RESEARCH   **Open Access**

CrossMark

Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications

Khadijah Bahwaireth[1], Lo'ai Tawalbeh[1,2], Elhadj Benkhelifa[3], Yaser Jararweh[2*] and Mohammad A. Tawalbeh[3]

Abstract

Cloud computing provides a convenient and on-demand access to virtually unlimited computing resources. Mobile cloud computing (MCC) is an emerging technology that integrates cloud computing technology with mobile devices. MCC provides access to cloud services for mobile devices. With the growing popularity of cloud computing, researchers in this area need to conduct real experiments in their studies. Setting up and running these experiments in real cloud environments are costly. However, modeling and simulation tools are suitable solutions that often provide good alternatives for emulating cloud computing environments. Several simulation tools have been developed especially for cloud computing. In this paper, we present the most powerful simulation tools in this research area. These include CloudSim, CloudAnalyst, CloudReports, CloudExp, GreenCloud, and iCanCloud. Also, we perform experiments for some of these tools to show their capabilities.

Keywords: Cloud computing, Modeling and simulation, Mobile cloud computing, Performance evaluation

1 Introduction

Nowadays, smart devices have become more popular and serve users in various ways. Using enhanced capabilities, smart devices have proved to be useful, providing tools that are ready anywhere and anytime, to serve a user's needs regardless of their location. Cloud computing integrates several scientific models and distributes resources over the network, thus, resolving the issue of massive computations being done on only one site. Additionally, cloud computing is based on a "pay-as-you-go" model.

Cloud computing is defined as an internet-based model, which provides on-demand access to a network of computing resources (e.g., networks, servers, storage, applications, and services), with minimal management effort [1]. The integration of cloud computing and mobile computing yields a new technology called mobile cloud computing (MCC). MCC technology provides access to cloud services for mobile devices, providing a significant number of advantages such as saving mobile

battery lifetime [2], improving data storage capacity [3], and improving reliability and security by storing data and applications on many servers [2].

Where researchers in cloud computing need to conduct real experiments in their studies, experimentations in cloud environments often come at a high cost quotient, for both setup and implementation. One suitable alternative is the use of models and simulation tools to emulate cloud environments and conduct the necessary experimentation. CloudSim, CloudAnalyst, CloudReports, CloudExp, GreenCloud, and iCanCloud are some of the most powerful simulation tools for cloud computing environments which enable researchers to model multiple complex scenarios.

In the remainder part of this paper, we focused on the abovementioned simulators. In Section 2, we review related works. In Section 3, we present detailed descriptions of the aforementioned simulators and conduct experiments for their efficacy. A comparison between the major cloud computing simulation frameworks is provided in Section 4, with a conclusion following in Section 5.

* Correspondence: yaser.amd@gmail.com
[2]Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan
Full list of author information is available at the end of the article

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 2 of 14

2 Related work

Recently, several research works have been done in the area of mobile cloud computing, for instance, authors in [4] provided a survey on mobile cloud computing and presented taxonomy of current approaches in MCC based on the critical issues in this area. Other authors [5] described the originality of mobile cloud computing and how it combines both cloud computing and device computing. They also reviewed the characteristics of MCC including some recently published research, in addition to discussing the challenges of working in this area and some proposed solutions. Benkhelifa et al. [6] attempted to provide an efficient green MCC platform by proposing a system that generates a profile to determine the users' power usage, based on their interaction and component usage. On the basis of evidence in the abovementioned works, it seems reasonable to suggest that many of these simulation tools were designed for analyzing different types of systems, and to study and observe the operations of each system. As suggested by Anu [7], simulation is a way of evaluating a proposed system for various parameters within a specific period of time [7]. The following are brief examples of simulation tools, they will be discussed in detail in the Section 3.

- SimGrid [8] is a simulation tool for grid computing and distributed systems that studies scheduling algorithms for distributed application.
- Buyya et al. [9] designed a simulation tool called CloudSim for cloud computing experiments. It supports modeling and simulation of cloud computing environments, and allows the simulation of multiple data centers.
- TeachCloud [3] is an extension of CloudSim framework and built to be convenient for teaching purposes. This tool adds a GUI along with several new key features.
- iCanCloud [10] is a simulation platform for conducting large-scale experiments in cloud computing. It can determine cost per user by predicting the trade-offs between cost and performance for each application executed on a particular hardware [11].
- Jararweh et al. [12] designed CloudExp tool to overcome certain limitations of the CloudSim tool. Its efficient GUI is built on top of CloudSim. In addition, this simulator is used to evaluate and analyze cloud components such as processing elements, data centers, storage platforms, and networking. It integrates most of the technologies related to cloud computing such as big data management and mobile cloud computing. CloudExp introduced a specialized modeling and simulation environment for mobile cloud computing [12, 13]
- CloudAnalyst [14] is an extension to CloudSim, and used to simulate a large-scale internet application on the cloud.
- CloudReport [15] is a GUI CloudSim-based tool with an easy-to-use interface.
- DARTCSIM [16] was also built based on CloudSim. It hides simulation details and provides friendly GUI for users to conduct their experiments. In addition, it allows the user to alter simulation parameters easily [16].

It is necessary to mention that some of these simulators are open source, giving opportunity to researchers to develop bespoke code extensions.

3 Cloud simulation tools

In this section, we have conducted a detailed study of commonly used cloud computing simulation tools and discussed their main features.

3.1 CloudSim

CloudSim project has found widespread use in both research-based cloud development and industry-based cloud development. It provides a generalized framework to apply experiments, according to specific design issues that can be seamlessly set to the offered generalized framework [9].

With its earlier versions based on GridSim, CloudSim was developed by the GRIDS laboratory of the University of Melbourne to model and simulate several cloud computing infrastructures. In its latter versions and in-depth re-implementation, CloudSim includes a full implementation of its own kernel to rid itself of reliance on GridSim [17].

Conceptually, CloudSim on one side offers classes representing data centers, physical hosts, virtual machines, services to be executed in the data centers, users of cloud services, internal data center networks, and energy consumption of physical hosts and data centers elements. On the other side, CloudSim supports dynamic insertion of simulation elements and provides message-passing application and data center network topology [9, 18].

In addition, CloudSim supports the modeling and simulating of different cloud computing environments, as well as simulation of many data centers. Furthermore, evidence in literature suggests that CloudSim is capable of instantiating 100,000 machines in less than 5 min [11]. It further enables studying on allocating policies and migrations of virtual machines (VMs) rendering its very useful in the research field for many researchers [19–21]. CloudSim's architecture consists of specific

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 3 of 14

entities that are represented as Java classes, and as such, developers can simulate their experiments by inheriting these classes or directly by changing the available methods or parameters easily. Figure 1 below illustrates the main components of CloudSim, with each described below.

Figure 1 represents the main entities in CloudSim [22]:

1- Data centers (DC): the resource provider—includes one or more hosts.
2- Host: the physical machine that allocates one or more VMs).
3- Virtual machine (VM): machines on which the cloudlet will be executed.
4- Cloud Information Service (CIS): responsible for registering all resources of data centers.
5- Broker: when a broker has the DC characteristics, it will submit VMs to the specific host in the specific DC and then allocate the cloudlets (tasks) to specific VMs. Finally, the broker will destroy the free VMs after execution of all cloudlets.
6- Cloudlets: In CloudSim, cloudlets are all tasks and applications that are executed on VMs.

These entities interact with each other to perform the simulation: Cloud Information Service (CIS), broker, data center, host, virtual machines (VM1, VM2), and cloudlets (C1, C2 ….).

Each simulation in CloudSim follows basic steps in java code from configuration to simulation. These are as follows:

1- Setting the number of users for a current simulation.
2- Initializing the simulation by instantiating the common variables (current time, trace flag, number of users).
3- Creating CIS instance.

