IEEE Access
The journal for rapid open access publishing
IEEE Xplore Digital Library
http://ieeexplore.ieee.org/Xplore

Characteristics of Co-allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study from Alibaba Cloud

Congfeng Jiang[1,2] , (Member, IEEE), Guangjie Han[3], (Member, IEEE), Jiangbin Lin[4], Gangyong Jia[1,2], (Member, IEEE), Weisong Shi[5], (Fellow, IEEE), Jian Wan[1,6](Member, IEEE)

Abstract—**In order to reduce power and energy costs, giant cloud providers now mix online and batch jobs on the same cluster. Although the co-allocation of such jobs improves machine utilization, it challenges the data center scheduler and workload assignment in terms of quality of service, fault tolerance, and failure recovery, especially for latency critical online services. In this paper, we explore various characteristics of co-allocated online services and batch jobs from a production cluster containing 1.3k servers in Alibaba Cloud. From the trace data we find the following:(1) For batch jobs with multiple tasks and instances, 50.8% failed tasks wait and halted after a very long time interval when their first and the only one instance fails. This wastes much time and resources as the remaining instances are running for an impossible successful termination. (2) For online services jobs, they are clustered in 25 categories according to their requested CPU, memory, and disk resources. Such clustering can help co-allocation of online services jobs with batch jobs. (3) Servers are clustered into 7 groups by CPU utilization, memory utilization, and their correlations. Machines with strong correlation between CPU and memory utilization provides opportunity for job co-allocation and resource utilization estimation. (4) The MTBF (mean time between failures) of instances are in the interval [400, 800] seconds while the average completion time of the 99th percentile is 1003 seconds. We also compare the cumulative distribution functions of jobs and servers and explain the differences and opportunities for workload assignment between them. Our findings and insights presented in this paper can help the community and data center operators better understand the workload characteristics, improve resource utilization, and failure recovery design.**

Index Terms—**co-allocated jobs, workload characterization, online services, batch jobs, data center, scheduling.**

[1] Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou Dianzi University, Hangzhou 310018, China

[2]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China;

[3]Key Laboratory for Ubiquitous Network and Service Software of Liaoning province, School of Software, Dalian University of Technology, Dalian, 116024, China.

[4]Alibaba Group, Hangzhou 311100, China

[5]Department of Computer Science, Wayne State University, Detroit, MI, USA

[6]School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Corresponding author: Jian Wan (wanjian@zust.edu.cn)

This work is supported by Natural Science Foundation of China (61472109, 61572163, 61672200, 61602137, and 61802093), Key Research and Development Program of Zhejiang Province (No. 2018C01098) and the Natural Science Foundation of Zhejiang Province (NO. LY18F020014). This work is also supported in part by National Science Foundation (NSF) grant CNS-1205338 and CNS-1561216.

Digital Object Identifier: 10.1109/ACCESS.2019.2897898

I. Introduction

Power consumptions have become major concern for not only cloud servers but also battery-powered devices [1], [2]. In the last decade, electricity use by data centers increased significantly, due largely to explosive growth in both the number and density of data centers. It is estimated that US data centers consumed about 70 billion kilowatt hours of kilowatt hours of electricity in 2014, representing 2% of the country's total energy consumption. This is a 4% increase in total data center energy consumption from 2010 to 2014, while it was a 24% increase from 2005 to 2010[3]. The speed of energy consumption increase is decreasing thanks to the energy efficiency improvement in commercial servers and actions on energy reduction in data centers. Specifically, energy efficiency improvements are the key factor in taming the growth rate of the data center industry's energy consumption. It is estimated that data centers would have consumed 40 billion kWh more energy than they did in 2014 if the energy efficiency improvements had stagnated at the levels of 2010 [4]. Except for energy efficiency improvement in hardware, various energy aware scheduling approaches have been proposed to save power and energy consumptions in data centers [5]-[8].

The varying workload in data centers results in fluctuation in resource utilization which provides opportunity for resource multiplexing [9], [10]. For example, current giant cloud service providers co-allocate online services and batch jobs on the same clusters to increase server utilization and reduce energy costs. Therefore, workload characterization helps understand workload patterns and design better job scheduling policy in cloud data centers [11]-[13]. However, the merging of online services and batch jobs also results in scheduling complexity and interferences among online services and batch jobs, which sometimes can deteriorate system performance.

Moreover, with the advances in communications and sensor technology, wireless sensors are deployed in smart cities, smart homes, autonomous vehicles, and industrial environments[14]. These smart sensors are capable of providing complex services with diverse requirements, including data aggregation and analytics [15]. In the emerging edge computing paradigm [16], data analytics can be performed in sensor nodes to save energy consumption or privacy preserve [17]-[20]. There are also some cloud based platforms providing data analytics for edge devices, such as AWS Greengrass [21] and Microsoft

Azure IoT Edge [22]. Due to the geographical distribution of servers and incoming workloads, cloud service providers tend to adopt an over provisioning strategy to respond to intermittent burst workloads in order to ensure the quality of service for different jobs. Therefore, a carefully designed mixture of online service and batch jobs in data centers can not only provide further server consolidation, but also ensure the quality of service guarantee of tenants as well as the reduction of energy consumption in data centers.

In this paper, we analyze the Alibaba Cloud trace data of a data center with online service and batch jobs mixture distribution [23]. It is 24-hour trace data collected from a data center with 1313 servers, including resource utilization and job status. Understanding of workload characteristics is vital for workload placement and scheduling as well as quality of service provisioning. We try to answer the following questions:

1) How to schedule jobs to appropriate servers according to jobs' resource requirements and a server's resource availability.
2) How to reduce the latency of online services at a specific percentile and increase the throughput of batch jobs when they are mixed on the same cluster, and how to mitigate the resource contention on this cluster and re-adjust the workload scheduling.
3) How to provide fault tolerant scheduling according to job-server affinity in case of hardware and software failures.
4) How to simulate a real Internet Data Center (IDC) based on the workload characteristics for better scheduler design.

The remainder of this paper is organized as follows. In Section II, we describe the trace data and give some notations and terms used in this paper. In section III, we analyze the batch jobs and characterize the instance completion time, jobs and task distribution, resource utilization, and job failures. Section IV describes the analysis of online service jobs. In Section V, we provide analysis on server nodes. We summarize related work in Section VI and conclude the paper in Section VII.

II. The Analyzed Datasets

A. THE Alibaba Trace Data

The trace data, ClusterData201708, contains cluster information of a production cluster in a 24-hour period, and contains 1313 machines that run both online service and batch jobs. The exposure of this data to the public can help address the challenges large IDCs face where online services and batch jobs are co-allocated. Characterization on this trace data may provide useful insights for online service and batch jobs scheduler cooperation. It can also help tradeoff resource allocation between online services and batch jobs to balancing improved throughput of batch jobs while maintaining acceptable service quality and fast failure recovery for online service.

B. Metric Notations and Terms

For convenience, we list the notations and terms used in this paper in Table 1.

TABLE I
Terms and Notations

Terms	Explanation [a]
Online service jobs	One online job is divided into multiple instances.
Batch jobs	Batch jobs are divided into tasks and tasks are divided into instances.
JobID	Each job has a unique identifier, and a job may have multiple tasks.
Job number	The amount of jobs
TaskID	Each task has a unique identifier and each task may have multiple instances.
Task number	The amount of tasks
InstanceID	Each instance has a unique identifier and each task may have multiple instances.
Instance number	The amount of instances

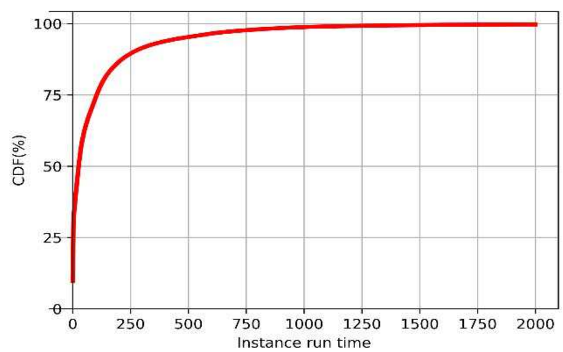


Fig. 1. CDF of batch instances completion time.

TABLE II
Percentiles of instance completion time

Percentile	50	60	80	90	99
Completion time(s)	26	44	132	260	1067

III. Batch Jobs Workload Characterization

A. Instances Completiont

Usually one batch job is divided into multiple tasks and each task executes different business logics. A task belonging to one job consists of a directed acyclic graph (DAG) due to data dependency (no DAG information is released in this dataset although they do have DAG dependencies). Instance is the smallest unit of batch job scheduling. For batch processing, all instances within a task execute the same application codes with the same resource request, but with different input data.

We give the CDF chart of completion time of all batch instances in Fig.1 and their percentiles in Table 2. The 80th, 90th, and 99th percentiles of instance completion time are 132, 260, and 1067 seconds, respectively.

