# Cloud Simulator, state of the art and requirements

Versione 0.1
Data 18/09/2014

Progetto iCaro
La piattaforma cloud per l'accelerazione del business delle PMI toscane
[CUP 6408.30122011.026000074]

# Informazioni sul documento

| | |
|---|---|
| ID Deliverable | 3.26.3 |
| Titolo Deliverable | Cloud Simulator, state of the art and requirements |
| ID Attività | 3.5 |
| N. Versione / Revisione | 0.1 |
| Natura: Bozza / Definitivo | Definitivo |
| Partner responsabile | UNIFI DISIT |
| Distribuzione: Riservato / Pubblico | Pubblico |
| Riferimenti Autore | DISIT lab |
| Data redazione | 18-09-2014 |
| Riferimenti revisore | Paolo Nesi |
| Data revisione | 18-09-2014 |
| Riferimenti soggetto che approva | Paolo Nesi |
| Data approvazione e consegna | 18-09-2014 |

# Controllo delle revisioni

| Oggetto | Numero | Data |
|---|---|---|
| Final version | 0.1 | 18-09-2014 |
| | | |
| | | |

# Nota di riservatezza

Pubblico

# Index



# Figure Index



# Table Index



# Legenda Acronimi e sigle

| Acronimo / Sigla | Dettaglio |
|---|---|
| SM | Supervisor & Monitor |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

# 1. State of the art

In recent years there has been an exponential growth of cloud platforms capable of offering every type of service (XaaS). These platforms are of priority importance for those in need of services, but do not have the economic requirements to put together a system on their own: the cloud lets you turn a fixed cost (CAPEX) in an operational expenditure (OPEX).
Companies offering services XaaS must be able to optimize the use of resources available to prevent damage and to minimize energy consumption, but this optimization cannot be tested on the resources that provide services, as it would risk impairment of the data or customer privacy.

For this reason, companies and researchers have developed, in the last years, many cloud simulation tools to test every new progress mentioned above (optimize the use of resources), avoiding that tests impair production machines.

The first step in developing a new simulator is to decide which of the below challenges the developer wants to engage. In fact, most of the simulators already developed are focused on one challenge only. In (1) (2) are described the major challenges that prevent the cloud as being a standard platform as:

- **Security:** since all the storage and computations are processed in cloud servers, the importance of confidentiality, data integrity and non-repudiation issues are predominant.
- **Cost Modeling:** cloud computing has a unique pay-as-you-go service model; through which organizations pay only for what is being used and nothing more. For example, it might be highly beneficial for a company if a brand new high powered server farm could be obtained to introduce a new web based market offering with zero upfront capital.
- **Energy Management:** due to fluctuation of workload, the average load is only 30% of data center resources and rest of the 70% account putting resources in sleep mode, so the main goal is to run an application with a minimum set of computing resources and maximize the resources that are in sleep mode (3) (4)
- **Virtual Machine Migration:** since cloud computing is a distributed system, when the workload is increased in a particular data center, virtual machine migration helps to prevent performance degradation of the system. (5)

Most part of cloud simulation tools are software because until a few years ago, it was very expensive to create a simulated data center buying the hardware infrastructure on which to run tests. Also, creating a practical test-bed, consisting of a reasonable number of servers (say, 40 machines) can still be out of reach for most researchers when one needs to consider space, power and cooling infrastructure (6). Nowadays some researchers have taken the effort to create a scaled data center using the Raspberry Pi device reducing drastically the cost of this solution.

The rest of the document is organized as follows. In the next section we will analyze the software cloud simulator providing a brief description of each tool, main features and known cons. At the end of this section there is a table in which the tools described above are compared.

## 1.1 Software Cloud Simulator

**CloudSim** (7)**:** A new, generalized, and extensible simulation framework that allows seamless modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services. By using CloudSim, researchers and industry-based developers can test the performance of a newly developed application service in a controlled and easy to set-up environment (7).

Features:

- Modeling and creating a huge data center, unlimited number of virtual machines, introducing brokering policy and support the important feature of cloud computing pay-as-you-go model (5). The time to instantiate an experiment setup with 1 million hosts is around 12 s (7).
- It implements generic application provisioning techniques that can be extended with ease and limited efforts (8).

