CloudSim Plus

A Modern Java 8 Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services

Manoel C. Silva Filho[1,2]; Raysa L. Oliveira[2]; Claudio C. Monteiro[1]; Pedro R. M. Inácio[2]; Mário M. Freire[2]

[1]*Departamento de Informática - Instituto Federal de Educação do Tocantins (IFTO).* [2]*Instituto de Telecomunicações (IT) and Departamento de Informática, Universidade da Beira Interior (UBI).*

Introduction

The Cloud Computing paradigm has been driving innovation in different areas of knowledge and economic sectors by providing distributed, scalable, manageable and fault-tolerant computing resources over the Internet. The large adoption of Cloud computing services can be also explained by diverse reasons such as: its pay-per-use charging model, which enables cost reduction for customers; the rapid and automated allocation of resources, which enable applications to quickly respond to bursts; and full-featured web and console interfaces, which enable customers to configure hosted services easily.

Considering the complexity of a cloud infrastructure, computer-based simulation constitutes a rather attractive tool to carry out research in this field. Additional reasons to use computer-based simulation include: the need to model proposed solutions and evaluate them in a quick, cheap and repeatable way; the use of a controlled environment, which makes it easy to monitor and collect metrics; and the easiness to setup the required environment for experimentation.

In this article we present CloudSim Plus: a new, full-featured, re-designed, highly extensible and modern Java 8 framework for modeling and simulation of cloud computing infrastructure, services, underlying mechanisms and algorithms. CloudSim Plus enables researchers to model and simulate different cloud scenarios, by implementing them using Java. Such scenarios can be used to experiment existing and potentially new solutions for the issues mentioned above.

CloudSim Plus is an open source project available at http://cloudsimplus.org and Maven Central.

Overview

CloudSim Plus is based on CloudSim 3. It went through an extensive re-design and re-engineering process to provide an updated, modern, highly extensible and easier-to-use cloud simulation framework. These changes aim to enable sustainable project maintainability for long-term evolution. To achieve such goals, CloudSim Plus is founded on several software design and engineering metrics, principles and practices such as Coupling, Cohesion, Design Patterns, SOLID principles and other ones like Don't Repeat Yourself (DRY) and KISS.

Architecture

CloudSim Plus is a Java project that has both simpler module and package structures. The entire

project is compounded of 4 modules that were re-organized to directly inherit from the parent project, allowing a researcher to quickly have an overview of the structure. Redundant and out-of-date modules such as "distribution" and "documentation" were removed since building distribution artifacts and documentation is already automated using Maven.

Figure 1 presents the current project architecture and its modules are described as follows. The highlighted modules are new in CloudSim Plus.
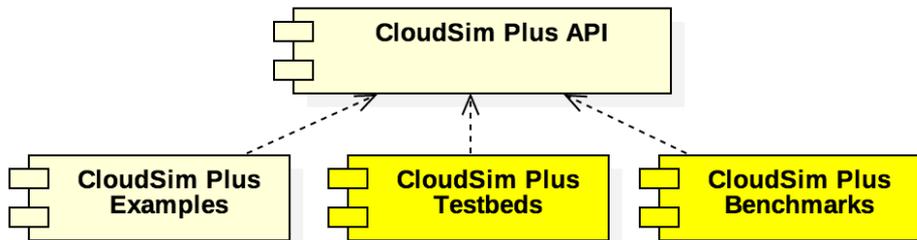

Figure 1. CloudSim Plus Modules

Modules

CloudSim Plus API is the main module that contains the framework API. It is the only independent module that is required to build simulation scenarios. Since such a module is available at Maven Central, there is no need to manually download the framework source code to build simulations. This module can just be added as a maven dependency to one's own project and he or she will be ready to start building simulation scenarios.

CloudSim Plus Examples provides the original CloudSim examples, with refactored, well organized and updated code to use the CloudSim Plus API. It also includes new examples for CloudSim Plus exclusive features.

CloudSim Plus Testbeds implements some simulation testbeds in a repeatable manner. It provides base classes that allow a researcher to collect valid scientific results, such as average values and standard deviations, considering a specific confidence interval. They serve as examples on how to create broader testbed experiments.

CloudSim Plus Benchmarks is used just internally to measure the overhead of some CloudSim Plus features.

