



# Análise de viabilidade da ferramenta de simulação CloudAnalyst para cálculo de performance e custo em redes de larga escala

*A. B. Fruch*      *L. F. Bittencourt*

Relatório Técnico - IC-PFG-21-48

Projeto Final de Graduação

2021 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.  
O conteúdo deste relatório é de única responsabilidade dos autores.

# Análise de viabilidade da ferramenta de simulação CloudAnalyst para cálculo de performance e custo em redes de larga escala

Arthur Biscaino Fruch\*

Luiz Fernando Bittencourt†

## Resumo

Nos últimos anos o conceito de computação em nuvem tem ganhado muito força e se popularizado muito rapidamente, nesse contexto foram desenvolvidas diversas ferramentas e *frameworks* para simular e lidar com esses tipos de soluções e auxiliar nas pesquisas e implantações fornecendo diversas métricas sobre o sistema.

O *CloudAnalyst* [2] foi desenvolvido como uma alternativa para redes de larga escala e com uma interface de usuário flexível permitindo a separação do exercício da simulação do exercício de desenvolvimento e depuração de código para executar as mesmas simulações. Este trabalho focará em explorar alguns dos recursos fornecidos pelo simulador e realizar uma análise de viabilidade apontando vantagens e desvantagens.

## 1 Introdução

É indiscutível que a computação em nuvem vem experimentando um acelerado crescimento a AWS (Amazon Web Services) uma das principais provedoras de soluções em nuvem teve um aumento na receita anual de \$ 42.262 bilhões de dólares entre 2013 e 2020 [6], essa incrível valorização já é prova suficiente do quão relevante os serviços em nuvem se tornaram nesses últimos anos. E isso se deve em grande parte ao aspecto flexível desse paradigma de computação, um dos conceitos mais interessantes é o do consumo sob demanda de recursos computacionais e por conseguinte o usuário só necessita pagar exatamente o que é utilizado e em qualquer momento for necessário um aumento de recursos computacionais, a computação em nuvem permite esse tipo de elasticidade e dinamicidade necessária para a aplicação de maneira automática e sem custo adicional, caso não seja necessária a ampliação dos recursos (mediante configuração correta).

Neste contexto foram surgindo diversas *frameworks* e ferramentas com o intuito de simular uma aplicação em nuvem para melhor entedimento de métricas básicas como por exemplo: quantidade de processamento necessária, custo, latência, etc. para executar algum evento. Uma dessas *frameworks* desenvolvidas que ganhou certa popularidade chama-se *CloudSim* [1], baseada em Java e de código aberto para a comunidade. Não demorou muito

---

\*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

†Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

tempo para que fossem criadas diversas extensões e ferramentas amparadas nesse *framework* e uma dessas ferramentas será o objetivo do estudo nesse relatório, *CloudAnalyst* [2] é um simulador com interface gráfica focada em redes de larga escala contemplando inclusive aspectos geográficos. O objetivo nesse trabalho será fazer uma análise da viabilidade dessa ferramenta para simulação dessas redes apontando vantagens e desvantagens focando principalmente em simular cenários utilizando diferentes políticas/algoritmos para controle do balanceamento de carga e seleção do *data center* que irá processar as requisições.

O relatório foi dividido de maneira que na seção 2 apresenta um resumo de conceitos básicos que serão necessários para o estudo. Na seção 3 será explicado a metodologia que foi utilizada para executar as simulações. Enquanto que na seção 4 será apresentada a análise dos resultados da simulação observando diferentes aspectos como tempo de resposta, tempo de processamento e custo total calculado da aplicação e por fim na seção 5 teremos a conclusão indicando as vantagens e desvantagens da ferramenta.

## 2 Conceitos Básicos

### 2.1 *Cloud Computing*

*Cloud computing* ou computação nas nuvens é um paradigma de computação relativamente novo que está cada vez mais sendo amplamente utilizado para diversas aplicações, devido principalmente às suas características dinâmicas, se refere tanto a aplicações entregues como serviços através da internet quanto ao hardware e software presentes nos data centers que provém esse serviço [3]. O principal ponto desse paradigma encontra-se justamente no processo de virtualização implementado pelos provedores deste tipo de serviço, permitindo uma eficiente separação dos recursos computacionais nos data centers de acordo com a demanda do usuário, além do rápido redimensionamento dos recursos computacionais fazendo uso do conceito de elasticidade dos recursos computacionais.

