2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)

How Does the Workload Look Like in Production Cloud? Analysis and Clustering of Workloads on Alibaba Cluster Trace

Wenyan Chen,[1,2], Kejiang Ye[1,*], Yang Wang[1], Guoyao Xu[1,3], Cheng-Zhong Xu[1]

[1]*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences*
[2]*College of Information Science and Technology, Northeast Normal University*
[3]*Department of Electrical and Computer Engineering, Wayne State University*
{wy.chen, kj.ye, yang.wang1, gy.xu, cz.xu}@siat.ac.cn

***Abstract*—Cloud computing technology is widely used in today's datacenters due to the benefits such as high scalability, on-demand services and low cost. An in-depth understanding of the characteristics of workloads running in production cloud environments is very important for improving the resource management efficiency. In this paper, we make a detailed analysis with visualization techniques and clustering methods on the trace dataset released by Alibaba which contains 11089 online services and 12951 batch jobs running on 1313 machines. Our methodology for clustering workloads contains: i) Select effective feature vectors as the dimensions of clustering; ii) Identify the cluster boundaries of each dimension using K-Means algorithm; iii) Classify jobs by combining the feature vectors which uses the results from previous step; iv) Analyze the characteristics of workload groups at runtime. Our analysis reveals several insights which previous work has not found on Alibaba cluster trace. For batch jobs: a) Average CPU cores of all batch jobs show bimodal-distribution obviously. b) At a random sampling time, more than 50% machines only run one group of jobs with a short duration, medium CPU cores and small memory utilization, the remaining machines run mixed groups of jobs. For online instances: a) The resource usage (CPU, Memory, and Disk) of most online instances is low; b) There are up to six groups running on the same machine according to our clustering method at a random sampling time.**

***Keywords*-Alibaba Trace; Workload Pattern; K-Means Clustering**

I. Introduction

Production cloud environment usually contains a huge number of machines, providing highly reliable services to end users. With the development of cloud computing technology and the ever-increasing number of cloud data centers, more and more users choose to migrate their services to clouds. By doing so, users can greatly reduce the total cost of ownership (TCO) and up-front commitments. However, this benefit is not free, it comes at the cost of bringing additional complexity to cloud managements. One of the challenges is how to implement efficient resource management for diverse workloads running in the same physical machine. Although resource management and scheduling has been widely studied by many researchers from different perspectives [1], [2], [3], there is still a large optimization space for the resource management of clouds running with mixed workloads (such as online services and offline batch jobs). A deep understanding of workload characterization benefits both machine capacity planning and job scheduling.

In typical production clouds, online services and offline batch jobs are usually co-located in the same machine to maximize resource utilization. However, co-location also brings great challenges to the existing cluster management methods. Resource utilization and QoS (Quality of Services) of diverse applications should be optimized together. Currently, Alibaba uses co-location strategy to support various application services, including online interactive services and offline computational services [4]. The conflict between different types of services on resource preemption and contention leads to certain services to be delayed and can not provide satisfied QoS for users. It is a common problem other cloud platforms are also faced. Understanding the workload characteristics and patterns will not only provide workload prediction and effective scheduling for cluster management, but also avoid resource preemption caused by uneven resource allocation strategy.

In this paper, we perform clustering and analysis of workloads on Alibaba cluster trace [5] which was released on September 2017 by Alibaba, one of the largest cloud computing companies in the world. Alibaba cloud processes millions of jobs or instances everyday in order to provide satisfied services to end-users. This dataset not only contains batch jobs, but also contains online services, whose instances running on containers. This paper is a follow-up work of our previous paper [4], in which we performed a detailed statistic studies on Alibaba trace and found multiple *imbalance* phenomena in the cloud. While in this paper, we find many new discoveries by using machine learning methods to deeply process the data. The main contribution of this paper is as follows:

- Perform clustering analysis on batch jobs and online service running on Alibaba cluster trace, and identify workload patterns. We first select effective feature vectors as the dimensions of clustering; then identify the cluster boundaries of each dimension using K-Means algorithm; After that, we cluster jobs by combining the

*Corresponding author

978-1-5386-7308-9/18/$31.00 ©2018 IEEE
DOI 10.1109/ICPADS.2018.00024

Table I
BASIC STATISTICS OF THE SIX DATA TABLES

Table name	Data entries	Data attributes	unzipped size(kb)
server_event	1352	7	47
server_usage	187963	8	18585
batch_task	80553	8	4524
batch_instance	16094656	12	838425
container_event	11102	8	855
container_usage	1480906	12	269525

feature vectors.

