Characterizing Task Usage Shapes in Google's Compute Clusters

Qi Zhang
University of Waterloo
q8zhang@uwaterloo.ca

Joseph L. Hellerstein
Google Inc.
jlh@google.com

Raouf Boutaba
University of Waterloo
rboutaba@uwaterloo.ca

ABSTRACT

The increase in scale and complexity of large compute clusters motivates a need for representative workload benchmarks to evaluate the performance impact of system changes, so as to assist in designing better scheduling algorithms and in carrying out management activities. To achieve this goal, it is necessary to construct workload characterizations from which realistic performance benchmarks can be created. In this paper, we focus on characterizing run-time task resource usage for CPU, memory and disk. The goal is to find an accurate characterization that can faithfully reproduce the performance of historical workload traces in terms of key performance metrics, such as task wait time and machine resource utilization. Through experiments using workload traces from Google production clusters, we find that simply using the mean of task usage can generate synthetic workload traces that accurately reproduce resource utilizations and task waiting time. This seemingly surprising result can be justified by the fact that resource usage for CPU, memory and disk are relatively stable over time for the majority of the tasks. Our work not only presents a simple technique for constructing realistic workload benchmarks, but also provides insights into understanding workload performance in production compute clusters.

1. INTRODUCTION

Cloud computing promises to deliver highly scalable, reliable and cost-efficient platforms for hosting enterprise applications and services. However, the rapid increase in scale, diversity and sophistication of cloud-based applications and infrastructures in recent years has also brought considerable management complexities. Google's cloud backend consists of hundreds of compute clusters, each of which contains thousands of machines that host hundreds of thousands of tasks, delivering a multitude of services including web search, web hosting, video streaming, as well as data intensive applications such as web crawling and data mining. Supporting such a large-scale and diverse workload is a challenging goal, as it requires a careful understanding of application performance requirements and resource consumption characteristics.

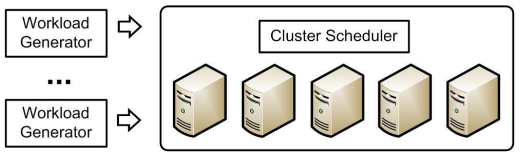


Figure 1: A Compute Cluster Benchmark

Traditionally, Google relies on performance benchmarks of compute clusters to quantify the effect of system changes, such as the introduction of new task scheduling algorithms, capacity upgrading, and change in application source code. As shown in Figure 1, a *performance benchmark* consists of one or more workload generators that generate synthetic tasks scheduled on serving machines. In all of the aforementioned scenarios, using historical workload traces can accurately determine the impact of changes to minimize the risk of performance regressions. However, this approach does not allow answering *what-if* questions about scaling workload or other scenarios that have not been observed previously.

To address this limitation, it is necessary to develop workload characterization models. We use the term *task usage shape* as a statistical model that describes run-time task resource consumption (CPU, memory, disk, etc.). Our goal is develop an accurate characterization of task usage shapes that is *sufficiently accurate* for producing synthetic workload benchmarks. The key performance metrics we are interested in are the average *task wait time* and *machine resource utilization* for CPU, memory and disk in each cluster. Task wait time is important because it is a common concern of cloud users. As the workload typically contains many long-running batch tasks that may alternate between waiting (this also includes the case of rescheduling due to preemption or machine failure) and running state, the total wait time experienced by each task is a main objective to be minimized. Similarly, machine resource utilization is important as it is a common objective of cloud operators to maintain high resource utilization.

In this paper, we present a characterization of task usage shape that accurately reproduces performance characteristics of historical traces, in terms of average task wait time and machine resource utilization. Through experiments using real workload traces from Google production clusters, we find that simply modeling the task mean usage can achieve