Entre os principais provedores de soluções nas nuvens encontram-se: *Amazon Web Services* (AWS) destacando-se o principal serviço deles o EC2 (*Elastic Compute Cloud*), *Google Cloud Platform* e *Microsoft Azure*.

### 2.2 *CloudSim* [1]

*CloudSim* é um *framework* baseado em Java, generalizado e extensível de simulação que permite modelagem, simulação e experimentação de ambientes e infraestruturas de *Cloud computing*. O *CloudSim* oferece suporte para modelagem e simulação de nuvem em grande escala, uma plataforma independente para modelagem de *data centers*, intermediários de serviços, além de políticas de alocação e agendamento para plataformas *Cloud* de larga escala. Além disso, fornece um ambiente de virtualização que auxilia na criação e gerenciamento de máquinas virtuais presentes em um *data center*.

### 2.3 *CloudAnalyst* [2]

A ferramenta de simulação *CloudAnalyst*<sup>1</sup> que será utilizada e analisada neste trabalho foi construída diretamente sobre o *framework* do *CloudSim* e faz uso de diversos recursos oferecidos pelo mesmo, para fornecer uma interface gráfica de fácil configuração e alta flexibilidade, que permite a simulação de redes de larga escala levando em consideração inclusive fatores geográficos.

O intuito da construção dessa ferramenta foi justamente apresentar uma interface simples de se utilizar mas ao mesmo tempo flexível, separando o exercício de simulação do exercício de programação permitindo maior foco nos parâmetros e modelo utilizado para o experimento. Além disso outro ponto interessante é o resultado da simulação onde os dados são expostos de maneira gráfica através de tabelas e diagramas com base nos parâmetros de entrada, fornecendo ao usuário diversas informações relevantes sobre a simulação executada.

## 3 Metodologia

O objetivo do estudo é analisar como a variação das políticas de balanceamento de carga (utilizado pelos *data centers* na alocação de requisições nas máquinas virtuais) e de seleção do *data center* para o processamento da requisição influenciam nos resultados de saída, sempre dedicando mais atenção ao tempo de processamento e resposta da requisição, além do custo estimado pela ferramenta.

### 3.1 Políticas e algoritmos oferecidos

O *CloudAnalyst* por padrão oferece três políticas de balanceamento de carga diferentes (será adicionada uma breve descrição para cada algoritmo entretanto os detalhes de cada política podem ser encontradas em detalhes aqui [2]), são elas<sup>2</sup>:

- *Round-Robin* (1): utiliza o algoritmo de round-robin para alocação das máquinas virtuais.
- *Equally Spread Current Execution Load* (de agora em diante abreviado pela sigla *ES-CEL*) (2): realiza o balanceamento das tarefas entre as máquinas virtuais disponíveis, de maneira a igualar o número de tarefas ativas em cada máquina virtual a cada dado momento. [2]
- *Throttled* (3): garante que apenas um número pré determinado de *Cloudlets* são alocados em uma única máquina virtual em um dado momento. No caso de mais grupos de requisições que máquinas virtuais disponíveis, as requisições são colocadas em uma fila de espera. [2]

Além disso a ferramenta por padrão também oferece três políticas para seleção do *data center* que irá atuar na requisição, entretanto serão utilizado apenas duas neste estudo, são elas:

---

<sup>1</sup>A ferramenta encontra-se disponível para download gratuito em: <http://cloudbus.org/cloudsim/>

<sup>2</sup>Os algoritmos serão numerados para uso posterior na seção de resultados.

- *Closest Datacenter Policy* (4): escolhe o *data center* com menor tempo de latência (sem considerar banda larga).
- *Optimize Response Time Policy* (5): esta política tenta escolher um *data center* que não esteja sobrecarregado, buscando o menor tempo de resposta.

