WEBCLoudSim: An Open Online Cloud Computing Simulation Tool for Algorithm Comparison
Youqi Chen, Qibo Sun, Ao Zhou, Shangguang Wang
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
{yqchen, qbsun, aozhou, sgwang}@bupt.edu.cn

Abstract
Comparatively evaluating cloud computing algorithms via credible simulation tools is desirable by many contemporary cloud service researchers and practitioners in terms of cost efficiency and effectiveness. Common cloud computing simulation tools need nontrivial improvements in extensibility, usability, and/or manageability. Aiming to facilitate developing, analyzing and comparing advanced cloud computing algorithms, we have developed the simulation tool WebCloudSim, which is built via open source code and can be accessed for free at http://webcloudsim.org via a web browser. Under the browser-server deployment model, WebCloudSim supports custom settings of target cloud datacenters. It is easy for a WebCloudSim user to configure a specific cloud datacenter environment for several algorithms under evaluation, and to exploit the comparative analysis data and charts that the tool generates automatically. This paper presents design, implementation and usage of the WebCloudSim tool with real use cases.

Keywords: cloud computing; cloud simulation; algorithm comparison; dynamic task

1. INTRODUCTION
Cloud computing has become the de facto enterprise computing model. Enterprises in the world are building their on-premises clouds and/or exploiting public clouds with the goal of optimizing their IT return-on-investment and/or accelerating their business innovations. A common challenge of optimizing the exploitation of shared cloud computing resources regards performance-assured deployment of scalable cloud services. For example, many optimization algorithms have been proposed to reduce the consumption of cloud CPU, memory, and/or bandwidth resources (Zhou et al., 2013, Zhou et al., 2015, Zhou et al., 2014).

Experimental evaluation of a specific cloud computing algorithm can be done on a commercial public cloud platform (e.g., Amazon AWS, Google App Engine, IBM Cloud Platform, Microsoft Azure, etc.) or on an on-premises cloud. However, it is cost-inefficient to provision a large number of customized cloud resources to quantitatively analyze the performance tradeoffs of several different cloud resource management approaches in a real cloud datacenter environment (Zhou et al., 2013, Zhou et al., 2015, Zhou et al., 2014). Therefore, it is desirable for cloud computing researchers to be able to use open and credible simulation tools to objectively evaluate their work and to facilitate replicating their evaluations by others. Examples of such tools are CloudSim¹, GreenCloud², and MDCSim³. However, none of these tools provide a browser-based user interface for online cloud simulation and analysis tasks, nor a user-friendly interface for configuring simulation settings.

This paper presents design, implementation, and usage of the open-source-based cloud computing simulation tool WebCloudSim (Jinglin Li et al., 2015) with real use cases. Under the browser-server deployment model, the tool can be accessed for free via a browser at http://webcloudsim.org through the Internet. Most of the necessary computing tasks are performed by shared WebCloudSim servers. It provides an integrated support all of the necessary lifecycle activities for simulation-based comparative analysis of several different cloud computing algorithms in various simulated cloud datacenter environments, e.g., configuring the target cloud computing environment variables, charting simulation results, downloading simulator-generated data, etc. A WebCloudSim user could obtain nontrivial comparative analysis insights via the detailed resource consumption statistics WebCloudSim generates. Users of the tool can efficiently build their respective simulation models and repeat a large number of cloud simulation experiments.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 illustrates the design and implementation of WebCloudSim. Section 4 exemplifies how WebCloudSim can be used to generate factual experimental evaluation results. We conclude the paper in Section 5.

2. RELATED WORK
In this section, we briefly discuss the design principles and implementation differences of three state-of-the-art cloud simulation tools: CloudSim, GreenCloud and MDCSim.

