

Contents lists available at [ScienceDirect](#)

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

ICARO Cloud Simulator exploiting knowledge base



Claudio Badii, Pierfrancesco Bellini, Ivan Bruno, Daniele Cenni,
Riccardo Mariucci, Paolo Nesi*

Distributed Systems and Internet Technology Lab, DISIT Lab Department of Information Engineering, University of Florence, Florence, Italy

ARTICLE INFO

Article history:

Received 6 June 2015

Revised 25 November 2015

Accepted 6 December 2015

Available online 16 February 2016

Keywords:

Cloud simulation

Cloud workload

Cloud simulation review

Knowledge modeling

Cloud ontology

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 workloads and for simulating complex business configurations and behaviours 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 real business configuration behaviour for days, weeks, months, etc. (for example, to predict workloads). The proposed research has been developed in the context of the ICARO Cloud research and development project.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Relevant multimedia infrastructures are using cloud based approaches to manage resources, and set up high availability solutions addressing National Institute of Standards and Technology (NIST) layers: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) [28]. Public and Private Cloud infrastructures are becoming every year more and more complex to be managed, especially as to process configuration and management, dynamic scaling for elastic computing, healthiness control, etc. [29]. Several different resource definitions are available and the 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. The latter 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 business costs on cloud on “as a service” basis. In Private and Public Cloud multimedia infrastructures such as social networks, content delivering networks, open data providers, smart city service providers, many different solutions can be adopted to manage the cloud by allocating services and their monitoring according to layers: IaaS, PaaS and SaaS. Solutions can have 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 [20], OpenNebula [19]. Public Cloud providers have currently used some cloud brokers or mall to present their range of services to customers 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,

* Corresponding author. Tel.: +39 55 2758515, fax: +39 55 2758570.

E-mail address: paolo.nesi@unifi.it, paonesi@gmail.com (P. Nesi).

URL: <http://www.disit.dinfo.unifi.it> (P. Nesi)

and 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 require complex business configuration and behaviors on week, month or season.

In the literature, a number of cloud simulators allows any reasoning on cloud configurations, mainly addressing low level aspects of communicating processes on cloud such as CloudSim [7], NetworkCloudSim [10], and MDCSim [13], or on energy consumption as GreenCloud [11], iCanCloud [18] even modeling the hardware aspects. A survey on mathematical models for cloud simulation is offered in [23]. 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 simulation of effects concerning complex business configurations with their real workload patterns in the cloud, so as to explore, assess and predict the best resource allocation for long temporal windows, for example for weeks and months.

This article describes the *ICARO Cloud Simulator (ICLOS)* for modeling and simulating complex cloud configurations such as those associated with multitier multimedia infrastructures. ICARO solution has been developed with the support of Tuscany with the European Funds on Regional Developments. ICARO is a platform for automating the cloud management for SME. ICARO itself is not an acronym it merely referring to Icarus of the Greek mythology joking on the fact that the cloud could save the life of SME preventing the burning their wings. The *ICARO Cloud Simulator* is based on (i) a Cloud Knowledge Base (KB) definitions for the modeling of cloud resources, model complex business configurations, SLAs, networks, and their evolution [4]; (ii) extended monitoring and supervising tools (shortly described in this paper); and (iii) smart cloud solution are described in [4]. 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 implementing strategies of smart cloud management and intelligence [5]. The proposed *ICARO Cloud Simulator* has been developed in the context of the ICARO Cloud research and development project (<http://www.disit.org/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 Business Configurations (BC), and related SLA, taking into account the real pattern workload behavior. For ICLOS a Business Configuration includes a number of resources that typically may range from a single or number of simple applications allocated on some VM, to a number of hosts, VMs and services connected together in somehow, belonging to the same company business, and allocated into some datacenter. A BC is a more general concept with respect to that of cloud application or subsystems. A BC has a specific SLA that governs the contractual rules connected with the general quality of service and are used by the Smart Cloud Engine (SCE) for activating eventual alarms but also reconfiguration actions. 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, where the allocated applications at SaaS level are provided by different vendors, and belong to categories of multitier solutions for Customer Relationship Management (CRM), Enterprise Resource Planner (ERP), workflow, marketing, business intelligence, social network, smart city, etc. This variety increases complexity of cloud management, and motivates the needs for flexible smart cloud engines which can carry out a workload simulation in advance for longer temporal windows, while always taking into account real workload patterns.

This paper is structured as follows. In Section 2, the related work regarding cloud simulator is reported highlighting the problems that ruled 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. Currently some cloud simulators have been proposed and are mainly suitable to simulate and assess specific cases and workloads, by 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 include 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 current conditions in a limited time interval, with simple workload patterns. A deeper simulation of the cloud conditions can be performed by using realistic workload partners and longer time forward is needed. Thus, the same host could have some VMs with heavy work during day-time and a bit less at night-time, while 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 resources exploited in the cloud. The duty of a cloud simulation should include the verification of resource consumption and the assessment of its capability. Not all the configurations can be viable. For example, by deploying a VM on a given Host may be unfeasible due to 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.

Table 1

Comparison of cloud simulators. “No” not provided in the tool, in some cases they can be customized.

| 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) |

The present state of the art of cloud simulators is quite wide. A number of surveys on cloud simulation have been presented [2,21,1,23,25] stressing the different kinds of goals meant for such simulators and their mathematical models.