Inicialmente foram realizados alguns testes empíricos através de simulações com grandes bases de usuários no intuito de observar o tempo necessário para finalizar a simulação e notou-se que com valores muito altos (na casa dos milhões) de usuários a ferramenta não apresentava boa performance (provavelmente devido à grande quantidade de requisições sendo processadas) demorando um elevado tempo para finalizar a execução.

Após a constatação do ponto levantado acima foi gerado o cenário com os parâmetros que serão expostos abaixo. Em seguida foram realizados testes variando as políticas de seleção do *data center*.

### 3.2 Parâmetros usados na simulação

Os parâmetros estabelecidos serão baseados em uma aplicação fictícia que possui uma grande base de usuários nas Américas dispondo de 2 *data centers* um na América do Sul e outro na América do Norte, além disso também há uma considerável quantidade de usuários na Ásia e Europa onde apenas um *data center* está instalado, abaixo seguem todos parâmetros tentando sempre se espelhar nas instâncias disponíveis na plataforma AWS EC2 [4].

### 3.3 Base de usuários

Foram definidas 4 bases de usuários em 4 diferentes regiões sendo que os horários de picos foram escolhidos de forma aleatória, segundo a tabela abaixo:

Nome	Região	Hora de Pico (GMT)	Média de usuários (pico)	Média de usuários
UB1	América do Norte	06:00-09:00	400.000	200.000
UB2	América do Sul	06:00-09:00	200.000	100.000
UB3	Europa	03:00-06:00	100.000	60.000
UB4	Ásia	03:00-06:00	100.000	50.000

Tabela 1: Tabela contendo as informações da base de usuários.

A ferramenta irá executar uma simulação de 24 horas ou seja um dia completo, cada pacote de requisição enviado por um usuário terá o tamanho de 800 bytes, um valor médio do *header* de uma requisição HTTP segundo [5] e será enviado a cada 1 minuto por usuário.

### 3.4 Data Centers

Para configurar os valores dos custos das instâncias das máquinas virtuais da simulação foi utilizado como base as o plano de custos do serviço da AWS comentado acima o EC2 [4],

além disso foram utilizados alguns parâmetros do seguinte estudo [2] para preencher os valores referentes aos três *data centers* localizados respectivamente na América do Norte, América do Sul e Europa:

Parâmetro	Valor
Custo por máquina virtual/hora	\$ 0.1
Custo de 1GB de transferência de dados para fora do <i>data center</i>	\$ 0.07
Tamanho da imagem da máquina virtual	10.000
Memória da máquina virtual	8.192 Mb
Banda larga da máquina virtual	10.000 Mb
<i>Data Center</i> - Arquitetura	x86
<i>Data Center</i> - Sistema Operacional	Linux
<i>Data Center</i> - Monitor da máquina virtual	Xen
<i>Data Center</i> - Número de máquinas	4
<i>Data Center</i> - Memória por máquina	8.192 Mb
<i>Data Center</i> - Armazenamento por máquina	100.000 Mb
<i>Data Center</i> - Banda larga por máquina	10.000 Mb
<i>Data Center</i> - Número de processadores por máquina	4
<i>Data Center</i> - Velocidade do processador	100.000 MIPS
<i>Data Center</i> - Política da máquina virtual	Partilhada no tempo

Tabela 2: Tabela contendo os parâmetros utilizados para a criação dos *data centers*.

### 3.5 Parâmetros avançados e características da internet

Além de todos os valores citados acima, ainda faltam dois pontos importantes para a simulação. O primeiro refere-se aos parâmetros que estão na aba "Advanced" do simulador, os dois parâmetros do topo equivalem respectivamente ao fator de agrupamento das bases de usuário e ao fator de agrupamento das requisições nos *data centers*. Essas propriedades apesar de que em um mundo ideal deveriam ser iguais a 1 (ou seja uma relação de um para um) para efeitos de performance da simulação precisam ser fixados com um valor mais alto como os autores explicam em [2] para que a execução do cenário não leve um tempo incrivelmente alto para finalizar. A outra propriedade que encontramos nessa aba é o tamanho de cada instrução executável por requisição que foi escolhido o valor de 300 bytes.

