promoting access to White Rose research papers

Universities of Leeds, Sheffield and York
http://eprints.whiterose.ac.uk/

This is the published version of a Proceedings Paper presented at the **2013 IEEE International Conference on Cloud Engineering**

Garraghan, P, Townend, PM and Xu, J (2013) *An Analysis of the Server Characteristics and Resource Utilization in Google Cloud.* In: IC2E '13 Proceedings of the 2013 IEEE International Conference on Cloud Engineering. UNSPECIFIED. IEEE , 124 - 131. ISBN 978-0-7695-4945-3

White Rose Research Online URL for this paper:

http://eprints.whiterose.ac.uk/id/eprint/76288

White Rose Research Online
eprints@whiterose.ac.uk

An Analysis of the Server Characteristics and Resource Utilization in Google Cloud

Peter Garraghan, Paul Townend, Jie Xu
School of Computing
University of Leeds
Leeds, UK
{*scpmg, p.m.townend, j.xu*}@leeds.ac.uk

Abstract. Understanding the resource utilization and server characteristics of large-scale systems is crucial if service providers are to optimize their operations whilst maintaining Quality of Service. For large-scale datacenters, identifying the characteristics of resource demand and the current availability of such resources, allows system managers to design and deploy mechanisms to improve datacenter utilization and meet Service Level Agreements with their customers, as well as facilitating business expansion. In this paper, we present a large-scale analysis of server resource utilization and a characterization of a production Cloud datacenter using the most recent datacenter trace logs made available by Google. We present their statistical properties, and a comprehensive coarse-grain analysis of the data, including submission rates, server classification, and server resource utilization. Additionally, we perform a fine-grained analysis to quantify the resource utilization of servers wasted due to the early termination of tasks. Our results show that datacenter resource utilization remains relatively stable at between 40 - 60%, that the degree of correlation between server utilization and Cloud workload environment varies by server architecture, and that the amount of resource utilization wasted varies between 4.53 - 14.22% for different server architectures. This provides invaluable real-world empirical data for Cloud researchers in many subject areas.

Index Terms— Cloud computing, empirical analysis, server characterization, resource utilization, dependability

I. INTRODUCTION

In recent years, there has been an increased effort by the research community to characterize Cloud computing environments. Cloud computing is defined as *a distributed paradigm that enables provisioning of computational resources as a service* [1]. This dynamicity is driven by user demand, and results in a variance of the Cloud workload and user behavior that together composes the Cloud environment [2]. However, to date there have been only very limited analyses of large-scale Cloud environments that consider real-world empirical evidence to show whether these characteristics are quantifiable. To further our understanding of Cloud environments, it is extremely important to analyze the trace logs derived from large-scale real-world Clouds.

Specifically, it is important to explore to what degree the dynamicity of Cloud environments impacts server resource utilization. Quantifying this dynamicity gives researchers and providers a greater insight into the nature of Cloud environments, and subsequently allows research involving resource optimization and data center simulation to be informed and tested with scenarios derived from real empirical data. Unfortunately, due to business and confidentiality concerns, there exists a lack of such data from real Cloud operational environments for analysis.

Recently, limited Cloud computing traces have been made available. In early 2012, Yahoo released traces from the M45 production cluster to a small selection of Universities [3], while Google publically released a relatively short cluster trace spanning seven hours [4] as well as a second version of the trace encompassing a larger time frame and containing more detail about their MapReduce cluster [5]. There has been some limited uptake in analyzing these trace logs, with work focusing on different research objectives including job behavior [6], statistical properties of workload [7][8], as well as machine events and job behavior [9]. However, as stated in [9], due to the massive dataset sizes as well as the required computation and storage power necessary to perform comprehensive analysis, until now it has only been possible to perform analysis at a coarse-grain or perform an in-depth analysis on a small time frame that represents a fraction of the entire trace log.

To improve upon this, we have built a large distributed processing environment that has allowed us to analyze the second version of the Google Cloud trace log, which contains over 12,000 heterogeneous servers and spans a period of 29 days. The aim of this paper is to use this large real-world dataset to characterize the server characteristics and resource utilization of the Google Cloud, in order to give insights into the dynamicity of realistic Cloud environments. Additionally, we explore and quantify the amount of server resource utilization that is wasted due to task failures and resubmissions. Such insights and quantification will be of great benefit to researchers working in the Cloud domain, with direct relevance to a range of topics such as dependability, energy-efficiency, and security.

