data

MDPI

Data Descriptor

GoCJ: Google Cloud Jobs Dataset for Distributed and Cloud Computing Infrastructures

Altaf Hussain * and Muhammad Aleem

Department of Computer Science, Faculty of Computing, Capital University of Science and Technology, Islamabad 44000, Pakistan; aleem@cust.edu.pk

* Correspondence: saddamaltaf@gmail.com; Tel.: +92-300-902-1208

Received: 24 August 2018; Accepted: 26 September 2018; Published: 28 September 2018

Abstract: Developers of resource-allocation and scheduling algorithms share test datasets (i.e., benchmarks) to enable others to compare the performance of newly developed algorithms. However, mostly it is hard to acquire real cloud datasets due to the users' data confidentiality issues and policies maintained by Cloud Service Providers (CSP). Accessibility of large-scale test datasets, depicting the realistic high-performance computing requirements of cloud users, is very limited. Therefore, the publicly available real cloud dataset will significantly encourage other researchers to compare and benchmark their applications using an open-source benchmark. To meet these objectives, the contemporary state of the art has been scrutinized to explore a real workload behavior in Google cluster traces. Starting from smaller- to moderate-size cloud computing infrastructures, the dataset generation process is demonstrated using the Monte Carlo simulation method to produce a Google Cloud Jobs (GoCJ) dataset based on the analysis of Google cluster traces. With this article, the dataset is made publicly available to enable other researchers in the field to investigate and benchmark their scheduling and resource-allocation schemes for the cloud. The GoCJ dataset is archived and available on the Mendeley Data repository.

Dataset: https://data.mendeley.com/datasets/b7bp6xhrcd/1.

Keywords: GoCJ dataset; meta-task dataset; HPC dataset; scientific dataset

1. Summary

Datasets are becoming increasingly more pertinent when executing the performance assessment of cloud-scheduling, resource-allocation, and load-balancing algorithms used for eagle-eyed examination of efficiency and performance in a real-world cloud. A minor change in the behavior and nature of the dataset is reflected in the performance of scheduling and resource-allocation policies. Assessing the scheduling and allocation policies on cloud infrastructures under a varying load and system size is a challenging problem. Real cloud workload is hard to acquire for performance analysis and investigation due to the users' data confidentiality and policies maintained by Cloud Service Providers (CSPs). In addition, using real testbeds limits the experiments to the scale of the testbed. Hence, testing the accuracy performance with real-world datasets is crucial in the field of research, and synthetic data does not realistically represent an actual dataset [1]. The most appropriate alternative is to make the investigation in a simulation environment with a load of varying behavior in the cloud environment. For cloud computing research, it is valuable to formulate and ensure a widespread availability of realistic datasets that show how resourcefully the cloud addresses the user requirements.

Distributed computing is comprised of a potentially large number of heterogeneous computing resources interconnected over recurrent network architecture to meet the computing requirements of varying high-performance computing (HPC) applications [2]. Cloud computing is a paradigm of distributed computing promised to deliver on-demand utility computing over the Internet. The resources are provisioned in the form of *Virtual Machines* (VMs) deployed within cloud datacenters consisting of physical host machines. These datacenters are generally over-provisioned to guarantee high service availability and Quality of Service (QoS) computing [3]. The QoS contracts are formally negotiated and self-proclaimed in the users' Service-Level Agreements (SLA) providing confidence to customers in outsourcing and executing HPC applications in cloud [4,5]. Cloud computing is a unique platform that offers solutions to small business users of computation-hungry large scientific applications. The cloud offers dynamically scalable access to the benefits of technology instead of worrying about involved deployment, building, investment, maintenance, and operation of physical infrastructure [6]. It is evident that several researchers and even different businesses are keenly interested in using the cloud infrastructure for executing scientific applications in remote datacenters [7]. The cloud provides its services in the form of a platform or infrastructure to real-time deployment, execution, or simulation of different computation-greedy applications i.e., big network traffic data visualizations [8], multi-threaded learning control mechanisms for neural networks [9], performance tests on merge sorts, recursive merge sorts for big data processing [10], and parallelization of modified merge sort algorithms [11] etc. Some parallel applications result in the degradation of resource use in cloud computing, undesirably affecting the performance of the computing environment. Thus, adopting an efficient and appropriate scheduling mechanism to achieve required scheduling objectives becomes a challenging research issue. Parenthetically, researchers need to archive scientific datasets that exhibit behaviors of realistic cloud workloads to enable other researchers to evaluate the scheduling mechanisms with desired performances.

Some datasets (i.e., Heterogeneous Computing Scheduling Problems (HCSP) Instances [12], and Task Execution Time Modeling (TETM) [13]) are publicly available for research purposes and these datasets are based on varying task and machine heterogeneity. However, the compositions of the job sizes in these datasets are not derived from any real cluster or cloud workload. Furthermore, there are some publicly available real workload traces (such as Google cluster traces [14], Yahoo cluster traces [15], Facebook Hadoop workload [16], OpenCloud Hadoop workload [17], Eucalyptus IaaS cloud Workload [18], and the Grid Workload Archiver TuDelft (GWA-T) traces [19], etc.); however, most of these require preprocessing and in-depth low-level details to be regenerated and used for experimentation.

