Gerenciamento de Máquinas Virtuais em uma Cloud Multimídia por meio do Metaescalonamento

Maycon Peixoto[1,2] Dionisio Leite[2] Marcos Santana[2] Regina Santana[2]

[1]Universidade Federal de Viçosa - Instituto de Ciências Exatas e Tecnológicas
[2]Universidade de São Paulo – Instituto de Ciências Matemáticas e de Computação
{maycon, dionisio, mjs, rcs}@icmc.usp.br

ABSTRACT

This paper proposes two algorithms for allocation of virtual machines which deal with multimedia services. The first one is based on information from services such as computational cost and deadline of the viewing frame to adjust the amount and configuration of virtual machines. The second one estimates the runtime of a multimedia service based on the characteristics of streaming and quantity of services per user, allocating a number of virtual machines in order to meet the required deadline. Both algorithms were tested and the results are demonstrated through the design of experiments.

RESUMO

Este artigo propõe dois algoritmos para alocação de máquinas virtuais os quais lidam com serviços multimídia. O primeiro baseia-se nas informações dos serviços como custo computacional e o *deadline* da visualização dos quadros para ajustar a quantidade e a configuração das máquinas virtuais. O segundo estima o tempo de execução de determinado serviço multimídia com base nas suas características de *streaming* e na quantidade de serviços por usuário, alocando uma quantidade de máquinas virtuais com o objetivo de cumprir o *deadline* exigido. Ambos os algoritmos foram testados e os resultados são demonstrados através do planejamento de experimentos.

Categories and Subject Descriptors

H.0 [**Information Systems**]: General; C.2.4 [**Computer Systems Organization**]: Computer-Communication Networks—*Distributed Systems*

General Terms

Management, Performance, Web Services

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. Full Papers.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9642.
SBC - Brazilian Computer Society

Keywords

Cloud Multimedia, Architecture, Metascheduling

1. INTRODUÇÃO

Cloud Computing é conceituada como uma nova plataforma de oferecimento de serviços escalável. Essa plataforma é formada pela combinação e a evolução da virtualização, que geralmente incorpora infraestrutura, plataforma e software como serviço (SaaS) [24] [20] [19] [2]. Para os usuários, a Cloud é um mecanismo de utilização de recursos (hardware e software) sob demanda, os quais são alugados para entregar armazenamento e serviços computacionais sobre a Internet.

A partir do desenvolvimento da Web 2.0, a Internet Multimídia tem surgido como um serviço. As aplicações multimídia sobre a Internet requerem uma quantidade significativa de recursos de computação para servir a muitos usuários da Internet ao mesmo tempo. Nesse novo paradigma baseado em Cloud multimídia, os usuários armazenam e processam os dados de suas aplicações multimídia na Cloud. Isso ocorre de maneira distribuída, eliminando a necessidade de instalação completa dos softwares dos usuários em seus computadores, aliviando assim o peso da manutenção e da atualização, bem como economizando processamento nos dispositivos dos usuários, que para o caso de dispositivos móveis reflete na economia de carga de bateria.

Uma maneira de gerenciar as operações em um ambiente de Cloud [6] é através de um Metaescalonador. Existem alguns Metaescalonadores para o ambiente de grid, tais como o Gridway [10], Nimrod-G [3] e o Condor-G [21]. Cada um desses Metaescalonadores possui uma função determinada dentro do contexto de grid. Entretanto, nenhum deles está preocupado com as questões adicionais que envolvem especificamente a Cloud multimídia [25]. Para Cloud multimídia é necessário que um Metaescalonador leve em conta as características deste tipo de ambiente, tais como: (i) a ausência de conhecimento por parte do cliente sobre os detalhes dos serviços de áudio, vídeo, imagens (do tipo SaaS); (ii) o tipo de hardware que está executando os serviços multimídia ; (iii) onde está localizado esse hardware e sobre as suas configurações; e o principal, (iv) o gerenciamento da virtualização.

Assim, os Metaescalonadores de propósito geral existentes para grid podem não estar preparados para lidar com a alocação de máquina virtual que ocorre na Cloud multimídia. Por isso, o objetivo deste trabalho é adicionar ao **MACC - Metascheduler Architecture to Provide QoS in Cloud Computing** os algoritmos de alocação e gerenciamento de recursos virtualizados apropriados para lidar

com questões de multimídia, usando algoritmos que utilizam heurísticas de tempo-real para obtenção de QoS - (*Quality of Service*) [18] [17].

Os ambientes de virtualização podem ser compostos de duas formas: (i) aqueles que garantem que a vCPU pode ser implementada sem compartilhamento entre os demais processos de outros usuários, como no Amazon EC2 [23] que utiliza o Xen [13], ou (ii) onde os usuários compartilham as vCPUs de acordo com a quantidade de usuários naquela máquina física, como é o caso do VMWare [8].

Levando em conta esses dois processos distintos de virtualização, os algoritmos desenvolvidos nesse projeto são construídos para garantir as restrições temporais do serviço desejado pelo cliente. Os algoritmos propostos pelo MACC apresentaram resultados adequados em relação ao cumprimento da SLA - (*Service Level Agreement*). Outros trabalhos na literatura não oferecem garantias orientadas ao usuário com as métricas adequadas, sendo que normalmente as métricas consideradas são orientadas ao sistema [12] [14] [15].

