Human Readable Scenario Specification for Automated Creation of Simulations on CloudSim

Manoel C. S. Filho
Computer Department
Instituto Federal de Educação do Tocantins (IFTO)
Palmas, Tocantins, Brazil 77021-090
Email: mcampos@ifto.edu.br
University of Beira Interior (UBI)
Covilhã, Castelo Branco, Portugal 6201-001
Email: d1365@ubi.pt

***Abstract*—Cloud Computing is been widely used as a computing model which enables customers to delivery services to their users, with reduced IT management and costs. However, the deployment of new applications to the cloud may require to project the environment that will be needed by it. CloudSim is a simulation tool ideal to project these cloud scenarios. However, each scenario creation requires programming. The creation of new scenarios is not direct and normally requires more programming efforts. This paper presents a tool to define cloud scenarios using YAML files and automate the creation of simulation at CloudSim. The proposed tool has been showed as a flexible and potential solution to create, share and extend CloudSim simulation scenarios.**

I. Introduction

Cloud Computing is a new paradigm to provide computing as utility. It is a consolidated way to provide computing services to customers in a pay-per-use basis. The customers account with virtually unlimited resources which can be used at any time, according with the demand of services hosted at cloud provider. The cloud brings the computing to a new level, joining computing power of heterogeneous physical hosts, at geographically distributed data centers. The resources at the cloud are managed autonomously to enable the multi-tenancy of the hosts, typically using virtualization [1]–[10].

The cloud enables companies, governments and universities to deliver services across the Internet, without high IT procurement costs. It tries to minimize resource underutilization, reducing costs, resource wastage and power consumption. Further, these benefits lead to reduce CO^2 emissions and contribute to Green Computing. It also enforces the entrepreneurship, enabling start-up companies to deliver new services to the cloud and be charged only by the used resources and accommodating the workload peaks [3], [5], [11]–[17].

However, the deployment of applications to the cloud may require tests prior they be make available to the users. These tests may define the feasibility of the project, but tests in a real cloud scenario may not be easy and represents a cost which may be avoided. Further, for research activities, the use of large real cloud scenarios can be impracticable due the associated costs and maybe requires to learn and interact with cloud specific APIs and/or tools to create the desired scenario. So, to newbie researchers in the cloud computing area, there is a learning curve associated to these tools and APIs which can be avoided using simulation tools such as CloudSim [18]. Simulation facilitate the cloud computing research process, avoiding costs and wasting time with concerns do not related to the desired research topic. For customers who want to deploy applications to the cloud, the resources required by the application can be measured, demand fluctuations can be tested and costs can be estimated in a easy way.

Nevertheless, the creation of simulation scenarios at CloudSim tool requires programming [18], [19]. So, the researcher will waste time with concerns not directly related to the scenario definition and does not have a split among the scenario specification and the code needed to create and run it. Every change in the scenario may require new changes on the code, so the researcher cannot focus only in the problem that he/she want to solve, such as implementing new algorithms to load balancing, new tasks and virtual machines (VM) schedule policies, VM placement, resource provisioning, workload prediction, server consolidation, energy efficiency, cost reduction and so on. Further, the definition of complex simulation scenarios may not be clear to others researches or to customers that will pay for a service been developed. This difficult the creation of new simulation scenarios, due the programming need, and does not give a complete overview of the scenario created.

The current paper presents a solution to solve these problems, completely avoiding the use of programming to define cloud simulation scenarios at CloudSim. The proposed tool allows the scenario specification using a YAML file that is a pretty human readable data format. So, the work focus only in simulation, despite there are other methodologies for lead works on cloud computing, such as statistical/mathematical, test beds in real commercial cloud environments (public clouds), test beds in local cloud computing environments (private clouds).

The main contributions of this paper are:

- to avoid programming to create cloud simulation environments on CloudSim tool;

Scholarship from CAPES

- to reduce the learning curve on creating CloudSim simulation scenarios;
- to facilitate and automate the process of cloud simulation environments creation;
- to use a human readable file format to specify cloud simulation scenarios and speed up this phase;
- to allow reuse, extension and share of simulations scenarios.

The paper is organized as follows. At Section II the related works are reviewed. At Section III the available tools used in this work are presented. At Section IV the proposal is presented. At Section V the work results are discussed. And finally, at Section VI are presented the conclusions and future works.

