ICARO Cloud Simulator Exploiting Knowledge Base

Claudio Badii, Pierfrancesco Bellini, Daniele Cenni, Ivan Bruno, Paolo Nesi
Distributed Systems and Internet Technology Lab, DISIT Lab,
Department of Information Engineering
University of Florence, Florence, Italy, tel: +39-055-2758515, fax: +39-055-2758570
http://www.disit.dinfo.unifi.it, paolo.nesi@unifi.it

Abstract
Allocation changes on cloud are complex and time consuming tasks, on cloning, scaling, etc. A solution to cope with these aspects is to perform a simulation. Cloud simulators have been proposed to assess conditions adopting specific models for energy, cloud capacity, allocations, networking, security, etc. In this paper, ICARO cloud simulator is proposed. It has been specifically designed for simulating the workload on the basis of real virtual machine loads and for simulating configuration and behaviour for wide temporal windows. This approach can be useful to predict and simulate the allocation of virtual machines on hosts and, thus, data centers on the basis of behaviour for days, weeks, months, etc. (for example for seasonal prediction of workloads). The proposed research has been developed in the context of ICARO Cloud research and development project.

Keywords: cloud simulation, cloud workload, cloud simulation review, knowledge modeling, cloud ontology.

1. Introduction

Relevant multimedia infrastructures are using cloud based approaches to manage resources, and set up high availability solutions addressing NIST (National Institute of Standards and Technology) layers: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) [Zhang et al., 2010]. Public and Private Cloud infrastructures are becoming every year more complex to be managed, especially for process configuration and management, dynamic scaling for elastic computing, healthiness control, etc. [Zhu et al., 2011]. Several different resource definitions are available and corresponding relationships among entities on the cloud can be established. Thus, every day new models and types are added especially at SaaS level, increasing complexity and demanding a very high level of flexibility in cloud management and definitions. These can be related to structures and resources on the cloud (e.g., hosts, VM, services, storages, process, applications, nets, etc.), on their corresponding Service Level Agreements, SLA; and on the metrics to be assessed for computing the business costs of the business on cloud on ”as a service” basis. In Private and Public Cloud multimedia infrastructures as social networks, content delivering network, open data provider, smart city service providers, many different solutions can be adopted for managing the cloud by the allocation of services and their monitoring according to layers: IaaS, PaaS and SaaS. The solutions can have a different coverage with respect to the cloud layers and are provided by industry (e.g., IBM, CISCO, HP, VMware, etc.) or open source (e.g., OpenStack [OpenStack], OpenNebula [OpenNebula]). Public Cloud providers have used some cloud brokers or mall for presenting the offer of services to their clients with a number of ready to deploy “almost standardized” configurations. Specific cloud market places can be adopted by the Public Cloud Service Provider, CSP, for example by using the solutions of IBM, CISCO, Parallels. In most cases, simple configurations at level of SaaS are supported in the offers and thus the whole cloud infrastructure is populated by a set of almost standardized simple single- and two-tiers solutions, while multimedia services are much more complex and imply multitier solutions. These complex solutions present periodic behaviors on week, month or season.

In the literature, a number of cloud simulators allow to make reasoning on cloud configurations, mainly addressing low level aspects of communicating processes on cloud such as CloudSim [Buyya et al., 2009], NetworkCloudSim [Kumar and Buyya, 2011], and MDCSim [Lim et al., 2009], or on energy consumption as

GreenCloud [Kliazovich et al., 2012], iCanCloud [Núñez et al., 2012] even modeling the hardware aspects. To this purpose, a number of simulators are based on direct mathematical models for: energy consumption (relating clock, storage access, and bandwidth to power consumption and temperature), network simulation in terms of packets, storage and database simulation in terms of latency, etc. Cloud simulators at the state of the art do not allow simulating effects of changing complex business configurations in the cloud, with the aim of exploring, assessing and predicting the best resource allocation for long temporal windows in advance; for example for weeks and months.

This article describes the ***ICARO Cloud Simulator*** for modeling and simulating complex cloud configurations such as those associated with multitier multimedia infrastructures. The *ICARO Cloud Simulator* is based on (i) a Cloud Knowledge Base, KB, definitions for the modeling of cloud resources, models, SLA, networks, software applications, and their evolution [Bellini et al, 2015]; (ii) extended monitoring and supervising tools (shortly described in this paper); (iii) smart cloud solution are described in [Bellini et al, 2015]. The adoption of a KB approach to model the cloud knowledge with a cloud ontology and its instances can be a solution to enable the reasoning on cloud structures, and thus for implementing strategies of smart cloud management and intelligence [Bellini, Cenni and Nesi, 2015]. The proposed *ICARO Cloud Simulator* has been developed in the context of the ICARO Cloud research and development project (http://www.disit.or/5482); it can be exploited in connection with other cloud tools such as configurators, orchestrators, monitoring, etc. Thanks to the Knowledge Base, KB, the proposed *ICARO Cloud Simulator* is particularly suitable for managing complex configurations and related SLA. The ICARO solution has been developed and tested on the cloud infrastructure of Computer Gross. Computer Gross is a cloud service provider for IaaS, PaaS and SaaS, in which allocated applications at SaaS level are provided by several different vendors, and belong to categories of multitier solutions for CRM (Customer Relationship Management), ERP (Enterprise Resource Planner), workflow, marketing, business intelligence, etc. This variety increases complexity of cloud management, and motivates the needs of flexible smart cloud engines that can simulate the workload in advance for longer temporal windows taking into account real workloads.

This paper is structured as follows. In Section 2, the related work regarding cloud simulator is reported putting in evidence the problems that guided our research and development. Section 3 describes the architecture of the ICARO Cloud Simulator in relationship with the classical elements of clouds. In Section 4, details about the ICARO cloud simulator are reported. Section 5 reports some experimental results. Conclusions are given in Section 6.

2. Related Work on Cloud Simulators

Allocation changes on cloud are complex and time consuming task, on cloning, scaling, etc. A solution to cope with these aspects is to perform a simulation. At the state of the art a number of cloud simulators have been proposed and are mainly suitable to simulating and assessing specific cases and workloads, adopting specific models for energy, cloud capacity, allocations, networking, security, etc.

Complex Business Configurations, BCs, need to be allocated on the cloud to satisfy specific demands. These configurations includes a number of hosts and Virtual Machines, VM, with many services/applications arranged as multitier solutions. When several BCs need to be allocated or changed on cloud the assessment of free resources into a set of hosts or external storages (CPU, memory, network and storage) cannot be based on the simple estimation of the current conditions in a limited time interval. A deeper simulation of the cloud conditions with a longer time forward is needed. Thus, the same host could have some VMs with heavy work during the day time and quite at nighttime, while the other VMs on the same host could have a complementary behavior in time, with typical weekly, monthly and seasonal behavior. A new allocation on the cloud may imply changes into the distribution of resource exploited in the cloud. The duty of a cloud simulation should include the verification of resource consumption and assessment of capability. Not all the configurations can be viable. For example, by deploying a VM on a given Host may be unfeasible for the lack of resources (i.e., CPU clocks and/or memory). The resources on a given host may be over-assigned by the VMs to exploit the compensation of the different CPU and memory exploitation during the day, week and months of different allocated VMs.

