
Ammara Sajjad, Aleena Aqdas Khan, Muhammad Aleem*

Accepted: 12/06/2018 DOI 10.18100/ijamec.2018241182

Abstract: Cloud computing is an emerging technology that offers pay-per-use IT services around the world. The daily increase in Cloud infrastructure and modern IT demands (such as web applications, scientific, and business) results in large-scale data centers that lead to additional electricity consumption. The high energy consumption causes a high operating cost and involves a large amount of carbon dioxide emissions that are harmful to the atmosphere. Many academic, governmental, and industrial studies show that energy consumed by computers and different communication units in a data center plays a significant role in the increase of data center operating costs. It is difficult to calculate the performance, security problems, and energy consumption in a true Cloud computing infrastructure. Therefore, in recent years, several simulation tools have been introduced to help researchers for the analysis of a Cloud computing environment. Cloud simulators need to perform simulation tests to reduce the various complexities of Cloud computing environments. Some Cloud simulators have been specially created to test the performance of Cloud computing environments. This study presents the comparative analysis of the state-of-the-art energy aware Cloud simulators. For comparison, several characteristics, functions, and models are employed for most famous energy aware Cloud simulators such as Green Cloud, CloudSim, iCanCloud, CloudAnalyst, NetworkCloudSim, and CloudReports.

Keywords: Cloud Computing, Cloud simulators, Energy-aware Cloud;

1. Introduction

Cloud computing is a technology that utilizes computing resources and Internet to provide scalable, on-demand, pay-as-you-go computing and storage services [1]. Cloud computing uses a large number of heterogeneously distributed resources to offer many services with unique requirements for Quality of Service (QoS) [2]. Some of the most prominent Cloud computing platforms are Microsoft Azure [18], Aneka, Amazon EC2 [17], Google App Engine [19], and GoGrid.

Generally, Clouds are classified as public, private, and hybrid Clouds [3]. In public Cloud model, a Cloud service provider offers Cloud services over the Internet. Public Cloud services are sold on demand, usually per minute or per hour basis. Customers only pay for the CPU, memory, and bandwidth they consume. Private Clouds and their infrastructure are owned and accessible by specific organizations [4]. By using a set of physical resources private Cloud provides computing power and other services within a specific organization or to the fixed small community of users. Moreover, users can only access the Cloud resources and services from a single organization, which provides more control and privacy. A hybrid Cloud is a combination of private and public Cloud [5]. Moreover, due to the availability issues of a single Cloud infrastructure, many efforts have been made towards multi-Clouds [6], which emphasize on the federation of different Clouds infrastructures [7, 8].

In addition, the most common services provided by the Cloud are classified as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [9]. In SaaS, users do not need to install software on their local machines [10]. PaaS provides a platform for customers who can develop their propriety applications on top of this platform [11]. IaaS provides an infrastructure for different purposes, such as storage systems, computing, and IT resources [12, 13]. Users of IaaS can adopt the available resources according to the required platform and application of their choice.

Cloud computing [14] focus on evolving data centers as a dynamic, scalable, and flexible platform for on-demand service and application delivery. This is done by revealing data-center capabilities as a network of virtual services for example user interface, database, hardware, and application logic so that the users can access and deploy applications over the Internet [15]. IT companies with new application service ideas are no need to make large expenditures on the hardware and software infrastructure. By using the Cloud as a platform for application hosting, IT companies are relieved of configuring basic hardware and software infrastructures. Therefore, for application service companies focus more on innovating and creating business values [16].