II. Related Work

There are a restrict set of original and independent cloud computing simulation tools. At this section some of these most recent tools are presented.

At [20] the CloudSched simulation tool is presented. It is a simulation tool aiming the evaluation of resource scheduling for cloud computing applications. The simulator can be used to test different scheduling policies and resource allocation schemes. The tool focus only in scheduling VMs on IaaS layer. It has a graphical user interface to create the simulation and allows the specification of simulation scenarios using files, but in non-standard text file format. The tools is developed using Java and is freely available on the web[1] but is not open source. So, this forbid the extension by creation and testing of new scheduling policies by other developers.

At [21], a survey on cloud-based simulation is presented. The goal of this type of simulation is to use simulations tools running into real cloud environment instead of the developer local machine. This type of simulation tools provides new cloud computing service models such as modeling as a service (MaaS), execution as a service (EaaS), analysis as a service (AaaS) and simulation resources as a service (SRaaS). These new service models enable developers to use simulation tools as a service, like software as a service (SaaS) model, but called simulation as a service (SIMaaS). The work presents an architecture to cloud-based simulation and reports about a prototype which, apparently, is not available in the web.

At [22], the design and development of a cloud-based simulation tool is presented, following the simulation service models presented at [21]. However, there is not mentioned a concrete tool or service freely available.

At [23], the Stratus cloud computing simulation framework is presented, inside the SciCloud[2] project [24]. It aims to provide simulations of data processing on the cloud, like the established Hadoop MapReduce framework, but using the bulk synchronous parallel (BSP) computing model.

Finally, at [25] is presented a review of some cloud simulation tools. The tools are summarized and their architectures presented. Last, they are compared showing the development platform and languages used.

III. Simulation Tools Used

For the development of this work, the CloudSim and CloudReports tools were used. They are discussed at the next subsections and represent the basis where the proposed tool was built and runs.

A. CloudSim Simulation Tool

CloudSim is a open source tool for modelling and simulation of cloud environments. It was developed on the top of GridSim simulation toolkit, that uses the SimJava discrete event simulation engine [18], as presented at Figure 2. It simplifies the process of asses large cloud environments and the test of algorithms for cloud, such as tasks and VM scheduling, VM placement and migration, load balancing and so on.

It is a lightweight tool that can make simulations with thousands VMs without requiring a powerful machine. At [18], was presented tests with 100000 VMs using a computer with a Celeron 1.86 GHz processor and 1 GB of RAM, that prove how efficient is the tool.

The tool can be used to reduce time and cost of tests at cloud scenarios and enables researches in the area without the need of real cloud infrastructure.

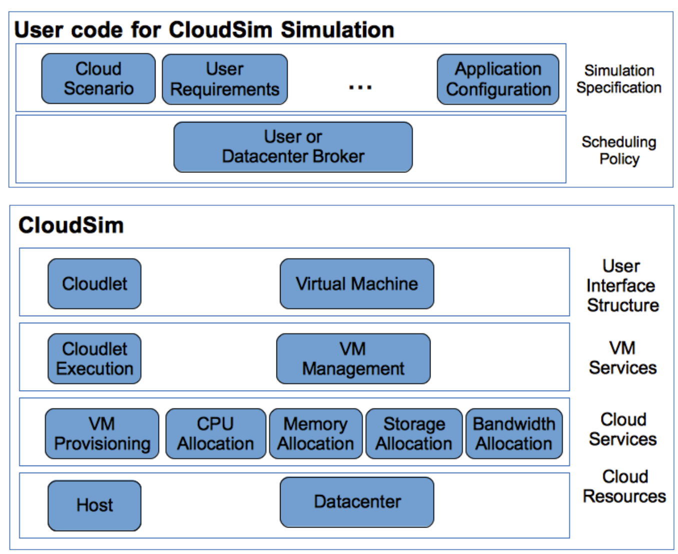

Fig. 1: CloudSim architecture (Adapted from [18])

However, the creation of cloud environments, and each object on it, is made programatically. The tool does not provide a simpler manner to specify the environment and run the simulation. Each simulation scenario need to be created by a developer using Java language. Some Java scenarios are provided with the tool, serving of start point to create new and more complex ones.

