Comprehensive Workload Analysis and Modeling of a Petascale Supercomputer

Hao Zhang[1] and Haihang You[2]

1. Department of Electrical Engineering and Computer Science,
University of Tennessee, Knoxville, TN 37996, USA
2. National Institute for Computational Sciences,
Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
{haozhang,hyou}@utk.edu

Abstract. The performance of supercomputer schedulers is greatly affected by the characteristics of the workload it serves. A good understanding of workload characteristics is always important to develop and evaluate different scheduling strategies for an HPC system. In this paper, we present a comprehensive analysis of the workload characteristics of Kraken, the world's fastest academic supercomputer and 11th on the latest Top500 list, with 112,896 compute cores and peak performance of 1.17 petaflops. In this study, we use twelve-month workload traces gathered on the system, which include around 700 thousand jobs submitted by more than one thousand users from 25 research areas. We investigate three categories of the workload characteristics: 1) general characteristics, including distribution of jobs over research fields and different queues, distribution of job size for an individual user, job cancellation rate, job termination rate, and walltime request accuracy; 2) temporal characteristics, including monthly machine utilization, job temporal distributions for different time periods, job inter-arrival time between temporally adjacent jobs and jobs submitted by the same user; 3) execution characteristics, including distributions of each job attribute, such as job queuing time, job actual runtime, job size, and memory usage, and the correlations between these job attributes. This work provides a realistic basis for scheduler design and comparison by studying the supercomputer's workload with new approaches such as using Gaussian mixture model, and new viewpoints such as from the perspective of user community. To the best of our knowledge, it's the first research to systematically investigate the workload characteristics of a petascale supercomputer that is dedicated to open scientific research.

Keywords: Workload Characterization and Modeling, Petascale Supercomputer, Academic Supercomputer, High Performance Computing

1 Introduction

As high performance computing (HPC) is becoming highly accessible, more researchers from a variety of research fields start to use supercomputers to solve large scientific problems. A supercomputer provides massive computational resources to achieve extremely fast processing speed in order to accelerate slow processes and generate outcomes with less time. The overall performance of a supercomputer depends heavily on the quality of the scheduling system. As indicated in [6], the performance of a supercomputer scheduler is greatly affected by the workload to which the supercomputer is applied, and there is no single scheduling algorithm working perfectly for all workloads. Therefore, workload characterization of a supercomputer is an important step to develop and evaluate scheduling strategies. Furthermore, a good understanding of workload characteristics can guide an HPC center to make decisions for purchasing or allocating specific hardware and software for the applications with different resource usage patterns.

Over the past decades, a variety of studies were conducted on workload analysis and modeling of parallel computers to evaluate scheduler performance [5] [11], and to predict job performance [6] [20]. Using historical data of workload traces that were recorded on real machines, statistical analysis of workloads was performed to understand the characteristics, such as distributions of job runtime and memory usage, of a single HPC system [3], a multi-cluster supercomputer [8], or a Grid computing environment [2]. Statistical workload models were also widely studied to generate abstract representation of job attributes, such as fitting distributions to job attributes [17] and modeling correlations between job attribute pairs [7]. Another type of workload models were developed on the basis of usage behavior classification. Wolter [21] experientially classified supercomputer users into three groups, and analyzed the job characteristics in each group. Song [18] used a mixed usage group model to classify jobs into predefined number of categories and investigated the workload traces in each category for job scheduling. Temporal characteristics of workloads in parallel systems were also analyzed and modeled. An infinite two-state Markov model was used to describe the characteristics of job inter-arrival in [13], which was extended to a n-state Markov modulated Poisson process to capture autocorrelations in [9]. Temporal locality of parallel system workloads was investigated in [14] to improve runtime prediction accuracy. However, the user community was relatively small in previous publications compared to a top ranking petascale supercomputers with a large user community and millions of job submissions. Previous researches failed to provide a comprehensive analysis of the characteristics of the user community, which greatly affects the patterns of the workload on a machine. For instance, the geographical distribution of users determines the temporal distribution of workload in a day.

In this paper, we present a comprehensive workload characterization of Kraken, an petascale supercomputer, which now ranks as the eleventh fastest computer in the world, and holds the title of world's fastest academic supercomputer [19]. Different from other top ranking HPC systems, Kraken is dedicated to academia, and it's user community consists of researchers from universities, research centers, institutes and laboratories. By analyzing Kraken workload, we can understand the patterns of how academia uses supercomputers to solve large scientific problems. Another distinguished characteristic of Kraken is the high utilization, which has been a sustained rate of 94 percents by average for the past year, and already contributed two billion CPU hours in total to open scientific research. The workload dataset used in this work contains about 700 thousand job traces, which were collected between November 2010 and October 2011 on Kraken. The large size of the dataset ensures that we are able to analyze the workload characteristics properly and come to solid conclusions. The job characteristics investigated in this work include general characteristics (e.g., distribution of the number of jobs submitted by a user), temporal characteristics (e.g., distribution of job inter-arrival time), and execution characteristics (e.g., distribution of job size).

The contributions of this paper are two fold. First, this paper analyzes a large workload dataset that was collected from the world's fastest petascale academic supercomputer. This is the first work, to the best of our knowledge, to systematically investigate the workload characteristics on a petascale supercomputer that is dedicated to open scientific research. Second, we provide new viewpoints and approaches with statistical models to comprehensively investigate a variety of the characteristics of Kraken workload. Besides investigating and modeling most of the workload characteristics introduced in previous research, we also analyze some characteristics which appear to be unique to Kraken, especially the characteristics of the user community, such as the user distribution and the distribution of the compute resource allocated to each research field. We also apply the Gaussian mixture models to fitting the distributions of some job attributes, such as job queuing time and actual runtime, which exhibit different patterns on a petascale supercomputer and cannot be modeled using a single distribution. We believe that this work is valuable in helping HPC centers to better understand the workload characteristics of a petascale supercomputer and the user community in academia, which is a necessary and important step to improve the overall performance of a supercomputer, and to prepare for the next generation of exascale HPC systems.