The present state of the art of cloud simulators is quite wide. A number of survey on cloud simulation has been presented [Ahmed and Sabyasachi, 2014], [Pandey and Gonnade, 2014], [Aggarwal 2013], putting in evidence the different kind of purpose of the simulators.

According to our analysis, **CloudSim** [Buyya et al., 2009] is the most popular cloud simulator, developed in Java as a library has been used as a basis for other simulators as **CloudAnalyst** [Wickremasinghe et al., 2010] in which the GUI and network modeling have been added, **NetworkCloudSim** [Kumar and Buyya, 2011] in which networks topologies/aspects are addressed supporting HPC, e-commerce and workflows. CloudSim is mainly focused on modeling IaaS aspects, allocating VM into single and multiple datacenters. CloudSim environment allows simulating specific configuration by programming, exploiting a limited number of aspects in modeling cloud resources at level of PaaS and SaaS that are left to the high level programming. On the other hand, it has been used for creating low level cloud simulators as: EMUSIM, CDOSim [Malhotra and Jain, 2013]. **GreenCloud** [Kliazovich et al., 2012] was based on Ns2 [McCanne and Floyd, 1997] a discrete cloud simulator implementing simulation of full TCP/IP. GreenCloud has been proposed for simulating the energy/power consumption aspects of cloud, and the networking level, thus suitable for simulating workload distributions and making decisions on the basis of mathematical models of energy consumption.
GreenCloud does not address higher level aspects of cloud stack and complex business configurations. GreenCloud presents a limited graphic user interface and provide low performance in simulation limiting the size of simulated clouds configurations. **MDCSim** [Lim et al., 2009] addressed the simulation of large scale multitier datacenters, taking into account the aspects related to NIST layers, communication aspects, etc. MDCSim is a library and does not provide a user interface, constraining to programming the cloud configuration to be simulated and the workload; and thus it presents limited capabilities in modeling and simulating complex business configurations that change over time. **iCanCloud** [Núñez et al., 2012] was developed with the aim of solving some of the limitation of CloudSim, GreenCloud and MDCSim. It is based on SIMCAN, OMNET, MPI, and provides the modeling of the infrastructure permitting the modeling of the hypervisor (with related math model that could be used for estimating power, temperature, costs, etc.) and can be executed on parallel instances. iCanCloud presents a relevant graphic user interface.
At the state of the art cloud simulators are mainly based on addressing low level aspects of communicating processes on cloud such as NetworkCloudSim and MDCSim, or on energy consumption as GreenCloud, iCanCloud even modeling the hardware aspects. To this purpose, a number of simulators are based on direct math model for: energy consumption (relating clock, storage access, and bandwidth to power consumption and temperature), network simulation in terms of packets, storage and database simulation in terms of latency, etc. Thus, it is very complex to make a full comparison of the different clouds since the consumed memory and speed in simulation strongly depends on the resource and mathematical models adopted [Núñez et al., 2012].

Attribute/ Simulator	CloudSim	CloudAnalyst	NetworkCloudSim	GreenCloud	MDCSim	iCanCloud
High level features						
GUI	No	Yes	No	Limited	No	Yes
Underlying Platform	SimJava	CloudSim	CloudSim	Ns2	CSIM	SIMCAN OMNET, MPI
Simulation Aim Focus						
Cost Model	Yes	Yes	Yes	No	No	Yes
Energy Model	No	Yes	Yes	Servers+ Network)	Rough Server	Yes
Power Saving Modes	partial	-	-	DVFS, DNS and both	No	-
Time and computing						
Simulation Type	Event Based	Event Based	Packet Level	Packet Level	Event Based	-
Parallel Simulation	No	No	No	No	No	--
Model of Resources						

Services	IaaS	IaaS	IaaS	IaaS	IaaS + service	IaaS
Description of VM	No	No	No	No	-	No
Communication Model	Limited	Limited	Full	Full	Limited	Full
Support of TCP/IP	No	-	Full	Full	No	-
Physical models	No	No	-	via plug-in	No	Full
Development						
Available	Open Source	Open Source	Open Source	Open Source	Commercial	Open Source
Programming Language	Java	Java	Java	C++/ OTcl	Java/ C++	C++
Programming	Yes (Java)	No	Yes (Java)	Yes (Tcl)	-	Yes (NED)

Table 1 – Comparison of Cloud Simulators. "No" not provided in the tool, in some cases can be customized.

A comparative overview is reported in **Table 1** in which the most relevant factors for comparing cloud simulators are reported and have been used to compare them. **High level features**: availability of a graphic user interface, the underlying platform. S**imulation aim and focus**, the main purpose of simulation such as for assessing: energy costs, power saving, capability model, scalability planning, etc. Depending on the simulation aim the resource model complexity can be more or less complex. For example, for the simple estimation of energy consumption a detailed modeling of services and connections among them would not be needed. **Time and computing aspects**: simulation type granularity, event address, parallel simulation, model, etc. This last feature refers to the possibility of simulating the whole data center on a parallel architecture. **Model of resources:** IaaS (host distribution and connections, network structure, and for each host CPUs, network, storage, memory), PaaS (operating system, VM modeling, allocated applications on the VM), SaaS (service modeling), description of VM, addressing communication model, support for the TCP/IP, addressing of t he physical model of the computer. **Development**: open sources or commercial, development language, API availability, etc.

As a result, the cloud simulators at the state of the art do not allow simulating the effects in the cloud related to the changes in complex business configurations, addressing Service Level Agreements, SLA, complex workload pattern models, with the aim of exploring, assessing and predicting the best resource allocation based on predicted consumption of resource in the real cloud infrastructure for a long time ahead. The above revised simulators are unsuitable to cope with huge amount of data produced by simulating the behavior for weeks. For example, the analysis of real monitored data from the services, VM and hosts in place, can be used to learn hurly, daily or weekly resources consumption patterns that can be used to produce a forward simulation and prediction.

3. ICARO Cloud Simulator Architecture

The proposed ICARO Cloud Simulator, **ICLOS**, is integrated in the context of the ICARO Cloud platform for Smart Cloud management [Belllini, et al., 2015]. In this Section, an overview of the ICARO architecture is reported to put in evidence the relationships of ICLOS with respect to the other components.