[1]http://sourceforge.net/projects/cloudsched/
[2]http://ds.cs.ut.ee/research/scicloud

B. CloudReports Graphical Tool

CloudReports [26] is a front-end to CloudSim [18]. It facilitates the process of creating cloud simulation scenarios using a graphical user interface and avoiding the need of programming them. It runs on top of CloudSim, creating and executing the simulation scenarios on it. It stores the scenarios created and generate several reports which allows the assessment of the simulation.

It is a ready and easy to use tool which can be used instead of directly programming the cloud scenarios at CloudSim. So, it facilitates very much the process of specify the cloud environment and run the simulation. Nevertheless, the tool does not allow the automation of the process of creating the whole scenario. The scenario is created graphically using the application interface. For larger environments, this can be boring and time-consumption. The tool does not allow sharing of specifications among different teams. The extension of one scenario using the specification of another one is not possible too.

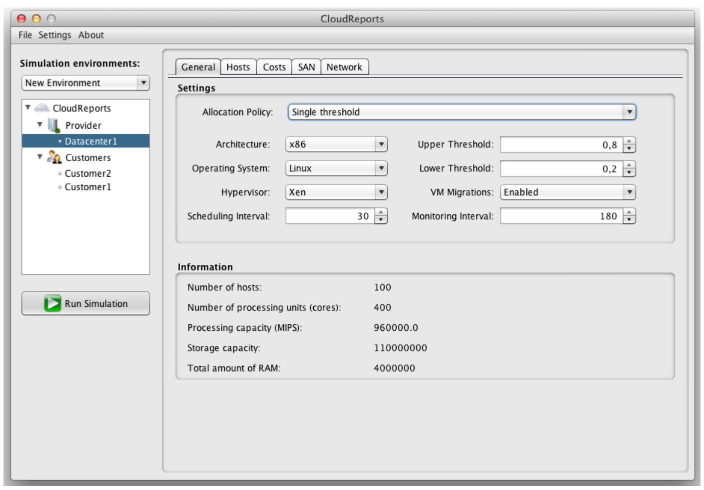

Fig. 2: CloudReports screenshot

C. Motivation to Choose the Tools

The CloudSim was chosen due it be an open source simulation tool and maybe the most well know for cloud simulation. It is a well designed and documented Java tool with a lot of others simulation tools based on it. There is a set of extensions which can be attached to it. Its modular project allows the developer to create new algorithms to VM and tasks scheduling policies, VM placement, resource provisioning and so on. The tool provides a set of examples which facilitate the process of understanding and creating new cloud simulation environments using the Java. Last, the tool has a support forum to help its users.

The CloudReports was chosen as a natural option after the choice of CloudSim, due it provides a set of classes used to abstractly defining CloudSim's objects.

IV. Proposal to automate the creation of CloudSim simulation environments

In the proposed work, cloud simulation scenarios are defined into a YAML file. YAML is the simpler human readable data serialization standard which was found in the literature and it make easy to define a cloud simulation scenario directly, using any YAML editor or even a simple text editor. It is a non-bureaucratic data format that, different from XML pattern, does not need to close tags (keys). This simplifies and speeds up manual creation of YAML files.

The YAML keys were defined based on the CloudReports' registries classes, as presented in Figure 3. These classes represent an abstract definition of CloudSim objects to the creation of simulation scenario. By means of these classes, objects like datacenters, storage area networks (SANs), hosts, virtual machines (VMs), customers and applications (cloudlets) can be defined at a higher level. Using they, the amount of determined objects with the same configuration, like hosts, can be specified only one time. So, this speed up the process of creating the cloud environment's objects. Thus, the creation of, for instance 1, 2, 10 or 100000 hosts with the same configuration has the same effort.

At Figure 3 there are the top level registries objects datacenterRegistries and customerRegistries, that represent, respectively, the abstract information of datacenters and customers in the cloud simulation scenario. These registries are a list of registry objects. The amount key, presents in almost all objects which can be specified in the YAML file, defines the amount of the current object to be created during the scenario creation. The scenario is created by the proposed tool and further ran in CloudSim. So, for instance, each datacenter in the datacenterRegistries list, represents a specific datacenter configuration. For each one in this list, will be created the concrete objects in the amount specified by the amount key.

The YAML snippet, presented at Listing 1, shows the creation of 1 datacenter with one specific configuration and the creation of 2 more datacenters that share another configuration. The first datacenter has different costs of the other two ones and has migration of VMs disabled.