The remainder of the paper is organized as follows. Section 2 introduces the Kraken supercomputer and describes the workload dataset used in this study. Section 3 discusses the general workload characteristics of Kraken, including distribution of research fields, distribution of job size for an individual user, job distribution over queues, job cancellation rate, job termination rate, and walltime request accuracy. The temporal workload characteristics of Kraken are investigated in Section 4, which are monthly utilization, job temporal distribution in a year, a month, a week, and a day, and inter-arrival time between temporally adjacent jobs and jobs submitted by the same user. Section 5 talks about the execution characteristics, such as distributions of each job attribute, job queuing time, job actual runtime, job size, memory usage, and the correlations between these job attributes. Finally, Section 6 concludes the paper and reiterates the important characteristics of a petascale academic supercomputer.

2 Workload Traces of the Kraken Supercomputer

The historical workload traces of the Kraken supercomputer [15] was used in this study. Kraken is managed by the National Institute for Computational Sciences (NICS) at the Oak Ridge National Laboratory (ORNL) in the United States, and is funded by the National Science Foundation (NSF). It provides a petascale computing environment that is fully integrated with the Extreme Science and Engineering Discovery Environment (XSEDE). The supercomputer is a Cray XT5 system consists of 9,408 compute nodes with 112,896 compute cores, 147 terabytes of memory, and 3.3 petabytes storage. Each compute node contains 12 AMD 2.6 GHz Istanbul compute cores and 16 GB memory. The peak performance of the Kraken supercomputer is now 1.17 petaflops. Access to Kraken compute resources is managed by the Portable Batch System (PBS). The Lustre file system is used to support I/O operations, with the peak performance of 30GB/s. Moab is used to schedule jobs on Kraken, with preference to large core count jobs. Backfilling is applied on Kraken to allow smaller, shorter jobs to use remaining idle resources. The supercomputer is funded for the NSF community, which enables the scientific discoveries of nationwide researchers. In general, experienced researchers from academic or nonprofit organizations of the United States are eligible to request allocations of compute time of Kraken.

Table 1: A typical historical usage data that records the workload traces of the Kraken supercomputer. The exemplary workload dataset contains five instances. Each row represents a job, and each column denotes a job attribute.

job_id	user_name	account	nproc	mem_used	submit_time	start_time	end_time	walltime_req	walltime	cpu_hours	queue	type	software	research_area	description
0000001.nid00016	hao	U1-INDEX001	12	7588	2011-01-27 08:10:13	2011-01-27 09:26:03	2011-01-27 09:36:14	00:30:00	00:10:11	3.22	small	batch	—	Physics	Simulation in Physics
0000002.nid00016	haihang	U2-INDEX001	1	142664	2011-03-28 14:15:50	2011-03-28 14:16:14	2011-03-28 14:35:01	05:30:00	00:18:47	0.0658	hpss	batch	wrf	Earth Sciences	Simulation in Earth Science
0000003.nid00016	hao	U2-INDEX001	1032	16276	2011-04-16 01:28:41	2011-04-16 11:51:30	2011-04-17 11:51:56	24:00:00	24:00:27	24775.74	medium	batch	mpcugles	Physics	Energy conserving eddy simulation
0000004.nid00016	admin	SUPPORT	288	11836	2011-05-31 09:43:51	2011-06-04 21:00:20	2011-06-05 21:00:57	23:59:59	24:00:37	6914.96	small	interactive	—	Benchmark	Interactive benchmark
0000005.nid00016	hao	U1-INDEX002	98304	71812	2011-07-05 17:01:08	2011-07-07 11:11:13	2011-07-08 13:11:34	32:00:00	26:00:21	2556477.44	capability	batch	gadget	Earth Sciences	Large earth simulation

Table 2: Classification of jobs that are submitted to Kraken, which is determined by the number of compute nodes. Each job category is associated with a unique queue that is managed by job scheduler.

Queue	Compute Cores			$\text{Walltime}_{\text{Max}}$
	Minimum	Maximum	Percentage (%)	(hours)
Small	1	512	0–0.45	24.0
Medium	513	8,192	0.45–7.26	24.0
Large	8,193	49,536	7.26–43.88	24.0
Capability	49,537	98,352	43.88–87.12	48.0
Dedicated	98,353	112,896	87.12–100	48.0
HPSS	N/A	N/A	N/A	24.0

The dataset of the Kraken workload traces were collected between November 2010 and October 2011. It contains 693,829 job traces submitted by more than one thousand researchers from 25 research fields. This dataset was obtained by tracking all jobs that were submitted to the Kraken supercomputer and documenting the information that was related to the jobs. Some instances of the Kraken workload traces are listed in Table 1, each of which contains sixteen attributes. The attributes *job_id*, *user_name*, and *account* are the unique identifiers of a job, an user, and an account, respectively. The attributes *submit_time*, *start_time* and *end_time* record the times when a job is submitted, executed, and finished, respectively. Jobs are automatically classified into several categories according to the number of requested compute nodes, denoted by *nproc*. Each category is associated with a *queue* that indicates the priority. The definitions of the job categories and queue types are described in Table 2. Because each compute node of Kraken contains twelve cores, the attribute *nproc* must be a multiple of 12, except for the jobs in the queue for accessing the High Performance Storage System (HPSS). These jobs are executed to transfer files to a storage system using a batch file. No compute nodes on Kraken are allocated to the jobs in HPSS queue. The attribute *walltime_req* is the requested walltime of a job, which is required to be estimated and provided by a user before submitting a job. The attributes *walltime*, *mem_used* and *cpu_hours* represent actual runtime, consumed memory and CPU hours of a job, respectively. The attribute *type* takes a boolean value (i.e., interactive or batch), which indicates whether a user has interactive access to the compute resources. The attribute *software* describes the name of the software or package, if any, used by a job. For instance, the job 0000003.nid00016 used MPCUGLES, which is a software for energy-conserving large eddy simulations. The attribute *research_area* indicates the research field of the target problem, or to which the user belongs. At last, the attribute *description* explains the objective of the project.

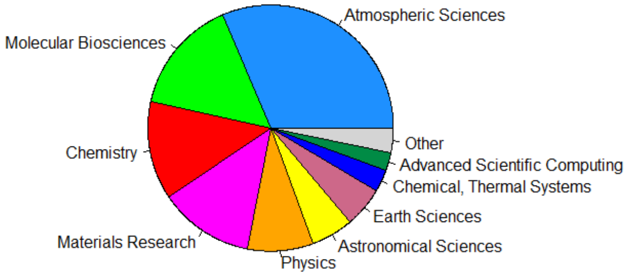


Fig. 1: Pie chart of the research fields sorted by the number of jobs submitted to Kraken in each field.