This paper presents three main contributions. Firstly, we present the statistical properties and a comprehensive coarse-grained analysis of the Google Cloud trace log, including characterizing a high-level view of the trace log

978-0-7695-4945-3/13 $26.00 © 2013 Crown Copyright
DOI 10.1109/IC2E.2013.40

IEEE computer society

TABLE 1 DATASET OVERVIEW

Trace span	29 Days
Average task length	61,575,043
Number of servers	12,532
Average tasks per server	16,653
Average task length (MI)	61,575,043

and server architectures over a period of 29 days. Secondly, we analyze and compare the resource utilization of servers within the trace log by server architecture type, at different time frames. Finally, we perform fine-grained analysis to explore and quantify the impact of tasks that are terminated before successful completion in terms of wasted resource utilization per server. Specifically, we contrast the required amount of resources utilized per server to successfully complete a task, against that of total resource utilization, inclusive of resource utilization that is wasted due to termination of tasks.

The rest of the paper is organized as follows: Section 2 presents a description of the trace log as well as the analysis infrastructure; Section 3 discusses the statistical properties and coarse-grain analysis of the Cloud trace log; Section 4 describes server characterization; Section 5 presents the server resource utilization; Section 6 analyzes wasted resource server utilization due to task failure; Section 7 presents the related work; finally, Section 8 presents the conclusions of our research and discusses future work directions.

II. RELATED WORK

Prior to this paper, there has been some initial research in characterizing Cloud environments by analyzing Cloud trace logs. Chen, et al. [6] focus on the behavior of jobs within the 7 hour trace log released by Google. Kayulya, et al. [7] present statistical properties of the Yahoo M45 cluster and present a simple analytical model to predict job-completion times in Hadoop environments. Reiss, et al. [8] describe the statistical properties of the second version of the Google trace log, presenting a coarse-grain analysis of data center CPU and memory utilization per day, as well as workload characterization. Liu, et al. [9] also analyze the second version of the Google trace log to characterize machine events, as well as analyze CPU usage of tasks within the time frame of a single day.

Our work differs from the above approaches in a number of ways. Aided by the construction of a large-scale analysis infrastructure, we firstly present a more detailed coarse-grain analysis of the second version of the Google trace including submission rates, task characteristics and server utilization. Secondly, our work analyses the CPU and memory utilization of different architecture types over different time periods. Finally, our work presents a comprehensive analysis of the amount of server utilization wasted due to termination of tasks per architecture type.

III. DATASET DESCRIPTION

The data was collected from the second version of the Google Cloud trace log that spans a period of 29 days and contains over 12,000 servers. The trace log consists of tens of millions of records for job, task and server events. A task is defined as the basic unit of computation assigned or performed in the Cloud e.g. MapReduce operations. Within the trace log, tasks are encapsulated within Linux containers. In addition, the log captures normalized CPU, memory and disk utilization per individual task every 5 minutes. The total size of the trace log is approximately 250GB. The trace log is available at [5], and further details about the data schema, and normalization process can be found in [10].

Due to the massive size of the trace log, as well as the query complexity required to extrapolate desired data, it was necessary to set up infrastructure for data analysis. To facilitate this, we used *Hadoop MapReduce 1.0.2* [11] to construct a cluster consisting of 50 physical nodes for data storage and processing power. In addition, queries were executed using *Apache Hive 0.90*, a data warehouse system that facilitates the analysis of large datasets [12]. Utilizing this analysis infrastructure allowed us to decrease query execution time from over 72 hours for a database installed on a single machine to approximately 15 minutes for a complex query.

IV. DATASET OVERVIEW

An overview of the statistics derived from our coarse-grain analysis of the trace log is presented in Table 1. The trace log contains a total of 25,375,377 tasks, 12,532 unique machines and 930 users. Next, we divided the trace log per day. We observe that the number of submitted tasks daily is widely heterogeneous as shown in Figure 1(a).

Furthermore, we observe that there is a loose correlation between the number of users using the Cloud as shown in Figure 1(b), and the number of tasks submitted within the same time frame at the coarse-grain level. Figure 1(c) presents the average length of tasks per day measured in Million Instructions (MI), and it is again observable that there exists a loose correlation between task length, submission rates and the number of users. These results suggest a large heterogeneity that exists within the Cloud environment from the perspective of the workload and user behavior, and this behavior varies in different time frames. An additional observation of interest is that the average number of machines operating per day does not deviate significantly from the average of 12,299 machines on a daily basis as shown in Figure 1(d).