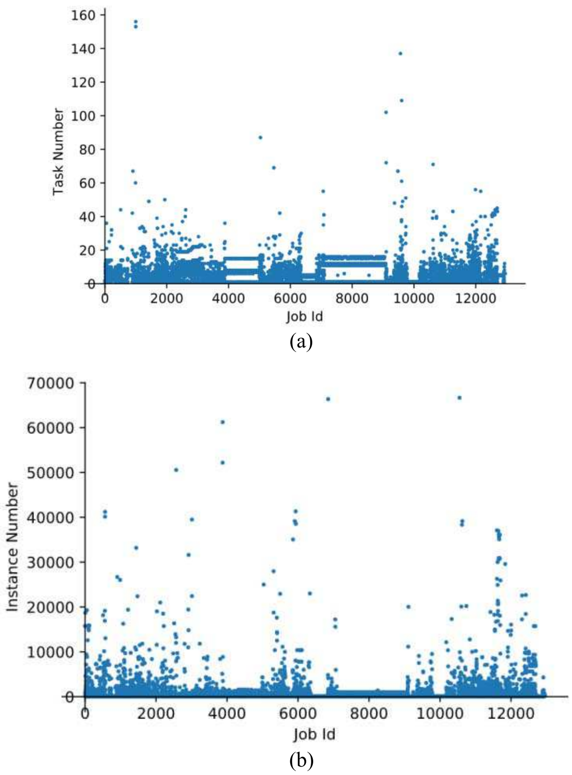


Fig. 2. Tasks and instance counts of each job.

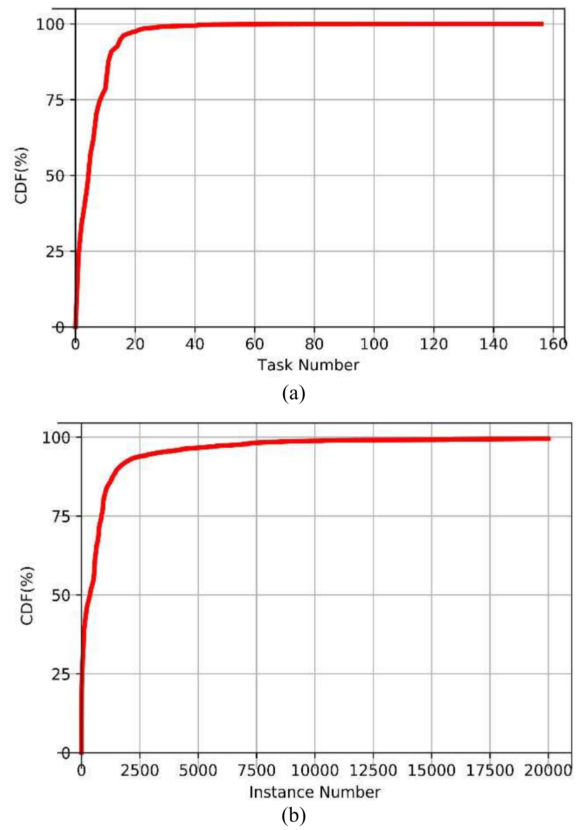


Fig. 3. CDF of tasks and instance of batch jobs.

TABLE III
PERCENTILES OF TASK NUMBER AND INSTANCE NUMBER OF JOBS

Percentile	50th	60th	80th	90th	95th	99th
Task Number	4	5	11	12	16	28
Instance Number	363	567	940	1552	3200	10749

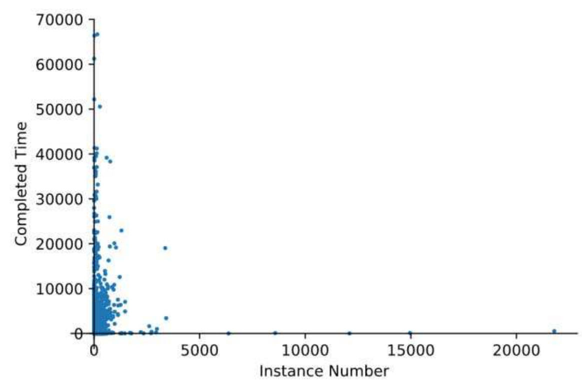


Fig. 4. Completion time of jobs with different instance numbers.

B. Jobs and Tasks Distribution

All batch jobs are divided into tasks and each task is divided into instances. In order to get the characteristics of tasks and jobs, we give their numbers in Fig.(2.a).We can see that the majority of jobs have tasks less than 40. Jobs with jobID from 4000 to 5000 and 7000 to 9000 have similar task numbers, and the possible reason is that these jobs are similar. We can also observed in Fig.(2.b) that jobs with jobID from 4000 to 5000 and 7000 to 9000 have similar instance numbers.

We present their CDF charts in Fig.3 and their percentiles in Table 3.

The task number and instance number of the 80th percentile is 11 and 940, respectively.

In a real job scheduling scenario, one practical problem is ascertaining if the job completion time is correlated to its division of tasks and instances, i.e. the job partition granularity. We give the job completion time, task number, and instance number in Fig.4 and Fig.5. The job completion time decreases significantly when the instance number increases after 1000.

From Fig.4, we can observe that most jobs has less than 1000 instance number and it's run time is quiet diversity, compare to Fig.4 and Fig.5, most task number and instance number are all less than 1000, according to data analysis we know that 99.2% job's instance number is less than 1000 and 99.4% job's task number is less than 1000.

We present the completion times of all jobs and jobs with completion times less than 1000 seconds in Fig.6

From Fig.6, we can see that jobs with an id around 10000 have significantly less completion time than do others. Investigation on these jobs may help design a better job scheduling policy to achieve shorter completion time of jobs.

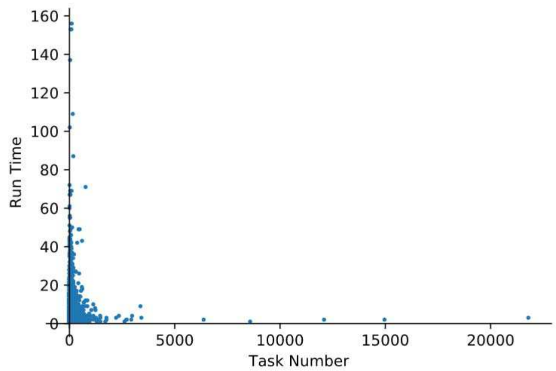
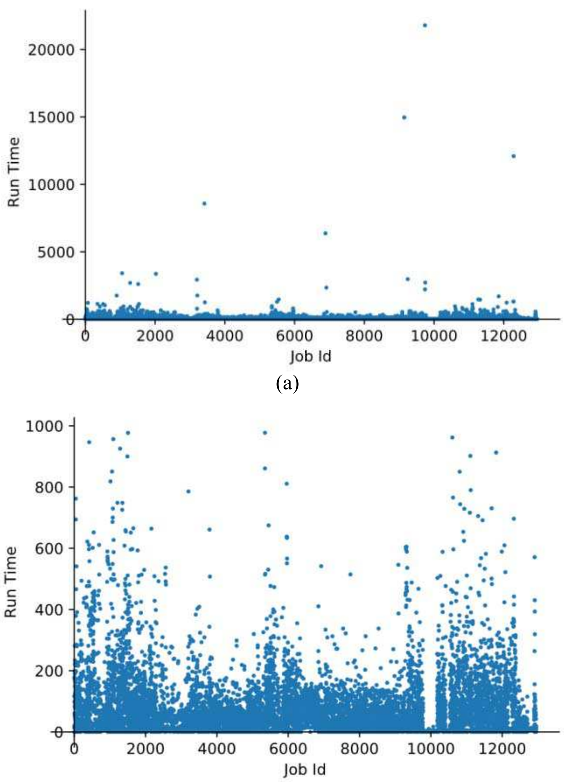


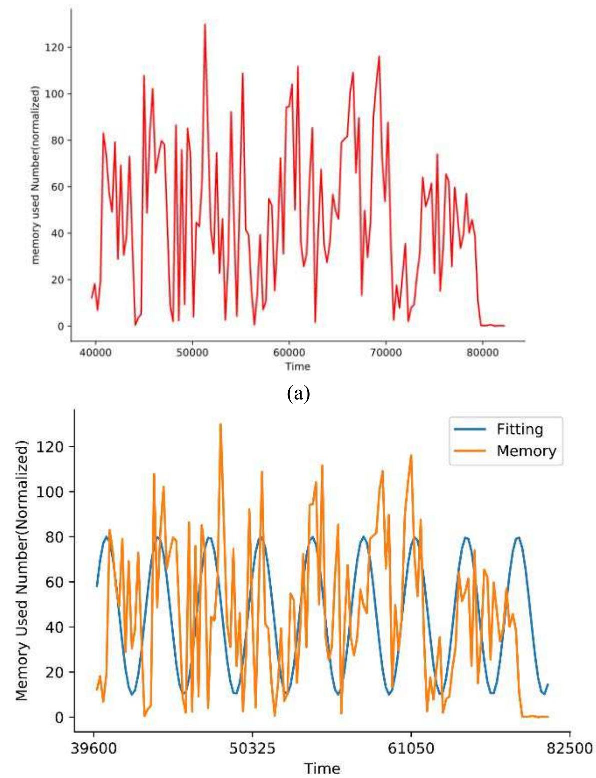
Fig. 5. Completion time of tasks with different task numbers.