3 Job General Characteristics

We first analyze the general characteristics of jobs running on Kraken and its user community. The objective of this analysis is to figure out how users from different research fields utilizing the petascale supercomputer, and how well they performed on using the supercomputer for open scientific research.

3.1 Distribution of Research Fields

There were 1,111 users from 473 different accounts submitted jobs to Kraken during the year when the workload traces were collected. The users scattered in 25 different research fields from all over the United States. The proportion of each research field sharing the supercomputer is illustrated in Figure 1. The top five research fields using Kraken were Atmospheric Sciences, Molecular Biosciences, Chemistry, Materials Research and Physics, which took over 75 percents of the job submissions to Kraken. However, the number of Kraken users in a research field was not necessarily proportional to the number of job submissions in the research field. For example, the research field of Molecular Biosciences had the most 205 Kraken users, but only 90 users came from Atmospheric Sciences who submitted the most jobs. On the other hand, the number of job submissions and the number of users in a research field were positively correlated, i.e., a larger user group often resulted in more job submissions.

3.2 Distribution of Number of Jobs for a User

Typically, a supercomputer user only submits jobs to address similar problems in the same discipline. On the other hand, it is very common that a user submits multiple jobs at one time or across multiple times. The distribution of the number of jobs submitted by a user is depicted in Figure 2 in a logarithm scale. On Kraken, there were 693,829 job submissions recorded in a year, and an average of 624 jobs were submitted by each user. However, it should be noted that the number of jobs submitted by a user was not uniformly distributed. Oppositely,

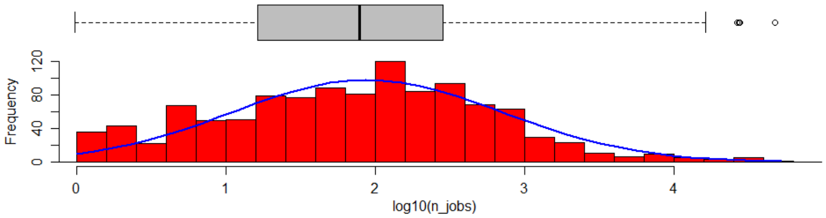


Fig. 2: Histogram (red rectangles) of the number of jobs submitted by a Kraken user, with density estimate (blue curve) and statistical summary (gray boxplot).

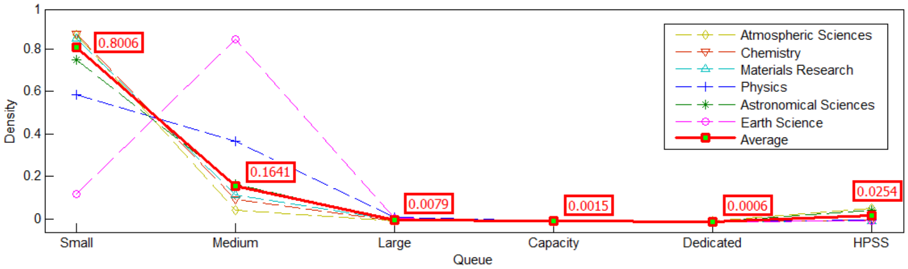


Fig. 3: Distribution of jobs over queues in different research fields. The number in a red rectangle indicates the probability that a job belongs to a queue.

the most active 10% users contributed 79.2% job submissions to the workload. The statistical summary of the jobs submitted by a user is also plotted with a boxplot in Figure 2, which indicates that 50% users had job submissions between 20 to 600 in a year. The number of jobs submitted by a user can be modeled using the log-normal distribution, i.e., the logarithm of the number of job submissions conforms to the Gaussian distribution. The parameters of the distribution can be computed using the maximum-likelihood estimation (MLE). The probability density function (PDF) is plotted with the blue curve in Figure 2.

3.3 Distribution of Jobs over Queues

The probability distribution of jobs over queues and the probability distributions of jobs over queues within different research fields are depicted in Figure 3. In general, the probability of the jobs belonging to a queue consistently decreased with the increase of the job size in the queues of *Small*, *Medium*, *Large*, *Capability*, and *Dedicated*, as shown by the average distribution in Figure 3. The probability that a job belongs to a queue is shown in a red rectangle in the figure. On average, over 80% jobs belonged to the queue of type *Small*, and only 1% jobs consumed significant compute resources, which belonged to the queues of type *Large*, *Capacity*, or *Dedicated*. As for the distributions of jobs over queues in each individual research field, although the distribution in Earth Science had

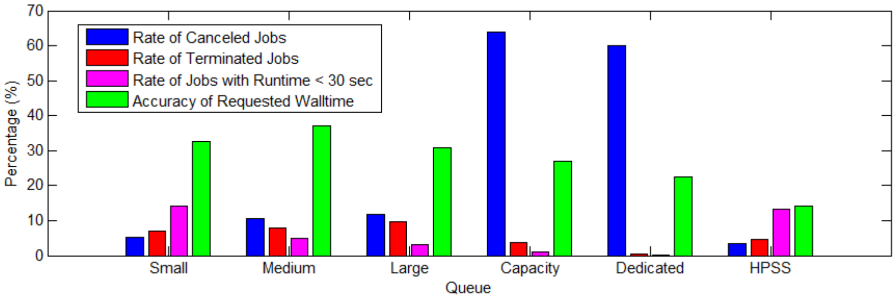


Fig. 4: General characteristics of workload traces in different queues.

great difference from the average distribution, in which case the probability that a job was in the queue of type *Medium* was greater than the probability that the job belonged to the queue of type *Small*, in general, most of the per-discipline distributions were quite similar to the average distribution, such as Atmospheric Sciences and Materials Research. Consequently, we can model the distribution of jobs over queues with a single multinomial distribution for all research fields, with each model parameter equal to the value of the corresponding average probability of a job belonging to a queue.