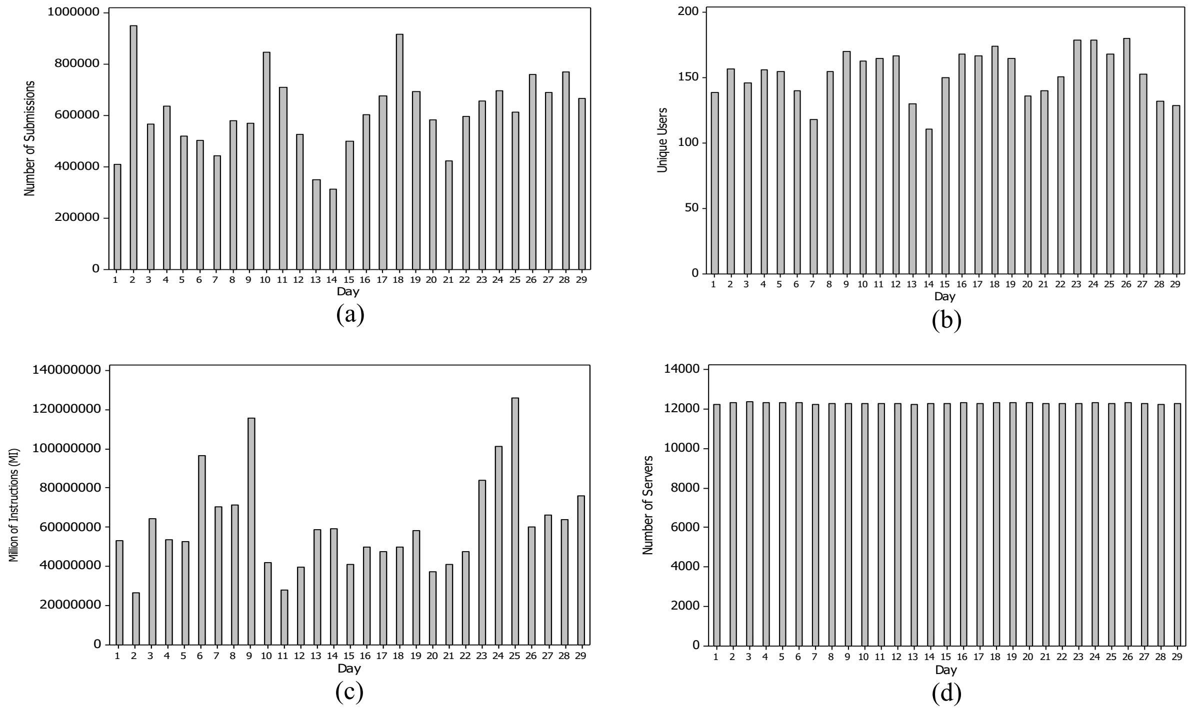

Figure 1. Statistical properties of (a) Submitted tasks, (b) Users, (c) Average task length, (d) Servers in the Google Trace Log

V. SERVER CHARACTERIZATION

Servers are characterized within the trace log by defining unique server architectures using three dimensions: Platform ID, which is a string representing the micro-architecture and chipset version of the machine, and CPU capacity and memory capacity which defines the normalized physical CPU cores and RAM capacity per machine. The last two dimensions are normalized independently [10]. Table 2 shows the breakdown of the number of servers within the trace log by architecture type.

The most populous architecture type is architecture 1, constituting 53.46% of the total number of server architectures within the trace log. Furthermore, architectures 1,2,3 and 8 constitute 98.49% of all architectures, as well as executing 99.1% of all tasks within the trace log. These results are in contrast to the values reported in [9], which claims that 93% of machines are homogenous due to sharing the same CPU capacity. However these observations are inaccurate when considering the heterogeneity of the RAM size assigned to architectures that exist within the trace.

VI. SERVER RESOURCE UTILIZATION

Using the dataset table "Task Resource Usage", which captures the resource utilization per task on a server at a 5 minute time interval, it is possible to extrapolate the total

TABLE 2 SERVER ARCHITECTURE CLASSIFICATION

Server Architecture	Platform	CPU Capacity	Memory Capacity	Server Population %	Task Submission %
1	A	0.5	0.5	53.46	57.89
2	A	0.5	0.25	30.76	25.93
3	A	0.5	0.75	7.93	8.36
4	A	0.5	0.12	0.43	0.19
5	A	0.5	< 0.06	0.03	0.00091
6	A	0.5	1	0.04	0.056
7	B	0.25	0.25	1.00	0.65
8	C	1	1	6.34	6.92