Cons:

- Lack of GUI
- Does not run in real time. So, you would not be able to evaluate how long does it take for your algorithm to make a decision it could take.
- For complex simulation you must learn the architecture of CloudSim and you must write in Java your simulation. Furthermore, because Java can only handle at most 2 GB of memory in 32 bits systems. This fact heavily affects the design of experiments (9).
- CloudSim is not a framework as it does not provide a ready to use environment for execution of a complete scenario with a specific input (8).
- Users of CloudSim have to develop the Cloud scenario it wishes to evaluate, define the required output, and provide the input parameters (8).
- Basically built for single server architecture and become insufficient for real cloud model, deploying different type of applications from different customer (5).
- CloudSim implements an HPC-style workload, with Cloudlets (jobs) submitted by users to VMs for processing. It can be used to simulate a transactional, continuous workload such as a web server or other service, but it lacks a detailed model of such an application (10).

**CloudAnalyst** (11)**:** Supports visual modeling and simulation of large-scale applications that are deployed on Cloud Infrastructures. CloudAnalyst, built on top of CloudSim, allows description of application workloads, including information of geographic location of users generating traffic and location of data centers, number of users and data centers, and number of resources in each data center (11).

Features:

- Provides modelers with a high degree of control over the experiment, by modeling entities and configuration options (5).
- Allows modelers to save simulation experiments input parameters and results in

the form of XML files so the experiments can be repeated (5).

Cons:

- CloudAnalyst is favorable for testing the performance of social networking sites such as Facebook, Twitter etc. (2)
- Need to restart of CloudAnalyst for every simulation.
- OS and architecture are transparent to simulation. Only x86 and Linux can be selected.
- Workload based on users' requests, a preliminary study is needed for the number and size of users' requests (Inheritance of CloudSim).
- Allows the configuration of high level parameters only (9).

**GreenCloud** (4)**:** For advanced energy-aware studies of cloud computing data centers in realistic setups. Extracts, aggregates, and makes information about the energy consumed by computing and communication elements of the data center available in an unprecedented fashion. In particular, a special focus is devoted to accurately capture communication patterns of currently deployed and future data center architectures (4).

Features:

- Developed as an extension of a packet-level network simulator Ns2 (12).
- Implements a full TCP/IP protocol reference model which allows integration of different communication protocols with the simulation (8).

Cons:

- The simulation duration is greatly influenced by the number of communication packets produced as well as the number of times they are processed at network routers during forwarding. As a result, a typical data center simulated in GreenCloud can be composed of thousands of nodes while the Java-based CloudSim and MDCSim can simulate millions of computers (4).
- User of this simulator needs to learn both of the programming languages i.e. C++ and Otcal to use this simulator, which is a noticeable drawback (5) two different languages must be used to implement one single experiment (9).
- Basically built for single server architecture and become insufficient for real cloud model, deploying different type of applications from different customer (5).
- Although it can support a relatively large number of servers, each server is considered to have a single core and there is no consideration of virtualization, storage area networks and resource management (13).
- Although it has a detailed workload model, it does not include any modeling of virtualization. As such, it is not suitable for virtualized resource management research (10).

**iCanCloud** (9)**:** The main objective of iCanCloud is to predict the trade-offs between cost and performance of a given set of applications executed in a specific hardware, and then provide to users useful information about such costs (9).

Features:

- Provides in-depth simulation of physical layer entities such as cache, allocation policies for memory and file system models (9).
- It has been designed to perform parallel simulations, so one experiment can be executed spanning several machines. In (9) this feature is not available yet.
- Several methods for modeling applications can be used in iCanCloud: using traces of real applications (9).

Cons:

- It does not provide models for power consumption, although this is included as future work (9).
- This tool is lacking the essential capability to simulate the variety of heterogeneous networks involved in the end-to-end cloud service supply chain (14).
- Aimed at simulating instance types provided by Amazon without considering the underlying network behavior (6).

**NetworkCloudSim** (15)**:** The main challenge addressed is to develop application and network models that are sophisticated enough to capture the relevant characteristics of real Cloud data centers, but simple enough to be amenable for analysis (13).

