A Comparison of Workload Traces from Two Production Parallel Machines

Kurt Windisch, Virginia Lo,
Dror Feitelson,
Bill Nitzberg, Reagan Moore

CIS-TR-96-06
April 1996

Department of Computer and Information Science
University of Oregon

A Comparison of Workload Traces from Two Production Parallel Machines

Kurt Windisch, Virginia Lo
kurtw, lo@cs.uoregon.edu
University of Oregon

Dror Feitelson
feit@cs.huji.ac.il
Hebrew University

Bill Nitzberg
nitzberg@nas.nasa.gov
NASA Ames Research Center

Reagan Moore
moore@sdsc.edu
San Diego Supercomputer Center

Abstract

The analysis of workload traces from real production parallel machines can aid a wide variety of parallel processing research, providing a realistic basis for experimentation in the management of resources over an entire workload. We analyze a five-month workload trace of an Intel Paragon machine supporting a production parallel workload at the San Diego Supercomputer Center (SDSC), comparing and contrasting it with a similar workload study of an Intel iPSC/860 machine at NASA Ames NAS. Our analysis of workload characteristics takes into account the job scheduling policies of the sites and focuses on characteristics such as job size distribution (job parallelism), resource usage, runtimes, submission patterns, and wait times. Despite fundamental differences in the two machines and their respective usage environments, we observe a number of interesting similarities with respect to job size distribution, system utilization, runtime distribution, and interarrival time distribution. We hope to gain insight into the potential use of workload traces for evaluating resource management polices at supercomputing sites and for providing both real-world job streams and accurate stochastic workload models for use in simulation analysis of resource management policies.

1 Introduction

Message-passing parallel machines are now being used to support diverse workloads in rich multi-user environments. In such environments, demand for system resources often exceeds those available, making careful resource management (i.e. job scheduling and processor allocation) an important concern. There is increasing interest in using analyses of the workload traces from these production parallel machines in the design and tuning of resource management algorithms. The study of workload traces has the potential to:

- **help administrators of parallel supercomputing sites evaluate and refine the resource management policies and algorithms for these machines.** This was done by Hotovy in [4], in which job scheduling policies for an IBM SP-2 at the Cornell Theory Center were modified in response to observed workload trends.
- **provide real-world job streams for direct use in more realistic simulation analysis of scheduling and allocation algorithms than is possible with purely stochastic workload models.** This was done by Suzuoka, Subholok, and Gross [7], who compared the performance of several scheduling mechanisms executing a set of jobs extracted from a workload trace of an Intel Paragon at ETH Zürich.
- **provide a basis for the development and validation of more realistic stochastic workload models for use in simulation analysis of scheduling and allocation algorithms.** In [2], Feitelson was able to create an accurate workload model based on observed workload characteristics from six separate supercomputer sites.

As a first step in this direction, we compare the characteristics of workload traces from two distinctly different machines. The first is an Intel iPSC/860 at NASA Ames Numerical Aerodynamic Simulation lab (NAS). The NAS machine is a 128-node hypercube. The three-month workload trace, taken from October 1 to December 31, 1993, was analyzed in detail by Feitelson and Nitzberg [3] and we use this analysis as a basis for comparison. The second machine in our comparison is an Intel Paragon at the San Diego Supercomputer Center (SDSC), containing 416 mesh-connected nodes. The mesh network has the dimensions 16×28, with 32 mesh slots unfilled. The five-month workload trace we analyze was recorded from December 1, 1994 to April 30, 1995.

While we recognize the inability to make generalizations from only two traces, we expect to gain insights that may help guide further investigation. Yet, because of the fundamental differences between the two machines, we hope to gain insight from any commonalities between the two workloads.

We have reproduced many of the observations and graphs from [3]; however, referring to the full NAS iPSC/860 technical report is helpful in understanding our comparison.

This paper is organized similarly to the NAS iPSC/860 paper [3]. Section 2 contains background on the NAS and SDSC scheduling policies. Sections 3 through 8 describe job size distributions, resource usage, system utilization, multiprogramming levels, runtimes, and submission characteristics. Section 9 concludes by summarizing the comparison of the two traces and suggests further studies.

2 Scheduling and Allocation Policies

When studying the characteristics of a workload trace, it is necessary to be aware of the scheduling policies of the machine which produced it. Although the NAS and SDSC machines that we study differ in many important ways, they share the same general scheduling strategies.

Both machines have comprehensive scheduling systems with two distinct subsystems from which the user can choose to schedule a job: interactive or batch. An interactive job is submitted directly to a specific physical partition of the machine. If and only if there are enough available processors in the interactive partition, the job is run, otherwise, the job is denied and must be resubmitted later. Interactively scheduled jobs are referred to as **direct** jobs. In addition, the administration of both machines allows for specially arranged and scheduled dedicated use by a single user.

The other physical partition of these machines, the batch partition, is governed by the Network Queuing System (**NQS**) [1]. The NQS batch partitioning system supports heterogeneous scheduling

time limit	number of nodes			
	16	32	64	128
20 minutes	q16s	q32s	q64s*	q128s*
1 hour	q16m	q32m	q64m*	q128m*
3 hours	q16l*	q32l*	q64l*	q128l*

* served during non-prime-time only

Table 1: NAS's NQS scheduling queue structure.

time limit	node memory	
	16MB	32MB
1 hour	q(2^n)s	qf(2^n)s
4 hours	q(2^n)m	qf(2^n)m
12 hours	q(2^n)l	qf(2^n)l
12 hours	standby	fstandby

(2^n) = the maximum number of nodes
($0 \leq n \leq 8$, though not all combinations exist)

Table 2: SDSC's NQS scheduling queue structure.

queues based on any of several characteristics: maximum runtime, maximum job size, and node type (16 megabyte regular nodes or 32 megabyte "fat" nodes in the SDSC machine). There are also two special low priority queues in the SDSC machine, *standby* and *fstandby* which have a maximum time limit of 12 hours [6]. Tables 1 and 2 show the queues available on each machine.

In both systems, scheduling policies change for non-prime-time periods to better serve the batch-oriented, nighttime workloads. In the NAS trace, prime-time scheduling was in effect Monday - Friday, from 6am to 8pm. During NAS prime-time, the only NQS queues served were those for small jobs (up to 32 nodes, 1 hour or less) and 64 compute nodes were reserved for interactive use. During non-prime-time, all nodes and queues were available for NQS use [3]. In the SDSC trace, prime-time was Monday - Friday, from 9am to 5pm. The number of nodes reserved for interactive or batch use remains constant between SDSC prime-time and non-prime-time, however, the set of nodes assigned to each NQS queue is reconfigured for non-prime-time (and these sets are not required to be disjoint) [8]. At the time of this study, 416 nodes in the SDSC Paragon were partitioned statically: 48 nodes were dedicated to the interactive partition, 352 nodes to the NQS compute partition, 6 nodes to the service partition, and 10 nodes to the I/O partition.

The NQS scheduler selects the highest priority job from among the primary queues shown in the table. Job priorities are calculated within each job queue utilizing an aging factor. The SDSC scheduler maintains special standby queues, *standby* and *fstandby*, from which jobs are selected when the regular queues are empty. The physical nodes are allocated to a given job corresponding to the actual number of nodes requested. In SDSC's Paragon, jobs need not request nodes numbering an exact power of two, but in the NAS iPSC/860, a hypercube topology, only power of two-sized subcubes may be requested.

Allocation of processors to a job in the NAS machine is restricted to contiguous subcubes. On the SDSC machine, requested nodes are allocated from the sets of nodes associated with the job's queue,

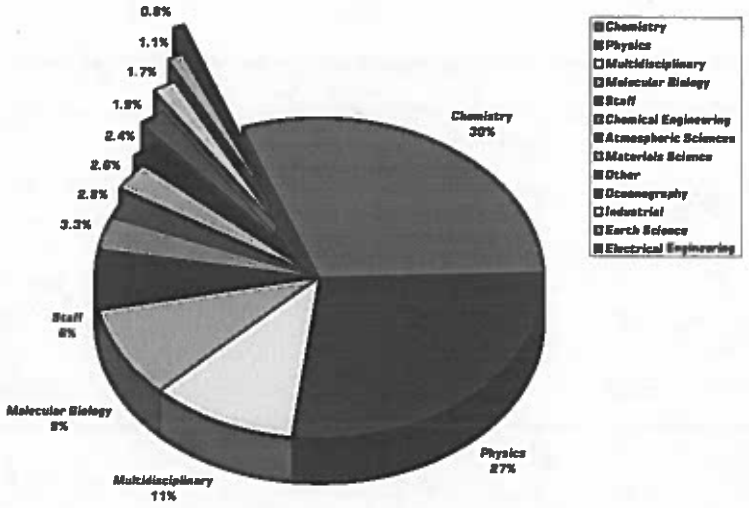

Figure 1: The application mix of the SDSC Paragon is very diverse.

using an extension to the 2-D Buddy Strategy [5][8]. An SDSC job may be allocated several blocks of processors which are not necessarily contiguous in the communication network. The first and largest block to be allocated is the anchor block. Subsequent blocks are chosen to be as close as possible to the anchor block, where *closeness* is the sum of the distances between the corresponding corners of the anchor block and the new candidate block.

3 General Job Mix

Number of Jobs

The SDSC trace recorded 33,283 jobs over a period of 5 months, an average of 6657 jobs per month. This is comparable to the equivalent job count from the NAS trace when numerous jobs executing the UNIX `pwd` command (for node fault detection) are omited: 18,037 jobs over a 3 month period, or 6,012 jobs per month on the average. Given the different machine sizes, the similarity in average number of jobs per month is a very interesting result.

Application Mix

One very significant difference between the job stream of the NAS and SDSC machines is the mix of applications which compose it. Figure 1 shows that of the jobs run on the SDSC Paragon, no single scientific field or application dominates the machine's usage. This large diversity of applications is in great contrast to the usage of the NAS machine, for which the user's applications were practically all in the domain of computational fluid dynamics.

Job Size Distribution - Sequential vs. Parallel Jobs

One of the most important general characteristics of a workload is the distribution of job sizes. This is shown in Figures 2 and 3 for the NAS and SDSC respectively. Both plots show the number of jobs for each job size broken into day, night and weekend. Day jobs are those that start between the hours of 6am-8pm Monday - Friday for the NAS trace, or 9am-5pm Monday - Friday for the SDSC trace. Night jobs are those that start during the remaining hours of the day on Monday - Friday, and weekend jobs are those that start at any time on Saturday or Sunday.

Job sizes in the SDSC trace are distributed similarly to the NAS trace with a few important differences. First, the SDSC trace does not have the huge number of single-node jobs that the NAS trace has

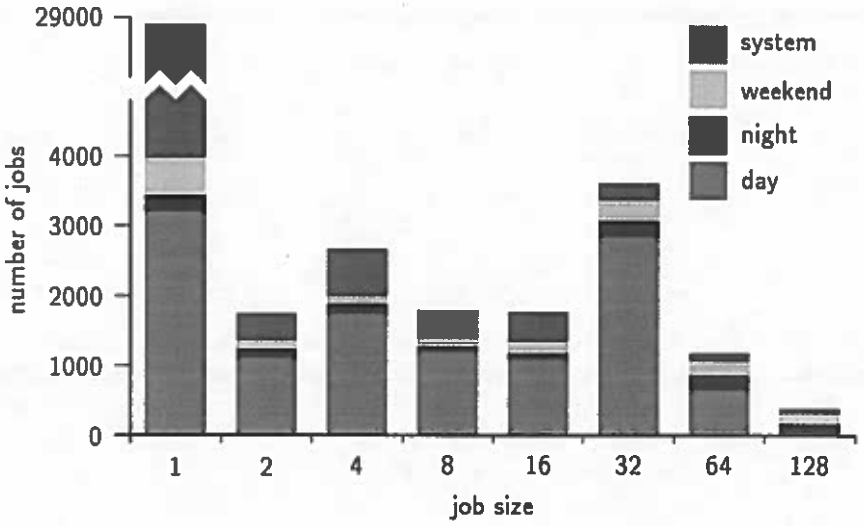

Figure 2: NAS job size distribution, classified by time period.