Compute Cluster		CPU (Cores)		Memory (GB)		Disk (GB)	
		Mean	Avg. cv	Mean	Avg. cv	Mean	Avg. cv
A	Type 1	0.25	0.3985	0.83	0.3576	1.69	0.3915
	Type 2	0.02	0.4755	0.06	0.446	0.12	0.4432
	Type 3	0.21	0.9143	0.79	0.6825	1.65	0.8225
	Type 4	0.16	1.1765	0.1	0.763	0.09	1.1585
B	Type 1	0.09	0.5922	0.55	0.845	1.62	0.5495
	Type 2	0.01	1.2285	0.05	1.0133	0.15	0.667
	Type 3	0.03	0.89	0.17	0.495	0.09	0.32385
	Type 4	0.22	1.076	0.11	1.0265	0.22	0.675
C	Type 1	0.14	0.3415	0.9	1.14	2.66	0.2195
	Type 2	0.38	1.4993	0.32	0.1325	2.31	0.6755
	Type 3	0.21	0.9325	0.15	0.7177	0.33	0.6015
	Type 4	0.1	1.2205	0.07	1.033	0.05	0.4205
D	Type 1	0.23	0.59	1.05	1.025	2.83	0.5475
	Type 2	0.04	0.8057	0.32	0.6265	0.11	0.8245
	Type 3	0.52	1.107	0.3	0.946	0.34	0.986
	Type 4	0.1	1.592	0.09	0.903	0.09	1.6625
E	Type 1	0.13	0.768	1.35	0.742	1	0.207
	Type 2	0	3.5888	0.01	0.1557	0	0.211
	Type 3	0.16	0.9128	4.58	0.484	0.3	0.5085
	Type 4	0.08	1.164	0.05	0.7995	0.03	0.4065
F	Type 1	0.36	0.5828	1.14	0.4005	2.58	0.218
	Type 2	0.38	1.0349	1.21	1.1935	0.08	0.258
	Type 3	0.22	0.54	0.15	0.595	0.32	0.8295
	Type 4	0.07	0.9976	0.18	0.848	0.14	0.4103

Table 1: Data set used in the experiment

high accuracy in terms of reproducing resource utilization and task wait time in Google's compute clusters. While this result may seem surprising at first glance, a closer examination shows that it is due to both (1) the low variability of task resource usage in the workload, and (2) the characteristics of evaluation metrics (i.e. task wait time and machine resource utilization) under different workload conditions. Our work not only presents a simple technique for generating workload traces that closely resemble real workload traces in terms of the key performance metrics, but also provides helpful insights into understanding workload performance in production compute clusters.

The rest of the paper is organized as follows: Section 2 describes the historical traces we used during our analysis. The experimental results are reported in Section 3. Section 4 is devoted to the discussion of the evaluation result. Specifically we analyze the correlation between the theoretical model errors (i.e. variability in task usage) and the empirical model errors observed in the simulations. Section 5 surveys related work in this area. Finally, section 6 concludes the paper.

2. DATASET DESCRIPTION

The data set we used in our study consists of historical traces of 6 compute clusters spanning 5 days (June 21 - 25, 2010). Together our analysis uses a total of 30-cluster days of traces from the production clusters. These historical traces contain CPU, memory and disk usage of every task scheduled in each cluster sampled at 5-minute intervals. Generally speaking, the workload running on Google compute clusters can be divided into 4 *task types*. Type 1 tasks correspond to production tasks that process end-user requests, whereas type 4 tasks correspond to low priority, non-production tasks that do not directly interact with users. Type 2 and Type 3 represent tasks that have characteristics falling between type 1 and 4. Table 2 summarizes the size of each cluster as well as the workload composition in terms of the 4 task types. We purposely select clusters of sizes ranging over two orders of magnitude. Typically tasks of type 4 have the highest task population, while tasks of type 1 have the lowest. There are exceptional cases, such as cluster F, which has a large percentage of Type 3 tasks.

Table 1 summarizes the mean and average coefficient of variation (CV) for CPU, memory and disk usage for tasks in every cluster over the course of 5 days. the task CV of a particular resource is computed by dividing the standard deviation of the measured usage values by their mean. From Table 1, it can be seen that CPU and disk have the highest and lowest CVs, respectively. Even though in many cases the average CV can exceed 1, it does not imply high resource usage variability since CV is generally sensitive to small mean value. For example, even though tasks of Type 2 in compute cluster E have the highest CV for CPU (i.e. 3.5888), the average CPU usage is very close to 0, hence the variability in resource usage is small. Similar results have also been reported in [9] and [11]. Hence we can conclude that the run-time variability of task resource usage is low.

The analysis above suggests that simply modeling the mean values of run-time tasks resource consumption is a promising way to model task usage shapes. As a starting point, we call this characterization model the *mean usage model* of tasks usage shapes. Specifically, the mean usage model stores the mean usage of CPU, memory and disk and running time of each task in the workload. Our hypothesis is that the mean usage model can perform reasonably well for reproducing the performance of real workload.