O restante desse artigo é estruturado como se segue. Na Seção 2 são apresentados os conceitos básicos sobre a Cloud multimídia. Na Seção 3 é feita uma discussão sobre o MACC e os seus componentes. Na Seção 4 é apresentado o planejamento de experimentos adotado no projeto. Já na Seção 5 são feitas as análises sobre os resultados obtidos. Na Seção 6 são apresentados os trabalhos relacionados na literatura. Finalmente, na Seção 7 são apresentadas as considerações finais e algumas direções para trabalhos futuros.

2. CLOUD MULTIMÍDIA

Segundo [25], o processamento de serviços multimídia em uma Cloud apresenta vários desafios que são identificados a seguir:

- Serviços multimídia heterogêneos: a Cloud multimídia deve lidar com os diferentes tipos de serviços, tais como voz sobre IP (VoIP), vídeo conferência, compartilhamento de fotos, *streaming* multimídia, etc.
- Heterogeneidade de QoS: como os serviços de multimídia possuem diferentes requisitos de QoS, a Cloud deve realizar o provisionamento de QoS para os diferentes tipos de serviços multimídia.
- Heterogeneidade da rede: as redes (Internet, LAN, wireless) possuem diferentes características, tais como largura de banda, atraso e variação no atraso. Portanto, a Cloud multimídia deve se adaptar aos diferentes tipos de contextos para adequar-se a entrega de serviços aos vários dispositivos dispersos geograficamente.
- Heterogeneidade de dispositivos: existem diferentes tipos de dispositivos, tais como TVs, computadores pessoais, celulares, e cada um deles possuem diferentes capacidades de processamento. A Cloud multimídia deve se adaptar às diversas capacidades para atender aos diferentes tipos de dispositivos, incluindo CPU, GPU, display, memória, armazenamento e carga da bateria.

Em Cloud computing, os mecanismos de gerência (dinamicamente) ou os próprios usuários (estaticamente) devem alocar a quantidade de processamento e armazenamento que eles precisam para executarem seus serviços de propósito geral. Entretanto, para aplicações multimídia, em adição aos requisitos de CPU e de armazenamento, um outro fator importante é o requisito de QoS em termos de largura de banda, atraso (*delay*), e variação do atraso (*jitter*) [7].

Para atender aos requisitos de QoS multimídia em Cloud sobre a Internet, é apresentado nesse artigo um mecanismo chamado de Metaescalonador. O Metaescalonador lida com os principais conceitos citados nesta Seção, especificamente ele leva em conta as características de aplicações multimídia e atrasos na rede com a adoção de técnicas de escalonamento de tempo-real.

3. MACC

O MACC, Figura 1, tem sido descrito em outros trabalhos [18] [17]. Neste artigo é apresentada a modelagem das políticas de gerenciamento e a alocação de recursos virtualizados do MACC. A abordagem foi dividida em dois momentos: (1) a escolha do datacenter que executará o serviço e (2) a configuração das máquinas virtuais a serem alocadas para a execução.

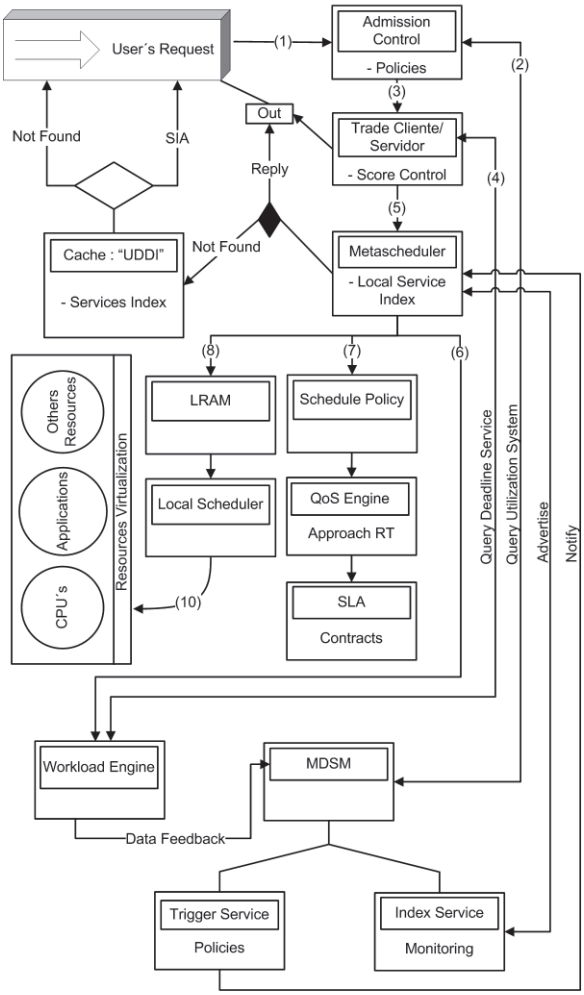

Figura 1: MACC.

Os principais componentes do MACC com suas respectivas funções são descritos da seguinte forma:

Controle de Admissão : Realiza o controle do fluxo de serviços multimídia que podem ser aceitos de acordo com métricas de utilização.

Trade Cliente/Servidor : Controle desde a construção do preço de um serviço até sua negociação com os clientes.

Trigger Service : Dispara um evento quando uma máquina virtual começa a entrar no estado de sobrecarga.

Index Service : Armazena informações sobre as variáveis do sistema e sobre os serviços multimídia em execução, realizando uma analogia ao UDDI (*Universal Description, Discovery and Integration*) distribuído.

Workload Engine : Manipula as informações sobre as condições do sistema e caracteriza o custo computacional de um serviço.

QoS Engine : Retrata as informações sobre a QoS atual de forma dinâmica.