According to our analysis, **CloudSim** [7] is the most popular cloud simulator; developed in Java as a library it has been used as a basis for other simulators as **CloudAnalyst** [27] where the GUI and network modeling have been added, **Network-CloudSim** [12] where 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 and exploiting a limited number of aspects in modeling cloud resources at level of PaaS and SaaS which are left to high level programming. On the other hand, it has been used to create low level cloud simulators as: EMUSIM, CDOSim [14]. **GreenCloud** [11] was based on Ns2 [15] a discrete cloud simulator implementing simulation of full TCP/IP. GreenCloud has been proposed to simulate the energy/power consumption aspects of cloud, and the networking level, thus suitable to simulate workload distributions and make decisions on the basis of mathematical models on energy consumption. GreenCloud does not address higher level aspects of cloud stack and complex business configurations. GreenCloud presents a limited graphic user interface and provides low performance in simulation by limiting the size of the simulated clouds configurations. **MDCSim** [13] 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, thus forcing to cloud configuration and related workload programming; therefore it shows limited capabilities in both modeling and simulating complex business configurations which change over time. **iCanCloud** [18] 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 its related math model that could be used for any estimation of power consumption, temperature, costs, etc.) and can be executed on parallel instances. iCanCloud has 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. For such reasons, it is very complex to make a full comparison of the different clouds, since the consumed memory and speed in simulation strongly depend on the resource and the adopted mathematical models [18].

A comparative overview is shown in Table 1 where 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. **Simulation aim and focus,** the main purpose of simulation, namely 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 relevant. For example, as to the simple estimation of energy consumption a detailed modeling of services with their related connections would be useless. **Time and computing aspects:** simulation type granularity, event address, parallel simulation, model, etc. This latter 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

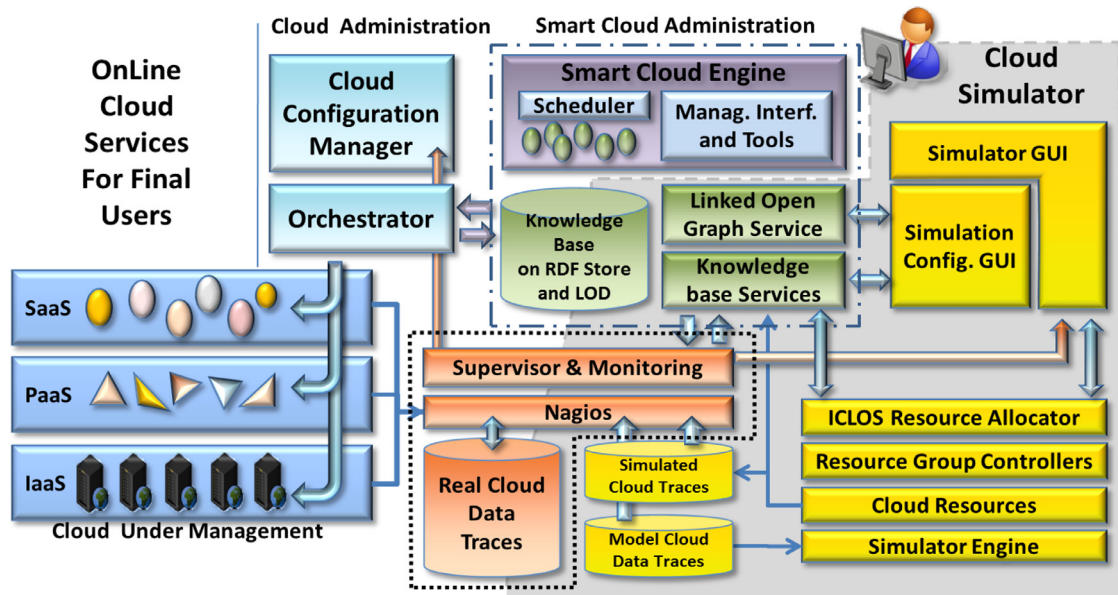


Fig. 1. ICARO cloud architecture with cloud simulator.

VM, addressing communication model, support for the TCP/IP, addressing the computer physical model. **Development:** open sources or commercial, development language, API availability, etc.

As a result, cloud simulators at present do not allow the simulation of complex business configurations, nor address Service Level Agreements, SLA, complex/real workload pattern models, with the aim of exploring, assessing and predicting the best resource allocation based on consumption of resource in the real cloud infrastructure for long time ahead. The above revised simulators hardly cope with the huge amount of data produced by simulating the behavior along several weeks, taking into account workload patterns describing the whole duration of the temporal windows. For example, the analysis of real monitored data from the services, VM and hosts in place, can be used to learn hourly, daily or weekly resource consumption patterns which can be used to produce a forward simulation and prediction.

3. ICARO Cloud Simulator architecture

The proposed ICARO Cloud Simulator is integrated in the context of the ICARO Cloud platform for Smart Cloud management [4]. In this Section, an overview of the ICARO architecture is reported to highlight the relationships of ICLOS with other components.

The ICARO Cloud architecture is reported in Fig. 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 as those mentioned above, and Orchestrators (e.g., VCO of VMware, Microsoft cloud solution). In pure simulation cases this part can be missing.
- (3) **Supervisor and Monitor (SM)** collects monitoring data from real 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 users, 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 structures, business configurations, SLA, resources, and corresponding actual values coming from the SM. 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. The use of a KB makes easier interoperability among public and private clouds, and/or among different cloud segments managed by different cloud orchestrators or CCM;
- (5) **Smart Cloud Engine (SCE)** exploits the Knowledge Base (KB) where the cloud under control, and the simulated clouds are modeled. The SCE allows the execution of making decision processes on a distributed and parallel architecture to assess cloud healthiness and reconfiguration strategies [4].
- (6) **Cloud Simulator (ICLOS)**, simulates cloud conditions hourly, daily, weekly and yearly, taking into account real resource consumption and workload patterns and exploiting complex business configuration modeled into the KB.

Areas (5), (4) and (3) of the architecture are summarized in the following subsections, while the ICLOS (6) is described in [Section 4](#) with further details.