The main contribution of this data descriptor is to introduce a realistic Google Cloud Jobs (GoCJ) dataset based on Google cloud infrastructure as a benchmark for cloud-scheduling researchers. This data descriptor is presented to address the research issue of ensuring the public availability of a realistic dataset that satisfies the need for researchers in performance analysis of cloud infrastructure. The GoCJ dataset generator is presented in the form of an Excel sheet generator and a Java-based tool. A sample dataset (with a small number of jobs) based on jobs trend behavior in Google cluster traces is formulated and provided to the GoCJ dataset generator, which simulates the desired dataset comprised of any required number of jobs. The proposed GoCJ is used in [20] for performance analysis of several load-balancing Cloud schedulers. The value and importance of the GoCJ dataset can be enumerated as follows:

- The GoCJ dataset provides a reflection of real workload behavior as perceived in Google cluster traces [21–26] and MapReduce logs from the M45 supercomputing cluster, so it has more significance and usefulness for researchers working in the scheduling of cluster and cloud-based applications;
- The GoCJ dataset can serve as an alternative to benchmark workload for scheduling and resource-allocation mechanisms using realistic HPC jobs in cloud computing.

The rest of the paper is organized as follows. Section 2 discusses some existing datasets in the research domain, presents the description of the proposed GoCJ dataset, the composition of jobs in GoCJ dataset and its comparison with the exiting datasets in the literature. The assumption and approach toward generating the GoCJ dataset, and the complexity of GoCJ generator tool, is presented in Section 3. The last section concludes the paper and identifies future directions for the GoCJ dataset.

2. Data Description

The GoCJ dataset is provided as a supplementary data in text and Excel file formats in amalgamation with the two dataset generator files: (1) an Excel worksheet generator, and (2) a Java tool generator. Each row in the text file describes the size of a specific job in terms of Millions of Instructions (MI). The Monte Carlo simulation method [2] is employed to generate the dataset comprised of any required number of jobs. The specification of the GoCJ dataset is presented in Table 1.

Table 1. Description of GoCJ dataset.

Feature	Description
Subject area	Computer science.
More specific subject area	Cloud scheduling.
Type of data	19 Text files comprised of job sizes (in MIs) in the dataset, 01 text file containing original dataset, and 01 text file comprised of a Java source code for GoCJ generator to be discussed in Section 3, and 01 Excel file as GoCJ dataset generator.
How data was acquired	Job-size behavior of Google cluster traces is presented and published in the literature survey [21–24,26]. The Google cluster traces are also available [Online] https://github.com/google/cluster-data.
Dataset generation	GoCJ dataset is created using Monte Carlo simulation.
Data format	Raw, filtered, analyzed.
Data accessibility	The data is available with this data descriptor article. To ensure publicly accessible availability of GoCJ dataset, the data is also archived and publicly available in a Mendeley Data repository.
Related research article	The presented dataset has been used in a research article [20]: "Resource-Aware Load-Balancing Algorithm (RALBA): a computation-aware load-balancing scheduler for cloud computing" (Hussain et al. 2018). The dataset is used to evaluate the RALBA and other eight scheduling heuristics in cloud computing in terms of resource use and makespan of the system.
Related dataset	- Heterogeneous Computing Scheduling Problem (HCSP) instances [12], - Task Execution Time Modeling (TETM) [13], - Yahoo cluster traces [15], - Facebook Hadoop Workload [16], - OpenCloud Hadoop workload [17], - Eucalyptus IaaS cloud Workload [18], - GWA-T traces [19].

The GoCJ dataset, stored in a Mendeley Data repository, is comprised of 21 text files. Each dataset file is named as "GoCJ_Dataset_XXX.txt", where XXX is the number of jobs in the file i.e., "GoCJ_Dataset_100.txt" has the sizes of 100 jobs. Each text file consists of a set of rows, where each row has a numeric value presenting the size of a job in terms of MI. Job completion time for GoCJ dataset follows a long-tailed distribution (with 90% of the jobs completing on average within 1.6 min). The longest-executing job witnessed in the dataset lasts up to 15 min (i.e., 6% of the jobs execute for less than 5 min and 4% of the jobs execute for 15 min). The average size of a job in the GoCJ dataset is 5 min. Figure 1 displays ratios and sizes of jobs distribution in GoCJ dataset in terms of percentage and MIs, respectively.