(a)

(b)

Fig. 6. Job completion time.



(a)

(b)

Fig. 7. Memory utilization of batch jobs.

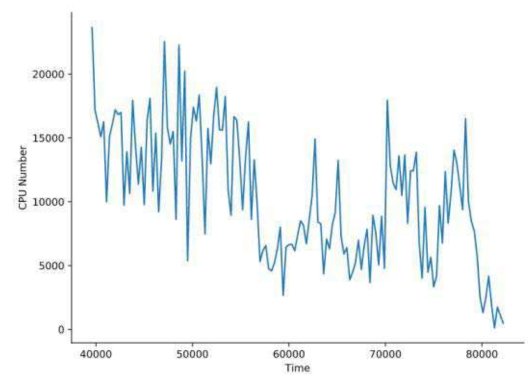


Fig. 8. CPU utilization of batch jobs.

C. Resource Utilization

Resource multiplexing and job co-allocation can significantly reduce energy consumption and increase resource utilization. Therefore, resource utilization analysis is vital for quality evaluation of such multiplexing and co-allocation. We present memory and CPU utilization in Fig.7 and Fig.8.

From Fig.7 we observe that memory utilization has higher fluctuations than CPU. Moreover, memory utilization is periodically changing during 24 hours. Therefore, job scheduling can adapt to this characteristic to smooth the memory utilization. We give the fitting curve of memory utilization in Fig.7 (b). The fitting is quantified as

$$y = 35 * (0.00129 * x - 50.828) + 45$$

where y is memory utilization and x is time.

The magnitude of the fitting sine curve is 35(%) and its period is 4870 seconds.

D. Job Failures Analysis

A robust job scheduling policy is very important for business consistence and quality service provisioning in data center operation. Among the trace data, there are 15739375 instances (excluding re-running instances) and 209168 failed instances representing a 1.32% failure rate. We list the failure statistics in Table 4 and Fig.9.

From Fig.9, we observe that the instances executed on some machines have a higher failure rate than other machines. Similarly, some tasks and jobs have higher failed instances

TABLE IV
INSTANCE FAILURE STATISTICS OF BATCH JOBS

Retry times	Failed	Total	Failed Rate	Dropped Number
1	148899	15739375	0.009%	0
2	35276	147778	23.871%	1121
3	19134	35276	54.241%	0
4	5599	19087	29.334%	47
5	175	5112	3.423%	487
6	57	172	33.14%	3
7	20	55	36.364%	2
8	7	19	36.842%	1
9	1	2	50.000%	5
10	0	1	0.000%	0

than others. For batch jobs with multiple tasks and instances, we find that among 1126 failed tasks, 572 failed tasks (representing 50.8% of total failed tasks) wait and halted after a very long time interval when their first and the only one instance of failure occurs. This wastes much time and resources as the remaining instances are running for an impossible successful termination. This also indicates that job scheduler should act more intelligently and quickly to reduce useless execution after an existing instance fails.

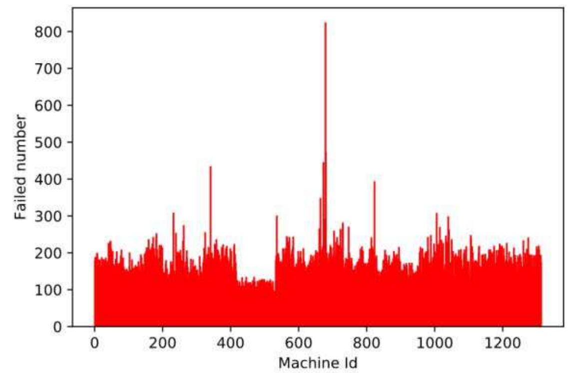
We list the top 15 machines that have the highest failed instance numbers in Table 5. These machines may have hardware related issues or software configuration incompatibility with instances running on them. Therefore, these machines should be investigated first for fault diagnostics. In Table 5, the total number of dropped instances is 1121. These 1121 instances may have been dropped manually by the system administrator on purpose, or automatically by the scheduler after instance staling, or by scheduler malfunction. Scanning on these dropped instances could provide insights on job scheduling and alleviation of these dropped instances.

In the following sections, server nodes are grouped and categorized to better understand the operating mode of the entire data center in more detail. One of the classification criteria is the number of failure instances on it.

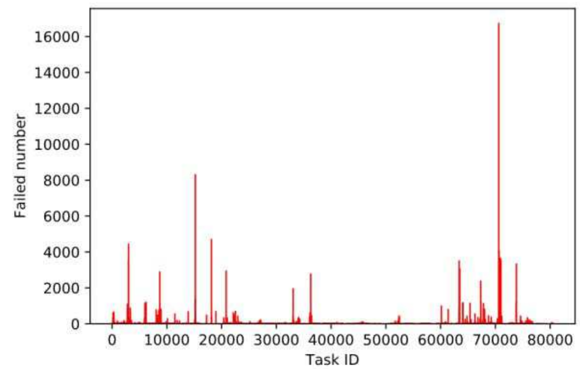
Among 148899 failed instances during the first execution of all instances, 1121 instances are dropped during the second execution. Eventually, jobs containing these 1121 instances are failed. Since these 1121 dropped failed instances leads to their parent tasks and jobs failure, we give the distribution on machines of these 1121 dropped instances in Fig.10.

From Fig.10, we observe that the majority of machines drop 0-5 failed instances, but some machines drop more than 10 instances.

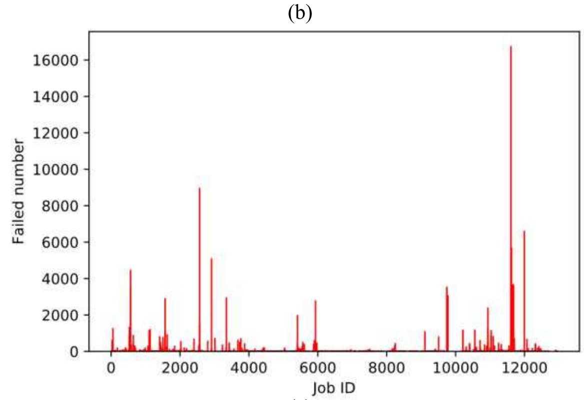
If an instance failed during execution but another instance is not notified of such failure by the scheduler and continues execution, their parent task, or job may fail eventually after extra prolonged execution. However, this can result in waste on resource usage and performance degradation or violation on service level agreements. In order to derive a temporal pattern of instance jobs, we give the failed jobs distribution with time in Fig.11.



(a)



(b)



(c)

Fig. 9. Instance failures on machines, tasks and jobs.

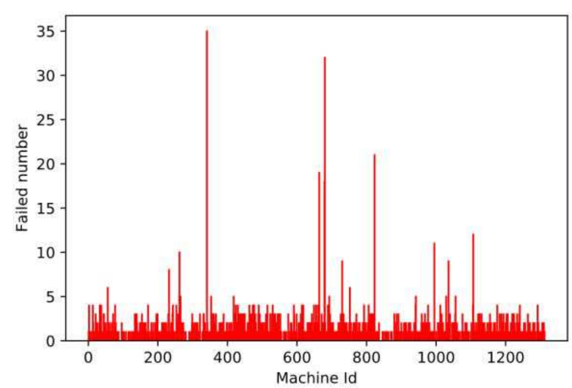


Fig. 10. Distribution of 1121 dropped failed instances after first execution.

TABLE V
TOP 15 SERVERS WITH THE HIGHEST FAILED INSTANCES

Machine Id	Failed Instance Number	Dropped Instances After 1st Execution
679	823	18
680	471	32
673	444	1
341	433	35
823	392	21
664	347	19
232	307	8
1006	306	2
536	299	1
1040	297	3
676	289	2
730	280	9
262	273	10
747	269	0
1015	268	0

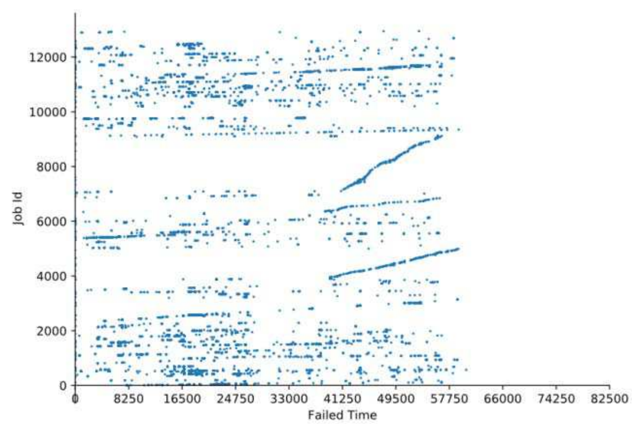


Fig. 11. Temporal distribution of job failures.