Features:

- It is equipped with more realistic application models than any other available Cloud simulator (15).
- Network flow model for Cloud data centers utilizing bandwidth sharing and latencies to enable scalable and fast simulations (15).
- For helping users to model such communicating tasks, we designed the *NetworkCloudlet* class that represents a task executing in several phases/stages of communication and computation (15).
- It uses Network Topology class which implements network layer in CloudSim, reads a BRITE file and generates a topological network (8).

Cons:

- Although, users can model complex applications in their simulation environment, still the precise execution of such applications depend highly on the underlying network model (15).

**EMUSIM** (16)**:** It is not only simulator; it provides both simulation and emulation of a cloud application. It is developed for software as a service (SaaS), applications having huge CPU-intensive and which are very costly for actual deployment. For these types of applications, customer needs to analyze before taking rent of the resources (5).

Features:

- For improving the accuracy, relevant information of the application are taken out during emulation and is used during the simulation (5).

- Output from the emulation stage and input to the simulation stage were loosely coupled; the emulation generated a performance file that was later translated into a simulation model of the application (16).
- Information that is typically not disclosed by platform owners, such as location of virtual machines and number of virtual machines per host in a given time, is not required (8).

Cons:

- For emulation it is necessary a real cluster with Xen Hypervisor installed on the host

**MDCSim** (17)**:** It is a commercial discrete event simulator developed at the Pennsylvania State University. It helps the analyzer to model unique hardware characteristics of different components of a data center such as servers, communication links and switches which are collected from different dealers and allows estimation of power consumption. (8)

Features:

- Allows measuring power and analyze each layer of 3-layer architecture model and can modify any layer without affecting other layer of the architecture (5).
- It can also model hardware characteristic such as links between two communication nodes and switches connected with these nodes (5).
- It is supplied with specific hardware characteristics of data server components such as servers, communication links and switches from different vendors and allows estimation of power consumption (4).
- The simulator featured IBA and Ethernet communication protocols over TCP/IP and support many functions of IBA. There is no restriction in adding any new communication protocol (5).

Cons:

- The drawback of this simulator is that it is commercial (since it is built on CSIM (18), a commercial product (9)), so users need to buy it for full functionality (5).

**DCSim** (10)**:** It is an extensible data center simulator implemented in Java, designed to provide an easy framework for developing and experimenting with data center management techniques and algorithms. It is an event-driven simulator, simulating a data center offering IaaS to multiple clients. It focuses on modeling transactional, continuous workloads (such as a web server), but can be extended to model other workloads as well (10).

Features:

- Provides the additional capability of modeling replicated VMs sharing incoming workload as well as dependencies between VMs that are part of a multi-tiered application. SLA achievement can also be more directly and easily measured and available to management elements within the simulation (10).

· The resource needs of each VM in DCSim are driven dynamically by an Application, which varies the level of resources required by the VM to simulate a real workload (10).
· Multi-tier application model which allows simulating interactions and dependencies between VMs, VM replication as a tool for handling increasing workload, and the ability to combine these features with a work conserving CPU scheduler (10).

Cons:

· Simulations as large as 10000 hosts and 40000 VMs can be executed in approximately 1 hour (10).

**IcaroCloudSimulator:** It is a cloud simulator developed in the iCaro project. The main aim of this simulator is to analyze the changes on workload in a data center when the structure of this last is modified dynamically in real time. In the simulators previously described the structure of data center is changed, adding virtual machine for satisfy a peak of workload. In this simulator, for the iCaro project, we are interested in “how the workload changes on the data center if a new set of virtual machine, with its own services and applications, is added to this last?”. That is, in this case the adding of virtual machine makes the management of workload more difficult.

Features:

· It is possible to save and to reload information about the structure of created data center from an ontological database.
· The simulated values are saved on RRD format so that is possible to read them, for example, with Nagios (Nagios is a powerful monitoring system). So it is possible to create a monitor tool on Nagios with simulated values
· The values are simulated by traces recorded from a real host. These traces include workload of CPU, Memory, Storage and Bandwidth takes at the same time. In this manner, the values of traces are correlated between them.
· The simulator as an intuitive and simple to understand GUI.