Listing 1: Cloud simulation environment YAML file sample: creation of datacenters with different configurations

```
datacenterRegistries:
  - !datacenter
    amount: 1
    allocationPolicyAlias: Simple
    vmMigration: false
    costPerSec: 0.2
    costPerMem: 0.04
    costPerStorage: 0.004
    costPerBw: 0.25
  - !datacenter
    amount: 2
    allocationPolicyAlias: Simple
    vmMigration: true
    costPerSec: 0.1
    costPerMem: 0.05
    costPerStorage: 0.002
    costPerBw: 0.3
```

How can be seen in the Listing 2, inside each registry key, another cloud objects can be specified. Inside each datacenter defined, a list of storage area networks (SANs) and physical hosts can be specified. Inside each customer defined, a list of virtual machines and cloudlets can be specified.

A. Architecture

The tool was developed in Java 7 and runs on top of CloudSim, using some CloudReports' classes. It is a command line application, like CloudSim, that loads a specified YAML file containing the definitions of simulation scenario(s). It uses the YamlBeans[3], a very simple and complete YAML parser for Java that can serialize plain old java objects (POJOs) to YAML and deserialize YAML content to POJOs.

The proposed tool parses the YAML file containing the simulation scenario specification, storing the definitions into CloudReports registries objects. Theses registries is used to autonomously creating the concrete objects that will constitute the cloud environment (datacenters, hosts, VMs, etc). Further, with all concrete objects created at the desired amount, specified at the YAML file, the tool runs the simulation in CloudSim, organizing the results into appropriated tables.

At Figure 3, the process is shown. The automation module will parse the YAML file, creating CloudReports' registries objects, and generates the the Java code needed to create the cloud environment and run the simulation on CloudSim.

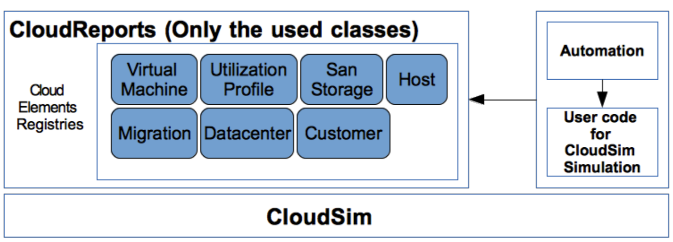

Fig. 3: Architecture to creation of cloud computing scenarios (Adapted from [18])

B. Demonstration Scenario

To demonstrate to use of the proposed tool, the Listing 2 presents a simulation scenario defined into a YAML file. The scenario has 1 datacenter with 4 physical hosts with the same configuration. Each host has the following configuration: 1 TB of RAM (1000000 MB), 100 Mbps of bandwidth (100000 bps), 40 TB of storage (40000 GB), and a quad-core processor (numOfPes - number of processing elements) with each core having a processing capacity of 50000 million instructions per second (MIPS). The virtual machines placed on each host will be executed using a time shared scheduler, allowing the VMs' execution to be scaled along the time. Each host will use a simple policy to provisioning RAM (ramProvisionerAlias), bandwidth (bwProvisionerAlias) and processor's cores (peProvisionerAlias) to the virtual machines allocated on it. The simple provisioning uses the best effort algorithm to allocate the requested resource. So, if there is enough resources, they will be provisioned, otherwise, the provisioning will fail. These simple provisioners are the only available with CloudSim. Whether the developer wants another provisioners policies, he/she needs to implement them, extending the CloudSim classes.

[3]http://github.com/EsotericSoftware/yamlbeans

At the scenario were defined 2 customers, that in CloudSim are represented by a broker. A broker is a component on the cloud that works on behalf of the service's users. So, when the users make a request to the service, the broker is accountable to negotiate with the cloud coordinator the provision of the resources needed to meet the application' QoS [18]. Each customer has 2 VMs with the following configuration: 500 MB of image size, a dual-core processor (numOfPes) with each core having a processing capacity of 1000 million instructions per second, 2 GB of RAM (2000 MB) and 1 Mbps of bandwidth (1000 bps). The tasks will be executed using a space shared scheduler. This means that when a application starts, it will use the provisioned processor cores until the end of its execution. This can cause a longer wait time to applications that depends of the busy cores.

For each customer created, were specified 6 cloudlets to be executed. The cloudlets are objects of CloudSim that represent the customer's applications in the simulation scenario. They are abstractions of the real cloud applications. So, one cloudlet only represents the computing requirements of a real application [18].