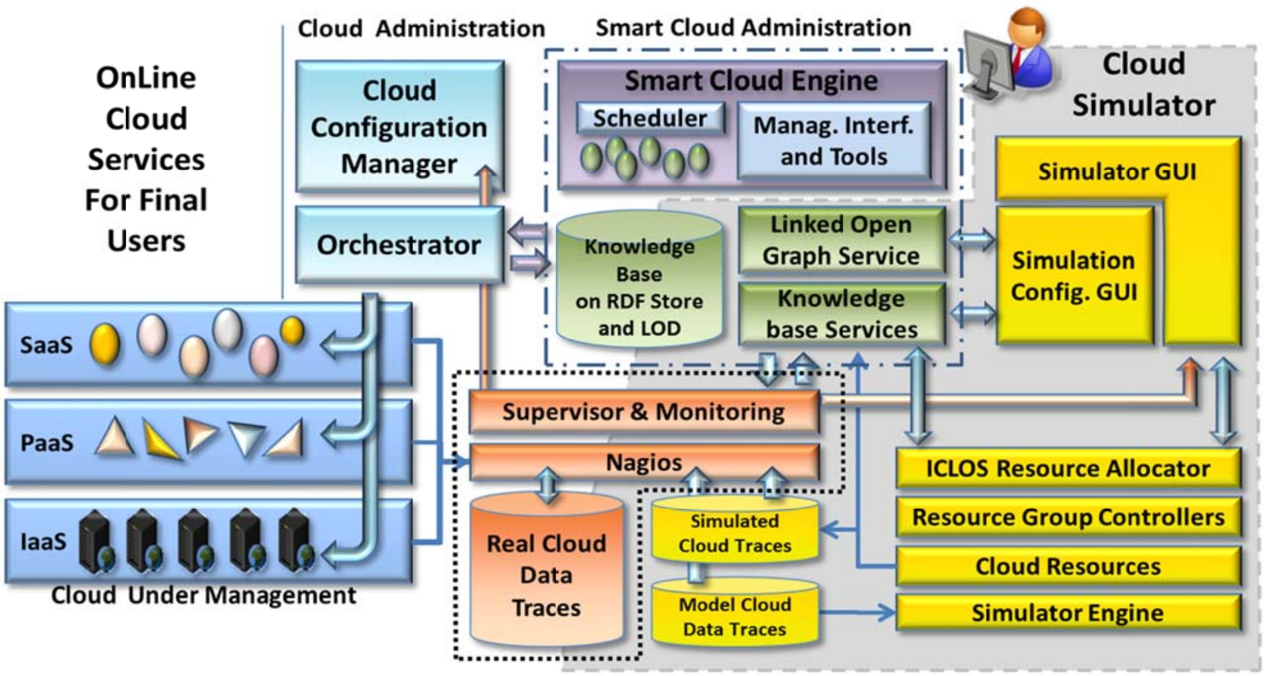

Figure 1 – ICARO Cloud Architecture with Cloud Simulator

The ICARO Cloud architecture is reported in **Figure 1** and includes six main areas:

(1) **Cloud under management**: on the left side the real cloud under management (including one or more datacenters) is depicted with its layers: IaaS, SaaS and PaaS. In pure simulation cases this part can be missing.

(2) **Cloud Administration area** including one or more commercial or open source Cloud Configuration Managers, CCMs, (any kind of cloud brokers and those mentioned in the introduction), and Orchestrators (e.g., VCO of VMware, or the Microsoft solution). In pure simulation cases this part can be missing.

(3) **Supervisor and Monitor, SM,** collects monitoring data from cloud resources, produces monitoring graphs and charts on demand, etc. Classical data are collected at level of IaaS (e.g., CPU, Memory, storage, network), at level of PaaS (e.g., operating system status), and SaaS (e.g., applicative metrics such as number of user, number of accesses, number of deploy/download, etc.).

(4) **Knowledge Base, KB**, can be invoked by any CCM or by the Orchestrator. The KB models the cloud knowledge in terms of structure, SLA, resources, actual values coming from the supervisor and monitor measured from the cloud. KB also manages the monitoring tools performing the automated configuration of monitoring issues related to the new resources configured by the CCM. The KB may model real as well simulated cloud configurations, datacenters and conditions;

(5) **Smart Cloud Engine**, **SCE**, exploits the Knowledge Base, KB, in which the cloud under control, and the simulated clouds are modelled. The SCE allows executing making decision processes on a distributed and parallel architecture to assess cloud healthiness and reconfiguration strategies [Belllini, et al., 2015].

(6) **Cloud Simulator, ICLOS,** simulates cloud conditions hourly, daily, weekly and yearly, taking into account real resource consumption patterns and exploiting complex configuration modelled into the KB.

Areas (3), (4) and (5) of the architecture are summarised in the following subsections, while the ICLOS (6) is deeply described in Section 4.

3.1. Smart Cloud Engine, SCE

The SCE is an autonomous engine for the supervised control of cloud resources, for the automation and optimization of services. The SCE [Belllini, et al., 2015] periodically checks the status of cloud resources in the cloud infrastructure (e.g., VM and application services) for each business configuration, BC, on the basis of the SLA. To this end, the SCE poses SPARQL queries to the KB modeling the cloud (real or simulated clouds) against additional rules with respect to those imposed in simulation. It can pose queries not only on the KB but also on any other external database. The KB has the detailed model of the cloud since any new resource allocated on the cloud is registered into the KB by the cloud administration tools. The KB is feed by the SM (which in turn is based on NAGIOS monitoring tool [Nagios]).

The SCE executes a set of decision rules associated with cloud resources (e.g., Host, VM, services, switch, etc.) and SLA/BC. Each decision rule is typically composed by:

- **An *assessment condition***, that if true activates the actions. The assessment condition estimates the resource healthiness, verifies the contractual conditions of the SLA, etc. For example, if a BC is getting low in resources, according to the SLA a scale out strategy is planned.
- **One or more *actions*** that correspond to the activation of strategies and procedures, for example for: scaling, reconfiguration, migration, cloning, balancing, etc. Actions can be configured to invoke remote calls (REST or WS or local calls) towards the CCM or the Orchestrator or other.

Thus, thousands and thousands of SCE processes are executed per day, on a distributed scheduler. With the aim of detecting critical conditions and making decision in real time, the provides a distributed scheduler engine with cluster functionality that allows adding new scheduling nodes and defining jobs, for smart cloud management, without service downtime.

The SCE presents a graphic user interface which includes: process definition and monitoring, decision configuration, connection to actions, etc.

3.2. Knowledge Base, KB

The cloud Knowledge Base, KB, stores the configuration and the status of the cloud under control (as well as of simulated cloud configurations or mixt). The collected model includes services ranging from the data center infrastructure to SW applications structure as well as the applicative metrics definitions and values. A review on knowledge base usage in the context of cloud can be recovered in [Bellini, Cenni and Nesi, 2015]. The variety of modelled resources in ICARO KB is higher with respect to the models adopted in the above mentioned simulators. In ICLOS, the KB is adopted for modeling configuration, making decision by comparing possible configurations with the actions, and at the basis of the simulation. Therefore, the KB models: datacenter, hosts, VMs, networks, net devices, SLAs, metrics, users, SW applications, operating systems, etc. (see **Figure 2**). The usage of a KB enables the reasoning on cloud structures and resources, and thus for implementing strategies of the SCE and simulation [Belllini, et al., 2015] .