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 cloud services [4]. The SCE 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 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 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**, when true it 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** corresponding 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 taking decisions in real time, the SCE provides a distributed scheduler engine with cluster functionality allowing adding new scheduling nodes and defining jobs, for smart cloud management, without service downtime. The SCE can take decision about the cloud reconfiguration, address aspects of energy consumption, capacity planning, etc., with the aim of maintaining a high quality 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. 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 general cloud configuration and the status of the cloud under control (as well as of simulated cloud configurations or mixt simulated over a basis of real conditions). The collected model includes services ranging from the data center infrastructure to SW application structure, as well as the applicative metrics definitions and values, referred to BC and SLA. A review on knowledge base usage in the context of cloud can be recovered in [5]. The variety of modeled resources in ICARO KB is higher if compared with the models adopted in the above mentioned simulators. In ICLOS, the KB is adopted for modeling configuration, taking decision by comparing possible configurations with actions, and it is the very basis of any simulation. Therefore, the KB models: datacenters, hosts (i.e., HW servers), VMs, networks, net devices, SLAs, metrics, users, SW applications, operating systems, etc., and their generic and specific relationships such as: is-a (specialization), needs, hasPart, isPartOf, has SLA, hasNetworkAdapter, isTenantOf, etc. (see [Fig. 2](#)). The model also allows to represent clusters as specific Business Configurations, racks as specific data centers elements, etc.

The usage of a KB enables the reasoning on cloud structures and resources by inference on the basis of the specialization (is-a), aggregation, equivalence, subclassing, etc., relationships. On this basis, the KB helps to implement SCE strategies and simulations [4], 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 belonging to components and their temporal trends (collected by the monitoring tools with the needed sub-sampling) to be able to answer questions such as “Which host machines can allocate a new VM?” or “Has the host machine H7897 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. The KB is feed by the SM with data regarding the monitored cloud resources (such action is based on Nagios monitoring tool [17]). The KB may refer to multiple Nagios installations collecting data from different data centers or segments of data centers.

KB stores both the application as a type and the application instances, and the latter can have specific constraints like the number of involved services (e.g., number of front-end web servers). Therefore, to avoid duplicating type/ instance relation (modeled in RDF) and to leverage on the modeling features available in OWL2 to express constraints (e.g., max/min cardinality) it has been decided to represent the application model as an OWL Class [10].

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 BCs. The KB has to contain the SLAs associated with application or application tenants or with a whole business configuration. The SLA has been modeled as a set of Boolean expressions relating high level metrics values belonging to a component with a reference value.

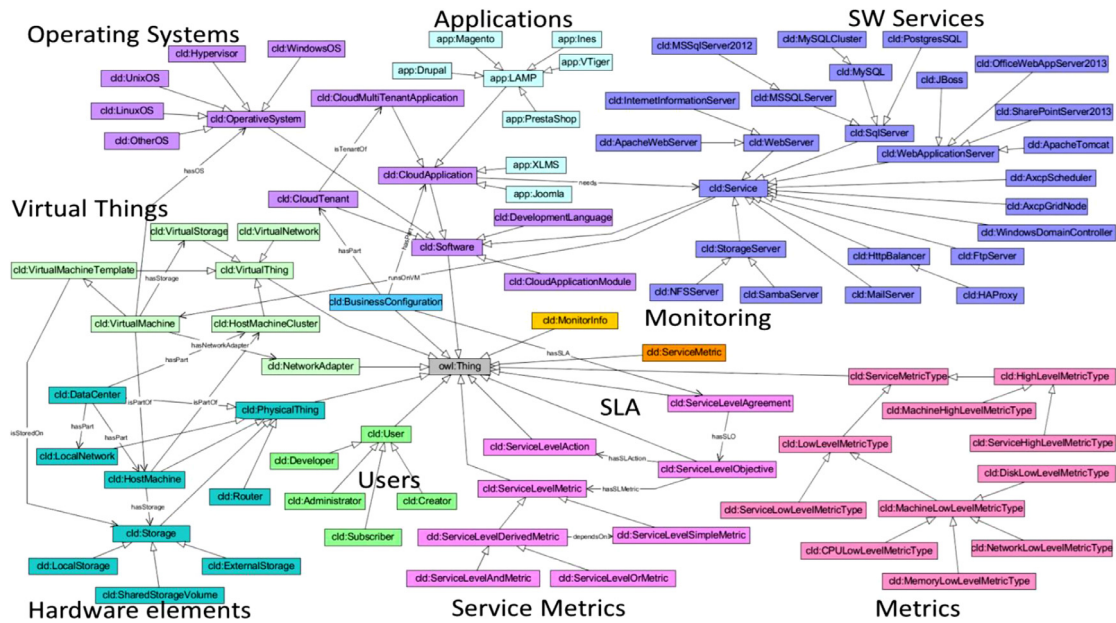


Fig. 2. Main entities and relationships in the ICARO Cloud Ontology of KB.

KB Services are provided as REST APIs for accessing, configuring, modeling, loading SLA, and manipulating any cloud element and metric value on an RDF Store (currently an OWLIM-SE instance). When a complex datum (e.g., a complex multitier business configuration) is provided to be stored on the KB via a RDF-XML description, it is first validated against completeness and consistency and then stored into the KB modeling the cloud. The KB provides a SPARQL endpoint allowing posing semantic queries for (i) the SCE assessment of the cloud element healthiness, (ii) the SCE decision criteria, (iii) the verification and validation of consistency and completeness of BC/SLA, (iv) any loading, storing and accessing the simulations, 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 to enable any services and resources control (see Fig. 3).

Moreover, in order to make the formalization of semantic queries easier, a suitable graphical user interface based on Linked Open Graph has been used to access the KB and browse the semantic model [6]. 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, stores 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 such 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, too. The approach of delegating the monitoring processes configuration to the KB (see Fig. 3) makes the work of the Orchestrator simpler, since each monitoring issue does not need to be programmed into the deploy workflow, thus reducing the error prone process, the distribution of passwords, etc. Besides, it allows to be sure that the SCE automatically adds all the monitoring processes allowing the SCE to have all the needed information to control the active BCs and SLAs. 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 customers), as well as to the user interface of the ICLOS.

4. ICARO Cloud Simulator (ICLOS)

Before discussing the structure of ICLOS, a short overview of the main identified requirements is needed to cope with the afore mentioned problems related to the Smart Cloud management for complex BC including SLA.