- Analyze and identify job group characteristics and the relationship among different job groups. Based on the job group characteristics, we can further schedule suitable jobs to the same machine and reduce the interference by conflicted jobs.

Our analysis reveals several insights which previous work has not found on Alibaba cluster trace.

- **For batch jobs:** i) Average CPU cores of all batch jobs show bimodal-distribution obviously; ii) At a random sampling time, more than 50% machines only run jobs with a short duration, medium CPU cores and small memory utilization. While the remaining groups of jobs are mixed running in other machines.
- **For online instances:** i) The resource usage (CPU, Memory and Disk) of most online instances are very small; ii) There are up to six groups running on the same machine according to our clustering method at sampling time.

The rest of the paper is organized as follows: Section II describes the Alibaba trace dataset. Section III uses K-Means algorithm to cluster workloads and identify workload patterns. Section IV discusses related works. The conclusion and future work are presented in Section V.

II. TRACE DATASET OVERVIEW

Alibaba released a cluster trace dataset on September 2017. This dataset contains online services and offline batch jobs which are co-located on 1313 machines. For online services, the trace collected almost 12 hours instances running on containers. For batch jobs, a job contains multiple tasks, different tasks execute different computing logics. Tasks form a DAG according to the data dependency. Instance is the smallest scheduling unit of batch workload.

The trace contains six files in CVS format in Table I. Server_event and server_usage describe the configuration information and occupancy of the machine CPU, memory, disk, and so on. Batch_task and batch_instance are the description of duration and resource consumption of the tasks and instances of the batch jobs running on the machines. Container_event and container_usage describe the running duration and resource occupation of the containers running on the machine. The online services run in containers.

III. WORKLOAD CLUSTERING

A. Step one: select clustering algorithm

K-Means clustering [6], mean shift clustering [7], density-based clustering method [8] are commonly used clustering algorithms in machine learning community. K-Means is usually used for workload clustering with following advantages: i) it is a classical algorithm to solve the clustering problem, the algorithm is simple and fast. ii) For processing large datasets, the algorithm is relatively scalable and efficient with a complexity of O(nkt), where n is the number of all objects, k is the number of clusters, and t is the iteration times (usually $k \ll n$). This algorithm often ends with a local optimum value. iii) The algorithm attempts to find the k partitions that minimize the value of the squared error function. When clusters are dense, spherical or lumpy, and the difference among clusters is obvious, the clustering works well.

K-Means is an unsupervised clustering method. It needs to specify the number of clusters k manually, then randomly select the cluster center point, perform multiple iterations until the center point is unchanged, or the euclidean distance between cluster points and the cluster center point of their cluster is unchanged. When the distance converges, or calculated cycle number is not less than the preset number of iterations, the iteration stops.

B. Step two: select feature vector

1) Batch Jobs: According to the trace data, batch jobs run on the Alibaba cluster over 12 hours. Among batch jobs, the task status of each batch job contains Ready/Waiting/Running/Terminated/Failed/Canceled, we only select the tasks with the terminated status for analysis in order to get more accurate clustering results.

- Job duration

According to trace dataset, 12,951 job samples are collected in batch jobs. The number of jobs with a status of terminated is 11,641, the maximum running duration is 8.2181 hours, and the minimum running duration is 0.0028 hour. From Fig. 1, the job distribution of each running duration can be seen intuitively.

It can be seen from Fig. 1 that the majority of job number is in the interval of $0 \sim 0.08$h, the number almost reaches 8000, the number of jobs in the interval $0.08 \sim 0.16$ is the second with more than 1000. The number of jobs in the remaining sections is much less. As the running time increases, the number of jobs tends to decrease, so it can be seen that the most batch jobs are short jobs. The average duration is 0.1009h, the maximum duration is 8.2181h, and the minimum duration is 0.0003h.

- Average CPU cores

Statistical analysis is performed on the average CPU usage of all jobs with the terminated status. Since a job contains multiple tasks, and a task contains a plurality of

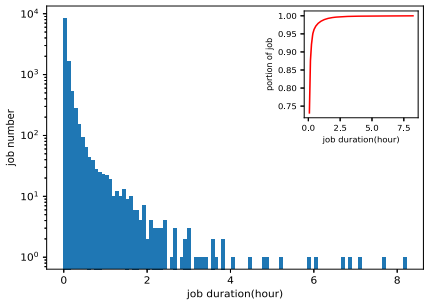


Figure 1. Batch job number each job duration period. Note the log_scale on the plot's y-axis.