Cons:

· The simulator is developed as a web application and this can affect the performance of the simulation, but improve portability.

In the following table the cloud simulator previously described are compared mainly on these attributes (5):

- ***Underlying Platform**:* Some simulators are built upon any existing simulation framework. The features of existing platform are inherited in the new simulation framework.
- ***Availability**:* This is important to know the availability of a simulator is commercial or open source.
- ***Programming Language**:* Most of simulator uses Java language for scripting or modeling any system. This is very important, since the users have to learn the language first to use the simulator.

- ***Cost Modeling**:* Since pay-as-you model go is one of fundamental service of cloud computing, or utility computing and one of the challenging issues of cloud simulator. The user can model any new policy by using the simulator that has this module.
- ***Graphical User Interface**:* Graphical user interface is for visual purpose and for simplicity when modeling. Many of the above simulators have an interactive GUI.
- ***Communication Model:*** Communication Model is one of the important in cloud computing, especially for networking within the data center and message passing between applications.
- ***Simulator Time**:* This is the execution time of the simulator during testing. This will determine whether simulator is heavy.
- ***Energy Modeling**:* Energy modeling is very important in cloud computing research because of huge energy consumption in the data center and various networking elements (router, switch etc.).
- ***Federation Policy**:* Since, cloud is distributed system. Many cloud service providers are located in different geographical locations. The federation policy allows coordinating different cloud service provider that supports inter-networking of application and workload migration to benefit high quality of service.
- ***Services*** (19)*:* The type of Cloud Services supported by the simulator (e.g. IaaS, PaaS, and SaaS).

At this list of attributes taken from (5), we added other self-explanatory attributes taken from paper indicated near the attribute name. The symbol '-' in a cell means that the simulator presents in the column was not compared on the attribute presents in the row with another simulator by the author of the above paper.

Other attributes are not self-explanatory and they are described here:

- ***Change Structure during Simulation**:* is set to yes if it is possible, for example, to add one or a set of new virtual machines during the simulation of a data center. The possibility to perform this operation it is important to analyze changes on the workload of the data center, if independent from previous workload a virtual machine is added. This operation is different from adding virtual machines to satisfy peak of workload.
- ***Full Description of Virtual Machine***: this attribute indicates if virtual machines in the simulation are fully described with information about, OS, CPU, memory, storage, service and application and not merely with CPU and memory. That is, the value is set to yes if each virtual machine has its own "identity". This attribute is important for high level simulation: if it is possible to add a virtual machine during the simulation that needs a service in other virtual machine, then the workload of the first machine affect the workload of the second.
- ***Programming Skill for Create Scenarios**:* is set to yes, with an indication of what language it is necessary, if the skill is required to generate scenarios of the simulation.

| ***Attribute/Simulator*** | ***CloudSim*** | ***CloudAnalyst*** | ***NetworkCloudSim*** | ***EMUSIM*** | ***GreenCloud*** | ***iCanCloud*** | ***MDCSim*** | ***DCSim*** | ***IcaroCloud Simulator*** |
|---|---|---|---|---|---|---|---|---|---|
| *Underlying Platform* | SimJava | CloudSim | CloudSim | AEF<br>CloudSim | Ns2 | SIMCAN (5)<br>OMNET, MPI (9) | CSIM | - | - |
| *Available* | Open Source | Open Source | Open Source | Open Source | Open Source | Open Source | Commercial | Open Source (GitHub) | - |
| *Programming Language* | Java | Java | Java | Java | C++/OTcl | C++ | Java/C++ | Java | Java |
| *GUI* | No | Yes | No | No | Limited | Yes | No | No | Yes |
| *Cost Model* | Yes | Yes | Yes | Yes | No | Yes | No | Yes | No |
| *Application model* (4) | Computation<br>Data transfer | Computation<br>Data transfer | Computation<br>Data transfer | - | Computation<br>Data transfer<br>Execution deadline | - | Computation | Multi-tier<br>Sharing workload (10) | Multi-Tier |
| *Services* (19) | IaaS | IaaS | IaaS | - | IaaS | IaaS | - | IaaS, PaaS | IaaS, Paas |
| *Communication Model* | Limited | Limited | Full | Limited | Full | Full | Limited | No | No |
| *Support of TCP/IP* (4) | No | - | - | - | Full | - | No | | No |
| *Energy Model* | Yes | Yes | Yes | Yes | Yes<br>Precise (Servers, Network) (4) | No | Rough<br>(Server Only) (4) | Rough<br>(Host Only) (10) | No |
| *Power Saving Modes* (4) | No (4)<br>Yes (If create by user) (7) | - | - | - | DVFS, DNS and both | - | No | - | No |
| *Simulation Time* | Seconds | Seconds | Seconds | Seconds | Minutes | Seconds | Seconds | Minutes | Seconds |