From Fig.11, we observe that:

1) There are no longer failures after the timestamp of 60352;
2) Jobs with jobID in [4000, 5000] and [7000, 9000] do not fail before timestamp. These jobs failed between timestamp 40000 and 60000, but the failed jobs have consecutive jobID along with time. This may be caused by some cascading job failures.

IV. ONLINE SERVICES JOBS WORKLOAD CHARACTERIZATION

A. Instances Clustering

In the Alibaba cloud data center, online services, and batch jobs are co-allocated on the same cluster. The online services are represented by instances. In this trace data, there are 11101 instances among which 10980 instances are created before the trace data sampling and only 193 instances are created during the trace data sampling.

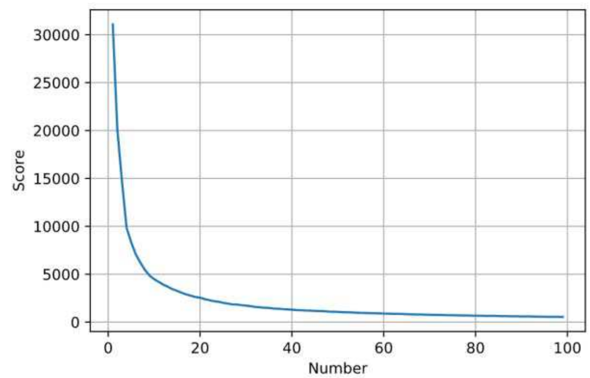


Fig. 12. Clustering performance and clustering number on resource allocation of online instances.

Typically, an instance will be allocated with memory, CPU, and disk capacity. CPU allocation is represented by the number of CPU cores such as 1, 2, 4, 8, or 16. Memory allocations are normalized values as 0.002651, 0.042430, 0.053012, 0.084819, 0.127228, 0.169637, 0.254456, 0.318000, 0.999963, 1.00002, 1.000002, or 1.000010. Disk allocations are also normalized values as 3.17e-11, 0.000142, 0.034085, 0.045446, 0.056808, 0.068170, 0.085212, and 0.113617. Thus, there are $5*13*8=520$ combinations of resource allocation and 107 CPU SET allocations. Although there are 520 combinations of resource allocation, we use K-means clustering on the resource allocation and find that the clustering performance is the best when we choose 25 as the clustering number as shown in Fig.12. (Please note that grouping resource allocation configurations into more than 25 categories is possible with lower clustering scores in our experiments.) The clustering results are listed in Table 6. The clustering results in Table 6 helps estimate resource allocation for incoming online instances and make better scheduling of these online instances to appropriate cluster nodes with available resources.

From Table 6, we know that majority instance has a low CPU usage rate and a higher memory usage rate, according to that we can add some task that has a higher CPU usage rate and lower memory usage rate. From my data analysis, 82.58% instances take less than 20th percentile disk capacity.

B. Resource Utilization

We list the sum of all online instances' CPU and memory utilization in Fig.13 and Fig.14. From Fig.13 and Fig.14 we can observe that memory used curve is more stable than CPU usage. We also observe that at both timestamp 50100 and 54300, there are two sharp declines in both CPU and memory utilization. This may represent service interruption or unavailability at these two spots. We can quickly diagnose and locate such exception of online services through real time system monitoring or log streaming processing.

TABLE VI
CLUSTERING OF RESOURCE ALLOCATION OF ONLINE INSTANCES

Category	CPU	Memory	Disk	Number
1	4.36	40.02	14.27	1151
2	3.89	24.69	6.25	1144
3	3.47	29.86	11.80	778
4	4.31	36.98	6.81	706
5	5.81	47.55	9.97	680
6	5.29	56.19	18.17	552
7	5.24	60.04	9.21	531
8	11.16	44.16	11.35	487
9	12.11	59.35	10.26	454
10	5.21	69.08	13.83	372
11	22.84	39.43	12.45	348
12	9.39	54.66	26.39	325
13	11.36	45.82	20.55	323
14	6.39	15.30	7.74	299
15	14.62	28.21	7.18	278
16	20.38	62.78	9.88	276
17	12.08	41.06	41.52	275
18	36.28	62.65	12.51	274
19	24.80	60.83	30.88	259
20	36.89	64.30	21.93	241
21	12.55	69.36	16.46	224
22	6.58	24.00	25.81	131
23	9.29	57.22	45.34	117
24	6.66	28.74	46.13	113
25	10.46	59.68	76.59	21

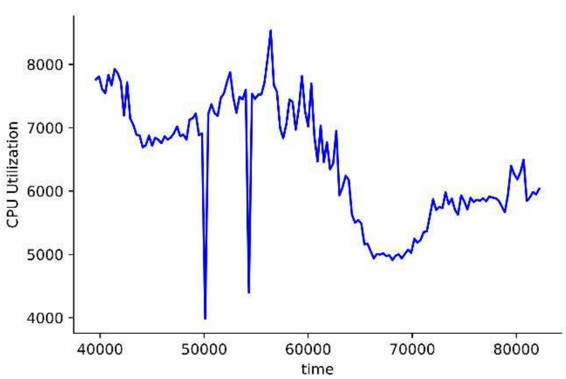


Fig. 13. CPU utilization of online services instances.

V. NODES USAGE ANALYSIS AND CHARACTERIZATION

A. Machine Clustering

Good understanding of workload characteristics is vital for coordinated job scheduling of online and batch jobs in data centers. In previous sections, we analyzed the jobs characteristics. In this section, we present the analysis of machine characteristics in the clusters.

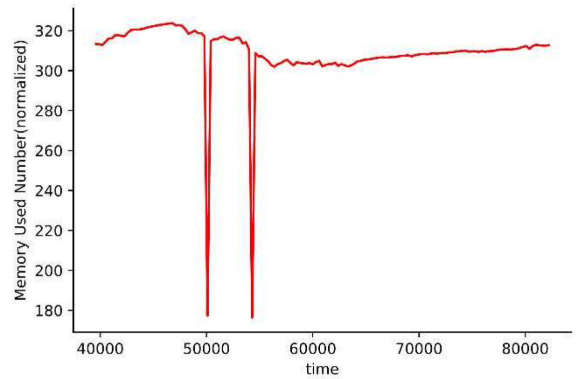


Fig. 14. Memory utilization of online services instances (sum of normalized data).

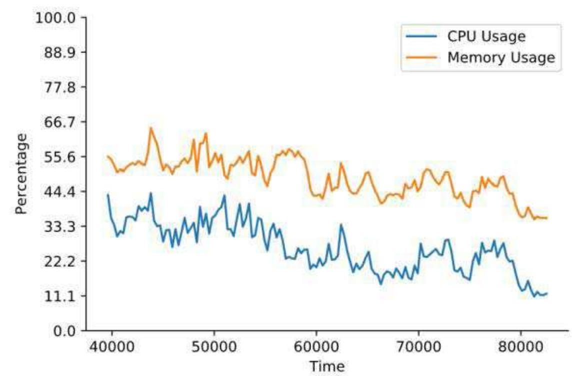


Fig. 15. CPU and memory utilization of the whole cluster.

We list the summarized CPU and memory utilization of all servers in the data center in Fig.15. The whole cluster's CPU utilization is between 13% and 40% and memory utilization is between 42% and 65%.

The co-allocation of online services and batch jobs on the same cluster makes it challenging for resource multiplexing and service provisioning. For example, online services have a rigorous latency requirement for jobs of the 99.9th percentile; therefore, existing resource utilization of each machine is an important reference before jobs co-allocation on dedicated machines. Therefore, we present the clustering results of all the machines in the whole cluster in Table 7. We observe that type A machines are the majority of the cluster and they have 27.5% CPU utilization and 50.67% memory utilization on average.

We list the CPU and memory utilization of each type of machine in Fig.16 including CPU usage and memory usage. According to its different curve shape classification, a total of seven categories are shown in the following figure. Table 7 is shown in Fig.16. The seven types are counted, the proportion of different categories, the average CPU and memory usage, and the median and median usage of CPU and memory. Type A is the largest proportion, reaching 84.77% of the total, and its curve is also an approximate straight line of low decline. Other categories have some abnormal curves appearing at a certain time or all time, and the overall graph calculated is