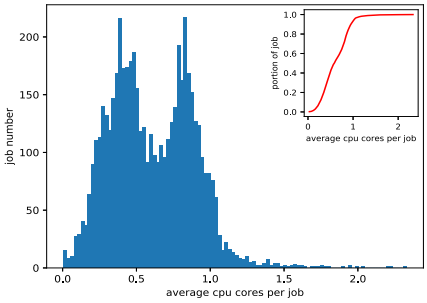


Figure 2. Batch job number of average CPU cores. Note the CPU cores is not normalized.

instances distributed on different machines. The average CPU occupancy of job_k is calculated as:

$$cpu\_avg\_of\_task_j = avg(\sum_{i=1}^{n}(real\_cpu\_avg\_of\_instance_i))$$

$$avg\_cpu\_job_k = avg(\sum_{j=1}^{m}(cpu\_avg\_of\_task_j)))$$

Similarly, average memory utilization of job_k is calculated in the same way as CPU:

$$mem\_avg\_of\_task_j = avg(\sum_{i=1}^{n}(real\_mem\_avg\_of\_instance\ _i))$$

$$avg\_mem\_job_k = avg(\sum_{j=1}^{m}(mem\_avg\_of\_task_j)))$$

Note: i = 1, 2, 3, ..., n, n is instance number of j-th task, j = 1, 2, 3, ..., m, m is task number of k-th job

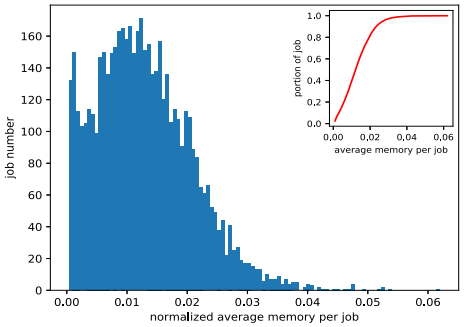


Figure 3. Batch job number of average memory utilization. Note the memory utilization is normalized.

It can be seen from Fig. 2 that the average utilization of CPU cores has a bimodal distribution phenomenon, that is, it mainly concentrates in two interval ranges: $0 \sim 0.7$ and $0.7 \sim 1.5$. The average value is 0.6075, maximum value is 2.3280 and minimum value is 0.0077.

- Average memory utilization

Fig. 3 describes the distribution of the average memory utilization of each job. In all sampled jobs, the maximum memory occupancy is 0.0620, the minimum is 0.0003, and more than 90% jobs are distributed in the interval less than 0.04.

2) Online Service: Fig. 4 describes the average resource (CPU/memory/disk) utilization distribution of an instance running in a container over all sampling times. For CPU, the maximum average CPU utilization is 47.8873%, the minimum is 0%, and more than 60% instants are distributed between 0% and 20%. For memory, the maximum average memory utilization is 76.0%, the minimum is 1.0%, and most instances are distributed between 25% and 50%. For disk, the maximum average disk utilization is 100.0%, the minimum is 2.0575%, and more than 90% instances are distributed between 0% and 50%. As can be seen from Fig. 4, the number of instances is about 30 when memory utilization is the largest. The total number of instances is 10359.

We select feature vectors according to the degree of influence of the attributes on the clustering results [9], [10]. For batch jobs, we select job duration, average CPU, average memory utilization as the feature vectors for clustering. For online services, we select the average CPU, average memory, average disk as feature vectors for clustering.

C. Step three: determine the boundaries of each dimension

The group number needs to be determined artificially when clustering groups. K-Means methodology provides a method for evaluating the clustering results - silhouette coefficient. Its evaluation criteria of clustering is the extent of dispersion among the objects belong to different clusters

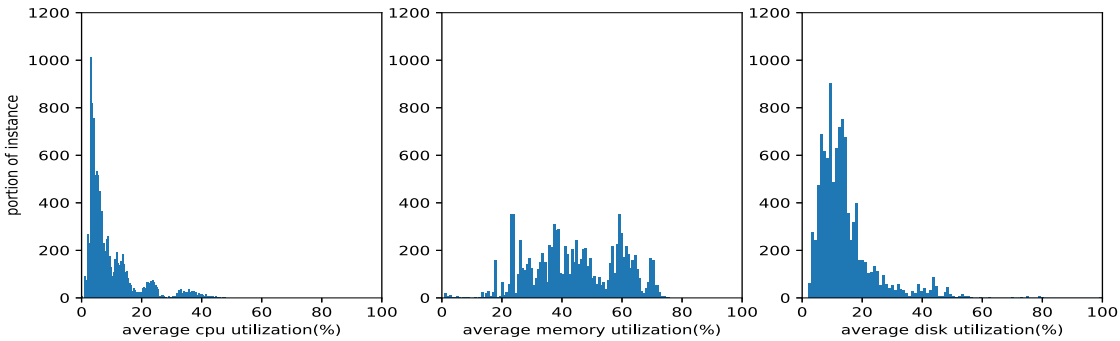


Figure 4. Average CPU/memory/disk utilization (%) of online service instances.