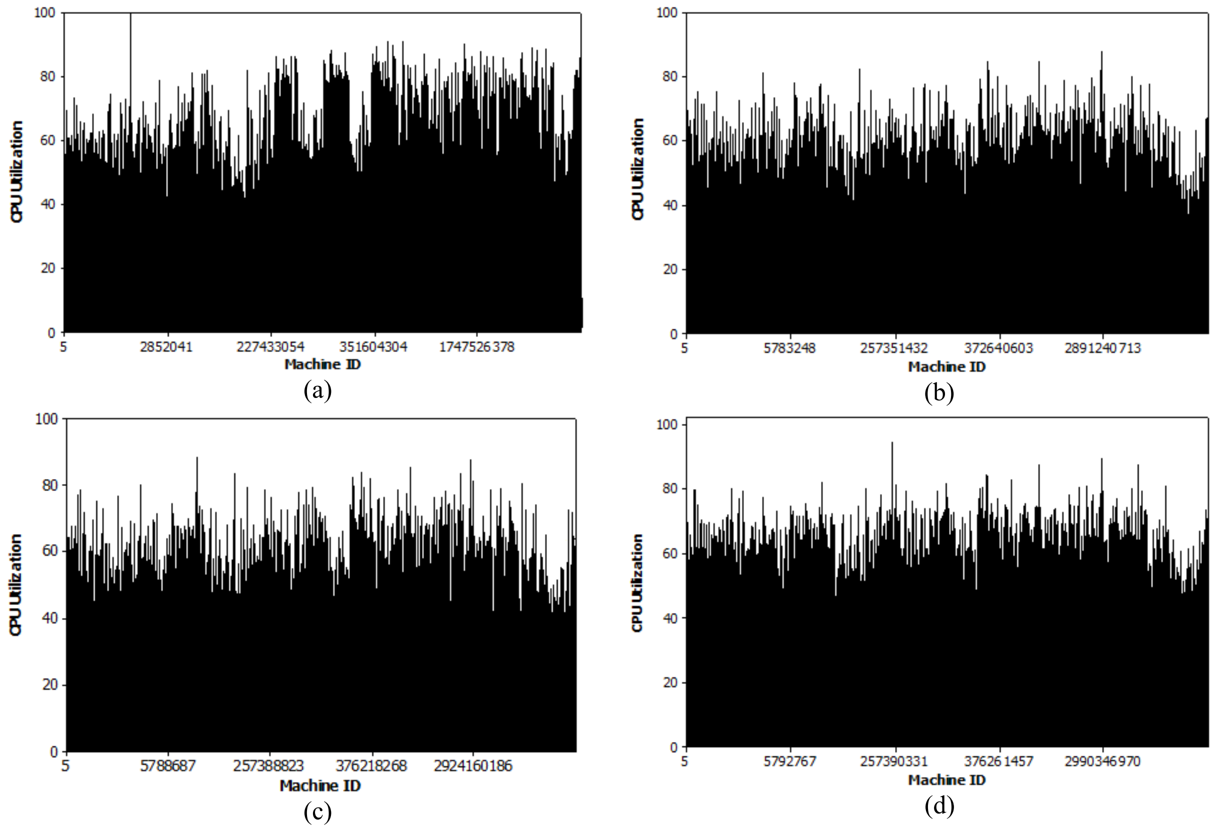

Figure 2. Average CPU utilization of all servers within (a) Day 2, (b) Day 13, (c) Day 14, (d) Day 18

resource utilization of tasks residing within the same physical server within a defined time period; this allows us to calculate the total resource utilization of an individual server. For our subsequent analysis, we focus on assessing server resource utilization through the following approach:

Firstly, server architectures 1,2,3 and 8 are selected for analysis, as they represent over 98% of the total architectures within the trace log population as discussed in Section 4. Furthermore, less than 0.2% of these architectures contain no record of resource utilization of tasks, and have been omitted from analysis. Secondly, we select a sample size of 4 days from the overall trace log population, sampling from days 2, 13, 14 and 18. This sample size was selected for a number of reasons: each day contains in the regions tens of millions of records, which introduces difficulties in term of computer processing for statistical analysis. Also, it is important to perform on a per day basis as this allows us to contrast behavioral patterns across different time periods within the trace log. Additionally, from our coarse-grain analysis presented in Section 3, we have identified days that exhibit high variance between the average task submission and task length in comparison to

TABLE 3 SERVER ARCHITECTURE RESOURCE UTILIZATION

	Architecture 1		Architecture 2		Architecture 3		Architecture 8	
	CPU %	**Memory %**	**CPU %**	**Memory %**	**CPU %**	**Memory %**	**CPU %**	**Memory %**
Day 2	41.55	49.86	32.86	50.83	55.66	39.11	29.18	47.70
Day 13	35.74	47.05	29.49	49.31	41.34	31.55	30.57	48.69
Day 14	35.08	47.21	28.34	48.85	41.46	31.51	31.26	50.04
Day 18	43.94	49.24	35.84	50.85	52.90	34.96	37.52	50.46
Average	39.08	48.34	31.63	49.96	47.84	34.28	32.13	49.22
St. Dev	4.35	1.42	3.40	1.03	7.52	3.60	3.69	1.27