4- Creating data center instance and then registering it with CIS.
5- Creating physical machines (hosts) with their characteristics.
6- Creating data center broker instance.
7- Creating VMs with their characteristics.
8- Submitting VMs to data center broker.
9- Creating cloudlets and specifying their characteristics.
10- Submitting cloudlets to data center broker.
11- Sending a call to start the simulation once there is an event to be executed.
12- Sending a call to stop the simulation once there is no event to be executed.
13- Printing the results of the simulation.

CIS registers all data center resources before they are used, where upon each data center contains one or more hosts. In addition, each host allocates one or more virtual machines which run the cloudlets, while the broker is responsible for submitting cloudlets to the specific data centers. Power consumption in CloudSim is described by a linear relationship between the power consumption and CPU utilization. As such, CloudSim provides several power models, such as PowerModelSqrt, PowerModelLinear, PowerModelSquare, and PowerModelCubic, all based on the linear relationship between power consumption and CPU utilization [23].

However, despite its said advantages, CloudSim still has its limitations, for instance, (1) the lack of a GUI—CloudSim does not have the ability to produce results in a graphical format; (2) CloudSim only includes a basic network model with a limited workload traffic generator [12]; (3) CloudSim also lacks support modeling for parallel experiments [12]; (4) lack of some essential cloud computing components such as BPM and SLA [12]; (5) analyzing simulation results in CloudSim is difficult, so a developer might need to analyze data using other tools such as Excel.
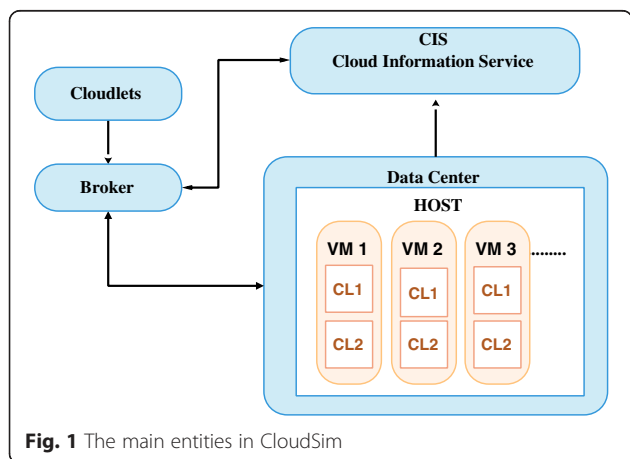
3.1.1 CloudSim experiment

The following scenario was assumed to test the capability of CloudSim for modeling and simulation. The objective of this scenario is to calculate the power consumption (watt) for a cloud data center. We assumed that the simulation environment consists of one data center with two hosts (physical machines). Each host has one CPU core (with computing capacity 1000 MIPS), 2 GB of RAM memory, and 1 TB of storage.

In this simulation, we have four VMs running on the host, with the following constraints: one CPU core, with 500 MIPS computing capacity, 512 MB RAM memory, and 10 GB of storage. A default policy allocates each VM to the first available host with the maximum

Fig. 1 The main entities in CloudSim

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 4 of 14

number of free processing elements (Pes) based on FCFS. Each host dynamically distributes/shares its capacity of a core among all VMs running concurrently (time-shared VM scheduler).

The environment is composed of ten cloudlets (task units), where upon each instance of cloudlet requires 400,000 million instructions, or about 6 min to complete execution on a host. Figure 2 shows the power consumption results of the simulation produced by CloudSim.

Figure 3 shows another power consumption results for simulating different configurations. It has only one difference, that is, the number of cloudlets. It is 100 cloudlet units.

Total power consumption of cloud data center is defined as the sum of power consumption for all nodes. The default power model in CloudSim is Linear power model as described in (1) [18].

$$P(u) = k * P_{\max} + (1 - k) * P_{\max} * u \qquad (1)$$

where k is the idle host power consumption, $P_{\max}$ is the maximum power consumed by the host in full utilization mode, and u is the CPU utilization [18]. However, CloudSim provides some power models that generate power consumption values by implementing linear relationship between power consumption and CPU utilization [23].

3.2 CloudAnalyst

CloudAnalyst is developed at the GRIDS laboratory at the University of Melbourne and designed for some specific goals. What is so special about this simulator is its graphical presentation of output, and the repetition of simulations to suit the user's requirements. As such, CloudAnalyst is beneficial where the main objective is to simulate distributed applications among several data centers and user groups. It inherits the original features

of CloudSim framework, extending it to simulate large-scale Internet applications and study their behaviors in cloud environments. Furthermore, it supports the evaluation of social network tools, based on the geographic distribution of users and data centers. It is our opinion that internet application users care more about the time needed to process their requests, regardless of the server location. When a user base (a group of users) requests cloudlets (tasks), a Service Broker entity in CloudAnalyst routes user base traffic to the best data center, according to a server brokerage policy. CloudAnalyst implements three brokerage policies; service proximity policy, best response time policy, and dynamically reconfiguring router [14].

CloudAnalyst domain model consists of the following:

1. Internet: an abstraction of the real word internet. Suitable bandwidth and transfer delays are introduced and are configurable for all six regions of globe.
2. Region: the world is divided into six different regions, where upon the user bases (UBs) and DCs belong to one or more of any one of these regions.
3. User base: it models the group of users who are considered as a single unit in simulation and generate traffic for the simulation, these can be a single user or a large number of users.
4. InternetCloudlet: grouping of request form users. In advance tab, requests can be bundled as a single InternetCloudlet, with information such as the size of request execution command and input output files.
5. CloudAppServiceBroker: In CloudAnalyst, the Service Broker controls the traffic between DCs and UBs and retains the authority to decide which data center services a request from the user base.
6. Data centre controller: most important entity in the CloudAnalyst domain model. It is the front face used by the CloudAnalyst to access the heart of CloudSim toolkit functions.
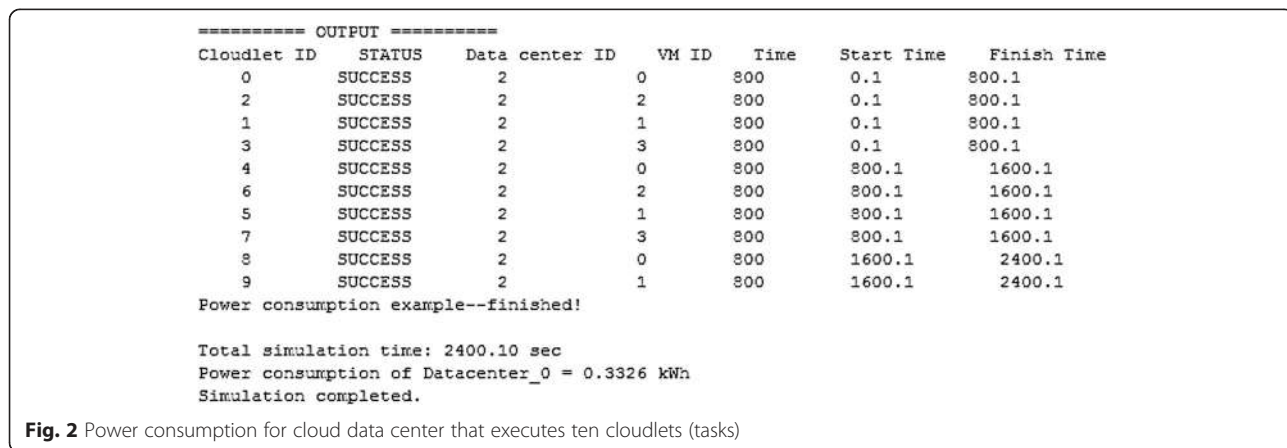
```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
       0       SUCCESS        2              0       800       0.1           800.1
       2       SUCCESS        2              2       800       0.1           800.1
       1       SUCCESS        2              1       800       0.1           800.1
       3       SUCCESS        2              3       800       0.1           800.1
       4       SUCCESS        2              0       800      800.1         1600.1
       6       SUCCESS        2              2       800      800.1         1600.1
       5       SUCCESS        2              1       800      800.1         1600.1
       7       SUCCESS        2              3       800      800.1         1600.1
       8       SUCCESS        2              0       800     1600.1         2400.1
       9       SUCCESS        2              1       800     1600.1         2400.1
Power consumption example--finished!

Total simulation time: 2400.10 sec
Power consumption of Datacenter_0 = 0.3326 kWh
Simulation completed.
```