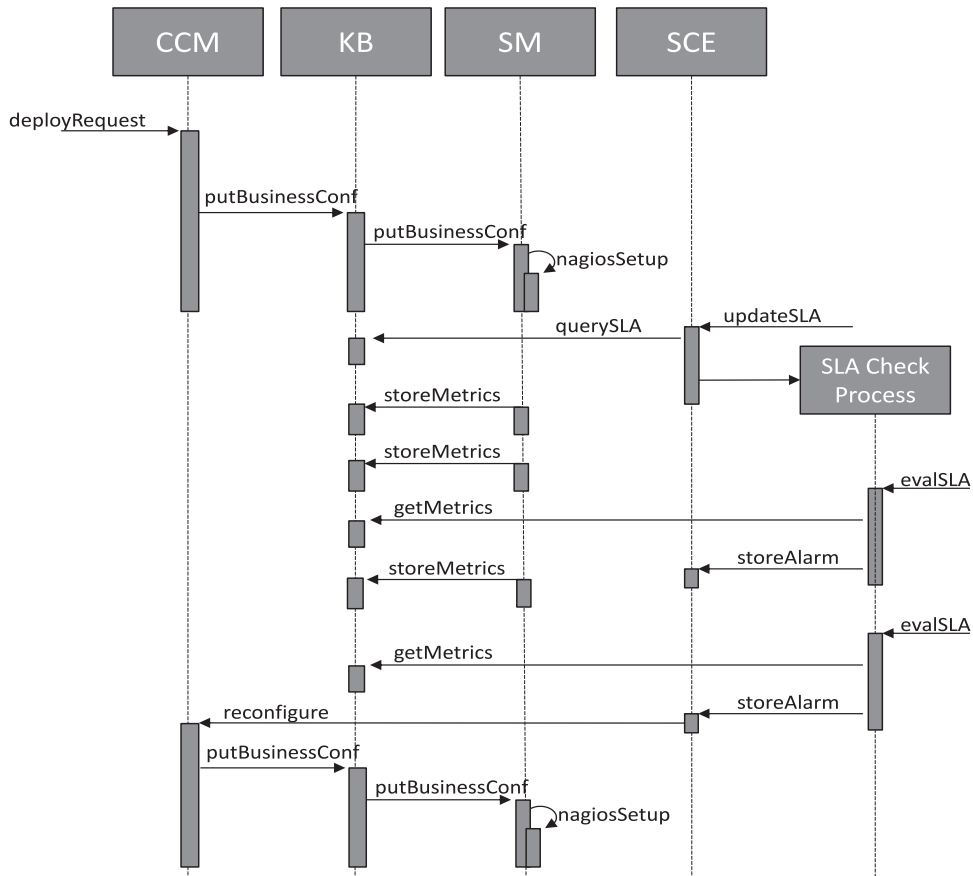


Fig. 3. ICARO supervisor and monitor.

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 architecture including several VMs, services, networks, services, processes;
- *Real resource consumption patterns* that may provide non-periodic 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 possible unexpected events, like the arrival of a storm, etc. The trends about resource consumption for CPU, memory, storage, network, etc. are related one another, and thus the real BC profiling has to be considered in terms of related patterns;
- *Simulation for longer time windows by using workload partners describing days, weeks, months*. Longer periods can be produce by replicating, while the modeling of long duration workload pattern strongly increase the simulation complexity;
- *Simulation of multiple objectives, for example, the energy consumption on viable cloud allocations*;
- *Articulated SLA* to avoid violation of SLA and to control major cost parameters, taking decision, informing the customer and administrators, etc., mainly connected to the Smart Cloud, SCE, features;
- *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., also connected to the Smart Cloud, SCE, features;

For the most part such aspects are not addressed in a satisfactory manner by the simulators at the state of the art, see [Section 2](#).

4.2. Architecture of ICLOS Cloud Simulator

[Fig. 1](#) has presented the general architecture of the ICLOS. As depicted in [Fig. 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

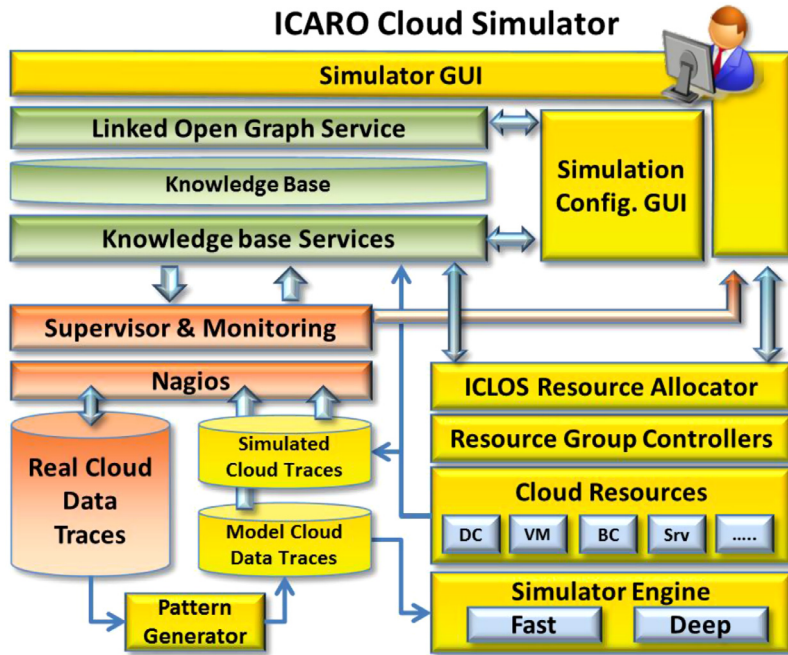


Fig. 4. ICLOS architecture.

general cloud management level. 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 to configure 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 produces 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 resource workload, always considering the related CPU, memory, network, storage, etc. along days, hours, week, months, etc., of different VM, hosts and services of a BC (see next subsection for a description). **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 the management of the allocated resources addressing events and harmonizing the math models for computation. **Cloud Resources**: a collection of allocated resources according to the produced configuration. It may take into account multiple and incremental configurations. The resources that can be allocated in the simulator are in principle the ones being modeled by the KB (see Fig. 2), while in reality only some of them are allocated and deployed as described hereafter. **Simulator Engine**: the simulation model can progress in estimating the output workload synchronously among all resources, time instant by time instant (deep mode), or it can compute 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 the same requested by the simulator during the configuration and coherently defined by the SLA for each BC. The results of the simulation is again generated in the RRD format, thus allowing the visualization of results on SM and any further reuse in more complex simulations.