SLA : Registra os contratos estabelecidos entre o cliente e o provedor.

LRAM : Gerenciador de Alocação de Recursos Local.

Cache:UDDI : Armazena informações sobre serviços multimídia locais e sobre os serviços remotos.

Políticas de Escalonamento : Diretório das políticas de roteamento e das políticas de alocação de máquinas virtuais.

3.1 Políticas de Escalonamento do MACC

Quanto à política de escolha do datacenter, este projeto adotou dois **algoritmos de roteamento**:

- Round-Robin - (RR): O algoritmo Round-Robin é um algoritmo de escalonamento que atribui requisições com base em lógica circular. No caso do ambiente proposto neste projeto, as diversas requisições serão escalonadas entre os três datacenters repetindo o ciclo.
- Network-based - (NB): Algoritmos Network-based utilizam a topologia e características da rede para o escalonamento de requisições. Neste projeto, o algoritmo adotado baseia-se no histórico da latência da conexão na topologia de rede.

Em relação à escolha do número de máquinas virtuais alocadas para uma determinada aplicação, foram consideradas duas **políticas de alocação de VM**:

- Alocação Dinâmica Adaptativa - (ADA): baseia-se nas informações dos serviços multimídia como custo computacional e o *deadline* para ajustar uma quantidade de máquinas virtuais com potência computacional variável, aproximando-se assim do número ideal de máquinas virtuais (em termos de custo e potência computacional) para atender os requisitos de QoS de determinada aplicação. Ressalta-se que a política em questão leva em consideração as limitações da máquina física durante a escolha da potência da máquina virtual, uma vez que este não pode ultrapassar o limite estabelecido na criação da máquina física. As máquinas virtuais criadas por esta política apresentam potências computacionais flexíveis, mas conservam os valores para os outros parâmetros já descritos.

- Alocação Dinâmica Fixa - (ADF): estima o tempo de execução de determinado serviço, com base nas suas características e na quantidade de serviços multimídia por usuário, alocando uma quantidade de máquinas virtuais de acordo com a estimação a fim de cumprir o *deadline* exigido. Para essa política, as máquinas virtuais criadas apresentam a mesma potência computacional.

3.2 Modelo Econômico do MACC

Um modelo econômico é determinado por um conjunto de recursos, um conjunto de agentes, (consumidores e produtores), e um conjunto de regras que especificam a interação entre os recursos e os agentes. A utilização de um modelo econômico ao MACC fornece aos participantes de uma Cloud Computing Colaborativa uma plataforma para realizarem as trocas de serviços multimídia [4] [9].

O modelo econômico adotado no MACC é o baseado na oferta/demanda. Este tipo de abordagem habilita o ambiente a compartilhar os seus recursos. A oferta refere-se à quantidade de um bem ou serviço que os produtores desejam vender. Já a demanda é a quantidade de um bem ou serviço que os consumidores desejam adquirir. Se a demanda é superior a oferta, os consumidores tendem a elevar o valor de lance, o que causa um aumento de preço sobre o serviço. Por outro lado, se a oferta é maior que a demanda, os vendedores tendem a diminuir os preços. Portanto, esse tipo de ambiente oferece flexibilidade e amplitude de desempenho para todos os interessados. Outro ponto a favor desta abordagem é que ela reage de forma adequada às mudanças do ambiente e não necessita de um coordenador.

Um dos pontos importantes do MACC é a adoção de um *deadline* como métrica de um serviço do cliente. De acordo com o *deadline* do serviço contratado, o componente *Trade Servidor* determinará qual o preço o cliente irá pagar. Por levar em conta as variáveis de custo e de *deadline*, essa abordagem provê um incentivo para os provedores compartilharem seus recursos (além de dados: vídeo, áudio, imagens) no ambiente de Cloud colaborativa.

O preço de um recurso pode influenciar a sua utilização, porque os consumidores tendem a utilizar aqueles que têm o menor preço, já que todos os serviços multimídia são funcionalmente iguais. O componente *Trade Servidor* lida com um mecanismo para ajuste automático do preço, objetivando maximizar a taxa de utilização dos recursos. O ajuste do preço deve ser realizado para atender as mudanças do ambiente econômico. O administrador do provedor pode determinar o peso de cada parâmetro envolvido, tais como: demanda (α), oferta (β), demanda/oferta (ϱ), taxa de utilização (δ) e (γ), e *deadline* (ϵ) variando de 0 a 1. Para o ajuste do preço são considerados os seguintes parâmetros:

- Demanda (d_i): O preço para executar um serviço é proporcional a sua demanda, onde (d_{i-1}) foi a última

demanda obtida. Equação 1.

$$d_i = \frac{(d_i - d_{i-1})}{d_{i-1}} * \alpha \quad (1)$$

- Oferta (O_i): O preço para executar um serviço é inversamente proporcional ao seu grau de oferecimento. Equação 2.

$$O_i = \frac{(O_i - O_{i-1})}{O_{i-1}} * \beta \quad (2)$$

- Demanda/Oferta (dO_i): Representa a relação entre o montante requisitado e o montante oferecido para um serviço. Se a oferta é mais alta do que a demanda, o preço tende a diminuir. Entretanto, se a demanda é mais alta do que a oferta, o preço tende a aumentar. Caso exista uma igualdade entre demanda e oferta, pode-se dizer que a Grade tem alcançado o equilíbrio. Equação 3.

