Towards Effective Selection of Collective XSEDE Resources

Vishal Shah
Rutgers University
Piscataway, NJ 08854
vishal.b.shah@rutgers.edu

Antons Treikalis
Rutgers University
Piscataway, NJ 08854
antons.treikalis@rutgers.edu

Shantenu Jha
Rutgers University
Piscataway, NJ 08854
shantenu.jha@rutgers.edu

ABSTRACT

Typically users view XSEDE as a set of individual isolated resources and generally do not consider the full collective capabilities of the resources available on XSEDE when determining where to run their workload. This results in sub-optimal resource selection, increased time-to-completion of workloads and inefficient resource utilization. Even though we find reasons not to be optimistic about prediction services, we believe that selecting resources by taking into account different measures of load (e.g. utilization) and availability (e.g. job queue waiting time) can alleviate the aforementioned problems. We propose and investigate different measures of load and availability to understand the correlation between the different measures. Our experiments are performed on multiple XSEDE resources and investigate these correlations with a view to determining efficient resource selection across all of XSEDE resource. Our experiments use historical information, current availability as well as future prediction services to determine appropriate resource selection. We use several existing services and tools such as NSF's XDMoD for much of the historical information and current availability, as well as Karnak for job-queue waiting predictions. This paper touches upon four specific questions: (i) how accurate are prediction tools, (ii) what is the correlation between queue waiting times and utilization, (iii) what are the correlation properties of the different measures of loads and availability, and, (iv) what is the time-scale over which the past is an indication of the future. Our work provides preliminary but useful and general basis for characterizing XSEDE resources so as to open the way towards ascertaining optimal metrics of utilization and closed-form answers.

General Terms

Performance, Design

Keywords

HPC, Distributed Computing, Performance, Prediction, Correlation

XSEDE14 'July 13 - 18 2014, 13 Atlanta, GA USA

http://dx.doi.org/10.1145/2484762.2484819.

1. INTRODUCTION

Distributed Cyberinfrastructure such as XSEDE are a federation of multiple individual resources. Whereas the management and provisioning of these resources is independent and isolated (and rightly so), the utilization of the resources does not have to be isolated. In some ways the grand vision of federated resources is the situation where resources can be utilized collectively such that the ẅhole is greater than the sum of its parts̈. This is not a reference to the raw total capacity (which by definition must be equal to the sum of its constituent parts) but is a reference to the ability of an end-user to be able to get better quality-of-service and/or lower time-to-completion (TTC) or even better estimates of the waiting times.

Better estimates of XSEDE's collective capacity and waiting times, are a first step towards more effective p̈lanningöf the science by the user. For example, if a Chemist can determine that she will be able to run a 100 Molecular Dynamics simulations in the next 48 hours, she will be better prepared to organize her scientific investigation; however, currently, most Chemists do not carefully plan their scientific investigation upon an estimate of when the required set of simulations will finish. At best, there is some very careful over-estimation (or careless under-estimation) involved. Either is an unsatisfactory state of planning and leads to inefficiency of sorts. There is clear need and role for more precise and accurate estimation of the collective capacity of XSEDE and thereby estimate of TTC of a set of tasks. Although a federation of resources, XSEDE is still presented to the user as an aggregation of independent, individual resources. There is currently no information service or capability based upon past performance, current status/availability or future estimates that guides an end-user towards effective distribution of their workload over XSEDE.

As mentioned there are three phases to the complexity of the solution: the first is the simple case when a resource is chosen a priori without any information; the second is when a resource is chosen out of a set of resources based upon some measure of goodness; the third situation is when a resource is chosen and the workload is decomposed collectively. In this paper, we will be concerned with only the second phase of the solution.

The ability to match applications to HPC resources such as Stampede [1] and Trestles [2] on XSEDE optimally is an important requirement. Although optimality in itself could be determined along different metrics and measures, one of the common metrics is the ẗotal time to completion(̈TTC). Being able to lower the TTC is for most scientists a measure of goodness of the underlying resource! Effective and efficient mapping of resources is also, at a minimum, an economic argument, for it will allow the optimal utilization of the resources, and thus is also in the resource providers interest. An important distinction between individual scientists and a r̈esource provider(̈RP) as a whole is that the former is only interested in optimizing the TTC of their job. The RP would like to optimize the entire workload on a resource so as to bring about maximal satisfaction and optimize overall utilization of the resource.