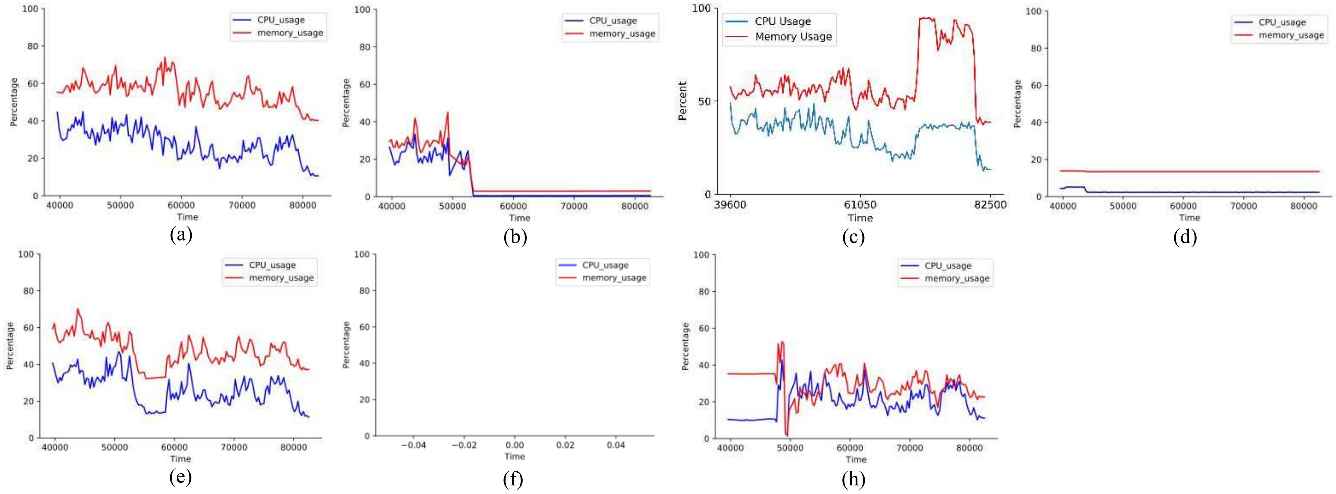


Fig. 16. CPU and memory utilization of all machine types.

very similar with Fig.(16.a). Combining the graph and the table, we know that the average memory and CPU of the Type B and Type C node are lower than those of the Type A, but the variance is higher than that of the Type A. It can be considered that the task mode on the Type B and Type C node is more complicated and needs some more dedicated task scheduling. In Fig.16, for type B machines, they become idle after timestamp 52000. These 154 type B machines may be scheduled for predefined maintenance or be out of service due to some reasons. If they are just idle and can serve incoming requests, it is not a good idea to let them idle for such a long time and they can be purposely powered off.

For type C machines, they have similar patterns on CPU and memory utilization, especially when there is a burst at timestamp 70000 and the burst lasts for about 10000 seconds. These 21 type C machines use almost 100% CPU during the burst of workload. The jobs on them should be migrated to other machines to keep the machine in a relatively lower CPU utilization for enough quality of service guarantee.

Seventeen type D machines seem idle during the sampling period of this trace data, which means these machines are idle, or out of service. Type E, F, and G machines are too few to represent any patterns. We give the difference of CPU utilization and memory utilization of seven types of machines in Fig.17.

B. Correlation Analysis of CPU and Memory Utilization

In order to quantify the interference of online service jobs and batch jobs, we categorize the machines into 3 types in Table 8 and provide the correlations of CPU and memory utilization of each machine in Fig.18. We illustrate the CPU and memory utilization of these three subsets of machines in Fig.19. On most of the machines, their CPU and memory utilization have a positive correlation. With the job type and instance distribution, we can infer the correlation of co-allocation of online and batch job scheduling and estimate their CPU and memory utilization after co-allocation.

We plot the correlation coefficients of CPU and memory utilization on each machine in Fig.20. For Type #2 machines

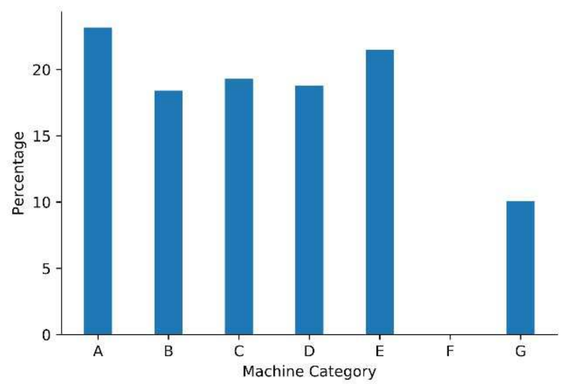


Fig. 17. The differences of CPU utilization and memory utilization of 7 types of machines.

where CPU and memory utilization have strong positive correlations, we identify these machines with machineId, i.e. machines 88-127, machine 261-296, and machines 830-906. They are probably three small clusters containing subsets of all 1313 machines that execute the same jobs with their neighbor machines. For these machines, machine utilization can be easily estimated if more jobs of the same type are scheduled to them. Specifically, if we schedule these machines to newly added machines in the cluster, we can also estimate the dedicated machines' utilization. Moreover, if we pack more computing or memory intensive jobs to the existing machines, the following resource utilization can also be estimated according to the correlation between CPU and memory utilization. Therefore, the correlation of CPU and memory utilization can help the co-allocation and consolidation of online services and batch jobs in data centers.

C. MTBF of Instances on Machine

As presented in the previous section, instances on some machines are more prone to failure than other machines, for example, machine #679, #680, #341, and #673. On the contrary, there are some machines where the instances on

TABLE VII
MACHINE CLUSTERING BY CPU AND MEMORY UTILIZATION

Type	Count	Percentage	Avg.cpu_usage	Avg.mem_usage	Med.cpu_usage	Med.mem_usage	Std.cpu_usage	Std.mem_usage
A	1113	84.77%	27.5	50.67	27.2	52.62	9.76	12.43
B	154	11.73%	21.11	39.52	15.04	35.7	13.26	11.63
C	21	1.60%	25.62	44.91	23.31	43.03	13.54	18.25
D	17	1.30%	9.26	28.04	9.71	30.5	6.62	12.82
E	4	0.30%	24.75	46.25	24.42	46.33	9.06	8.39
F	3	0.23%	0	0	0	0	0	0
G	1	0.08%	19.58	29.66	18.98	29.52	0	0

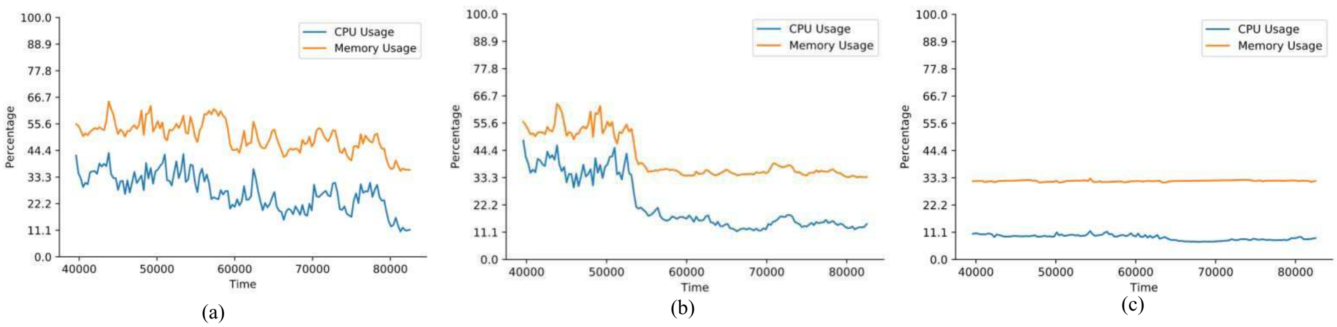


Fig. 18. CPU and memory utilization clustering by correlation coefficients

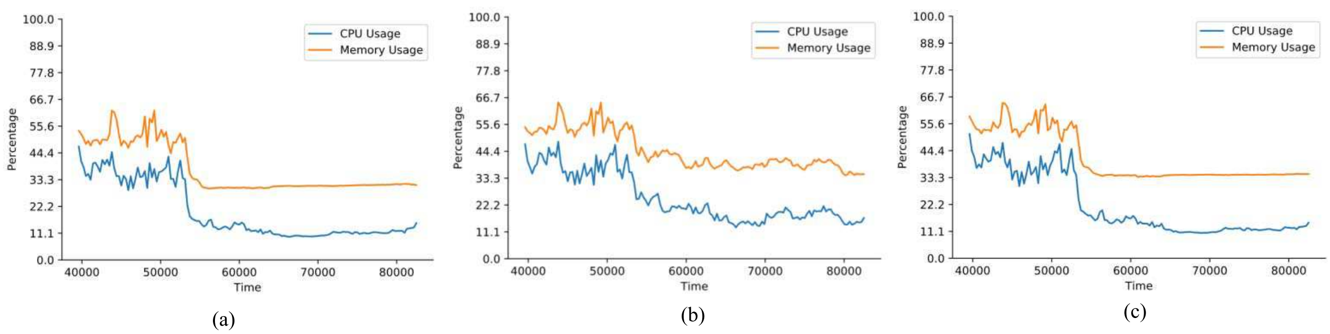


Fig. 19. Resource utilization of three subsets of machines with strong correlation coefficients between CPU and memory

them never fail, for example machine #167, #478, and #1075. Comparison of these machines can provide useful insights and directions for diagnosing hardware or software related issues to avoid instance failures. We calculate the MTBF (mean time between failures) of instances on all machines as depicted in Fig.21. Since different machines have different failure patterns, we do not give the MTBF numbers on each machine. Alternatively, we give the MTBF range on most machines. On most of the machines, the MTBF of instances is between 400 seconds to 800 seconds. The MTBF of instances is very important in 3 aspects:

1) It should be a consideration to determine if it is worthy to schedule jobs to these machines, especially for jobs that require a high success rate or long execution time.

