficult, but as we shall argue later may be related to the hierarchical structure that underlies the process.

In the analysis that follows we will be looking at the interarrival times of jobs. Heuristically speaking from the standpoint of the above derivation, the pdf_i's may be such things as the steps mentioned above and each has a distribution associated with it and this feeds into the product that eventually leads to stretch exponential. For example, the time needed to analyze results will depend in part on the complexity of results themselves as well as distractions like phone calls, meetings, and so on, all of which influence the time at which the next job may be submitted. These constitute the multiplicative steps in the formulation of the stretched exponential.

4. The ASCI Queueing Algorithms

In this paper we analyze job logs from the ASCI supercomputers ASCI-BlueMountain (at Los Alamos National Laboratory, LANL, with 6144 CPUs), and ASCI-BluePacific (at Lawrence Livermore National Laboratory, LLNL, with 972 CPUs)[1]. Each lab has devised its own method for queuing jobs based in part on the historical political realities at each lab.

The ASCI queueing algorithms have been described elsewhere[3]. Briefly, the Load Sharing Facility(LSF)[15] on Blue Mountain and the Distributed Production Control System(DPCS)[4] on Blue Pacific are both so-called "fair share" schemes[9]. In fair share algorithms, a user, or group of users is allocated a certain "share" of an allocation which may be part of an entire share hierarchy of groups. In other words, the share hierarchy of the queuing system becomes part of the decision-making hierarchy we have been talking about.

The share is a scale factor that goes into determining a user's or group's priority in addition to other factors. For example, when a user or group member submits a job or one is dispatched to run, depending on the algorithm, the priority is decreased according to some function. All other jobs submitted by that user or that user's group will have their priority affected by this activity as well. Once the job runs, the priority is gradually restored to its original value. When there are not enough processors for the highest priority job to run immediately, smaller jobs are "backfilled"[19] to make use of the spare processors until enough processors are freed for the big job to run.

In any event, the important thing to keep in mind is that the queuing algorithm through its prioritization and backfilling, acts to alter the order that jobs were submitted and thus when they will be dispatched, run, and finally analyzed, all affecting the next job to be submitted and thus the distribution of interarrival submission times as part of the loop shown in Fig. 1.

5. Analysis

For all the analysis shown below we tried to fit other distributions such as the exponential, lognormal, and power law functions, but none provided as good a fit and over such a long range as the stretched exponential. Qualitatively, the exponential fell off more rapidly than the data and the power law not fast enough. The lognormal fit well for smaller values, but did poorly at larger values, as one would expect from its functional form. From the heuristic argument given earlier, we might expect the stretched exponential to be applicable and fill in this intermediate range with a moderately heavy tail and a characteristic scale.

5.1 Blue Mountain

Blue Mountain at Los Alamos has 6144 processors, although only 5418 were available for the production queues we studied. Blue Mountain is partitioned into 2 sections, one for smaller jobs and the other for larger jobs.

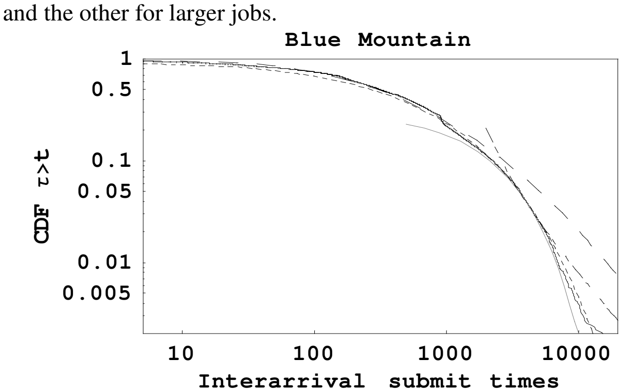

Fig. 3. The cumulative distribution function for the time between job submissions.

From the standpoint of our analysis, these can be considered as two separate machines. Since we are most interested in studying large jobs, we will focus on jobs from the large partition. This left us with 8171 jobs out of an original total of 86,403 jobs, which constituted 88% of the usage on the entire machine. The average number of processors was 415, the average run time was 8920 sec, and the average job size was 3.3×10^6 CPU-sec. The distribution of interarrival times is shown in Fig. 3. The best fit (short dashes) to a stretched exponential is shown with a characteristic time of t_0 = 524 sec and β = 0.57. The average is 879 sec and the maximum is 161,311 sec. The solid curve is the data, the dashed are from a stretched exponential fit. The long dashed curve is a best fit to a lognormal, the gray curve is the best fit to an exponential, and the dot-dash is a power law over a restricted region. As can clearly be seen, only the stretched exponential is able to model the data well over its entire range.