| ***Attribute/Simulator*** | ***CloudSim*** | ***CloudAnalyst*** | ***NetworkCloudSim*** | ***EMUSIM*** | ***GreenCloud*** | ***iCanCloud*** | ***MDCSim*** | ***DCSim*** | ***IcaroCloud Simulator*** |
|---|---|---|---|---|---|---|---|---|---|
| *Simulation Type* (8) | Event Based | Event Based | Packet Level | Event Based | Packet Level | - | Event Based | Event Based (10) | Event Based |
| *Federation Policy* | Yes | Yes | Yes | No | No | No | No | No | No |
| *Models for public cloud providers* (9) | No | No | - | - | No | Amazon | No | No | No |
| *Physical models* (9) | No | No | - | - | Available using plug-in | Full | No | No | No |
| *Change Structure During Simulation* | No | No | No | No | No | No | - | No | No |
| *Full Description of Virtual Machine* | No<br>Yes (if created by user) | No | No | No | No | No | - | No | Yes |
| *Programming Skill For Create Scenarios* | Yes (Java) | No | Yes (Java) | No | Yes (Tcl) | Yes (NED) | - | Yes (Java) | No |
| *Inserting in other project* | Yes (Java Library) | No | Yes | No | No | No | No | No | No |

**Table 1 – Comparison between software cloud simulators previously developed**

# 2. Requirements

- **R1**: The system should allow simple and fast generation of all entities connected to a data center or a business configuration. It should be possible, therefore, the insertion of information related to these entities and, with these data, the generation of an XML file described syntactically by the schema provided by KB.
- **R2**: The system should allow simple and fast generation of entities related to ServiceMetric, to consent study of other tools that analyze the KB.
- **R3**: For all entities generated as XML, the system should allow saving files in local (in the device that is used to access the system). Furthermore, it should be possible sending the XML file to KB in order to make information persistent.
- **R4**: Business configurations should be created over an existing data center (contained in KB), that is, the system should allow a choice of one data center contained in the KB, over which to generate a business configuration.
- **R5**: The system should allow analysis of metrics, which are contained in KB and are associated with host and virtual machines.
- **R6**: The system should allow the collection of data related to real working Host Machines. From these data should be extracted patterns to obtain models of workload for using them during simulation.
- **R7**: The system should provide two types of simulation
  - A ***fast*** simulation in which is possible to select entities (host and virtual machine) that are to be simulated and to which should be possible associate one of the models generated at point R6.
  - Simulated data must be written in an RRD file and must be sent to NAGIOS server, so that another tool can calculate on simulated data, as well as real data, the High Level Metrics to save in KB.
  - A ***real-time*** simulation in which is possible to generate new patterns from those calculated at point R6. The user should be capable to modify and to add entities to simulated data center, for example, with the addition of new virtual machines. These changes are necessary for analyzing workload generated on the real data center from such modifications.

# 3. Domain Model

The domain model used by this cloud simulator is based on the ontology under the KB. For this reason, many entities are similar, if not equal, to those contained on KB, except for few entities that have been created to simplify development of the simulator.