Currently users of XSEDE rely upon s̈eat of the pantsëxperience to determine where to submit their workloads optimally. There is little by the way of guidance, empirical or theoretical, to determine which resources should be chosen. To solve the full blown problem is a major intellectual challenge. It requires a deep understanding of the characteristics of application workload: their flexibility and ëlasticity;̈ the metrics of s̈atisfactionäs well as the characterization of the properties of the underlying resources, e.g., what is the capacity of the resources, what is the execution time of a given workload on a given resource, how long would a given task have to wait before it would execute. Evidently, this is a multi-dimensional problem, but for the sake of simplicity and making progress we will, for the purposes of this paper, reduce it to the single dimension of understanding the properties of the resources.

There has been significant work in the area of q̈ueue wait timep̈rediction [3, 4, 5, 6]. However, the measure-of-goodness of all prediction approaches is at best poor or insufficiently specified. In general, we believe that predicting individual waiting times is an unproductive venture. It is difficult to establish exactly why this might be the case, as underlying reasons will vary between different prediction approaches. Although we can't describe accurately the causes of this deviation, we can have empirical evidence to suggest that determining which resources to use via a commonly-available prediction service is no better than choosing a resource at random! This is not a commentary on the prediction service, but a reflection on the difficulty of generalizing prediction services to different systems, characteristics and load profiles. It also points to the inherent challenges of prediction at the individual job level.

Thus, if the future is unpredictable, at least without hindsight, a natural logical follow-on question is: without resorting to prediction, how can we best determine which resources to utilize and how? Can we use some available information – either historical or current, to get a measure of the global status of XSEDE and determine its temporal properties? If so, can this available information tell us how to plan our workload execution?

This work attempts to answer the above question in its simplest incarnation. Whereas the full blown problem has many dimensions and aspects, here we attempt to incorporate the temporal properties of XSEDE resources. We do not have closed form answers; however, we have empirically determined insight. We are interested in determining appropriate/novel m̈easuresöf load/utilization and response, so as to discern any possible connection between system load utilization and availability. Thus we will be concerned with weak causation, i.e., correlation between these different measures/variables.

Two underlying assumptions of our work are that it is important to look for metrics other than the Queue Waiting Time of individual jobs submitted to individual XSEDE resources and the fact that it is possible (and meaningful) to look at the capacity of distributed resources as a collective whole rather than a single resources.

It is also important to note that we will be concerned with optimality from the individual user's point of view and not net utilization of resources, though it will be interesting to inquire into the relationship between the two. For example, how are the two types of optimality related? Is the former a necessary condition for the latter, or might they be mutually exclusive? We will revisit these questions in future work. The TTC on a given resources has two major components: the Queue Waitng Time (T_q) and Execution Time (T_x). To a first approximation, we can assume T_x to be a constant, without loss of generality. Admittedly the speed/power of individual processors on resources vary, but it can be assumed that the number of processors (cores) requested can vary so as to keep T_x constant across resources, without any discernible effect on T_q.

There exists a need to characterize the capacity of these resources for the goal of ultimately decreasing the TTC across a set of resources. Most effort has been associated with characterising a single resource and not a collective set of resources. In order to extend the advantages of characterizing a single resource, we believe a characterization of XSEDE [7] as a whole is a useful venture, i.e., capacity characterization of XSEDE would be useful due to the general user having access to multiple resources.

In this paper, characterizations of load and available/instantaneous capacity of resources is examined through various measures from a variety of resources. Representative examples of the different measures of a resource include, utilization and T_q. We do this by measuring the load of the resources and TTC of a measurable workload easily suitable for a heterogeneous environment. These measures have multiple measures that will be examined: historical data obtained through XDMoD [8], instantaneous data obtained at time of submission, and predictions through current prediction tools such as Karnak [3].

This paper examines four specific questions: (i) how accurate are current prediction tools?, (ii) what is the nature of the correlation between a measure of load/utilization and availability/T_q, (iii) of the different measures of load and availability which are better suited for the purposes at hand, viz., do instantaneous workloads better correlate than historical measures in optimizing individual capacity, and (iv) what is the time scale over which a user should/can/is able to plan, i.e., does there exist a time scale after which any of the measures of load/utilization is no longer a reliable indication of the availability of a resource, namely that the use of that measure/information is no better than random?!. Al-