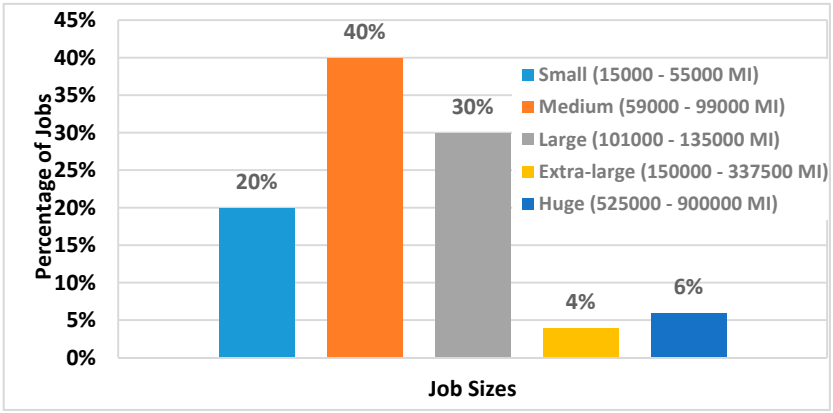

Figure 1. The composition of the GoCJ dataset.

Some existing datasets (i.e., HCSP, TETM, and GWA-T instances) are discussed and compared with the GoCJ dataset. The HCSP instances and TETM dataset are used in research work [24–26]. However, HCSP and TETM datasets are not based on compute-traces of any real Grid or Cloud system. On the other hand, GWA-T traces are based on distributed datacenters of Bitbrains and relates to a research work in [27]. Similarly, GWA-T traces exhibit the performance metrics of VMs in the datacenters instead of jobs behavior. The comparison of existing datasets with the proposed GoCJ dataset is presented in Table 2.

Table 2. Comparison of the proposed and existing Cloud datasets.

Dataset	Repository	Main Features and Limitation	Potential Users
HCSP instances [12]	HCSP website [12]	HCSP instances is a standardized benchmark-based on a range-based method to produce Expected Time to Compute (ETC) matrices with a variation of heterogeneity in the tasks to be executed and machines in the system. Likewise, small size HCSP instances (up to 1024 tasks and 32 machines) are provided to direct download, while for the larger instances a generator program along with the seeds used for the random number generator (execute the generator program using the correspondent seeds to replicate the instances) is provided.	Research community in heterogeneous computing systems
TETM instances [13]	Not available; instead a method for dataset creation is available in [13]	To simulate a heterogeneous computing environment, a coefficient-of-variation-based technique to produce ETC matrices with a variation of heterogeneity in the tasks to be executed, and computing machines in the distributed system is presented. The model is capable of generating the dataset to evaluate the performance of scheduling heuristics. It generates a dataset reflecting the required task and machine heterogeneity; however, the TETM dataset does not reflect the realistic workload behavior i.e., the workload behavior on Google cluster traces or real compute Cloud etc. Similar to GoCJ, any number of tasks can be created with the TETM method.	Researchers in a distributed heterogeneous computing environment

Table 2. *Cont.*

GWA-T traces [19]	The Grid Workloads Archive [19]	The dataset contains the performance metrics of 1750 VMs from distributed datacenters of Bitbrains, which is a service provider of hosting and business computation. The clients of Bitbrains are many major banks, credit card operators, and insurers etc. GWA-T traces focus on performance metrics of VMs; however, the GoCJ exhibits the jobs sizes and behaviors in a workload.	Users of Grid computing/distributed datacenters
Facebook Hadoop workload [16]	GitHub Data repository [16]	The workload is based on Hadoop traces on 600 machine cluster on Facebook spans over 6 months duration from May 2009 to October 2009 containing 1 million jobs. The jobs in the workload are recorded with submit time and inter-job_submit_gap parameters.	Heterogeneous computing/cluster computing systems
GoCJ [27]	Mendeley Data [27]	The GoCJ dataset is based on jobs size behaviors witnessed in the analysis of the Google cluster traces of 29 days, and the MapReduce logs from the M45 supercomputing cluster of 10 months. Hence, the GoCJ dataset is an alternative to the real workload on a Cloud. Two dataset generators (i.e., one Excel sheet and the other is the Java generator program) are provided to produce the GoCJ dataset comprised of any number of Cloud jobs.	Research community in cloud computing/ heterogeneous computing systems

3. Methods

In this section, the data acquisition for original dataset, and the process of generating the GoCJ dataset, is described in detail. In addition, the complexity of and data distribution in GoCJ dataset is also presented.

3.1. Data Acquisition for Original Dataset

The Monte Carlo simulation method is used to generate the GoCJ dataset. A sample original dataset is formulated based on workload behavior in Google cluster traces, which is input into the Monte Carlo simulation. The composition of jobs in the simulated GoCJ dataset is generated based on the job sizes in the original dataset.

The contemporary state of the art has been scrutinized to explore a real workload behavior in Google cluster traces [21–24,26] and MapReduce logs from the M45 supercomputing cluster [25]. Liu and Cho analyzed large-scale Google cluster traces of 29-days to examine the machine properties, workload behavior and resource use [21]. The analysis affirms that the majority of the jobs execute for fairly a short duration of fewer than 15 min, while a few jobs execute over 300 min. The median length of a job in Google cluster traces is witnessed as approximately 3 min. Furthermore, it establishes the fact that approximately two-thirds of the jobs execute for less than 5 min and approximately 20% of the jobs execute for less than one minute. It is found that most jobs in Google cluster traces are shorter length. The shorter jobs are generally used for test runs on Google cluster [21]. Likewise, the MapReduce logs of the M45 supercomputing cluster presented in [7] is also scrutinized. The MapReduce logs for the duration of 10 months (i.e., Hadoop logs from 25 April 2008, to 24 April 2009, except logs from 12 Nov 2008, to 18 Jan 2009) have been released by the Yahoo. It is shown that most of jobs (i.e., 95% of the jobs) complete the execution within 20 min, and approximately 4% of jobs take up to 30 min [7] to execute.