In detail the classes that begin with word *Group* are used to take the data inserted by a user on the forms and with these to create a determinate number (indicated by a user on the forms) of an object whose name is indicated by the rest of the name of the creator class eliminating the word *Group* (i.e. *GroupHostMachine* is the class responsible to create object of class *HostMachine*). While the ontology is represented (Figure 2) in hierarchical schema, the domain model (Figure 1) highlights the associations from the classes. A very important association to note is one that associates the *VirtualMachine* class to *IcaroService* class: with this association is possible to connect the part of the ontology/domain model describing business configuration to part describing data center. This association allows connection from perspective of the virtual things to perspective of the physical things.

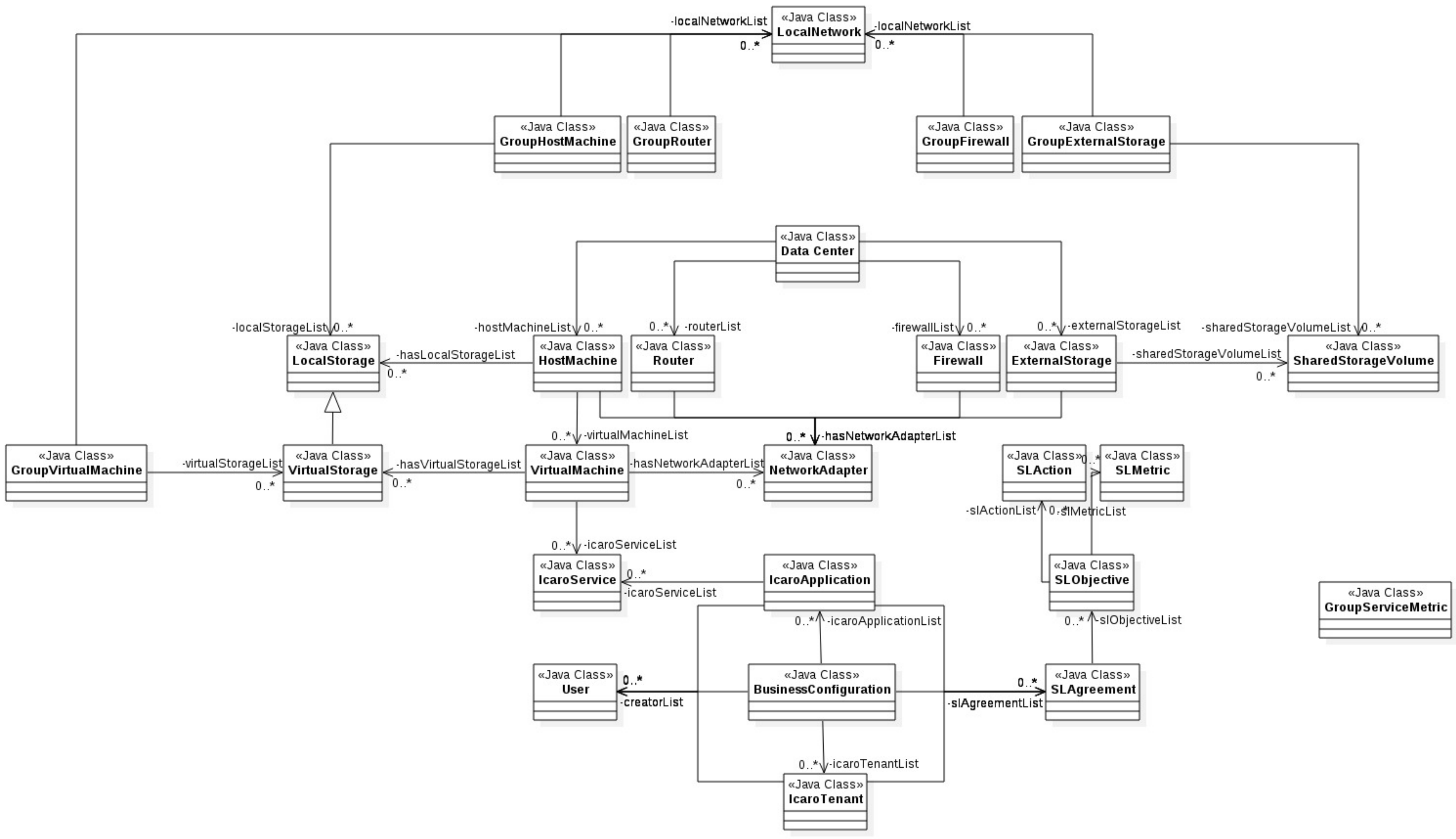


**Figure 1 – Domain model under Icaro Cloud Simulator**

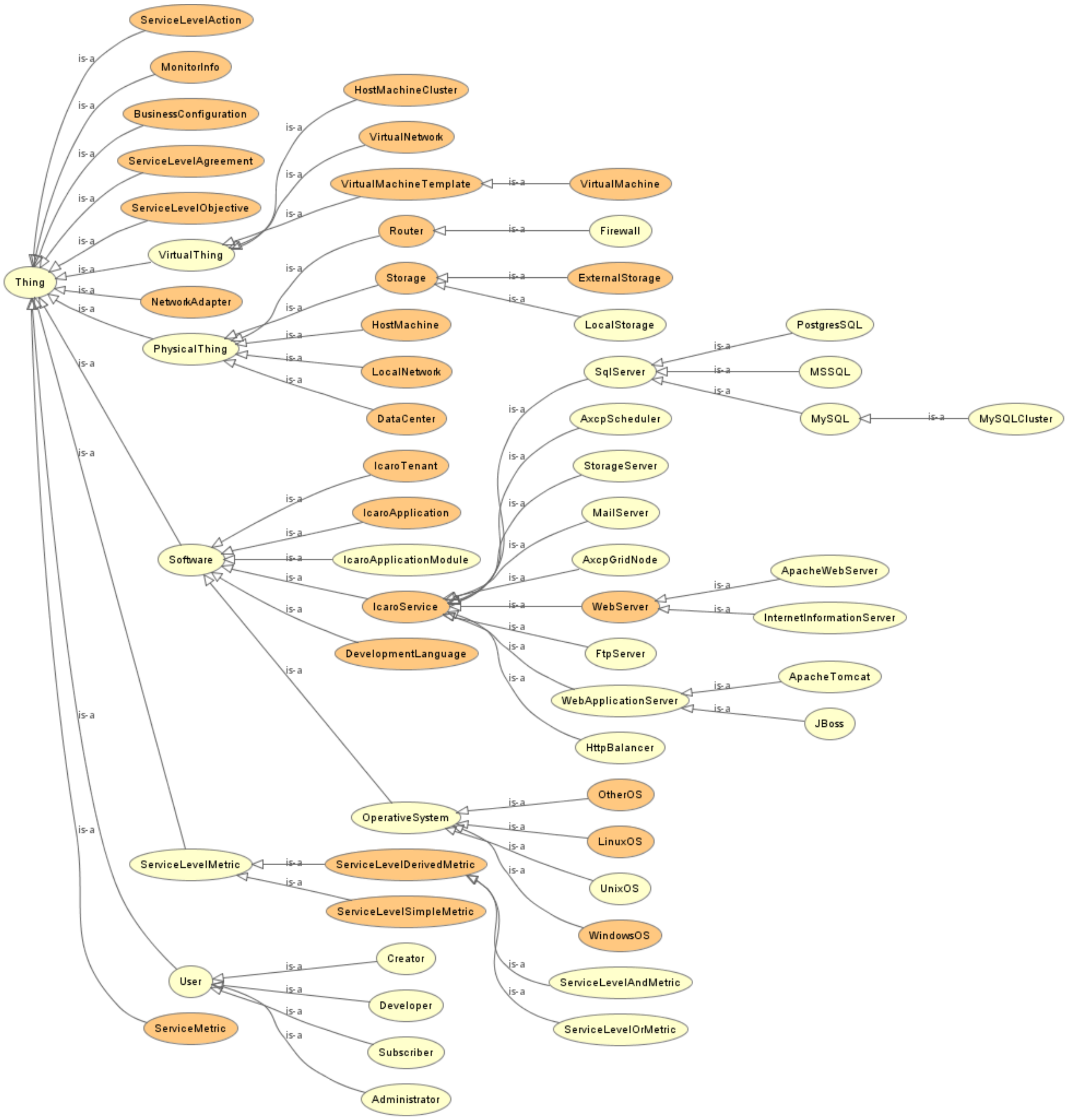


Figure 2 – Ontology under Knowledge Base

# 4. References

1. *Cloud computing: state-of-the-art and research challenges.* **Q. Zhang, L. Cheng, R. Boutaba.** s.l. : Journal of Internet Services and Applications, Vol. 1, pp. 7-18.