Por fim a ferramenta *CloudAnalyst* ainda fornece a possibilidade de configurar as características geográficas da internet que será utilizada na simulação na forma de uma matriz de latências para emular a quantidade de tempo em milisegundos que uma requisição partindo de um continente chegará ao outro, para o objetivo desse estudo foi mantida a matriz padrão que já vem configurada na ferramenta. Além disso o *CloudAnalyst* ainda oferece a possibilidade de definir uma matriz com a banda larga dentro de cada região geográfica, para este estudo teve todos valores de banda larga fixados em 10.000 Mbps.

### 3.6 Cenários simulados

Ao todo serão 6 cenários diferentes onde serão analisados e comparados os valores referentes à performance (tempo de resposta/processamento mínimo, médio e máximo) e também ao custo gerado pelas instâncias das máquinas virtuais, sendo que a aplicação será implantada com o *data center* da América do Norte (que possui mais usuários) com 25 máquinas virtuais, o *data center* da América do sul com 10 e por fim o da Europa com 10 também.

A diferença entre cada cenário simulado será justamente o tipo de política/algoritmo utilizado para o balanceamento de cargas nas máquinas virtuais juntamente com a política de seleção de *data centers* combinando todos esses algoritmos que foram mencionados acima e gerando os 6 cenários diferentes.

Os diferentes cenários estudados estão expostos na tabela abaixo, onde as diferentes políticas estão representadas de acordo com a numeração que foi determinada anteriormente na seção 3.1.

Cenário	1	2	3	4	5
1	-	X	-	X	-
2	-	X	-	-	X
3	X	-	-	X	-
4	X	-	-	-	X
5	-	-	X	X	-
6	-	-	X	-	X

Tabela 3: Representação dos cenários utilizados para o estudo

## 4 Resultados das simulações

A seguir serão apresentados, comparados e discutidos os resultados obtidos nas simulações dos cenários expostos acima; ressaltando as simulações não foram executadas mais de uma vez e calculadas as médias dos valores, pois foram feitos testes anteriormente e os valores eram extremamente próximos com diferença mínima, logo foi utilizado somente os resultados de uma única simulação para cada cenário diferente, sem nenhum tipo de prejuízo para a qualidade dos dados apresentados.

### 4.1 Análise dos tempos de resposta

Através da tabela 4 nota-se claramente pouca mudança causada pela mudança de cenário nos tempos de resposta médio de cada base de usuário, entretanto é importante salientar aqui dois pontos. O primeiro ponto é que houve um melhor desempenho considerando as políticas usadas nos cenários 5 e 6 diminuindo de maneira relativamente adequada os tempos de resposta para quase todas bases de usuário (com exceção à UB4 que apresentou desempenhos parecidos nos cenários 1 e 2). O segundo ponto importante de notar é que os tempos de resposta da quarta base de usuários são bem mais elevados que os demais,

Cenário	$T_{resposta}$ (UB1)	$T_{resposta}$ (UB2)	$T_{resposta}$ (UB3)	$T_{resposta}$ (UB4)
1	95,63 ms	84,30 ms	68,18 ms	309,83 ms
2	95,64 ms	84,29 ms	68,17 ms	309,99 ms
3	95,76 ms	83,89 ms	68,25 ms	310,05 ms
4	95,75 ms	83,89 ms	68,25 ms	310,26 ms
5	86,63 ms	73,52 ms	63,52 ms	309,78 ms
6	86,63 ms	73,50 ms	63,52 ms	309,96 ms

Tabela 4: Relação entre os cenários e os tempos de resposta médio para cada base de usuários.

isso ocorre pela distância aos *data centers*, visto que a mencionada base de usuários está localizada na Ásia causando assim uma latência maior gerada pelo aspecto geográfico.

Cenário	$T_{resposta}$ mínimo	$T_{resposta}$ médio	$T_{resposta}$ máximo
1	44,06 ms	115,30 ms	418,36 ms
2	42,36 ms	115,32 ms	408,53 ms
3	44,06 ms	115,30 ms	418,36 ms
4	42,36 ms	115,32 ms	408,53 ms
5	44,06 ms	107,58 ms	418,36 ms
6	42,36 ms	107,60 ms	507,92 ms

Tabela 5: Relação entre o tempo geral de resposta (considerando as quatro base de usuários) para os diferentes cenários e levando em conta os diferentes algoritmos apresentados e numerados anteriormente.