Based on the analysis of [21–26], a GoCJ realistic data set is generated using the bootstrapped Monte Carlo (MC) simulation method [28]. Instead of Random Number Generation (RNG), the original dataset is repeatedly sampled by selecting one of the data points from the original dataset in bootstrapping [28]. A list of 50 different-size jobs (i.e., presented in Table 3) are identified and

input into MC bootstrapping as the original dataset. Each job size in the dataset is treated with equal probability in repeated sampling by bootstrapping. The average power of machine in the distributed computing environment for finding job sizes is assumed as 1000 Million Instructions Per Second (MIPS). The majority of the smaller jobs in Google-like realistic dataset (i.e., 90% of jobs) execute for up to 1.6 min. However, the longest-executing jobs in the GoCJ dataset execute for up to 15 min (i.e., 900,000 MIs/1000 MIPS = 900 s = 15 min). The sizes of jobs in the GoCJ dataset is presented in terms of MIs instead of ETC as used in other available datasets [12,13]. The job size is calculated from the ETC of the job, using the following relationship, where Job_{MI} presents the size of job in MIs, $Machine_{MIPS}$ is the power of computing machine in MIPS, and ETC is the expected time to complete a job in seconds [20]. The job size can be calculated using Equation (1).

$$Job_{MI} = Machine_{MIPS} \times ETC_{Second} \quad (1)$$

The converse of Equation (1) is to determine the ETC of a job, which is presented [20] as:

$$ETC_{Second} = \frac{Job_{MI}}{Machine_{MIPS}} \quad (2)$$

The job sizes in HCSP instances, GWA-T traces, ETM datasets, Facebook Hadoop workload [16], and Yahoo cluster traces [15] are presented in terms of ETC. On the other hand, the composition of the original dataset is presented in Table 3 (which is derived using Equation (1)). Fifty different job sizes are identified based on the analysis of Google cluster traces with an equal probability of occurrences in the desired GoCJ dataset. Using Equation (1) by considering the computing power of machine as 1000 MIPS and the ETC of jobs (derived from workload behavior studied in Google cloud infrastructure [21–24,26]), the original dataset is created and presented in Table 3.

Table 3. Sizes of jobs in original dataset for GoCJ (in MIs).

Small	Medium	Large	Extra-Large	Huge
15,000, 27,500, 40,000, 45,000, 47,000, 49,000, 51,000, 53,000, 55,000	59,000, 61,000, 63,000, 65,000, 67,000, 71,000, 73,000, 75,000, 77,000, 79,000, 81,000, 83,000, 85,000, 87,000, 89,000, 91,000, 93,000, 95,000, 97,000, 99,000	101,000, 103,000, 105,000, 107,000, 109,000, 111,000, 113,000, 115,000, 117,000, 119,000, 121,000, 123,000, 125,000, 127,000, 129,000, 135,000	150,000, 525,000	525,000, 712,500, 900,000

3.2. Reproduction of GoCJ Realistic Dataset

The dataset is generated by bootstrapped MC simulation using an Excel worksheet (as shown in Figure 2). The original dataset is input in column C from cell C1 through C50, highlighted with a yellow background in Figure 2. The individual probability of occurrence of each job is placed in the corresponding row of column A (i.e., from cell A1 through A50). Congruently, the cumulative probability of each job is placed in column B from cell B1 through B50. Therefore, a data table (i.e., from cell B1 through C50) is generated where each job size in column C is referenced by a cumulative probability in column B. As presented in column I in the worksheet, uniform RNG is used to generate a random number among the indices of job sizes in the data table. Now, a built-in function VLOOKUP in Excel is used to create the GoCJ dataset based on the original dataset provided in column C. The VLOOKUP function inputs the generated random number (i.e., in the corresponding row of column E), and the data table (i.e., from cells B1 through C50). Then, VLOOKUP searches for the entry in the data table based on the input random number (i.e., search in cumulative probabilities

in column B) and returns the corresponding job-size entry to store it in column F, highlighted with a green background as shown in Figure 2. The VLOOKUP-based formula used to find a job size is as follows:

$$F1 = VLOOKUP\ (E1, \$B\$1:\$C\$50, 2) \quad (3)$$

The GoCJ dataset with the specified number of jobs can be created by extending the formulas in cells E1 and F1 by copy/paste to the row equal to the desired number of jobs in the dataset.

One Excel file, named "GoCJ_Dataset_Monte_Carlo.xlsx", is also placed along with the text files in the dataset. This Excel file can be used to generate a GoCJ dataset comprised of the desired number of jobs. A new dataset can be generated by copying any of the worksheets in the "GoCJ_Dataset_Monte_Carlo.xlsx" Excel file, and then extending the formulas in column E and F by copy/paste, as discussed.