¹ http://www.cloudbus.org/cloudsim/
² http://greencloud.gforge.uni.lu/
³ http://gcclab.org/tagc/research/data-center.html
CloudSim (Calheiros et al., 2011) is a discrete event simulation package developed via the SimJava function library, which inherits the GridSim programming modeling. Fig. 1 shows the three-layer framework of CloudSim: Simulation Core, Cloudsim, and User Code. Simulation Core is the bottommost layer, and implements the core functions CloudSim. The Cloudsim layer simulates the target cloud datacenter with support for custom configuration. It can concurrently instantiate and transparently manage many simulation runs targeted at various large-scale infrastructure-as-a-service (IaaS) clouds, which hosts virtual machines (VM) and other types of computing resources. The User Code layer hides the platform simulation and algorithm configuration details from the CloudSim user-level code, which focuses on fulfilling user requirements.

GreenCloud (Kliazovich, Bouvry and Khan, 2012) is a sophisticated packet-level simulator for energy-aware cloud datacenters with a focus on cloud communications. It is an open source tool developed mainly for analyzing the performance of a datacenter. Cloud simulation customization can be done via a set of GreenCloud built-in parameters and/or by modifying the GreenCloud code. The modeling entities it supports regard servers, switches, communication links, and energy consumption. It can be used to facilitate developing new cloud solutions (regarding system monitoring, resource allocation, workload scheduling) as well as optimizing communication protocols and network infrastructures. It is uneasy for computing novices to exploit the tool due to complexity of the tool’s implementation and configuration process as well as the required programming skills in C++ and Tcl. The tool’s output can be viewed via a web browser by running script files.

MDCSim (Seung-Hwan et al., 2009) is a discrete event simulator that supports business level requirements based upon a datacenter simulation system. One key feature of MDCSim is that its datacenter component allows for mixing different hardware models, each of which has its own characteristics profile. Moreover, the component is a three-tier platform comprising of communication layer, kernel layer, and user-level layer. The communication layer sends and receives messages between hosts in a server cluster. The kernel layer imitates the scheduling strategy in Linux core via a CPU allocation queue. The user-level layer comprises of web server, application server, and database server. The web server receives requests and tasks from MDCSim users or memory system, and delivers them to the nodes hosted by the application server.

From our viewpoint, major differences among CloudSim, GreenCloud, and MDCSim are summarized below. (1) During the simulation process, GreenCloud uses datagrams with protocol header to transfer messages between entities, instead of accomplishing it by transferring events. This approach greatly shortens its simulation time. (2) CloudSim and GreenCloud are free open source tools, and MDCSim is not. (3) A common big issue with these three simulators regards user friendliness of the GUI. Exploiting any of the simulators requires the user to first have an in-depth understanding of a non-trivial software package, followed by mastering an error-prone system configuration process. (4) When it comes to datacenter network, GreenCloud supports detailed modeling of the communication stack, which supports TCP/IP protocol, such as TCP and UDP, and imitates link failure and congestion. (5) GreenCloud supports the use of energy modeling to control energy consumption of servers, networks or switches, and conserve energy. CloudSim provides very limited support for energy control, and MDCSim only monitors energy consumption of servers. (6) CloudSim provides API support for implementing custom algorithms for VM scheduling and task allocation.

![Figure 1. The CloudSim framework](image-url)
WebCloudSim. Moreover, CloudSim provides the best support for implementing sophisticated algorithms (e.g., VM allocation policies and task allocation policies). It is convenient for researches to do thousands of experiments by extending or modifying CloudSim. Our decision on using JAVA to build our Web-based cloud computing simulator also makes CloudSim appealing. Finally, CloudSim is more extensible in terms of our requirements.

We note that when conducting a simulation experiment via CloudSim, the user has to extend the JAVA abstract classes of CloudSim. Moreover, CloudSim supports only one simple type of user task. The simulation results could not be presented well to the user in terms of quality and performance bottleneck of the algorithm under evaluation. Since the key simulation component must reflect reality, we need our simulator to be able to provide several types of dynamic task.