3. EXPERIMENTS

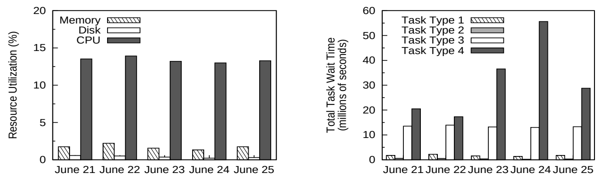
This section presents our experiment results. We first describe our evaluation methodology. Given a historical workload trace from real compute clusters, We modify the trace by over-writing the actual task resource usage by the model-predicted usage values. Specifically, to evaluate the mean usage model, we need to replace measured resource usage records by their mean value for each task and each resource type. The other components of the workload, including user-specified resource requirements, task placement constraints [10] and request arrival times, are kept intact. We then run two experiments. The first one runs the benchmark using the unmodified historical trace. The second one runs the benchmark using the modified trace after the treatment. Once finished, we compare the benchmark results of both experiments. As mentioned previously, two performance metrics of interests are task wait time and machine resource utilization.

In addition, during our experiments we realized that it is necessary to increase the load on individual clusters in order to make the difference more apparent. For example, when there is ample free capacity in a cluster, every task can almost immediately be scheduled and never have to wait during its course of execution. In this case, the task wait time will be low regardless of the quality of the characterization. Hence, we developed a *stress generator* that increases the load on the cluster by randomly removing a fraction of its machines. We will discuss the effect of load increase on the performance metrics in Section 4.

We conducted trace-driven simulation for all 30 cluster-days. We first report the basic characteristics of our performance metrics. Specifically, Figure 2 shows the total task wait time and resource utilization for cluster A across 5 days. It can be observed that the day-to-day variability for resource utilization is rather small. On the other hand, the day-to-day variability for task wait time can be quite high, especially for the tasks of type 4, where total task wait time

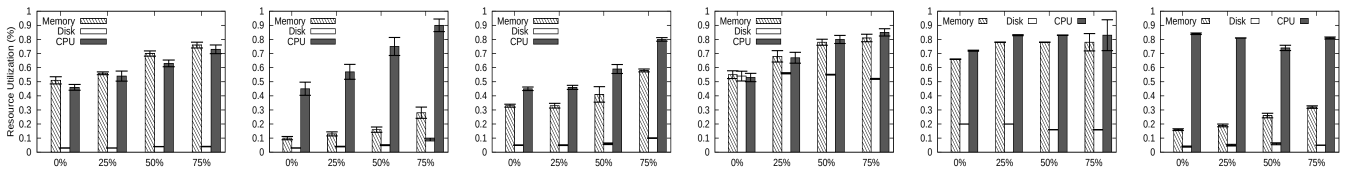
Compute Cluster	No. of machines	Type 1 (%)	Type 2 (%)	Type 3 (%)	Type 4 (%)
A	10000s	3.12	0.26	3.14	93.47
B	1000s	1.46	0.86	2.52	95.16
C	1000s	4.54	0.34	4.67	90.45
D	1000s	5.86	2.42	31.77	59.95
E	1000s	39.26	1.48	34.27	24.99
F	10s	1.23	0.2	72.93	25.64

Table 2: Cluster Size and Workload Composition



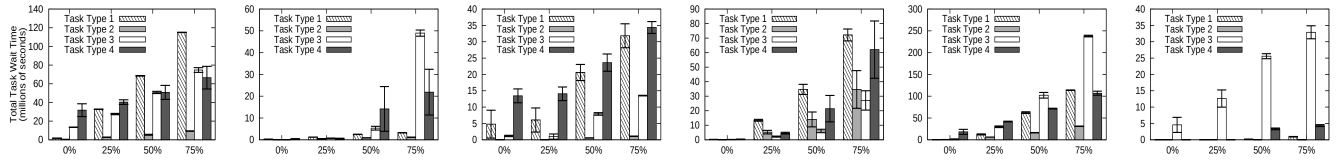
(a) Resource utilization (b) Task wait time