	A	B	C	D	E	F	G
1	0.02	0	15000		0.82477	125000	
2	0.02	0.02	27500		0.0067	15000	
3	0.02	0.04	40000		0.23684	63000	
4	0.02	0.06	45000		0.92653	337500	
5	0.02	0.08	47000		0.89473	135000	
6	0.02	0.1	49000		0.72494	115000	
7	0.02	0.12	51000		0.91522	150000	
8	0.02	0.14	53000		0.06835	45000	
9	0.02	0.16	55000		0.02644	27500	
10	0.02	0.18	59000		0.15855	53000	
11	0.02	0.2	61000		0.41961	83000	
12	0.02	0.22	63000		0.0429	40000	
13	0.02	0.24	65000		0.32682	75000	
14	0.02	0.26	67000		0.77935	119000	
15	0.02	0.28	71000		0.08815	47000	
16	0.02	0.3	73000		0.40771	83000	
17	0.02	0.32	75000		0.66957	109000	
18	0.02	0.34	77000		0.9136	150000	
19	0.02	0.36	79000		0.35568	77000	
20	0.02	0.38	81000		0.51804	93000	
21	0.02	0.4	83000		0.29206	71000	
22	0.02	0.42	85000		0.90112	150000	
23	0.02	0.44	87000		0.20031	61000	
24	0.02	0.46	89000		0.82516	125000	
25	0.02	0.48	91000		0.41658	83000	
26	0.02	0.5	93000		0.0184	15000	
27	0.02	0.52	95000		0.98706	900000	
28	0.02	0.54	97000		0.51058	93000	
29	0.02	0.56	99000		0.80718	123000	
30	0.02	0.58	101000		0.19399	59000	
31	0.02	0.6	103000		0.84291	127000	
32	0.02	0.62	105000		0.04853	40000	
33	0.02	0.64	107000		0.65054	107000	
34	0.02	0.66	109000		0.66589	109000	
35	0.02	0.68	111000		0.00755	15000	
36	0.02	0.7	113000		0.49215	91000	
37	0.02	0.72	115000		0.99745	900000	
38	0.02	0.74	117000		0.67561	109000	
39	0.02	0.76	119000		0.12788	51000	
40	0.02	0.78	121000		0.15827	53000	
41	0.02	0.8	123000		0.68198	111000	
42	0.02	0.82	125000		0.66557	109000	
43	0.02	0.84	127000		0.02807	27500	
44	0.02	0.86	129000		0.12812	51000	
45	0.02	0.88	135000		0.67987	109000	
46	0.02	0.9	150000		0.88851	135000	
47	0.02	0.92	337500		0.70781	113000	
48	0.02	0.94	525000		0.46747	89000	
49	0.02	0.96	712500		0.3209	75000	
50	0.02	0.98	900000		0.48174	91000	

Figure 2. GoCJ Excel worksheet generator.

3.3. GoCJ Dataset Generator Tool

Alternatively to the Excel sheet generator, an automated dataset generator program is provided using Java programming. The algorithm is presented as Algorithm 1, which presents the formal algorithm for creating a dataset as discussed in Section 3.1. The GoCJ generator performs the necessary initialization (lines 1–4). cPer variable presents the cumulative percentage of probability for each job in the original dataset to occur in the simulated dataset, jobSize is the size of the job to occur in the simulated dataset, and jList is the list of jobs finally produced in the simulated dataset. Afterward, the original dataset from a text file named "Original_Dataset" is read and resided in bufferReader (in lines 5–6). Line 5 reads all the job sizes in the original dataset derived from the Google cluster traces. A while-loop is used to copy the sample job sizes from bufferReader to fill the dataTable. The dataTable contains the job sizes in original dataset along with its cumulative probability (i.e., lines 7–10 of Algorithm 1). Line 9 maps the cumulative probability of each job in the dataTable. The second while-loop is used to produce the GoCJ dataset with the desired number of jobs. The job size produced in each iteration of the loop (i.e., lines 12–15) is stored in a job list (i.e., jList variable). The source code of the given algorithm in Java program is available on Mendeley along with the GoCJ dataset files and provided with the supplementary files as well.

Algorithm 1: GoCJ Generator

Input: num — desired number of jobs in dataset,
Original_DataSet — file of original dataset sample
Output: jList — list of job sizes in the desired dataset

```
1  cP er = 0
2  jobSize = 0
3  jList = N ull
4  dataT able < cP er, jobSize ≥ N ull
5  fileReader = readFile(Original_DataSet)
6  buf f erReader = read(f ileReader)
7  while buf f erReader is Not Empty do
8           jobSize = long.parseLong(bufferRedear.readLine())
9           dataT able.add(cPer,jobSize)
10           cP er = cP er + 2
11 a=1
12 while num ≥ a do
13           rand = Random.nextInt(100)
14           jList.add((rand Mod 2)?dataT able.get(rand): getJobSize(rand))
15           a++
16 return jList
```