The use of a KB facilitates interoperability among public and private clouds, and/or among different cloud segments managed by different cloud orchestrators or CCM. The KB allows formal verification and validation of resource cloud configuration, discovering and brokering services and resources, reasoning about cloud security, computing capability for horizontal or vertical scaling, thus elastic computing. The KB models and stores not only the structure of cloud components (infrastructure, applications, and configurations) but also the values of metrics of the components and their temporal trends (collected by the monitoring tools) to be able to answer questions such as “Which host machines can allocate a new VM?” or “The host machine H7897 have been over used in the last week?”, “Which VM is using most resources in the Host?”. However storing the full history of all metric values on the KB can be too expensive and unnecessary. Only high level metrics values are stored on the KB while the low level metrics are stored in the monitoring service (e.g., Nagios).

KB stores the application as a type and the application instances, and these can have specific constraints, as the number of services involved (e.g. number of front-end web servers). Therefore, to avoid duplicating type/ instance relation (modelled in RDF) and to leverage on the modeling features available in OWL2 to express constraints (e.g. max/min cardinality) we decided to represent the application model as an OWL Class [ICARO KB].

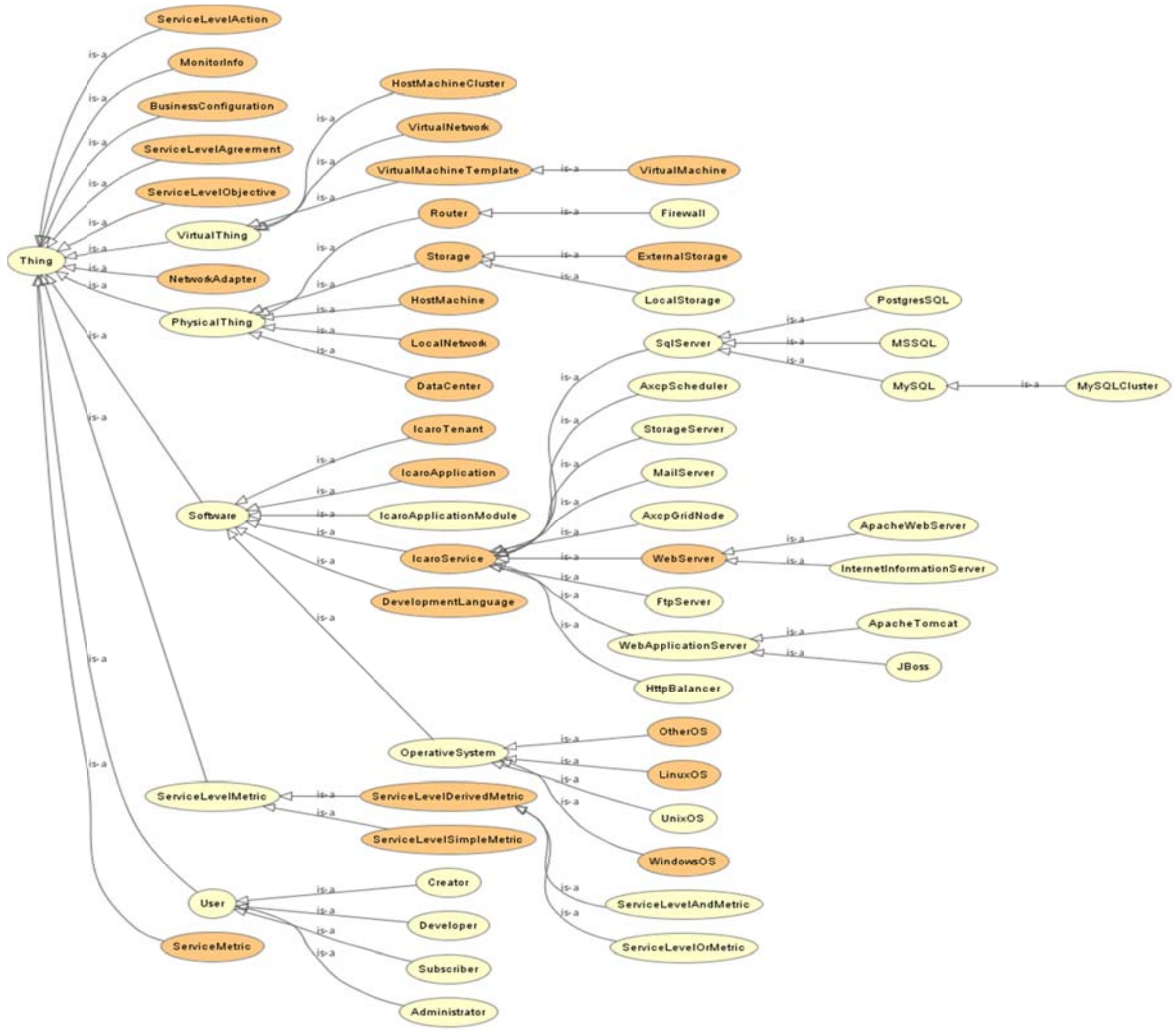

Figure 2 – Main "is-a" relationships in the ICARO Ontology of KB.

Another need is the possibility to aggregate different applications, servers, VMs to build a complete BC (e.g., an ERP with a CRM) and also to model applications tenants to be able to put application tenants in business configurations. The KB has to contain the SLAs associated with application or application tenants or with a whole business configuration. The SLA has been modelled as a set of Boolean expressions that relate high level metrics values of a component with a reference value.

KB Services are provided as REST APIs for accessing, configuring, modeling, setting SLA, and manipulating any cloud element and metric values on an RDF Store (currently an OWLIM-SE instance). When a complex data (e.g., a complex multitier configuration) is provided to be stored on the KB via an RDF-XML description, it is firstly validated and then stored. The KB also provides a SPARQL endpoint allowing making semantic queries for (i) the SCE assessment of the healthiness, (ii) the SCE decision criteria, (ii) the verification and validation of consistency and completeness of BC/SLA, etc.

Every time the KB is configured with a new resource to be monitored (a new host, VM, service, connection, etc.), it automatically sends a corresponding command to the SM to set up the specific monitoring processes for controlling services and resources (see **Figure 3**). Moreover, for facilitating the formalization of semantics queries a suitable graphical user interface based on Linked Open Graph has been used to access at the KB and browsing the semantic model [Bellini, Nesi, Venturi, 2014]. An instance of the ICARO KB applied to the DISIT data center can be accessed by the Linked Open Graph tool on the real time RDF store of the ICARO cloud tools at http://log.disit.org .