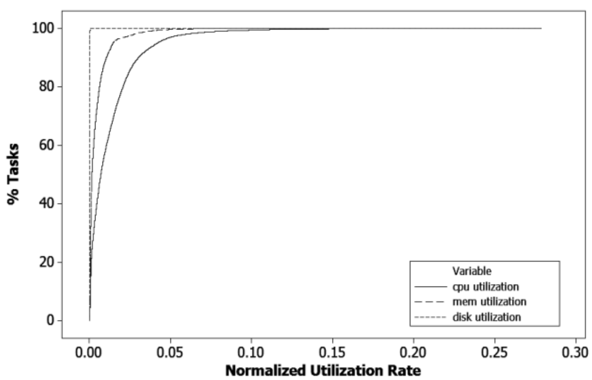

Figure 3 Task utilization resource utilization rates.

the entire trace log. Days 2 and 18 consist of high task submissions rates and low and average task lengths respectively, while days 13 and 14 consist of low task submission rates and average task lengths. Finally, we only consider analyzing the resources utilization of CPU and memory. Disk usage is omitted as a result of 98% of tasks presenting similar usage patterns, as supported in [10,13].

Figures 2(a), 2(b), 2(c) and 2(d) present the total CPU utilization of all servers operating in days 2, 13, 14 and 18 respectively. We observe that the average resource utilization per day is between 40-60% across all sampled days reflecting similar numbers reported in [8]. Next, we analyze the average utilization per architecture type for CPU and memory utilization, with the results shown in Table 3. From this, we can observe a range of average utilization patterns across different architecture types, ranging from 28.34 - 55.66% and 31.51 - 50.83% for CPU and memory respectively, with architecture 3 possessing the highest CPU utilization and lowest memory utilization on average across sampled days.

Furthermore, we observe that the average utilization for architectures 1 and 3 in days 13 and 14 is approximately 7% and 12% lower respectively for CPU than that of days 2 and 18, while the CPU utilization of architectures 2 and 8 remains relatively stable. This is a result of interest due to days 2 and 18 containing nearly three times the amount the amount of tasks compared to day 13 and 14, presenting a stronger correlation between task submission rate and server utilization rates for architectures 1 and 3, compared to architectures 2 and 8. This suggests that within the trace log, the CPU utilization for some architectures is more strongly influenced by the Cloud environment than in other architectures, and that CPU utilization is correlated with the workload environment as depicted in Figure 1(a).

Moreover, utilization of memory remains relatively stable for architectures across all days, suggesting that there is a loose correlation between the utilization of memory in a server and workload behavior in the Cloud environment. This is also a result of interest, as it is intuitive to assume that all resource utilization at t[1]he server level would be correlated strongly with the workload environment.

We postulate a number of reasons for the above observations and behaviour. Firstly, the Cloud environment analyzed may deploy a load balancing technique to attempt to keep utilization levels of servers stable regardless of the behavior of the Cloud workload environment. Secondly, certain architecture types may be assigned tasks to possessing certain conditions that can only be fulfilled by a specific architecture type. However initial analysis shows that only 5% of tasks possess one or more constraints when allocated to specific server architecture. Additionally, the utilization rate variability, which is higher for CPU in comparison to Memory or Disk as shown in Figure 3, can also affect the correlation between server utilization and workload. This is because a few tasks have such high utilization rates that they require specific server characteristics that are not common in the analyzed datacenter. In this scenario, utilization is high in those common servers that are used by low utilization rate tasks, and low in those uncommon servers used by the high utilization rate tasks.

VII. TASK TERMINATION RESOURCE UTILIZATION

The remainder of the paper is dedicated to exploring and quantifying the degree of resource utilization wasted due to

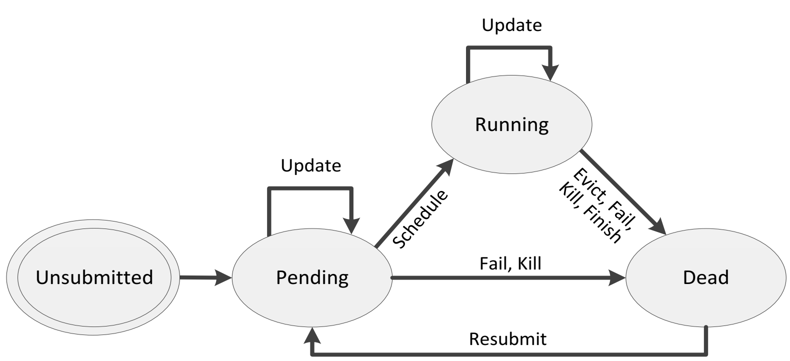

Figure 4. Task life cycle in Google Cloud

TABLE 4. SUMMARY OF DAY 2
WASTED CPU RESOURCE UTILIZATION

	CPU	Arch. 1	Arch. 2	Arch. 3	Arch. 8
Full Task	Average	41.55	32.86	55.66	39.12
	St. Dev	14.16	12.38	17.24	8.44
Comp. Task	Average	35.08	28.33	41.44	31.26
	St. Dev	13.77	12.49	15.06	7.79
	Disparity	6.46	4.53	14.22	7.86