3.4. Data Distribution and Complexity of GoCJ Generator

To scrutinize the complexity and efficiency of the GoCJ generator, N number of desired jobs are considered in the GoCJ dataset. As mentioned in Section 3, the average computing power of a machine that is used to generate job sizes in the original dataset is 1000 MIPS. Algorithm 1 inputs two parameters; first num is desired number of jobs in the simulated dataset and second is the Original_Dataset with a fixed number of jobs (i.e., 50 jobs). Therefore, the first while-loop in Algorithm 1 iterates for a fixed number of times (i.e., equal to the number of job sizes in the original dataset). On the other hand, the second while-loop iterates till N that is the desired number of jobs in the simulated dataset. The computational complexity of GoCJ generator is linear or O(N) depending only on the desired number of jobs in the simulated dataset. Due to the linear complexity of GoCJ generator, it will not create any overhead even for a dataset creation with a large number (N) of jobs.

To test the compliance of the data distribution trend in the original dataset to the simulated dataset produced by the GoCJ generator, the covariance statistical test is performed. The average data distribution of 19 GoCJ dataset files is determined and its covariance with the original dataset is calculated to find the correlation of data distribution. The covariance of the original dataset to the average simulated dataset is 2.49, demonstrating that both the original and simulated datasets are positively dependent. Furthermore, the ratios of job-size distribution are also presented in Figure 3 to highlight the job distribution trends in original and simulated datasets.

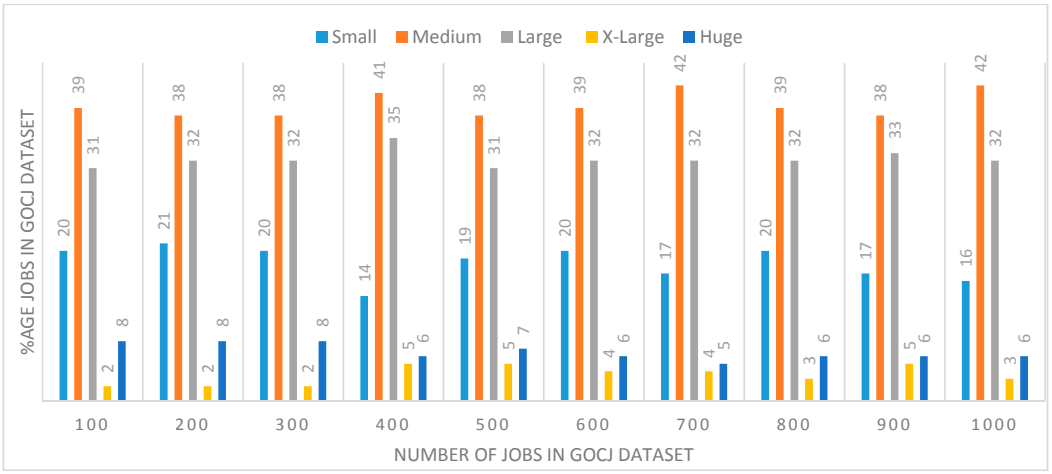

Figure 3. Ratios of jobs distribution in the GoCJ dataset.

Another statistical measure that displays the data distribution based on five-number summary (i.e., minimum, first quartile, median, third quartile and maximum) is boxplot visualization. The five-number summary for original datasets and ten sample datasets produced using GoCJ generated is determined. To reflect the job-size trends in the original and simulated dataset, a boxplot visualization is shown in Figure 4. The boxplot presents the median of job sizes in the original, and ten simulated GoCJ datasets which are 92,000, 91,000, 87,000, 89,000, 95,000, 93,000, 93,000, 93,000, 91,000, 93,000, and 93,000 MIs, respectively. These job sizes belong to the medium job sizes as shown in Figure 1 and Table 3. However, the minimum and maximum job sizes in both original and simulated GoCJ datasets are same as 15,000 and 900,000 MIs, respectively. Similarly, the first quartile and third quartile of original and simulated GoCJ datasets belongs to a category of medium- and large-size jobs as presented in Figure 1. Thus, the boxplot visualization proves that the data distribution in the simulated GoCJ datasets complies with the data distribution in the original dataset in the same manner.

