

- Sílabas tônicas estão em **negrito**.
- Uma letra sublinhada indica que ela é pronunciada como no português.
- “Palavras entre aspas” devem ser pronunciadas juntas (como uma terminando em consoante e outra começando em vogal).
- Letras ~~riseadas~~ não devem ser pronunciadas.

CloudSim Plus Presentation

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 **present** "CloudSim Plus: A cloud computing simulation **Framework** 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.

Comment [MCdSF1]: En

Slide 3

- CloudSim Plus is “an-in**dependent**” CloudSim fork for cloud computing simulations 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 twenty **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 **Framework** significantly reduces code duplication by **thirty** percent, 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 **eighty** percent, while fixing **Several** issues, providing more **Accuracy** and safety to perform changes.

Comment [MCdSF2]: tidy

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.

Comment [MCdSF3]: meintein

Slide 5

- CloudSim Plus is compounded of four 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 **excl**usive to CloudSim Plus.
- The **Examples** module was updated for removal of code duplication and better organization, including some **excl**usive examples.
- The **Testbeds** module provides a way to execute simulations multiple times, applying different seeds for **p**seudo random number generators and allowing collection and **a**nalysis of scientifically valid results.
- The **Benchmarks** module is used for performance assessment of cumbersome features such as **H**euristics.
It enables a researcher to get metrics such as number of operations per second, which may be used to guide the tuning of **a**lgorithms and **H**euristics.

Comment [MCdSF4]: s00d0

Comment [MCdSF5]: Riuristics

Comment [MCdSF6]: algoridms

Comment [MCdSF7]: Riuristics

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 **excl**usive 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 classes were updated to fix bugs, improve documentation and design and provide new features.

Slide 7

- "There are" more than **t**wenty **excl**usive features.
Due to time limitation, only the most “important ones” are going to be **p**resented.
- The official website presents “an-extended” list.

Comment [MCdSF8]: tweny

Comment [MCdSF9]: d(y)00

Slide 8

- “One of the” most interesting CloudSim Plus new features is VM scaling.
“There are” two types of scaling.
- Vertical scaling enables specific VM resources, such as RAM or CPU, to be scaled up or down, according to current load and defined static or dynamic **th**resholds.
This way, it **a**llows fitting VM resources to current workload, aiming to decrease resource under and over provisioning, as well as **S**LA violations.
- Horizontal scaling enables creation or destruction of VM instances to balance the load, also according to defined **th**resholds.
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 **a**lternative for VM **m**igration.

Comment [MCdSF10]: allows

Comment [MCdSF11]: Ess-EL-Ei

Comment [MCdSF12]: maigreixon

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 “run” method, which builds the simulation scenario and runs it, such a line creates the required threads to execute each simulation instance.

Comment [MCdSF13]: perallell

Comment [MCdSF14]: perallell

Comment [MCdSF15]: perallell

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 dynamically created in CloudSim Plus, during simulation runtime.
- It doesn't require new Datacenter Brokers to be instantiated.
- You just have to submit new VMs or Cloudlets to “an existing” broker.

Comment [MCdSF16]: en

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.

Comment [MCdSF17]: allows

Slide 12

- **Event** **Listeners** 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.
- “There are” **Listeners** for **Events** generated from Hosts, VMs, Cloudlets and more.

Comment [MCdSF18]: Ivént

Comment [MCdSF19]: Lisseners

Comment [MCdSF20]: Lisseners

Comment [MCdSF21]: Ivénts

Slide 13

- CloudSim Plus is a strongly “object-oriented” **Framework** in which objects are used to create actual relationships, instead of using **Integer** IDs.
- It has a fluent API, allowing chained calls such as this one. This way, it is very easy to know, for instance, the Datacenter where a Cloudlet was **executed**.
- And don't worry, since it uses the “Null Object” Pattern to avoid Null Pointer Exceptions.

Comment [MCdSF22]: ekzecuied

Slide 14

- CloudSim Plus introduces classes and interfaces which specify a contract to implement **Heuristics** 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 **Heuristics** include **Tabu Search**, **Simulated Annealing** and **Ant Colony Systems**.
- It is included a Simulated Annealing **Heuristic** for mapping Cloudlets to VMs.

Comment [MCdSF24]: Riuristics

Comment [MCdSF23]: en

Comment [MCdSF25]: ekzempous

Comment [MCdSF26]: Riuristics

Comment [MCdSF27]: Tiaboo

Comment [MCdSF28]: Aniling

Comment [MCdSF29]: Riuristic

Slide 15

- The Linux Completely Fair Scheduler.
- A Cloudlet Scheduler defines how a Vm schedules the execution of Cloudlets.
- Bad scheduling may cause starvation, wastage of CPU cycles and SLA violations.
- The Completely Fair Scheduler reduces these issues, but it 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 can use the CPU at a given round.
- The current Cloudlet Scheduler Time Shared has a simplistic implementation, ignores task's priority and doesn't perform actual process preemption, as it is shown in this link.
- The Cloudlet Scheduler Completely Fair is a more realistic implementation provided by CloudSim Plus.

Slide 16

- CloudSim Plus applies functional programming to provide a functional implementation of the Datacenter Broker.
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 Datacenter Broker 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 creation of new Datacenter Broker classes.

Slide 17

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 and extensively tested;
 - get away from code duplication to avoid code degeneration;
 - and provide classes, following software engineering principles.
- CloudSim Plus "is aligned" with all these requirements.
- Proposed future work is available at the issues page.

Thank you.