TABLE 5. SUMMARY OF DAY 2
WASTED MEMORY RESOURCE UTILIZATION

	Memory	Arch. 1	Arch. 2	Arch. 3	Arch. 8
Full Task	Average	49.86	50.83	39.11	51.32
	St. Dev	15.71	13.01	13.71	13.39
Comp. Task	Average	47.21	48.84	31.51	50.03
	St. Dev	17.54	14.70	15.13	13.54
	Disparity	2.66	1.99	7.61	1.29

tasks terminations within the Cloud environment. The trace log contains an event log for task events. In this context an *event* is defined as an action that causes the state of a task to change at a specific time and space. The task event log describes the life cycle of tasks as depicted in Figure 4. After a task is submitted into the system it is scheduled by the Cloud scheduler to a server and then executed.

An individual task may only be scheduled and executed on one server at a time. It is possible for a task to transition to a DEAD state in a variety of ways, including successful completion (FINISH) as well as task termination. In this context, we define the *termination* of a task as an event that results in unsuccessful task completion. Task termination is defined by four events as described in [10]. A task descheduled due to task failure is classified as a FAIL event, a task that is rescheduled as a result of a higher priority task due to scheduler over commitment or server disk failure is defined as an EVICT event, while a task cancelled by a user or program, or a failure due to a job dependency, is a KILL event. If a task becomes DEAD for any other reason besides

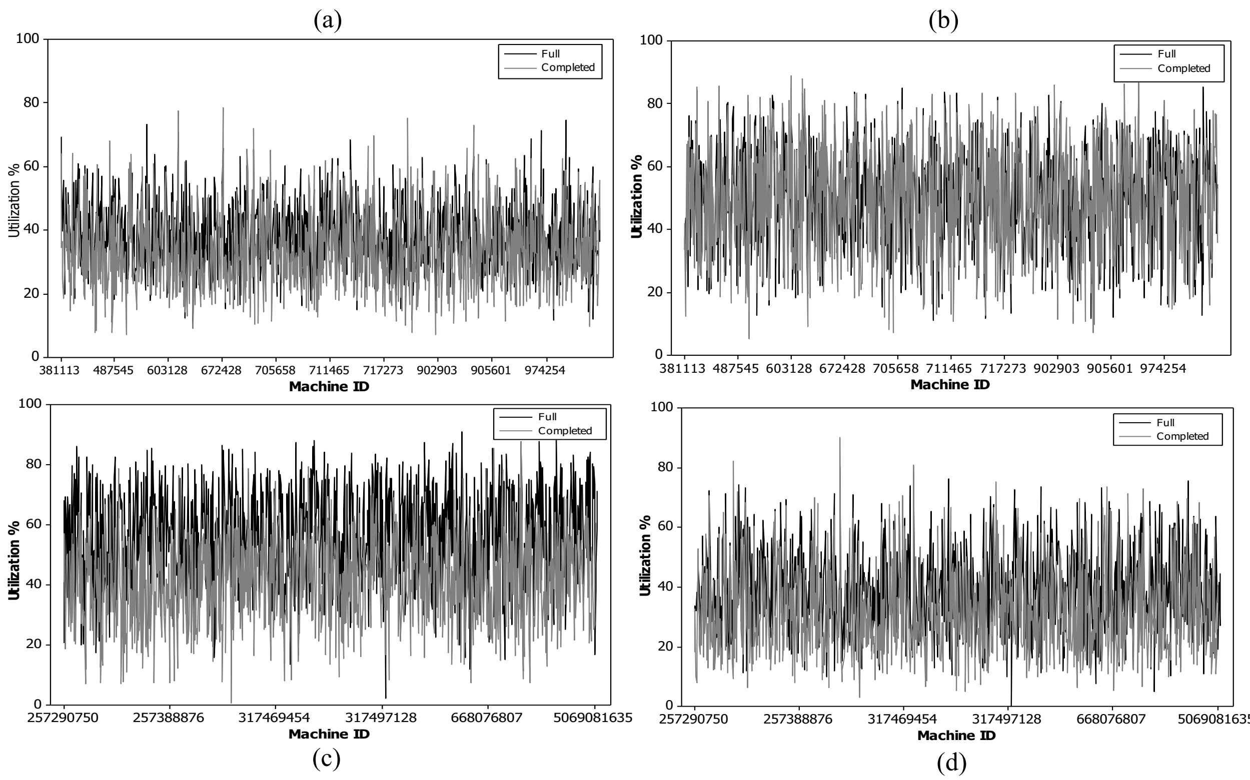

Figure 5. Comparison of full and completed tasks in day 2 for (a) Arch. 1 CPU utilization, (b) Arch. 1 memory utilization, (c) Arch. 8 CPU utilization, (d) Arch. 8 memory utilization.

successful task completion, it is resubmitted to the scheduler and allocated to another server.