3.4 Cancellation/Termination Rate and Walltime Request Accuracy

The general characteristics of the jobs in different queues are also investigated in this study, which include job termination rate, job cancellation rate, and average accuracy of walltime estimation. In most HPC systems, a job can be canceled manually by a user before the job starts executing, in which case the job actual runtime is zero. On the other hand, a job can be forcefully terminated by the HPC system if the job runs out of the requested walltime, in which case the job actual runtime is equal to the requested walltime. Moreover, a job might fail and exit during its execution due to some runtime error, in which case the job actual runtime is often significantly smaller than the requested walltime. If a job neither completes correctly, nor provides meaningful results, it is considered incorrect. Because the incorrect jobs are always misleading for meaningful analysis, we evaluate the accuracy of the requested walltime with correct jobs. A job is considered *correct*, if it belongs to the job set:

$$\begin{aligned}\mathcal{J}_c = \{j \mid (j.mem\_used \neq 0) \wedge (j.walltime > 30)\\ \wedge\ (j.walltime < j.walltime\_req)\}\end{aligned} \tag{1}$$

where "." represents attribute relationship, and the unit of the attribute *walltime* is second. The three literals in (1) remove the canceled jobs, terminated jobs, and a part of jobs with runtime errors at the beginning of job execution, respectively. The second literal also contributes to remove some "hello world" jobs, in which case a user often does not care about job runtime. On the other hand, it should

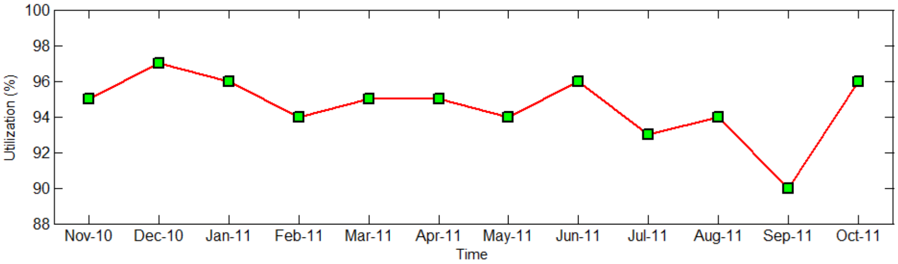
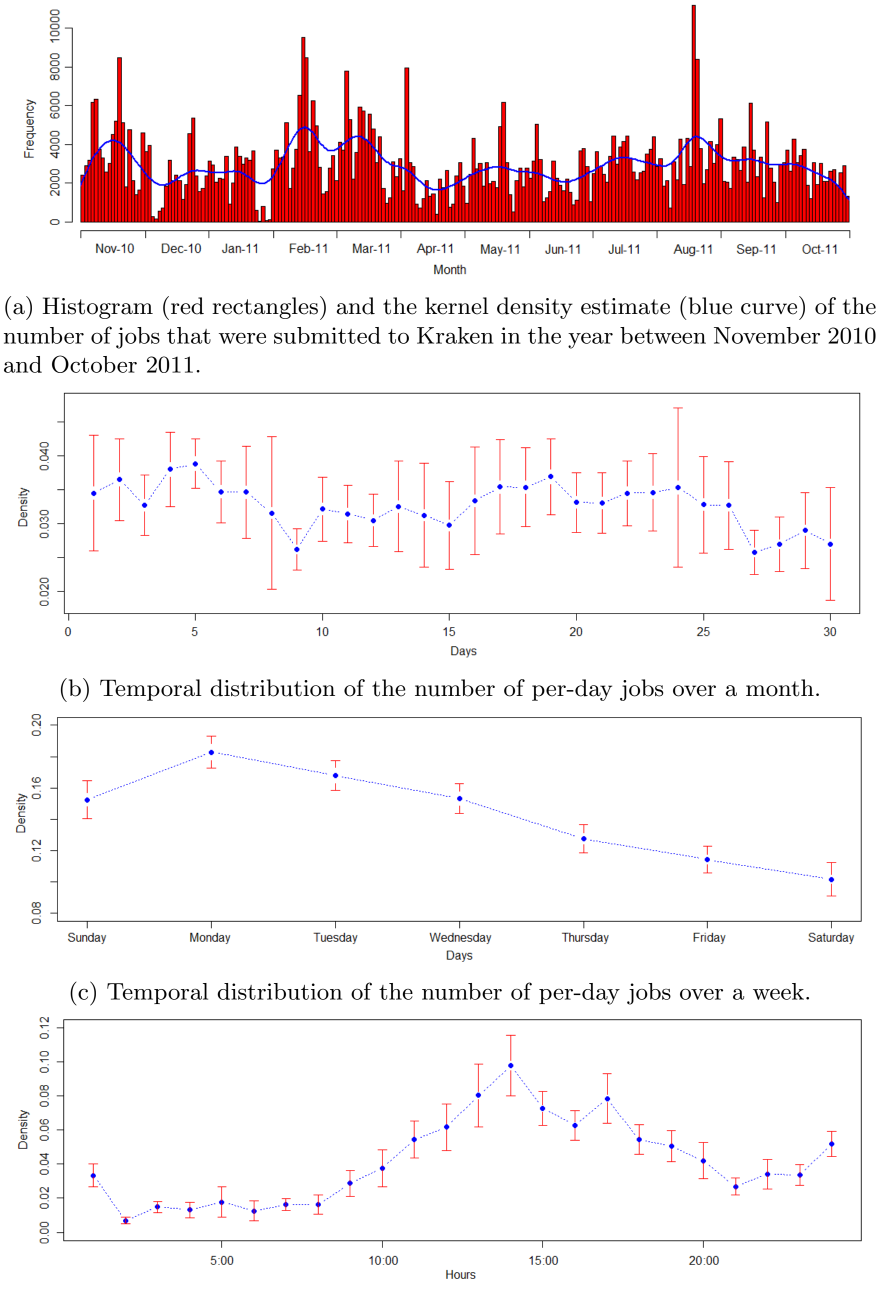