3.3. Supervisor and Monitoring, SM

The SM is a cloud monitoring engine. It collects data from cloud resources, storing them for historical reasons, provides relevant data to the KB, and produces monitoring graphs and charts. For the low level monitoring, the SM specifically uses drivers to manage multiple Nagios instances (not discussed in this article). The SM collects monitored values from the cloud IaaS, PaaS and SaaS levels and high level metrics, HLM. The SLAs are typically based on HLM as: the number of users registered on a social network, the number of downloads, the average number of connections, etc. All the collected data are stored in RRD (round-robin database) format. In this case, Nagios has been chosen but a different low level monitoring tool could be used. The approach of delegating the configuration of monitoring processes to the KB (see **Figure 3**) simplifies the work of the Orchestrator since all the monitoring issues do not have to be programmed into the deploy workflow, thus reducing the error prone process, the distribution of passwords, etc. Secondly, it allows to be sure that the SCE automatically adds all the monitoring processes that allow at the SCE to have all the needed information for controlling the BC and SLA. The SM is therefore automatically managed and configured by the KB. For all the collected data, the SM provides graphics and charts on demand to CCM (to be shown to the customers), as well as to the user interface of the ICLOS.

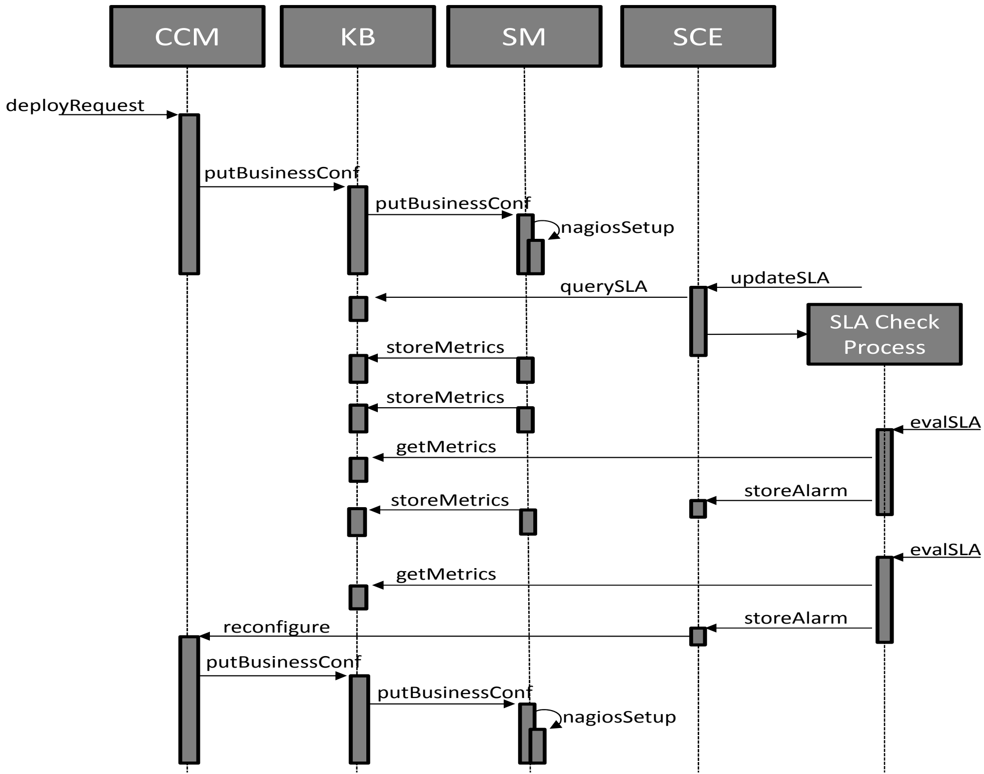

Figure 3 – ICARO Supervisor and Monitor

4. ICARO Cloud Simulator, ICLOS

Before discussing the structure of **ICLOS**, we present a short overview of the main requirements that have been identified to cope with the mentioned problems related to the Smart Cloud management for complex BC including SLA.

4.1. Requirements for ICLOS

As discussed in previous sections, real multimedia services, social networks, large web sites with CDN, crowdsourcing solutions, and smart city solutions, typically they need to manage:

- *complex BCs* as multitier architectures including several VMs, services, networks, services, processes, etc.;
- *articulated SLA* for avoiding violation of SLA and for controlling major cost parameters, taking decision, informing the customer and administrators, etc.;

- *strategies* activating elastic configuration processes for scaling on the front end, scaling on the database, scaling on the content ingestion of user generated content, scaling for computing suggestions, etc.;
- *several dependent resources*: hosts, VM, services, networks, applications, storage access, etc.;
- *resource consumption patters* that may provide aperiodic behavior as well as overlapped with periodic behavior at level of: hour, day, week, month and/or year. These factors can be due to the alternation of working hours, vacations, business orientation, seasonal commercial factors, and to eventual unexpected events, as the arrival of storm, etc. The trends about resource consumption for CPU, memory, storage, network, etc. are related each other, and thus the real BC profiling has to be take them into account as related patterns.

Most of the aspects are not addressed in a satisfactory manner by the simulators at the state of the art, see Section 2.

The problem of pattern production for cloud simulation has been addressed by Google Cloud Backend which performs a characterization according to their duration, CPU and memory requirements [Hellerstein, 2010]. The analysis of the data collected by the Performance Monitor may be used to perform a workload classification [Mishra et al., 2010], [Amoretti et al., 2013]. The workload patterns are exploited in the cloud simulation in the ICLOS solution. In reality, the sole statistical characterization of VM or hosts on the basis of CPU and Memory workload is not enough to cope with complex BCs like those described at the beginning of this section. The exploitation of SCE and Cloud Simulation based on real workload patterns derived from the monitoring log of the SM can be the path to setup a smarter cloud management engine [Germain-Renaud, Rana, 2009]. This can take decision about the cloud reconfiguration, addressing aspects of energy consumption, capacity planning, etc., with the aim of maintaining a high quality of service according to the SLA, and to the general objectives of the cloud service provider in terms of energy, costs, etc. Thus the SCE can activate reconfigurations, in/out scaling, load balancing, moving, cloning, etc.

4.2. Architecture of ICLOS Cloud Simulator

Figure 1 presented the general architecture of the ICLOS. As depicted in Figure 4 the ICLOS consists of a number of subsystems. SM and the KB subsystems have been described in Section 3 with the aim of presenting their role for the general cloud management level.