Task resubmission due to task termination results in the preceding work being lost, and is therefore a waste of server resource utilization. To quantify the exact resource utilization wasted per server as a result of task terminations within the Cloud environment, we contrast the sum of task resource consumption required to successfully complete a task in ideal environmental conditions (omitting events that can cause the task to terminate), against the sum of actual server resource utilization by tasks, including preceding work lost due to terminations within the trace log. To facilitate this, we define a number of definitions and assumptions.

A *completed task* is defined as the period of time and resource consumption of a task between the latest scheduling, and the completion event with no termination events present. We assume that tasks do not use check-pointing, and that therefore a task resubmitted into the Cloud results in the task restarting from the beginning. This is supported by [10,14], which states that a task failure results in "*...an interruption on a running task, requiring the system to re-execute the interrupted task*". This re-execution represents meaningful work performed by the task towards ultimate task completion.

A *full task* is defined by the total task duration and resource utilization from the first submission of the task into the Cloud until successful completion, inclusive of work performed before task failures and resubmissions. This encompasses both the expended time and resource consumption of a completed task, as well as work performed by the task that has been wasted due to termination events. The cause of a *task termination* within the Cloud environment encompasses individual task failure, the failure of a server, and the eviction of a task by the scheduler. The analysis and breakdown of the failure rate and root cause of those termination events that lead to wasted resource utilization is not within the scope of this paper. These events are very frequent, and occur in terms of hundreds of thousands (as shown in Figure 6), representing considerable resource utilization in the analyzed datacenter.

Furthermore, all tasks that contain both a schedule and a completion event are considered for analysis. Tasks that are not scheduled or completed within the total trace log time frame are not considered, as it is impossible to define task duration if the schedule or completion time is unknown. The majority of tasks excluded from analysis as a result of this condition are tasks used for monitoring purposes within the trace log, which represent 13% of the total tasks within the observational period.

Figures 5(a) and 5(c) show the CPU utilization disparity between full tasks and completed tasks for architectures 1 and 8 in day 2, and Figures 5(b) and 5(d) present the

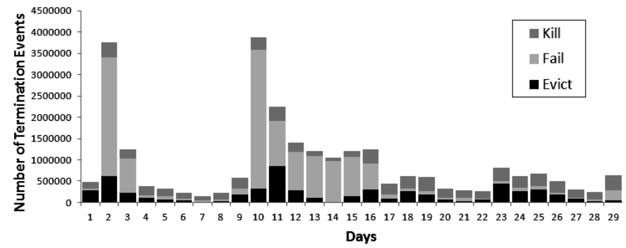

Figure 6 Termination events breakdown per day.

utilization disparity for memory within the same time frame as well as per server architecture type. It is observable that there exists a noticeable level of resource disparity between full tasks and completed tasks, and that different architecture types within the same time frame experience different levels of disparity. Tables 4 and 5 show the wasted CPU and Memory utilization disparity in more detail for all four architectures in day 2. These results show that there is a 4.53 - 14.22% and 1.29 - 7.61% resource disparity between full and completed tasks for CPU and memory utilization respectively.

We postulate two possible causes of this resource disparity. The first is the result of the Cloud scheduler evicting tasks across all servers. However as discussed previously and presented in Table 2, the number of tasks submitted to a particular server architecture type is in proportion to the total server population, agnostic of server type, and only 5% of tasks contain one or more constraints to scheduling location. The second reason for resource waste is the result of the Cloud workload environment being driven by user behavior. An example of this behavior is presented in [8], which observed that Day 2 exhibits a high number of task failure events within the time period. These results should be emphasized, as they offer *empirical evidence of the economic consequences of task failure within the Cloud environment*. Furthermore, a resource disparity of 4.54% for CPU should not be overlooked as an inconsequential figure when considering that there could be potentially thousands of servers with similar values. This waste of resources translates into economic loss for providers in the form of energy consumption, as well as reduced availability of servers.