2. *Cloud computing: Issues and challenges.* **T. Dillon, C. Wu and E. Chang.** Perth : IEEE, 2010. Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. pp. 27-33.

3. *Challenges towards Elastic Power Management in Internet Data Centers.* **J. Liu, F. Zhao, X. Liu, and W. He.** Montreal, Quebec, Canada : s.n., 2009. Workshop on Cyber-Physical Systems (WCPS),.

4. *GreenCloud: A Packetlevel Simulator of Energy-aware Cloud Computing Data Centers.* **D. Kliazovich, P. Bouvry, S. U. Khan.** 3, s.l. : Journal of Supercomputing, 2012, Vol. 62, p. 1263-1283.

5. *Cloud computing simulators: A detailed survey and future direction.* **A. Ahmed and A. S. Sabyasachi.** Guargon : IEEE, 2014. Advance Computing Conference (IACC), 2014 IEEE International. pp. 866 - 872.

6. *The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures.* **F. P. Tso, D. R. White, S. Jouet, J. Singer and D. P. Pezaros.** 2013. Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on. p. 108-112.

7. *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms.* **R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and Rajkumar Buyya.** 1, New York : Wiley Press, 2011, Software: Practice and Experience (SPE), Vol. 41, pp. 23-50. ISSN: 0038-0644.

8. *Study and Comparison of CloudSim Simulators in the Cloud Computing.* **R. Malhotra and P. Jain.** 4, 2013, SIJ Transactions on Computer Science Engineering and its Applications (CSEA), Vol. 1, pp. 111-115.

9. *iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator.* **A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero and I. M. Llorente.** 1, s.l. : Springer, 2012, Journal of Grid Computing, Vol. 10, p. 185-209.

10. *DCSim: A data centre simulation tool for evaluating dynamic virtualized resource management.* **M. Tighe, G. Keller, M. Bauer, and H. Lutfiyya,.** 2012. The 6th International DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standard and the Cloud.

11. *CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications.* **B. Wickremasinghe, R. N. Calheiros and R. Buyya.** s.l. : IEEE Computer Society, 2010. 24th International Conference on Advanced Information Networking and Applications (AINA). p. 446-452.

12. ***The Network Simulator Ns2.*** **[Online] 2010. http://nsnam.isi.edu/nsnam/index.php/Main_Page.**

**13.** ***Introduction of Cloud Computing and Survey of Simulation Software for Cloud.*** **M. Aggarwal. 13, 2013, TIJ's Research Journal of Science & IT Management, Vol. 2. 2251-1563.**

**14.** ***CloudNetSim - Simulation of Real-Time Cloud Computing Applications.*** **T. Cucinotta and Santogidis A. Paris : s.n., 2013. 4th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS).**

**15.** ***NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations.*** **S. Kumar Garg and R. Buyya. Melbourne : IEEE CS Press, 2011. Proceedings of the 4th IEEE/ACM International Conference on Utility and Cloud Computing.**

**16.** ***EMUSIM: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of Cloud computing applications.*** **R. N. Calheiros, M. A.S. Netto, César A.F. De Rose and R. Buyya,. 5, Software: Practice and Experience, Vol. 43, p. 595–612.**

**17.** ***MDCSim: A multi-tier data center simulation, platform.*** **Seung-Hwan Lim, Bikash Sharma, Gunwoo Nam, Eun Kyoung Kim and Chita R. Das,. New Orleans : IEEE, 2009. Cluster Computing and Workshops, IEEE International Conference on.**

**18. CSIM Development Toolkit for Simulation and Modeling. [Online] http://www.mesquite.com/..**

**19.** ***Modeling and Simulation Frameworks for Cloud Computing Environment: A Critical Evaluation.*** **A. Bashar.**