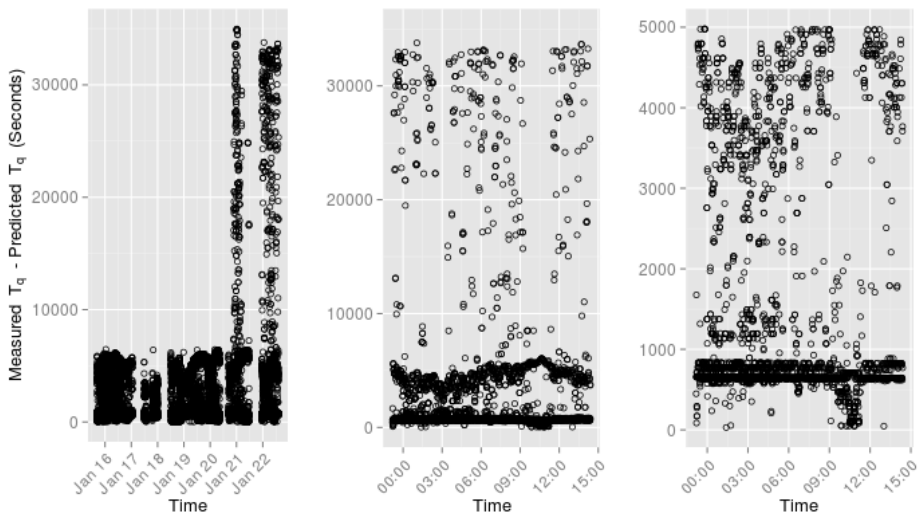

Figure 1: Difference between measured T_q and predicted T_q by Karnak on Trestles. From the left, the first plot represents the total amount of data collected throughout the experiment. The second plot represents a snapshot of the data on January 22, 2014 to show more fine grained behavior. The third plot represents a snapshot of the data where T_q is less than 5000 seconds to show further finer grain behavior within the majority of clustered data.

though these questions will not give us a theoretical framework to determine an optimal distribution/decomposition of the workloads over XSEDE, together, insight gained from examining these four questions will enhance our understanding of the larger issue of collective capacity of XSEDE from an individual scientists point of view.

This is just a snapshot/checkpoint of full work. We want to share excitement and plans of future investigation. We've only characterized (partially) the system. We do not close the loop and answer thh question we set out to answer (viz. how to use collectively) but that will appear in future work.

The remainder of this paper is organized as follows: Section 2 motivates the problem investigated in this paper. Here we provide a short overview of XSEDE as a whole and discuss characteristics of some XSEDE resources relevant in the context of this paper. Next we introduce Karnak Prediction Service and XDMoD. In Section 3, we explain how we used XDMoD and Karnak to develop a better understanding of the problem we are addressing. In addition to that, we provided a set of metrics used to facilitate our research. We also present experiments conducted as a part of this effort. In Section 4, we analyze composed graphs and discuss how these helped to answer questions posed in Section 1. In Section 5 we provide conclusions made as a result of our investigations. In Section 6, we discuss possible directions for future work.

2. BACKGROUND

Resources that comprise XSEDE are both seen and utilized as individual resources. The average user in the most common case chooses an XSEDE resource on which to execute based upon familiarity, availability of software tools, experience and comfort, as opposed to determining the optimal resource amongst a set of XSEDE resources. This is a sub-optimal decision for the end-user, but also leads to inefficiency in resource utilization, as some XSEDE resources are over-subscribed (and hence longer waiting times) whilst some are under-subscribed (and shorter waiting times). Also, we currently see utilization across XSEDE resources is not balanced, which may be addressed by users making optimal decisions.

In addition, through this process, we believe that TTC for the average user can be reduced by choosing an optimal set of resources on which users can execute their workloads. As a result, there exists a necessity to determine which resources characteristics and metrics can be used to enable informed decisions to improve utilization and to reduce TTC of workloads.

Karnak, presented in [9], is a service that provides T_q estimates and resource information. This service is currently available on XSEDE resources for which it provides information that can be used by user-defined heuristic methods to decide where to run a given job. The Karnak service can be accessed via web [3], command line client [3] or via XSEDE portal [10]. Prediction information provided by Karnak can be divided into two types: (i) predictions for the jobs that have already been submitted and (ii) predictions for the jobs that may be submitted. For both types, prediction is either a T_q duration or a time when the job will start execution (timestamp). T_q for jobs that have not yet been submitted are calculated based on resource name, queue name, number of cores, wall time requested and a confidence interval. The X% confidence interval for a prediction p is defined as a duration d, such that X% of the time, the actual value will fall within the interval [p - d, p + d] [9]. The underlying prediction algorithm used by the Karnak utilizes instance-based learning techniques [11].

In [9], it was demonstrated that Karnak overestimates/underestimates it's predictions in the range from 66% to 43% for the following resources - Ranger [12], Lonestar [13], Abe, Cobalt and Pople. The latter three from this list are old clusters located at the National Center for Supercomputing and Applications (NCSA) [14]. Currently, Karnak does not provide functionality to give a T_q estimates for all tasks or even a sub-set of tasks in a complete workflow. In addition, Karnak predictions do not take into accout user profiles: for a given job, two users with different user accounts, allocations, and number of previously submitted jobs will receive identical T_q estimates.