5.2 Blue Pacific

In this section we show the results from Blue Pacific consisting of 57,430 jobs. The average number of processors was 71, the average run time was 3,511 sec, the average job size 372 kCPU-sec, and average job interarrival time of 391 sec. Unlike Blue Mountain, Blue Pacific did not have any partitions and used about 1000 CPUs, although the full machine has more. Fig. 4 shows the cumulative probability distribution function for interarrival times. The solid curve is the log data; the dashed curve is the fitted stretched exponential. We have truncated our fit at 10,000 seconds because beyond that time the interarrival times are likely due to system issues and not user issues. For example, these events may correspond to outages in the machine or logging errors (about 10% of the log had bogus entries and was not used) that could anomalously effect the very long portion of the tail. For interarrival times up to 10,000 sec and using the 15% largest jobs, the parameters for the stretched exponential are a characteristic time of t_0 = 1659 sec and β = 0.61.

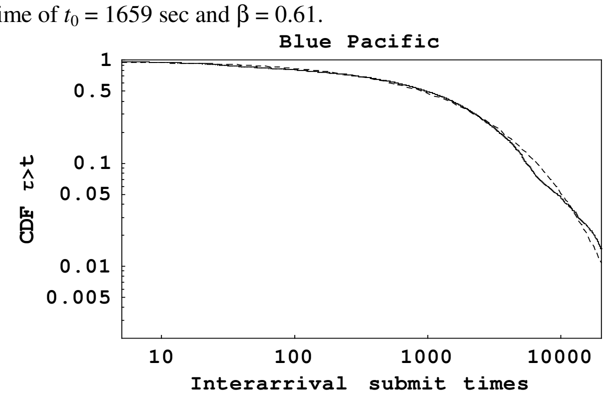

Fig. 4. The cumulative distribution function for the interarrival time between job submissions.

5.3 Hierarchical Dynamics

In this section we discuss some of the implications of the value of the stretched exponential exponent, β. As mentioned earlier, a smaller β implies a heavier tailed distribution and is therefore subject to larger fluctuations than those with lighter tails. According to [6], β is related to the number of levels in a hierarchy of levels, n, by the relation:

$$\beta = \alpha / (\alpha + n)$$

where α is related to the rate that each level makes its decisions. In principle, we could estimate α from a trace of the decision-making process, but it is unlikely that this record is available or that it could be interpreted properly. However, it is probably the case that the effective number of levels is only a few. From [10], we have the relationship between the characteristic times that decisions are made at a given level n:

$$\tau_{n+1} = 2^{\mu_n} \tau_n$$

where μ_n is the number of decision-makers at a given level. If we consider the faster decision to be on the scale of the average time between jobs, that is about 15 minutes, i.e., O(hour), and at the next highest level suppose the major users decide on the time scale of O(day), at the next level perhaps a week or so, then at the level of several months, and so on., then these ratios imply that μ_n ≈ 3-6, and given that $\mu_0 \approx \alpha$ [6, 17], we have an estimate for n as well. For simplicity's sake we will take the smallest integer value for n and α corresponding to β = 0.57 (Blue Mountain). Thus, we fix α = 4 and so the effective number of levels is n = 3, then. Larger values of α would imply that β is less sensitive to changes in the number of levels. We can now study the effects of a different number of levels in the decision-making hierarchy by fixing α and varying n, as shown in Table 1.

Table 1. Relationship between β and number of levels, *n* for α=4.

β	*n*
4/6 = 0.66	2
4/7 = 0.57	3
4/8 = 0.50	4

By generating synthetic jobs based on the actual log data as discussed in [11], we can study the effect of different values of β. The queuing algorithm used in the simulation is First-Come-First-Served with backfilling. This allows us to focus on the differences due to the interarrival distribution, rather than the idiosynchracies of the queuing algorithm.