Realizando a análise do tempo geral de resposta (considerando todos as quatro base de usuários) exposto na tabela 5 acima, nota-se que houve pouca mudança nos valores, ou seja os algoritmos influenciaram minimamente nos valores obtidos, entretanto percebe-se uma leve vantagem no tempo de resposta médio para as combinações de políticas apresentadas nos cenários 5 e 6, em especial no cenário 5 visto que no pior caso o tempo de resposta máximo é significativamente menor que o do cenário 6. Outro ponto que deve ser ressaltado é que o tempo de resposta geral máximo se mostrou bem mais elevado que os demais devido à quarta base de usuários que está localizada na Ásia resultando em um alto tempo de latência, assim como foi descrito na análise acima.

Um ponto interessante de se notar nos cenários de 1 e 2 inclui o fato de que provavelmente os *data centers* não se sobrecarregaram nenhuma vez, ou se sobrecarregaram provavelmente não era vantajoso enviar a requisição para o outro *data center* devido à latência da rede, pois se esse caso tivesse ocorrido teríamos visualizado diferença entre os tempos de respostas dos cenários 1 e 2 com os cenários 3 e 4.

## 4.2 Análise do tempo de processamento nos *data centers*

Com a ajuda da tabela 6 observa-se novamente valores bem próximos para os cenários de 1 a 4, da mesma forma também nota-se uma vantagem de desempenho nos cenários 5 e 6, entretanto desta vez a melhoria em performance desta vez é por volta de 70% nos dois



Cenário	$T_{processamento}$ (DC1)	$T_{processamento}$ (DC2)	$T_{processamento}$ (DC3)
1	27,79 ms	25,46 ms	8,61 ms
2	27,79 ms	25,47 ms	8,61 ms
3	28,25 ms	25,07 ms	8,80 ms
4	28,24 ms	25,10 ms	8,80 ms
5	16,12 ms	13,90 ms	5,85 ms
6	16,12 ms	13,90 ms	5,85 ms

Tabela 6: Relação entre os cenários e os tempos de processamento médio para cada *data center*

primeiros *data centers* e de aproximadamente 40% para o terceiro *data center*, o que é uma diferença expressiva em relação aos outros cenários. Essa grande diferença no cenário 5 e 6 se deu pelo uso da política de *throttling*, provavelmente essa política se mostrou mais efetiva nesse caso devido às boas configurações dos *data centers* do ponto de vista de performance pois acabou garantindo que pouca ou nenhuma das instâncias das máquinas virtuais estivesse indisponível (o que prejudicaria o algoritmo, tendo que colocar a requisição em uma fila até que uma máquina virtual estivesse disponível).

### 4.3 Análise de custo

A tabela abaixo detalha o total dos custos dos *data centers* para cada cenário:

Cenário	Custo total máq. virtuais (\$)	Custo total transf. dados (\$)	Custo total (\$)
1	108,01	35.115,22	35.223,23
2	108,01	35.115,22	35.223,23
3	108,01	35.115,22	35.223,23
4	108,01	35.115,22	35.223,23
5	108,01	35.115,22	35.223,23
6	108,01	35.115,22	35.223,23

Tabela 7: Resultados referentes aos custos totais dos *data centers*

Observando a tabela 7 imediatamente fica evidente dois pontos, o primeiro é que todos valores de custo são iguais, o que de certo modo faz bastante sentido, já que em todos cenários independentemente das diferentes políticas utilizadas, a quantidade de requisições que foram enviadas para processamento foi igual, independentemente de como ou por qual *data center* a requisição foi processada, o custo de processar e enviar um retorno será sempre o mesmo.