Due to the lack of inclusion of Trestles and Stampede in Reference [9], we measured the accuracy of Karnak on these resources. We measured T_q for a given job configuration and then obtained the estimated T_q provided by Karnak for the same configuration. We have graphically represented the time difference between the measured values of T_q and predicted values of T_q provided by Karnak. Results are shown in Figure 1. Large variation occurred at any given moment in time and the average difference between the Karnak prediction and the actual value measured was 4313.649 seconds with a standard deviation of 7554.752 seconds. We also found that prediction errors ranged to beyond 8 hours. It should be noted that the statistics provided take into consideration all jobs independent of job configuration. Based on these results, we found that for Trestles and Stampede, Karnak did not provide the insight required to distinguish execution resources.

We obtain our historical information for XSEDE through XDMoD [15]. XDMoD (XSEDE Metrics on Demand) is an NSF-funded open source tool designed to audit and facilitate the utilization of the XSEDE cyberinfrastructure by providing a wide range of metrics on XSEDE resources, including resource utilization, resource performance, and impact on scholarship and research. The XDMoD framework is designed to meet the following objectives: (1) provide the user community with a tool to manage their allocations and optimize their resource utilization, (2) provide operational staff with the ability to monitor and tune resource performance, (3) provide management with a tool to monitor utilization, user base, and performance of resources, and (4) provide metrics to help measure scientific impact. While initially focused on the XSEDE program, Open XDMoD has been created to be adaptable to any HPC environment [15].

3. EXPERIMENTS AND METHODOLOGY

We seek to understand resources both individually and collectively. In order to gain a better understanding of resources, various measures of load and availability are examined. These measures give insight into the behavior of a resource in the past, present, and future with the use of various metrics representative of a behavior. The various metrics are represented by data collected through XDMoD, instantaneous data collection, and Karnak, representing past, present, and ¨future(̈probabilistic) metrics respectively. The specific metrics that are used in this paper are: utilization and T_q defined as, the time a job spends in a resource queue before it begins executing. We define utilization as:

$$U = \frac{h_c}{h_p} \tag{1}$$

where h_c is the total number of CPU hours consumed by XSEDE jobs over a given time period and h_p is the total CPU hours that the resource could have potentially provided during that period.

The following is the global experimental configuration: (i) We used Trestles and Stampede as the resources on which we executed our workloads. We considered these two resources for the availability of the prediction tool Karnak and the relatively high load factor (the ratio of number of cores in use and the number of cores available). (ii) The workload that we utilized was a single task executing ̈/bin/sleep 5ẅith a core count (N) and a requested time (M) seconds. N for a given run is randomly chosen from [256, 512, 768, 1024] and M is randomly chosen from [5, 10, 30, 60, 720, 1440]. (iii) The metrics that we measure are T_q, utilization, and predicted T_q. Throughout the experimentation, no more than two jobs are queued at any given moment on a resource in order to prevent jobs from self-conflicting and there by biasing our results.

We utilized the global configuration in the following experiments: (i) We measured T_q, utilization, and Karnak predictions to determine if Karnak provided an accurate resource decision. We submitted jobs to our subset of resources and determined whether our decision for resource collection based on the lowest T_q provided by Karnak was correct. We also chose a resource randomly for each job submission and compared the results. Jobs of identical configuration (core count and requested time) were submitted to two resources. (ii) We did an auto-correlation analysis with T_q and utilization. The auto-correlations were done individually for T_q and utilization to determine how quickly each metric de-correlated with itself. This was done to show whether there existed a consistency in T_q and utilization over a time period and to understand the temporal effects of measures of a resource. (iii) We did a cross correlation analysis for T_q and utilization. The cross-correlation was done between the metrics to determine how each of these metrics behave with respect to each other in order to understand the dependence that utilization may have on T_q.

Historical data obtained from XDMoD was also examined. We transformed auto-correlation and cross-correlation analysis for the same metrics for which measured data was obtained, T_q and utilization. The analysis was expanded to historical data to understand whether there existed a deviation in behavior in T_q and utilization from our measured data. Due to the limitation of the data received from the utility, the core counts and request time for the jobs represented in the data encompassed all job configurations, not only the subset experimentally obtained. It should also be noted that the historical data encompassed aggregates for all users on XSEDE as compared to our individually measured data.

