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

**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 \_nd 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 \_rst 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 the co-allocation of online services jobs with batch jobs. 3) Servers are clustered into seven groups by CPU utilization, memory utilization, and their correlations. Machines with a strong correlation between CPU and memory utilization provides an 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 \_ndings 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.

**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 signi\_cantly, 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 ef\_ciency improvement in commercial servers and actions on energy reduction in data centers. Speci\_cally, energy ef\_ciency 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 ef\_ciency improvements had stagnated at the levels of 2010 [4]. Except for energy ef\_ciency improvement in hardware, var- ious energy aware scheduling approaches have been pro- posed to save power and energy consumptions in data centers [5]\_[8]. The varying workload in data centers results in \_uctua- tion 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 sen- sor technology, wireless sensors are deployed in smart cities, smart homes, autonomous vehicles, and industrial environ- ments [14]. These smart sensors are capable of providing complex services with diverse requirements, including data aggregation and analytics [15]. In the emerging edge com- puting paradigm [16], data analytics can be performed in sensor nodes to save energy consumption or privacy pre- serve [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 work- loads, cloud service providers tend to adopt an over provi- sioning 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 qual- ity 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 spe- ci\_c 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 af\_nity in case of hardware and software failures. 4) How to simulate a real Internet Data Center (IDC) based on the workload characteristics for better sched- uler 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 ana- lyze 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 informa- tion 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 balanc- ing 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.

**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 process- ing, 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. 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 divi- sion 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 signi\_cantly 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 signi\_cantly 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. C. RESOURCE UTILIZATION Resource multiplexing and job co-allocation can signi\_cantly 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 \_uctuations than CPU. Moreover, memory utilization is peri- odically changing during 24 hours. Therefore, job scheduling can adapt to this characteristic to smooth the memory utiliza- tion. We give the \_tting curve of memory utilization in Fig.7 (b). The \_tting is quanti\_ed as y D 35\_ .0:00129\_*x* 􀀀 50:828/ C 45 where y is memory utilization and x is time. The magnitude of the \_tting 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 than others. For batch jobs with multiple tasks and instances, we \_nd that among 1126 failed tasks, 572 failed tasks (repre- senting 50.8% of total failed tasks)wait and halted after a very long time interval when their \_rst 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 con\_guration incom- patibility with instances running on them. Therefore, these machines should be investigated \_rst 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 classi\_cation criteria is the number of failure instances on it. Among 148899 failed instances during the \_rst exe- cution of all instances, 1121 instances are dropped dur- ing 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 noti\_ed 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. 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. Typically, an instance will be allocated with mem- ory, CPU, and disk capacity. CPU allocation is rep- resented 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 val- ues 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 D 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 alloca- tion and \_nd 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 con\_gurations 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, accord- ing 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 mem- ory 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 times- tamp 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 ser- vices through real time system monitoring or log streaming processing.

**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. 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 classi\_cation, a total of seven categories are shown in the following \_gure. 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 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 prede\_ned 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 coef\_cients of CPU and memory utilization on each machine in Fig.20. For Type #2 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 neigh- bor machines. For these machines, machine utilization can be easily estimated if more jobs of the same type are sched- uled to them. Speci\_cally, 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 esti- mated 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 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 mainte- nance 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 \_gure out solutions for optimizing performance. Single server workload characterization has been studied exten- sively. 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 geographically dis- tributed 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 signi\_- cantly 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 sys- tem. Fiandrino *et al*. [31] propose a framework of newmetrics able to assess performance and energy ef\_ciency of cloud computing communication systems. In order to limit peak power costs without any workload performance degrada- tion, Aksanli *et al*. [26], Aksanli and Rosing [32] adopt a battery-based peak shaving method, and Liu [33] develop two algorithms for data centers. Server consolidation and VM migration are typical methods that improve data center resource utilization ef\_ciency. Ahmad *et al*. [34] analyze and compare the current VM migration and servers consolidation frame- work. Varasteh and Goudarzi [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 plat- forms [37]. In addition, since consolidating different appli- cations may lead to a drop in performance, Chen *et al.* [38] develop a light-weight, non-intrusive methodology to achieve application-centric performance targets, while consolidating homogeneous and heterogeneous application. There is signi\_cant work on resource allocation in data centers. Sun *et al.* [39], present an overview on different kinds of resource management mechanisms for data cen- ters. Tan *et al*. [40] and Mazumdar and Kumar [41] propose various methods for analyzing resource usage and model- ing resource usage patterns. In addition, to maintain high resource utilization, new resource allocation strategies have been proposed for CPU and memory [42], [43]. Warneke and Leng [44] propose an approach to improve memory utilization. Shojafar *et al.* [45] and Bari *et al.* [46] pro- pose approaches to dynamically recon\_gure 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 condi- tions [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 esti- mate 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 ef\_cient data center because some servers may fail due to some reasons, and prior work proposes several solutions. Chalermarrewong *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 \_ndings 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 clus- ters 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 \_ndings and insights presented here can help a data center operator better understand the workload characteristics, improve resource utilization, and failure recovery capabilities.