We preserve the average time between submitted jobs by normalizing the value of t_0 through the relation:

$$t_0 = \frac{\beta}{\beta_{ref}} t_{0ref} \frac{\Gamma(1/\beta_{ref})}{\Gamma(1/\beta)}.$$

where t_{0ref} = 524 sec and β_{ref} = 0.57 (both from Blue Mountain). By normalizing the average time between jobs we isolate the effect of the fluctuations due to the tail of the distribution. In other words, the fluctuations correspond to the degree of complexity of the decision-making process. From the viewpoint of the decision-making process this makes sense, as we would expect different organizations to be operating under the same deadlines, but have different means of arriving at those decisions. Further, by preserving the mean and varying the heaviness of the tail, we are also affecting the short times as well. Thus heavy tails also have sharper peaks (for the same average of the distribution) and this peaking of job interarrival times is what ultimately leads to the decrease in performance. This is what causes jobs to pile-up. Table 2 shows the peak and tail probabilities for the values of β we studied. The significant peak for lower values of β is also shown in Fig. 5. The curves are for values of β=0.1,0.2, etc. to

1.0(dashed). The straightest lines are for smaller values of β (sharp peak and long tail).

Table 2. Peak and Tail Probabilities

β	Prob(< 60 sec)	Prob(>3600 sec)
0.50	0.314	0.054
0.57	0.252	0.050
0.66	0.191	0.042

The sharp peak and long tail are a twofold nuisance as far a performance goes because the sharp peak at short interarrival times leads to jobs piling up in the queue, while long times between interarrival intervals mean that work isn't being submitted. In whichever case, work is not getting done, either because of resource conflicts with others or one's own self-imposed delays.

The interpretation of these results in light of their relationship to the number of levels is instructive. First we define the expansion factor, which is a measure of the increased time that the job takes to finish due to its waiting to run and is defined as,

$$EF = \frac{finish - submit}{run} = 1 + \frac{wait}{run}.$$

From a simulation of jobs, for both wait times and expansion factor we see that heavier-tails, lower β, lead to increased probability of long wait and higher expansion factors (see Table 3 and Figs. 6 and 7). Interpolating the results for β=0.61 (Blue Pacific) we see that the average wait time is 23½ minutes longer than for β=0.57 (Blue Mountain). Taken over a period of months with many thousands of jobs, this is a significant effect for users and the facility.

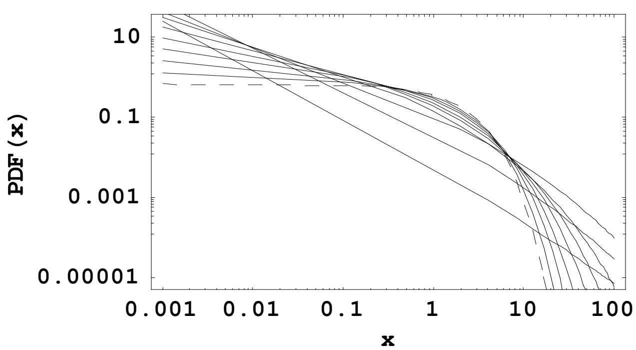

Fig. 5, Theoretical PDF of the stretched exponential using an x_0=1 and normalizing to β=0.6.

The interesting implication is that these results suggest that the more levels there are in a hierarchy, the greater the chance for a large fluctuation in the decision making process that goes into ultimately submitting a job. So even though the *average* time to make a decision is the same in the simulated cases we studied, it is the *spread* in those times that leads to noticeably longer wait times. The lesson from this analysis is that the decision-making process should be streamlined in order to reduce these fluctuations and thereby improve the turnaround of jobs.

Table 3: Relation between β and wait times and expansion factors.

β	avg. wait(sec)	median wait(sec)	avg. EF	median EF
0.50	24,514	11,970	68	4.1
0.57	20,417	10,320	58	3.5
0.66	17,238	8,310	50	3.0

It is also interesting to note that the difference in the queuing algorithms between BlueMountain and BluePacific may account, in part, for the difference between β_{BlueMt} =0.57 and $\beta_{BluePac}$ =0.61. For example, DPCS involves more complicated sets of conditions that on the one hand may seem to correspond to more "levels" in the decision-making process. On the other hand, this greater degree of automation may serve to obviate levels of human decision-making that are much more prone to fluctuations in their decision-making than is a queuing algorithm. If true, this argues for more sophisticated queuing algorithms to remove the degree of fluctuation implied when human decision-making is involved.