2) It helps estimate when and how many jobs will fail in the near future, or migrate instances to other machines to avoid failures.

3) It is a good reference for scheduled machine maintenance or patching if it is out of service.

VI. RELATED WORK

Workload characterization is the essential process for data center operators to identify system bottleneck and figure out solutions for optimizing performance. Single server workload characterization has been studied extensively. Cortez *et al.* [24] present a detailed characterization of several VM workload behaviors from Microsoft Azure. They analyze the key characteristics of the workloads by VMs' lifetime, deployment size, and resource consumption. Guo *et al.* [25] divide the workload into two categories: delay-sensitive interactive workload and delay-tolerant batch workload, and they present Stochastic Cost Minimization Algorithm (SCMA). In the time-sensitive

TABLE VIII
MACHINE CLUSTERING CPU AND MEMORY UTILIZATION

Type	1	2	3
count	1109	198	6
Corre.Coeff	0~0.85	>=0.85	<0
Avg.cpu_util (%)	27.11	22.87	25.90
Avg.mem_util(%)	50.11	42.39	49.07
Med.cpu_util(%)	26.86	18.88	25.98
Med.mem_util(%)	52.26	39.28	49.70

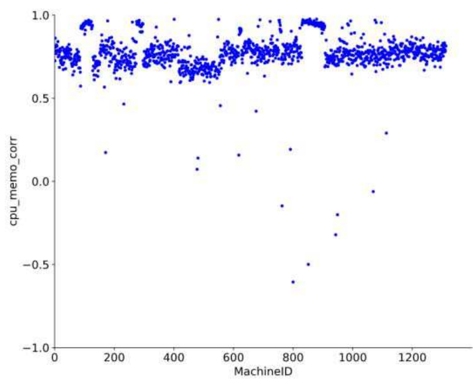


Fig. 20. correlation coefficients of CPU and memory utilization on each machine.

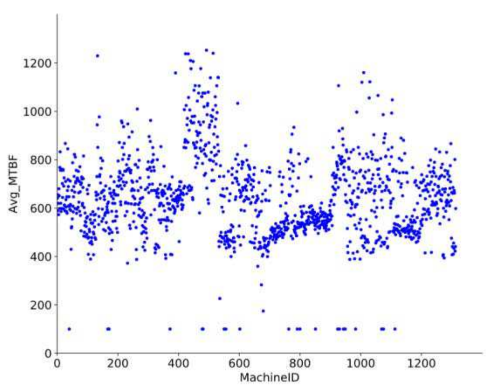


Fig. 21. MTBF of instances on machines.

geographically distributed data centers with renewable generation [26], the authors adopt a green energy prediction to schedule mixed batch and service jobs in data centers. Likewise, some researchers show the data analysis workloads are significantly diverse in terms of both speedup performance and micro-architectural characteristics [27]. Chen *et al*. [28] and Chong *et al*. [29] argue that workload characterization is important to optimize the workload management in data centers, and they can dynamically make decisions with knowledge of the newly emerging workload. Wu *et al*. [30] propose Dynamo-a data center-wide power management system. Fiandrino *et al*. [31] propose a framework of new metrics able to assess performance and energy efficiency of cloud computing communication systems. In order to limit peak power costs without any workload performance degradation, Aksanli *et al*. [26], [32] adopt a battery-based peak shaving method, and Liu *et al*. [33] develop two algorithms for data centers.

Server consolidation and VM migration are typical methods that improve data center resource utilization efficiency. Ahmad *et al*. [34] analyze and compare the current VM migration and servers consolidation framework. Varasteh *et al*. [35] survey virtual machine migration and server consolidation, the parameters and algorithmic approaches used to consolidate VMs onto PMs. Mastroianni *et al*. [36] present ecoCloud, a self-organizing and adaptive approach for the consolidation of VMs on two resources, namely CPU and RAM, in order to limit the number of VM migrations and server switches. Meanwhile, others focus on the Xen and VMware virtualization platforms [37]. In addition, since consolidating different applications may lead to a drop in performance, Chen *et al.* [38], the authors develop a light-weight, non-intrusive methodology to achieve application-centric performance targets, while consolidating homogeneous and heterogeneous application.

There is significant work on resource allocation in data centers. Sun *et al.* [39], present an overview on different kinds of resource management mechanisms for data centers. Tan *et al*. [40] and Mazumdar *et al.* [41] propose various methods for analyzing resource usage and modeling resource usage patterns. In addition, to maintain high resource utilization, new resource allocation strategies have been proposed for CPU and memory [42], [43]. Warneke *et al*. [44] propose an approach to improve memory utilization. Shojafar *et al.* [45] and Bari *et al.* [46] propose approaches to dynamically reconfigure the computing-plus-communication resources of networked data centers to improve resource utilization. Reiss *et al.* [47] analyzed the google trace data, and their result is helpful for resource schedulers. Besides, researchers present an analytical model, based on stochastic reward nets (SRNs), which can set the data center parameters under different working conditions [48].

There are some works in literature on predicting workload behaviors. These works predict resource demand, resource utilization, or job/task length for provisioning or scheduling purposes [49], [50]. They propose many methods to estimate the future need of applications in terms of resources and allocate them in advance, releasing them once they are not required. However, it is not enough to ensure an efficient data center because some servers may fail due to some reasons, and prior work proposes several solutions. Tiranee *et al.* [51] use ARMA (Auto Regressive Moving Average) and Fault Tree Analysis to predict online failure. Sedaghat *et al.* [52] present a statistical model for job reliability in a cloud data center, in the presence of stochastic and correlated failures. Itani *et al.* [53] propose a solution for node failures.

Although there are research efforts on traditional workload characterization, server consolidation and VM migration in data centers, workload characterization on co-allocated jobs in data centers is rare. There are some works on analysis of Alibaba's trace data [54]-[59]. They focus on imbalance phenomena in the cloud. In this paper, we characterize the contemporary IDCs with co-allocation of online services and batch jobs in the literature in new dimensions including failure patterns and correlations among CPU and memory. Our findings in this paper can help the data center operators better understand the workload characteristics and implement workload driven job scheduling and workload placement.

VII. Conclusions

The contemporary giants of cloud service providers co-allocate online services and batch jobs on the same clusters to increase machine utilization and reduce energy cost. However, the mixture of online services and batch jobs also result in scheduling complexity and interferences among online services and batch jobs. Moreover, rigorous latency control for online services limits the resource multiplexing between online services and batch jobs. Good knowledge of pioneered operating of IDCs that co-allocate online services and batch jobs can help the community build a more robust fault tolerant scheduler for IDCs.

In this paper, we analyze various characteristics of co-allocated online services and batch jobs from a production cluster in Alibaba Cloud. We present detailed analysis on batch instances completion time, resource utilization, failure distribution, correlation and interference between resource, and machine operating characteristics. Our findings and insights presented here can help a data center operator better understand the workload characteristics, improve resource utilization, and failure recovery capabilities.

References

[1] C. Jiang, Y. Wang, D. Ou, B. Luo, W. Shi, "Energy proportional servers: Where are we in 2016?," in *Proc. IEEE 37th Int. Conf. Dist. Comp. Sys.(ICDCS)*, Atlanta, GA,USA,2017,pp.1649-1660.

[2] G. Han, X. Yang, L. Liu, W. Zhang, "A joint energy replenishment and data collection algorithm in wireless rechargeable sensor networks," *IEEE Internet Things J.,* vol.5, no.4, pp.2596-2604, 2018.

[3] A. Shehabi, S. J. Smith, D. A. Sartor, R. E. Brown, M. Herrlin, J. G. Koomey, E. R. Masanet, N. Horner, I. L. Azevedo, W. Lintner, "United States data center energy usage report," Lawrence Berkeley National Laboratory, Berkeley, CA,USA, Rep. No. LBNL-1005775, 2016

[4] H. Fuchs, A. Shehabi, M. Ganeshalingam, L. Desroches, B. Lim, K. Roth, A. Tsao, "Characteristics and energy use of volume servers in the United States," Lawrence Berkeley National Laboratory, Berkeley, CA,USA, Rep. No. LBNL-2001074, 2017.

[5] M. A. Khoshkholghi , M. N. Derahman, A. Abdullah, S. Subramaniam ,M. Othman, "Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers," *IEEE Access*, vol. 5, pp.10709-10722,2017.

[6] C. Jiang, Y. Wang, D. Ou, Y. Qiu, Y. Li, J. Wan, B. Luo, W. Shi, C. Cerin, "EASE: Energy efficiency and proportionality aware virtual machine scheduling," in *Proc. 30th Int. Symp. Comp. Arch. High Perf. Com. (SBAC-PAD)*, Lyon, France, 2018.

[7] Y. Qiu, C. Jiang, Y. Wang, D. Ou, Y. Li, J. Wan, "Energy aware virtual machine scheduling in data centers," *Energies*, 2019, 12(3).

