Escalabilidade e Paralelismo em Ambiente de Nuvem: Uma Solução para Processar Trâmites Judiciais

Emmanoel M. Sousa Junior, Frederico Lopes, Idalmis Milian

Instituto Metrópole Digital - Universidade Federal do Rio Grande do Norte
Natal - RN - Brasil

emmanoeljr@gmail.com, fred@imd.ufrn.br, idalmismilian@ect.ufrn.br

Abstract. *This paper proposes a scheduling mechanism for data processing in cloud computing environments. Such mechanism analyzes some specific variables in the business context of a company incubated at Metropole Digital Institute incubator, located at Federal University of Rio Grande do Norte. The main goal of this mechanism is to fulfill the sazonal demand using IaaS services and always considering two policies: (i) the maximum execution time allowed by the application may not be exceeded and (ii) the data have to be processed considering the lowest possible monetary cost. The proposed solution generates strategies to select the best set of virtual machines to process the current bunch of data. Such selection considers the amount of data, the estimated execution time for each specific strategy and the monetary cost of the virtual machines sets. In the context of this work, the strategy concept means the schedule of a set of virtual machines to process a specific amount of data, load balacing decisions and the paralelism of application's execution flow. The solution proposed in this work has resulted in great impact for that company since it allowed the vertiginous increase of the amount of clients served.*

Resumo. *Este artigo propõe um mecanismo de escalonamento para o processamento de dados em ambientes de computação em nuvem. Tal mecanismo analisa algumas variáveis específicas no contexto de negócios de uma empresa incubada no Instituto Metropole Digital incubadora, localizada na Universidade Federal do Rio Grande do Norte. O principal objectivo deste mecanismo é atender a demanda sazonal e altamente variável da empresa usando serviços de IaaS. O mecanismo deve sempre considerar duas políticas concomitantemente: (i) o tempo máximo de execução permitido pela aplicação não pode ser ultrapassado e (ii) os dados devem de ser processados considerando o menor custo monetário possível. A solução proposta gera estratégias para selecionar o melhor conjunto de máquinas virtuais para processar o grupo atual de dados. Tal seleção considera a quantidade de dados, o tempo de execução estimado para cada estratégia específica e o custo monetário dos conjuntos de máquinas virtuais. No contexto deste trabalho, o conceito estratégia significa a programação de um conjunto de máquinas virtuais para processar uma quantidade específica de dados, decisões de balanceamento de carga e o paralelismo de fluxo de execução da aplicação. A solução proposta resultou em grande impacto para referida empresa uma vez que permitiu o aumento vertiginoso da quantidade de clientes atendidos.*

1. Introdução

A computação em nuvem é uma visão conceitual que vem sendo utilizada nos últimos anos para descrever uma nova forma de se trabalhar a infraestrutura de TI, através da rede mundial de computadores, ou em ambientes internos das organizações, obedecendo critérios de alta disponibilidade, escalabilidade e elasticidade dos recursos computacionais para apoiar um modelo de consumo dos recursos de acordo com as demandas funcionais dos sistemas. Este consumo não envolve apenas recursos de

infraestrutura e armazenamento, mas também níveis de processamento de dados, serviços de *software* pré-configurados com pagamento de acordo com o consumo. Segundo Veras(2012), a visão dos principais serviços de um provedor de nuvem pode ser dividida nas seguintes camadas: Saas (*Software as a Service*) aplicações de interesse de uma grande quantidade de usuários passam a ser hospedadas na nuvem como uma alternativa ao ambiente local; Paas (*Plataform as a Service*) capacidade oferecida pelo provedor ao desenvolvedor de aplicação que serão executadas e disponibilizadas na nuvem e Iaas (*Infrastructure as a Service*) capacidade que o provedor tem de oferecer uma estrutura de processamento e armazenamento de forma transparente.

O NIST (*National Institute of Standards and Technology)* argumenta que a computação em nuvem é um paradigma em evolução, um modelo que possibilita acesso, de modo conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços (Mell and Grance 2009). Este contexto de TI surge da necessidade de trabalhos complexos, onde os usuários têm que realizar instalação, configuração e atualização de sistemas de *software* em uma arquitetura de solução mais inteligente, buscando uma melhor relação entre *hardware* e *software*.

O escalonamento de aplicações é outro aspecto importante no ambiente de nuvem. Selecionar, dentro de uma lógica controlável, os recursos necessários para a execução de determinado trabalho/tarefa, obedecendo a critérios de definição de ambiente como tempo e volume preestabelecidos. A escolha da melhor ou da mais apropriada estrutura de execução para o trabalho é determinada pelos algoritmos de escalonamento. Segundo TANEMBAUM (2008), dentro de uma visão de consumo de CPU, o escalonador é responsável por decidir qual trabalho será executado primeiro, caso haja vários trabalhos definidos para executar competindo pelo uso da CPU. Sendo também responsável por decidir qual é o processador mais adequado para executar cada tarefa, decidir qual é o intervalo de tempo que cada tarefa executará em cada processador e alocar os recursos necessários para cada tarefa; além de disparar a sua execução. Segundo Borges e Augusto (2011) é possível observar o comportamento do escalonador ao ver como a política utilizada afeta os recursos e os consumidores. Seguindo esta visão, este artigo mostra a necessidade de uma estratégia de escalonamento na nuvem, analisando o comportamento da aplicação no ambiente atual e como este atinge os clientes envolvidos. A nova estratégia deve estar baseada em um inteligenciamento da relação demanda x capacidade de processamento disponível no ambiente.