The ICLOS simulator has been designed to model into the simulation the main KB classes and structures. In Fig. 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 Fig. 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 to feed the KB. The main goal of the simulator is to simulate the workload and cloud model in general and save them along days, weeks, months, etc. in the SM and KM. This allows to: (i) model and simulate larger cloud and more complex configurations, (ii) activate the SCE rules for further analysis.

4.3. Cloud workload, Pattern Generator

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 [9]. The analysis of the data collected by the Performance Monitor may be used to perform a workload classification [16, 3]. Such workload patterns are exploited in the

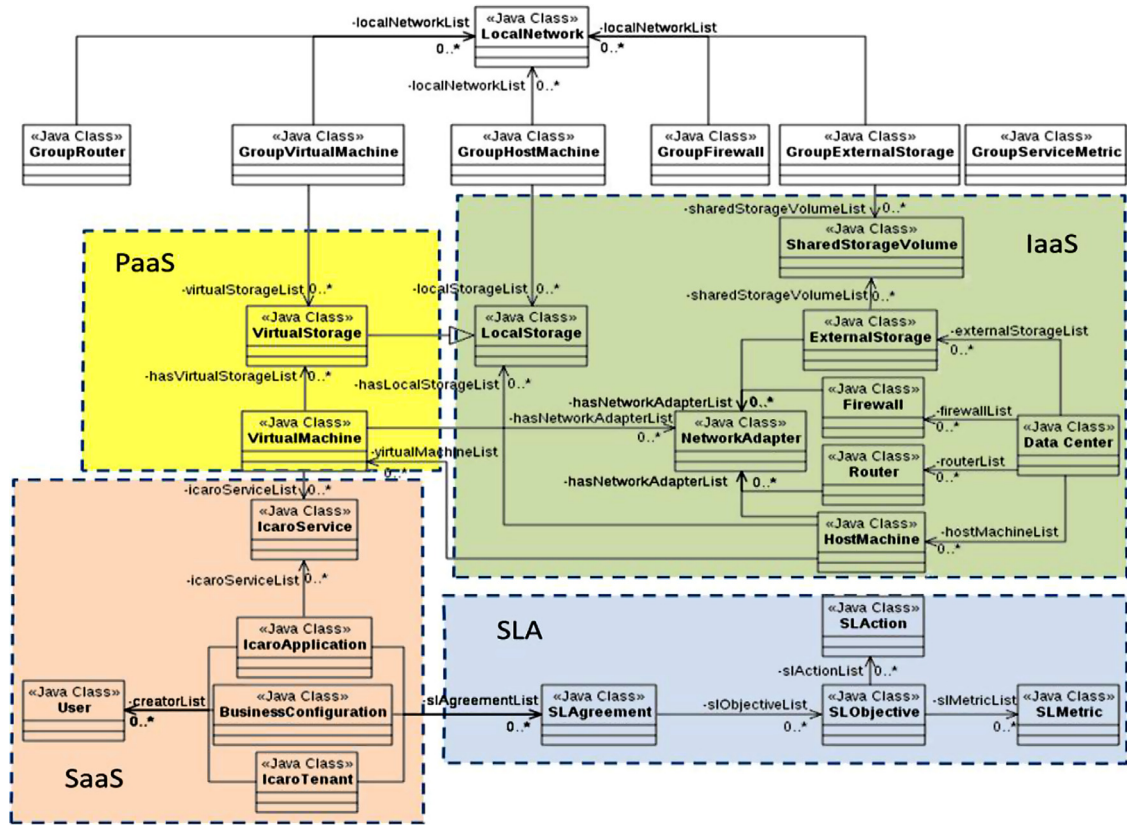


Fig. 5. ICLOS modeling main classes.

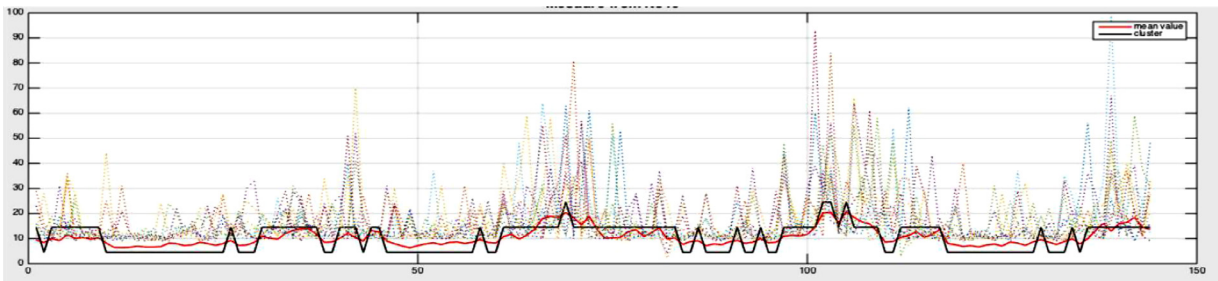


Fig. 6. Example of normalized workload pattern generated by clustering real trends along an hour of work in the context of the smart city data ingestion VM.

cloud simulation in the ICLOS solution. In reality, the mere statistical characterization of VM or hosts on the basis of CPU and Memory workload is not enough to cope with complex BCs like the ones 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 [8]. The **Pattern Generator** perform a clustering analysis to identify the most probable workload patterns from real resource consumption trend of classical cloud resources and/or high level metrics such as: CPU, memory, network, storage, user activity, disk usage, etc. These trends can be computed per hour, day, week, month, etc., from real business configurations including hosts, VM and services of a BC. The Pattern Generator tools exploit Real Cloud Data Traces in RRD format collected from Nagios/SM on the real cloud to perform cluster analysis and produce the most likely family of patterns for a given BC to be used into the ICLOS simulation phases (see example in Fig. 6). The family of patters of each single BCs are coherently selected (associating coherent values among resources, avoiding of making simulations with CPU workload unrealistic with respect to the memory usage or HD access). Moreover, they are randomly selected among the most probable patterns to create the simulation workload. In addition, the same pattern is normalized and used to create different kind of workloads, for example with 10%, 30%, 60%, 90% of load, and/or adding some random changes of limited value.