We define auto-correlation as the following:

$$r_k = \frac{\sum_{i=1}^{N-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \tag{2}$$

where: $\bar{x}$ is the mean of the observable being investigated,

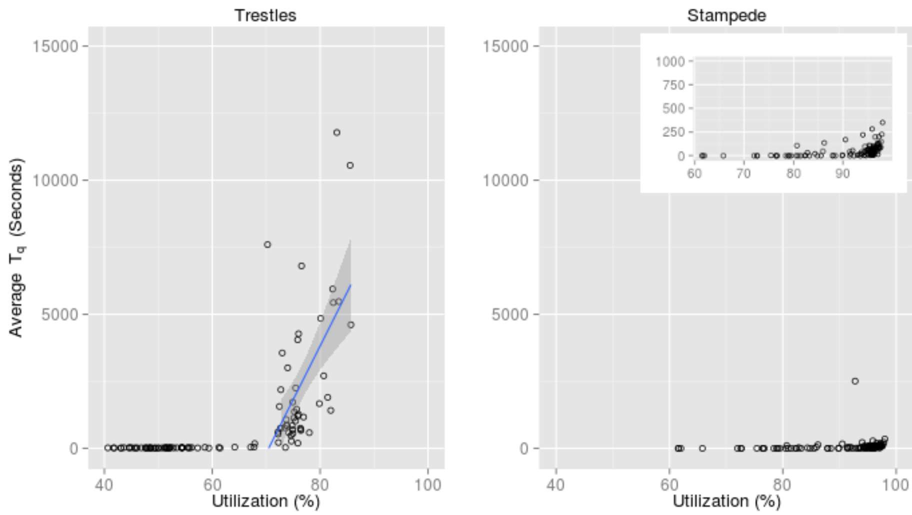

Figure 2: Scatter plot of measured utilization and average T_q for data granularity of an hour. The plot on the left shows data for Trestles and the plot on the right shows data for Stampede. The data represented is the raw data binned to obtain a granularity of an hour. The linear regression on the plot on the left is for a utilization greater than 70% and is represented as the line with a confidence interval of 90% shown as the shaded area around the linear regression. On the plot on the right a the data for an average T_q up to 1000 seconds is shown on the top right to show finer grain behavior.

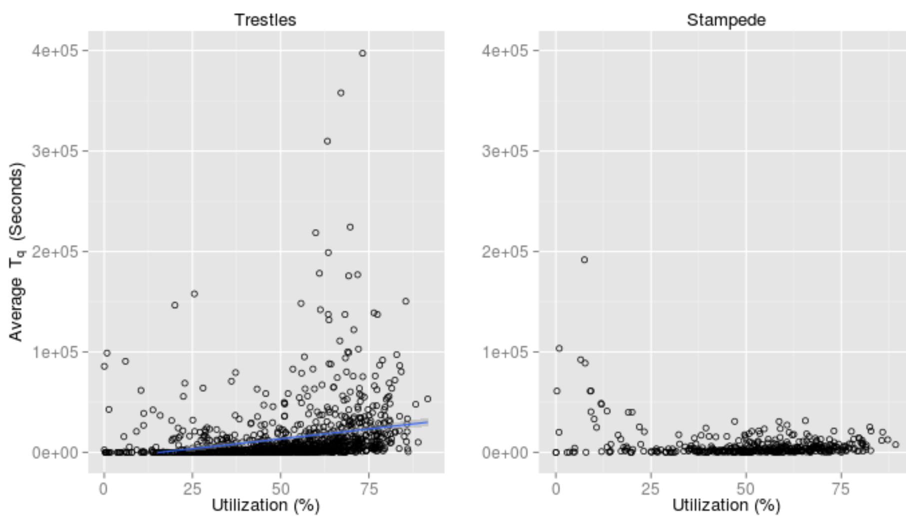

Figure 3: Scatter plot of historical utilization and average T_q for a data point granularity of a day. The plot on the left shows data for Trestles over a span of 3 years and the plot on the right shows data for Stampede over a span of 1 year. The data represented includes jobs of all sizes due to the data not being separable by configurations. The data was obtained from XDMoD with a granularity of one day for each data point. A linear regression is shown as the line with a confidence interval of 90% shown as the shaded area around the linear regression.

x_i is the value of x at instance i, and k is the time lag

We define cross-correlation as the following:

$$r_k = \frac{\sum_{i=1}^{N-k} (x_i - \bar{x})(y_{i+k} - \bar{y})}{\sum_{i=1}^{N} (x_i - \bar{x}) \sum_{i=1}^{N} (y_i - \bar{y})} \tag{3}$$

where, $\bar{x}$ is the mean of the first observable, $\bar{y}$ is the mean of the second observable, x_i is the value of x at instance i, y_i is the value of y at instance i, and k is the time lag.