Some of the basic and developing Cloud-based application services in real time are content delivery, social networks, data processing, and web hosting. These applications have a different composition, configuration, and implementation requirements. In real Cloud computing environments such as (Microsoft Azure [18], Google App Engine [19], and Amazon EC2 [17]) and the evaluation of the provisioning policies for different applications under the temporary condition is very difficult. The use of real infrastructures, for benchmarking application performance (cost benefits, performance analysis) in variable conditions (workload, availability) is often limited by the inflexibility of the infrastructure. Therefore, in real infrastructures, an extremely difficult task is the reproduction of the reliable results. Furthermore, in extensive Cloud infrastructure reconfiguration of many experiments is a tedious and time-consuming task. The...
existing conditions in Cloud result in these limitations that are not under the control of application service providers. Therefore, benchmarking experiments cannot be performed in a repeatable, reliable, and scalable manner in a real Cloud environment. Therefore, many research efforts have been focused on developing Cloud simulators which are useful for benchmarking, analysis, and study the Cloud-related policies and techniques. In this work, we provide a comparative survey of the state-of-the-art in Cloud computing simulation. This study will aid researcher to select and employ the appropriate Cloud simulation framework for the experimentation and analysis of the proposed Cloud heuristics. The rest of this paper is organized as follows. In Section 2, a general description of Cloud simulation, an overview of Cloud simulators, a general and a policy based comparison of Cloud simulators is presented. Section 3 includes brief conclusive remarks about the Cloud simulators.

2. Cloud Simulation

Table 1. Cloud Simulators: a generalized comparison.

<table>
<thead>
<tr>
<th>Compare Factors</th>
<th>CloudSim</th>
<th>Green Cloud</th>
<th>iCanCloud</th>
<th>CloudAnalyst</th>
<th>CloudReports</th>
<th>Network CloudSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI Support</td>
<td>Limited</td>
<td>Limited</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Availability</td>
<td>Open Source</td>
<td>Open Source</td>
<td>Open Source</td>
<td>Open Source</td>
<td>Open Source</td>
<td>Open Source</td>
</tr>
<tr>
<td>Scalability</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Programming Language</td>
<td>Java</td>
<td>C++, otcal</td>
<td>C++</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
</tr>
<tr>
<td>Platform</td>
<td>SimJava</td>
<td>Ns2</td>
<td>INET, OMNET++</td>
<td>CloudSim</td>
<td>CloudSim</td>
<td>CloudSim</td>
</tr>
<tr>
<td>Energy-Aware</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Cloud Security</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Task Scheduling</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Resource management</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Migration Support</td>
<td>Auto Migr</td>
<td>Auto Migr</td>
<td>-</td>
<td>Auto Migr</td>
<td>Auto Migr</td>
<td>-</td>
</tr>
<tr>
<td>Topologies in data centers</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Simulation Model</td>
<td>Event Oriented</td>
<td>Packet Level</td>
<td>-</td>
<td>Event Oriented</td>
<td>Event Oriented</td>
<td>-</td>
</tr>
<tr>
<td>Accuracy</td>
<td>High</td>
<td>-</td>
<td>-</td>
<td>High</td>
<td>High</td>
<td>-</td>
</tr>
<tr>
<td>Services</td>
<td>IaaS,Paas,SaaS</td>
<td>IaaS</td>
<td>IaaS</td>
<td>IaaS, Saas</td>
<td>IaaS, Saas</td>
<td>IaaS</td>
</tr>
<tr>
<td>OS</td>
<td>Linux, Windows, MAC</td>
<td>Linux</td>
<td>Linux</td>
<td>Windows, MAC</td>
<td>-</td>
<td>Linux</td>
</tr>
<tr>
<td>Power Consumption Support</td>
<td>Limited</td>
<td>Full</td>
<td>Wip</td>
<td>Limited</td>
<td>Limited</td>
<td>-</td>
</tr>
<tr>
<td>TCP/IP Support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Graphical Output</td>
<td>Partial</td>
<td>Partial</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Supports Parallel Experiments</td>
<td>No</td>
<td>No</td>
<td>Wip</td>
<td>No</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>Communication Model</td>
<td>Limited</td>
<td>Full</td>
<td>Limited</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Energy Model</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Simulators should provide an easy-to-use Graphical User Interface that supports visual modeling [26]. Availability refers to a user's ability to access information or resources from a specific location and in the desired and correct format. Scalability indicates the enhanced operating speed of the simulator (with respect to the simulation time) and the amount of memory space that the simulator utilizes with the increased workload. Most of the Cloud simulators use programming languages such as Java and C++ for the implementation of simulation programs. Moreover, as most of these simulators are themselves implemented using these programming languages (i.e., C++, Java, etc.) and support
extensions to the simulator frameworks [26]. The platform features mentioned in Table 1 represent the implementation base of a simulator. Energy efficiency has become a major concern for Cloud computing platforms and data centers and can be considered at different levels such as energy-aware scheduling, resource management, etc. Cloud security refers to a broad set of policies used for the protection of data, applications, and incorporated Cloud computing infrastructure. Task scheduling determines the shared processing power between tasks and the available resources. In Cloud computing, resource management means the efficient usage of various geographically distributed resources [36]. Migration support (mentioned in Table 1) refers to the migrating possibilities of VMs from one computing resource to another [31]. A simulator may support multiple topologies and also allows the addition of new ones. Tree-based topologies are commonly used in Cloud data-center design [30]. Moreover, Table 1 also provides several other parameters such as simulation model that represents simulation granularity at an event or at packet levels. In an event-based simulation, communication takes place via message passing between different system entities. Packet-level simulation relates to the ad-hoc network supported network simulators such as NS-2 [37]. Accuracy in simulation defines the degree to which the results of an experiment are semantically correct. The Cloud services parameter defined in Table 1 is commonly categorized as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [9]. Cloud simulators are available for many operating system platforms as mentioned in Table 1. The quantification of network-related parameters requires the support of communication and networking protocols (such as TCP/IP) and several simulators provide such support (as shown in Table 1) [26]. The graphical output provided by the simulators is the only format results in a quick and easy understanding of the produced information as compared to the command line based [27]. The parameter Parallel experiment support (mentioned in Table 1) means that a particular simulator supports concurrent execution of the simulation on multi-processor machines [29]. Communication model is a very important requirement for a Cloud simulator to model and profile communication among data-centers and Cloud applications [28]. A Cloud simulator providing an energy consumption model facilitates to compare the energy consumption of different simulated algorithms and scheduling strategies. With the help of cost model, researchers can easily evaluate the cost-effectiveness of the new policies [28].