Exclusive Characteristics and Features

CloudSim Plus is a full-featured simulation framework that has introduced exclusive features, as presented below.

Dynamic Creation of Vms and Applications (Cloudlets)

CloudSim Plus allows on-demand creation of `Vms` and `Cloudlets`, without requiring creation of `DatacenterBrokers` at runtime. The `DatacenterBroker` class was refactored to enable submission of new `Vms` and `Cloudlets` during simulation execution, accordingly requesting the creation of such objects into the cloud infrastructure. It also enables delaying the creation of submitted `Cloudlets`

and Vms, which may be used to control the time when the researcher wants these objects to be created. For instance, Vm creation can be delayed to avoid upfront allocation of resource that may not be required immediately.

Vm Scaling

Vm migration is a well-known mechanism to optimize allocation of physical resources. However, it is an expensive operation that causes service downtime, introduces overhead and must be performed carefully. Sometimes Vm migrations can be avoided by simply scaling under or overloaded Vms. Appropriately, CloudSim Plus provides vertical and horizontal Vm scaling mechanisms.

These two different kinds of Vm scaling, additionally with Vm migration algorithms, can be used selectively by an Hypervisor to provide a very efficient Vm allocation policy mechanism to achieve intended goals. This mechanism can decide the time to perform Vm migration, vertical or horizontal Vm scale. Depending on specific conditions, one action can be favored over other ones or even different actions can be performed at a given time. CloudSim Plus Vm Scaling mechanisms are discussed below.

Vertical Vm Scaling

Vertical Vm Scaling performs on-demand down or up allocation of Vm resources such as RAM, Bandwidth and CPUs, according to under or overloaded Vm condition, respectively. Since actual hypervisors such as KVM and VMware ESX allows dynamically changing allocation of RAM and resources for a Vm, that can be used to avoid VM migration in specific conditions.

Horizontal Vm Scaling

Horizontal Vm Scaling allows dynamic destruction or creation of Vms, according to an under or overload condition, respectively. Such conditions are defined by a predicate that can check different Vm resources usage such as CPU, RAM or Bandwidth, to define if a Vm is under or overloaded.

This feature allows a researcher to implement and evaluate load balancing algorithms for dynamic workloads and burst conditions, by enabling the creation of new Vms to attend the demand. Some cloud platforms such as Amazon Web Services provide an Auto Scaling feature, that can be alike simulated in CloudSim Plus.

Parallel Execution of Simulations

CloudSim Plus was re-designed to enable running multiple experiments in parallel, in a multi-core machine, to reduce simulation time. The real time reduction that can be achieved by running simulations in parallel is tightly dependent of the simulation scenario and its scale. If the simulation is CPU-bound and is comprised of several runs, then the parallelization might provide large time reduction. On the other hand, small scale simulations or I/O-bound ones are not expected to take advantage of this feature.

An example available here shows how it is simple to parallelize simulation experiments in

CloudSim Plus, using the Java 8 Stream API.

Event Listeners

One of the features a cloud infrastructure must provide is the ability to monitor running services. Monitoring capabilities can be used in different ways by involved parties. The cloud provider can, for instance: collect resource utilization to charge customers in a pay-per-use basis; assess fulfillment of customer SLA; or optimize resource allocation to avoid under and over resource provisioning. Customers can, for instance, assess if the kind of resources he/she has contracted is appropriated to his/her demand and then take the required actions if they are not.

CloudSim Plus provides Listeners as a mechanism to monitor simulation in runtime, allowing collection of metrics, resource allocation decision making (such as Vm scaling) and granular simulation execution feedback. Since the final goal of a simulation is the collection of data to be processed, assessed and validated, Listeners enable researchers to collect such data at any time interval they need.

Strongly Object-oriented Framework

CloudSim Plus was comprehensively re-engineered to create relationships among classes, enabling chained calls such as `cloudlet.getVm().getHost().getDatacenter()`. This way, it stores references to actual objects, instead of just integer IDs to represent these relationships, which does not conform to an object-oriented design.

The line of code shown above provides a direct way to know what Virtual Machine (`Vm`) an application (`Cloudlet`) is running or will run, what `Host` such a `Vm` is or was placed into, and finally what `Datacenter` such a `Host` is settled down. The Null Object Design Pattern was also implemented to avoid the so propagated `NullPointerException` when making such a chained call.