4. ANALYSIS

In addition to Karnak used as an estimate for T_q in Reference [9], we used Karnak in another capacity where it was used to determine the optimal resource on which to run our workload defined in Section 3. We submitted the workload to two resources, then obtained the Karnak prediction for T_q for the configuration of the workloads on each resource and measured the T_q experienced on the queues of the re-

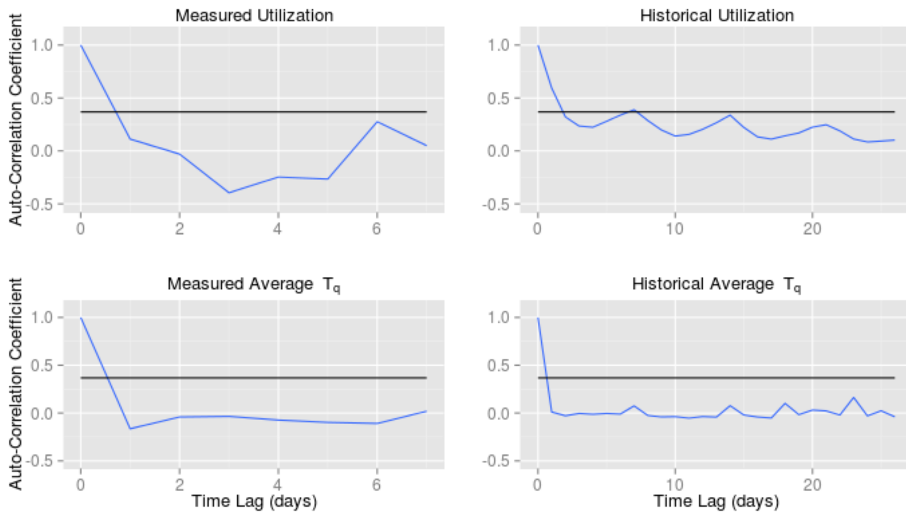

Figure 4: Auto-correlation for measured and historical utilization and T_q on Stampede. The horizontal axis represents the lag introduced to show how each time series correlates with itself in time. The vertical axis is the correlation coefficient defined in (2). Due to the limited data measured the scales on the horizontal axes is not identical between the measured and historical plots. The horizontal line represents the expression 1/e.

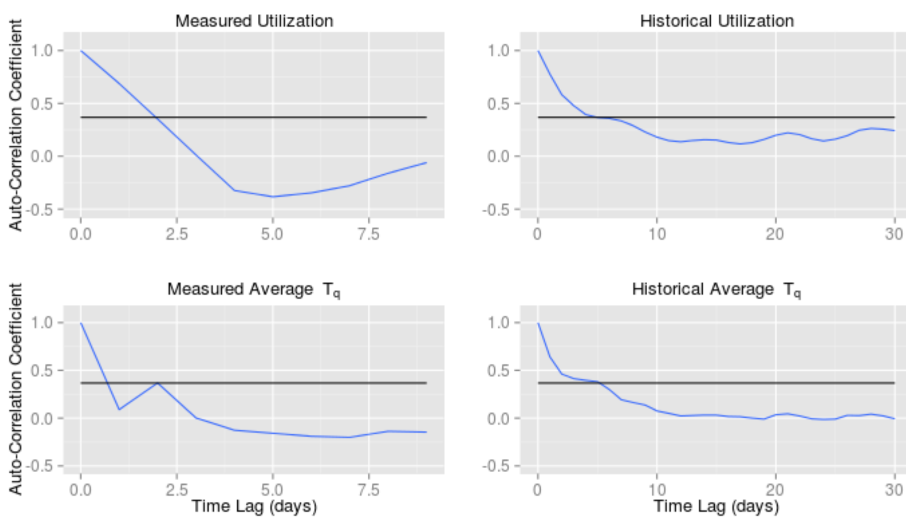

Figure 5: Auto-correlation for measured and historical utilization and T_q on Trestles. The horizontal axis represents the lag introduced to show how the each time series correlates with itself in time. The vertical axis is the correlation coefficient defined in (2). Due to the limited data measured the scales on the horizontal axes is not identical between the measured and historical plots. The horizontal line represents the expression 1/e.

sources. We also made a random guess as to which resource would have a lower T_q and we compared the result to that determined by the Karnak predictions. We found that on average a random guess was correct 48% of the time, where by definition a random choice should have a symmetric distribution around 50%. For any given set of 100 trials, the averages for a random choice ranged from 32% to 70%. Karnak on the other hand produced a result of 40% with a range of 16% to 90% for any given set of 100 trials. The results from this experiment suggest that Karnak as a capability does not provided the accuracy to make a well informed decision.

Driven by the difficulty of prediction, we examined different measures of load and availability. We first measured utilization and queue waiting times and examined the relationship between the two parameters on Trestles and Stampede. We found that for utilization less than 70%, there was very little variation in queue waiting times. Variation occurred for higher utilization seen in Figure 2. We see that there was a general linear positive trend that exists for utilization greater than 70%. on Trestles. The same trend exists for Stampede; however, the trend is weak.