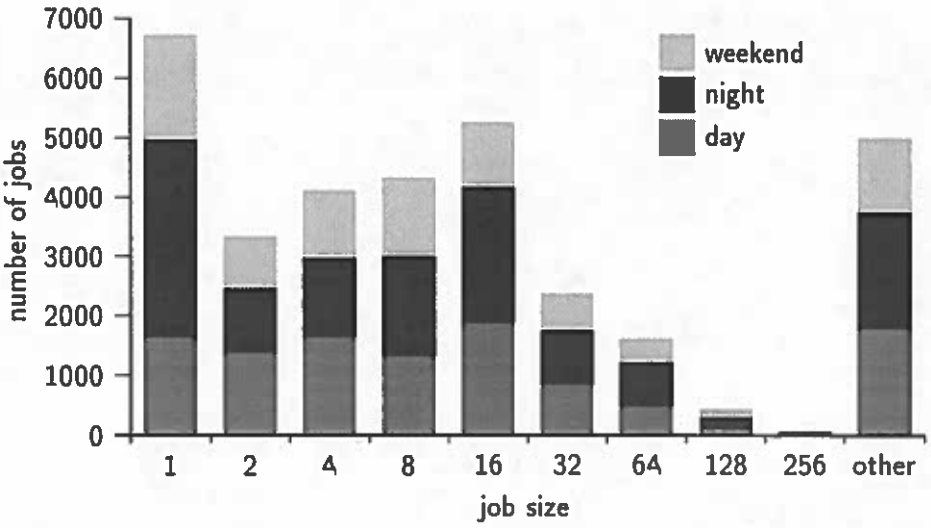

Figure 3: SDSC job size distribution, classified by time period.

as a result of frequent system jobs (those executing UNIX `pwd` as mentioned above). In both traces, the number of jobs falls rapidly after the level of peak parallelism, 32 nodes for NAS and 16 nodes for SDSC. These peaks are especially interesting since, despite the fact that the NAS machine is about one-third of the of the SDSC machine, the distribution's peak parallelism occurs at a larger job size than for the SDSC distribution [1].

The irregularity of Figures 2 and 3 illustrate that modeling workload job distributions with proba-

[1] However, preliminary statistics compiled for more recent SDSC traces indicate that larger job sizes have become more common. This suggests a changing, maturing workload similar to that reported for the Cornell SP-2 trace[4].

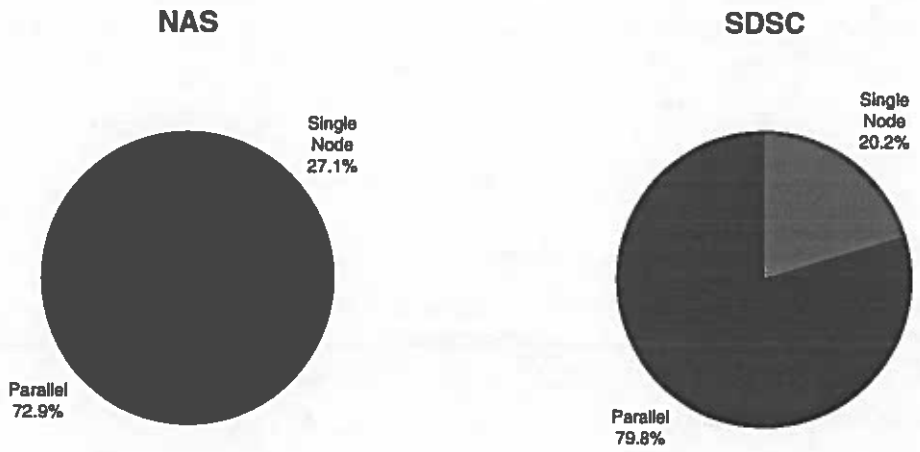

Figure 4: Pie charts illustrating the proportions of single-node and parallel jobs (excluding the NAS pwd commands) in the NAS and SDSC workloads.

	Mean Job Size	Standard Deviation	Coefficient of Variance	Percentage of Jobs
All	13.763843	24.258052	1.762448	100%
Day	14.329133	26.645758	1.859551	33.4%
Night	13.586864	23.227347	1.709544	40.8%
Weekend	13.312034	22.536799	1.692964	25.8%
Direct	8.489726	14.69598	1.731032	66.4%
NQS	24.178211	34.067217	1.409005	33.6%

Table 3: SDSC job size distribution parameters for each job classification.

bilistic functions is a difficult problem for which simple distribution functions will not suffice. However, using a harmonic distribution for job sizes and then manually emphasizing certain "interesting sizes", Feitelson [2] was able to create a surprisingly accurate workload model based on traces from six sites, including this SDSC trace.

In the SDSC trace, only 6,719 jobs (20.2%) ran on a single node, while 26,564 jobs (79.8%) were parallel. This is also comparable to the NAS trace if the frequent diagnostic `pwd` jobs are omitted: 27.1% ran on a single node and 72.9% were parallel. This comparison is illustrated in Figure 4.

Table 3 shows the mean, standard deviation, and coefficient of variance for the SDSC job size distributions.

	All	Direct	NQS	Day	Night	Weekend
Total	33283	22094	11189	11111	13596	8576
Power-of-two	28276 (85.0%)	17547 (79.4%)	10729 (95.9%)	9326 (83.9%)	11626 (85.5%)	7324 (85.4%)
Non-power-of-two	5007 (15.0%)	4547 (20.6%)	460 (4.1%)	1785 (16.1%)	1970 (14.5%)	1252 (14.6%)

Table 4: The number of jobs in SDSC with power-of-two sizes, for each job classification.

Power-of-two Sized Jobs

Despite the fact that the SDSC Paragon permitted jobs of any size, Figure 3 shows that most jobs have sizes that are exactly a power of two. Table 4, giving the breakdown of power-of-two jobs in the SDSC trace, shows clearly that throughout each different category of jobs, the great majority of them have sizes that are powers of two (85.0% of the total). It is especially interesting that this trend is even greater for NQS jobs, of which 95.9% are power-of-two sizes. Furthermore, of all the non-power-of-two jobs, 90.8% of them were directly submitted (interactive). Thus, there appears to be a strong correlation between the type of job (direct vs. NQS) and whether the job size is a power-of-two.

The power-of-two job size distribution at SDSC may be driven by the default job size values associated with the NQS queues. Jobs that do not use the default size must have the actual value specified on the queue submission input line. It is worth speculating that a different distribution would have been obtained if the default queue job sizes had been specified as multiples of 10.

Time Period

In the SDSC trace, 11,111 jobs (33.4%) start during the day, 13,596 jobs (40.8%) start during the night, and 8,576 jobs (25.8%) start during the weekend. This differs from the NAS trace, in which 80.4% of all user jobs began during the day, however, due to scheduling policies at at the NAS site, the *day* period lasted four hours longer than at SDSC.

4 Total Resource Usage

While the job size distribution profiles the number of jobs and their sizes, resource usage statistics tell us which jobs actually used the machine's computational resources by factoring in execution time. Figures 5 and 6 show the total resource usage (the product of job size and execution time) by jobs for varying degrees of parallelism.

Distribution

While Figure 3 showed that most of the SDSC jobs were 16 nodes or smaller, the total resource usage graphs show that practically all of that system's resources (measured in node-seconds) were used by jobs of size 8 or larger. In fact, single-node jobs used only 0.12% of the total resources, which is similar to the NAS trace (0.28%). Furthermore, 73% of the SDSC total resources were used by jobs having the sizes 32, 64 or 128 nodes. Almost all resources were used by jobs with power-of-two sizes, however, the common non-power-of-two job sizes were 11, 14, 22, 24, and 96 nodes.

Time Period - Day/Night/Weekend

In the SDSC trace, non-prime-time (night and weekend) usage accounts for the most total resources used by almost all significant job sizes 8 or larger. This is very similar to the NAS trace, in which daytime resource usage decreased after 32 nodes due to the scheduling policy limiting NQS-queued jobs

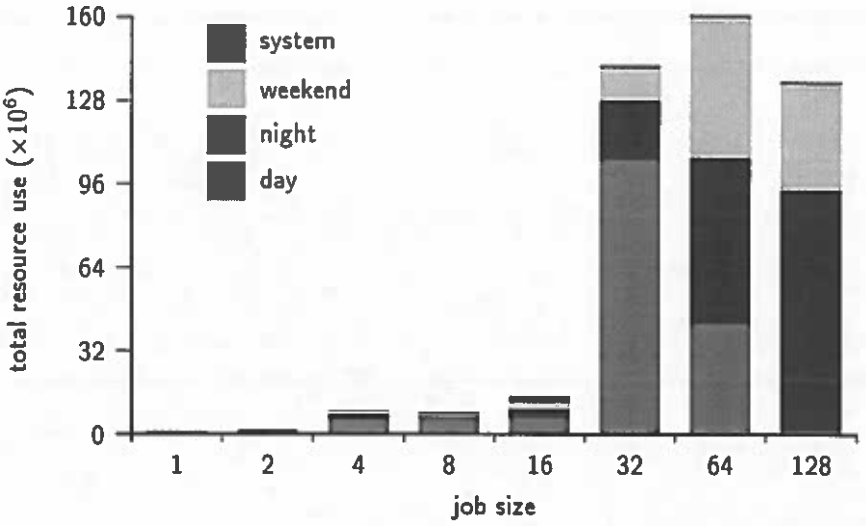

Figure 5: NAS total resource usage by each jobs size, classified by time period (node-seconds).

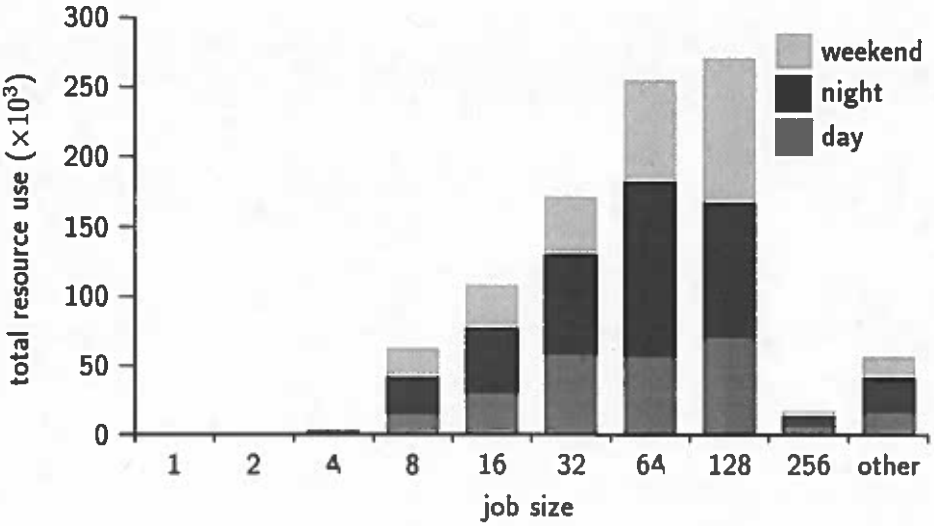

Figure 6: SDSC total resource usage by each jobs size, classified by time period (node-seconds).

to 32 or fewer nodes (refer back to Table 1). However, resource use by large jobs in the SDSC trace is not as heavily dominated by night and weekend usage, since its scheduling policies also allowed large jobs during the day.

Table 5 contains statistical information on the SDSC workload's resource usage distribution. One of the most striking features of this table is the extremely high variance in resources used by directly-submitted jobs. This indicates the direct scheduling system was used for a very wide range of job types, both in terms of their parallelism and runtimes. As with the job size distributions, the resource use distribution cannot be modeled accurately with only a simple probabilistic model.