Each cloudlet was defined with the following requirements: 2 processor's cores (cloudletsPesNumber) needed, 100 million of instructions (MI) to be executed, with 50 bytes of input file size (application code + input data) and 70 bytes of output file size (input file size + output data). The cloudlets were defined to use the whole required resources (RAM, bandwidth and CPU) during 100% of its execution time. This is defined by the utilizationModel*XXX* keys.

Listing 2: Sample of cloud simulation scenario defined into a human readable YAML file

```
datacenterRegistries:
  - !datacenter
    amount: 1
    allocationPolicyAlias: Simple
    vmMigration: enabled
    costPerSec: 0.1
    costPerMem: 0.05
    costPerStorage: 0.001
    costPerBw: 0.1
    sanList:
      - !san
        capacity: 10000
        bandwidth: 10000
        networkLatency: 5
    hostList:
      - !host
        amount: 4
        ram: 1000000
        bw: 100000
        storage: 40000
        numOfPes: 4
        mipsPerPe: 50000
        schedulingPolicyAlias: TimeShared
        ramProvisionerAlias: Simple
        bwProvisionerAlias: Simple
        peProvisionerAlias: Simple
customerRegistries:
  - !customer
    amount: 2
    vmList:
      - !vm
        amount: 10
        size: 500
        pesNumber: 4
        mips: 1000
```

```
        ram: 2000
        bw: 1000
        schedulingPolicyAlias: SpaceShared
        vmm: Xen
    utilizationProfile:
      - !profile
        numOfCloudlets: 6
        cloudletsPesNumber: 2
        length: 100
        fileSize: 50
        outputSize: 70
        utilizationModelCpuAlias: Full
        utilizationModelRamAlias: Full
        utilizationModelBwAlias: Full
```

At the next section, the simulation's results will be demonstrated.

V. Result Analysis

How can be seen in the scenario presented at Listing 2, there are 2 dual-core VMs to each one of the 2 customers. However, for each customer there are 6 cloudlets (applications) requiring 2 processors' cores each one. Therefore, there are not enough VMs for each customer to run him/her applications at the same time. So, how was defined a space shared scheduler to the VM's tasks and each cloudlet will require all the 2 cores of one customer's VMs, only one cloudlet will be executed per time. By this way, the others cloudlets will wait in a queue. This can be seen at the Figure 4.

```
Broker: name broker1 id 3 cloudlets executed: 6============================================================
###|    CloudletID|     STATUS  |       DataCenterID|   VmID|    HostID| ExecTime|       Start Time|      Finish Time|
int|    int       |     string  |       int         |   int |    int   | secs    |       secs      |      secs       |
  1|    1         |     SUCCESS |       2           |   1   |          | 0,11    |       0,1       |      0,21       |
  2|    2         |     SUCCESS |       2           |   2   |          | 0,11    |       0,1       |      0,21       |
  3|    3         |     SUCCESS |       2           |   1   |          | 0,11    |       0,21      |      0,32       |
  4|    4         |     SUCCESS |       2           |   2   |          | 0,11    |       0,21      |      0,32       |
  5|    5         |     SUCCESS |       2           |   1   |          | 0,11    |       0,32      |      0,43       |
  6|    6         |     SUCCESS |       2           |   2   |          | 0,11    |       0,32      |      0,43       |

Broker: name broker2 id 4 cloudlets executed: 6============================================================
###|    CloudletID|     STATUS  |       DataCenterID|   VmID|    HostID| ExecTime|       Start Time|      Finish Time|
int|    int       |     string  |       int         |   int |    int   | secs    |       secs      |      secs       |
  1|    7         |     SUCCESS |       2           |   3   |          | 0,11    |       0,1       |      0,21       |
  2|    8         |     SUCCESS |       2           |   4   |          | 0,11    |       0,1       |      0,21       |
  3|    9         |     SUCCESS |       2           |   3   |          | 0,11    |       0,21      |      0,32       |
  4|    10        |     SUCCESS |       2           |   4   |          | 0,11    |       0,21      |      0,32       |
  5|    11        |     SUCCESS |       2           |   3   |          | 0,11    |       0,32      |      0,43       |
  6|    12        |     SUCCESS |       2           |   4   |          | 0,11    |       0,32      |      0,43       |

CloudEnvironment Simulation finished!
```