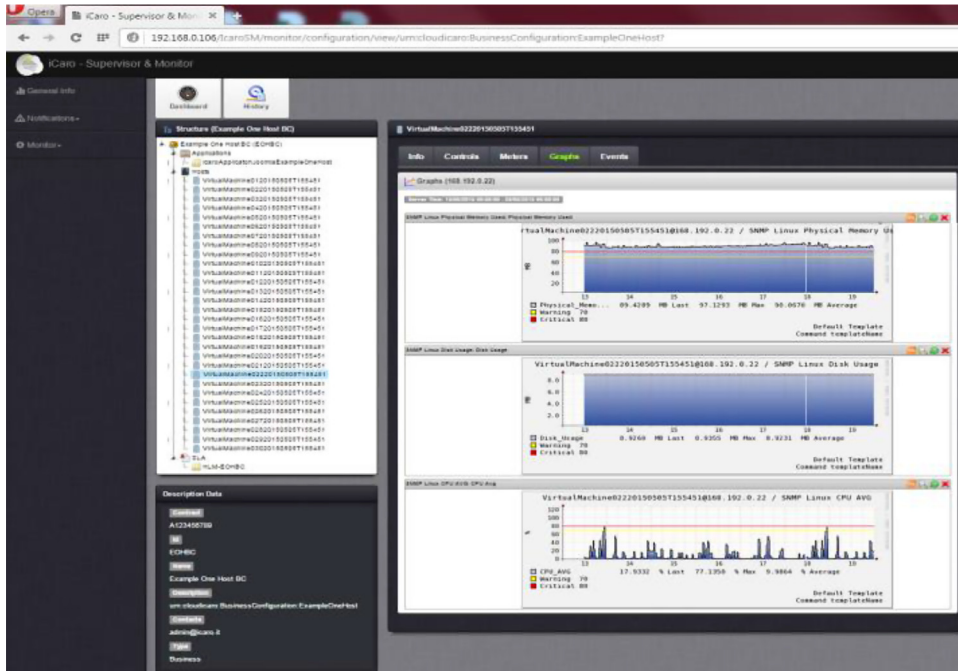


Fig. 7. ICLOS simulator results on SM.

5. Experimental results

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

5.1. Experimental results in simulating

In [26], the simulation of 1000 Hosts with 4 VM each, for a total of 40,000 *In this paper*, ICARO cloud simulator the energy consumption model only, was performed in 3597 s on an Intel Core i7 930 processor and 6GB of RAM. The estimation has been assessed by performing 5 repetitions and the simulations were done along 10 days, with a single data value every 10 min. The power consumption model has been modeled by using SPECpower benchmark [24]. For comparison purposes, a similar simulation has been performed with ICLOS. Thus, taking 1000 Hosts with 4 VM each, a total of 40,000 VMs has been simulated by computing the energy consumption model SPECpower benchmark [24] and using input values every 5 min, while generating output simulated values every 5 min. The simulation has been performed 5 times on Debian 64 bit, 6 Gbyte of memory, CPU 4 core, 2000Mhz, obtaining average time of 1985 s and a Std.Dev.=245,89. As a result, the ICLOS and DC simulators are comparable in terms of execution time.

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 more time in saving the output data resulting from the simulation of all the VM and Hosts on the hard-disk, with a sample every 5 min. Fig. 7 reports the ICLOS simulation directly monitored into the SM tool which exploited Nagios libraries to access and render the RRD storages.

Moreover, Table 2 reports details of a number of simulations / configurations 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 GB, reservation memory space of 1 GB; Hosts (cases 1 and 2) ranging from 1 to 10 (each of them with: 32 cores, 2500 MHz per core and 128 GB 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.

The ICLOS simulations have been performed by using workload patterns of 1 week forward for resources (CPU, storage and memory) from the RRD of the SM with a measure every 5 min, thus simulating a whole week for the VM and hosts. Therefore, the input workload patterns have a value every 5 min 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/Hosts has been performed in about 80 min 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

Table 2
ICLOS Simulations for power consumption assessment.

| 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 3

ICLOS simulations for allocation by using different algorithms. The execution time refers to 20 executions of the allocation algorithm in simulation. The “host number” refers to the most probable number of hosts identified among the set of 20 simulations; in most cases, this number is the minimum number of hosts according to the goals of the adopted bin packing algorithm.

| Algorithms | 494 | | 998 | | 2000 | | 4000 | |
|--------------------|---------------------------|-------------|---------------------------|-------------|---------------------------|-------------|---------------------------|-------------|
| | Execution time in seconds | Host number | Execution time in seconds | Host number | Execution time in seconds | Host number | Execution time in seconds | Host number |
| <i>Dot Product</i> | 49,36 | 37 | 169,89 | 75 | 708,64 | 150 | 3169,60 | 300 |
| <i>L2 Norm</i> | 43,62 | 38 | 163,88 | 77 | 739,93 | 153 | 3457,49 | 305 |
| <i>FFD Sum</i> | 2,95 | 37 | 14,84 | 75 | 59,09 | 150 | 266,38 | 300 |
| <i>FFD Prod</i> | 5,77 | 38 | 14,06 | 75 | 47,96 | 150 | 231,36 | 300 |

numbers from simulations as reported in Table 2 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 GB 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 any saving of the resulting data on SM in RDD format in a remote HD. The “averaged time / #VM” grows 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 Fig. 7). The computational time to simulate the 10 Hosts with 30VMs 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 includes for each VM the access on HD to take the pattern, the XML parsing, the computation of simulation and the writing of the RRD/XML with the simulation. Therefore, 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 quite the same along a week. Provided that the simulation time is quite constant, it is almost useless to perform simulations with higher number of VM and Hosts, with a needed storage of about 1.2Mbyte 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, keeping its values under 120Mbyte in all cases.