	Total Resources ($10^6 node \cdot sec$)	Mean Resources ($10^6 node \cdot sec$)	Std Deviation ($10^6 node \cdot sec$)	Coefficient of Variance	Percentage of Jobs
All	3423.3	0.10286	0.48165	4.6828	100%
Day	928.84	0.083696	0.43187	5.1661	33.4%
Night	1455.5	0.10705	0.46829	4.3745	40.8%
Weekend	1039.0	0.12116	0.55693	4.5968	25.8%
Direct	115.91	0.005246	0.11535	22.987	66.4%
NQS	3307.4	0.29560	0.77963	2.6375	33.6%

Table 5: SDSC total resource use distribution parameters for each job classification.

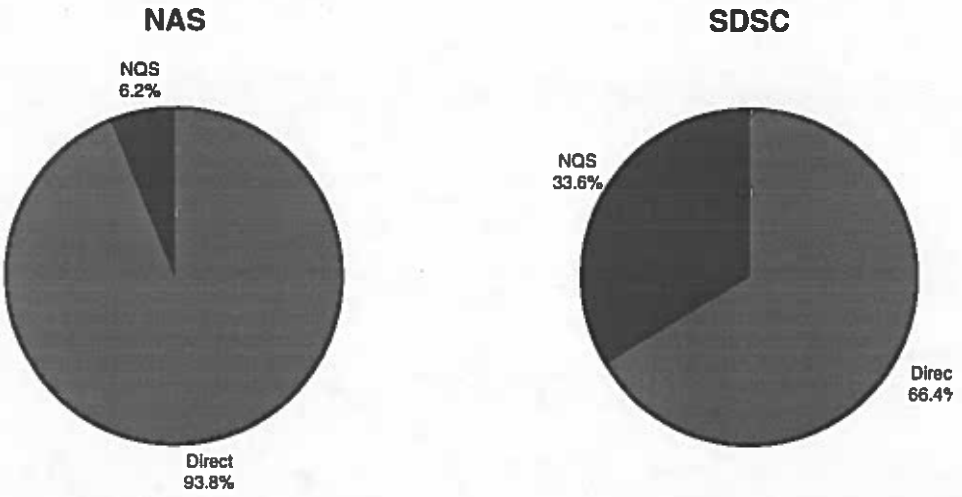

Figure 7: Pie charts illustrating the proportions of direct and NQS user jobs in the NAS and SDSC workloads.

Of the SDSC's total resource usage, 27.1% was attributed to jobs started during the day, 42.5% to jobs started at night, and 30.4% to jobs started on the weekend. To put this in relation to the total wall clock time that the system ran, accounting for down time, 22.1% of the traced time was during the day (9am - 5pm weekdays), 48.4% was during the night, and 29.5% was during the weekend.

5 Job Type - Direct vs. NQS

Job Size Distribution

Figures 8 and 9 show the number of jobs of each size broken down by their type, either directly submitted, or submitted through NQS. Overall, 66.4% of all jobs submitted to the SDSC machine were submitted directly to the interactive partition, while 33.6% were submitted through NQS. This differs from the NAS trace, in which only 6.2% of jobs were submitted through NQS. The proportions of direct to NQS jobs are illustrated in Figure 7. The greater proportion of NQS jobs in the SDSC trace may result from the fact that SDSC is a more general-purpose batch environment with a broad

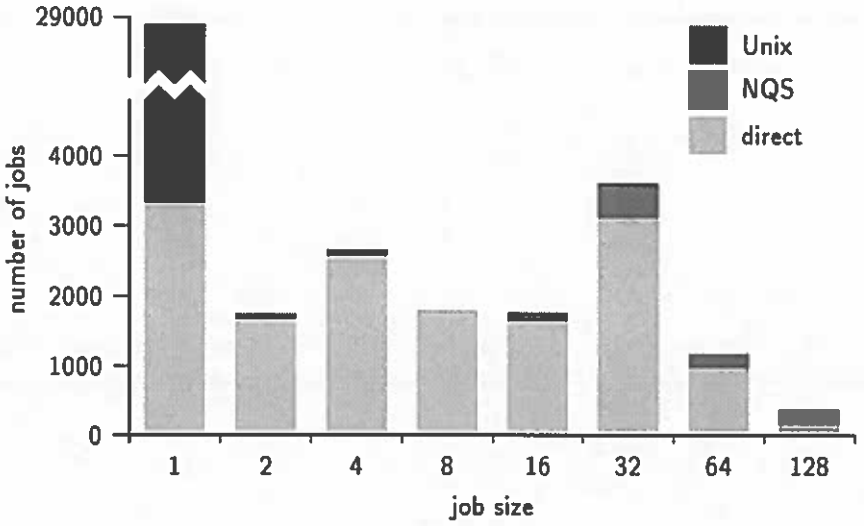

Figure 8: NAS job size distribution, classified by job type.

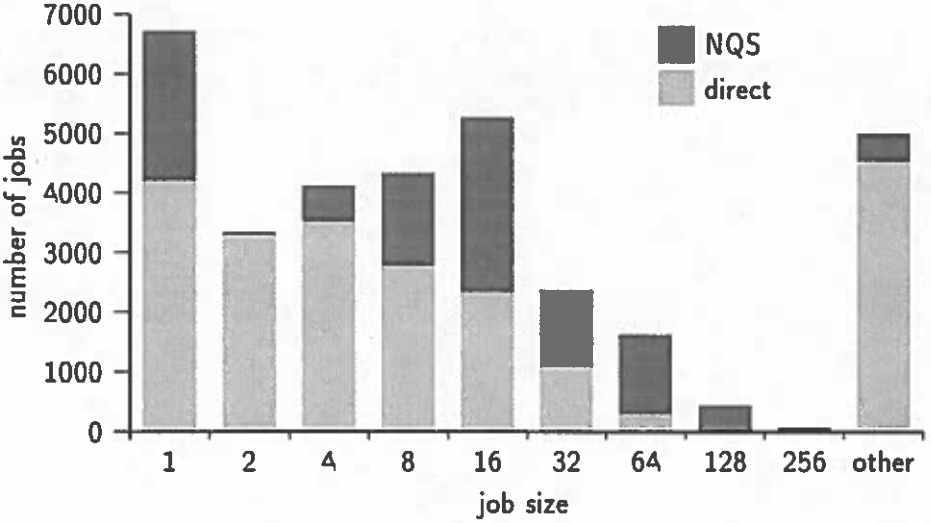

Figure 9: SDSC job size distribution, classified by job type.

user community, while NAS is primarily for specialists desiring more direct control. Another significant difference is the distribution among job sizes. In the SDSC trace, there are significant numbers of NQS jobs of nearly all sizes (predominantly powers of two), but the NAS trace had very few NQS jobs smaller than 16 nodes. However, in both traces, there are very few direct jobs greater than 64 nodes.

Total Resource Usage

Figures 10 and 11 show the total resource usage by jobs of different sizes, broken down by their type, either direct or NQS. It's very interesting to note that although NQS jobs accounted for only 33.6% of the total number in the SDSC trace, they account for almost *all* (96.7%) of the resource usage. In contrast to the NAS trace, this is only true for the largest job size (total NQS resource usage in NAS

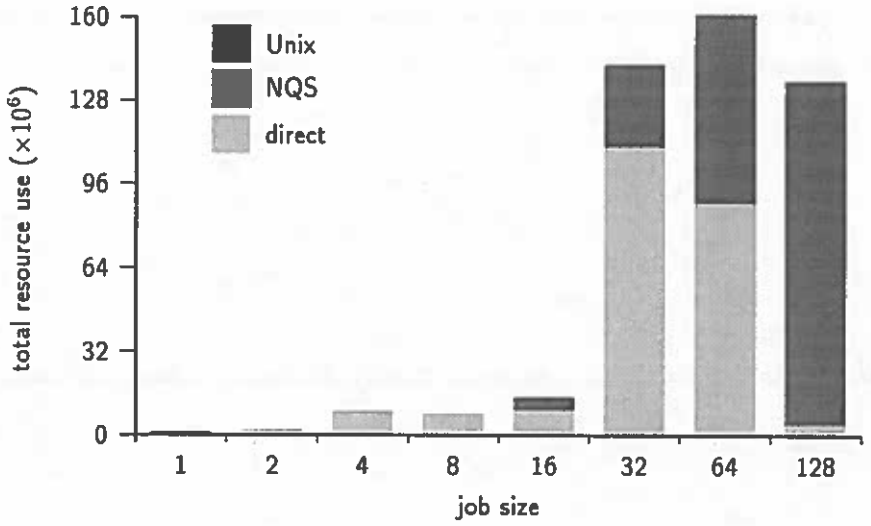

Figure 10: NAS total resource usage of each job size, classified by job type (node-seconds).

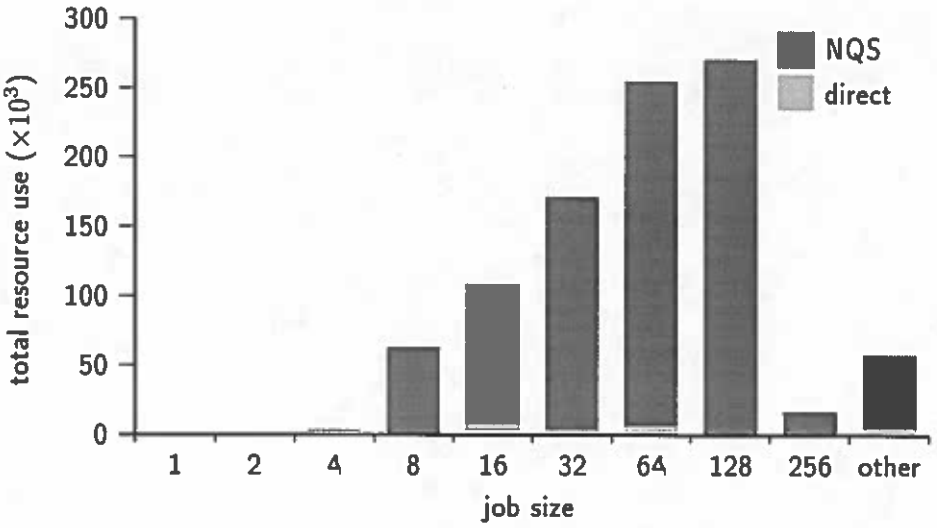

Figure 11: SDSC total resource usage of each job size, classified by job type.

was 50.5%).

6 System Utilization and Multiprogramming Level

System Utilization

Figures 12 and 13 show the average system utilization for both machines, throughout the day. In general, the SDSC utilization follows the same patterns as the NAS workload, except that SDSC's utilization is much more steady and moderated. It does not show the extreme surges and lulls of

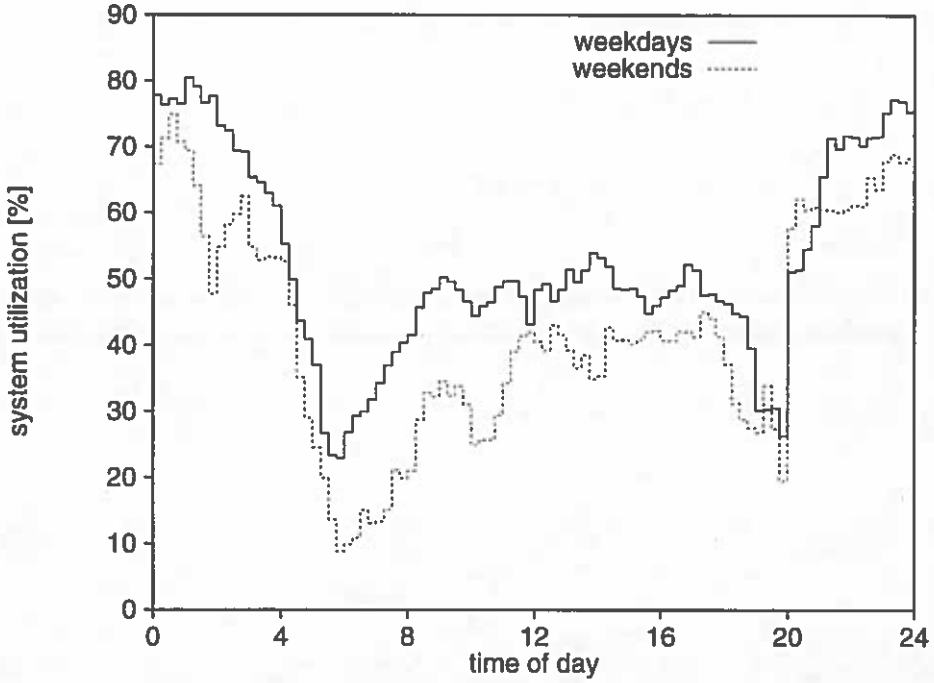

Figure 12: NAS average system utilization as a function of time of day.