3. **Designed the WebCloudSim**

WebCloudSim aims to enable simulating, with high credibility, the target real cloud datacenter and the cloud computing algorithm under evaluation in that datacenter environment. It also aims to facilitate comparing one’s cloud computing algorithms with the classic ones and others’ as well as evaluating the algorithms using different cloud datacenter configurations by themselves. For these goals, we need to accomplish the tasks listed below:

- Accurately model the effects of the algorithm under evaluation as per the WebCloudSim user’s requests, with an adequate set of APIs and user-friendly UIs for the users (e.g., easy-to-use configuration UIs)
- Correctly model the entities of the datacenter with support for an adequate set of datacenter specification parameters
- Correctly model the behavior of the cloud datacenter with support for continuous running of datacenters
- Clearly display the simulation results

![Figure 2. User interaction structure](image)

Fig. 2 shows the user provides, among other input data, the algorithm file and the configuration settings for the target cloud datacenter, services and tasks. We use the inputs to build our cloud environment. Outputs are result charts, log, excel table, etc.

Fig. 3 shows the component structure of WebCloudSim. Before using the WebCloudSim online simulation tool, the user must first register with the tool a new id, and then sign into the system using that id. Each simulation run is associated with an experimental setup, which must be done via a four-step process: (1) compared algorithm uploading, (2) datacenter setup, (3) service setup, and (4) dynamic task setup. After all configuration tasks are done, the can start the simulation setup via the WebCloudSim’s browser interface.
After a simulation run completes, the results can be shown to the requesting user through a web browser, and can be downloaded by the requesting user as CSV-formatted data files.

Below is a summary of the functions of all of the WebCloudSim modules (shown in Fig. 3).

**User operation module:** This browser-facing module provides two main functions: receive input from the user and illustrate simulation results. The input message contains information on user id, uploaded algorithm, user-defined datacenter, and service configuration. Comparison of different algorithms’ simulation results can be illustrated in two forms: logs and line charts. The frame-based browser GUI is easy to use.

**Account management module:** This module manages user id registration, user id information, and user session (e.g., sign in and sign out processes). A user id is a validated email address of the user. Every WebCloudSim user must sign in with a valid user id and credential first before using the web-based cloud computing simulation tool (or service).

**Cloud environment configuration module:** This module provides a differentiating feature of WebCloudSim in terms of configurability of the target cloud environment. WebCloudSim can be used to compare the algorithm under investigation with other algorithms, which may have already been implemented per the WebCloudSim framework and stored in the WebCloudSim service system. This module comprises of four sub-modules: (1) algorithm uploading, (2) datacenter setup, (3) services setup and (4) tasks setup. WebCloudSim presently provides two types of algorithm: virtual machine allocation policy and task allocation policy.

Moreover, it provides several different algorithms for each of the policies. The user can choose to use the “built-in” algorithms by uploading theirs. The datacenter setup module configures the target cloud datacenter in WebCloudSim, including the number of hosts, the size of the processing units and so on. The service setup module mainly allocates computing tasks to the virtual machines in use. Default service setup settings can be overridden by the user. Finally, in support of dynamic tasks, the task setup module provides two types of distribution of resource utilization rate: Linear and Gauss distributions.

**Database:** This module stores and manages the operational data for the whole system. It has three duties. First, this module stores and manages the operational data for user accounts, including user id information and uploaded algorithms. Second, before a specific simulation run starts, it assures that information about the cloud environment configuration for the simulation is written into the data store it owns. Finally, this module collects various measurement data while the simulation runs. It acts as the core of the data flows in WebCloudSim because it communicates with the other modules on most data management related activities.

**Datacenter creation module:** This module activates simulation instances of various cloud environments per user requests (including the algorithms in use). It is in charge of creating entities and allocating resources. It establishes a simulated datacenter via abstractions for hosts, virtual machines, and network. It manages the allocation relation between virtual machines and hosts as well as between tasks and virtual machines.
**CloudSim engine module:** This module is an extended version CloudSim. We modified and extended the CloudSim source code per our requirements. This module executes the requested simulation runs per the information provided by the aforementioned modules. We changed the attributes and functions in CloudSim. We added VM Dependency part into CloudSim to support inter-VM dependencies (which enables the users to manage virtual machines and task execution by exploiting the important dependencies in practice). We also added some JAVA classes to support dynamic tasks and network. All of the data generated in a simulation run is stored into the WebCloudSim’s database.

