

Simulação de nuvens computacionais

Alunos: João Pedro Ladeira Rezende e Matheus da Silva Jahnke

Data: 28/10/2021

1 Especificação

O trabalho consiste em implementar, utilizando o *cloudsimplus*, uma simulação de uma nuvem federada com algumas instituições de ensino e pesquisa, em particular, uma contendo pelo menos 3 das 5 instituições especificadas na tabela 1.

Nome	Nivel	Módulos	PEs	RAM
UFRGS	1	8	512	4096
UFSM	1	8	512	4096
UFPel	2	4	256	2048
UNIPAMPA	3	2	128	1024
IFSUL	4	1	64	512

Tabela 1: Instituições com o número de módulos dedicados para a nuvem a ser simulada

2 Definições utilizadas pelo *cloudsimplus*

A ferramenta utiliza algumas definições que devem ser esclarecidas:

2.1 *Cloudlet*

Cloudlet se refere a uma tarefa independente das outras, que é executada em uma máquina virtual. Essa é a unidade tratada pelo *cloudsimplus*. Ela representa o processo em execução

2.2 *Virtual Machine*

Virtual Machine se refere a virtualização de forma que seja possível fornecer um computador simulado para os *cloudlets*.

2.3 *Host*

Host representa a um computador físico presente em um *Datacenter*.

2.4 *Datacenter*

Datacenter representa um conjunto de *Hosts*.

2.5 *Broker*

O *broker* é responsável por gerenciar as tarefas e as máquinas virtuais. Nele são alocadas as máquinas virtuais e as tarefas - as *cloudlets* são alocadas para as máquinas virtuais.

3 Algoritmo proposto

O algoritmo de escalonamento de *cloudlets* em máquinas virtuais proposto aqui é uma variação do *First Come, First Served*, na qual os primeiros *cloudlets* que aparecem serão processados primeiro, sem que eles sejam preemptados por outros processos. Porém o algoritmo considera a proporção das CPU utilizadas - caso 2 *cloudlets* tentem executar em uma máquina, cada uma utilizando 50% da CPU, o escalonador irá permitir que ambas executem, pois, em tese, não haverá contenção. Podemos chamá-lo na linha de comando configurando o segundo argumento para **refined**.

Para comparação, foi utilizado um algoritmo que considera somente se uma CPU está livre. Nesse caso, caso 2 *cloudlets* tentem executar em uma CPU, cada uma utilizando 50% da CPU, somente um deles irá executar, e, após ele termine, o próximo será executado. Podemos chamá-lo na linha de comando configurando o segundo argumento para **crude**.

Processos podem não utilizar totalmente a CPU mesmo que ela esteja disponível pois eles dependem de recursos diversos - tais como acesso ao disco, à internet, entre outros - esses processos são referidos como *IO-Bound*, dado que a velocidade de processamento é vinculada não ao desempenho da CPU (tal como seria se ele fosse *CPU-Bound*), mas sim ao desempenho da entrada e saída, ou seja, *IO*.

Também será utilizada a configuração padrão para escalonamento.

4 Comparação com os algoritmos existentes

Para propósitos de validação, foram elaborados três cenários possíveis que envolvem as cinco instituições de ensino e pesquisa: *small*, *medium* and *big*.

Utilizando os conjuntos de dados *big*, utilizamos os 3 escalonamentos - 2 implementados e um incluído na biblioteca *cloudsimplus*.

Como a simulação não considera o sobrecusto do chaveamento de processo, todos os escalonadores que maximizam a utilização das CPUs têm o mesmo tempo de execução.

Os resultados podem ser observados na tabela 2.

Escalonamento	Tempo de simulação	Maior atraso
crude	554	439
refined	536	469
default	536	469

Tabela 2: Resultados obtidos em função dos escalonamentos

5 Conclusão

Este trabalho desenvolveu uma simulação de nuvens utilizando a biblioteca *cloudsimplus*. Nesse trabalho foi desenvolvido um algoritmo de escalonamento com desempenho similar ao escalonamento padrão da biblioteca.