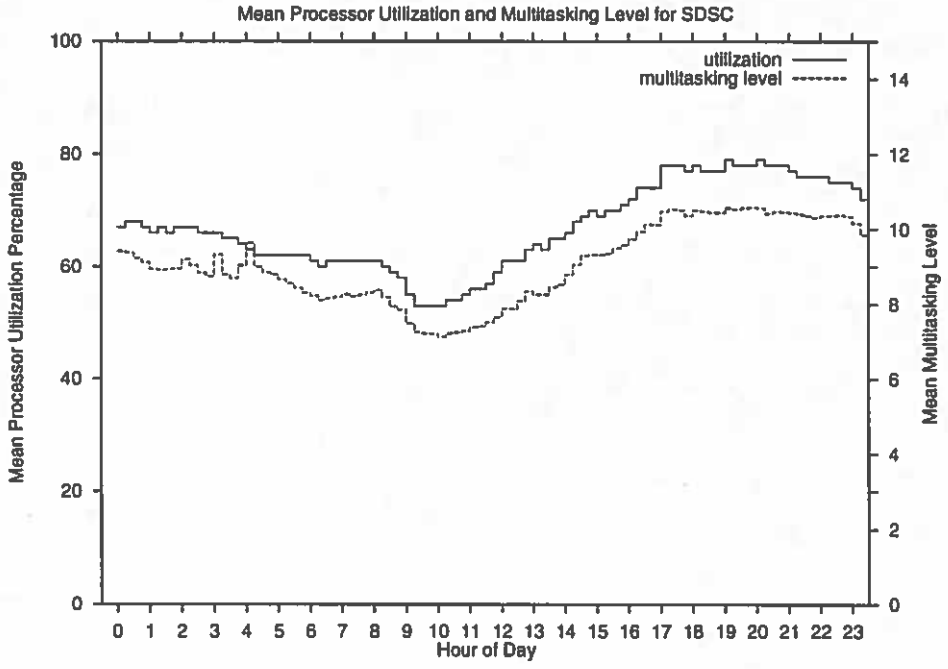

Figure 13: SDSC average system utilization and mean multitasking level as a function of time of day.

utilization that characterizes the NAS workload. Similar to the NAS utilization, the SDSC shows very distinct differences between day (9am - 5pm) and night. At the beginning of the day (9am), utilization is at its lowest point, which is still more than 50% (NAS is at about 25% at the beginning of its prime-time at 6am). SDSC's Utilization increases steadily throughout the workday and when prime-time ends at 5pm, utilization reaches its peak of about 80%, falling steadily throughout the night. The SDSC utilization does not exhibit the decline at the end of its prime-time period shown in the NAS trace. The utilization drop in the NAS trace is probably an artifact of the scheduling policies, which do not allow for direct jobs to execute except during prime-time. Thus, at the end of the day, users must cease submitting direct jobs while large NQS are held up until non-prime-time starts, decreasing utilization. The SDSC scheduling policies do not repartition the direct and NQS nodes, so direct jobs can continue to be served.

Multiprogramming level

Figure 13 also shows the multiprogramming level of the SDSC workload throughout the day compared to its system utilization. The average multiprogramming level tends to mirror the average utilization almost exactly, starting at about seven concurrent jobs at the start of prime-time and increasing during the day to its peak of over ten concurrent jobs in the early evening.

Figures 14 and 15 show the total time spent at each multiprogramming level. The analysis of the SDSC's average multiprogramming level reveals a very important difference in the way the machine is used compared to the NAS machine. The NAS machine, restricted to 9 concurrently executing jobs, spent 32.0% of its total traced time running only a single job and this was the dominant multiprogramming level. However, the SDSC machine spent the majority of its traced time at multiprogramming levels of between 5 and 15 concurrent jobs. In fact, it spent only 1.3% of its traced time running only a single job and multiprogramming levels of up to 27 were observed. The machine was idle 4.1% of the total traced time, and down for another 4.0%. The dramatically higher multiprogramming levels observed in the SDSC trace help explain the more continuous utilization shown in Figure 13. The presence of a large number of jobs in the system results in utilizations and other system metrics that are more independent of the behavior of individual jobs or individual users.

Figure 16 shows the average processor utilization observed at each multiprogramming level for the SDSC trace. As expected, the utilization tends to increase as more jobs are run concurrently in the system. Generally, more jobs were able to run concurrently during the night and weekends, up to 27, rather than the maximum of 23 during the day.

Down Time

The SDSC machine was down for a total of 146.5 hours (4.0% of the total traced time). Of this down time, 58.7% occurred during the day (prime-time), 19.4% during the night, and 21.8% during the weekend. There were 33 hardware failures, 50 software failures, and 16 down times due to other or unknown causes. Further, of the 99 down time periods, only 37 of them were pre-scheduled.

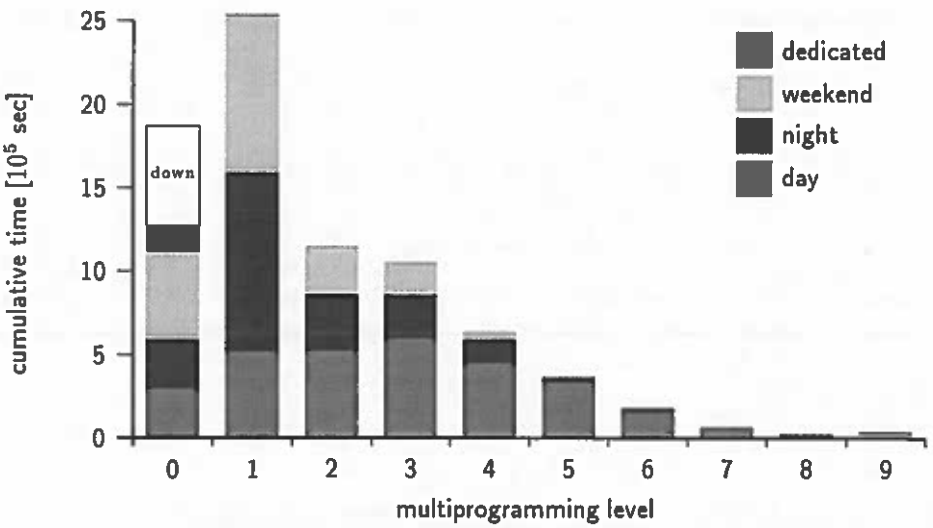

Figure 14: NAS cumulative time spent in each multiprogramming level.

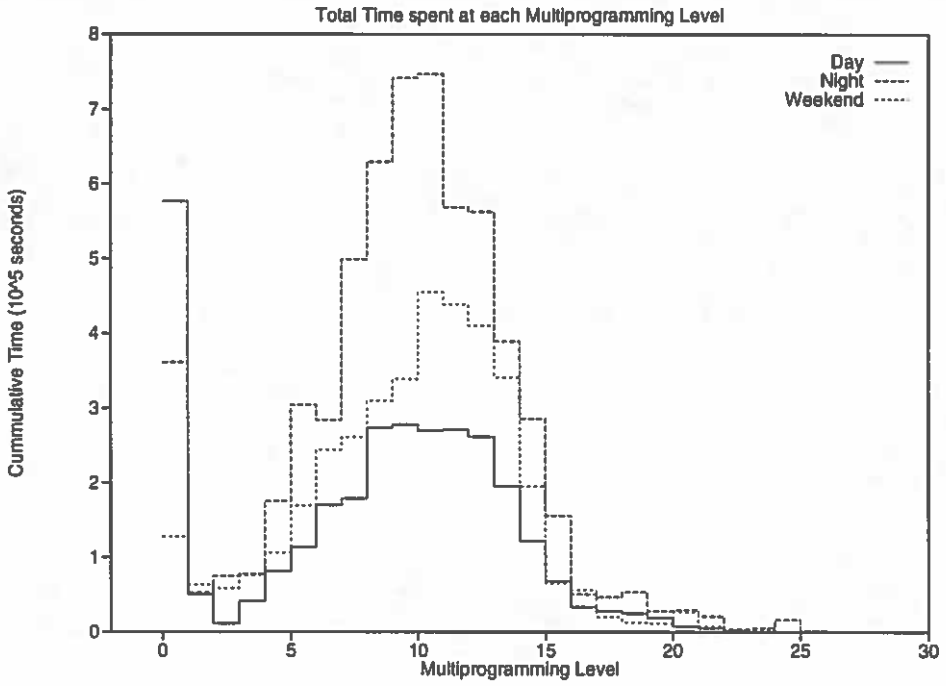

Figure 15: SDSC cumulative time spent in each multiprogramming level.

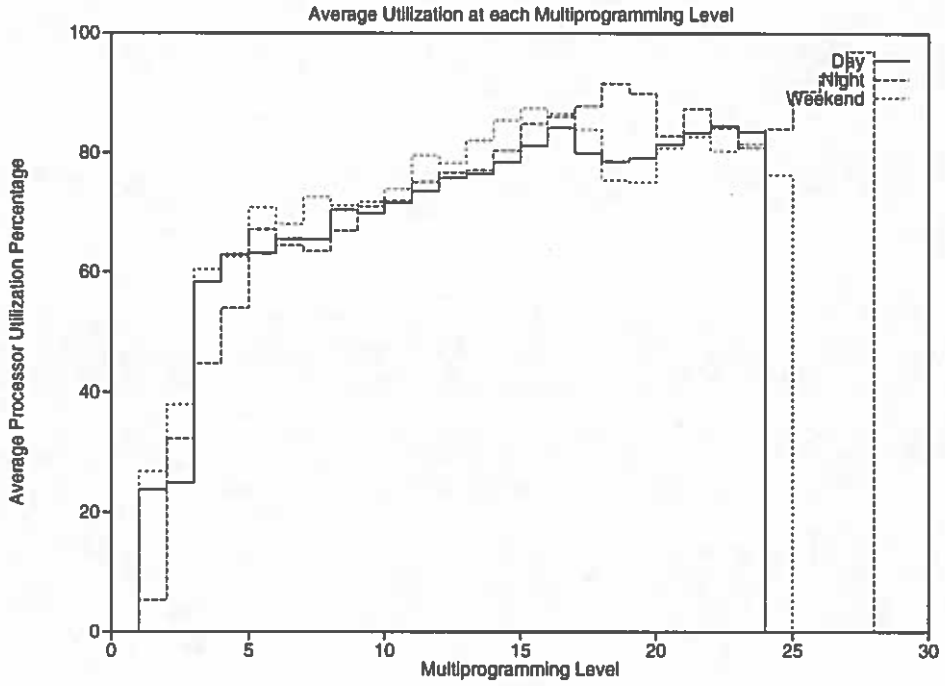

Figure 16: SDSC average utilization at each multiprogramming level.

7 Runtime Distribution

Figures 17 and 18 show the average resource usage for jobs of each size. We can see that the resource requirements of nighttime jobs are usually more than those for daytime jobs. However, the differences are not nearly as dramatic for the SDSC trace as they are for the NAS trace, in which nighttime jobs often require up to an order of magnitude more resources than daytime jobs. Night jobs in the SDSC trace generally use no more than twice the resources of day jobs.

Table 6, giving the mean runtimes for the SDSC trace, shows that (for up to 128 nodes) runtime tends to increase and the variance in runtimes decreases for progressively larger jobs. However, for the few jobs larger than 128 nodes, the opposite trend was observed: runtime decreases while the variance increases. The NAS trace shows similar trends in runtimes and variances for jobs of each size, with mean runtimes generally increasing from about 712.2 seconds (with variance of 3.4) for two-node jobs to 3280.1 seconds for 128-node jobs. However, we can see that large jobs in the SDSC machine tended to run for longer amounts of time, as we would expect given that the SDSC NQS queues allow for longer running jobs.