Figure 2: Day-to-Day variability of Two Metrics for Cluster A



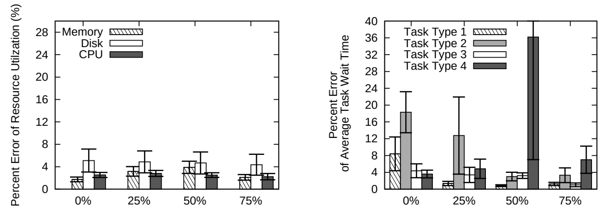
(a) Cluster A (b) Cluster B (c) Cluster C (d) Cluster D (e) Cluster E (f) Cluster F

Figure 3: Average Machine Resource Utilization over 5 Days after removing 0%, 25%, 50% and 75% of the machines



(a) Cluster A (b) Cluster B (c) Cluster C (d) Cluster D (e) Cluster E (f) Cluster F

Figure 4: Average Task Wait Time over 5 Days after removing 0%, 25%, 50% and 75% of the machines



(a) Machine Resource utilization (b) Task Wait Time

Figure 7: Summary of the Percent Model Error of Performance Metrics for the Mean Usage Model

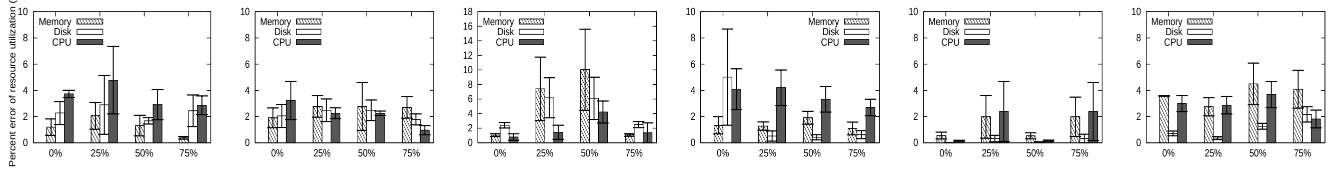
on June 22 is 2 times larger than the one on June 25. These observations are consistent across clusters, which suggests that resource utilization is a more robust metric than total task wait time. The average machine resource utilization and task wait time for all 6 clusters under 4 different utilization levels are shown in Figure 3 and Figure 4 respectively. As expected, both the utilization and total task wait time grow with the utilization level (i.e. the percentage of machines removed). The task wait time seems to grow rapidly at high utilization level. More analysis on this observation will be described in Section 4.

Next we present our evaluation of the mean usage model. The results for resource utilization and task wait time are shown in Figure 5 and 6, respectively. It can be observed that the model error for resource utilization is quite small ($\leq$ 10%) under all circumstances. However, for task wait time, the percent error has very high variability. For example, Cluster D produces a significant error for tasks of type 4 when number of machines removed is 50%. However, the large error bar (representing the standard error) indicates that the error is likely caused by one or 2 samples. This is also explained by our previous result that task wait time is a less robust metric compared to resource utilization.

The average performance of machine resource utilization and task wait time across all 6 clusters are summarized in Figure 7. The model error for machine resource utilization seems to be uniformly low under all 4 utilization levels. On the other hand, despite the large variation in results, the model errors of task wait time seem to follow decreasing trends for task type 1 and 2 and increasing trends for task type 4. As type 4 tasks typically have the largest population in the workload, It is reasonable to say that the task wait time seems to increase with machine resource utilization. Overall, these observations suggest that the mean usage model performs well for reproducing the performance of real workload in terms of task wait time and resource utilization.

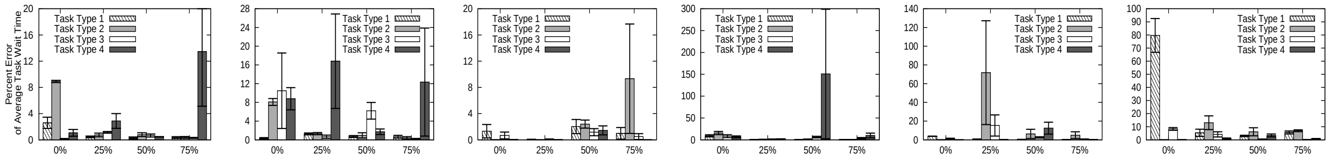
4. DISCUSSION

The experiment results described in Section 3 suggest that the mean usage model performs well in terms of reproducing the average task wait time and machine resource utilization. It seems intuitive to explain why machine resource utilization performs well, as most of tasks have low resource usage variability for all 3 resource types. However, it is the fact