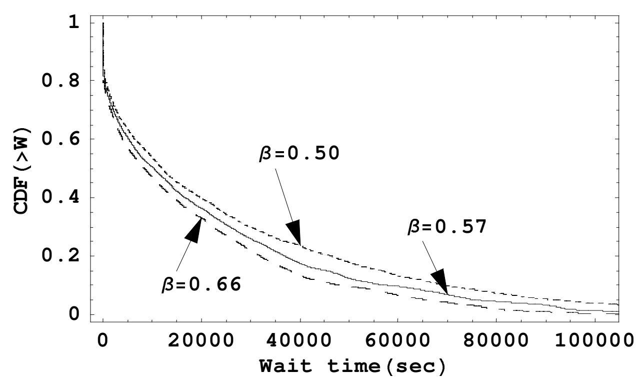

Fig. 6.CDF for the wait time for different values of β.

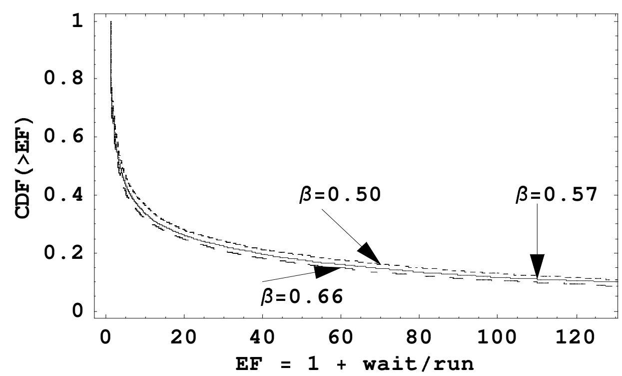

Fig. 7. CDFs for Expansion Factor for different values of β.

6. Discussion

We have shown that the interarrival time of jobs to the machines are not exponential nor do they posses pure power-law tails, but are somewhere in between and can be fit by stretched exponentials (Weibulls) over a large and important part of their range. These are indicative of finite scaling behaviors and have implications for the ultimate performance of these facilities because they relate to the frequency, and therefore the turnaround, of big jobs that are the bread and butter of the ASCI supercomputers.

Also, we note that Blue Mountain and Blue Pacific utilize a Fair Share algorithm[19] so it will be interesting to see, when data becomes available, if another queuing algorithm, such as NQS[16] at ASCI Red at Sandia has a similar characteristic exponent for job sizes and job interarrival times.

The characteristic scale implied by the stretched exponential distribution may prompt another look at some computer phenomena previously thought to exhibit scale-free behavior. The characteristic scale implied by a stretched exponential tells us that the deviations from a power law are a fundamental part of the phenomena[14]. After all, as big as these supercomputers are, they are still finite and their operators have put in additional constraints as well to satisfy administrative requirements, i.e., political realities. Together these constraints act to define a characteristic size of the distribution as well as the heaviness of the tail.

Through a simulation we showed that the performance of the supercomputer, as measured by wait time and expansion factor, was quite sensitive to the heaviness of the tail as parameterized by β. We also found that adding or removing a layer of decision-making is significant to the performance. Specifically, we found that the heaviness of the interarrival job submission tail can be explained by fluctuations due to the number of levels in a decision-making hierarchy. According the models presented here, more levels have fatter tails and worse performance due to the larger degree of fluctuations present with more levels. To overcome this problem, either the decisions have to be made more quickly, or the number of levels in the hierarchy needs to be reduced.

While our results provide support for the important role of the hierarchy in performance we should point out that like astronomers and economists we cannot perform direct experiments to test our hypothesis, we can only observe the phenomenon. Another point of interest is that our hierarchical model only considers marginal distributions. That is, we have not investigated the effect of burstiness through correlations among agents at the same level in the hierarchy. Generally speaking, correlations and the resulting burstiness leads to poorer performance[11].

Nevertheless, our results provide ammunition for those seeking to streamline the entire practice of using supercomputers. To the extent that a queuing system reduces the need for human intervention and the concomitant fluctuations in those interventions, more automation in the queuing process will improve performance. We should point out that some of the interventions involve fairness conditions and that these notions of fairness may need to be modified for the sake of other performance metrics.

In future work we will extend our work to include other distributions such as dispatch time that are more directly influenced by the queuing algorithm.

7. Acknowledgements

The authors gratefully acknowledge the assistance of Stephany Boucher, Charles Hales, Michael Hannah, Steve Humphreys, Moe Jette, Wilbur Johnson, Tom Klingner, Jerry Melendez, Amy Pezzoni, Randall Rheinheimer, Phil Salazar, Bob Wood, and Andy Yoo. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

8. References

[1] http://www.llnl.gov/asci/

[2] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", *IEEE Transactions on Communications*, vol. 43, no. 2/3/4 Feb./Mar./Apr. 1995, pp. 1566-1579.

