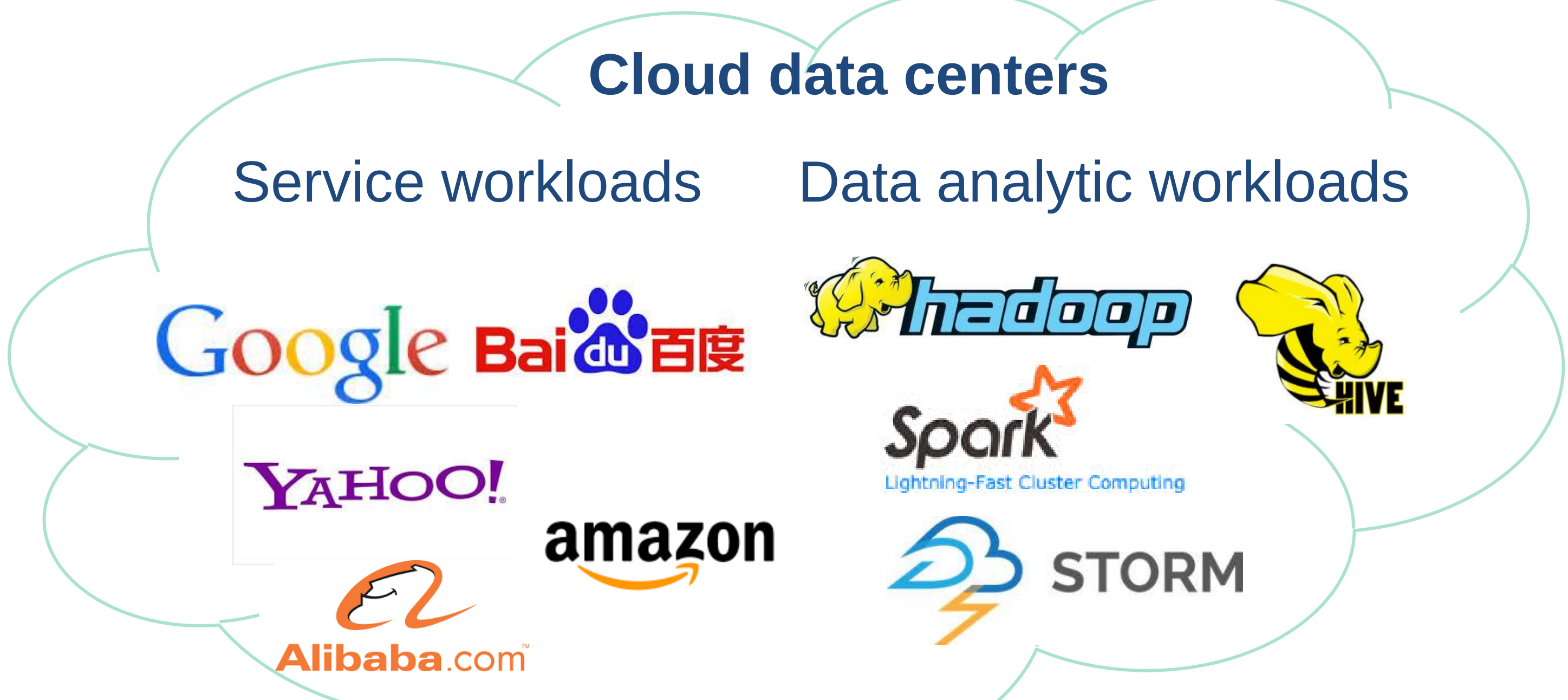
BigDataBench-MT: A Benchmark Tool for Generating Realistic Mixed Data Center Workloads

Rui Han,[1] Gang Lu,[1] Shulin Zhan,[1] Chenrong Shao,[1] Junwei Wang,[1] Lizy K. John,[2] Lei Wang[1], Jianfeng Zhan,[1] and Xinhui Tian.[1]
[1] Institute Of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] Department of Electrical and Computer Engineering, the University of Texas at Austin, TX, USA

1. Background


Cloud data centers

Service workloads

Data analytic workloads

FIGURE 1. Co-located workloads in today's cloud data centers

Workload heterogeneity and dynamicity:

1. Real workloads of diverse types
2. Dynamic request/job arrival patterns

2. Challenges

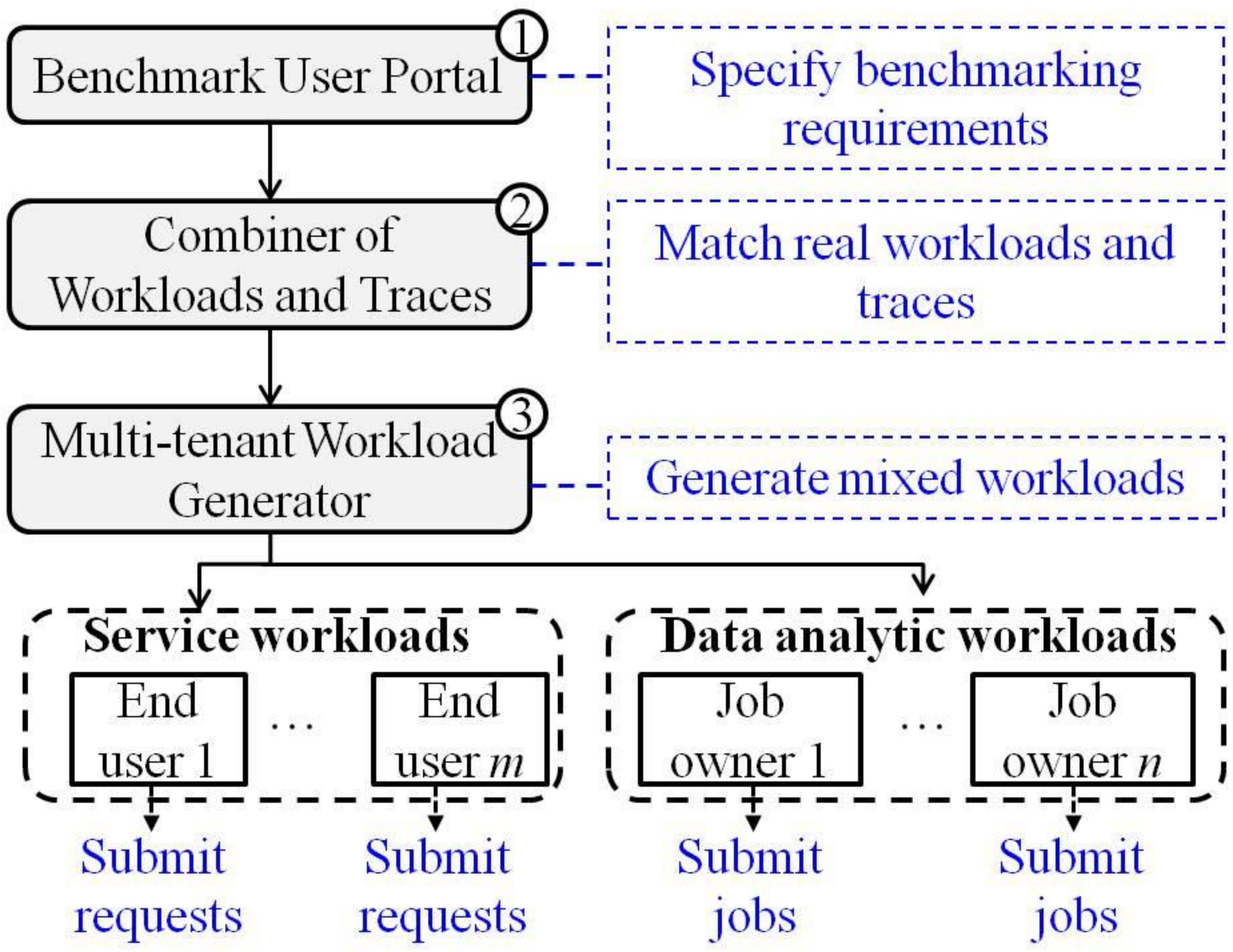
Existing data center benchmarks:

Benchmarks	Real workloads	Real workload traces
AMPLab benchmark, Linkbench , Bigbench , YCSB, CloudSuite	Yes	No
GridMix , SWIM	No	Yes

Open problem:

Generate real workloads following arrival patterns hidden in real traces

3. BigDataBench-MT Overview


Benchmark User Portal ① Specify benchmarking requirements

Combiner of Workloads and Traces ② Match real workloads and traces

Multi-tenant Workload Generator ③ Generate mixed workloads

Service workloads: End user 1 … End user *m* — Submit requests, Submit requests

Data analytic workloads: Job owner 1 … Job owner *n* — Submit jobs, Submit jobs

FIGURE 2. Three modules in BigDataBenchmark-MT

Publicly available from:

http://prof.ict.ac.cn/BigDataBench/multi-tenancyversion

TABLE 1. Real workloads and traces in BigDataBench-MT

Workload types	Real workloads	Real traces
Service workloads	Nutch search engine	Sogou user query logs
Data analytic workloads	Hadoop MapReduce and Spark	Facebook and Google cluster traces

5. Combination of real workloads and traces

Problem: Real traces usually contain anonymous workloads

Solution:

1. Derive workload characteristics

TABLE 2. Metrics to represent workload characteristics

Metric	Description
Execution time	Measured in seconds
CPU usage	Total CPU time per second
Memory usage	Measured in GB
CPI	Cycles per instruction
MAI	The number of memory accesses per instruction

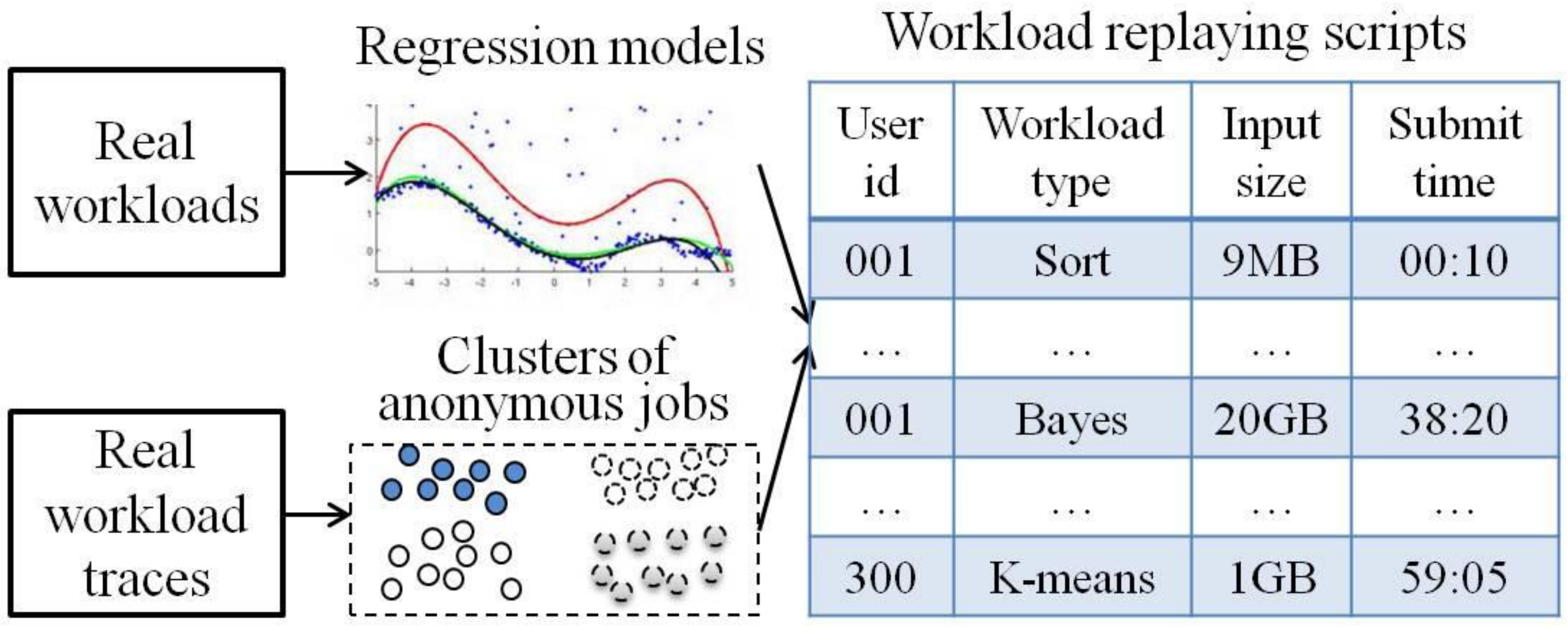
2. Matching *real* and *anonymous* workloads (Figure 3)

a) Train regression models
- ✓ *Input*: real workloads of different input sizes
- ✓ *Output*: five workload characteristic metrics

b) Group anonymous workloads into different clusters

c) Match each cluster to one real workload of a specific input

d) Output workload replaying scripts


Real workloads

Regression models

Real workload traces

Clusters of anonymous jobs

Workload replaying scripts

User id	Workload type	Input size	Submit time
001	Sort	9MB	00:10
…	…	…	…
001	Bayes	20GB	38:20
…	…	…	…
300	K-means	1GB	59:05

FIGURE 3. The matching process of real and synthetic workloads

6. Future Work

1. Employ basic operations to compose real workloads
2. Customize mixed workloads for private clouds

1. CONTEXT

In modern cloud data centers, a large number of tenants are consolidated to share a common computing infrastructure and execute a diverse mix of workloads. This paper focuses on two classes of popular workloads: long-running service workloads (e.g. web search engines and e-commerce sites) and short-term data analytic workloads (e.g. Hadoop MapReduce jobs). For example, in Google and Facebook data centers, a majority (over 90%) of analytic jobs complete within a few minutes [6]. We believe it will be of interest to the cloud community and a large user base to develop realistic benchmarks reflecting the practical scenarios of such mixed workloads.