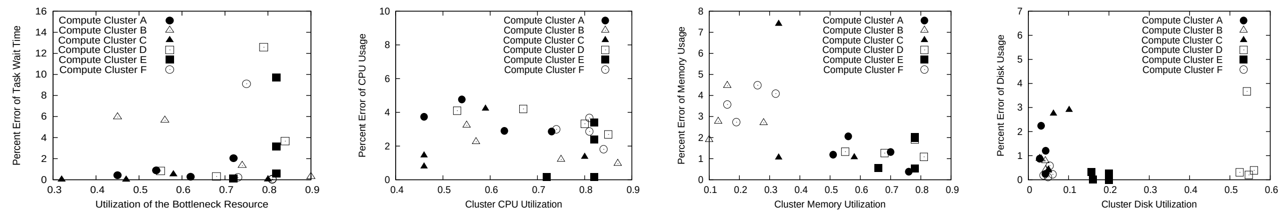
(a) Cluster A (b) Cluster B (c) Cluster C (d) Cluster D (e) Cluster E (f) Cluster F

Figure 5: Percent Model Error for Resource Utilization after removing 0%, 25%, 50% and 75% of the machines



(a) Cluster A (b) Cluster B (c) Cluster C (d) Cluster D (e) Cluster E (f) Cluster F

Figure 6: Percent Model Error for Task Wait Time after removing 0%, 25%, 50% and 75% of the machines



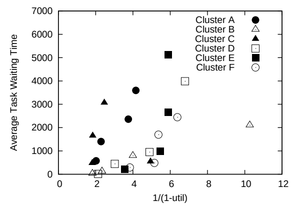
(a) Task Wait Time (b) CPU Utilization (c) Memory Utilization (d) Disk Utilization

Figure 8: Percent Model Error of Performance Metrics vs. Machine Resource Utilization

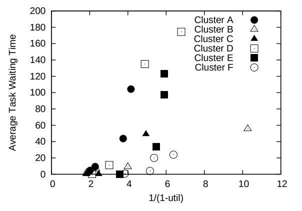
that the mean usage model also accurately reproduce task wait time makes the result surprising. It should be pointed out that occasionally we may still see errors$\geq$ 10% for task wait time. Hence this section is dedicated to analyzing the model errors for both task wait time and machine resource utilization.

To start our analysis, note that in addition to modifying the task shapes in the treatment process, we have also used a stress generator to introduce additional load in order to make task wait times more apparent. The stress generator increases the utilization of the cluster by randomly removing a percentage of machines from the cluster. To understand the impact of resource usage variability on model errors for both task wait time and machine resource utilization, we must first determine the impact of utilization on the model errors. From the discussion in Section 3, we know that the average task wait time increase with resource utilization due to the large population of type 4 tasks. For the model errors of machine cluster utilization, our hypothesis was that it should decrease with the utilization level of the cluster, as higher utilization implies less room for model errors. Furthermore, as there are many tasks waiting to be scheduled, the scheduler in this case will try to "bin-pack" tasks on physical machines as much as possible, further reducing the model error. To validate this hypothesis, we plot the model errors of the performance metrics against utilization for all the clusters in Figure 8. However, even though there seems to be a trend that the model errors for machine cluster utilization decrease with utilization level, the trend is not significant enough as the noise in the percent error in both cases can be of equal magnitude. This is mainly because the the model errors for machine cluster utilizations are small (i.e. $\leq 5\%$).

For task wait time, from queuing theory we know that average task wait time ($E(w_i)$) grows hyperbolically with respect to resource utilization ($util$) (i.e. $E(w_i) \propto \frac{1}{1-util}$) for every compute cluster i [7]. Specifically, as $util$ approaches 1, $E(w_i)$ grows towards infinity. To see this, we plot $E(w_i)$ against $\frac{1}{1-util}$ for every cluster $1 \leq i \leq 6$ in Figure 9(a). The diagram clearly indicates this relationship, as the points for each compute cluster roughly lie on a same line. We also plotted the average difference in task wait time $E(\Delta w)$ against $\frac{1}{1-util}$ in Figure 9(b). It turns out that the points for each compute cluster again roughly lie on the same line in Figure 9(b). Denote by r_{w_i} and $r_{\Delta w_i}$ the slope of the lines for each cluster in Figure 9(a) and 9(b) respectively. Our hypothesis is that higher task resource variability will cause higher growth rate difference in task wait time, as difference in scheduling decisions at higher utilization level will have higher impact on task wait time. To validate this hypothesis, we plotted ratio of the two slopes for each cluster (i.e. $r_{\Delta w_i}(i)/r_{w_i}(i)$) against the average CV of the bottleneck resource type (i.e. resource type with the highest utilization as it generally has the largest impact on task schedulability) in Figure 10(a). The average CV is weighted by task duration, as long running tasks have higher impact on the model error than short running tasks. It turned out there is a direct relationship between these two quantities, as shown in Figure 10(a). Another way to interpret this re-