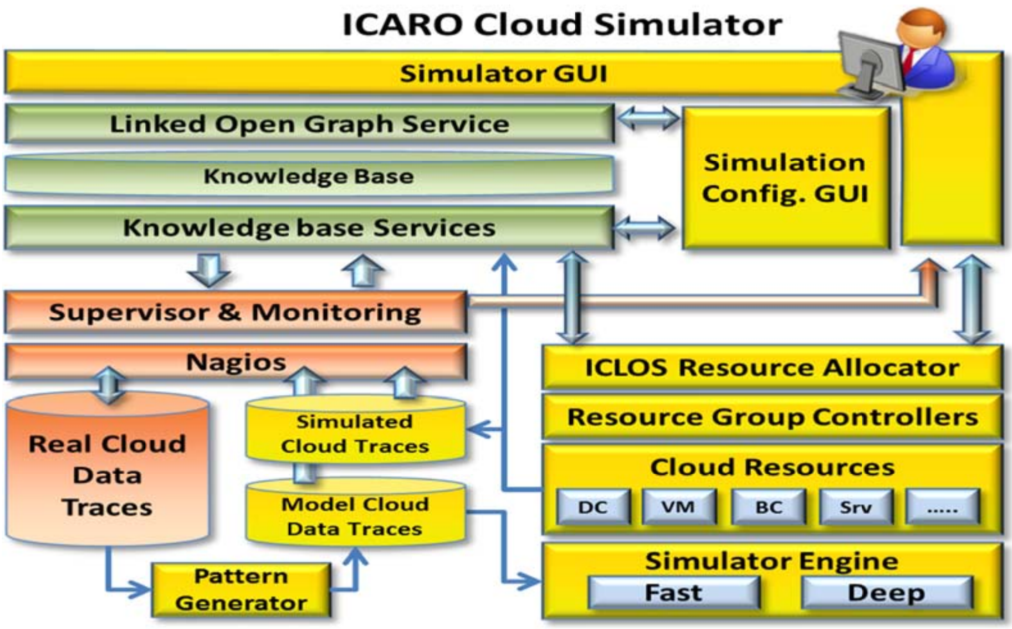

Figure 4 – ICLOS Architecture.

The elements of the ICLOS solution are described as follows.

Simulator GUI: is the user interface of the ICLOS simulator to: (i) set up a new configuration to be simulated, (ii) impose the configuration data, (iii) obtain the simulation results in terms of resource consumption graphs and general assessment results.

Simulation Configuration GUI: is a specific user interface for configuring parameters of the resources involved into the configuration to be simulated. A configuration to be simulated is produced and stored into the KB by sending an XML file. The ICLOS simulator starts from the KB to perform the simulation, and produce the result corresponding to the allocated resources into the Simulated Cloud Traces saved into RRD (round-robin database) format.

Pattern Generator: a set of tools to estimate patterns for workload of resources taking into account related CPU, memory, network, storage, etc. for day, hours, week, months, etc., of different VM, hosts and services of a BC. The tools exploit Real Cloud Data Traces in RRD format collected from Nagios/SM on the real cloud to produce patterns in the same format to be used into ICLOS. This allows the SM to show and export graphs of the workload patterns and of simulated results.

ICLOS Resource Allocator: on the basis of the configuration of resources it allows to allocate them into the cloud simulator memory.

Resource Group Controller: it allows managing the allocated resources addressing events and harmonising the math models for computation.

Cloud Resources: a collection of allocated resources according to the configuration produced. It may take into account multiple and incremental configurations. The resources that can be allocated in the simulator are in principle those modelled by the KB (see **Figure 2**), while in reality only some of them are allocated and deployed as described in the following.

Simulator Engine: the simulation model that can progress in estimating the output workload synchronously among all resources time instant by time instant (deep mode), or by computing the results on the basis of workload patterns associated to resources in the configuration phase and taken from the Model Cloud Data Traces in RRD format; thus, resulting in a faster simulation (Fast mode). The simulated values are those requested by the simulator during the configuration and coherently defined by the SLA for each BC.

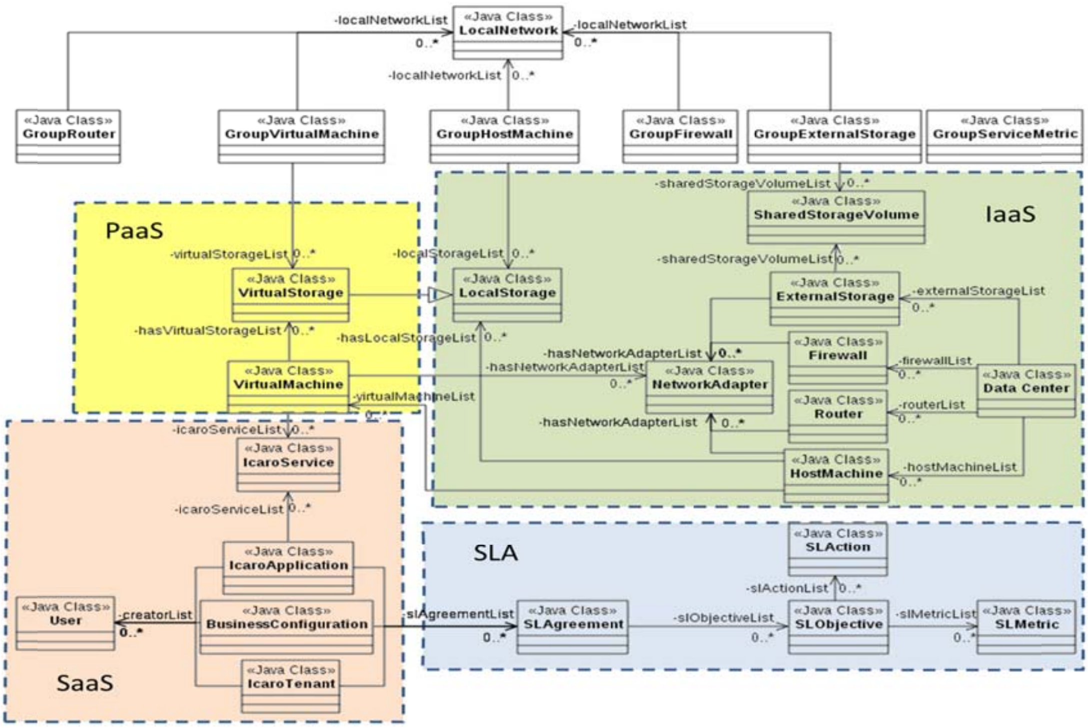

Figure 5 – ICLOS Modeling main classes.

The ICLOS simulator has been designed to model into the simulation the main KB classes and structures. In **Figure 5**, the main classes modeling layers IaaS, PaaS and SaaS aspects, the SLA and the group controllers are reported. According to the design pattern of Model View Control, a number of classes have been developed (not reported in Figure 5). They allow to view and model the inputting of data for each of the addressed cloud resources. On the other hand, their purpose is limited to the production of the XML file for feeding the KB. The main goal of the simulator is to simulate the workload and cloud model in general and save them for day, week, months, etc. in the SM and KM. This allows to: (i) model and simulate large cloud and complex configurations, (ii) activate the SCE rules for further analysis.

