* Sílabas tônicas estão em **negrito**.
* Letra sublinhada pronuncia-se como em português
* “Palavras entre aspas” são pronunciadas juntas.
* Letras ~~riscadas~~ não devem ser pronunciada

# # 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 ex**clu**sive features;
* Conclusions and future work.

# # Slide 3

* CloudSim Plus “is an-inde**pen**dent” CloudSim fork for cloud computing simulations   
  which uses the most recent features from Java 8.
* It's a highly ex**ten**sible, completely redesigned and refactored **Fra**mework,   
  “making easier” to create simulation scenarios.
* It has more than “twenty ex**clu**sive” 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 sig**ni**ficantly 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 **Co**verage by “eigh**ty** percent”, 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 Con**cer**ns principle,   
  placing only classes with the same goal into the same package.
* Finally, new inter**fa**ces were introduced to increase abstraction and define contracts for implementing classes.

**Re**searchers can “rely on these” public inter**fa**ces to create their simulations and build tools on “top of” CloudSim Plus.

# # 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 “**dar**k yellow ones” “are ex**clu**sive” to CloudSim Plus.

* The **Examples** module was updated for removal of code duplication and better organization, including “some ex**clu**sive” 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**na**lysis of scien**ti**fic~~a~~lly valid results.
* The **Benchmarks** module is used for performance assessment of cumbersome features   
  such as Heu**ri**stics.   
  It enables a **re**searcher 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 “package structure” “make easier” to find a given class.

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

* “**Dar**k yellow packages” “include ex**clu**sive” CloudSim Plus features.
* **“Li**ght 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 inter**fa**ces.   
  Existing classes were updated to fix bugs, improve documentation and design   
  and provide new features.

# # Slide 7

* "There are” more than “twenty ex**clu**sive” features.

Due to time limitation, only the most “important ones” are going to be pre**sen**ted.

* The official website **pre**sents “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 VM resources, such as RAM or CPU,   
  to be scaled up or down, according to current load and “defined static” or dynamic **thre**sholds.   
  This way, “it allows” fitting VM resources to current workload,   
  aiming to decrease 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 **thre**sholds.

“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 al**ter**native” for VM migration.

# # Slide 9

* Sometimes, simulations may take “**Se**veral 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.

# # 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 Datacenter Brokers to be instanti**a**ted.
* You just have to submit 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 **Mo**nitor 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" **Li**steners for E**ven**ts generated from Hosts, VMs, Cloudlets and more.

# # Slide 13

* CloudSim Plus is a strongly “object-oriented” **Fra**mework “in which objects” “are used to create” actual relationships, instead of using “**In**teger IDs”.
* It has a “fluent API”, allowing chained calls such as this one. This way,   
  it’s 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.

# # Slide 14

* “CloudSim Plus” introduces classes and inter**fa**ces 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 are Tabu Search, Simulated Annealing and Ant Colony Systems.
* “It’s included” a Simulated Annealing *H*eu**ris**tic for mapping Cloudlets to VMs.

# # 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 per**for**m actual process preemption,   
  “as it’s 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” “which 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’s difficult to replicate” “a real” system in simulation,   
  mainly con**cer**ned in **mo**delling “the arrival of” “stochastic e**ven**ts” such as workload bursts.
* To contribute for valid results, a simulation **Fra**mework 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.**