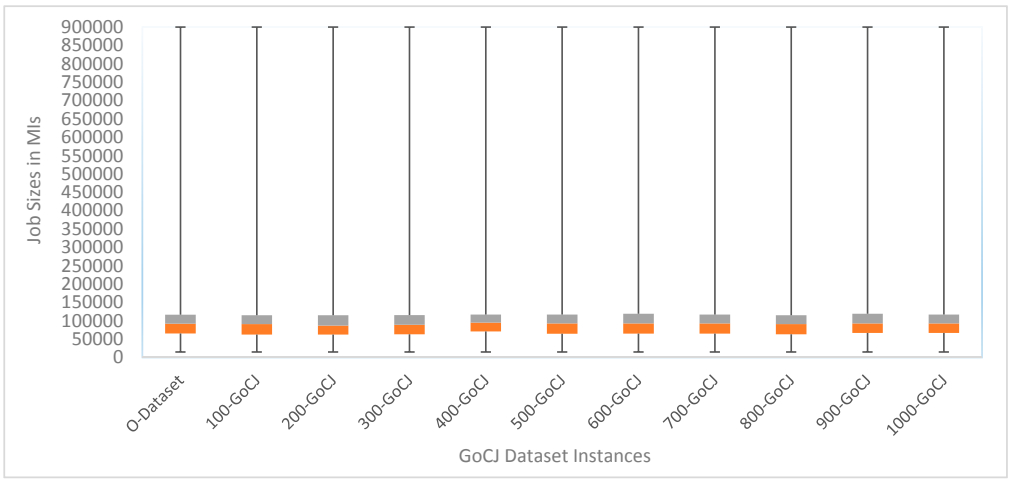

Figure 4. Boxplot visualization of jobs distribution in original and simulated GoCJ datasets.

4. User Notes

It is very hard to acquire real cloud datasets due to user data confidentiality and policies maintained by CSPs. This restricts the performance analysis to the scale of the real dataset and making diversified empirical examination impossible as per varying requirements of the researchers. Thus, realizing the need for a varying size realistic dataset, the GoCJ dataset is generated based on the real jobs trend in Google cluster traces. A very comprehensive analysis of the Google cluster traces [21–24] is conducted and presented in the form of GoCJ dataset. An original dataset based on this analysis is derived depicting the jobs behavior in real Google cluster. Then, MC bootstrapping is used to generate the GoCJ dataset based on this original dataset (i.e., dataset comprised of 50 different fix-sized jobs) as a seed to the GoCJ workload generator. Two GoCJ dataset generators (i.e., Excel worksheet and Java tool generators) are presented so that a researcher can easily create a GoCJ dataset comprised of any desired number of jobs. The job sizes in GoCJ are presented in terms of MIs, where the same is presented in the HCSP, TETM, and GWA-T instances as ETC. However, the conversion of the GoCJ to an ETC-based dataset is also presented and explained. The proposed GoCJ dataset can be used by the cloud research community for cloud scheduling, resource-allocation policies, and benchmark-based performance analysis. GoCJ dataset is also used in the performance analysis of a resource-aware load-balancing technique as a benchmark in cloud infrastructure [20].

However, the GoCJ dataset is based on the jobs trend observed in one specific real cloud infrastructure (i.e., Google cluster traces). As a future direction of this data descriptor, the GoCJ dataset can be enhanced with a generalized original dataset (to be input into the MC simulation) based on a diversified analysis of different multiple real cloud infrastructure (i.e., Google cluster traces, Facebook Hadoop workload, Yahoo cluster traces, etc.). In addition, the GoCJ can be equipped with SLA- and deadline-based jobs parameters based on a comprehensive literature review that would be useful in the performance analysis of SLA-aware and constraints-oriented resource-allocation and scheduling policies in cloud.

Supplementary Materials: The following 21 text files and one Excel file are available online at https://data.mendeley.com/datasets/b7bp6xhrcd/1. Names of 21 Text files: GoCJ_Dataset_100.txt, GoCJ_Dataset_150.txt, GoCJ_Dataset_200.txt, GoCJ_Dataset_250.txt, GoCJ_Dataset_300.txt, GoCJ_Dataset_350.txt, GoCJ_Dataset_400.txt, GoCJ_Dataset_450.txt, GoCJ_Dataset_500.txt, GoCJ_Dataset_550.txt, GoCJ_Dataset_600.txt, GoCJ_Dataset_650.txt, GoCJ_Dataset_700.txt, GoCJ_Dataset_750.txt, GoCJ_Dataset_800.txt, GoCJ_Dataset_850.txt, GoCJ_Dataset_900.txt, GoCJ_Dataset_950.txt, GoCJ_Dataset_1000.txt, Original_DataSet.txt, GoCJ Java Generator.txt. Name of 01 Excel file: GoCJ_Dataset_Monte_Carlo.txt.

Author Contributions: A.H. conceived the study, scrutinized the existing literature, developed the tools employed for the dataset generation, and written the initial draft of the paper. M.A. contributed to the analysis of results, revising the paper, and overall supervision of the work.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Makonin, S.; Wang, Z.J.; Tumpach, Z.J. 'RAE: The Rainforest Automation Energy Dataset for Smart Grid Meter Data Analysis'. *Data* **2018**, *3*, 8. [CrossRef]
2. Ghorbannia, A.; Yalda, D. 'HSGA: A hybrid heuristic algorithm for workflow scheduling in cloud systems HSGA: A hybrid heuristic algorithm for workflow scheduling in cloud systems'. *Cluster Comput.* **2014**, *17*, 129–137. [CrossRef]
3. Beloglazov, A.; Buyya, R. 'Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints'. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 1366–1379. [CrossRef]