$$dO_i = \frac{(d_i - O_i)}{min(d_i, O_i)} * \varrho \quad (3)$$

- Taxa de utilização (U_i): Se a taxa de utilização é alta, é possível aumentar o valor dado ao serviço. Se esse serviço está sendo pouco utilizado, o preço cobrado pode estar alto e deve ser reduzido. Equação 4.

$$U_i = \left(\left(1 - \frac{U_{i-1}}{100}\right) * \delta\right) + \left\lfloor \frac{U_{i-1}}{100} \right\rfloor * \gamma \quad (4)$$

- *Deadline* (D_i): Essa variável referencia o tempo de entrega de um serviço. Se esse tempo é alto, o preço torna-se baixo. Por outro lado, se esse tempo é baixo, o preço torna-se alto. Equação 5.

$$D_i = \frac{(D_i - D_{i-1})}{D_{i-1}} * \epsilon \quad (5)$$

A Equação 6 mostra como obter o novo preço (P_i) em relação a estes parâmetros que mudam sobre o tempo (dado por i).

$$P_i = P_{i-1}\,(1 + (d_i - O_i) + dO_i - D_i - U_i) \quad (6)$$

Depois do estabelecimento do preço base (P_i), o valor do serviço é negociado. O MACC procura atender a requisição dentro do prazo contratado. Será possível verificar nos resultados desse projeto que a responsabilidade de garantir esse critério é do algoritmo de alocação de máquinas virtuais.

3.3 Modelo de Comunicação do MACC

A Figura 2 apresenta o modelo de comunicação entre os vários Metaescalonadores de uma federação. Assim, cada Metaescalonador é um nodo na camada lógica P2P e entre eles é eleito um Superescalonador. O Superescalonador é aquele que tem a maior potência computacional. Ele recebe uma requisição e descobre quem poderá fornecer aquele serviço.

O MACC possui uma camada lógica P2P entre as entidades para melhorar o desempenho, disponibilidade, e escalabilidade dos serviços multimídia da Cloud. O MACC explora alguns pontos desse tipo de ambiente para prover QoS na Cloud multimídia, tais como:

- Coleta de informações: considerando a carga e a interconexão dos nodos na rede P2P;

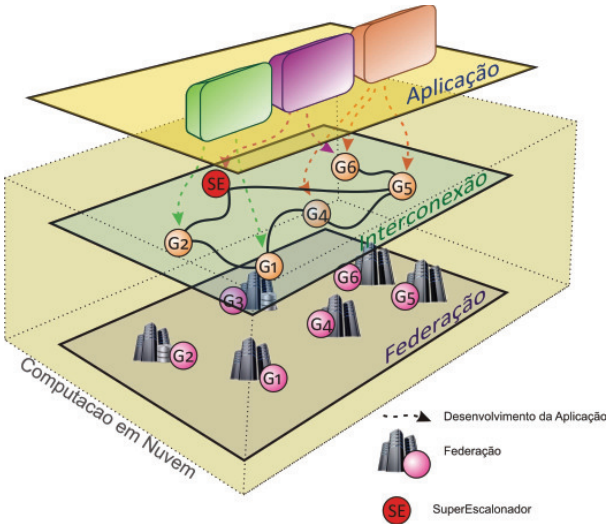

Figura 2: Interligação das Camadas da Cloud Colaborativa. Adaptado de [6].

- Crescimento escalável: os protocolos de roteamento devem considerar que o ambiente estará inclinado a crescer;
- Eleição do Metaescalonador: define políticas e algoritmos para eleição do Metaescalonador;
- Eleição do Superescalonador: baseado na capacidade dos provedores de recursos em uma federação;

O MACC utiliza o modelo hierárquico P2P. Neste trabalho a rede de interconexões, formado pelos clientes e os datacenters, é estruturada segundo mostrado na Figura 3 e na Figura 4. A ideia sobre esse modelo BRITE [16] de rede é usar o Superescalonador para desempenhar duas funções: i) controlar parcialmente a sobrecarga na rede de serviços multimídia e ii) melhorar o desempenho na consulta para serviços multimídia no grafo de conectividade minimizando o congestionamento.

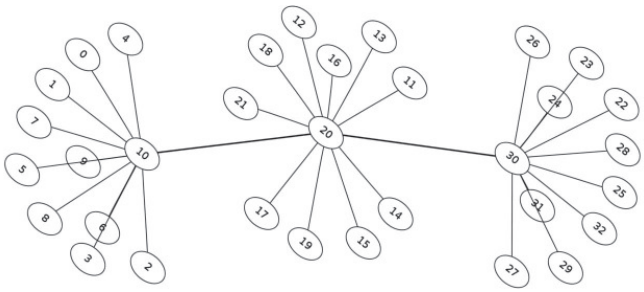

Figura 3: Grafo para o Cenário 1: Experimentos Ímpares.

O acesso a uma entidade utiliza $O(logN)$ mensagens para um sistema DHT - (*Distributed hash table*) localizar quem possui o serviço desejado. Uma vez que o serviço foi localizado na árvore são necessários em média $O(log_t n)$ acessos sobre a rede da entidade, isso significa a altura da árvore. O t é o número de índices em uma entidade e n é o total de

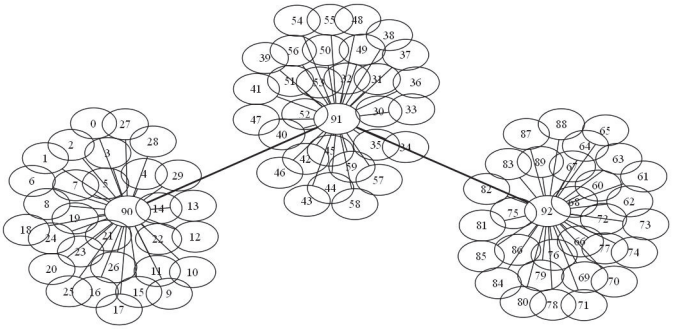

Figura 4: Grafo para o Cenário 2: Experimentos Pares.

números de elementos. Em termos gerais, é assumido que para localizar uma entidade que contem o serviço requerido são enviadas $O(logN * log_t n)$ mensagens, para N nodos no ambiente.