CloudSim [20] is a new, open-source, widespread, and flexible simulation framework [33]. It enables the modeling and simulation of Cloud computing infrastructures and emerging application services [20]. CloudSim is an event-based simulator which means that messages are sequenced as events in time. CloudSim provides limited support for the GUI and provides high simulation scalability. Moreover, it provides a limited energy based profiling for the executing simulations [29]. It performs task scheduling by using different scheduling techniques such as dynamic or static scheduling [50]. CloudSim supports auto-migration of VMs. In CloudSim BRITE format [40] is used to store the topology description that contains a number of network nodes. BRITE is a universal parameterized topology generation tool which can be used to study various properties of generated topologies such as power laws and average path length. CloudSim [20] provides highly accurate simulation results and supports modeling of IaaS, PaaS, and SaaS-based resource management policies. CloudSim is available on several platforms such as Linux, Windows, and MAC operating systems. To model communication in Cloud services, it supports TCP/IP based profiling [42]. CloudSim supports different energy consumption models such as Dynamic Voltage Frequency Scheduling (DVFS).

Green Cloud [35] is an energy-aware and packet-level simulator. Green Cloud is based on NS2 [37] simulation framework and focuses on simulating the communication between Cloud services at packet level [41]. In the same way as NS2, GreenCloud is written in C++ and OTCI [28], and this is a disadvantage of this simulator as two different languages must be used to implement one single experiment. GreenCloud is an open-source simulator that supports limited GUI [42]. This simulator can be used to model and simulate Cloud services at IaaS level. Green Cloud provides profiling of energy efficiency which is helpful in simulating green resource allocation and scheduling policies.