For both Trestles and Stampede, the same experimental configurations are used and the core counts that we use range from 128 to 1024 explained in Section 4. Stampede is larger than Trestles, i.e., has more cores [16]. Due to the differ-

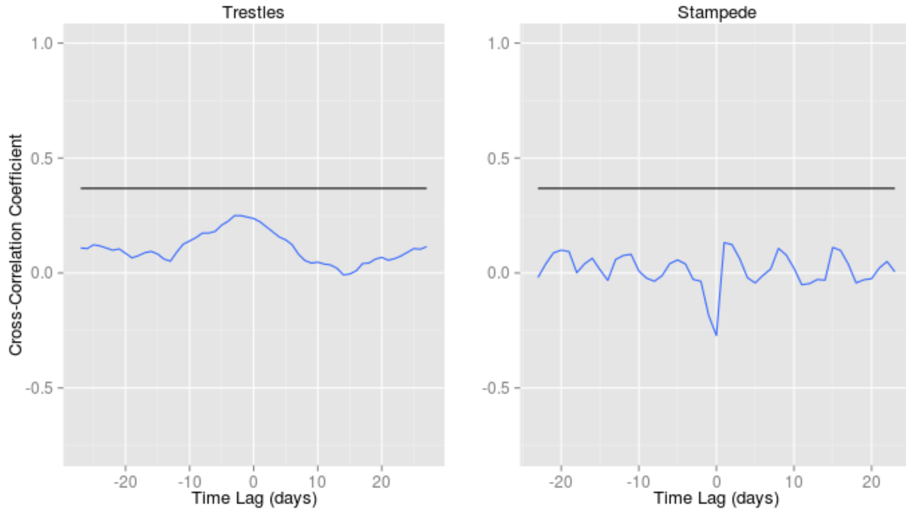

Figure 6: Cross-correlation between historical utilization and T_q. The horizontal axis represents the lag in the data set for queue waiting times introduced to show how utilization correlates with queue waiting times in the past and the future. The vertical axis represents the cross correlation coefficient as defined in (3). The horizontal line represents the expression 1/e.

ence in total core count for both resources, T_q had very little variation and was relatively close to 0 if, (i) utilization was low seen on Trestles (ii) the size of the job requested was insignificant compared to the total core count of the resource seen on Stampede. This can be seen in Figure 2.

Utilization and T_q held the same trends regardless of the granularity and measure. This can be seen in the representation in Figure 3. It should be noted that the anomalies of very large T_q for low value of utilization are attributed to jobs being submitted to the resource before it is taken down for maintenance, technical difficulties, etc., and beginning after the resource is brought online.

We examined the behavior of T_q and utilization by removing the time dimension in Figure 2 and Figure 3, which plays a role in measures of load and availability. In response, the behavior of the metrics were examined with the dimension of time by the use of an auto-correlation. An auto-correlation shows the degree of similarity between observations in a time series as a function of a time lag. In the auto-correlation analysis, we defined decorrelation as the point at which the correlation coefficient passes 1/e, where e is Euler's number estimated to be 2.718. This numerical interpretation is generally taken as the point at which a time series being examined is no longer statistically consistent, or the point at which a time series has decorrelated.

We found that after roughly a span of 1 day, both utilization and T_q decorrelate with respect to themselves. We saw that this was consistent regardless of the resource, shown in Figure 4 and Figure 5. From both the measured and historical measures, we found that the behavior of the autocorrelation was very similar. There do exist finer grained details within each distribution; however, the overall behavior was invariant across resources. Based on the results, T_q decorrelates more quickly than utilization. Taking into consideration that T_q is the dependent variable, this is consistent with the understanding that utilization changing in behavior should cause a faster change in behavior for T_q. Due to the limitation of the granularity of the data from the historical point of view (one day), we cannot provide a more accurate or precise analysis as there may be a finer grained behavior that may be present for finer granularities of data.

Understanding the behavior of both utilization and T_q within their respective time series, we examined the behavior of each with respect to each other through a cross-correlation. We found that at best there exists weak correlation between utilization and T_q due to the correlation coefficient for a time lag of 0 falling below the statistical threshold of 1/e shown in Figure 6. As expected, the correlation between utilization and T_q decreases as T_q is offset by a time lag.