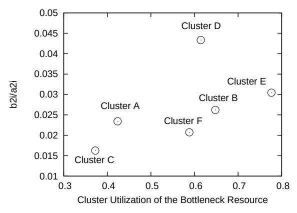


(a) Task Wait Time

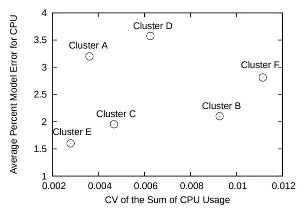


(b) Difference in Task Wait Time

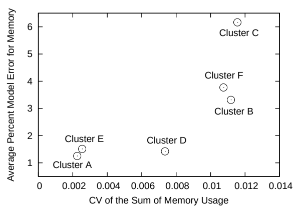
Figure 9: Total and Difference in Task Wait Time vs $\frac{1}{1-util}$ of the bottleneck resource



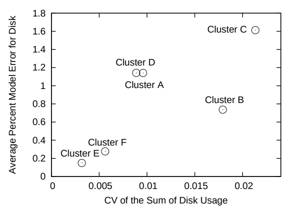
(a) $\frac{b_{2i}}{a_{2i}}$ vs. Avg. weighted task CV



(b) Percent Error for CPU vs. Est. CV of total CPU usage



(c) Percent Error for Memory vs. Est. CV of total Memory Usage



(d) Percent Error for Disk vs. Est. CV of total Disk Usage

Figure 10: Correlating Model Error in Performance Metrics with Variability in Task Usage Shapes

lationship is as follows: we can model $E(w_i) = a_{1i} + \frac{a_{2i}}{1-util}$ and $E(\Delta w_i) = b_{1i} + \frac{b_{2i}}{1-util}$. Thus $E(w_i) = a_{1i} + \frac{a_{2i}}{1-util}$ and $E(\Delta w_i) = b_{1i} + \frac{b_{2i}}{1-util}$. The model error hence can be expressed as $Err = \frac{E(\Delta w_i)}{E(w_i)} = \frac{(1-util)b_{1i}+b_{2i}}{(1-util)a_{1i}+a_{2i}} \approx \frac{b_{2i}}{a_{2i}}$, the ratio of the two slopes. Intuitively, this result means that the task usage variability does cause a difference in task wait time, but the difference is not significant considering the wait time for most of the tasks also grow at a rapid rate.

For machine resource utilization, unlike the case for task wait time, the average model error tends to be quite small (i.e. around 3%), and the impact of utilization on the model error is also quite small. In this case, we can simply compute the average model errors under all utilization levels and plot them against CVs of each cluster. Notice that since the utilization is the sum of resource usage, the CVs we used should be the CV of the sum of the total usage (i.e. CV_{sum}). To estimate this value, assuming the resource usage variability follows a normal distribution, then CV_{sum} can be estimated by summing up the variance (i.e. $(mean_t \cdot CV_t)^2$) of each task t weighted by its duration d_t, divided by the simulation interval. Using the fact that the sum of the variances is the variance of the sum for normal distributions, we can then compute CV_{sum} accordingly. The results are shown in Figure 10(b),(c) and (d). There seems to be a correlation between the resource variability and model error observed in the experiment for Memory and Disk. On the other hand, the correlation for CPU seems less accurate. The reason is that task usage for CPU generally has much higher variability than memory and disk, hence the benchmark is more conservative in computing the resource utilization to account for potential future variability of CPU usage. This leads to the inaccuracy observed in Figure 10(b).