Figures 19 and 20 show the distribution of runtimes for all jobs, sequential jobs, and parallel jobs. Similar to the NAS workload, sequential jobs in the SDSC trace have a wide distribution, mostly spread very evenly between 90 seconds and 6 minutes, and with a large number running 3 seconds or less. Parallel jobs are distributed heavily between 3 seconds and 1 minute, but their numbers decrease as runtime increases, except for local peaks just before 1 and 12 hours. As with the NAS workload, these peaks are exclusively the result of the NQS queue time limits (1, 4, and 12 hours). A large number of parallel jobs also ran for 3 seconds or less.

Figures 21 and 22 shows the cumulative distribution of runtimes for sequential and parallel jobs. The SDSC workload is again characterized by a more uniform distribution of runtimes.

Figures 23 and 24 present same information as above (Figures 19 and 20), except that jobs are differentiated as either direct or NQS jobs. It becomes very apparent that the peaks in the SDSC trace at 1 and 12 hours are the result of NQS jobs exclusively. Also, we can see that the large majority of short-running jobs (3 seconds or less) are those that were submitted directly, rather than with NQS. But still, runtimes of NQS jobs are quite evenly distributed, while direct jobs become fewer as runtimes increase.

Figures 25 and 26 show the runtime distribution broken down by exact degrees of parallelism of the jobs. Here, we can see that the peak in the SDSC trace at one hour is almost totally the result of jobs with between 9 and 16 nodes. Also, the 12 hour peak is mostly the result of jobs using between 5 and 32 nodes.

When analyzing the distribution of runtimes in the NAS trace in [3] the authors conclude that the runtimes for parallel jobs tend to increase as the parallelism of the jobs increases. Our study of the SDSC trace appears to support this claim, as seen in Table 6 and Figure 26. However, it must be noted that runtime is also influenced by many other factors, such as time of day or submission method. An accurate probabilistic model for the runtime distribution may need to classify jobs by types (direct vs. NQS, parallel vs. sequential, etc), associate these types with probabilities, and then generate runtimes based on stochastic functions specifically tailored to the job type.

Figure 17: NAS average resource use by job of each size (node-seconds).

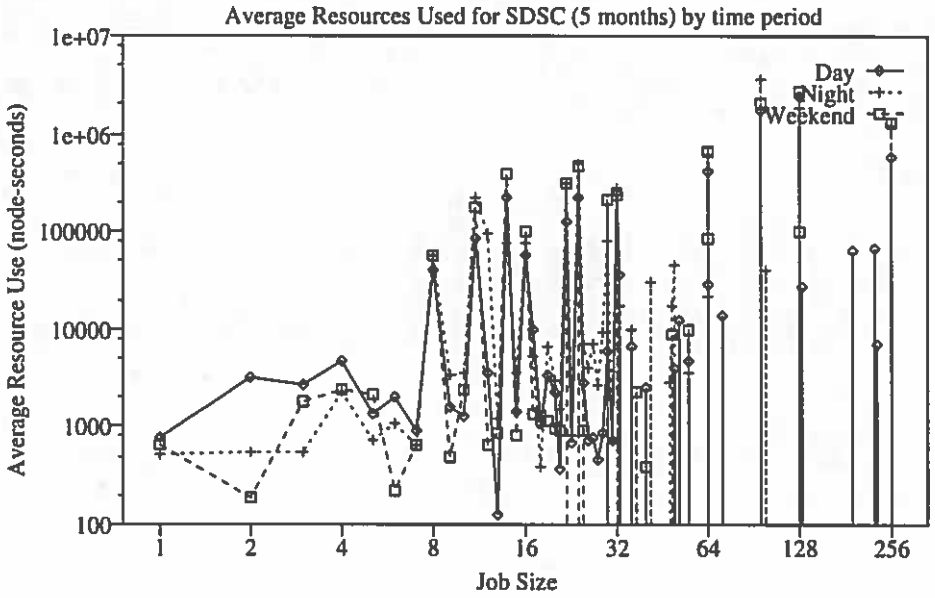

Figure 18: SDSC average resource use by job of each size (node-seconds).

	Mean Runtime (sec)	Standard Deviation	Coefficient of Variance	Percentage of Jobs
All	3479.4	10,805.6	3.1056	100%
1 node	618.49	4396.5	7.1085	20.2%
2 nodes	774.21	12,082.0	15.606	10.0%
3-4 nodes	743.98	7988.1	10.737	16.3%
5-8 nodes	5046.7	10,390.3	2.0588	17.1%
9-16 nodes	4526.5	11,250.0	2.4854	20.1%
17-32 nodes	7372.4	14,261.9	1.9345	9.5%
33-64 nodes	8611.4	13,985.7	1.6241	5.0%
65-128 nodes	16,694	18,661.6	1.1178	1.4%
129-256 nodes	3009.5	7491.3	2.4892	0.2%
257-400 nodes	508.13	914.18	1.7991	0.003%
Day	3072.8	12,144.0	3.9521	33.4%
Night	3560.2	9630.8	2.7051	40.8%
Weekend	3878.1	10,707.5	2.7610	25.8%
Direct	505.47	7344.0	14.529	66.4%
NQS	9351.9	13,743.1	1.4696	33.6%

Table 6: SDSC runtime distribution parameters for each job classification and degree of parallelism.

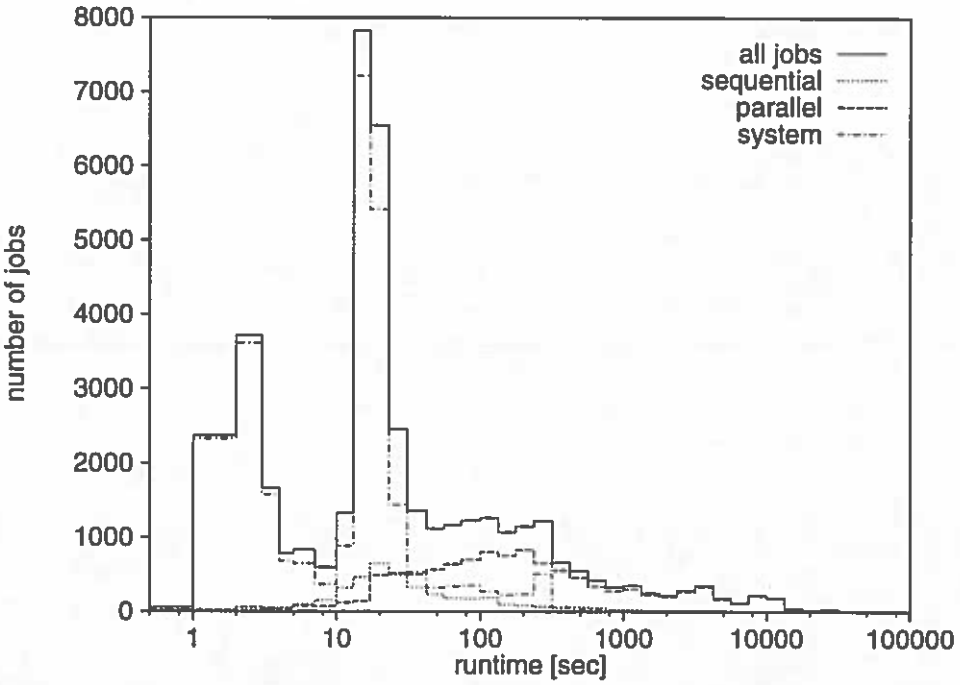

Figure 19: NAS pointwise distribution of runtimes, classified by job parallelism.

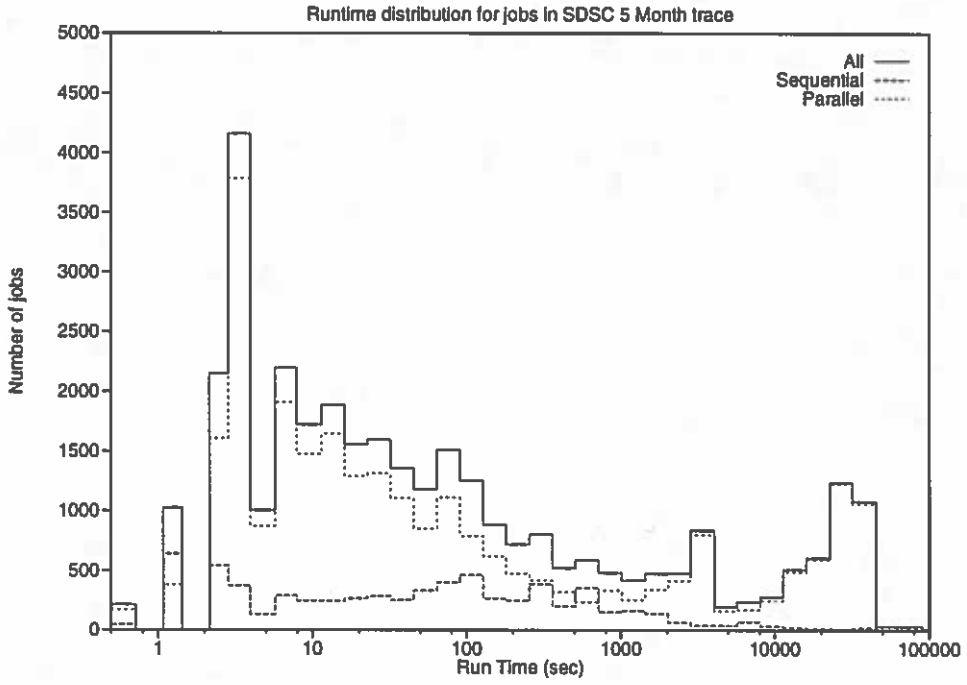

Figure 20: SDSC pointwise distribution of runtimes, classified by job parallelism.

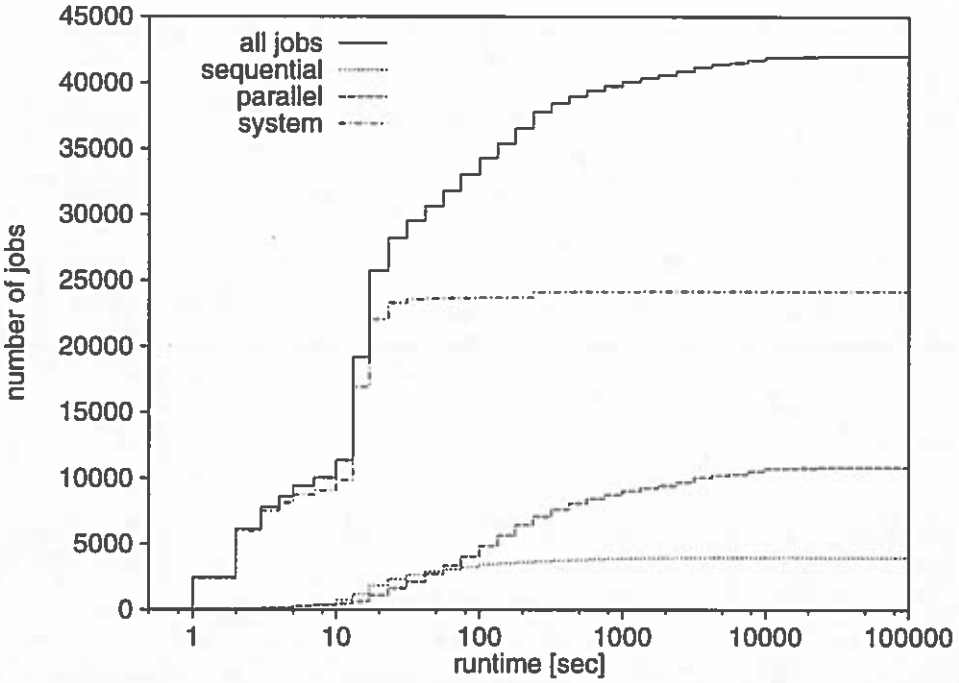

Figure 21: NAS cumulative distribution of runtimes, classified by job parallelism.