5.2. Experimental results

In this section simulation results for VMs allocation are reported.

The first case is reported in Table 3, where different bin-packing algorithms have been used in order to identify the most probable number of hosts needed to allocate a number of VM (from 500 to 4000 including BCs). The bin-packing algorithms try to compose the VM while respecting the possible configurations and composing the resources patterns, so as to always keep in mind the limits of the host capacity. The algorithms selected have been already adopted for cloud resource allocation [22], and in particular the FDD (First Fit Decreasing) by sum and by product weight, the Dot product, and the L2 Norm. When the patterns are complex, the bin packing goal is to find the compromise from the most probable number of minimum host for allocating a number of VMs belonging to a set of different BCs. The simulations have been performed using generated patterns from real cases and simulating one working day. As to the obtained data, it can be remarked that the FFD Prod algorithm provides good results with shorter execution time in almost all cases. The execution times have been estimated on a 24 CPU core host at 3.0 GHz with 64 GB of RAM on 20 simulations. On the other hand, in most cases, 10 simulations could be enough to estimate the configurations to obtain the most probable number of needed hosts, as

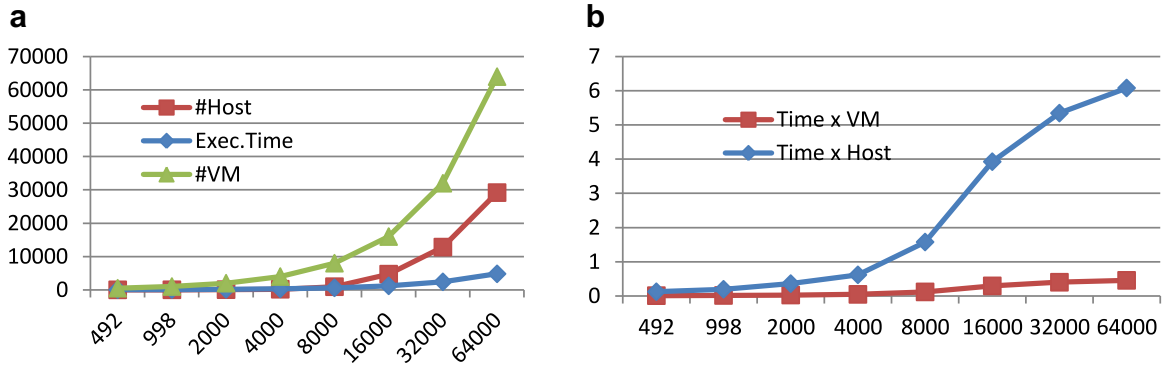


Fig. 8. Simulations for allocating VMs by using FFD Prod algorithm from 500 to 64000 VMs: (a) trend of execution time and number of VMs and Hosts (20 simulations), (b) estimation of the executing time cost with respect to the number of VMs and Hosts.

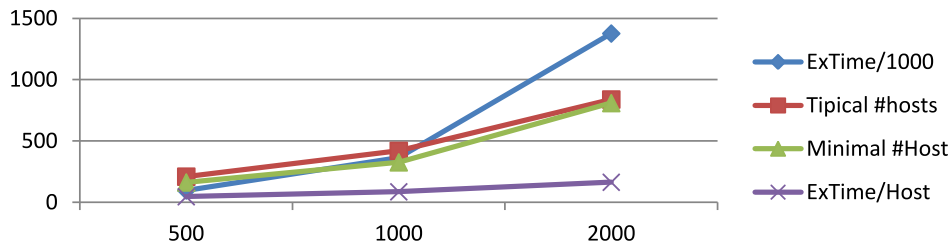


Fig. 9. Simulations for allocating VM by using FFD Prod algorithm from 500 to 2000 VMs, workload patterns for 1 month according to the described mix of BC, 10 simulations for each estimation.

reported in Table 3. The patterns were referred to a distribution of 4 different BCs: a 4 tier architecture for data warehouse (5 VM as a balancer, 2 web server, one database and 6 computational nodes); a three tier solution of small social network, a simple two tier solution for a web server application, and a single tier solution with a web application. The simulations have been addressed starting from their real workload patterns and producing clusters simulation patterns for a 1 whole day of 24 h.

A second simulation experiment shows the execution time for VM allocation from 500 to 64000 VMs. The simulations have been addressed with the above described BC workload patterns for a day. Fig. 8a reports the trends of the execution time in seconds, with respect to the number of VMs and the identified most probable number of hosts. Fig. 8b describes the execution time in seconds for simulating a VM and a Host respectively. In both cases, simulations with more than 32000 VMs tend to stabilize the execution time per VM and per Host.

In Fig. 9, the trends of simulation execution time are reported for the case of workload of 1 month, for the same complex BC described in the first case of this section. The simulations have been addressed with complex workload patterns of 1 month according to the mix of BCs described above. A huge complexity is added when long time durations are taken into account. In fact, the case of 2000 VMs for a day produced results of packing all into 150 hosts, while in this case of 1 month, the packing leads to 839 hosts, thus taking into account critical longer period behaviors for the VMs. In terms of execution time for the 2000 VMs, for a day costs 47 s (see Table 3), while for the case of 1 month pattern the averaged execution time for 20 simulations 2753.6 s that leads to about 91.8 s per day per simulation.

6. Conclusions

In this paper, ICARO cloud simulator (ICLOS) developed has been presented. It has been specifically designed for simulating the workload on the basis of real virtual machine patterns for their resources and behavior within wide temporal windows, and in connection with the SCE, Smart Cloud Engine. 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 the supposed behavior along days, weeks, months, etc. (for example for seasonal prediction of workloads). The proposed research has been developed in the context of the 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. On such grounds, the simulation is sustainable for large data centers obtaining predictions and computing sustainable allocations dimensioning the number of hosts needed, assessing the power consumption, etc., and it can be scaled up by using multiple servers. The ICLOS simulator has been also used to simulate power consumption and it obtained simulation time comparable with other simulators and yet it could grant more complete functionalities.