Fig. 2 Power consumption for cloud data center that executes ten cloudlets (tasks)

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 5 of 14


```
95        SUCCESS        2        3        800        18400.1        19200.1
96        SUCCESS        2        0        800        19200.1        20000.1
98        SUCCESS        2        2        800        19200.1        20000.1
97        SUCCESS        2        1        800        19200.1        20000.1
99        SUCCESS        2        3        800        19200.1        20000.1
Power consumption example--finished!

Total simulation time: 20000.10 sec
Power consumption of Datacenter_0 = 2.7771 kWh
Simulation completed.
```

Fig. 3 Power consumption for cloud data center that executes 100 cloudlets (tasks)

7. VmLoadBalancer: virtual machine load balancer is used in data center controller to select the virtual machine to assign the next cloudlet for processing. There are three different load balancers used—round robin, equally spread current execution load, and throttled load balancer.
8. GUI: this is the set of different screens that allow a user to define the simulation parameters and save, load saved configurations, and execute or cancel simulation.

CloudAnalyst includes new powerful features such as an easy-to-use GUI, the ability to separate a simulation from the programming code, quick simulation setup, and an enhanced graphical results display, featuring useful formats such as tables and charts [14]. Installing CloudAnalyst is very simple and time efficient.

The main window of CloudAnalyst lets the user configure the simulation by determining data centers, hosts, VMs, and user base that represent the number of users (e.g., 1000 users), as shown in Fig. 4. The user base sends requests that are executed according to a routing policy in a specific data center.

When the simulation completes, the output panel displays the response time for each user base. The simulator provides a detailed screen with results that include response time for each user, request servicing time by each data center, and a number of all requests serviced by the data center. All of these results are displayed in tables and charts to simplify the analysis as shown in Fig. 5.
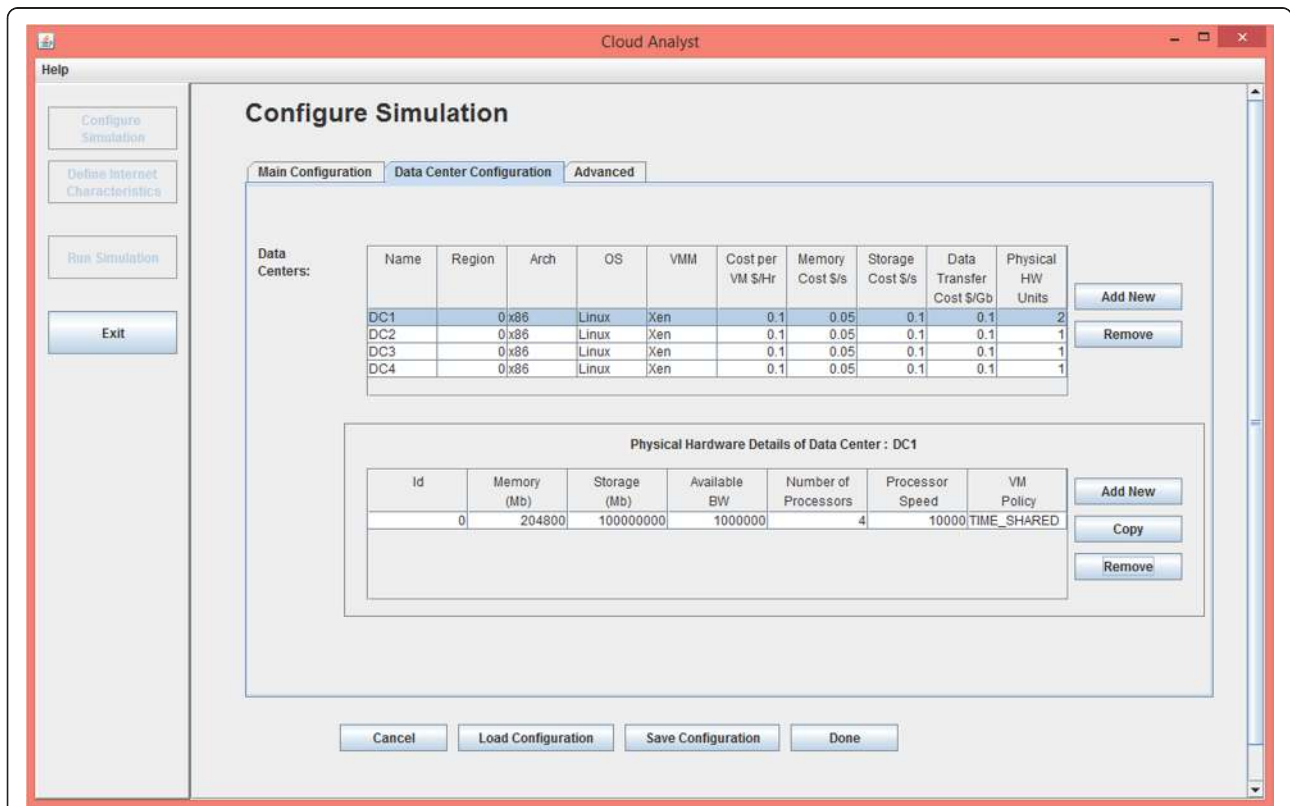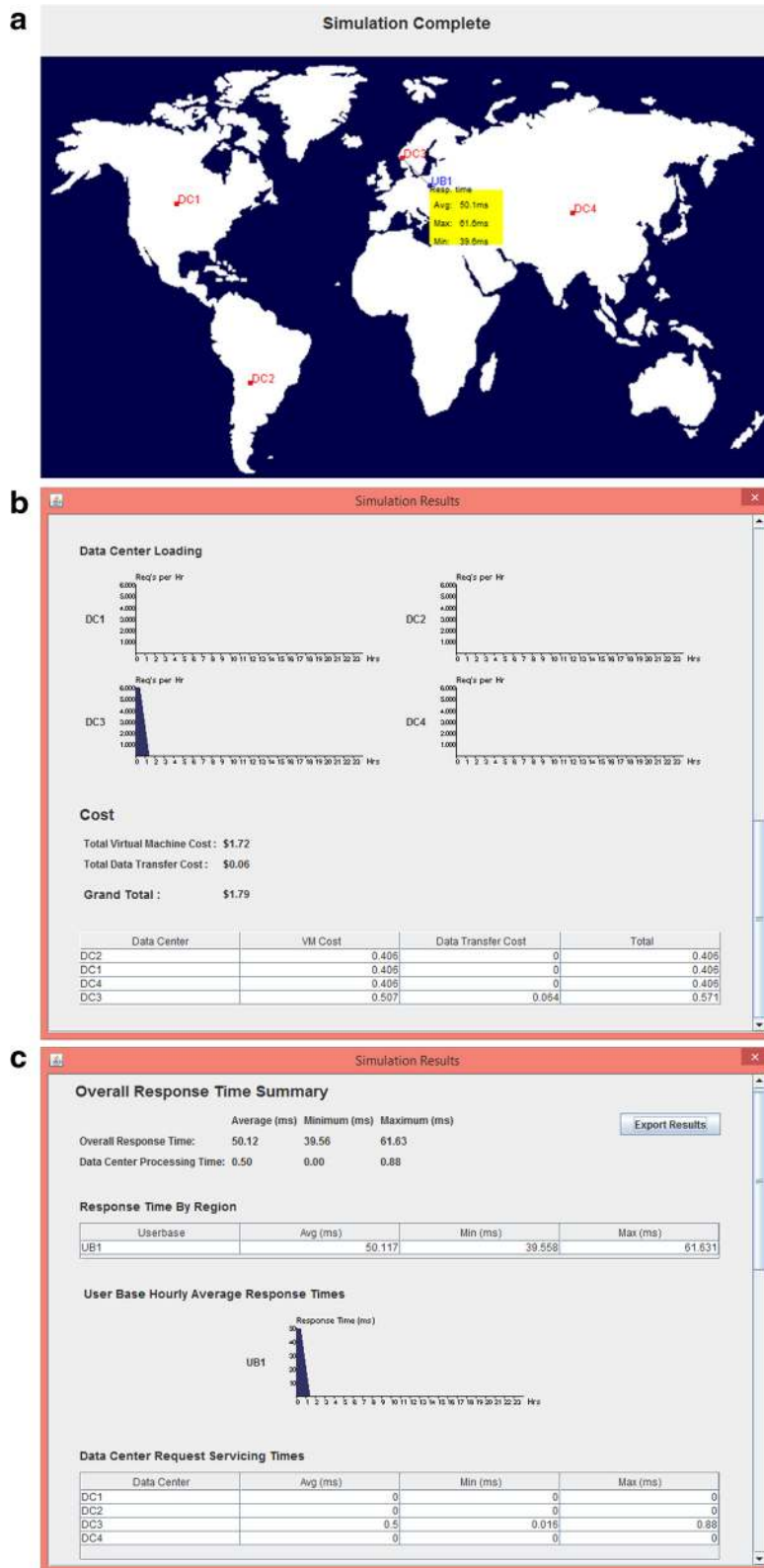