4. PLANEJAMENTO DE EXPERIMENTOS

Foi utilizado para realizar os testes o ambiente de simulação CloudSim 2.1 [5]. Já o planejamento de experimentos segue as sugestões apresentadas em [11]. Assim, cada federação da Cloud multimídia objetiva desempenhar todos os serviços de um usuário seguindo três métricas principais, elas são:

- Tempo de Resposta Médio (*MRT - Mean Response Time*)
- Custo (C - Cost)
- Confiabilidade (R - Reliability)

Essas métricas são orientadas ao usuário. Nesse projeto o foco primário é o lado do usuário. Em especial, existe a métrica confiabilidade. A confiabilidade está ligada a dependabilidade e ela cobre a maioria dos aspectos relacionados à satisfação de um usuário nesse contexto. Confiabilidade leva em consideração o tempo médio de resposta através do atendimento da variável *deadline*. Os principais fatores que influenciam as variáveis de reposta são:

A - Algoritmo de Roteamento do MACC (Round Robin e Network Based).

B - Algoritmo de Alocação de Máquinas Virtuais do MACC (ADF e ADA).

C - Número de serviços multimídia por usuário (10 e 30).

D - Número de usuários (30 e 90).

Nesse trabalho, foi escolhido o projeto fatorial completo. O projeto fatorial completo mede cada variável de resposta usando todos os tratamentos (combinações entre os níveis de fatores). Um projeto fatorial completo para n fatores com $N1, ..., Nn$ níveis, requer $N1 \times ... \times Nn$ execuções de experimentos. Assim, os experimentos para o algoritmo de roteamento (A), algoritmo de alocação (B), serviços multimídia (C), e usuários (D) são: Nn = {{Round Robin, Network Based} + {ADF, ADA} + {10, 30} + {30, 90}}!, produzindo a Tabela 1.

Tabela 1: Projeto de Experimentos

Exp	Roteamento (A)	Alocação de VM (B)	Serviços (C)	Usuários (D)
n_1	Round Robin	ADF	10	30
n_2	Round Robin	ADF	10	90
n_3	Round Robin	ADF	30	30
n_4	Round Robin	ADF	30	90
n_5	Round Robin	ADA	10	30
n_6	Round Robin	ADA	10	90
n_7	Round Robin	ADA	30	30
n_8	Round Robin	ADA	30	90
n_9	Network Based	ADF	10	30
n_{10}	Network Based	ADF	10	90
n_{11}	Network Based	ADF	30	30
n_{12}	Network Based	ADF	30	90
n_{13}	Network Based	ADA	10	30
n_{14}	Network Based	ADA	10	90
n_{15}	Network Based	ADA	30	30
n_{16}	Network Based	ADA	30	90

O modelo para um projeto 2^4 é dado pela Equação abaixo:

$$\begin{aligned} y &= q_0 + q_A x_A + q_B x_B + q_C x_C + q_D x_D + q_{AB} x_{AB} \\ &+ q_{AC} x_{AC} + q_{AD} x_{AD} + q_{BC} x_{BC} + q_{BD} x_{BD} + q_{CD} x_{CD} \\ &+ q_{ABC} x_{ABC} + q_{ABD} x_{ABD} + q_{ACD} x_{ACD} + q_{BCD} x_{BCD} \\ &+ q_{ABCD} x_{ABCD} \end{aligned} \quad (7)$$

Substituindo as observações no modelo, o novo arranjo de valores fica da seguinte maneira: $q_A, q_B, q_C, q_D, q_{AB}, q_{AC}, q_{AD}, q_{BC}, q_{BD}, q_{CD}, q_{ABC}, q_{ABD}, q_{ACD}, q_{BCD}, q_{ABCD}$. Por exemplo, existe q_0, como se segue a Equação abaixo:

$$\begin{aligned} q_0 = 1/16 * (&y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 \\ &+ y_{10} + y_{11} + y_{12} + y_{13} + y_{14} + y_{15} + y_{16}) \end{aligned} \quad (8)$$

A partir dos valores obtidos pode-se determinar a soma total de quadrados. A variação total ou soma total dos quadrados (SST) é dada por:

$$SST = \sum_{i,j} (y_{ij} - \bar{y}) \quad (9)$$

Aqui, $\bar{y}$ denota a média de resposta a partir de todas as replicações de todos os experimentos. Assim, ele é adicionado nos termos dessa equação em todas as 2^4 observações, gerando:

$$SST = 2^4(q_A^2 + q_B^2 + q_C^2 + q_D^2 + ... + q_{ABCD}^2) \quad (10)$$

O SST oferece a variação total da variável resposta medida e fornece também a influência de cada fator e interação no sistema, com base no modelo de regressão. Por exemplo, supõe-se querer saber o valor do Fator de A. Isto é obtido por: $y = SSA/SST$, onde $SSA = 2^4 q_A^2$. Na Tabela 2 são mostrados todos os fatores e suas interações.