5. CONCLUSIONS

The experiments were performed to determine öptimalřesource selection methods using historical, instantaneous, and prediction based measures. It is worth reiterating that Karnak-based predictions were not effective in selecting optimal resources, thus indicating the need for non-prediction based approaches to resource selection. We investigated different measures of load and availability, such as utilization and T_q. Through this we have determined correlation between different measures as well as auto-correlation between the individual measures. Consistent with expectation, we found that when the load on resources was low, e.g., for utilization less than 70%, there was very little variation in queue waiting times. However, for the measures of load and availability we examined, we found that there exists a weak correlation, providing an approximate time-scale over which they are correlated. Our results indicate that counter to simple assumptions, a measure of availability such as T_q is only weakly correlated to a measure of load (such as utilization). However, further work is required to determine precise temporal relationship between the two measures. This paper represents preliminary work towards the semi-empirical modeling of capacity-based resource selection of multiple (distributed) resources such as those found on XSEDE. In the absence of formal methods, we believe semi-empirical models are likely to provide an optimal point between accuracy, simplicity, and error (uncertainty) analysis.

6. FUTURE WORK

Although we began with a seat-of-the-pants analysis of the difficulty of prediction, we do believe there might still be a role for prediction services, but just not as we/XSEDE are currently using them. For example, we believe that in lieu of individual jobs, predicting the T_q for an ensemble/aggregation/collection of jobs is likely to be more reliable, i.e., a statistical approach for prediction. Not only does this have the obvious property of not being subject to the same levels of randomness (c.f. law of large numbers), there are reasons to believe that the aggregated throughput is a more reliable measure than individual job TTC. We will examine this proposition in future work.

In the future, we will continue to examine the various measures and parameters established in this paper. We will collect more data through measurements experimentally as well as attempt to obtain historical data at a finer granularity. We will also extend the work to more resources in order for generic resource capabilities to be established. In addition, further measures, parameters, and statistical analysis will be established to determine resource capabilities with the ultimate goal of global characterization of resources. [7]

Author Contributions Most of the experiments were performed and analyzed by Vishal Shah, who is an undergraduate majoring in Computer Engineering. Experiments related to Karnak were performed by Antons Trekalis a 1st year graduate student. The bulk of the writing was done by VS with important contribution AT. SJ set the overall motivation of the paper, and helped with the introduction.

Acknowledgement Vishal Shah is funded by Research Experience for Undergraduates (REU) supplement to NSF award ACI-1235085. This work is also funded by NSF OCI-1253644. This work has also been made possible thanks to computer resources provided by XRAC award TG-MCB090174.

7. REFERENCES

[1] "Stampede (stampede.tacc.xsede.org) - a dell linux cluster.." `https://www.xsede.org/web/xup/knowledge-base/-/kb/document/bcib`. Accessed: 2014-03-20.

[2] "Trestles - xsede cluster.." `https://www.sdsc.edu/us/resources/trestles/`. Accessed: 2014-03-20.

[3] "The karnak prediction service.." `http://karnak.teragrid.org/karnak/index.html`. Accessed: 2014-03-20.

[4] J. Brevik, D. Nurmi, and R. Wolski, "Predicting bounds on queuing delay for batch-scheduled parallel machines," in *Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming*, pp. 110–118, ACM, 2006.

[5] H. Li, D. Groep, J. Templon, and L. Wolters, "Predicting job start times on clusters," in *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*, pp. 301–308, IEEE, 2004.

[6] O. Sonmez, N. Yigitbasi, A. Iosup, and D. Epema, "Trace-based evaluation of job runtime and queue wait time predictions in grids," in *Proceedings of the 18th ACM international symposium on High performance distributed computing*, pp. 111–120, ACM, 2009.

[7] "Extreme science and engineering discovery environment - XSEDE." `https://www.xsede.org/`. Accessed: 2014-03-20.

[8] "Xsede metrics on demand - XDMoD." `https://xdmod.ccr.buffalo.edu/`. Accessed: 2014-03-20.

[9] W. Smith, "A service for queue prediction and job statistics," in *Gateway Computing Environments Workshop (GCE), 2010*, pp. 1–8, IEEE, 2010.

[10] "The karnak prediction service on xsede.." `https://www.xsede.org/web/xup/queue-prediction`. Accessed: 2014-03-20.

[11] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, 1999.

[12] "Ranger - sun constellation linux cluster.." `https://www.tacc.utexas.edu/resources/hpc/ranger`. Accessed: 2014-03-20.

[13] "Lonestar - dell linux cluster.." `https://www.tacc.utexas.edu/resources/hpc/lonestar`. Accessed: 2014-03-20.

[14] "National center for supercomputing and applications.." `http://www.ncsa.illinois.edu/`. Accessed: 2014-03-20.

[15] "Xsede metrics on demand - XDMoD." `http://xdmod.sourceforge.net/`. Accessed: 2014-03-20.

[16] "XSEDE resource overview." `https://www.xsede.org/resources/overview`. Accessed: 2014-03-20.