Fig. 4 CloudAnalyst configuration panel

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 6 of 14

Fig. 5 a CloudAnalyst simulation result window; **b**, **c** detailed simulation result window

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 7 of 14

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	0	60	100	13	15	400000	40000
UB2	1	60	100	15	17	100000	10000
UB3	2	60	100	20	22	300000	30000
UB4	3	60	100	1	3	150000	15000
UB5	4	60	100	21	23	50000	5000

Data Center	# VMs	Image Size	Memory	BW
DC1	20	10000	1024	1000
DC2	30	10000	1024	1000
DC3	40	10000	1024	1000

Fig. 6 User base and data center configurations

Evidence in research suggests that many authors have indeed used CloudAnalyst to conduct experiments. For instance, the authors in [24] used CloudAnalyst to contrast various load balancing policies. A service broker strategy also is proposed in [25] which routes the user requests to the cost-effective data center.

3.2.1 CloudAnalyst experiment

In this experiment, we assume that there is an internet application that is located in three data centers around the world. The cloud environment consists of six user bases which are geographically distributed and three data centers with 20, 30, and 40 VMs as shown in Fig. 6 below. The user base traffic is routed to the data center according to the service proximity policy that selects the closest data center in terms of the lowest network delay [14]. The duration of this simulation is 1 h.

After the simulation is completed, CloudAnalyst shows the regions with response to time results. This is shown in Fig. 7.

It is also possible to analyze the results by exporting a .pdf file that includes detailed charts and tables. Tables 1 and 2 are illustrative screenshots of a request servicing time for each data center and cost details, respectively.

3.3 CloudReports

CloudReport is another tool which is built upon CloudSim's framework (see Fig. 8 for CloudReport architecture). However, CloudReport enhances many features
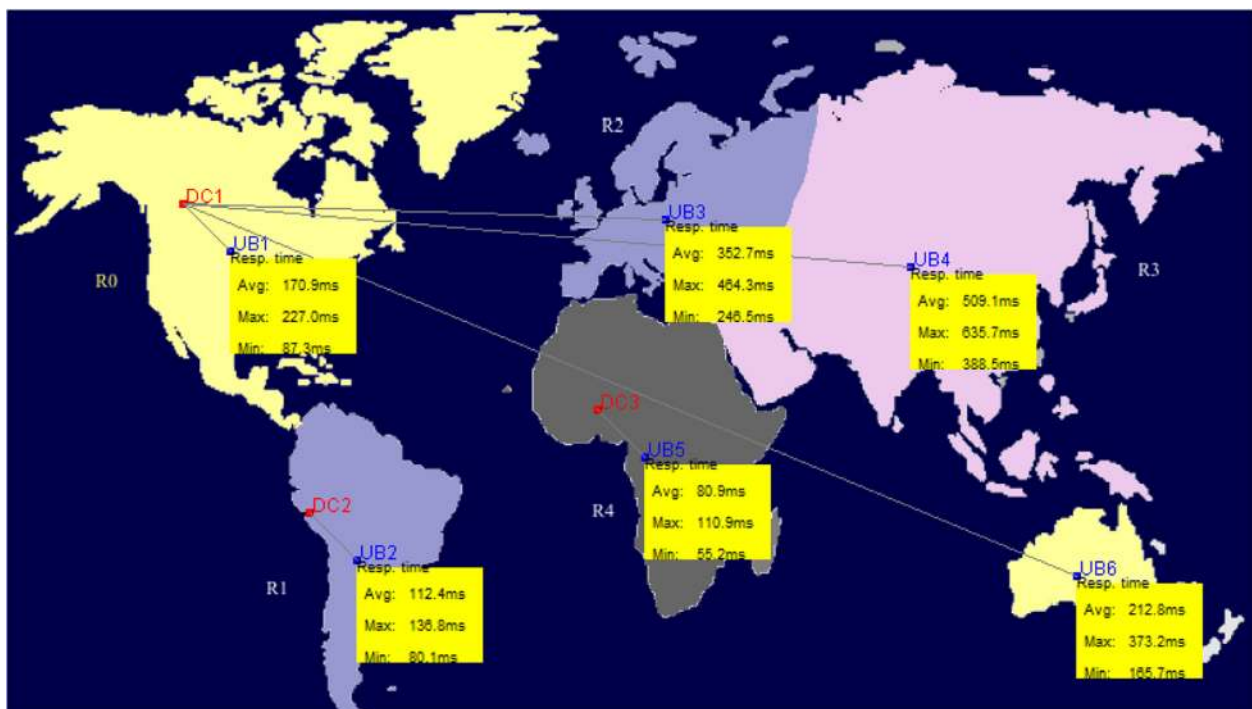

Fig. 7 A view of geographical response time results

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 8 of 14

Table 1 Data center request servicing time

Data center	AVG (ms)	Min (ms)	Max (ms)
DC1	69.62	1.94	168.53
DC2	61.58	33.47	81.84
DC3	30.27	12.50	50.96

of CloudSim such as a friendly graphical user interface, running multiple simulations at the same time, and enhanced simulation results. In addition, CloudReport benefits from the use of Application Programmable Interface (API) for creating specific extensions to simulate specific algorithms.

When simulation completes, CloudReport results are shown in tables and charts with information about execution time and resource usage costs. It also provides information about VM allocation, energy consumption, and any other user-defined characteristic [15].

3.3.1 CloudReports experiment

To demonstrate the capabilities of CloudReport, in this experiment we assumed that the simulation environment consists of one data center and two hosts (physical machines). Each host has one CPU core (with computing capacity of 1000 MIPS), 2 GB of RAM memory, and 1 TB of storage. In this simulation, we have four VMs running on the host with the following computing capacities 500, 1000, 500, and 1000 MIPS. These VMs have 512 MB of RAM memory and 10 GB of storage. To initiate this experiment in CloudReports, we configure the data center, change the host configuration, and then add user proprieties, as shown in Fig. 9.