5. Experimental Results

In this section, experimental results about the usage of the ICLOS simulator are reported by providing some examples of simulations; details are reported in Table 2.

The reported measures have been performed by simulating a number of configurations to provide examples of the simulation costs, for example by considering:

- VM ranging from 1 to 3000, each of them with: CPU clocks per second equal to 2000 MHz, reserved CPU clocks per second equal to 800 MHz; RAM memory of 3 Gbyte, reservation memory space of 1 Gbyte.
- Hosts (case 1 and case 2) ranging from 1 to 10 (each of them with: 32 cores, 2500 MHz per core and 128 Gbyte Ram). Hosts in cases 3 and 4 have been scaled up consequently. In ICLOS, the costs of Host computing simulation is included into the VM model, so that the simulation time and storage is linear with the number of VMs, as it is shown in Table 2.

Simulation parameters and general measures	Case 1	Case 2	Case 3	Case 4
#host	1	10	1	1
#VM per Host	30	30	300	3000
Total number of VM	30	300	300	3000
HD space used for data output on RRD format, in Mbyte	36,1	361,2	350,7	3503
Simulation: measured times and computer metrics	Case 1	Case 2	Case 3	Case 4
Mean Total Time in ms	37500	385042	421954	4797872
Std Dev in the Mean Total Time	1316	16845	19117	228470
averaged total time / #VM	1250,01	1283,47	1406,51	1599,29
Mean Time Simulation of VMs + Hosts, in ms	9907	93437	90462	1061463
averaged computing time for simulating a VM +Host	330	311	301	298
Mean Time Simulation Hosts structure, in ms	274	2843	265	274
averaged computing time for simulating a host	274	284,35	265	274
Mean Time for Saving RRD data of VMs in SM storage via network, in ms	26353	279602	330001	3721905
averaged storage for simulation data per VM in Mbyte	1,20	1,20	1,169	1,167
Mean Time for Saving RRD data of Hosts in SM storage via network, in ms	962	9036	1159	1654

Table 2 – ICLOS Simulations

The ICLOS simulations have been performed by using workload patterns of 1 week in advance for resources (CPU, storage and memory) from the RRD of the SM with a measure every 5 minutes, thus simulating a whole week for the VM and hosts. Therefore, the input workload patterns have a value every 5 minutes and they can be specifically assigned or randomly selected from a set of real patterns taken from ECLAP social network, Sii-Mobility smart city aggregator tools, etc. from the DISIT data center in XML format (coming from RRD of SM).

Please note that the simulation of 1 week for 3000 VM/Host has been performed in about 80 minutes on a single server. The computing time can be allocated on multiple servers hosting the simulators, taking different segments of the cloud on KB to be simulated, since all computations are independent, and produce results directly on the ICARO RRD/XML of the SM (the SM provides high level results to KB). Please note that, the registered numbers from simulations as reported in Table 1 have been obtained as mean value taken from 20 simulations with the same parameters. The simulations have been executed on a Debian 64 bit, 6 Gbyte of memory, CPU 4 core, 2000 Mhz. ICARO Simulator has been developed in Java and runs on Tomcat.

The Mean Total Time refers to the time needed to execute the whole simulation including the reading of the patterns (CPU, memory, storage) for the whole VM, the computation of the VM and Host load and the save of the resulting data on SM in RDD format in a remote HD. The “averaged time / #VM” growths marginally passing from 300 to 3000 VMs (at 1599 ms) with an increment of the 13% of the mean computational and saving cost per VM. This increment is mainly due to the cost of writing and sending the RRD of VM into the store of the SM (see Figure 6). The computational time to simulate the 10 Hosts with 30 VMs for week (CPU, mem and storage) is of about 800 ms. On the other hand, the “Mean time Simulation of VMs + Hosts” reported in Table 2 also include for each VM the access on HD for taking the pattern, the XML parsing, the computation of simulation and the writing of the RRD/XML with the simulation. There, the mean time for simulating the host structure includes only the saving of the XML for the host, and thus it is almost constant being always for a week. Being the simulation time quite constant it is almost un-useful to perform simulations with higher number of VM and Hosts, with a storage needed of about 1.2 Mbyte of HD per each VM for a week.

Each Host simulation is performed autonomously and thus also the RAM memory used by the simulator is almost constant remaining under 120 Mbyte in all cases.

The simulation time cannot be easily compared with other simulators since in the case of ICLOS the simulations address longer time windows, and longer time lead also to spend time in saving the output data resulting from the simulation of all the VM and Hosts on the harddisk, with a sample every 5 minutes. Figure 6 reports the ICLOS simulation directly monitored into the SM tool that exploiting Nagios libraries to access and rendering the RRD storages.

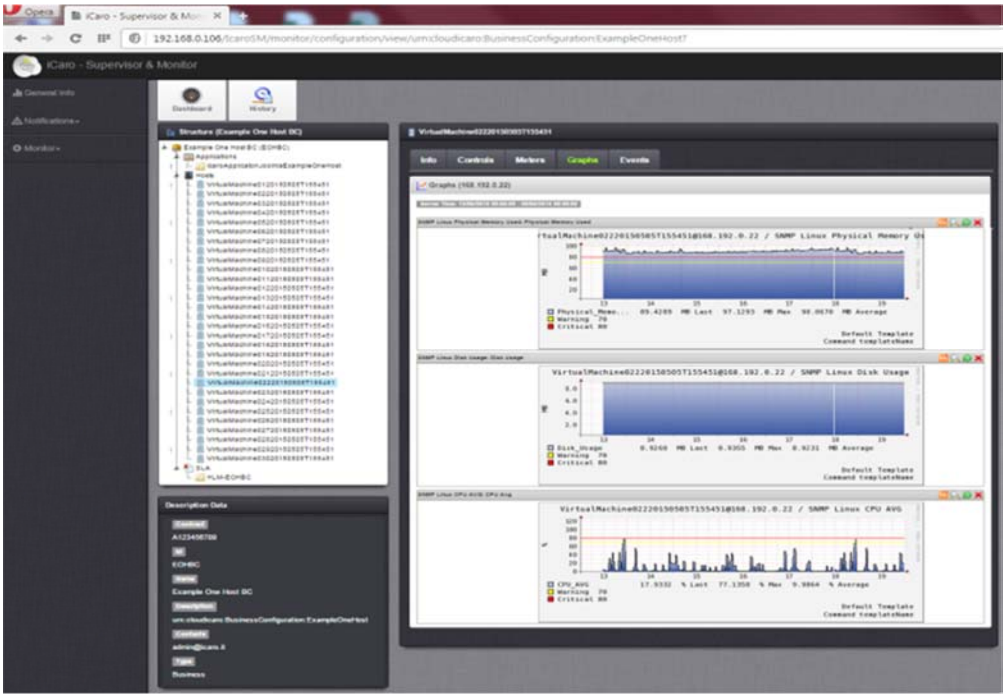

Figure 6 – ICLOS Simulator results on SM.