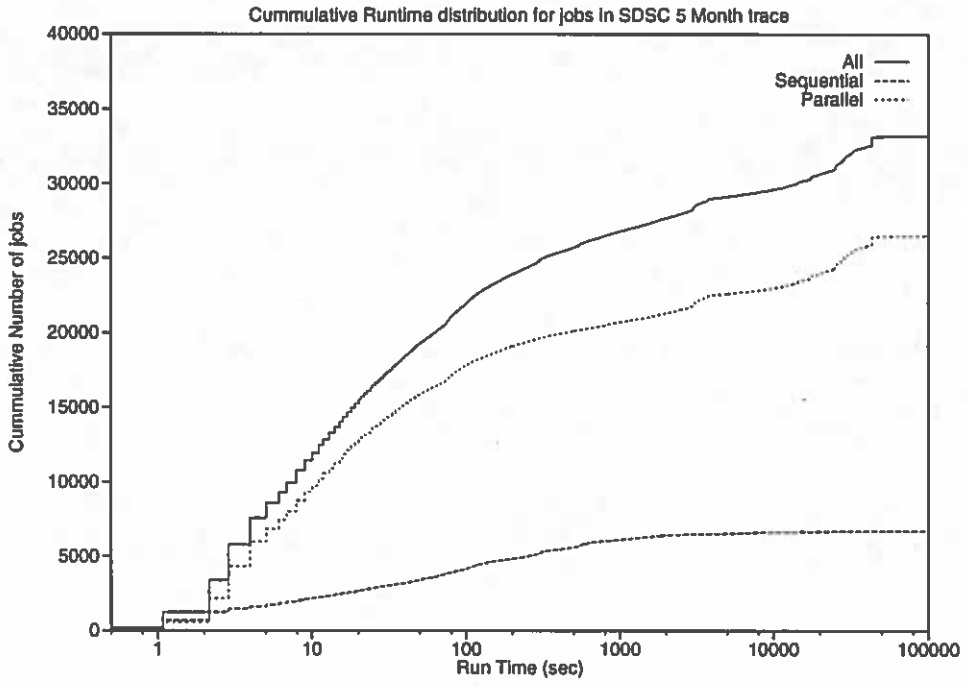

Figure 22: SDSC cumulative distribution of runtimes, classified by job parallelism.

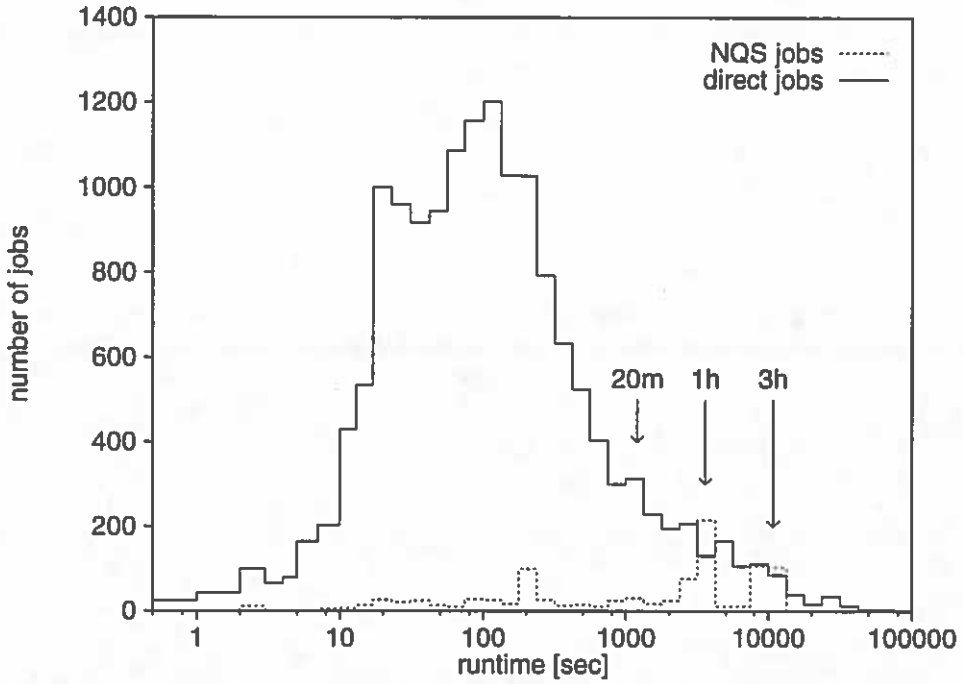

Figure 23: NAS pointwise distribution of runtimes, classified by job type.

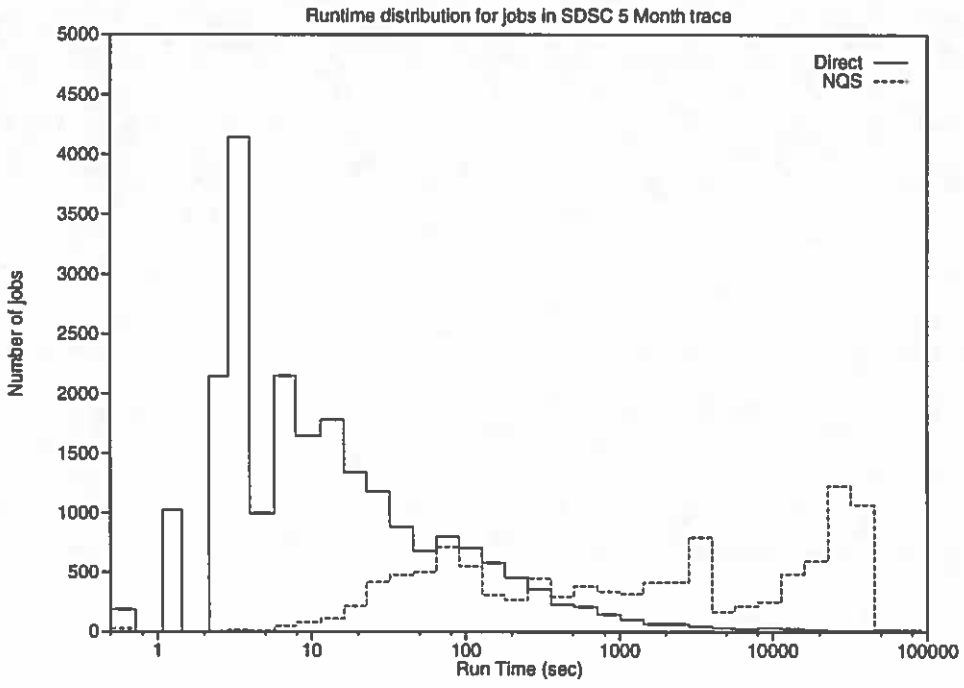

Figure 24: SDSC pointwise distribution of runtimes, classified by job type.

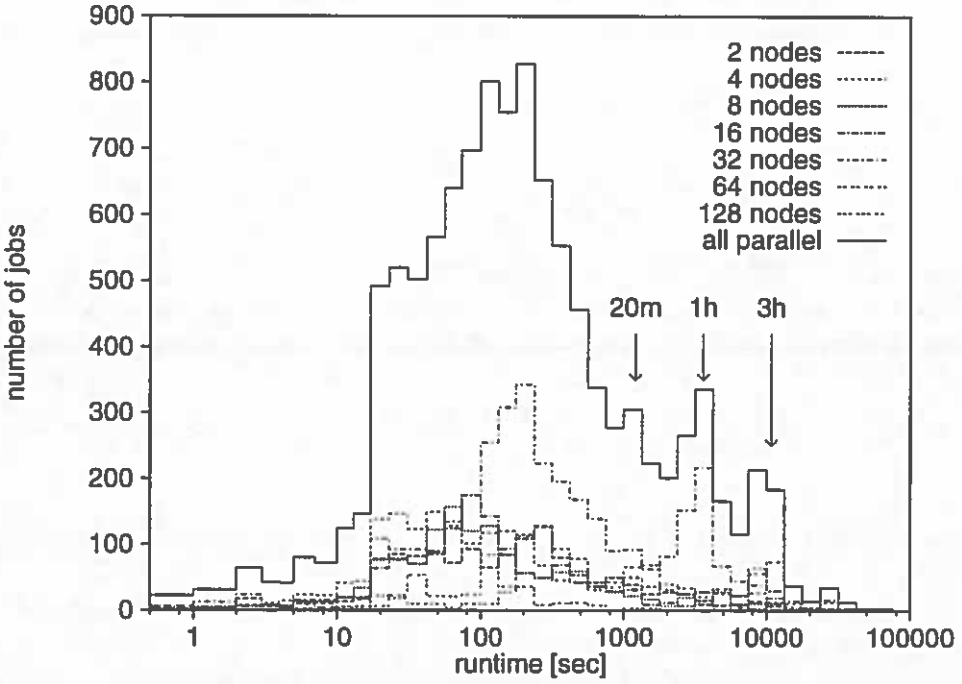

Figure 25: NAS pointwise distribution of runtimes classified by degrees of job parallelism.

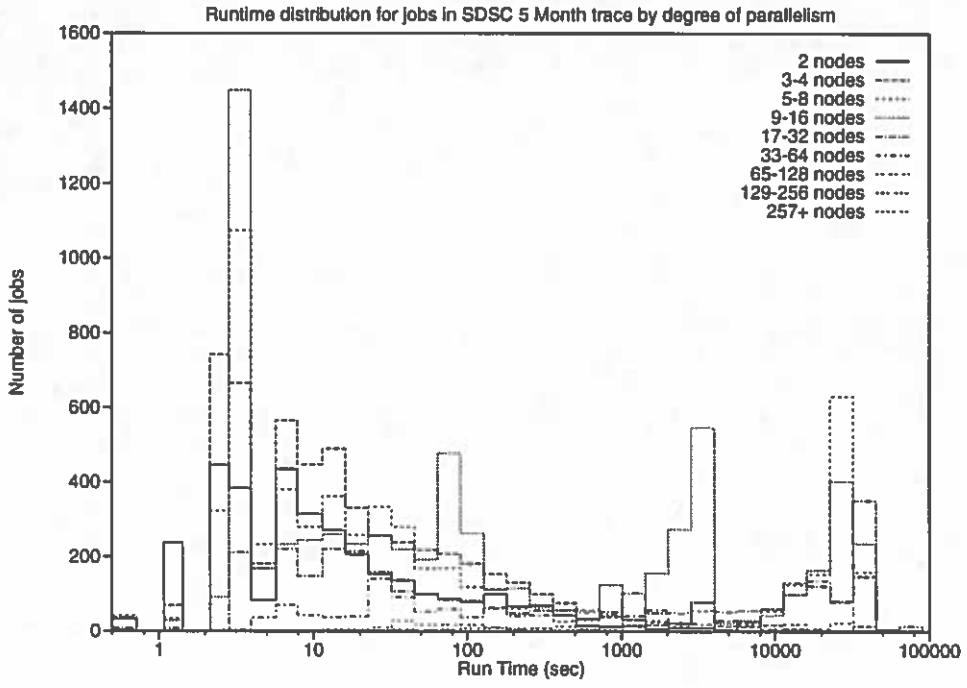

Figure 26: SDSC pointwise distribution of runtimes classified by degrees of job parallelism.

8 Job Submission Statistics and Interarrival Times

Investigating job submission rates and characteristics yields insight into the temporal usage patterns of a machine's workload. For interactively scheduled jobs, the user submission (arrival) and start times are essentially the same, since interactive jobs are scheduled immediately. But for batch (NQS) jobs, arrival and start times are distinct due to job wait times accumulated when held in scheduling queues. Unfortunately, in contrast to the SDSC trace, the NAS trace did not include actual job arrival times, only start times. Therefore, for the purposes of comparison, job start times are used to represent arrival times for both traces. An exception occurs in our analysis of SDSC wait times, when actual arrival times are considered. However, even for NQS jobs, start times are interesting, because they characterize when resources become available for scheduling.