When the simulation completes, CloudReports generates the results and represents them as HTML reports. These include charts for representing resource utilization, execution time, power consumption, and the overall resource utilization of each data center. Figure 10 is generated using CloudReports and shows the number of cloudlets which have been successfully executed by the current user on the data center.

Figure 11 is generated also by CloudReports and shows the total time in seconds for each cloudlet that is executed successfully in the current data center.

3.4 CloudExp simulator

CloudExp is a modeling and simulation environment which introduced a specialized mobile cloud computing

Table 2 Cloud environment cost results

Data center	VM cost $	Data transfer	Total $
DC2	3.01	5.91	8.92
DC1	2.01	53.40	55.41
DC3	4.01	2.98	6.99
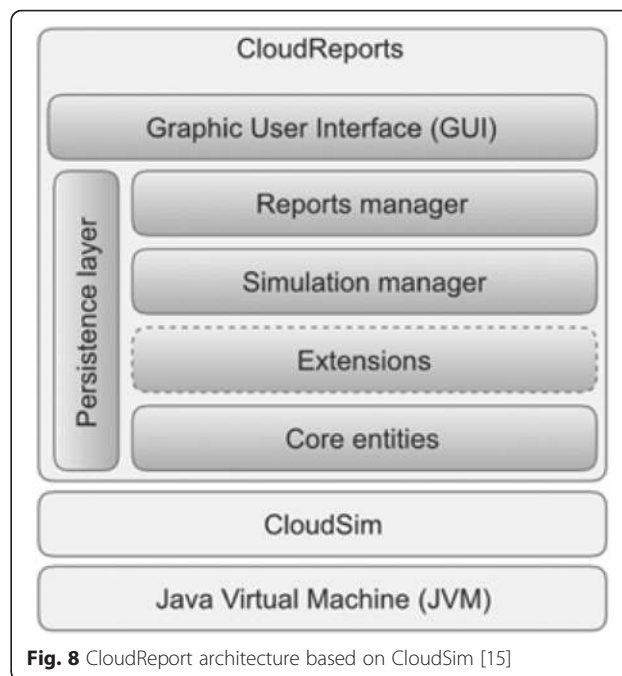
Fig. 8 CloudReport architecture based on CloudSim [15]

experimental framework [12, 13]. It is built on top of "CloudExp" framework with several new features. It provides user-friendly GUI to enhance the users' experience in building their own infrastructure. This simulator allows researchers to study the communication cost between users and clouds. In CloudExp, cloudlets are created using a simple drag and drop. Additionally, CloudExp conducts various mobility scenarios for mobile devices.

In CloudExp, the users need to configure the workplace properties and then decide the number of users, number of cloudlet in that region, the cloudlet type (fixed or mobile), and other properties of cloudlet; these are shown in Fig. 12 When the simulation completes, CloudExp results are saved as Excel sheets to simplify the analysis of results. CloudExp used to simulate different types of configurations with different objectives. The work in [26] presented a multi-agent-based dynamic resource provisioning and monitoring for cloud computing system infrastructure. CloudExp is extended to support the multi-agent approach for resource management. An evaluation study for map-reduce task scheduling algorithms over cloud computing infrastructure is presented in [27]. The CloudExp tool is extend to implement a number of map-reduce tasks over a cloud-based infrastructure. Evaluating a large-scale health system is presented at [28] which aims at supporting global health awareness. CloudExp is used to evaluate large-scale cloudlet deployment in [29]. An integrated framework for large-scale mobile cloud computing environment is presented in [20].
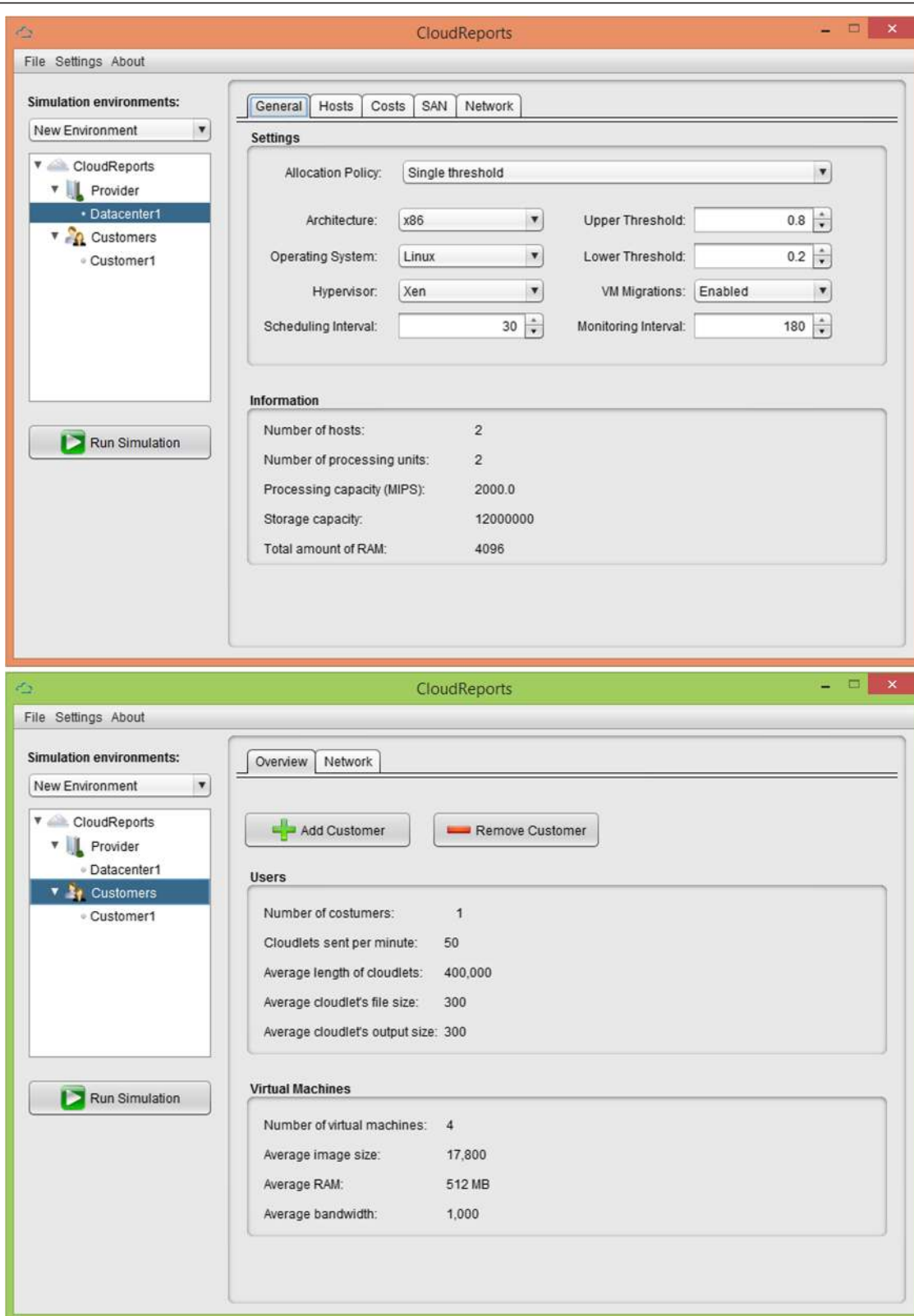
Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 9 of 14