O segundo ponto evidente se encontra no fato de que a maior parte do custo calculado pelo simulador foi sobre a taxa cobrada na transferência de dados, aqui deve ficar claro que o custo se apresentou de forma elevada devido à alguns fatores que podem até mesmo ser apresentados como uma limitação da ferramenta. O primeiro fator é que o parâmetro

que é configurado na ferramenta aplica a taxa de transferência de dados tanto para entrada quanto para saída dos dados no *data center*, o que em um cenário real nem sempre é válido, segundo este artigo [7] da AWS não há nenhum custo envolvido na transferência de dados da internet para o *data center*, logo esse valor produzido pelo *CloudAnalyst* já possui uma inconsistência quando comparado a um cenário real. Outro fator de limitação da ferramenta consiste no evento de transferência de dados dentro da mesma região que o provedor oferece, também explicado em detalhes neste artigo [7] da AWS, sendo que em alguns casos não há nenhum tipo de taxa cobrada ou o valor cobrado é bem inferior ao preço base cobrado pela transferência de dados de saída, ou seja do *data center* para a internet.

## 5 Conclusão

Neste trabalho foi apresentada a ferramenta de simulação de aplicações nas nuvens *CloudAnalyst*, o objetivo foi analisar sua viabilidade para simulação de redes de larga escala focando principalmente na variação das políticas de balanceamento de cargas nas máquinas virtuais e das políticas de seleção de *data center* para envio/processamento das requisições dos usuários.

Observando-se os dados que foram coletados das execuções das simulações é possível notar que não apresentaram grandes disparidades ao alterar os tipos de políticas que foram utilizadas, com exceção dos cenários 5 e 6 que tiveram como base a política de *throttling* onde os tempos de resposta foram ligeiramente menores e, no caso dos tempos de processamento dentro dos *data centers* houve grande melhoria de performance ao optar por esta política. Possíveis causas para que as outras políticas terem apresentado resultados similares pode-se ter dado devido à aplicação fictícia que foi criada, ou seja o contexto das bases de usuário e *data centers* não permitiu mostrar os pontos fracos/fortes das outras políticas.

Além disso foi efetuada a análise das informações de custo total gerado pela simulação e como já foi pontuado na seção 4.3 a ferramenta apresentou algumas limitações para fornecer uma geração de preço adequada, o que fica inclusive sugerido como um trabalho futuro para o simulador.

De forma geral a ferramenta apresenta uma boa interface de usuário relativamente fácil de ser usada, apesar de possuir ainda vários problemas de usabilidade como por exemplo ao finalizar uma simulação não é possível realizar outra em seguida, ou seja deve-se fechar a ferramenta e abrir novamente para ser possível realizar uma nova simulação, além de alguns outros defeitos inesperados. Entretanto é uma das poucas ferramentas desse tipo que sequer apresenta uma interface de usuário, o que por si só já é uma ótima vantagem, eliminando a parte de desenvolvimento e depuração de código, e sem dúvidas, caso fossem realizadas melhorias principalmente no quesito de usabilidade da ferramenta e apresentando mais flexibilidade para a simulação, além de gerar configurações pré estabelecidas referentes aos *data centers* para trazer opções utilizando como base as especificações dos principais provedores de serviços nas nuvens seria de grande ajuda para profissionais da área auxiliando na implantação de aplicações em grande escala.

## Referências

- [1] R. Buyya, R. Ranjan and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," *2009 International Conference on High Performance Computing & Simulation*, **2009**, pp. **1-11**, doi: **10.1109/HPCSIM.2009.5192685**.
- [2] Bhathiya Wickremasinghe and Assoc Prof and Rajkumar Buyya Contents, "CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments".
- [3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, *A view of cloud computing*, *Commun. ACM* **53** (4) (2010) **50–58**, doi:10.1145/1721654.1721672.
- [4] "Instâncias spot do Amazon EC2 (Amazon Elastic Cloud)", <https://aws.amazon.com/pt/ec2/spot/pricing/>
- [5] "Google's SPDY research project whitepaper", <http://dev.chromium.org/spdy/spdy-whitepaper>
- [6] "Annual revenue of Amazon Web Services from 2013 to 2020", <https://www.statista.com/statistics/233725/development-of-amazon-web-services-revenue/>
- [7] "Overview of Data Transfer Costs for Common Architectures", <https://aws.amazon.com/pt/blogs/architecture/overview-of-data-transfer-costs-for-common-architectures/>