Fig. 4: Execution's results of cloud simulation scenario presented at Listing 2

At the refereed figure, can be seen 2 brokers representing the customers defined in the simulation scenario. For each customer, were executed 6 cloudlets. The two VMs of each customer were instantiated. The VMs number 1 and 2 were assigned to the customer 1 (represented by broker 1) and the VMs number 3 and 4 were assigned to the customer 2 (represented by broker 2). At the column "ExecTime" can be seen that all cloudlets took the same time to finish, because all cloudlets and VMs have the same configurations. However, the finish time was variable. According to the VMs and cloudlets configurations, only one cloudlet can be executed at a VM per time. So, how each customer has two VMs, two customer's cloudlets can be executed per time. By this way, the first two cloudlets of each customer have the same finish time. The next two have another finish time and the next ones also.

If the cores of the VMs be changed to 4, will be possible to run two cloudlets at the same VM so, the wait time will be reduced. This can be seen at the Figure 5. The cloudlets 1, 2, 3 and 4 had the same finish time due they were started at the same time. Cloudlets 1 and 3 were executed at VM 1 and cloudlets 2 and 4 were executed at VM 2. The cloudlets 5 and 6 had to wait until the end of the other ones, so their start time is exactly the finish time of the previous cloudlets. In this case, the broker used chose to allocate one customer's VM to each cloudlet, even if the two remaining cloudlets would be executed at the same VM. To change this behaviour is needed to extend the DatacenterBroker CloudSim's class. Only the first customer results were shown because they are identical, due the fact of him/her VMs and cloudlets configurations be the same.

```
Broker: name broker1 id 3 cloudlets executed: 6============================================================
###|    CloudletID|     STATUS  |       DataCenterID|   VmID|    HostID| ExecTime|       Start Time|      Finish Time|
int|    int       |     string  |       int         |   int |    int   | secs    |       secs      |      secs       |
  1|    1         |     SUCCESS |       2           |   1   |          | 0,11    |       0,1       |      0,21       |
  2|    3         |     SUCCESS |       2           |   1   |          | 0,11    |       0,1       |      0,21       |
  3|    2         |     SUCCESS |       2           |   2   |          | 0,11    |       0,1       |      0,21       |
  4|    4         |     SUCCESS |       2           |   2   |          | 0,11    |       0,1       |      0,21       |
  5|    5         |     SUCCESS |       2           |   1   |          | 0,11    |       0,21      |      0,32       |
  6|    6         |     SUCCESS |       2           |   2   |          | 0,11    |       0,21      |      0,32       |
```

Fig. 5: Execution's results of cloud simulation scenario with quad-core VMs

The presented results was formatted to make easy the understanding and can be easily changed at the proposed tool.

VI. Conclusion and Future Works

At the paper, a flexible, open source and simple solution was proposed to easy the process of creating cloud simulation scenarios at CloudSim tool. The tool has been shown easy to use and the specification of the simulation scenario using the very human readable YAML data format simplify and speed up this task. The scenarios can be easily changed in the YAML file to test different cloud architectures characteristics.

The YAML defined scenario easy the process of overview, understanding, extend and share the scenarios created. They can be created using any YAML editor or even a simple text file editor. So, the tool can reduce the learning curve in understanding cloud technologies and simulation. Further, it may motivate researches in the cloud computing area due the easiness of creating any desired cloud simulation scenarios.

As future works are proposed implementation of different resource allocation, VM scheduling and placement policies; creation of random workload; create policies to define elasticity for VMs resources and VMs replication to enable load balancing and integration with CloudReports' graphical user interface.

Acknowledgement

The authors would like to thank CAPES Brazilian agency for the scholarship to support this work.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, Jun. 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167739X08001957

[2] S. Patidar, D. Rane, and P. Jain, "A Survey Paper on Cloud Computing," *2012 Second International Conference on Advanced Computing and Communication Technologies*, pp. 394–398, Jan. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6168399

[3] A. J. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for Cloud computing environments," *International Conference on Green Computing*, pp. 357–364, Aug. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5598294

[4] B. Armbrust, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A view of cloud computing," *Communications of the ACM*, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1721672

[5] J. Espadas, A. Molina, G. Jiménez, M. Molina, R. Ramírez, and D. Concha, "A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 273–286, Jan. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167739X1100210X