[8] S. Wang, Z. Qian, J. Yuan, I. You, "A DVFS based energy-efficient tasks scheduling in a data center," *IEEE Access*, vol.5, pp.13090-13102, 2017.

[9] C. Jiang, W. Huang, Z. Ren, Y. Li, J. Wan, F. Cao, J. Lin, "Towards building a scalable data analytics system on clouds: An early experience on AliCloud," in *Proc. IEEE 11th Int. Conf. Cloud Com(CLOUD)*. San Francisco, CA, USA, 2018, pp.891-895.

[10] Z. Ren, J. Wan, W. Shi, X. Xu, M. Zhou, "Workload analysis, implications, and optimization on a production Hadoop cluster: A case study on Taobao," *IEEE Trans. Serv. Com.*,vol.7,no. 2, pp.307-321, 2014.

[11] Z. Zhou, J. H. Abawajy, F. Li, Z. Hu, M. U. Chowdhury, A. Alelaiwi , K. Li, "Fine-grained energy consumption model of servers based on task characteristics in cloud data center," *IEEE Access*,vol.6,pp.27080 -27090,2018.

[12] X. Cao, Y. Zhong, Y. Zhou ,J. Wang ,C. Zhu ,W. Zhang, "Interactive temporal recurrent convolution network for traffic prediction in data centers," *IEEE Access*, vol.6,pp.5276 – 5289,2018.

[13] Z. Ren, W. Shi, J. Wan, F. Cao, J. Lin, "Realistic and scalable benchmarking cloud file systems: Practices and lessons from AliCloud," *IEEE Trans. Para. Dist. Syst*., vol.28, no.11, pp.3272- 3285, 2017.

[14] G. Han, L. Liu, W. Zhang, S. Chan, "A hierarchical jammed-area mapping service for ubiquitous communication in smart communities, *IEEE Comm. Mag.*, vol.56, no.1, pp.92-98, 2018.

[15] S. Deng, Z. Xiang, J. Yin, J. Taheri, A. Y. Zomaya, "Composition-Driven IoT Service Provisioning in Distributed Edges," *IEEE Access*,vol.6, pp.54258-54269,2018.

[16] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J*., vol.3, pp.637-646,2016.

[17] C. Jiang, T. Fan, Y. Qiu, H. Wu, J. Zhang, N. Xiong, J. Wan, "Inter-domain I/O Optimization in Virtualized Sensor Networks," *Sensors*, vol.18,no.12, art. no.4395, 2018

[18] G. Han, H. Wang, M. Guizani, S. Chan, W. Zhang, "KCLP: A k-means cluster-based location privacy protection scheme in WSNs for IoT," *IEEE Wireless Comm. Mag.*, vol.25, no.6, pp.84-90, 2018.

[19] C. Jiang, Y. Wang, D. Ou, Y. Li, J. Zhang, J. Wan, B. Luo, W. Shi, "Energy efficiency comparison of hypervisors," *Sustain. Comput.: Inform. Syst.* (2017), doi:10.1016/j.suscom.2017.09.005

[20] G. Han, H. Wang, J. Jiang, W. Zhang, S. Chan, "CASLP: A confused arc-based source location privacy protection scheme in WSNs for IoT," *IEEE Comm. Mag.*,vol.56, no.9, pp.42-47, 2018.

[21] https://aws.amazon.com/greengrass

[22] https://azure.microsoft.com/en-us/services/iot-edge

[23] The Alibaba ClusterData201708 trace data https://github.com/alibaba/clusterdata.

[24] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, R. Bianchini, "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms," in *Proc. 26th Symp. Operating Syst. Prin. (SOSP)*. ACM, 2017.

[25] Y. Guo, Y. Gong, Y. Fang, P. Khargonekar, X. Geng, "Energy and network aware workload management for sustainable data centers with thermal storage," *IEEE Trans. Para. Dist. Syst.* vol.25, no. 8, pp. 2030-2042, 2014.

[26] B. Aksanli , J. Venkatesh , L. Zhang , T. Rosing, "Utilizing green energy prediction to schedule mixed batch and service jobs in data centers," in *Proc. the 4th Workshop Power-Aware Comp. Syst.*, Cascais, Portugal ,2011.

[27] Z. Jia, L. Wang, J. Zhan, L. Zhang, C. Luo, "Characterizing data analysis workloads in data centers," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC),* Portland, Oregon, USA, 2013, pp.66-76.

[28] T. Chen, X. Wang, G. B. Giannakis, "Cooling-aware energy and workload management in data centers via stochastic optimization," *IEEE J. Sel. Topics Signal Proc.*, vol.10, no.2, pp. 402-415, 2016.

[29] F. T. Chong, M. J. R. Heck, P. Ranganathan, A. A. M. Saleh, H. M. G. Wassel, "Data center energy efficiency: Improving energy efficiency in data centers beyond technology scaling," *IEEE Design & Test,* vol.31, no.1, pp. 93-104, 2014.

[30] Q. Wu, Q. Deng, L. Ganesh ,C. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, Y. J. Song, "Dynamo: Facebook's data center-wide power management system," *in Proc. ACM/IEEE 43rd Annual Int. Symp. Comp. Arch. (ISCA)*, Seoul, South Korea, 2016.

[31] C. Fiandrino, D. Kliazovich, P. Bouvry, A. Y. Zomaya, "Performance and energy efficiency metrics for communication systems of cloud computing data centers," *IEEE Trans. Cloud Com.*, vol.5, no.4,pp. 738-750, 2017.

[32] B. Aksanli, T. Rosing, "Providing regulation services and managing data center peak power budgets," in *Proc. Design, Auto. Test Europe Conf. Exhib. (DATE)*, Dresden, Germany, 2014, pp.1-4.

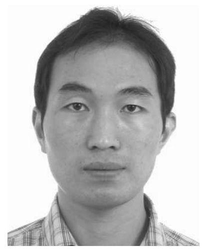
[33] Z. Liu, "Data center demand response: Avoiding the coincident peak via workload shifting and local generation." *Perf. Eval.*, vol.70, no.10, pp. 7870-791, 2013.

[34] R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai, F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," *J. Net. Comp. App.,* vol.52, pp.11-25, 2015.

[35] A. Varasteh, M. Goudarzi, "Server consolidation techniques in virtualized data centers: A survey," *IEEE Systems J.*, vol.11, no.2, pp.772-783, 2017.

[36] C. Mastroianni, M. Meo, G. Papuzzo, "Probabilistic consolidation of virtual machines in self-organizing cloud data centers," *IEEE Trans. Cloud Com.*, vol.1, no.2, pp.215-228, 2013.