Fig. 5: Monthly utilization of the Kraken supercomputer over a year

be noted that this definition does not remove all incorrect jobs, such as jobs with runtime errors at the end of their execution. These remaining incorrect jobs are treated as noise in this work. To quantitatively measure the accuracy of the requested job runtime, the walltime request accuracy (WRA) is applied. For a correct job j in $\mathcal{J}_c$, WRA is defined as:

$$WRA(j) = \frac{j.walltime}{j.walltime\_req} \times 100\% \tag{2}$$

The rates of canceled jobs, terminated jobs and jobs with runtime less than 30 seconds, along with the WRA for each queue are illustrated in Figure 4. A most obvious phenomenon is that the jobs requesting significant compute resources, in the queues of type *Capability* and *Dedicated*, had the highest cancellation rate of more than 60%, which might be resulted from the long queuing time. Moreover, the jobs in the queue *Capability* and *Dedicated* had the lowest termination rate of less than 5%, and the lowest 30-second job quitting rate of less than 1%, along with the lowest WRA. This phenomenon can be partially explained by the fact that researchers submitted these jobs often had rich HPC experience to reduce errors in the code, and they also tended to request longer walltime to decreased the probability that their jobs were forcefully killed by the system. Third, a high rate of the jobs quitting within 30 seconds in the queues of type *Small* and *HPSS*, which at least doubled the same rate of other queues, indicates that new users of Kraken were more probable to submit small jobs to figure out how to access the compute and storage resources. Consequently, it is necessary for the scheduling system of an HPC system to have backfilling policy, which allows small jobs to be backfilled. The last important observation is that the overall accuracy of the requested walltime was around 33%, which was typically estimated and provided by a user. The low accuracy of the job runtime estimate indicates that many users were lack of experience with the HPC system. It is of great necessity for HPC centers to provide users with more supports, such as a recommendation system to guide users to predict the runtime of their jobs.



(a) Histogram (red rectangles) and the kernel density estimate (blue curve) of the number of jobs that were submitted to Kraken in the year between November 2010 and October 2011.

(b) Temporal distribution of the number of per-day jobs over a month.

(c) Temporal distribution of the number of per-day jobs over a week.

(d) Temporal distribution of the number of per-hour jobs over a day.

Fig. 6: Temporal distributions of the Kraken workload.

4 Job Temporal Characteristics

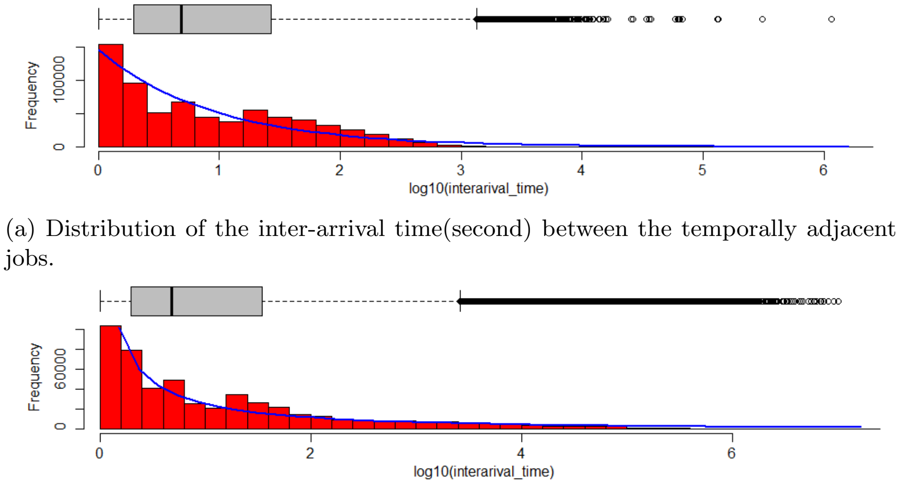
In this section, we investigate the temporal characteristics of the jobs submitted to Kraken, which include 1) the monthly utilization of the supercomputer in a year; 2) temporal distributions of jobs in a year, a month, a week, and a day; and 3) the inter-arrival time of job submissions. The goal is to unveil the workload dynamics, i.e., the time-based patterns in the workload.

4.1 Supercomputer Utilization

The monthly utilization of Kraken over a year is depicted in Figure 5. The system utilization for each month was very consistent on Kraken, which oscillated between 90% and 97%, and an average monthly utilization of 94.6% was observed. Kraken has been providing two thirds of cycles that are available to the national research community funded by NSF in the United States. Projects are selected by xRAC [22] through process which is designed to provide independent merit-review of proposals for the XSEDE. At NICS, teams of operations, user support, computational science and education, outreach and training provide support at various levels. The combination of project selection, management, support, and expertise provided on site at NICS result in such high utilization rate for the petaflop supercomputer, which serves as a good example to increase the overall utilization of an HPC system to open scientific research.

4.2 Temporal Distributions

The temporal distributions of the Kraken job submissions over a year, a month, a week, and a day are illustrated in Figure 6a, 6b, 6c, and 6d, respectively. In Figure 6a, the kernel density estimate [16] is computed to fit the dataset and to provide a smooth representation. An important observation from this figure is that the job submissions were not uniformly distributed over time, and the bursty behavior of job arrivals can be observed, such as the burstiness of the job arrivals in the middle of February and August, 2011. This phenomenon is possibly caused by the training activities using the supercomputer at the beginning of spring and fall semester. On the other hand, the low job arrival can be also observed, which might be caused by system upgrade, system failure, or occupation by capacity or dedicated jobs. An interesting observation is that the number of job submissions was not necessarily correlated with the utilization. For instance, running dedicated jobs might lead to very high system utilization with very low job counts. At last, the workload of Kraken did not exhibit obvious patterns over a year. The temporal distribution of the daily job arrivals over a month is computed using the 12-month Kraken workload dataset, and the standard deviations are also calculated to indicate the reliability of the estimates, which are presented with the error bars in Figure 6b. The large errors for the daily job arrival estimates indicate that the workload of the same day (e.g., the 8th day) in a month was not stationary, which varied greatly from month to month.



(a) Distribution of the inter-arrival time(second) between the temporally adjacent jobs.

(b) Distribution of the inter-arrival time(second) of jobs from the same user.

Fig. 7: Comparisons between the inter-arrival time of the continuous job stream and the inter-arrival time of the job sequence submitted by the same user.