4. Yeo, C.S.; Buyya, R. Service Level Agreement based Allocation of Cluster Resources: Handling Penalty to Enhance Utility. In Proceedings of the 7th IEEE International Conference on Cluster Computing, Burlington, MA, USA, 27–30 September 2005.
5. Durao, F.; Fernando, J.; Carvalho, S.; Fonseka, A. A systematic review on cloud computing. *J. Supercomput.* **2014**, *68*, 1321–1346. [CrossRef]
6. Vaquero, L.M.; Rodero-Merino, L.; Buyya, R. Dynamically Scaling Applications in the Cloud. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 45–52. [CrossRef]
7. Tripathy, L.; Patra, R.R. Scheduling in Cloud Computing. *Int. J. Cloud Comput. Serv. Archit.* **2014**, *4*, 21–27. [CrossRef]
8. Ruan, Z.; Miao, Y.; Pan, L.; Xiang, Y.; Zhang, J. Big network traffic data visualization. *Multimed. Tools Appl.* **2018**, *77*, 11459–11487. [CrossRef]
9. Połap, D.; Woźniak, M.; Wei, W.; Damaševičius, R. Multi-threaded learning control mechanism for neural networks. *Futur. Gener. Comput. Syst.* **2018**, *87*, 16–34. [CrossRef]
10. Marszałek, Z. Performance tests on merge sort and recursive merge sort for big data processing. *Tech. Sci.* **2018**, *21*, 19–35.
11. Marszałek, Z. Parallelization of Modified Merge Sort Algorithm. *Symmetry* **2017**, *9*, 176. [CrossRef]
12. Heterogeneous Computing Scheduling Problem (HCSP) Instances. Available online: https://www.fing.edu.uy/inco/grupos/cecal/hpc/HCSP/HCSP_inst.htm (accessed on 24 August 2018).
13. Ali, S.; Siegel, H.J.; Maheswaran, M.; Hensgen, D.; Ali, S. Task execution time modeling for heterogeneous computing systems. In Proceedings of the 9th Heterogeneous Computing Workshop, Cancun, Mexico, 1 May 2000; pp. 185–199.
14. Google cluster traces. Available online: https://github.com/google/cluster-data (accessed on 24 August 2018).
15. Yahoo Cluster traces. Available online: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&guccounter=1 (accessed on 24 August 2018).
16. Facebook Hadoop Workload. Available online: https://github.com/SWIMProjectUCB/SWIM/wiki/Workloads-repository (accessed on 22 August 2018).
17. OpenCloud Hadoop workload. Available online: http://ftp.pdl.cmu.edu/pub/datasets/hla/ (accessed on 20 August 2018).
18. Eucalyptus IaaS cloud Workload. Available online: https://www.cs.ucsb.edu/~rich/workload/ (accessed on 20 August 2018).
19. GWA-T-12 traces. Available online: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains (accessed on 24 August 2018).
20. Hussain, A.; Aleem, M.; Khan, A.; Iqbal, M.A.; Islam, M.A. RALBA: A computation-aware load balancing scheduler for cloud computing. *Clust. Comput.* **2018**, 1–14. [CrossRef]
21. Liu, Z.; Cho, S. Characterizing machines and workloads on a Google cluster. In Proceedings of the 41st International Conference on Parallel Processing Workshops, Pittsburgh, PA, USA, 10–13 September 2012; pp. 397–403.
22. Moreno, I.S.; Garraghan, P.; Townend, P.; Xu, J. An approach for characterizing workloads in google cloud to derive realistic resource utilization models. In Proceedings of the 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, Redwood City, CA, USA, 25–28 March 2013; pp. 49–60.
23. Chen, Y.; Ganapathi, A.S.; Griffith, R.; Katz, R.H. Analysis and Lessons from a Publicly Available Google Cluster Trace. Available online: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-95.html (accessed on 24 August 2018).
24. Reiss, C.; Tumanov, A.; Ganger, G.R.; Katz, R.H.; Kozuch, M.A. Towards understanding heterogeneous clouds at scale: Google trace analysis. Available online: http://www.pdl.cmu.edu/PDL-FTP/CloudComputing/ISTC-CC-TR-12-101.pdf (accessed on 22 August 2018).
25. Kavulya, S.; Tany, J.; Gandhi, R.; Narasimhan, P. An analysis of traces from a production MapReduce cluster. In Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (CCGrid), Melbourne, Australia, 17–20 May 2010; pp. 94–103.
26. Liu, C.; Liu, C.; Shang, Y.; Chen, S.; Cheng, B.; Chen, J. An adaptive prediction approach based on workload pattern discrimination in the cloud. *J. Netw. Comput. Appl.* **2017**, *80*, 35–44. [CrossRef]

27. Hussain, A.; Aleem, M. GoCJ: Google Cloud Jobs Dataset, 2018. Available online: https://data.mendeley.com/datasets/b7bp6xhrcd/1 (accessed on 24 August 2018).
28. Mason, S.J.; Hill, R.R.; Mönch, L.; Rose, O.; Jefferson, T.; Fowler, J.W. Introduction to Monte Carlo Simulation. In Proceedings of the 2008 Winter Simulation Conference, Miami, FL, USA, 7–10 December 2008; pp. 91–100.