VIII. CONCLUSIONS

In this paper, we have presented a characterization and analysis of server resource utilization within a large-scale production Cloud consisting of over 12,500 servers over a 29 day time span. Furthermore, we have presented a coarse-grain analysis of the statistical properties of the trace log, giving an insight into the degree of dynamicity and variability of workload that exists within a real Cloud environment. Finally, we have explored and quantified the amount of resource utilization wasted by servers due to task failure for CPU and memory. In this work we have been able to make a number of observations and conclusions. They are summarized as follows:

- Analyzing the statistical properties of the trace log quantifies and presents empirical evidence to substantiate the claim of workload diversity and dynamicity that exists within the Cloud environment. Our analysis demonstrates a high level of variability in workload characteristics and submission rates across different time spans in the Cloud environment that is not cyclical.
- Our analysis shows that utilization within the trace log remains at a constant 40-60% per architecture type across sampled days, and that the level of correlation between resource utilization and workload variability for a server is dependent on architecture type. This observation suggests that server resource utilization is not heavily influenced by the dynamicity of workload within the Cloud environment.
- We present empirical evidence of wasted resource utilization within servers due to task failure. We have quantified the amount of wasted utilization, and have discovered that the average server architecture type within the trace log wastes between 4.54 - 14.22% and 1.26 - 7.61% utilization for CPU and memory respectively. Furthermore, this level of wasted utilization varies by the server architecture type. In addition, we postulate that the cause of resource waste due to task termination is primarily driven by the Cloud workload environment, more so than the Cloud scheduler.

This work has shown great potential, and has the potential for a great deal of future endeavor. Our plans for future work include a more fine-grained analysis of resource utilization across the entire trace log, as well as analysis of resource utilization on an hourly basis to assess the behavior of "Cloud bursting". Furthermore, we plan to include a more fine-grained analysis on the correlation and effects of *workload types and behavior* on server behavior and characterization.

An interesting research direction based on the results presented in this paper would be a detailed failure analysis across different server architecture types, as well as characterizing specific failures to calculate the resultant cost of energy consumption and economic impact for providers. Furthremore, using the results presented this paper, it is possible to develop distributions of the utilization and characteristics of servers that can then be integrated into Cloud simulation tools to facilitate more realistic assessment of Cloud architectures and algorithms.

ACKNOWLEDGMENTS

This work is supported by the National Basic Research Program of China (973) (No. 2011CB302602), the UK EPSRC WRG platform project (No. EP/F057644/1), and the Major Program of the National Natural Science Foundation of China (No. 90818028). We would like to thank John Hodrien at the University of Leeds for his assistance and support in developing the Hadoop MapReduce analysis infrastructure.

REFERENCES

[1] P. Mell and T. Grance. NIST definition of Cloud computing. National Institute of Standards and Technology. October 7, 2009.

[2] R. Buyya, et al., "InterCloud: utility-oriented federation of cloud computing environments for scaling of application services," presented at the Proceedings of the 10th international conference on Algorithms and Architectures for Parallel Processing - Volume Part I, Busan, Korea, 2010.

[3] Yahoo. Yahoo! M45 Supercomputing Project. Available: http://research.yahoo.com/node/1884

[4] Google. Google Cluster Data V1. Available: http://code.google.com/p/googleclusterdata/wiki/TraceVersion1

[5] Google. Google Cluster Data V2. Available: http://code.google.com/p/googleclusterdata/wiki/ClusterData2011_1

[6] Y. Chen, A. S. Ganapathi, R. Griffith, and R. H. Katz, "Analysisand lessons from a publicly available google cluster trace," EECSDepartment, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-95, Jun 2010. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-95.html

[7] R. G. Soila Kavulya, Jiaqi Tan and P. Narasimhan, "An analysis of tracesfrom a production mapreduce cluster," Carnegie Mellon University, Tech. Rep., 2009.

[8] C.Reiss, A,Turnanov, G.R.Ganger, R.H.Katz and M.A.Kozuch, "Towards Understanding Heterogeneous Clouds at a Scale: Google Trace Analysis," Carnegie Mellon University, Tech.Rep., 2012

[9] Z.Liu, S.Chi, "Characterizing Machines and Workloads on a Google Cluster," 8th International Workshop on Scheduling and Resource Management for Parallel and Distributed Systems, 2012

[10] C. Reiss, et al., "Google Cluster-Usage Traces: Format + Schema," Google Inc., White Paper, 2011

[11] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," presented at the Proceedings of the 6th conference on Symposium on Opearting Systems Design \& Implementation - Volume 6, San Francisco, CA, 2004.

[12] A. Thusoo, et al., "Hive: a warehousing solution over a map-reduce framework," Proc. VLDB Endow., vol. 2, pp. 1626-1629, 2009.

[13] A. K. Mishra, et al., "Towards characterizing cloud backend workloads: insights from Google compute clusters," SIGMETRICS Perform. Eval. Rev., vol. 37, pp. 34-41, 2010.

[14] J. A. Quiane-Ruiz, et al., "RAFTing MapReduce: Fast recovery on the RAFT," in Data Engineering (ICDE), 2011 IEEE 27th International Conference on, 2011, pp. 589-600.