However, the workload exhibited obvious and stationary patterns in weekly cycles. As illustrated in Figure 6c, the daily workload presented obvious cyclical patterns in a week, which started at the maximum on Mondays, then constantly decreased to its minimum on Saturdays, and bounced back on Sundays towards the peak. Moreover, the small standard variations presented by the short error bars in Figure 6c also indicate a high degree of consistency in the daily workload over a week. In general, the workload at the end of a week was quite different from the workload at the beginning of a week. The maximum daily workload on Mondays was 1.8 times of the minimum daily workload on Saturdays. Similarly, the hourly workload also showed stationary, obvious, but more complicated patterns in daily cycles, which is illustrated in Figure 6d. The peak of the hourly workload arrival appeared at around 3:00 PM in the afternoon. In general, the hourly workload in the midnight was much lower than the hourly workload in the afternoon. Another interesting phenomenon is that researchers tended to submit more jobs before getting off work and going to sleep, which is well supported by the local maximum at around 5:00 PM and 12:00 PM in Figure 6d, respectively. However, it should be noted that the cyclical patterns of hourly workload over a day on a supercomputer is significantly dependent on the spatial distribution of its users.

4.3 Inter-Arrival Time

We analyze the inter-arrival time of the jobs submitted in sequence to Kraken, and compare the results with the inter-arrival time of the jobs submitted by the same user. The inter-arrival time distributions are illustrated in Figure 7, in which the inter-arrival time is plotted in a log scale to reduce a wide range to a manageable size. From Figure 7a, we can observe that the time intervals between two temporally adjacent job submissions were very small. As indicated by the statistical summary in the box plot, around 25% inter-arrival times were less than three seconds, around 50% inter-arrival times less than six seconds, and around 75% inter-arrival times less than 30 seconds. The large inter-arrival time might be resulted from system upgrade or system failure, in which case, users cannot access or submit jobs to the supercomputer. The inter-arrival time distribution of the jobs submitted by the same user exhibited similar characteristics to the distribution of the overall inter-arrival time, as shown in Figure 7b. This distribution indicates the temporal locality, or burstiness, of the jobs submitted by an individual user, which is well supported by the observation that 25% inter-arrival times were less than two seconds, and 50% inter-arrival times were less than five seconds. The job burstiness was generally caused by the fact that users usually repeated submitting the same jobs or jobs with similar attributes in a short time span. Due to the similarity of these distributions, a single statistical model can be applied to model the inter-arrival times of the workload. In this work, we apply a Weibull distribution to model the logarithm of the job inter-arrival times, with MLE to estimate the model parameters. The results are shown with the blue curves in Figure 7.

5 Job Execution Characteristics

In this section, the job execution characteristics are investigated. First, we analyze the characteristics of each individual job attribute in the Kraken workload, including actual job queuing time, actual job runtime, number of compute nodes used by a job, and total memory consumed by a job. Then, the correlations between these job attributes are studied. The results of the characteristics of each job attribute and the correlations between job attributes are drawn with scatter-plot matrices which is depicted in Figure 8. Each diagonal plot presents the univariate histogram of the logarithm of one job attribute, along with the fitted distribution. The plots above the diagonal, with different shapes and colors, present the queue types to which each data point belongs, which also indicate the geographical distributions of the jobs in the 2-dimensional attribute space. Although plots below the diagonal contain the same data points as the plots above the diagonal, a smooth curve is fitted to each plot using the locally weighted scatter plot smoothing [4] to intuitively present the correlations between the job attributes. It should be noted that, because incorrect jobs are always distracting in investigating the underlying distributions of the job attributes, we intentionally remove the incorrect jobs, and only use the correct jobs in the set $\mathcal{J}_c$, defined in (1), for the following analysis. Moreover, the characteristics of the jobs in the HPSS queue are not analyzed, because these jobs do not consume any compute cores and memory, and users generally do not care about the performance of such jobs.

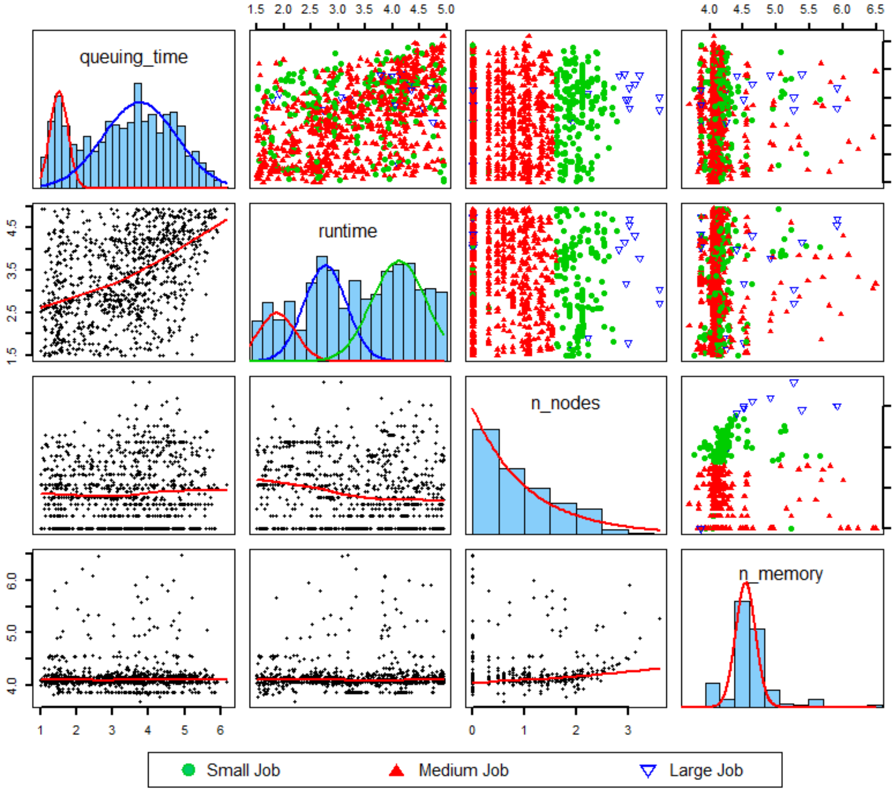


Fig. 8: Execution characteristics of actual queuing time, actual runtime, requested compute nodes, and consumed memory in the Kraken workload. All quantities in the figure are in a logarithmic scale based on 10.