Fig. 9 Add/configure data center, hosts, user properties in CloudReports

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15
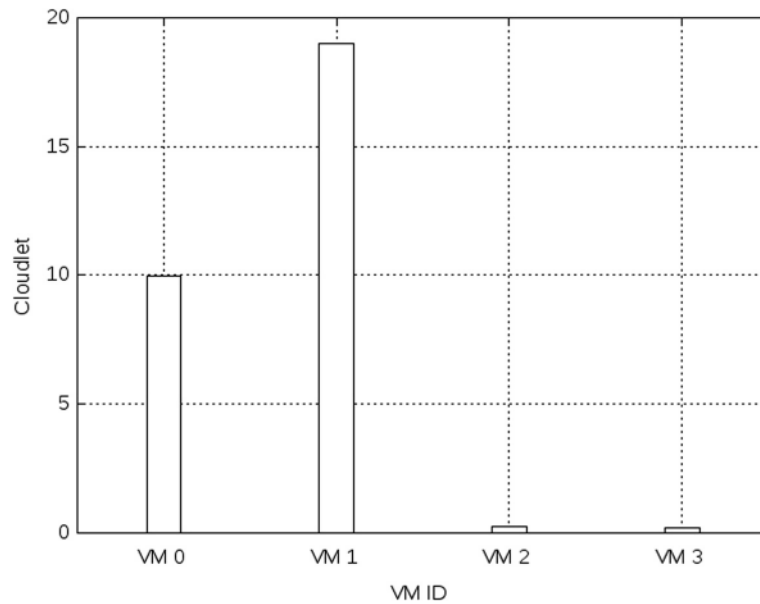
Page 10 of 14

Fig. 10 Number of successful cloudlets executed in each VM in the current data center

3.4.1 CloudExp simulator experiment

The goal of the following experiment is to study the user load changing behavior on CloudExp power consumption and delay. We study the power consumption for each mobile user in CloudExp by looking at the impact of load size on the performance of power and delay. The load produced by each user is measured by packet rate per second. By using this simulator, we conducted ten separate experiments for both heavy and light user loads. The following table summarizes the experiment's configuration in CloudExp (Table 3).

All users move in a closed area, and each of them sends a packet to the cloudlet using Wi-Fi or cellular technology with 2 m/s speed. In each experiment, the cloudlet is incremented from one to ten cloudlets, with the maximum number of users in each cloudlet set to 500.

This experiment shows performance results of packet transmission power for each user in MCC, for both heavy and normal user loads. From Fig. 13, we can observe that the power consumption and delay values for each user in the heavy load are higher than those in the light load. This is an expected outcome since heavy loads such as 400 packets are transmitted in 1 ms, whereas light loads such as 100 packets are transmitted in less than 1 ms. Furthermore, both transmission power and delay can be reduced by increasing the cloudlets. A detailed account of the foregoing is provided in [12, 13]. Finally, in contrast to other simulators, CloudExp seems
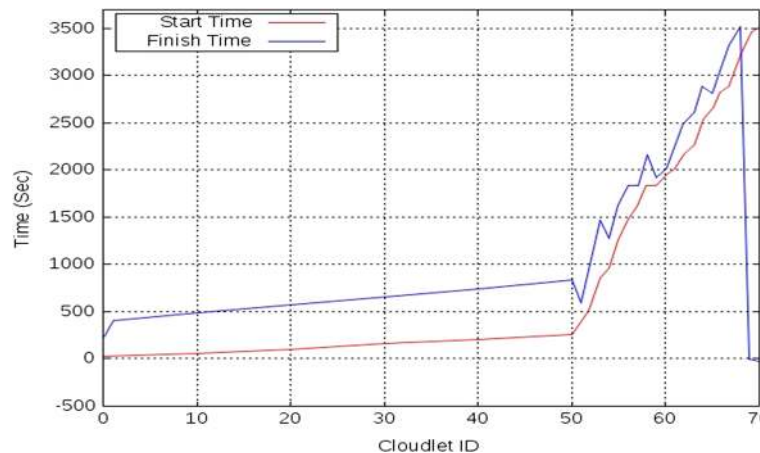

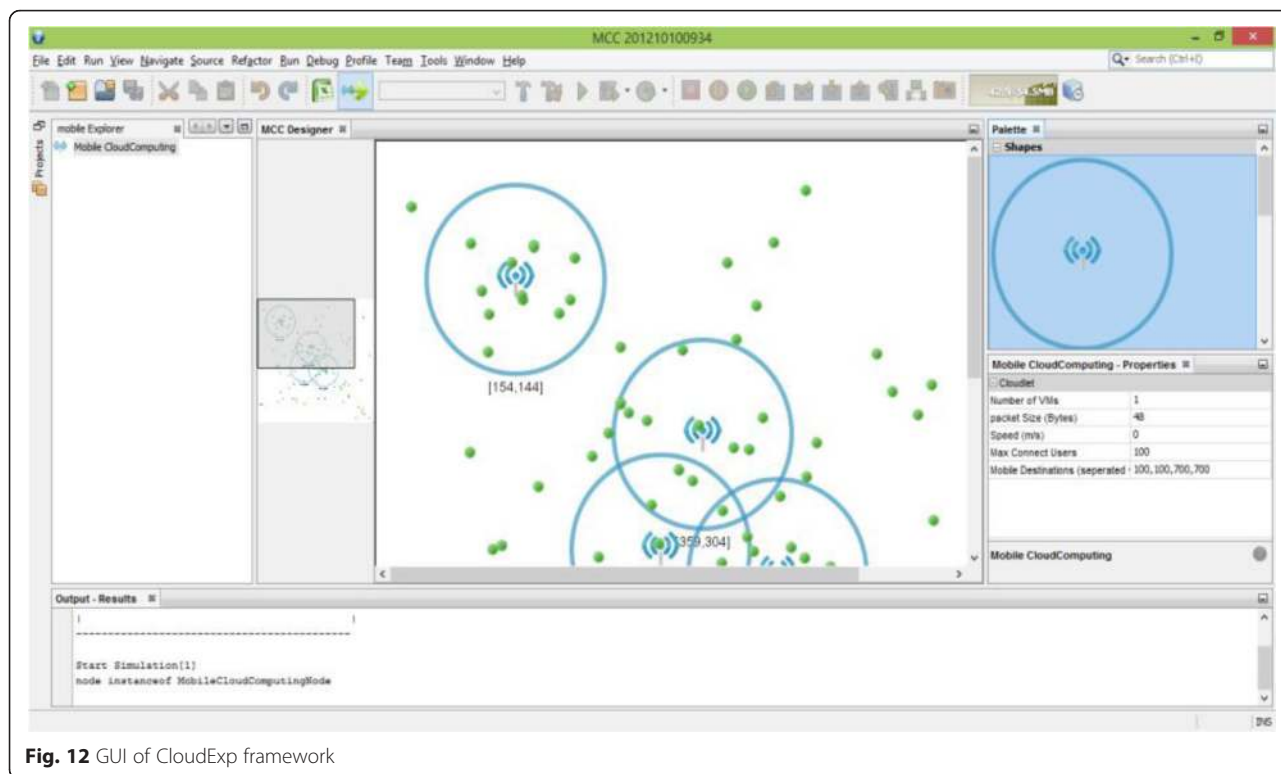**Fig. 11** Execution time for each customer's cloudlet

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 11 of 14

Fig. 12 GUI of CloudExp framework

superior when similar attributes are used, with better precision in experiments.