For example, for DC in [Tighe et al., 2012] the simulation of 1000 Hosts with 4 VM each, for a total of 40000 VMs computing a the energy consumption model only was performed in 3597 s on a Intel Core i7 930 processor and 6GB of RAM. The estimation has been assessment by performing 5 repetitions and the simulations were for 10 days, with a single data value every 10 minutes. The power consumption model has been modeled by using SPECpower benchmark [SPECpower 2012].

For the comparison, a similar simulation has been taken with ICLOS. Therefore, 1000 Hosts with 4 VM each, for a total of 40000 VMs were simulated by computing the energy consumption model SPECpower

benchmark [SPECpower 2012] by using input values every 5 minute and generating output simulated values very 5 minutes. The simulation has been performed 5 times on Debian 64 bit, 6 Gbyte of memory, CPU 4 core, 2000 Mhz, obtaining average time of 1985 s and a Std.Dev. = 245,89.
As a results, the ICLOS and DC simulators are comparable in terms of execution time.

6. Conclusions

In this paper, ICARO cloud simulator, ICLOS, has been proposed. It has been specifically designed for simulating the workload on the basis of real virtual machine patterns for their resources and behaviour for wide temporal windows. This approach can be useful to compute predictions via simulations of the allocation of virtual machines on hosts and, thus, data centers on the basis of supposed behaviour for days, weeks, months, etc. (for example for seasonal prediction of workloads). The proposed research has been developed in the context of ICARO Cloud research and development project. All the computations are directly producing results on RRD format on ICARO SM or Nagios and on KB. This means that the SLA and other analysis can be performed via the Smart Cloud Engine, SCE, and other tools. So that the simulation is sustainable for large data centers obtaining forecasts saved data for 3000 VMs for a week in 80 minutes accessible on hard disk on a single server, and can be scaled up by using multiple servers. The ICLOS simulator has been also used for simulating power consumption obtaining simulation time comparable with other simulator but providing more complete functionalities.

Acknowledgements

The authors would like to thanks to ICARO partners as Computer Gross and LiberoLogico and to the many people involved in the validation of results. The project has been founded by the Tuscany region in the Programme POR CREO.

Bibliography

[Aggarwal 2013] M. Aggarwal, Introduction of Cloud Computing and Survey of Simulation Software for Cloud, Research Journal of Science & IT Management, Vol. 2, 2013.

[Ahmed and Sabyasachi,2014] Ahmed, A.; Sabyasachi, A.S., "Cloud computing simulators: A detailed survey and future direction," IEEE International Advance Computing Conference (IACC), pp.866-872, 21-22 Feb. 2014, doi: 10.1109/IAdCC.2014.6779436

[Amoretti et al., 2013] Michele Amoretti, Francesco Zanichelli, Gianni Conte, Efficient autonomic cloud computing using online discrete event simulation, J. Parallel Distrib. Comput. Vol.73, 2013, pp.767–776.

[Bellini et al, 2015] Pierfrancesco Bellini, Daniele Cenni, Paolo Nesi, A Knowledge Base Driven Solution for Smart Cloud Management, IEEE Cloud 2015, NewYork, July 2015.

[Bellini, Cenni and Nesi, 2015] P. Bellini, D. Cenni, P. Nesi, Cloud Knowledge Modeling and Management, Chapter on Encyclopaedia on Cloud Computing, Wiley Press, 2015.

[Bellini, Nesi, Venturi, 2014] P. Bellini, P. Nesi, A. Venturi, "Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views", International Journal of Visual Language and Computing, Elsevier, 2014.

[Buyya et al., 2009] R. Buyya, R. Ranjan, and R. N. Calheiros, Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities, 7th Int. Conf. on High Performance Computing and Simulation, June 2009, pp.1–11.

[Germain-Renaud, Rana, 2009] C. Germain-Renaud, O.F. Rana, The convergence of clouds, grids, and autonomics, IEEE Internet Computing n.13, vol.6, 2009.

[Hellerstein, 2010] J.L. Hellerstein, Google cluster data. Posted at: http://googleresearch.blogspot.com/2010/01/google-cluster-data.html

[ICARO KB] ICARO Cloud KB, Ontology. D2.9.2, http://www.disit.org/5604

[Kliazovich et al., 2012] D. Kliazovich, P. Bouvry, S. U. Khan, GreenCloud: A Packet level Simulator of Energy-aware Cloud Computing Data Centers, Journal of Supercomputing, 2012, Vol. 62, p. 1263-1283.

[Kumar and Buyya, 2011] S. Kumar Garg and R. Buyya. Melbourne, NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations, Proceedings of the 4th IEEE/ACM Int. Conf. Utility and Cloud Computing, 2011.

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

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

[McCanne and Floyd, 1997] S. McCanne and S. Floyd, "Network Simulator ns-2," http://www.isi.edu/nsnam/ns/, 1997. http://nsnam.isi.edu/nsnam/index.php/Main_Page

[Mishra et al., 2010] A.K. Mishra, J.L. Hellerstein, W. Cirne, C.R. Das, Towards characterizing cloud backend workloads: insights from google compute clusters, ACM SIGMETRICS Performance Evaluation Rev., n.37, vol.4, 2010, pp.34–41.

[Nagios] http://*www.nagios.org*

[Núñez et al., 2012] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero and I. M. Llorente, iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator, Springer, Journal of Grid Computing, Vol.10, 2012, pp.185-209.

[OpenNebula] http://opennebula.org/

[OpenStack] *https://www.openstack.org/*

[Pandey and Gonnade, 2014] Ranu Pandey, Sandeep Gonnade, Comparative Study of Simulation Tools in Cloud Computing Environment, International Journal of Scientific & Engineering Research, Vol.5, n.5, May-2014.

[SPECpower 2012] "Standard Performance Evaluation Corporation," http://www.spec.org/, Aug 2012.

[Tighe et al., 2012] Tighe, M.; Keller, G.; Bauer, M.; Lutfiyya, H., "DCSim: A data centre simulation tool for evaluating dynamic virtualized resource management," Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm) , vol., no., pp.385,392, 22-26 Oct. 2012

[Wickremasinghe et al., 2010] B. Wickremasinghe, R. N. Calheiros and R. Buyya, CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications, 24th Int. Conf. on Adv. Inf. Networking and App. (AINA), IEEE, 2010, pp.446-452.

[Zhang et al., 2010] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." Journal of internet services and applications, 2010, pp.7-18.

[Zhu et al., 2011] Wenwu Zhu, Chong Luo, Jianfeng Wang, Shipeng Li, Multimedia Cloud Computing, Signal Processing Magazine, IEEE, Vol.28, n.3, pp.59-69.