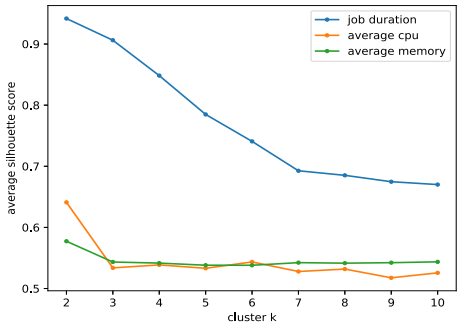


Figure 5. Average silhouette coefficients for batch workload's feature vector.

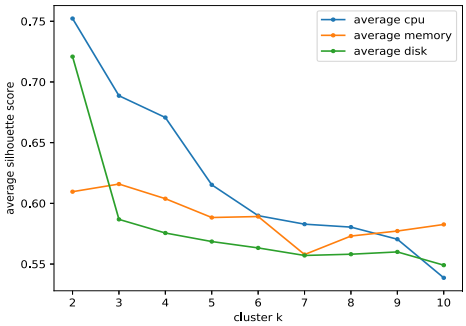


Figure 6. Average silhouette score for online service's feature vector.

Table II
THE BOUNDARIES OF FEATURE VECTORS FOR BATCH JOBS

Feature vector	Range (short)	Range (medium)
job duration	0.0∼0.1534	0.1534∼8.02181
average CPU	0.0∼0.6238	0.6238∼2.3280
average memory	0.0∼0.0144	0.0144∼0.0621

Table III
THE BOUNDARIES OF FEATURE VECTORS FOR ONLINE SERVICE

Feature vector	Range (short)	Range (medium)
average CPU	0∼17.0986	17.0986∼48.0
average memory	0∼45.5775	45.5775∼76.0
average disk	0∼22.8332	22.8332∼100.0

or belong to same cluster. It combines the cohesion and separation where cohesion is the average distance from i-th object to other objects in the same cluster, denoted as a_i; the separation is the average distance of the i-th object from all objects in the cluster that does not contain it, denoted as b_i; and the silhouette coefficient of the i-th object is $s_i = \frac{b_i - a_i}{max(a_i, b_i)}$, while s_i value is between -1 and 1. The s_i value is larger, the clustering result is better. The k-values for clustering are respectively evaluated for the feature vectors which are selected by the two kinds of services, and the clustering group number is generally not greater than 10. Therefore, the selection range of k is set between 2 and 10.

1) Batch Jobs: It can be seen from Fig. 5 that when evaluating the clustering effect of job duration, average CPU, and average memory, the k values corresponding to maximum average silhouette coefficients are 2, 2, 2 respectively. Therefore, the three feature vectors can be divided into short and medium groups by using K-Means. Table II describes these boundaries of feature vectors.

2) Online Service: Fig. 6 shows the average silhouette coefficient where the three lines represent the three feature vectors of the online service which are average CPU, average memory, and average disk utilization. It can be seen that the clustering effect is the best when the k values of the three feature vectors are 2, 3, and 2 respectively. However, the number of groups classified into 12 is too large, and the feature vector of average memory has a small difference when the values are 2 and 3. In order to facilitate analysis, the average memory is also divided into two intervals. The results of the K-Means algorithm are shown in Table III.

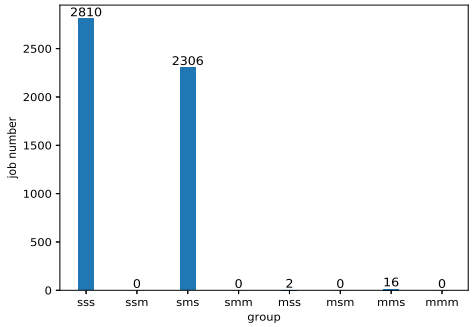


Figure 7. Batch job number of per group with K-Means.

