**CloudSim Plus Presentation**

* Sílabas tônicas estão em **negrito**.
* Letras maiúsculas no meio de uma palavra indicam que ela é pronunciada como no português.
* “Palavras-interligadas” devem ser pronunciadas juntas (como uma terminando em consoante e outra começando em vogal).
* Letras ~~riscadas~~ não devem ser pronunciadas.

**# Slide 1**

Good morning,

My name is Manoel Campos, I'm a PhD student at University of Beira Interior and a professor at a federal institute of education in Brazil.

I'm going to **prE**sent "CloudSim Plus: A cloud computing simulation **Fra**mework pursuing software engineering principles for improved modularity, extensibility and correctness".

**# Slide 2**

The following topics are going to be covered today:

* “An-introduction” to CloudSim Plus;
* It's architecture, modules and main packages;
* Main exclusive features;
* Conclusions and future work.

**# Slide 3**

* CloudSim Plus is “an-inde**pen**dent” CloudSim fork for cloud computing simulation which uses the most recent features from Java 8.
* It's a highly extensible, completely redesigned and refactored Framework, making it easier to create simulation scenarios.
* It has more than 20 exclusive features, enabling implementation of complex and more realistic simulations.
* It's heavily founded in Design Patterns, SOLID principles, Clean Code programming and other software engineering practices.
* The **Fra**mework significantly reduces code duplication by 30%, removing redundancy to provide a simplified design.

A side-by-side comparison between a simulation scenario in CloudSim and CloudSim Plus is available at this link.

The link to the presentation is provided in the end.

* Finally, it increases test coverage by 80%, while fixing **Se**veral issues, providing more **A**ccuracy and safety to perform changes.

**# Slide 4**

* CloudSim Plus is a maven project available at maven central, enabling new tools to be built on “top-of-it”, “in-an-easier” way.
* It has a simplified module structure which is easier to understand and maintain. It also introduces some new modules.
* It has a totally re-organized package structure for compliance with Separation of Concerns principle, placing only classes with the same goal into the same package.
* Finally, new interfaces were introduced to increase abstraction and define contracts for implementing classes.

Researchers can rely on these public interfaces to create their simulations and build tools on “top-of” CloudSim Plus.

**# Slide 5**

* CloudSim Plus is compounded of 4 modules. The **API** module is the main, independent and single-required one for building simulations.

All the other modules “depend-on-it**”**. The dark yellow ones are exclusive to CloudSim Plus.

* The **Examples** module was updated for removal of code duplication and better organization, including some exclusive examples.
* The **Testbeds** module provides implementation of simulations to be executed multiple times, applying different seeds for pseudo random number generators and allowing collection and a**nA**lysis of scien**ti**fic~~a~~lly valid results.
* The **Benchmarks** module is used for performance assessment of cumbersome features such as heuristics. It enables a researcher to get metrics such as number of operations per second, which may be used to guide the tuning of algo**ri**thms and *H*eu**ris**tics.

**# Slide 6**

* The new packages structure makes it easier to find a given class.

For instance, if you are looking for a Host implementation, you'll find it inside the hosts package.

* Dark yellow packages include exclusive CloudSim Plus features.
* Light yellow ones were introduced to better organize existing CloudSim classes and introduce new implementations.
* Finally, white ones are existing CloudSim packages which also received new classes and interfaces. Existing ones were updated to fix bugs, improve documentation and design and provide new features.

**# Slide 7**

* There are more than twenty exclusive features.

Due to time limitation, only the most important ones are going to be presented.

* The official website presents “an-extended**”** list.

**# Slide 8**

* One of the most interesting CloudSim Plus new features is VM scaling. “There-are” two types of scaling.
* Vertical scaling enables specific resources of a VM, such as RAM or CPU, to be scaled up or down,

according to current load and a defined static or dynamic threshold.   
This way, it allows fitting VM resources to current workload, aiming to reduce resource under and over provisioning, as well as SLA violations.

* Horizontal scaling enables creation or destruction of VM instances to balance the load, also according to defined thresholds.

Since sometimes a Host doesn't have enough resources to scale a VM up, or the vertical scaling is not enough to meet the workload, horizontal scale is one alternative for VM migration.