3.5 iCanCloud

iCanCloud [30] simulator is used to model and simulate a cloud computing system with a wide range of users. The user continuum ranges from basic users on one end to developers of large distributed applications on the other. iCanCloud can predict trade-offs between the cost and performance of a given set of applications that execute in specific hardware. This simulator is developed on top of [31], which is an open source communication network simulation package for OMNeT++ (modular, extensible, component-based C++ library and framework which are being used for building network simulators), as such, both are required to develop and execute new modules for iCanCloud.
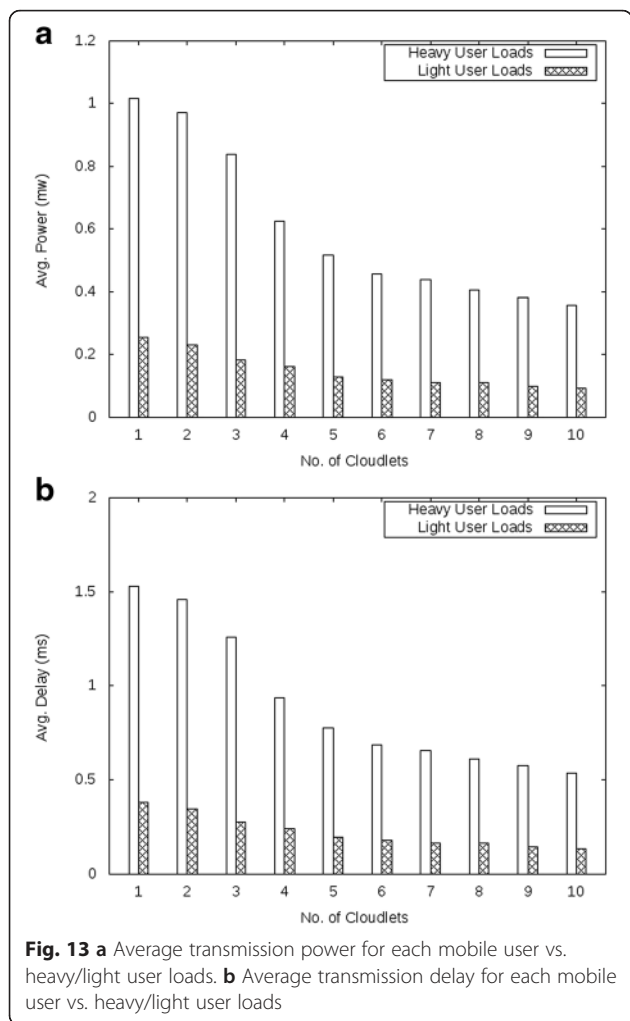
Table 3 Experiment properties

Property	Light load	Heavy load
Number of cloudlets	Changed from 1 to 10	Changed from 1 to 10
Area of workspace	800*800 m	800*800 m
Each experiment duration	3600 s (1 h)	3600 s (1 h)
Number of mobile users	500	500
Packet rate per second	0.1 (100 packets per ms)	0.4 (400 packets per ms)

Since cloud computing provides a variety of features, it is our opinion that a majority of users and consumers have differing interests; however, they share the common objective of optimizing trade-offs between cost and performance. This is a hard task that iCanCloud tries to alleviate.

iCanCloud offers flexible, scalable, fast, and easy-to-use environment for users to obtain results quickly, and enables users make swift decisions about the value of the services. Scalability is paramount as it determines the ability of a simulator to simulate large-scale systems without losing performance. This is particularly critical considering that the time required to run a simulation is directly proportional to the size of the infrastructure.

A flexible simulator allows its users to build environments easily by using different components with different levels of detail. Indeed, the hypervisor model allows users to integrate any cloud brokering policy to manage a set of fully customizable VMs, so an iCanCloud user can fully customize different brokering policies. By using iCanCloud, researchers in [32] simulate Multi-Agent System Cloud (MASCloud) for optimizing cost in cloud computing.

iCanCloud provides a set of components which emulate the behavior of real components in the real architecture, for instance, networks, disks, file systems, and memory. These components constitute the core simulation engine, while additional components can be added

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 12 of 14

Fig. 13 a Average transmission power for each mobile user vs. heavy/light user loads. **b** Average transmission delay for each mobile user vs. heavy/light user loads

and fully customizable global hypervisor to integrate a cloud brokering policy; (iii) it should have GUI for user convenience, to configure and launch simulations which can scale up to handle a single VM, to large systems that comprise of thousands of machines; (iv) it should reproduce the instance types which are provided by a given cloud infrastructure.

The following are the features of iCanCloud:

1. Existing and non-existing cloud architectures can be modeled and simulated on iCanCloud.
2. Uni-core/multi-core systems can be quickly simulated by using customizable VMs.
3. An easy method to integrate and test existent and new cloud brokering policies is provided by a flexible cloud hypervisor.
4. This simulator offers wide range of configurations to storage systems which include a model for remote storage system and local storage systems, for example NFS, parallel file systems, parallel storage systems, and RAID systems.
5. iCanCloud offers graphical user interface which enables easy generation and customization of large-scale distributed models. With the help of this graphical user interface, a user can easily manage a repository of pre-configured VMs, cloud system, pre-configured experiments, launch experiments from graphical user interface, and generate graphical reports.
6. To increase the functionality of simulator platforms, new components can be added to the repository of iCanCloud.
7. iCanCloud offer a POSIX-based [34] API and Message Passing Interface (MPI) library to model and simulate application.

3.6 GreenCloud
Data centres (DC) distributed over a large geographical area require a considerable amount of energy, which

to its repository, and the substitution of substitution of components.

Along similar lines to authors in [33], we postulate iCanCloud's design principles as (i) it should be able to conduct large experiments as compared to other simulators discussed in this paper; (ii) it should provide flexible

Table 4 Cloud computing simulator comparison

Parameter	CloudSim	CloudAnalyst	CloudReport	CloudExp	GreenCloud	iCanCloud
Platform	SimJava (GridSim)	Java SE 1.6, Swing, SimJava (CloudSim)	Ant, Maven, Java development kit (CloudSim)	CloudSim	NS2	OMNET, MPI
Availability	Open source	Open source	Open Source	No	Open source	Open source
Language/Script	Java	Java	Java	Java	C++/OTcl	C++
Graphical support	Limited by using CloudAnalyst and CloudReport	Full	Full	Full	Limited by using Nam	Full
Communication Models	Limited	Limited	Limited	Full	Full	Full
Power consumption modeling support	Limited	Limited	Limited	Yes	Yes	Work in progress
Run on windows environment	Yes	Yes	Yes	Yes	No	Yes

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 13 of 14

impacts the total operational cost for Cloud DC. The GreenCloud [35] simulator is designed to retrieve the details of the energy consumed by DC components, as well as packet level communication patterns between them. It presents the energy-aware studies of cloud computing DC in a realistic setup. GreenCloud is developed on top of NS2 (a packet level network simulator) to extract aggregates and avail information about the energy consumed by computing and communication components of DC. It is an open source tool, however, with a limited graphical user interface using Nam [36]. In addition, it cannot run on window environments.

A huge advantage of GreenCloud rests in its capabilities to measure DC power consumption, DC load, and DC job load balancing. Around 80 % code of this simulator is implemented in C++, while the rest 20 % is in the form of tool command language scripts.

GreenCloud simulator's source code [37] have the following major classes: (i) TskObject—it defines tasks representing a single workload which is executed in DC; (ii) TskComAgent and TskComSink classes are protocol agent classes that fragment a workload task into IP packets and perform their delivery; (iii) CloudUser is the parent class of all workload task generators and supports task generation; (iv) ExpCloudUser implements generation of workload task objects with exponentially distributed inter-arrival time; (v) CBRCloudUser class implements constant bit rate generation of workload objects; (vi) DcHost, in DC, this class is the main class coordinating execution and workload distribution; (vii) DcHost computing servers are represented by this class, and receive workload task through TxkComSink, scheduling for it to execute locally. It also constantly maintains a list of active tasks; (viii) SwitchEnergyModel monitors energy consumption of network switches according to load and energy saving mode enabled.

4 Comparison between major mobile cloud computing simulation tools

Table 4 summarizes the major cloud simulation models discussed in this research (CloudSim, CloudAnalyst, CloudReports, CloudExp, GreenCloud, and iCanCloud), for different parameters (language, platform, GUI, availability), communication modes, and the support level that the simulator provides.