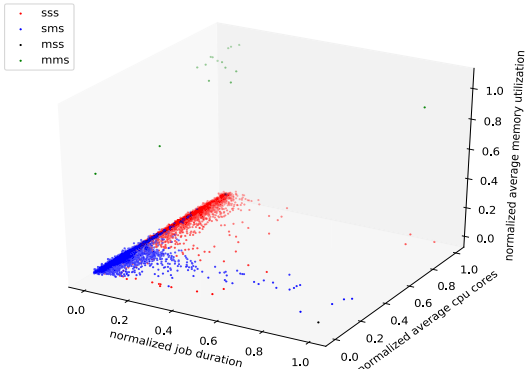


Figure 8. Three-dimensional map of batch job groups.

D. Step four: clustering results and feature analysis

1) Batch Jobs: The batch jobs can be divided into 2*2*2 clusters, and the statistical analysis of the eight groups for batch jobs is performed according to the feature boundaries division as shown in Fig. 7. Note that s is denoted as short or small, m as medium, and the selected feature order is job duration, average CPU and average memory. For example, the group sms means short job duration, medium CPU cores and small memory utilization.

As can be seen from Fig. 7, only four groups of job numbers are greater than zero, which are sss, sms, mss, mms. Fig. 8 is the three-dimensional map of offline batch job groups. Obviously the number of sss group and sms group almost occupies 90% while group mss only has 2 jobs, group mms has 16 jobs.

We use CDF (Cumulative Distribution Function) figures to describe changes of resource utilizations and job execution duration for batch job groups as shown in Fig. 9. We can see each line of different color represents a group of batch jobs, the job duration of sms group is very short, all of them are less than one hour. While the average CPU cores and the average memory utilization are more than others. And the sss also executes about ten minutes while the utilization of CPU and memory is less than sms. The mms group has one job executed nearly 9 hours while its CPU is almost more than sss and sms. But for memory, one part of it is less than sss and sms, and the other part is more than them.

The trace collects instances whose status is running in the batch job within 24 hours. By counting the job groups running on each machine at a certain time, it can be seen which types of jobs are often co-located together. We select time 43200s as an example. From previous analysis there are four groups of jobs: sss, sms, mss, mms. If there is a certain group of instances running on the machine, the value of the group is set to 1, otherwise set to 0. That is, the number of combination of all modes is 16 (i.e., 0000, 0001, ..., 1111), as shown in Table IV.

As shown in Fig. 10, there are five modes are greater than zero which are 0000, 0100, 0101, 1100, 1101. Only 757 machines (57.65%) are assigned with sms group. 468 machines (35.64%) run sms and mms, which means sms and mms are often allocated together. 24 machines (1.83%) run only sss and sms, 43 machines run sss, sms, and mms (3.27 %). According to these observations, we find that sms and mms are complementary in CPU. While in other resource dimensions, they may also be complementary due to the experimental result.

Table IV
WORKLOAD PATTERNS FOR BATCH JOBS

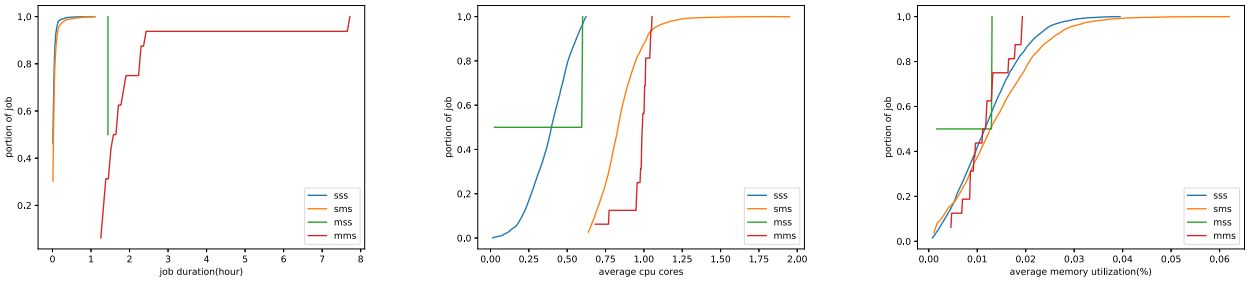
Group	sss	sms	mss	mms
pattern 1	0	0	0	0
pattern 2	0	1	0	0
pattern 3	0	1	0	1
pattern 4	1	1	0	0
pattern 5	1	1	0	1
...				
pattern 16	1	1	1	1

2) Online Service: The online service can be divided into 2*2*2=8 groups. The number of instances for each group is shown in Fig. 11. Fig. 12 is the three-dimensional map of online service groups.