**Data management module:** This module manages data generated in simulation runs. The operational data stored in the database are key-value pairs, recorded at important points in time during a simulation run. Since WebCloudSim needs to offer numerous forms to show simulation results, we use this module to facilitate displaying the results to the user. For example, using line charts to show the resource consumption differences of two algorithms, and displaying datacenter situations in detail via the simulation log. Users could download all the operational data they own from the database as CSV-formatted files.

### 4. Experiment

#### 4.1 Experimental setup

Compared to other cloud simulators, WebCloudSim could run remotely and be accessed by the user via a web browser. Thus, turnaround time of a simulation run could have little to do with the load of the browser machine in use.

As shown in Fig. 4, user could choose the algorithm for Virtual Machine Allocation and Task Allocation. Then the user must choose the compared algorithm and performance metrics. Finally, the user own algorithm needs be uploaded according to WebCloudSim file format rules.

As shown in Fig. 5, WebCloudSim users could set user-defined environments per their simulation needs. For each simulation run, a datacenter and a service must be specified. When adding a datacenter, the system sets some initial parameters by default (see Fig. 5(a)), such as the number of hosts and processing units, processing capacity (in terms of MIPS), storage capacity, and total amount of RAM. The default parameters of a service are divided into two parts, user description (see Fig. 5(b)) and virtual machine description (see Fig. 5(c)). The user description specifies the number of service, the number of tasks, and information about the tasks. The virtual machine description specifies the number of virtual machines, the image size, and RAM and network bandwidth capacity.

#### 4.2 Experimental Evaluation results

After a simulation run ends, the user can obtain the comparative experimental evaluation results of the two algorithms and download them as CSV-formatted files.

Fig. 6 and Fig. 7 exemplify the simulation-based experimental evaluation results of the two virtual machine allocation algorithms: First Fit (Algorithm 1 in the figures) and Random Fit (Algorithm 2 in the figures). Fig. 6 shows the CPU utilization of Algorithm 1 is higher than that of Algorithm 2 from 33min to 78min, and is lower than that of Algorithm 2 after 78min. Fig. 7 shows the RAM utilization of Algorithm 1 is lower than that of Algorithm 2 before 17min, and higher after 17min.
5. CONCLUSIONS AND FUTURE WORK
This paper presents the design, implementation, and usage of the current version of WebCloudSim, an open online cloud computing simulation tool for algorithm comparison. We aim to develop an education version as well as a government and enterprise version of WebCloudSim. We plan to extend WebCloudSim with support for multiple network topologies, e.g., fat-tree network topology and communication between different datacenters. We endeavor to advance service composition and selection research via WebCloudSim. Finally, we plan to enrich the simulating process with support for algorithms like particle swarm optimization, genetic algorithm, and mixed integer programming.

6. ACKNOWLEDGMENT
The work presented in this paper is supported in part by the NSFC (61202435), and the NSFC (61272521).

7. REFERENCES


His current research interests include network security, network intelligence and services.

Ao Zhou is an assistant professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She received her Ph.D. degree in computer science at Beijing University of Posts and Telecommunications in 2015. Her research interests include cloud computing, service reliability.

Shangguang Wang is an associate professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). He received his Ph.D. degree at BUPT in 2011. He is Vice Chair of IEEE Computer Society Technical Committee on Services Computing, President of the Service Society in China, General Chiar of ICCSA 2016, Application Track Chair of IEEE SCC 2015, Program Chair of the 2014 International Conference on Internet of Vehicles (IOV), and Program Chair of the 2014 International Symposium on Cloud and Service Computing (SC2). His research interests include Service Computing, Mobile and Cloud Computing.

Authors

Youqi Chen received his bachelor degree in computer science and technology from the University of Electronic Science and Technology of China in 2014. He is currently a master candidate at Beijing University of Posts and Telecommunications. His research interests including cloud computing.

Qibo Sun received the Ph.D. degree in Communication and Electronic System from the Beijing University of Posts and Telecommunication, in 2002. He is currently an Associate Professor at the Beijing University of Posts and Telecommunication in China. He is a member of the CCF.