Job Submission Times

Figures 27 and 28 show the job submission rate for each hour throughout the day. Despite their differing hours for prime-time service, the NAS and SDSC machines have strong similarities in their daily job submission rates. Both have predictably reduced submission rates at the end of their prime-time periods and higher rates early in the morning. However, while the NAS submission rate drops to almost nothing at night, SDSC retains a moderate submission rate throughout most of the 24-hour period.

Average Job Size vs. Submission Time

Figures 29 and 30 show the average size of jobs submitted during each hour of the day. The SDSC machine, with a large, active NQS partition throughout all periods of the day, had a more even distribution of job sizes submitted over time, except in the very early morning, when the largest NQS jobs tended to run. The NAS machine, on the other hand, had large surges in submitted job sizes throughout night, while jobs submitted during the day tended to be very small.

Wait Times

In addition to the jobs start times provided in both traces, the SDSC trace contained actual job arrival times for NQS jobs (the NAS trace did not have any arrival times). Interactive jobs did not have distinct arrival times since, by definition, they were allocated immediately, without being queued. The presence of arrival times allowed for an analysis of SDSC's job wait times, the amount of time jobs wait in the NQS queues before being allocated.

Figure 31 shows the overall distribution of wait times for SDSC's NQS jobs. Two distinct groupings of wait times are clearly visible. The most common wait times were very small, less than one minute. In fact, of the 10,622 NQS jobs having usable arrival information (a few jobs were missing information or had earlier start times than arrival times), the most frequent wait time was only 5 seconds, attributed to 752 jobs. The median wait time was 29 seconds, while the average was 4.95 hours, indicating that a relatively smaller number of jobs had very large wait times.

The second distinct group of job wait times are broadly distributed between about 30 minutes and 32 hours. Figure 32 shows a more detailed plot of this range of wait times by hours. Most of the large quantity of jobs that wait exactly 24 hours belong to system administrators and selected users; these are very short-running, sequential programs that automatically reschedule themselves to run at the same time the following day.

Interarrival Time

Figures 33 and 34 show the distribution of interarrival times for the two traces. This SDSC distribution appears to be very similar to that for the NAS trace, with the distribution peaking between 90 seconds and 270 seconds. Note, once again, that since the NAS trace did not include actual arrival times for NQS jobs, job start times are used to represent arrival times for both traces. Regardless, for

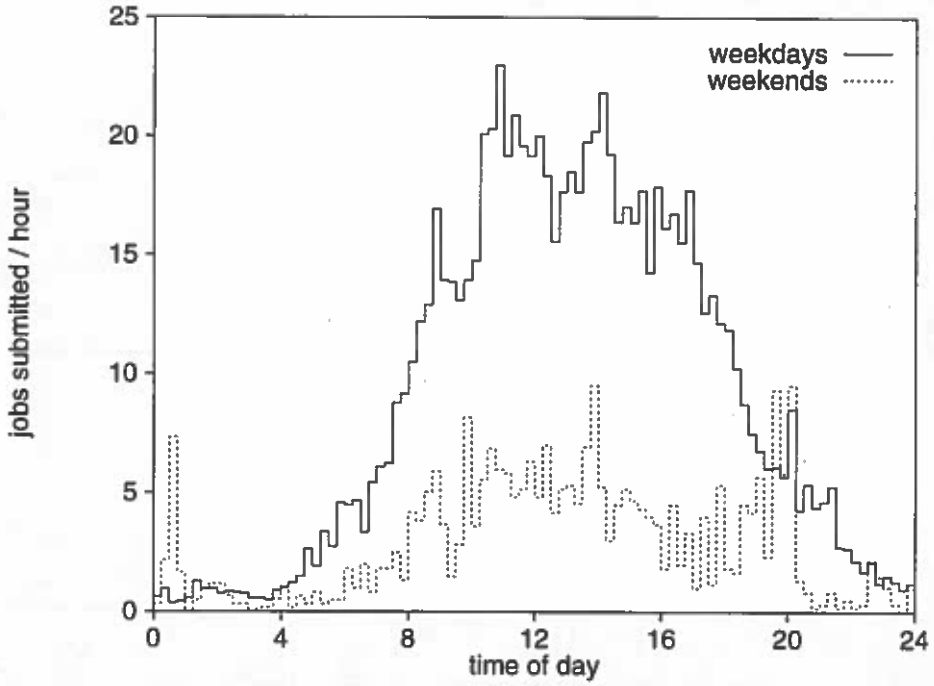

Figure 27: NAS job submission rate as a function of time of day.

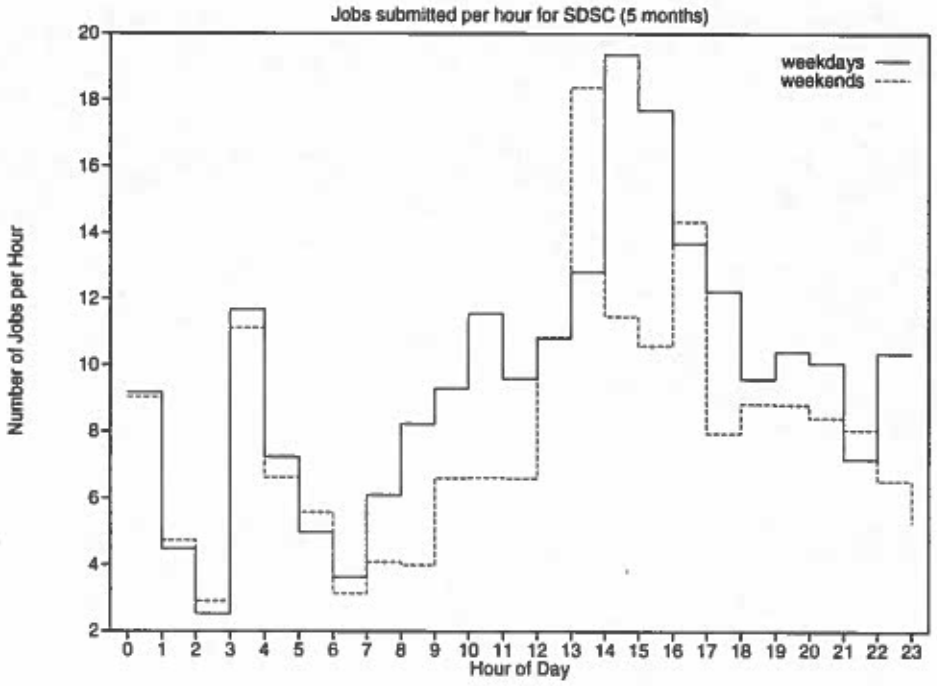

Figure 28: SDSC job submission rate as a function of time of day.

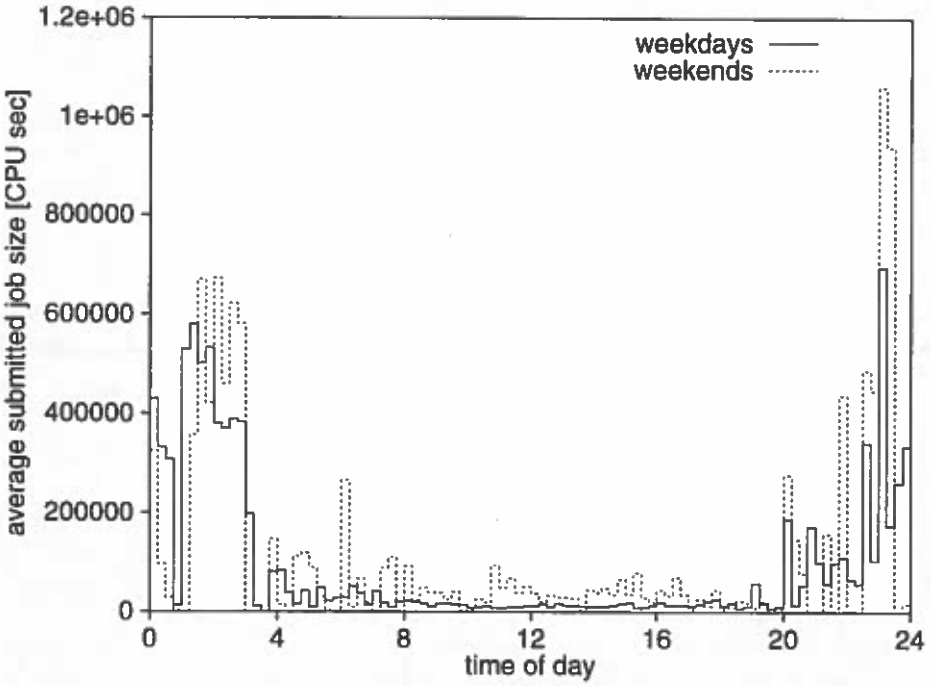

Figure 29: NAS job size as a function of time of day.

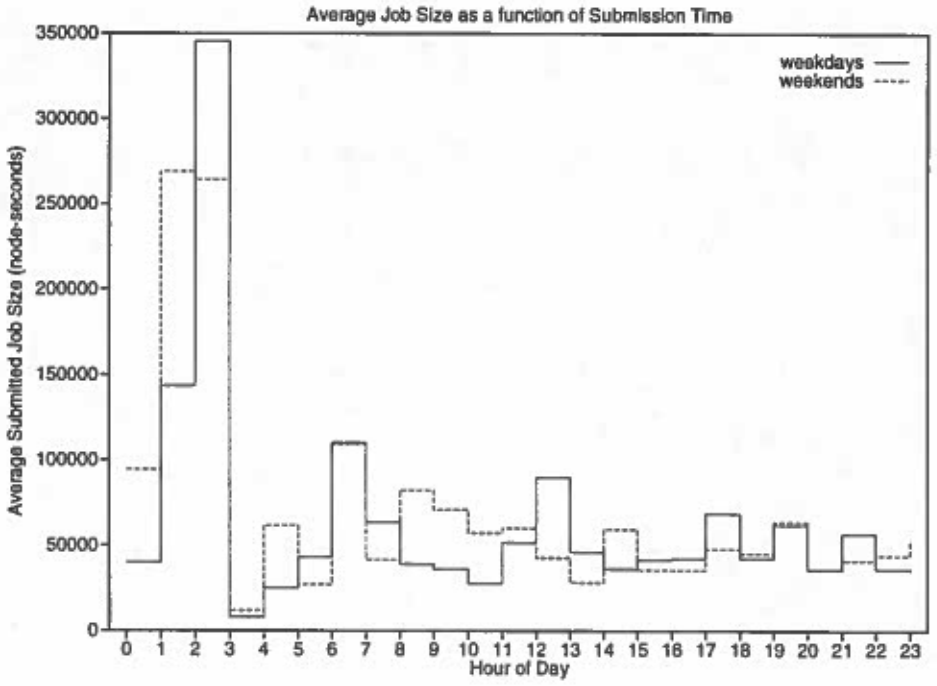

Figure 30: SDSC job size as a function of time of day.

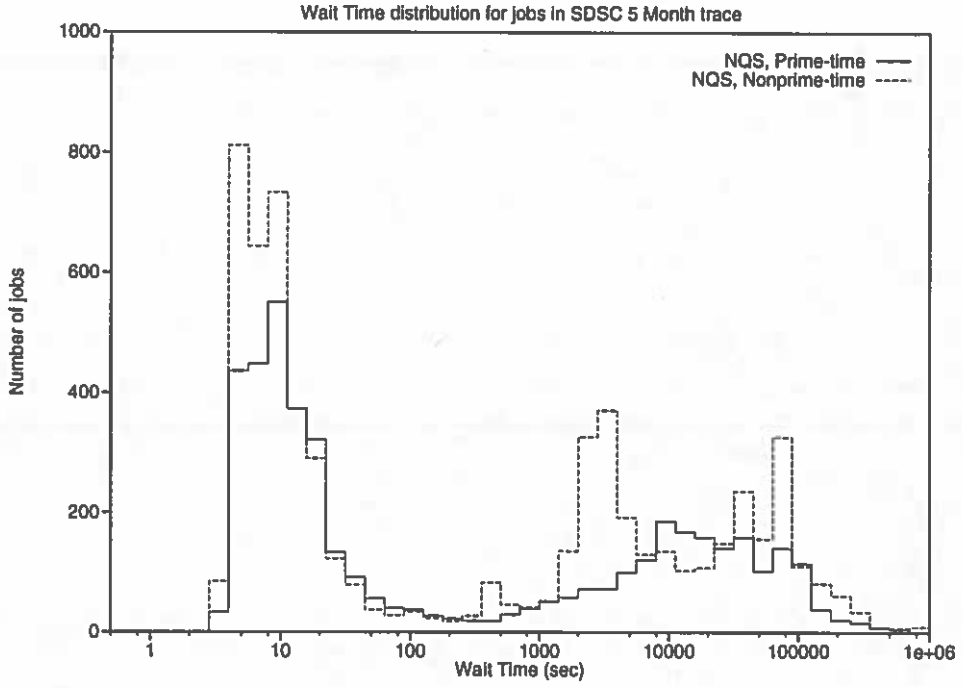

Figure 31: SDSC distribution of job wait times, plotted logarithmically.