From the figures, the number of instances with the group sss is the highest, with 4869 instances, and the number of sms group is 3052. These two groups occupy more than 70%. And the number of msm is the least, with only three instants.

We also use Fig. 13 to describe the CDF resource utilizations of offline batch jobs. We find there are three lines which represent ssm, sms, smm are similar in the distribution of resource utilization, while mss, msm, mms, mmm groups have some difference.

We sample a time 43200s randomly in order to analyze instance group pattern on machines. In Fig. 11, all groups are more than zero, so if we calculate the combinations of all groups, there are totally $2^8 = 256$ groups. From Fig. 14, we find that not all combination patterns are larger than zero and the workload patterns 10100000 (i.e., sss-sms) and



(a) job durations (b) job average CPU cores (c) job average memory utilization

Figure 9. CDF of job durations, job average CPU cores and job average memory utilization for batch job groups.

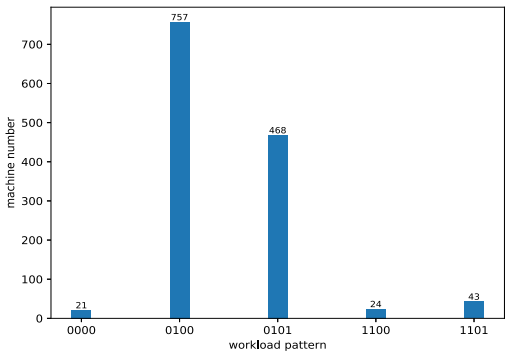


Figure 10. Machine distribution for job groups pattern at 43200s.

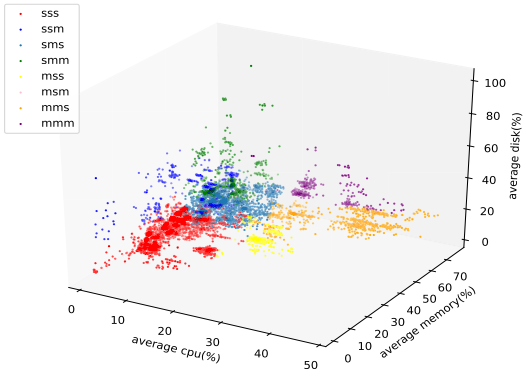


Figure 12. Three-dimensional map of online service group.

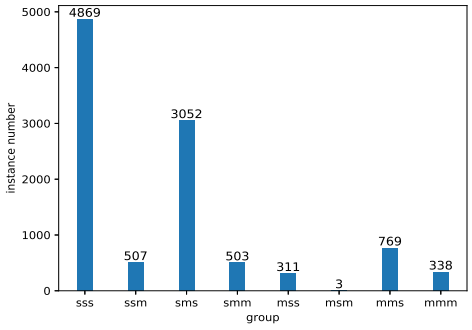


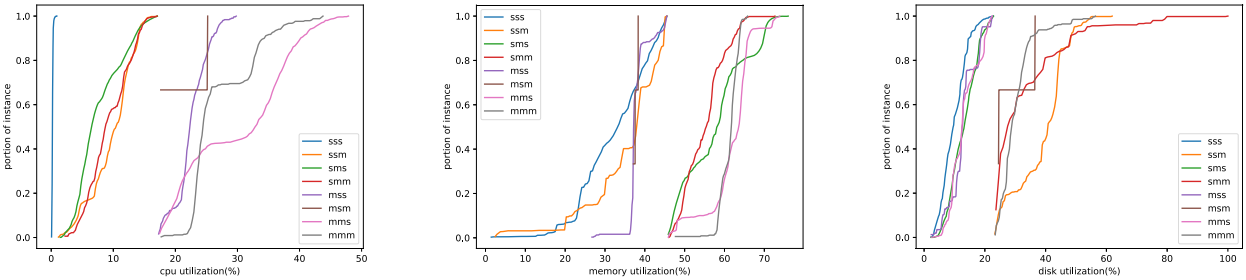
Figure 11. Instance number distribution of online service groups using K-Means Clustering.