[37] L. Lu, E. Smirni, "Effective resource and workload management in data centers," in *Proc. IEEE Net. Oper. Manag. Symp. (NOMS)*, Krakow, Poland, 2014, pp.1-7.
[38] L. Y. Chen, G. Serazzi, D. Ansaloni, E. Smirni, W. Binder, "What to expect when you are consolidating: effective prediction models of application performance on multicores," *Clus. Com.*, vol.17, no.1, pp. 19-37, 2014.
[39] X. Sun, N. Ansari, R. Wang, "Optimizing resource utilization of a data center," *IEEE Comm. Surveys & Tutorials,* vol.18, no.4, pp. 2822-2846, 2016.
[40] J. Tan , P. Dube, X. Meng, L. Zhang, "Exploiting resource usage patterns for better utilization prediction," in *Proc. 31st Int. Conf. Dist. Com. Syst. Workshops (ICDCSW)*, Minneapolis, MN, USA, 2011, pp.14-19.
[41] S. Mazumdar, A. S. Kumar, "Statistical Analysis of a Data Centre Resource Usage Patterns: A Case Study," in *Proc. Int. Conf. Com. Comm. Syst.*, Singapore, 2018.
[42] A. Wolke, B. Tsend-Ayush, C. Pfeiffer, M. Bichler, "More than bin packing: Dynamic resource allocation strategies in cloud data centers," *Info. Syst.*, vol.52, pp.83-95, 2015.
[43] D. Lo, L. Cheng, R. Govindaraju, P. Ranganathan, C. Kozyrakis, "Heracles: Improving resource efficiency at scale," In *Proc. 42nd Annual Int. Symp. Comp. Arch. (ISCA)*, Portland, Oregon, USA, 2015, pp.450-462.
[44] D. Warneke, C. Leng, "A case for dynamic memory partitioning in data centers," In *Proc. 2nd Workshop Data Analy. Cloud(DanaC)*. New York, NY, USA, pp.41-45, 2013.
[45] M. Shojafar, N. Cordeschi, D. Amendola, E. Baccarelli , "Energy-saving adaptive computing and traffic engineering for real-time-service data centers," In *Proc. IEEE Int. Conf. Comm. Workshop (ICCW)*, London, UK, 2015.
[46] Md. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, Md G. Rabbani, Q. Zhang , M. F. Zhani, "Data center network virtualization: A survey," *IEEE Comm. Surv. & Tutor.*, vol.15, no.2, pp.909-928, 2013.
[47] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," In *Proc. 3rd ACM Symp. Cloud Com.*, San Jose, CA, USA, 2012.
[48] D. Bruneo, "A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems," *IEEE Trans. Para. Dist. Syst.*, vol.25, no.3, pp.560-569, 2014.
[49] A. Khan, X. Yan, S. Tao, N. Anerousis, "Workload characterization and prediction in the cloud: A multiple time series approach," In *Proc. IEEE Net. Oper. Manag. Symp.(NOMS)*, Maui, HI, USA, 2012.
[50] R. N. Calheiros, E. Masoumi, R. Ranjan, R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," *IEEE Trans. Cloud Com.*, vol.3, no.4, pp. 449-458, 2015.
[51] T. Chalermarrewong, T. Achalakul, S. C. W. See, "Failure prediction of data centers using time series and fault tree analysis," In *Proc. IEEE 18th Para. Dist. Syst. (ICPADS)*, Singapore, 2012, pp.794-799.
[52] M. Sedaghat, E. Wadbro, J. Wilkes, S. De Luna, O. Seleznjev, E. Elmroth, "Diehard: reliable scheduling to survive correlated failures in cloud data centers," In *Proc. IEEE Int. Symp. Clus. Cloud Grid Com. (CCGrid)*, Cartagena de Indias, Colombia, 2016, pp.52-59.
[53] M. Itani, S. Sharafeddine, I. Elkabbani, "Practical single node failure recovery using fractional repetition codes in data centers," In *Proc. 2016 IEEE 30th Int. Conf. .Adv. Info. Net. App. (AINA)*, Crans-Montana, Switzerland, 2016, pp.762-768.
[54] C. Lu, K. Ye, G. Xu, C. Xu, T. Bai, "Imbalance in the Cloud: an Analysis on Alibaba Cluster Trace," In *Proc. IEEE Int. Conf. Big Data (BigData)*, Boston, MA, USA, 2017, pp. 2884 - 2892.
[55] W. Chen, K. Ye, Y. Wang, G. Xu, C. Xu, "How Does the Workload Look Like in Production Cloud? Analysis and Clustering of Workloads on Alibaba Cluster Trace," In *Proc. 24th IEEE Int. Conf. Para. Dist. Syst. (ICPADS)*, Singapore, 2018, pp.102-109.
[56] Y. Cheng, A. Anwar, X. Duan, "Analyzing Alibaba's Co-located Datacenter Workloads," In *Proc. IEEE Int. Conf. Big Data(BigData)*, Seattle, WA, USA , 2018, pp.292-297.
[57] Y. Cheng, Z. Chai, A. Anwar, "Characterizing Co-located Datacenter Workloads: An Alibaba Case Study," In *Proc. 9th ACM SIGOPS Asia-Pacific Workshop Syst. (APSys 2018)*, Jeju Island, South Korea, 2018.
[58] L. Deng, Y. Ren, F. Xu, H. He, C. Li, "Resource Utilization Analysis of Alibaba Cloud," In *Proc. Int. Conf. Intel. Com. (ICIC)*, Wuhan, China, 2018, pp 183-194.
[59] Q. Liu, Z. Yu, "The Elasticity and Plasticity in Semi-Containerized Co-locating Cloud Workload: A View from Alibaba Trace," In *Proc. ACM Symp. Cloud Com. (SoCC)*, Carlsbad, CA, USA 2018, pp.347-360.



CONGFENG JIANG is currently an Associate Professor of Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou. He is also with the School of Computer Science and Technology, Hangzhou Dianzi University, China. He received his B.E. and Ph.D. both in Hydro-electrical Engineering from North China University of Water Resources and Electric Power in 2002, and Huazhong University of Science and Technology in 2007, respectively.

Since 2007, he has been an Assistant Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, China. He has published more than 50 articles in grid computing, cloud computing, virtualization, and big data systems. His research interests include system optimization, performance evaluation, and distributed system benchmarking. Dr. Jiang is a member of ACM, IEEE, and CCF.

GUANGJIE HAN is currently a Professor with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning province, School of Software, Dalian University of Technology, Dalian, China. He received the Ph.D. degree from Northeastern University, Shenyang, China, in 2004. From 2004 to 2006, he was a Product Manager for the ZTE Company. In February 2008, he finished his work as a Postdoctoral Researcher with the Department of Computer Science, Chonnam National University, Gwangju, Korea. From October 2010 to 2011, he was a Visiting Research Scholar with Osaka University, Suita, Japan. He is the author of over 270 papers published in related international conference proceedings and journals, and is the holder of 110 patents. His current research interests include sensor networks, computer communications, mobile cloud computing, and multimedia communication and security. Dr. Han has served as a Co-chair for more than 50 international conferences/workshops and as a Technical Program Committee member of more than 150 conferences. He has served on the Editorial Boards of up to 16 international journals, including the IEEE ACCESS, Telecommunication Systems, International Journal of Ad Hoc and Ubiquitous Computing, Journal of Internet Technology and KSII Transactions on Internet and Information Systems. He has guest edited a number of special issues in IEEE Journals and Magazines. He has served as a Reviewer of more than 50 journals. He had been awarded the ComManTel 2014, ComComAP 2014, ChinaCom 2014 and Qshine 2016 Best Paper Awards. He is a member of IEEE and ACM.

JIANGBIN LIN is currently an expert software engineer of AliCloud. He received the BS degree from Peking University in 2008. He was the innovator of Pangu Testing Framework. His research interests include distributed storage systems, system benchmarking, and reliability optimization.

GANGYONG JIA is currently an Associate Professor of School of Computer Science and Technology at Hangzhou Dianzi University, China. He received his Ph.D. degree in Department of Computer Science from University of Science and Technology of China, Hefei, China, in 2013. He has published over 60 papers in related international conferences and journals. He has served as a reviewer of IEEE Transactions on Computers, IEEE Transactions on Industrial Informatics, IEEE Communications Magazine, IEEE Transactions on Modeling and Performance Evaluation of Computing Systems, IEEE Transactions on Sustainable Computing, Applied Soft Computing, International Journal of Parallel Programming, Microprocessors and Microsystems. His current research interests are IoT, cloud computing, edge computing, power management, and operating system. He is a member of IEEE and CCF.

WEISONG SHI is a Charles H. Gershenson Distinguished Faculty Fellow and a Professor of Computer Science at Wayne State University. There he directs the Mobile and Internet Systems Laboratory, Connected and Autonomous Driving Laboratory, Intel IoT Innovators Lab, and Wayne State Data Science Initiative and Wireless Health Initiative, investigating performance, reliability, power- and energy-efficiency, trust and privacy issues of networked computer systems and applications. He founded the Metro Detroit Workshop on Connected and Autonomous Driving (MetroCAD).

Dr. Shi is one of the world leaders in the edge computing research community, and has been advocating Edge Computing (Fog Computing) since 2014, a new computing paradigm in which the computing resources are placed at the edge of the Internet, in close proximity to mobile devices, sensors, end users and the emerging Internet of Everything. In 2016, he co-chaired the NSF Workshop on Grand Challenges in Edge Computing, and was the founding steering committee chair of ACM/IEEE Symposium on Edge Computing (SEC). He was interviewed by China Science Daily on the Rise of Edge Computing in December 2016. In 2018, Dr. Shi has led the development of IEEE Course on Edge Computing.

Dr. Shi is an expert in energy efficient computer systems research, including battery management for mobile systems and energy efficient data center design. He is also very active on the workload characterization and has received the best paper award of IISWC 2012 for their work on Taobao Hadoop workload analysis. His work has been adopted and used by industry, such as Baidu, Alibaba, and Intel. In addition to publications, his group has developed several tools that have been widely used by the community, including pTop, a process-level power profiling tool, has been downloaded more than 600 times from more than 30 countries; SPAN, a software power analyzer, has been downloaded more than 400 times since its release in 2012. Dr. Shi is a recipient of the National Outstanding Ph.D. dissertation award of China (2002), the NSF CAREER award (2007), Wayne State University Career Development Chair award (2009), Charles H. Gershenson Distinguished Faculty Fellow (2015), College of Engineering Faculty Research Excellence Award (2016), the Best Paper award of ICWE'04, IEEE IPDPS'05, HPCChina'12, IEEE IISWC'12, the Best Paper Nominee award of ACM UbiComp'14, the Best Student Paper Award of IEEE HealthCom'15, IEEE eHealth Best Paper Award 2017. He is an IEEE Fellow and a Distinguished Scientist of ACM.

JIAN WAN is currently a Professor of School of Information Engineering, Zhejiang University of Science and Technology, Hangzhou, China. He is the vice President of Zhejiang University of Science and Technology and vice director of Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou. He obtained his B.S. in Mechanical Engineering in 1990, M.S. in Mathematics in 1993, and Ph.D. in Computer Science in 1996, all from Zhejiang University. From 2000 to 2015, he was the dean of School of Computer Science and Technology, Hangzhou Dianzi University. He has published more than 150 articles in grid and services computing, cloud computing, virtualization, and distributed systems. His research interests include distributed systems, computer networks, and big data analytics.

He is a member of IEEE and CCF.