2. RELATED WORK

Considering the heterogeneity and dynamicity of data center workloads, this requires using both real workloads and traces such that trustworthy benchmarking reflecting the practical data center scenarios can be conducted. Existing benchmarks designed for service or data analytical systems can be divided into two complementary categories.

The first category of benchmarks such as YCSB, CloudSuite, AMPLab benchmark, Linkbench and Bigbench supports real workloads but the arrival patterns of these workloads are determined by probability models. By contrast, in the second category of benchmarks such as GridMix and SWIM, the workload arrival patterns faithfully follow real workload traces, but only synthetic workloads are supported. To date, how to generate real workloads on the basis of real workload traces, is still an open question.

3. APPROACH

To fill this gap, we propose a benchmark tool that is a first step towards generating realistic mixed data center workloads. This tool, called BigDataBench-MT, is a multi-tenancy version of our open-source project BigDataBench, which is a comprehensive benchmark suite including 14 real-world data sets and 33 real workloads covering five application domains. The goal of BigDataBench-MT is not only supporting the generation of service and data analytic workloads based on real workload traces, but also providing a multi-tenant framework to enable the scaling up and down of such workloads while keeping their realistic mixes.

In the current version of BigDataBench-MT, we employ the Sogou user query logs [4] as the basis to generate the service workload (i.e. the Nutch search engine [1]) and the Facebook and Google cluster workload traces [6] as the basis to generate data analytic workloads (i.e. Hadoop MapReduce and Spark workloads). The Sogou trace includes over 9 million users and 43 million queries. The Google trace includes over 5K users, 40K workload types, 1000K jobs and 144 million tasks.

BigDataBench-MT consists of three main modules (see Figure 2 in the poster). In the *Benchmark User Portal*, users can first specify their benchmarking requirements, including the machine type and number to be tested, and the types of workload to use. A set of workload traces following these requirements are then selected. The next step (i.e. *Combiner of Workloads and Traces*) is to match the real workloads and the selected workload traces (Section 4), and outputs a workload replaying script to guide the benchmarking process. Finally, the *Multi-tenant Workload Generator* first extracts the tenant information from the script. For the service and data analytic workloads, this tenant information represents the number of concurrent end users and submitters of analytic jobs, respectively. The generator then creates a client for each tenant and emulates the scenarios that a number of end users/job submitters concurrently submit requests/jobs to the system.

4. KEY TECHNIQUE: COMBINATION OF REAL AND SYNTHETIC DATA ANALYTIC JOBS

Figure 3 in the poster illustrates the process of matching real and anonymous data analytic jobs and it consists of two parallel sub-processes. First, the real jobs with different input data sizes are tested and their workload characteristic metrics are collected. In BigDataBench-MT, we provide auto-running scripts to collect performance metrics and hardware performance counters (Perf [3] and Oprofile [2] for Linux 2.6+ based systems) to obtain micro-architectural metrics. Using the testing results as samples, the combiner trains the multivariate regression model to describe the relationship between a real job (including both its workload type and input size as the independent variables) and its workload characteristic metrics (one metric is a dependent variable). Second, the combiner views each anonymous job as an entity and the five workload characteristic metrics as its attributes, and employs the Bayesian Information Criterion (BIC)-based k-means clustering algorithm [5] to cluster anonymous jobs in the trace into different clusters.

Based on the constructed regression models and clusters, the combiner further matches each cluster to one real job with a specific input data. In the matching, the coefficient of variation (CV) measure, defined as the ratio of the standard deviation σ to the mean μ, is used to describe the dispersion of jobs in the same cluster. The combiner iteratively tests real jobs of different workload types and input sizes, and matches a real job with a cluster under two conditions: (i) the CV of the cluster is smaller than a specified threshold (e.g. 0.5), which indicates the anonymous jobs in this cluster are closely similar to each other; (ii) the change in this CV is smaller than a threshold (e.g. 0.1) after the real job is added to the cluster. This means the workload characteristics of the added job are sufficiently similar to those of the anonymous jobs in the cluster. If multiple matched real jobs are found for one cluster, the combiner selects the job resulting the smallest CV change. Finally, the combiner produces workload replaying scripts as the output.

5. REFERENCES

[1] Apache nutch search. `http://nutch.apache.org/`.

[2] Oprofile. http://oprofile.sourceforge.net/.

[3] Perf. https://perf.wiki.kernel.org/.

[4] Sogou user query logs. `http://www.sogou.com/labs/dl/q-e.html`.

[5] D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.

[6] 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 *SoCC'12*, pages 7–19. ACM, 2012.