Overall, our analysis shows that ignoring task usage variability at run-time does introduce inaccuracies compared to real historical traces, the difference seems to be small in all the cases. Hence, we believe that mean usage model is sufficiently accurate for reproducing the performance of real workloads.

5. RELATED WORK

There is a long history of research on workload characterization. Specifically, There has been work on characterizing workload in various application domains , such as the Web [2], multimedia [5], distributed file systems [3], databases [12] and scientific computing [1]. Furthermore, different aspects of workload characterization, including arrival patterns [4], resource requirements [8] and network traffic [6] have also been studied. However, the focus of existing work has been on revealing workload characteristics, rather than evaluating the quality of the workload characterization. In contrast, our work focuses on studying the quality of characterizations using performance benchmarks.

Our work is directly related to our previous work on task shape classification [9]. The goal in [9] is to construct a task classification model that divides workload into distinct classes using the K-means clustering algorithm. The features used by the clustering algorithm are the mean cpu usage, mean memory usage and task execution time. The accuracy of the model is evaluated by computing the intra and inter cluster similarity in terms of standard deviation from the mean values of the cluster. However, it is unclear whether the task classification criteria are sufficient for generating synthetic workloads that can reproduce the performance characteristics of real workloads. More recently, Chen *et. al.* [11] analyzed the publicly available traces from Google's clouds and performed K-means on jobs using a variety of features. They also used correlation scores to infer relationships between job types and job clusters. However this is different from our work that focus on task shape characterization.

6. CONCLUSIONS

In this paper we studied the problem of deriving characterization models for task usage shapes in Google's compute cloud. Our goal is to construct workload models that accurately reproduce the performance characteristics of real workloads. To our surprise, we find that simply capturing the mean usage of each task (i.e., the mean usage model) is sufficient for generating synthetic workload that produces low model error for both resource utilization and task wait time. The direct implication of our work is that we can realistically estimate the total wait time and resource utilization for existing or imaginary workloads (e.g. workload scaled up by $\times 10$) using synthetic workload generated from the distribution of task mean usages. Our future work includes using compute cluster benchmarks to find effective clustering algorithms that will produce simpler task shape characterization models with similar performance as the mean usage model.

7. REFERENCES

[1] S. Alarm, R. Barrett, J. Kuehn, P. Roth, and J. Vetter. Characterization of scientific workloads on systems with multi-core processors. In *International Symposium on Workload Characterization*, pages 225–236. IEEE, 2007.

[2] M. Arlitt and C. Williamson. Internet web servers: Workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, 2002.

[3] R. Bodnarchuk and R. Bunt. A synthetic workload model for a distributed system file server. In *Proceedings of the 1991 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, pages 50–59. ACM, 1991.

[4] M. Calzarossa and G. Serazzi. A characterization of the variation in time of workload arrival patterns. *IEEE Transactions on Computers*, pages 156–162, 1985.

[5] M. Chesire, A. Wolman, G. Voelker, and H. Levy. Measurement and analysis of a streaming-media workload. In *Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems-Volume 3*, page 1. USENIX Association, 2001.

[6] D. Ersoz, M. Yousif, and C. Das. Characterizing network traffic in a cluster-based, multi-tier data center. In *International Conference on Distributed Computing Systems*, page 59. IEEE, 2007.

[7] L. Kleinrock and R. Gail. *Queueing systems*, volume 1. Wiley New York, 1975.

[8] A. Maxiaguine, S. Kunzli, and L. Thiele. Workload characterization model for tasks with variable execution demand. 2004.

[9] A. Mishra, J. Hellerstein, W. Cirne, and C. Das. Towards characterizing cloud backend workloads: insights from Google compute clusters. *ACM SIGMETRICS Performance Evaluation Review*, 37(4), 2010.

[10] B. Sharma, V. Chudnovsky, J. Hellerstein, R. Rifaat, and C. Das. Modeling and Synthesizing Task Placement Constraints in Google Compute Clusters. In *Proceedings of ACM Symposium on Cloud Computing (SOCC)*, 2011.

[11] Y. Chen, and A. Ganapathi et. al. Analysis and Lessons from a Publicly Available Google Cluster Trace. *UC Berkeley Technical Report*, 2010.

[12] P. Yu, M. Chen, H. Heiss, and S. Lee. On workload characterization of relational database environments. *IEEE Transactions on Software Engineering*, pages 347–355, 1992.