Acknowledgments

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

References

- [1] M. Aggarwal, Introduction of cloud computing and survey of simulation software for cloud, *Res. J. Sci. IT Manag.* 2 (2013).
- [2] A., Ahmed, A.S. Sabyasachi, Cloud computing simulators: a detailed survey and future direction, in: *Proceedings of IEEE International Advance Computing Conference (IACC)* 21–22 Feb., 2014, pp. 866–872, doi:10.1109/IAdCC.2014.6779436.
- [3] M. Amoretti, F. Zanichelli, G. Conte, Efficient autonomic cloud computing using online discrete event simulation, *J. Parallel Distrib. Comput.* 73 (2013) 767–776.
- [4] Pierfrancesco Bellini, Daniele Cenni, Paolo Nesi, A knowledge base driven solution for smart cloud management, *IEEE Cloud (2015) NewYork*, July 2015.
- [5] P. Bellini, D. Cenni, P. Nesi, *Cloud Knowledge Modeling and Management*, Chapter on *Encyclopaedia on Cloud Computing*, Wiley Press, 2015.
- [6] P. Bellini, P. Nesi, A. Venturi, Linked open graph: browsing multiple SPARQL entry points to build your own LOD views, *Int. J. Vis. Lang. Comput., Elsevier*, 2014.
- [7] R. Buyya, R. Ranjan, R.N. Calheiros, Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities, in: *Proceedings of 7th Int. Conf. on High Performance Computing and Simulation*, June 2009, pp. 1–11.
- [8] C. Germain-Renaud, O.F. Rana, The convergence of clouds, grids, and autonomies, *IEEE Internet Comput.* 6 (13) (2009).
- [9] J.L. Hellerstein, Google Cluster Data. Posted at: <http://googleresearch.blogspot.com/2010/01/google-cluster-data.html> (accessed 10.02.16).
- [10] ICARO Cloud KB, Ontology. D2.9.2, <http://www.disit.org/5604> (accessed 10.02.16).
- [11] D. Kliazovich, P. Bouvry, S.U. Khan, GreenCloud: a packet level simulator of energy-aware cloud computing data centers, *J. Supercomput.* 62 (2012) 1263–1283.
- [12] S. Kumar Garg, R. Buyya, Melbourne, NetworkCloudSim: modelling parallel applications in cloud simulations, in: *Proceedings of the 4th IEEE/ACM International Conference on Utility and Cloud Computing*, 2011.
- [13] Seung-Hwan Lim, Bikash Sharma, Gunwoo Nam, EunKyoung Kim, ChitaR. Das, MDCSim: a multi-tier data center simulation, platform, in: *Proceedings of IEEE International Conference on Cluster Computing and Workshops*, 2009.
- [14] R. Malhotra, P. Jain, Study and Comparison of CloudSim Simulators in the Cloud Computing, *Trans. Comput. Sci. Eng. Appl. (CSEA)*, 1 (2013) 111–115.
- [15] S. McCanne and S. Floyd, “Network Simulator ns-2,” <http://www.isi.edu/nsnam/ns/>, 1997 (accessed 10.02.16).
- [16] A.K. Mishra, J.L. Hellerstein, W. Cirne, C.R. Das, Towards characterizing cloud backend workloads: insights from google compute clusters, *ACM SIGMETRICS Perform. Eval. Rev.* 4 (37) (2010) 34–41.
- [17] <http://www.nagios.org> (accessed 10.02.16).
- [18] A. Núñez, J.L. Vázquez-Poletti, A.C. Caminero, G.G. Castañé, J. Carretero, I.M. Llorente, iCanCloud: a flexible and scalable cloud infrastructure simulator, *Springer, J. Grid Comput.*, 10, 2012, pp. 185–209.
- [19] <http://opennebula.org/> (accessed 10.02.16).
- [20] <https://www.openstack.org/> (accessed 10.02.16).
- [21] Ranu Pandey, Sandeep Gonnade, Comparative study of simulation tools in cloud computing environment, *Int. J. Sci. Eng. Res.* 5 (5) (May-2014).
- [22] Rina Panigrahy, Kunal Talwar, Lincoln Uyeda, and Udi Wieder, “Heuristics for Vector Bin Packing”, <http://research.microsoft.com/apps/pubs/default.aspx?id=147927> (accessed 10.02.16).
- [23] G. Sakellari, G. Loukas, A Survey of mathematical model, simulation approaches and testbeds used for research in cloud computing, *Simul. Model. Pract. Theory* (2013).
- [24] Standard Performance Evaluation Corporation <http://www.spec.org/>, Aug 2012 (accessed 10.02.16).
- [25] W. Tian, Xu M, Chen A, Li G, Wang X, Chen Yu, Open-source simulators for Cloud computing: Comparative study and challenging issues, *Simulation Modelling Practice and Theory* 58 (Part 2) (2015) 239–254.
- [26] M. Tighe, G. Keller, M. Bauer, H. Lutfiyya, DCSim: A data centre simulation tool for evaluating dynamic virtualized resource management, in: *Proceedings of 2012 8th International Conference And 2012 Workshop On Systems Virtualization Management (Svm), Network And Service Management (CNSM)*, 22–26 Oct., 2012, p. 385,392.
- [27] B. Wickremasinghe, R.N. Calheiros, R. Buyya, CloudAnalyst: A CloudSim-based visual modeller for analysing cloud computing environments and applications, in: *Proceedings of 24th International Conference on Advanced Information Networking and Application (AINA)*, IEEE, 2010, pp. 446–452.
- [28] Qi Zhang, Lu Cheng, Raouf Boutaba, Cloud computing: state-of-the-art and research challenges, *J. Int. Serv. Appl.* (2010) 7–18.
- [29] Wenwu Zhu, Chong Luo, Jianfeng Wang, Shipeng Li, Multimedia Cloud Computing, *Signal Processing Magazine, IEEE*, vol.28, no.3, pp.59–69.