10100001 (i.e., sss-sms-mmm) occupy 26% while the other patterns occupy less number of machines. According to the above figures, we have four observations as follows:

- ***Observation 1:*** *sss and sms are often running on same machine together, it may due to the complementary resource demand of this two groups of workloads running on the same machine.*
- ***Observation 2:*** *sss, ssm, sms are often allocated together running on physical machines.*
- ***Observation 3:*** *sss, ssm, sms, smm are allocated together running on about 20% physical machines.*
- ***Observation 4:*** *there are up to six groups running on the same machine, showing the ability to host diverse workloads with different characteristics.*

IV. Related Work

Workload characterization has been studied in many past works. Calzarossa et al. did a survey on workload characterization [11] in 1993. Wang et al. [12] analyzed the workloads of distributed file system. The predictive accuracy reaches 95% through analyzing similar workloads behavior and features on a six months of data for 139 enterprise applications. With the release of Google cluster dataset [13] in 2010, many researchers carried out workload analysis on this dataset in order to provide insights for resource scheduling and planning in cloud environment. Reiss et al. [14] characterized cluster's resource requests, and the actual resource utilization. Fan et al. [15] proposed a function based generalized feature generation method for the analysis of a one-month trace of Google trace. Dong et al. [16] proposed the most efficient server first (MESF) task scheduling scheme to minimize the energy consumed by data-center servers though analysis of Google cluster trace. Reiss et



(a) average CPU utilization(%) (b) average memory utilization(%) (c) average disk utilization(%)

Figure 13. CDF of average CPU utilization(%), memory utilization(%) and disk utilization(%) for online service groups.

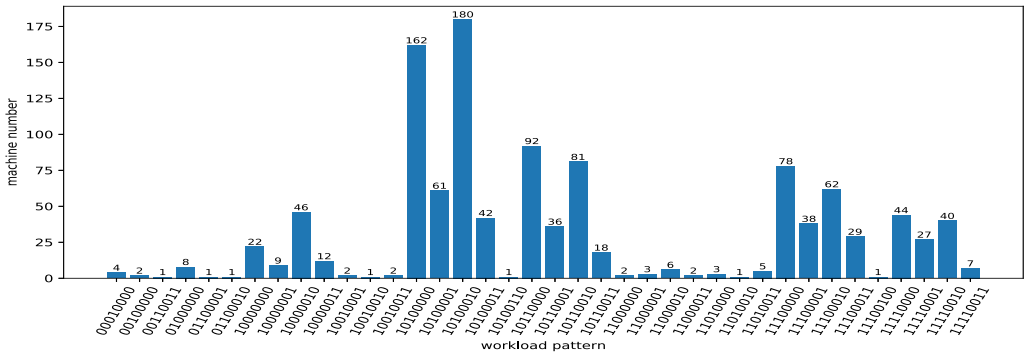


Figure 14. Machine distribution for online service groups pattern at 43200s.

al. [17] analyzed the workloads features and find out that workload characteristic is heterogeneity in resource types (CPU cores, memory and executed duration) and their usage, some tasks are constrained to some kind of machine types they can use. Zhang et al. [18] suggested to model task usage shapes using the mean values of run-time tasks resource consumption. One traditional solution to improve the system resource utilization is scheduling, which throws all jobs to a process pool and relies on a scheduler to schedule the jobs [19].

Meanwhile, there are some studies on clustering workloads on trace dataset in order to identify job groups. Inspired by text categorization, the ”Bag of Words” model was proposed and a job clustering method called ”Bejo” was constructed [20] by using SVM. Rasheduzzaman et al.[21] discussed statistical summary of the data, and performed K-Means clustering to identify common groups of job. Chen et al. [9] offered a statistical profile of the data with several interesting discoveries regarding job arrival patterns, CPU and memory consumption, task duration, and others. They further performed K-Means clustering to cluster jobs and did correlation analysis between job semantics. Alam et al. [22] provided analysis of resource usage and requirements in this trace and attempted to give an insight into such kind of production trace, clustering of workload patterns and classification of jobs into different types based on K-Means clustering. Mishra et al. [10] modeled workloads of Google clusters and proposed an approach to classify tasks using K-Means clustering, represented by average resource consumption. Zhang et al. [23] made a deep analysis on the same dataset analyzed by Reiss. They proposed an approach to classify tasks, represented by task size (CPU, memory) and running time, for dynamic heterogeneity-aware resource provisioning. Di et al. [24] characterized and modeled Google applications and jobs, also, they analyzed the classification of applications using K-Means with optimized number of sets, based on task events and resource usage.

While our work is the first to cluster workloads on Alibaba trace and identify workload patterns on machines at sampling time. What’s more, we also find many interesting insights in production cloud environment.