A primeira métrica analisada é o custo total no ambiente da nuvem, a influência sobre ela tem sido mais impactada pelo total de serviços multimídia por usuário (C). O custo é proporcional à carga de trabalho do usuário, referindo-se a quantidade de usuários (D) e a sua quantidade de serviços multimídia (C). O custo também está relacionado com um total de máquinas virtuais criadas, tornando-se, em casos

Tabela 2: Influência dos Fatores.

	Custo (%)	MRT (%)	Confiabilidade (%)
q_A	0,00	0,02	0,02
q_B	1,69	3,58	82,51
q_C	31,02	1,79	0,02
q_D	52,09	90,42	8,65
q_{AB}	0,00	0,00	0,02
q_{AC}	0,00	0,01	0,02
q_{AD}	0,00	0,02	0,02
q_{BC}	3,75	1,71	0,02
q_{BD}	0,35	0,90	8,65
q_{CD}	10,23	1,06	0,02
q_{ABC}	0,00	0,00	0,02
q_{ABD}	0,00	0,00	0,02
q_{ACD}	0,00	0,01	0,02
q_{BCD}	0,87	0,49	0,02
q_{ABCD}	0,00	0,00	0,02

normais equivalente à quantidade de máquinas virtuais que foram criadas.

Por outro lado, a confiabilidade é influenciada pelo algoritmo de alocação de máquina virtual (B). Esse parâmetro está relacionado com o cumprimento do *deadline*, ou seja, pode-se concluir que o responsável pelo cumprimento da SLA será o algoritmo de alocação de máquina virtual, representado pelos algoritmos ADF e ADA. O algoritmo realiza uma importante função de decidir quantas máquinas virtuais devem ser criadas, objetivando atender o *deadline* tanto quanto economizando o custo para o usuário. Se o MACC tomar uma decisão errada, ele poderia gastar mais recursos do que o necessitado e ainda assim não atender a confiabilidade exigida. Principalmente, no caso da confiabilidade, o algoritmo de alocação de máquinas virtuais tem uma importante função de determinar a porcentagem de atendimentos dos *deadlines* impostos, porque ele te uma influência na confiabilidade igual a $82,51\%$.

Também é possível realizar uma observação direcionada aos fatores do ambiente. A primeira delas é o algoritmo de roteamento (A). Ele não faz qualquer diferença, porque todas as federações têm o mesmo custo por máquina virtual criada. Com relação às diferenças computacionais entre as federações, o impacto das variáveis de resposta não são afetadas pelo algoritmo de roteamento, porque o MACC pode criar qualquer quantidade de máquinas virtuais desejadas, dependendo somente do algoritmo de alocação de máquina virtual, e levando em conta a quantidade de serviços multimídia por usuário.

Outra observação é a ausência de sobrecarga na federação, pois todos os datacenters possuem máquinas físicas suficientes para atender a demanda. Portanto, não é necessário o balanceamento de carga entre os datacenters, pois para obter bons resultados para as métricas em questão é a política de alocação de máquinas virtuais a responsável por lidar com as questões da SLA. Isso deve-se ao fato de não haver interferência de sobrecarga entre as máquinas virtuais criadas, como é o caso das máquinas virtuais com rodam sobre o Xen, onde pode-se dedicar núcleos de processamento (vCPUs), isolando os códigos das máquinas virtuais sobre a mesma máquina física. Em uma análise mais detalhada, podemos concluir que apesar dos datacenters possuírem diferentes características, será o algoritmo de criação de máquinas virtuais o responsável pela criação de uma máquina virtual adequada, ou seja, que leve em consideração as características do datacenter e do serviço multimídia. Uma métrica que poderia ser afetada aqui seria o Custo, caso fosse adotada um tipo de economia onde cada provedor/datacenter altera-se os preços dinamicamente, não ocorrendo a entrada do estado de equilíbrio, como é visto na Figura 5.

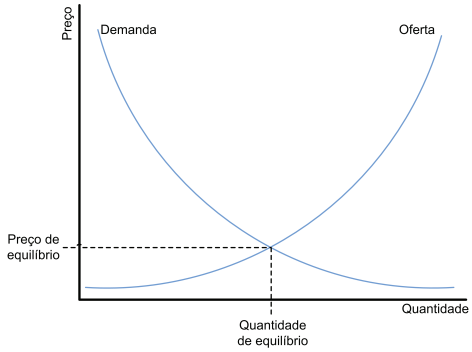

Figura 5: Relação entre a demanda e a oferta.

Uma última observação pode ser agrupada na classe de carga de trabalho do usuário, composta pelo total de usuários (D) e o total de serviços multimídia por usuário (C). Como já foi dito, ambos os fatores tem mais impacto na variável de resposta do custo do sistema, apresentando uma influência de $93,24\%$, que é representada por q_C, q_D e q_{CD}.

5. ANÁLISE DOS RESULTADOS

A Figura 6 apresenta a comparação entre as duas políticas de alocação de máquinas virtuais. As figuras consideradas nesta Seção são configuradas através das proporcionalidades das variáveis. Analisando a Figura 6 nota-se um ganho considerável da política ADA em relação a política ADF. Isso deve ao fato da política ADA adaptar-se as necessidade do cumprimento da SLA do cliente, criando a máquina virtual com a potência computacional adequada para aquele cenário.