[6] A. Goyal and S. Dadizadeh, "A survey on cloud computing," *University of British Columbia, Technical Report for CS 508*, no. December, pp. 1–14, 2009. [Online]. Available: http://unbreakablecloud.com/wordpress/wp-content/uploads/2011/02/A-Survey-On-Cloud-Computing.pdf

[7] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, Apr. 2010. [Online]. Available: http://www.springerlink.com/index/10.1007/s13174-010-0007-6

[8] Z. Zhang, C. Wu, and D. Cheung, "A survey on cloud interoperability: taxonomies, standards, and practice," *ACM SIGMETRICS Performance . . .*, no. 4, pp. 13–22, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2479945

[9] T. Dillon, C. Wu, and E. Chang, "Cloud Computing: Issues and Challenges," *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 27–33, 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5474674

[10] S. S. Manvi and G. Krishna Shyam, "Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey," *Journal of Network and Computer Applications*, pp. 1–17, Oct. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1084804513002099

[11] B. Sotomayor, S. Montero, I. M. Llorente, and I. Foster, "An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Clouds," *Ieee Internet Computing, Special Issue on Cloud Computing*, pp. 1–11, 2009.

[12] M. Maurer, I. Brandic, and R. Sakellariou, "Adaptive resource configuration for Cloud infrastructure management," *Future Generation Computer Systems*, vol. 29, no. 2, pp. 472–487, Feb. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167739X12001525

[13] E. A. Fischer and P. M. Figliola, "Overview and Issues for Implementation of the Federal Cloud Computing Initiative : Implications for Federal Information Technology Reform Management," 2013.

[14] D. Warneke and O. Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 985–997, Jun. 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5710902

[15] S. Vijayakumar, Q. Zhu, and G. Agrawal, "Dynamic Resource Provisioning for Data Streaming Applications in a Cloud Environment," *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pp. 441–448, Nov. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5708483

[16] R. N. Calheiros, M. A. S. Netto, C. A. F. D. Rose, and R. Buyya, "EMUSIM : an integrated emulation and simulation environment for modeling , evaluation , and validation of performance of Cloud computing applications," *Software - Practice and Experience*, no. April 2012, pp. 595–612, 2013.

[17] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing — The business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, Apr. 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167923610002393

[18] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," *2009 International Conference on High Performance Computing and Simulation*, pp. 1–11, Jun. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5192685

[19] R. N. Calheiros, R. Ranjan, A. Beloglazov, and A. F. D. Rose, "CloudSim : a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software - Practice and Experience*, no. August 2010, pp. 23–50, 2011.

[20] W. Tian, Y. Zhao, M. Xu, Y. Zhong, and X. Sun, "A Toolkit for Modeling and Simulation of Real-Time Virtual Machine Allocation in a Cloud Data Center," *IEEE Transactions on Automation Science and Engineering*, pp. 1–9, 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6558510

[21] X. Liu, X. Qiu, B. Chen, and K. Huang, "Cloud-Based Simulation: The State-of-the-Art Computer Simulation Paradigm," *2012 ACM/IEEE/SCS 26th Workshop on Principles of Advanced and Distributed Simulation*, pp. 71–74, Jul. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6305887

[22] Q. Pan, J. Pan, and C. Wang, "Simulation in Cloud Computing Envrionment," *2013 International Conference on Service Sciences (ICSS)*, pp. 107–112, Apr. 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6519772

[23] P. Jakovits, S. N. Srirama, and I. Kromonov, "Stratus: A Distributed Computing Framework for Scientific Simulations on the Cloud," *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, pp. 1053–1059, Jun. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6332290

[24] S. Srirama, O. Batrashev, and E. Vainikko, "SciCloud: Scientific Computing on the Cloud," *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 579–580, 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5493427

[25] W. Zhao, Y. Peng, F. Xie, and Z. Dai, "Modeling and simulation of cloud computing: A review," *2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC)*, pp. 20–24, Nov. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6486505

[26] T. Teixeira Sá, "CloudReports: uma ferramenta gráfica para a simulação de ambientes computacionais em nuvem baseada no framework CloudSim," *Simpósio Brasileiro de Sistemas Distribuídos e Redes de Computadores (SBRC)*, pp. 103–116, 2011. [Online]. Available: http://sbrc2011.facom.ufms.br/files/workshops/wcga/ST03_2.pdf