Classes and Interfaces Allowing Implementation of Heuristics

Considering the large scale of cloud infrastructures, finding an optimal solution for issues such as Vm Placement is impracticable, since this is a NP-hard problem. Alternatively, heuristic techniques can be used to find a sub-optimal and satisfactory solution in a reasonable time. Some well-know heuristic methods include Tabu Search, Simulated Annealing and Ant Colony Systems.

CloudSim Plus provides a set of classes and interfaces to enable a researcher to build such heuristics for solving problems like Vm placement and migration. The interfaces provide a contract, by defining method signatures to: implement a solution generation and solution cost function (the fitness function is just the inverse of the cost); implement a function to update the solution search state; specify the number of maximum iterations, the probability for accepting each random solution and the predicate that defines when the solution finding must stop. The package org.cloudsimplus.heuristics contains such classes and interfaces and also includes a Simulated annealing heuristic to perform the map between `Cloudlets` and `Vms`.

Implementation of the Linux Completely Fair Scheduler

`CloudletScheduler` is an interface implemented by classes that provide scheduling algorithms for `Cloudlets` execution inside a `Vm`. One of the criticisms against simulation experiments is differences between some behaviors of the actual system being simulated and the simulation itself, which may reduce the simulation accuracy. Process scheduling is one of the behaviors that was neglected in cloud computing simulations up to now. The scheduling algorithm impacts some application metrics such as wait time and task completion time. A bad scheduling may lead to processes waiting for long time periods to use the CPU or, when a process is assigned to a CPU, it is not given enough CPU time. That situation is called starvation and may cause SLA violations.

The Completely Fair Scheduler used in recent version of the Linux Kernel provides a very efficient policy to avoid the mentioned issues. As an actual scheduler, it considers assigned tasks priorities to define the time slice that each process is allowed to use the CPU. It also tries to be fair when allocating these time slices to avoid starvation of low priority processes. CloudSim Plus introduces a implementation of the Completely Fair Scheduler to increase the accuracy of processes execution in simulation environments.

Additional Characteristics

Besides all the exclusive features that have been presented, CloudSim Plus has additional characteristics that make it a promising cloud simulation framework. Some of them include:

- Completely re-designed and reusable network module. Totally refactored network examples to make them clear and easy to change.

- Throughout documentation update, improvement and extension.

- Improved class hierarchy, modules and package structure that is easier to understand, following the Separation of Concerns principle (SoC). For instance, power-aware `Host` classes and interfaces are included into the intuitive org.cloudbus.cloudsim.hosts.power package, as well as network-enabled ones are included into the org.cloudbus.cloudsim.hosts.network package. And if one needs to find a power or network-enabled `Vm`, he/she will intuitively know where to find it.

- As it is usual to extend framework classes to provide some specific behaviors, a researcher will find a totally refactored code that follows clean code programming, SOLID, Design Patterns and several other software engineering principles and practices. This way it is far easier to understand the code and implement a required feature.

- Integration Tests to increase framework accuracy by testing entire simulation scenarios.

- Updated to Java 8, making extensive use of Lambda Expressions and Streams API to improve efficiency and provide a cleaner and easier-to-maintain code.

Conclusion

CloudSim Plus is an updated cloud simulation framework that relies on the most recent advances of

the Java language. It provides a more extensible, cleaner and easy-to-understand code that encourage developers to contribute. It uses industry-standard tools to:

- measure code quality;

- automate builds and the execution of unit and integration tests into a continuous integration environment;

- host its meaningful and extended documentation in a searchable way, enabling documentation versioning.

The redesign and refactoring performed in CloudSim Plus enabled reducing code duplication, making it easier to extend. The tools presented above provide an ecosystem to properly support contributions by tracking code quality and software regression. In this process, several issues were detected and fixed, improving the framework correctness.

Finally, all the new CloudSim Plus features allow researchers to implement more realistic, complex and accurate simulations. Even the scale of simulation experiments may be enlarged by running experiments in parallel. All these characteristics and features make CloudSim Plus a promising cloud simulation framework.

Acknowledgements