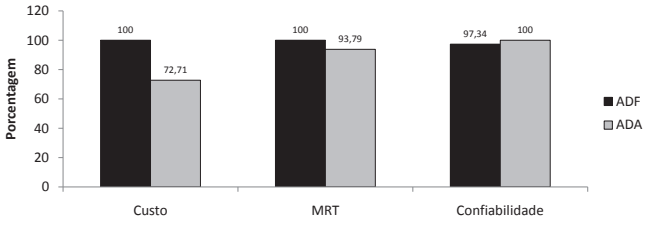

Figura 6: Comparação entre as políticas de alocação de VMs.

É importante ressaltar que o algoritmo de alocação de máquinas virtuais ADF havia sido um avanço em relação aos algoritmos que não levam em consideração o *deadline* imposto na SLA. Em experimentos realizados anteriormente pode-se perceber, Figura 7, o ganho do ADF em relação ao algoritmo de criação de máquinas virtuais aleatório de vetor numerado $[1-5]$, chamado AVN. O AVN não realiza nenhum tipo de consideração da métrica de *deadline*, criando as máquinas virtuais de forma aleatórias dentro de um subconjunto numerado que vai de 1 a 5.

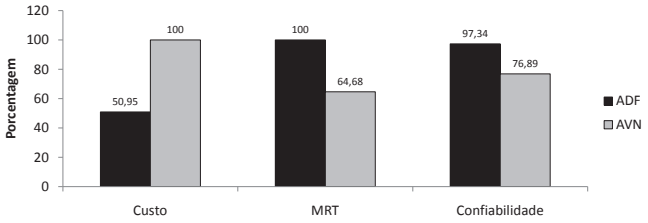

Figura 7: Comparação entre as políticas de alocação de VMs.

A análise do gráfico da Figura 8 mostra que a política de alocação flexível, proposta neste projeto apresentou melhores resultados em relação às variáveis de resposta consideradas. Nota-se que apesar da alta confiabilidade de ambas as políticas (97,34% e 100%), a política se flexível se mostrou melhor, apresentando também um menor tempo de resposta e um menor custo total. Essa configuração se deve à diferença na criação de máquinas virtuais entre as duas políticas. Enquanto a ADF cria máquinas virtuais de mesma potência computacional, a ADA se adapta à carga de trabalho, criando máquinas cuja potência de computacional seja o mais próximo da ideal para cada aplicação.

A diferença no custo pode ser explicada da seguinte forma: em algumas situações, a potência computacional das máquinas virtuais criadas pela ADF (tamanho fixo) ultrapassa o ideal para a execução de determinada aplicação, ou seja, existe um gasto desnecessário quando da criação de tais máquinas, fato este que tende a ser evitado ao máximo pela ADA; outra situação leva em consideração também o número de máquinas criadas, podendo ocorrer que o custo total para criação de 5 VMs com 1 vCPU na ADF seja maior que o custo total para criação de 3 VMs de 2 vCPUs criadas pela ADA. Como o cálculo do custo final engloba a quantidade e potência computacional das máquinas virtuais criadas, esta diferença de custo é assim justificada. A Figura 8 realça o desempenho da ADA em garantir a confiabilidade mesmo com o aumento da carga de trabalho.

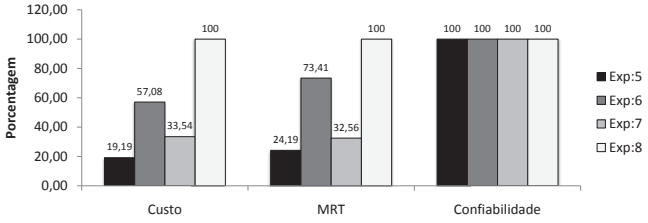

Figura 8: ADA em relação ao aumento da carga de trabalho.

Pode-se observar que apesar do aumento da carga de trabalho, o algoritmo mantém a confiabilidade em 100%, isto é, cumpre os *deadlines* exigidos pelos usuários na execução de seus serviços. Este é um comportamento interessante para ambientes Cloud cuja principal preocupação seja a manutenção do QoS no provisionamento de serviços multimídia mesmo com possíveis picos de demanda, com picos de demandas para o caso de transmissão de video.

6. TRABALHOS RELACIONADOS

Segundo [25], a computação em alta escala de multimídia tem sido relacionada a computação em grids, CDN (*content delivery network*) e P2P. Mais especificamente sobre a computação sobre grids, a computação é aplicada a alto desempenho (HPC) [1]. Por outro lado, a CDN lida com questões de entrega de serviço, reduzindo a latência ou maximizando a largura de banda para os clientes acessarem os dados. Alguns exemplos são: Akamai Technologies, Amazon CloudFront, e Limelight Network. O Youtube utiliza a CDN da Akamai para fornecer os vídeos na Internet. Por fim, a computação em P2P refere-se a aplicações distribuídas que dividem as tarefas de computação de multimídia entre os nodos da rede. Exemplos incluem: Skype, PPlive.

A Cloud multimídia apresentada nesse artigo identifica como a Cloud pode prover QoS para serviços de áudio, vídeo, e imagens em grande escala por meio de um mecanismo de gerência chamado Metaescalonador.

Para conhecimento, existem poucos trabalhos sobre Cloud multimídia na literatura. A IBM teve uma iniciativa de âmbito geral (http://www.ibm.com/ibm/cloud/resources.html#3). Já Trajdovska propõe uma união entre P2P e a arquitetura de Cloud para o fornecimento de fluxo de multimídia através de funções de custo de QoS [22].