Green Cloud is available for Linux platform and supports modeling the auto-migration of VMs (transferring a VM across host machines in a Cloud). Green Cloud gives a wide range of power consumption profiling support; however, it does not support parallel simulations. To model communication, Green Cloud provides complete TCP/IP implementation [47] with partial support of GUI. The energy model provided by the green Cloud offers to model energy consumption for several communication devices in data-center such as a router, switches, gateways, and communication links [29]. Green Cloud fully supports modeling of power consumption which is the distinguished attribute of this simulator [47].

iCanCloud [29] is an open-source simulation platform that is designed to model and simulate Cloud computing systems [29]. Simulation of large storage networks is the core function of iCanCloud. iCanCloud provides an easy-to-use GUI which helps to manage, configure, and launch Cloud-based heuristic experiments. iCanCloud is one of the few simulators that provides a comprehensive GUI support [47]. A simulation in iCanCloud can be programmed using C++ [29]. iCanCloud provides scalable simulation at the expense of more consumed memory as compared to CloudSim. iCanCloud has been developed on the top of the INET and OMNeT++ [48] simulators. It is used to model IaaS level Cloud services and is available for Linux platform. Although iCanCloud has been designed to perform parallel simulations so that one experiment can be executed by spreading over several machines but this feature is not completely implemented yet. iCanCloud claims parallel experimentation and power consumption model as Work in Progress (WIP) [47]. For modeling of communication, iCanCloud supports TCP/IP based profiling of the execution entities.

CloudAnalyst [39] is an event-oriented graphical Cloud simulator and build on top of the CloudSim simulation framework [20]. CloudAnalyst supports visual modeling of Cloud-based resource allocation, scheduling policies, and large-scale Internet applications [49]. It is an open source simulator that is implemented in Java language. CloudAnalyst can generate graphical results of simulation in the form of tables and graphs, which is useful for efficiently summarizing a large amount of data collected during simulation [39]. CloudAnalyst is compatible with Linux, Windows, and MAC operating systems. It has no TCP/IP implementation available and provides a limited support for power consumption. Additionally, auto VM migration support is available in Cloud Analyst to model dynamic and fault-tolerance related Cloud services. There is a partial support of communication-related performance profiling in CloudAnalyst.

CloudReports [25] uses the CloudSim toolkit as underlying...
simulation framework [20]. Using the GUI support of CloudReports, researchers can easily model complex Cloud simulations. CloudReports is an open-source simulator [25] that provides the simulation at core Cloud service levels such as IaaS and SaaS. CloudReports supports limited power consumption modeling and is based on event-oriented simulation mechanism. Notable features of CloudReports include high accuracy and adaptive simulations based on auto-VM migration support. It provides Java Reflection API [25] to create extensions that are loaded as plugins.

The main cause of static power dissipation is current leakages and the dynamic power dissipation is related to the workload during CPU utilization [35]. CloudSim is released under Apache version 2.0 that is an energy and traffic aware simulator [20, 45]. CloudSim supports programming and simulation of both static and dynamic Cloud scheduling policies [44]. CloudSim is capable of managing scheduled processes based on both the space and time-shared VM scheduling heuristics. A space-shared policy assigns specific CPU cores to specific VMs and a time-shared policy dynamically distributes the capacity of a processor among VMs [20]. CloudSim provides VM's live migration. In CloudSim default VM allocation policy is First Come First Serve (FCFS) [40] and it provides four VM Migration policies (i) Minimum Migration Time (MRT) policy, (ii) Random Choice (RC) policy, (iii) Maximum Correlation (MC) policy, and (iv) Minimum Utilization (MU) policy [33]. DVFS policy can be utilized in CloudSim for the simulation of green scheduling heuristics [46]. CloudSim provides an implementation of SLA based policies and scheduling

Next, we compare these six Cloud simulators considering the features related to modeling of Cloud policies and heuristics (as shown in Table II).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Green Cloud</td>
<td>GPLA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>FCFS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CloudSim</td>
<td>Apache</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>FCFS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CloudSim Version 2.0</td>
<td>License</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>Time</td>
<td>Round</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Cloud Analyst</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Cloud Reports</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NetworkCloudSim</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Space</td>
<td>Space</td>
<td>-</td>
<td>FCFS</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>iCanCloud</td>
<td>GPLA version 3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Time</td>
<td>-</td>
<td>max</td>
<td>min, min-min</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 2. Cloud simulators: policies based comparison**