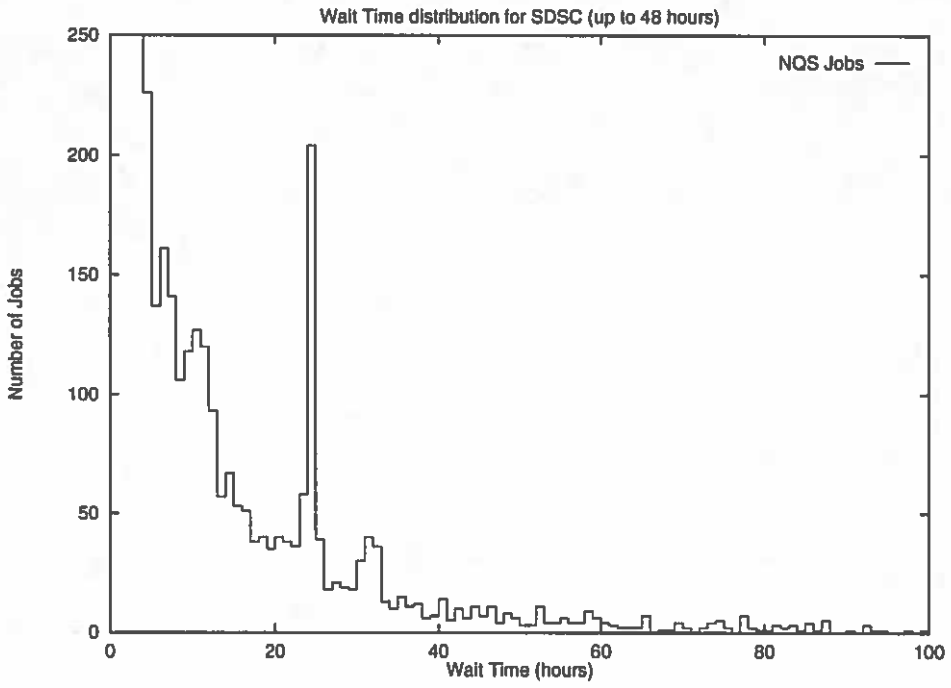

Figure 32: Detail of SDSC distribution of job wait times up to 100 hours, plotted hourly with a linear scale.

	Mean Inter-arrival Time (sec)	Standard Deviation	Coefficient of Variance
All	391.82	2091.6	5.3383
Day	234.15	703.29	3.0035
Night	665.06	3880.0	5.8340
Weekend	433.96	1361.4	3.1371

Table 7: SDSC job interarrival time distribution parameters for jobs arriving in each time period.

the interactive (direct) jobs, start times actually do reflect submission times.

Table 7, giving interarrival times for the SDSC trace, shows again that jobs arrive less frequently and with greater variance in their interarrival times at night. During the day, the SDSC mean interarrival time of 234 seconds compares well with the NAS mean interarrival time of 270 seconds. However, during the night and weekends, SDSC jobs arrive at more than twice the rate of NAS jobs, showing much more sustained usage of the machine. Compared to the NAS workload, with interarrival time variances of 3.56, 2.11, and 2.83 for day, night, and weekend, respectively, SDSC shows greater variance in the interarrival times of incoming jobs.

While the interarrival time distribution has many subtleties, its behavior is much more regular than any of the other distributions characterizing the workloads, and could be modeled reasonably well by simpler probabilistic functions.

9 Conclusions

We conducted much the same analysis on the 5 month SDSC Intel Paragon trace as was done for the NASA Ames NAS iPSC/860 [3]. Despite many fundamental differences in the two machines and the way in which they were used, there are a surprising number of similarities in their workloads.

- The overall job size distributions had many similar characteristics. Specifically, the shape of the distributions were very similar and the numbers of user jobs submitted per month and the proportions of those which were parallel vs. sequential jobs were very close.
- Jobs submitted to both machines had sizes that were primarily powers of two. This was an inherent constraint in the NAS iPSC/860, while arbitrarily sized jobs were permitted in the SDSC Paragon.
- General trends in the way that system utilization changes during the day are similar, but peaks and lulls are more pronounced for the NAS machine.
- The overall runtime distribution shows the same trends of increasing execution times and decreasing variance as job parallelism increases.
- The interarrival time distribution is very similar, especially for daytime jobs.

However, aside from the obvious differences in architecture and machine size, there were several significant differences in the workloads recorded for the two machines. Most of the differences reflect the much more heterogeneous and multiprogrammed workload of the SDSC system, in which overall system performance is more independent of individual jobs or users. To summarize:

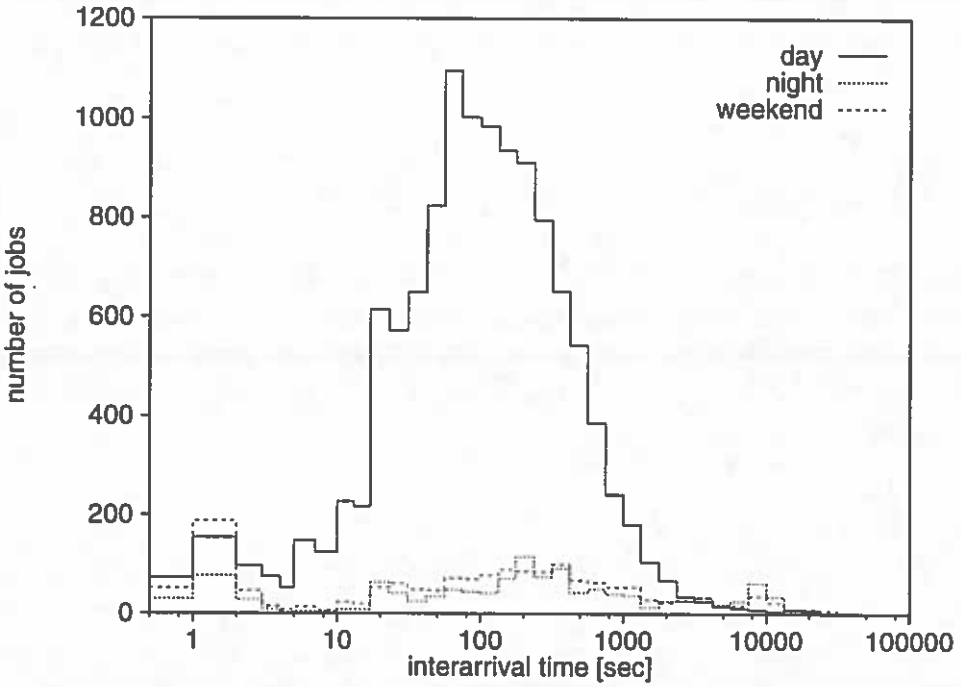

Figure 33: NAS pointwise distribution of job interarrival times.

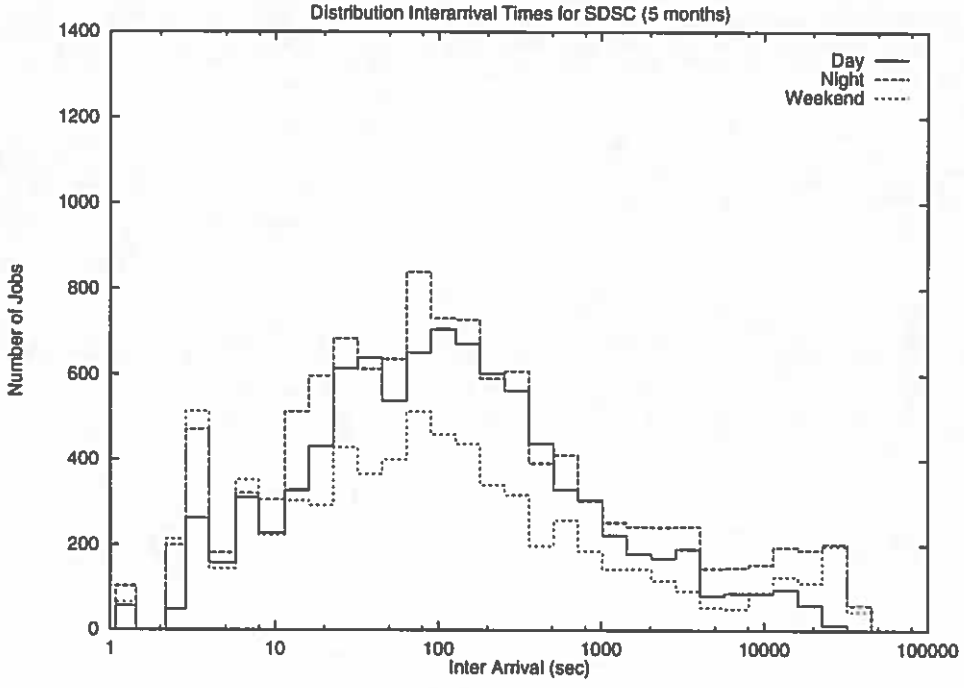

Figure 34: SDSC pointwise distribution of job interarrival times.

- The NAS workload made much less use of the NQS scheduling system.
- The application mix was much more diverse in the SDSC system.
- The average system utilizations and job submission rates were much more moderated and sustained in the SDSC system.
- Multiprogramming levels were much higher in the SDSC system.

One area of relevance of this work for researchers studying resource allocation issues (particularly scheduling and allocation), is to help guide the development of accurate, realistic workload models for resource management experimentation. While parameters for stochastic modeling of job size, runtime, and interarrival time distributions can be extracted from this work, our analysis of the distribution shapes and influencing factors indicates that stochastic job stream models must be much more complex that what is traditionally done.

Our future work includes using this and other workload studies to refine and validate the use of (1) real-world job streams and (2) stochastic job streams modeled using parameters of real workloads, in resource management simulations that are as realistic as possible.

References

[1] Intel Corp. Paragon Network Queuing System manual. October 1993.

[2] D. Feitelson. Packing schemes for gang scheduling. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing, IPPS 96 Workshop*, Lecture Notes in Computer Science. Springer, 1996.

[3] D. G. Feitelson and B. Nitzberg. Job characteristics of a production parallel scientific workload on the NASA Ames iPSC/860. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing, IPPS 95 Workshop*, Lecture Notes in Computer Science, 949, pages 337–360. Springer, 1995.

[4] S. Hotovy. Workload evolution on the Cornell Theory Center IBM SP2. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing, IPPS 96 Workshop*, Lecture Notes in Computer Science. Springer, 1996.

[5] K. Li and K. Cheng. A two-dimensional buddy system for dynamic resource allocation in a partitionable mesh connected system. *Journal of Parallel and Distributed Computing*, 12:79–83, 1991.

[6] R. Moore and M. Wan. Intel Paragon allocation algorithms. Personal communication, 1995.

[7] T. Suzuoka, J. Subhlok, and T. Gross. Evaluating job scheduling techniques for highly parallel computers. Technical Report CMU-CS-95-149, School of Computer Science, Carnegie Mellon University, 1995.

[8] M. Wan, R. Moore, G. Kremenek, and K. Steube. A batch scheduler for the Intel Paragon MPP system with a non-contiguous node allocation algorithm. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing, IPPS 96 Workshop*, Lecture Notes in Computer Science. Springer, 1996.