7. CONCLUSÕES FINAIS

Este artigo apresentou um estudo e uma proposta de modelagem e avaliação de desempenho para ambientes de Cloud multimídia. Apesar de ser um paradigma relativamente novo, a Cloud multimídia está em crescente expansão, fato este que expõe suas principais limitações e desafios futuros. Nesse contexto, este artigo concentrou-se no que é hoje um dos principais desafios para as Clouds: a manutenção da qualidade de serviço através da adoção de políticas de gerenciamento dinâmicas. A política de alocação de máquinas virtuais neste projeto apresentou bons resultados em relação às variáveis de resposta consideradas, sendo, portanto, uma escolha adequada para a análise e comparação em ambientes e projetos futuros.

8. AGRADECIMENTOS

Os autores agradecem a ajuda financeira da CAPES, CNPq e a FAPESP que tem contribuído com esse projeto.

9. REFERÊNCIAS

[1] B. Aljaber, T. Jacobs, K. Nadiminti, and R. Buyya. Abstract multimedia on global grids: A case study in distributed ray tracing. In *Malays. J. Comput. Sci., vol 20, no. 1, pp. 1-11*, 2007.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[3] R. Buyya, D. Abramson, and J. Giddy. Nimrod/g: an architecture for a resource management and scheduling system in a global computational grid. In *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International*

Conference/Exhibition on, volume 1, pages 283–289 vol.1, 2000.

[4] R. Buyya, S. Pandey, and C. Vecchiola. Cloudbus toolkit for market-oriented cloud computing. In *Proceedings of the 1st International Conference on Cloud Computing*, CloudCom '09, pages 24–44, Berlin, Heidelberg, 2009. Springer-Verlag.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In *Proceedings of the 7th High Performance Computing and Simulation Conference*, volume abs/0907.4878 of *HPCS 2009*, Leipzig, Germany, 2009. IEEE Press.

[6] R. Buyya, R. Ranjan, and R. N. Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010, Busan, South Korea, May 21-23), LNCS, Springer, Germany, 2010.*, abs/1003.3920, 2010.

[7] K. De Moor, I. Ketyko, W. Joseph, T. Deryckere, L. De Marez, L. Martens, and G. Verleye. Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting. *Mob. Netw. Appl.*, 15:378–391, June 2010.

[8] M. Dowty and J. Sugerman. Gpu virtualization on vmware's hosted i/o architecture. *SIGOPS Oper. Syst. Rev.*, 43:73–82, July 2009.

[9] S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh. Efficient service recommendation system for cloud computing market. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, pages 839–845, New York, NY, USA, 2009. ACM.

[10] E. Huedo, R. S. Montero, and I. M. Llorente. A framework for adaptive execution in grids. *Softw. Pract. Exper.*, 34:631–651, June 2004.

[11] R. Jain. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley, 1991.

[12] R. Jeyarani, R. V. Ram, and N. Nagaveni. Design and implementation of an efficient two-level scheduler for cloud computing environment. In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, CCGRID '10, pages 585–586, Washington, DC, USA, 2010. IEEE Computer Society.

[13] M. Lee, A. S. Krishnakumar, P. Krishnan, N. Singh, and S. Yajnik. Supporting soft real-time tasks in the xen hypervisor. *SIGPLAN Not.*, 45:97–108, March 2010.

[14] L. Li. An optimistic differentiated service job scheduling system for cloud computing service users and providers. In *Proceedings of the 2009 Third International Conference on Multimedia and Ubiquitous Engineering*, MUE '09, pages 295–299, Washington, DC, USA, 2009. IEEE Computer Society.

[15] Q. Li, Q. Hao, L. Xiao, and Z. Li. Adaptive management of virtualized resources in cloud computing using feedback control. In *Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering*, ICISE '09, pages 99–102, Washington, DC, USA, 2009. IEEE Computer Society.

[16] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: Universal topology generation from a users perspective. Technical report, 2001.

[17] M. L. Peixoto, M. J. Santana, and R. H. Santana. A p2p hierarchical metascheduler to obtain qos in a grid economy services. *Computational Science and Engineering, IEEE International Conference on*, 1:292–297, 2009.

[18] M. L. M. Peixoto, M. J. Santana, J. C. Estrella, T. C. Tavares, B. Kuehne, and R. H. C. Santana. A metascheduler architecture to provide qos on the cloud computing. In *ICT '10: 17th International Conference on Telecommunications*, pages 650–657, Doha, Qatar, 2010. IEEE Computer Society.

[19] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li. Comparison of several cloud computing platforms. In *ISISE '09: Proceedings of the 2009 Second International Symposium on Information Science and Engineering*, pages 23–27, Washington, DC, USA, 2009. IEEE Computer Society.

[20] B. P. Rimal, E. Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *NCM '09: Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 44–51, Washington, DC, USA, 2009. IEEE Computer Society.

[21] D. Thain, T. Tannenbaum, and M. Livny. Condor and the grid. In *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley, 2003.

[22] I. Trajkovska, J. Salvachua Rodriguez, and A. Mozo Velasco. A novel p2p and cloud computing hybrid architecture for multimedia streaming with qos cost functions. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1227–1230, New York, NY, USA, 2010. ACM.

[23] G. Wang and T. S. E. Ng. The impact of virtualization on network performance of amazon ec2 data center. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 1163–1171, Piscataway, NJ, USA, 2010. IEEE Press.

[24] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl. Scientific cloud computing: Early definition and experience. In *Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications*, pages 825–830, Washington, DC, USA, 2008. IEEE Computer Society.

[25] W. Zhu, C. Luo, J. Wang, and S. Li. Multimedia cloud computing. *IEEE Signal Processing Magazine - DOI: 10.1109/MSP.2011.940269*, 28 Issue:3, 2011.