A Cloud scheduling algorithm can be classified into static or dynamic. In static scheduling, task priorities and allocations are pre-determined before execution of the simulation while in dynamic scheduling tasks priorities are determined at runtime. VM migration is used to migrate a VM between host machines within a data-center. VM migration is useful for implementing dynamic or adaptive policies and to support fault tolerance. Generally, there are two types of migrations: [34, 1] regular or cold migration and 2) live migration. In live VM migration, a VM switches from one host machine to another while continuing its execution resulting in no obvious effect on end-user performance [35]. The cold or regular migration involves halting execution of a VM before migration to the destination host in a Cloud data-center. The scheduling policies at VM level (allocation of processes to VMs) are comprised of two sub-policies: 1) space-shared and 2) time-shared [20]. VM allocation is the process of creating VMs on hosts. In this allocation, hosts and VMs should match the important characteristics (memory, configurations (software environment), and requirements of the SaaS provider [33]. The VM selection problem defines that which VM should be migrated to a new host machine with respect to power and other Cloud policy considerations [33]. The Service Level Agreement (SLA) is a contract negotiated between a customer and a service provider which defines all the services that must be provided [32]. Power consumption has two types, static and dynamic power dissipation.
heuristics. Green Cloud [35] is released under General Public License Agreement (GPLA). Green Cloud is a traffic and energy-aware simulator. It supports both static and dynamic scheduling [44] heuristics with live VM migrations. Green Cloud implements specifications based on VM scheduling policies. It provides Load-Aware Allocation (LLA) VM allocation policy [35], by default VM allocation policy in Green Cloud is FCFS. Green Cloud performs a heuristic search for VM selection. Power saving mode in Green Cloud is based on DVFS model. Green Cloud also provides an implementation of SLA agreed on Cloud policies. In Green Cloud, both static and dynamic power consumption policies can be accommodated for the executing simulation.

iCanCloud [29] is released under General Public License (GPL). It supports time shared VM scheduling policy. iCanCloud implements two default resource provisioning and scheduling algorithms (i.e., Min-Min and Max-Min) [29]. In CloudAnalyst [39], the default VM allocation policy uses a round robin algorithm. By using this algorithm all requests are assigned to the available VMs in a round robin manner, regardless of the current load on each virtual machine [39]. SLAs are met at a minimal cost for consumers in CloudAnalyst. A time-shared policy is used to resource scheduling for virtual machines in CloudAnalyst. Resource provisioning is a configuration for extensive applications in which resources are dynamically assigned by geographic location based on workload [39].

CloudReports [25] is an energy-aware simulator. This simulator provides resource utilization models, VM allocation policies, VM schedulers, which can be further extendable for the customized user needs.

NetworkCloudSim [38] implements time-shared VM scheduling policy. Default VM allocation policy in this simulator is based on FCFS [38].

3. Conclusions

To ensure Green Cloud computing environment many efforts have been conducted to reduce the worldwide energy crisis to some extent. Researchers have used Cloud computing simulators for testing and evaluating these techniques. In this paper, we compare Cloud computing simulators named as Green Cloud, CloudSim, iCanCloud, CloudAnalyst, CloudReports, and NetworkCloudSim on the basis of different parameters. We have found that different simulators have distinct characteristics, features, and developed to meet specific requirements of the researchers. For energy aware simulation Green Cloud is the most suitable simulator. Green Cloud has many advantages over other Cloud simulators where energy consumption is the key concern. Green Cloud is based on the popular NS2 network simulator, which permits it to keenly observe packet-level details. Green Cloud accurately calculates the energy consumption because energy consumption is computed on packet levels whereas the CloudSim does not provide a comprehensive energy model for packet level simulation. CloudSim can be more beneficial when a precise analysis of complex resource allocation policies and scheduling heuristics is required. iCanCloud claims parallel experimentation feature in progress and yet not provided by any other simulator. Moreover, iCanCloud provides full GUI based interface support. CloudAnalyst supports the evaluation of social network tools employing the geographically distributed data-centers and users. CloudAnalyst also provides results in full graphical mode based on several types of visualizations (such as Graphs, charts etc.)

CloudReports is one of the unique simulators to simulate and evaluate the cost of implementation, the degree of resource utilization, and power consumption of the simulated algorithm. Finally, we observe that most of the Cloud computing simulators lack features related to analyzing security issues in Cloud.

References