[3] S. H. Clearwater and S. D. Kleban, "ASCI Queuing Systems: Overview and Comparison", *Proc. 16th International Parallel & Distributed Processing Symposium* (IPDPS 2002).

[4] DPCS Manual (unpublished) and Moe Jette and Bob Wood, private communication, 2001-2002.

[5] D. Feitelson "Sensitivity of Parallel Job Scheduling to Fat-Tailed Distributions" Technical Report 2000-44, Inst. Computer Science, The Hebrew University of Jerusalem, Oct 2000 and http://www.cs.huji.ac.il/~feit/pub.html.

[6] U. Frisch and D. Sornette, "Extreme Deviations and Applications", *Phys. I France*, vol. 7, 1997, pp. 1155-1171.

[7] S. D. Gribble, G. S. Manku, D. S. Roselli, E. A. Brewer, T. J. Gibson, and E. L. Miller, "Self-Similarity in File Systems", *Measurement and Modeling of Computer Systems*, 1998, pp. 141-150.

[8] M. Harchol-Balter, "The Effect of Heavy-Tailed Job Size Distributions on Computer System Design", *Proceedings of ASA-IMS Conference on Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics*, June, 1999.

[9] J. Kay and P. Lauder, "A Fair Share Scheduler" *Communications of the ACM*, vol. 31, no. 1, Jan. 1988, pp. 44-55.

[10] J. Klafter and M. F. Shlesinger, "On the relationship among three theories of relaxation in disordered systems" *Proceedings of the National Academy of Sciences*, vol. 83, Feb. 1986, pp. 848-851.

[11] S. D. Kleban and S. H. Clearwater "Quelling Queue Storms" *12th International Symposium on High-Performance Distributed Computing* (HPDC-2003), IEEE CS Press, Los Alamitos, CA, 2003, pp. 162-171.

[12] S. Kleban and S. Clearwater. "A Market-Based Architecture for Supercomputer Resource Management" *Proceedings of the ACM/IEEE Super Computing 2001* (SC-2001), 2001.

[13] R. Kohlrausch, "Theorie des electrischen Rückstandes in der Leidner Flasche" Pogg. Ann. Phys., vol. 91, 1854, pp.179-214 (in German).

[14] J. Laherrere and D. Sornette, "Stretched exponential distributions in Nature and Economy: 'Fat tails' with characteristic tails", *European Physics Journal B*, vol. 2, 1998, pp. 525-539.

[15] "LSF Batch Administrator's Guide, Sixth Edition", Platform Computing Corporation, 1998.

[16] http://umbc7.umbc.edu/nqs/nqsguide.html

[17] R. G. Palmer, D. L. Stein, E. Abrahams, and P. W. Anderson, “Models of Hierarchically Constrained Dynamics for Glassy Systems” *Phys. Rev. Lett.* vol. 53, no. 10, Sep. 1984, pp. 958-961.

[18] V. Paxson and S. Floyd, “Wide-area traffic: The failure of Poisson modeling”, *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, 1995, pp. 226-244.

[19] J. Skovira, J. W. Chan, H. Zhou, D. and Lifka, “The EASY-LoadLeveler API Project”, D. G. Feitelson and L. Rudolph, (eds.), *Proceedings of 2nd Workshop on Job Scheduling Strategies for Parallel Processing*, vol. 1162, Springer Verlag, 1996, pp. 41-47.

[20] J. Voldman, B.B. Mandelbrot, L.W. Hoevel, J. Knight, and P. Rosenfeld, “Fractal Nature of Software-Cache Interaction”, *IBM Journal of Research and Development*, vol. 27, no.6, Nov. 1981, pp. 164-170.

[21] W. Willinger, R. Govindan, S. Jamin, V. Paxson, and S. Shenker, “Scaling phenomena in the Internet: Critically examining criticality”, *Proceedings of the National Academy of Sciences*, vol. 99, Feb. 2002, pp. 2573-2580.

[22] W. Willinger, M. Taqqu, and A. Erramilli, “A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks”, *Stochastic Networks*, F. P. Kelly, S. Zachary and I. Ziedins (eds.), Oxford University Press, Oxford, 1996, pp. 339-366.