V. Conclusions and Future Work

This paper provides a statistical profile of workload behavior and clustering based on several typical features on the Alibaba trace dataset which was released in 2017. Although previous work has some analysis on offline batch jobs and online services, our analysis provides some new discoveries, such as job groups which have similar resource utilization or executing duration, the features of every group

and which groups often run on the same machine together and so on. In the future, we plan to identify the dynamic workload characteristics which may change over time. We are interested in optimizing job scheduling and resource allocation algorithms in large-scale clouds based on the discoveries of our workload characterization.

ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (No. 2018YFB1004804), National Natural Science Foundation of China (No. 61702492, U1401258), and Shenzhen Basic Research Program (No. JCYJ20170818153016513, JCYJ20170307164747920).

REFERENCES

[1] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis lectures on computer architecture*, vol. 8, no. 3, pp. 1–154, 2013.

[2] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, and J. Wilkes, "Omega: flexible, scalable schedulers for large compute clusters," in *Proceedings of the 8th ACM European Conference on Computer Systems*. ACM, 2013, pp. 351–364.

[3] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at google with borg," in *Proceedings of the Tenth European Conference on Computer Systems*. ACM, 2015, p. 18.

[4] C. Lu, K. Ye, G. Xu, C.-Z. Xu, and T. Bai, "Imbalance in the cloud: an analysis on alibaba cluster trace," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2884–2892.

[5] "Alibaba trace," *https://github.com/alibaba/clusterdata*, 2017.

[6] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[7] K. G. Derpanis, "Mean shift clustering," *Lecture Notes*, 2005.

[8] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[9] Y. Chen, A. S. Ganapathi, R. Griffith, and R. H. Katz, "Analysis and lessons from a publicly available google cluster trace," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-95*, vol. 94, 2010.

[10] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das, "Towards characterizing cloud backend workloads: insights from google compute clusters," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 4, pp. 34–41, 2010.

[11] M. Calzarossa and G. Serazzi, "Workload characterization: A survey," *Proceedings of the IEEE*, vol. 81, no. 8, pp. 1136–1150, 1993.

[12] F. Wang, Q. Xin, B. Hong, S. A. Brandt, E. Miller, D. Long, and T. McLarty, "File system workload analysis for large scale scientific computing applications," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2004.

[13] "Google trace," *https://github.com/google/cluster-data*, 2011.

[14] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Towards understanding heterogeneous clouds at scale: Google trace analysis," *Intel Science and Technology Center for Cloud Computing, Tech. Rep*, vol. 84, 2012.

[15] Z.-W. Fan, P.-J. Huang, P.-S. Huang, L.-X. Chen, Y.-Q. Xiao, M.-X. Huo, and Y. Liang, "A feature generation framework for google trace analysis," in *Machine Learning and Cybernetics (ICMLC), 2015 International Conference on*, vol. 1. IEEE, 2015, pp. 229–234.

[16] Z. Dong, W. Zhuang, and R. Rojas-Cessa, "Energy-aware scheduling schemes for cloud data centers on google trace data," in *Green Communications (OnlineGreencomm), 2014 IEEE Online Conference on*. IEEE, 2014, pp. 1–6.

[17] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proceedings of the Third ACM Symposium on Cloud Computing*. ACM, 2012, p. 7.

[18] Q. Zhang, J. L. Hellerstein, R. Boutaba *et al.*, "Characterizing task usage shapes in googles compute clusters," in *Proceedings of the 5th international workshop on large scale distributed systems and middleware*. sn, 2011, pp. 1–6.

[19] C. Delimitrou and C. Kozyrakis, "Qos-aware scheduling in heterogeneous datacenters with paragon," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 4, p. 12, 2013.

[20] L. Xu, J. Cao, Y. Wang, L. Yang, and J. Li, "Bejo: Behavior based job classification for resource consumption prediction in the cloud," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*. IEEE, 2014, pp. 10–17.

[21] M. Rasheduzzaman, M. A. Islam, T. Islam, T. Hossain, and R. M. Rahman, "Task shape classification and workload characterization of google cluster trace," in *Advance Computing Conference (IACC), 2014 IEEE International*. IEEE, 2014, pp. 893–898.

[22] M. Alam, K. A. Shakil, and S. Sethi, "Analysis and clustering of workload in google cluster trace based on resource usage," in *IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE, 2016, pp. 740–747.

[23] Q. Zhang, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, "Harmony: Dynamic heterogeneity-aware resource provisioning in the cloud," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*. IEEE, 2013, pp. 510–519.

[24] S. Di, D. Kondo, and F. Cappello, "Characterizing and modeling cloud applications/jobs on a google data center," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 139–160, 2014.