5 Conclusions

The ever increasing popularity and complexity of computing systems makes simulation software an important choice for designing, configuring, managing, and analyzing the performance of a system. Cloud computing simulators have found use in evaluating trade-offs between cost and performance in pay-as-you-go (PAYG) environments. Our work therefore sought to present a survey of simulation tools suitable for cloud architectures. A variety of tools, including CloudSim, CloudAnalyst, and CloudReport, CloudExp, iCanCloud, and GreenCloud, are discussed. By comparing the different simulators, it is our opinion that iCanCloud is a superior platform as it provides a full GUI support to the user and runs on JVM. In addition, it is specifically designed for cloud simulation.

Author details
[1]Department of Computer Engineering, Umm al-Qura University, Makkah, Saudi Arabia. [2]Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan. [3]Cloud Computing and Applications Research Lab, Staffordshire University, Stoke-on-Trent, UK.

References
1. ZA Khan, J Siddiqui, SS Sohail, An extensive survey on mobile cloud computing. Open J. Comput. Sci. Eng. Survey. **3**(1), 52–61 (2015)
2. D. Dev, K.L. Baishnab. *A Review and Research Towards Mobile Cloud Computing. Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on. IEEE*, (2014)
3. Y Jararweh, Z Alshara, M Jarrah, TeachCloud: a cloud computing educational toolkit. Int. J. Cloud Comput. **2-3**, 237–257 (2013)
4. N Fernando, SW Loke, W Rahayu, Mobile cloud computing: a survey. Futur. Gener. Comput. Syst. **29**(1), 84–106 (2013)
5. J Shuja, A Gani, K Bilal, AUR Khan, SA Madani, SU Khan, AY Zomaya, A survey of mobile device virtualization: taxonomy and state of the art. ACM Comput. Surveys. **49**(1), 1 (2016)
6. E. Benkhelifa, et al. *User Profiling for Energy Optimisation in Mobile Cloud Computing." The 6th International Conference on Ambient Systems, Networks and Technologies, ANT*. (2015)
7. A. Maria. *Introduction to modeling and simulation. In: Proceedings of the 29th conference on Winter simulation. IEEE Computer Society*, (1997), p. 7-13
8. H. Casanova, A. Legrand, M. Quinson. *Simgrid: A generic framework for large-scale distributed experiments. In Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on*, (2008), p. 126-131. IEEE
9. R. Buyya, R. Ranjan, R.N. Calheiros. *Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. High Performance Computing & Simulation, 2009. HPCS'09. International Conference on. IEEE*, (2009)
10. A Núñez, JL Vázquez-Poletti, AC Caminero, GG Castañé, J Carretero, IM Llorente, iCanCloud: a flexible and scalable cloud infrastructure simulator. J. Grid. Comput. **10**(1), 185–209 (2012)
11. G Sakellari, G Loukas, A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing. Simul. Model Pract. Theory. **39**, 92–103 (2013)
12. Y Jararweh, M Jarrah, Z Alshara, MN Alsaleh, CloudExp: a comprehensive cloud computing experimental framework. Simul. Model Pract. Theory. **49**, 180–192 (2014)

Bahwaireth *et al. EURASIP Journal on Information Security* (2016) 2016:15

Page 14 of 14

13. M. Quwaider, Y. Jararweh, M. Al-Alyyoub, R. Duwairi. *Experimental Framework for Mobile Cloud Computing System* (1't International Workshop on Mobile Cloud Computing Systems, Management, and Security (MCSMS-2015))

14. B Wickremasinghe, Cloudanalyst: a Cloudsim-based tool for modelling and analysis of large scale cloud computing environments. MEDC project report. **22**(6), 433–659 (2009)

15. T.T.Sá, R.N. Calheiros, D.G. Gomes. *CloudReports: an Extensible Simulation Tool for Energy-Aware Cloud Computing Environments* (Cloud Computing. Springer International Publishing, 2014), p. 127-142

16. X Li, X Jiang, P Huang, K Ye, DartCSim: an enhanced user-friendly cloud simulation system based on CloudSim with better performance. Cloud Comp. Intel. Syst. 2012 IEEE 2nd Int. Conf. **1**, 392–396 (2012). IEEE

17. R. Buyya, GridSim (2008), http://www.buyya.com/gridsim/. Accessed 24 June 2016

18. R. Buyya, A. Beloglazov, J. Abawajy, *Energy-efficient management of data centre resources for Cloud Computing: a vision, architectural elements, and open challenges*. arXiv preprint arXiv:1006.0308 (2010)

19. G Geronimo, J Werner, R Weingartner, C Westphall, Provisioning, resource allocation, and DVFS in green clouds. Int. J. Adv. Netw. Serv. **7**, 108–117 (2014)

20. Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, E. Benkhelifa, *The Future of Mobile Cloud Computing: Integrating Cloudlets and Mobile Edge Computing*. In proceeding of the 23rd IEEE conference in Telecommunication (ICT2016), (2016)

21. J. Werner, G. A.Geronimo, C. B. Westphall, F. L. Koch, R. R. Freitas, C. M. Westphall, Environment, services and network management for green clouds. CLEI Electronic Journal. **15**(2), 1–13 (2012)

22. P. Humane, J. N. Varshapriya, *Simulation of Cloud infrastructure using CloudSim simulator: a Practical Approach for Researchers.* Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on. IEEE, (2015)

23. W. Long, L. Yuqing, X. Qingxin, *Using Cloudsim to Model and Simulate Cloud Computing Environment.* Computational Intelligence and Security (CIS), 2013 9th International Conference on. IEEE, (2013)

24. T Ahmed, Y Singh, Analytic study of load balancing techniques using tool Cloud Analyst. Int. J. Eng. Res. Appl. **2**(2), 4 (2012)

25. D Limbani, B Oza, A proposed service broker strategy in Cloudanalyst for cost-effective data centre selection. Int. J. Eng. Res. Appl. India. **2**(1), 793–797 (2012)

26. M Al-Ayyoub, Y Jararweh, M Daraghmeh, Q Althebyan, Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure. Clust. Comput. **18**(2), 919–932 (2015)

27. Q Althebyan, Y Jararweh, Q Yaseen, O AlQudah, M Al-Ayyoub, Evaluating map reduce tasks scheduling algorithms over cloud computing infrastructure. Concurrency Comput: Pract Experience. **27**(18), 5686–5699 (2015)

28. M Quwaider, Y Jararweh, A cloud supported model for efficient community health awareness. Perv. Mobile Comp. **28**, 35–50 (2016)

29. LA Tawalbeh, Y Jararweh, F Dosari, Large scale cloudlets deployment for efficient mobile cloud computing. J. Networks. **10**(01), 70–76 (2015)

30. A. N. Covarrubias, G. G. Castane, 'iCanCloud: Quick guide'. Version 0.9 (Beta) (2011), www.iCanCloudSim.org. Accessed 24 June 2016

31. INET Framework (2012, Welcome to the INET Framework. http://inet.omnetpp.org/. Accessed 24 June 2016

32. A Núñez, C Andrés, MG Merayo, 'MASCloud: a framework based on multi-agent systems for optimizing cost in cloud computing'. Computational Collective Intelligence. Technologies and Applications. Lect. Notes Comput. Sci. **7653**(2012), 436–445 (2012)

33. A Núñez, JL Vázquez-Poletti, iCanCloud: a flexible and scalable cloud infrastructure simulator'. J. Grid. Comput. **10**, 185–209 (2012). doi:10.1007/s10723-012-9208-5. Springer Science+Business Media B.V

34. Wikipedia (2010), POSIX, http://en.wikipedia.org/wiki/POSIX. Accessed 24 June 2016

35. D Kliazovich, P Bouvry, SU Khan, GreenCloud: a packet-level simulator of energy-aware. Cloud Comput. data centres. **62**, 1263–1283 (2010) doi:10.1007/s11227-010-0504-1

36. Nam (2002), The network animator, http://www.isi.edu/nsnam/nam/. Accessed 24 June 2016

37. GreenCloud (2012), The green cloud simulator, http://greenCloud.gforge.uni.lu/. Accessed 24 June 2016