5.1 Individual Attribute Characteristics

When analyzing the characteristics of each individual job attribute and plotting the scatter-plot matrices, the job submissions belonging to the queues of type *Capacity* and *Dedicated* are also intentionally removed, because these jobs only take over 0.21% of the entire job traces in the Kraken workload, which are always treated as outliers by the statistical models that are used to fit the individual attribute distributions.

We first analyze the characteristics of the job attribute *queuing time*, denoted as T_q, which is the actual waiting time of a job in a queue between job submission and execution. Different from the conclusions in previous research that a single distribution, such as the log-uniform distribution discussed in [10], can be used to present the queuing time distribution in a small system, for the workload of Kraken, a petascale supercomputer, any single distribution does not well fit to the job queuing time. In this study, the Gaussian mixture model (GMM), with two Gaussian component distributions, is applied to model the logarithm of the job queuing time, as illustrated in Figure 8-$(1, 1)$. The parameters of the GMM model are estimated using MLE. The data of the job queuing time are well fitted by the GMM model with two components, which actually shows the characteristics of the scheduling queues. The first Gaussian component with a smaller mean is resulted from the backfilling policy of the scheduler, in which case the jobs requesting a small portion of compute resources can be backfilled and executed faster. The second Gaussian component represents the distribution of the queuing time of the non-backfilled jobs. On average, a job needs to wait in the queue for 34.5 minutes before starting execution.

Similarly, the distribution of the actual job runtime, denoted as T_r, cannot be simply modeled with a single distribution as well, such as the log-uniform [10], hypergamma [12], or Webull [1] distributions. In this study, we apply the GMM model to present the distribution of the logarithm of the actual job runtime, as shown in Figure 8-$(2, 2)$. The GMM model contains three Gaussian components, which well fits the distribution of the logarithm of the job runtime. A possible explanation for the three Gaussian components is that the actual runtime of the jobs in each queue conforms to a logarithm-normal distribution, in which case the logarithm of the actual runtime of all jobs conforms to a mixture Gaussian distribution. The average job actual runtime is 36.8 minutes.

The characteristics of job size, denoted as N_n in number of compute nodes, for the Kraken workload are also investigated. On Kraken, there are 12 cores per node, and it is not possible to allocate part of a node. After applying logarithm on the number of nodes, the resulted distribution still has a short tail, as shown in Figure 8-$(3, 3)$. This tailed distribution indicates that there are a great number of small jobs along with few large jobs are submitted to Kraken, which is consistent with the results shown in Figure 3. In more detail, there are 80.1% jobs submitted to the queue of type *Small*, and 26.5% jobs requesting only one node with twelve compute cores for learning HPC or debugging code. Consequently, we argue that, while the scheduler is collecting resources for larger jobs, backfilling policy is important for a petascale supercomputer to make jobs with short wall-clock limits and small core counts to start execution faster, in order to respond to the users with small job submissions quickly.

Memory usage is defined as the maximum amount of physical memory that is consumed by a job at some time point during its execution, which is recorded by the PBS on Kraken. Kraken is a distributed-memory system, with 16 GB memory for each compute node. The univariate histogram and the fitted distribution of the logarithm of the memory usage are depicted in Figure 8-$(4, 4)$. In this study, a single Gaussian distribution is applied to fit the memory usage represented in a log scale. Although the Gaussian distribution fits most memory usage data very well, it cannot well model the heavy tail, which actually indicates that the entire workload is dominated by the computationally-intensive jobs with few memory-intensive jobs. A possible solution to address this problem is first to cluster the jobs into computationally-intensive or memory-intensive categories, then to model each job category separately. On Kraken, an average

Table 3: Correlation coefficients between job attributes

Correlation	$T_q - T_r$	$T_q - N_n$	$T_q - M_u$	$T_r - N_n$	$T_r - M_u$	$N_n - M_u$
Pearson	0.2439	0.0061	0.0210	-0.0481	-0.0177	0.0620
Log-Pearson	0.3946	0.0768	0.0286	-0.2092	0.0070	0.2501
Spearman	0.3951	0.0593	0.0367	-0.2293	-0.0716	0.4893

memory usage for the computationally-intensive jobs is around 12.3 GB, and the memory-intensive jobs usually consume over 80 GB memory.

5.2 Attribute Correlations

The correlations between job attributes are computed using all jobs in the queues of type *Small*, *Medium*, *Large*, *Capacity*, and *Dedicated* in the job set $\mathcal{J}_c$. Pearson product-moment correlation coefficient is the most commonly used measure of the strength of linear dependence between object attributes. Pearson correlation works well in the cases that the values of the attributes are roughly normally distributed. But Pearson correlation is very sensitive to the strong outliers, and works poorly on modeling correlations between data sampled from heavily tailed distributions. Thus, we also apply the Spearman's rank correlation coefficient to study the correlations between job attributes, which is a non-parametric correlation measure that is less sensitive to strong outliers. The correlations between job attributes are intuitively presented with the red smooth curves in the plots below the diagonal in Figure 8. We also applied the Pearson measure on both the raw job attributes and the logarithmic-scaled job attributes. Because Spearman measure is invariant to the logarithm operation, the Spearman correlation coefficients are only computed using the raw job attributes. The resulted coefficients are quantitatively listed in Table 3.

A strong positive correlation between job queuing time T_q and actual runtime T_r is observed from all three correlation measures, which can also be intuitively observed from the curve with a large positive slope in Figure 8-$(2, 1)$. Because the Spearman correlation and Pearson correlation have similar values when the data are roughly normally distributed with few outliers, the almost identical coefficients computed from the Log-Pearson measure and the Spearman measure indicate that the GMM model is a good model for fitting logarithm-scaled job queuing time and actual runtime. The second observation of job attribute correlations is that the job size N_n is positively correlated, in a non-linear manner, with the memory usage M_u, which can be observed from the curve with a positive slope in Figure 8-$(4, 3)$. A possible explanation for this observation is that Kraken is a distributed-memory system which always allocates 16 GB memory with each compute node allocation. Because each core can only access to the memory on the same node, the total allocated memory is always proportional to the number of allocated nodes. In this sense, a user tends to use more memory, when more compute cores are allocated with a larger amount of memory available. Third, there is a negative correlation between job size N_n and job actual runtime T_r, which can be easily observed from the curve with a negative slope in Figure 8-$(3, 2)$. This phenomenon indicates that, in general, using more compute resources reduces the wall-clock time of a job, which is actually the initial reason to use a supercomputer. Fourth, job queuing time T_q can be considered independent with job size and memory usage, as indicated by the coefficients that are close to zero. Similarly, job actual runtime T_r is statistically independent with job memory usage M_u. These independences indicate that job queuing time and memory usage are more random than other job attributes, which means it is harder to predict these two job attributes. The fitting curves with slopes close to zero graphically indicate the independences between the job attribute pairs, as shown in Figure 8-$(3, 1)$, 8-$(4, 1)$, and 8-$(4, 2)$. Finally, for the correlation measures, the Spearman measure generally performs better than the Pearson measure in the sense of detecting the non-linear correlations between job attribute pairs.