**# Slide 9**

* Sometimes simulations may take several minutes to run. Parallel execution enables multiple simulations to be run at the same time, in a multi-core CPU machine, which may reduce the overall simulation time.
* CloudSim Plus relies on Java 8 Parallel Streams mechanism to enable execution of simulations in parallel.
* Using this feature may be as simple as calling a single line of code, like this one.

Here we have a list of simulation instances to be executed, and considering there is a method "run" which builds the simulation scenario and runs it, such a line creates the required threads to execute each simulation instance.

**# Slide 10**

* A cloud infrastructure is a dynamic environment where requests to create VMs and run applications arrive all the time.
* To simulate this behavior, VMs and Cloudlets can be dynamic~~a~~lly created in CloudSim Plus, during simulation runtime.
* It doesn’t require new DatacenterBrokers to be instantiated.
* You just have to submit the new VMs or Cloudlets to “an-existing” broker.

**# Slide 11**

* CloudSim Plus also allows delaying the creation of VMs and Cloudlets, before starting the simulation.
* Commonly used when the arrival time of objects to be created are known in advance.
* It’s a different and easier way to simulate the dynamic arrival of such objects, however it doesn’t provide all the flexibility of the previous feature.

**# Slide 12**

* E**ven**t **Li**steners is the most general purpose feature in CloudSim Plus, which may be used in lots of different ways, such as to monitor the simulation to:
  + collect resource utilization data
  + assess fulfillment of customer SLA
  + optimize resource allocation to avoid under and over resource provisioning
  + and for granular simulation execution feedback
* For instance, “there-are" **Li**steners for E**ven**ts generated from Hosts, VMs and Cloudlets.

**# Slide 13**

* It’s a strongly object-oriented **Fra**mework in which objects are used to create actual relationships, instead of using integer IDs
* It has a fluent API, allowing direct chained calls such as this one.

This way, it is very easy to know, for instance, the Datacenter where a Cloudlets was executed.

* And don’t worry, since it uses the Null Object Pattern to avoid Null Pointer Exception.

**# Slide 14**

* CloudSim Plus introduces a set of classes and interfaces which specify a contract to implement *H*eu**ris**tics in the following steps:
  + initial solution generation
  + generation of neighbor solutions
  + definition “of-an-utility” function to be minimized or maximized
  + and then the solution finding stop criteria
* Examples of *H*eu**ris**tics include Tabu Search, Simulated Annealing and Ant Colony Systems.
* It is included a Simulated Annealing *H*eu**ris**tic for mapping Cloudlets to VMs.

**# Slide 15**

* A `CloudletScheduler` defines how a Vm schedules the execution of Cloudlets.
* Bad scheduling may cause starvation, wastage of CPU cycles and SLA violations.
* Completely Fair Scheduler reduces these issues, but needs improvements, as can be seen in this paper.
* It considers tasks priorities to define CPU time slices, which is the amount of time a process is allowed to use the CPU at a given round.
* The current CloudletSchedulerTimeShared has a simplistic implementation, ignores task's priority and doesn’t perform actual process preemption, as can be seen in this link.
* CloudletSchedulerCompletelyFair is a more realistic implementation.

**# Slide 15**

* CloudSim Plus applies functional programming to provide a functional implementation of the DatacenterBroker. This is one fundamental object that is accountable to make decisions on behalf of a cloud customer, such as the allocation of VMs and Cloudlets.
* The redesigned DatacenterBroker enables changing in runtime, the policies used to select:
  + a Datacenter to place waiting VMs;
  + a fallback Datacenter when a previous one doesn't have a suitable Host for a VM;
  + and a VM to run each Cloudlet.
* It allows implementing new policies, without requiring new to create DatacenterBroker classes.

**# Slide 16**

Finally, let-me read the conclusions.

* It is difficult to replicate a real system in simulation, mainly concerned in modelling the arrival of stochastic events such as workload bursts.
* To contribute for valid results, a simulation framework has to:
  + be well-designed;
  + get away from code duplication to avoid code degeneration;
  + be extensively tested;
  + and provide classes following software engineering principles.
* CloudSim Plus is aligned with all these requirements.
* Proposed future work is available at the issues page.