6 Conclusion

In this paper, the workload characteristics of a petascale academic supercomputer is comprehensively and systematically investigated, based on the twelve-month workload dataset collected from the world's most powerful academic supercomputer, with around 700 thousand jobs submitted by more than one thousand users from 25 research fields. The general characteristics, temporal characteristics, and execution characteristics of the workload traces are investigated and well represented with statistical models, with the objective of providing a realistic basis for scheduler design and comparison, as well as helping HPC centers to better understand the characteristics of workload and user community. For the highly-utilized petascale academic Kraken supercomputer, the most important observations and conclusions are reiterated as follows, according to the order they are discussed in the paper:

- Jobs are not uniformly distributed over research fields and users, and several users from a few research fields usually dominate the job submissions.
- The distributions of jobs over queues are very similar in different research fields. Thus, the same multinomial distribution can be applied, in all research fields, to model how jobs distribute over queues.
- Users with large job submissions generally have more HPC knowledge than users only with small job submissions. However, in general, the job runtime estimate is shown to be highly inaccurate.
- The Kraken workload does not show stationary patterns over a month or a year. But the workload exhibits obvious patterns in weekly or daily cycles.
- Due to the strong similarity between the patterns of overall job inter-arrival time and inter-arrival time of the jobs submitted by a user, a single statistical model is enough to represent the distributions of job inter-arrival time.
- Backfilling policy is important for a petascale supercomputer to enable small jobs to run effectively, which take over 80% job submissions in Kraken.

- The distributions of job queuing time and actual runtime on a multi-queue petascale supercomputer cannot be modeled with a single distribution. Gaussian mixture models perform well on modeling the log-scaled attributes.
- Job actual runtime is positively correlated with job queuing time, and negatively correlated with job size. Job memory usage has positive correlation with job size. Job queuing time can be considered independent with job size and memory usage. Job actual runtime and memory usage are also statistically independent.

References

1. S.-H. Chiang and M. K. Vernon. Characteristics of a large shared memory production workload. In *Revised Papers from the 7th International Workshop on Job Scheduling Strategies for Parallel Processing*, pages 159–187, 2001.
2. K. Christodoulopoulos, V. Gkamas, and E. Varvarigos. Statistical analysis and modeling of jobs in a grid environment. *Journal of Grid Computing*, 6:77–101, 2008.
3. W. Cirne and F. Berman. A comprehensive model of the supercomputer workload. In *IEEE International Workshop on Workload Characterization*, pages 140–148, Dec. 2001.
4. W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.
5. Y. Denneulin, E. Romagnoli, and D. Trystram. A synthetic workload generator for cluster computing. In *International Parallel and Distributed Processing Symposium*, page 243, april 2004.
6. D. G. Feitelson. Workload modeling for performance evaluation. In *Performance Evaluation of Complex Systems: Techniques and Tools*, volume 2459, pages 114–141, 2002.
7. H. Li. Workload dynamics on clusters and grids. *The Journal of Supercomputing*, 47:1–20, 2009.
8. H. Li, D. Groep, and L. Walters. Workload characteristics of a multi-cluster supercomputer. *Job Scheduling Strategies for Parallel Processing*, 3277, 2004.
9. H. Li and M. Muskulus. Analysis and modeling of job arrivals in a production grid. *SIGMETRICS Perform. Eval. Rev.*, 34:59–70, March 2007.
10. H. Li, L. Wolters, and D. Groep. Workload characteristics of the das-2 supercomputer, Jun. 2004.
11. V. Lo, J. Mache, and K. Windisch. A comparative study of real workload traces and synthetic workload models for parallel job scheduling. In *Job Scheduling Strategies for Parallel Processing*, volume 1459, pages 25–46. 1998.
12. U. Lublin and D. G. Feitelson. The workload on parallel supercomputers: Modeling the characteristics of rigid jobs. *Journal of Parallel and Distributed Computing*, 63:2003, 2001.
13. E. Medernach. Workload Analysis of a Cluster in a Grid Environment. In *Job Scheduling Strategies for Parallel Processing*, volume 3834. 2005.
14. T. Minh and L. Wolters. Modeling parallel system workloads with temporal locality. In *Job Scheduling Strategies for Parallel Processing*, volume 5798, pages 101–115. 2009.
15. National Institute for Computational Sciences. Running jobs on Kraken. `http://www.nics.tennessee.edu/node/16`. Accessed: 11/11/2011.

16. M. Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832–837, Sept. 1956.
17. B. Song, C. Ernemann, and R. Yahyapour. Modelling of parameters in supercomputer workloads. In *International Conference on Architecture of Computing Systems*, pages 400–409, 2004.
18. B. Song, C. Ernemann, and R. Yahyapour. User group-based workload analysis and modelling. In *IEEE International Symposium on Cluster Computing and the Grid*, volume 2, pages 953–961, may 2005.
19. Top500. Application area share for 06/2011. `http://www.top500.org/list/2011/11/100`. Accessed: 11/11/2011.
20. D. Tsafrir, Y. Etsion, and D. Feitelson. Modeling user runtime estimates. In *Job Scheduling Strategies for Parallel Processing*, volume 3834, pages 1–35. 2005.
21. N. Wolter, M. McCracken, A. Snavely, L. Hochstein, T. Nakamura, and V. Basili. What's working in HPC: Investigating HPC user behavior and productivity. *CT-Watch Quarterly*, 2(4A), 2006.
22. xRAC. xrac. `http://www.teragridforum.org/mediawiki/index.php?title=XRAC`.