2. Descrição do problema

A empresa ATI *Service* LTDA, para a qual se desenvolveu este trabalho, tem sua estrutura de negócio relacionada a serviços que visam atender a provisão de informações jurídicas para advocacia em geral. Ela atua coletando processamento e gerindo informações relacionadas a processos judiciais de forma inteligente para disponibilizar aos interessados, através de uma plataforma de *software* em ambiente de nuvem. As informações enviadas variam em termos de volume de texto por registro, quantidade de registro por usuários responsável e quantidade de usuários responsáveis por clientes a cada dia. Estas informações são disponibilizadas até às 09 horas da manhã, devendo ser processadas e enviadas no formato do e-mail até às 10 horas do dia corrente, pois elas são fundamentais para tomada de decisão e trâmite de ações operacionais pelos clientes, relacionadas a prazos administrativos e judiciais. Segundo Maciel et al. (2005), o uso das novas tecnologias tem apoiado as ações do governo

eletrônico, em especial no eixo de serviços, onde o cidadão, advogados e gestores acessam processos judiciais via Internet.

A Tabela 1 apresenta os dados levantados e demonstram que a estrutura atual, operando com um volume de 120 de clientes, já não atende em alguns dias o requisito de processamento das informações para envio dos e-mails aos clientes, em relação ao contexto ideal que a empresa deseja atingir (abrangência nacional). No momento analisado, o sistema chega a um mínimo de dez vezes o volume atual de clientes, que é de no máximo 1 hora de processamento após a disponibilização, como pode ser visualizado do dia 20 ao dia 27 da amostragem.

Tabela 1. Amostragem de desempenho da rotina

Data	20/02/2015	23/02/2015	24/02/2015	25/02/2015	26/02/2015	27/02/2015	02/03/2015	03/03/2015
Registros	**909**	**913**	**1765**	**1432**	**1406**	**1529**	**1531**	**1108**
Hora inicio	09:00:00	09:00:00	09:00:00	09:00:00	09:00:00	09:00:00	09:00:00	09:00:00
Hora fim	11:41:02	10:38:25	10:53:50	10:45:30	10:05:50	10:05:27	09:52:22	09:44:38
Tempo	**02:41:02**	**01:38:25**	**01:53:50**	**01:45:30**	**01:05:50**	**01:05:27**	**00:52:22**	**00:44:38**

Esta amostragem do acompanhamento deixa explicita a necessidade de uma revisão da arquitetura do sistema. O objetivo de tal revisão é construir uma estrutura de operação do serviço que possa ser facilmente redimensionada em relação ao crescimento da demanda. Tal necessidade de reengenharia do sistema ficou ainda mais evidenciada quando, no dia 11/05/2015, a empresa vivenciou a situação em que, por um problema de captura das informações no sistema, o número de registros disponibilizados para envio aos clientes ultrapassou 9.000 para um único dia. Esse evento resultou em uma sobrecarga no processamento do servidor, provocando o atraso na entrega das mensagens aos clientes e a lentidão das aplicações que estão operantes no mesmo ambiente. Esta situação do dia 11/05/2015 demonstrou na prática que o sistema atual não estava estruturado para atender um nível de demanda tão alto. Um ponto positivo foi que, devido a estudos preliminares já realizados neste trabalho até a data do ocorrido, foi possível identificar o problema rapidamente e montar uma estratégia de ação imediata para minimizar o transtorno junto aos clientes.

3. Em busca de uma solução para o problema

Para achar um padrão mínimo de infraestrutura que possibilitasse o devido planejamento das atividades e um escalonamento controlado dos serviços, em relação a demanda na nuvem, foram utilizados os dados do universo apresentado na Tabela 1. Neste cenário, foi possível observar a quantidade de registros em relação ao tempo de operação para montar as mensagens a serem enviadas aos responsáveis de cada cliente. Para realizar as simulações necessárias, a empresa resolveu utilizar o ambiente de nuvem da *Amazon* (AMAZON WEB SERVICES), devido a mesma já possuir alguns serviços operando neste ambiente. A *Amazon* apresenta uma proposta de escalonamento de infraestrutura bem ágil e com custo de utilização por tempo ativo do serviço, o que facilita o controle do investimento. Neste momento, o principal foco foi buscar o entendimento dos fatores e pesos que podiam influenciar no resultado do serviço, segundo a visão de escalonamento no ambiente de nuvem. A configuração do ambiente selecionado é apresentada na Tabela 2.

Tabela 2. Dados de configuração do ambiente original do serviço

Modelo	Memória	Processamento	Sistema Operacional.	Serv. WEB	Linguagem
t2.*micro*	1 GiB	1 vCPUs, 2.5 GHz, Intel Xeon Family	Win 2003	IIS	PHP

Para atender as condições deste cenário e analisando a aplicação operando tanto, em termos das tecnologias utilizadas, como também em relação a versão destas tecnologias;

foi selecionado o sexto dia da amostragem da Tabela 1 que contem 1.529 registros como referência para a simulação. Observou-se que o sexto dia (27/02/2015) já estava no limite aproximado das necessidades de negócio da empresa, ou seja, com um tempo aproximado de uma hora, replicando em um desempenho similar ao quinto dia (26/02/2015) de operação do serviço com 1.406 registros. Para achar uma solução a este problema foram implementadas sequencialmente, quatro fases distintas e complementares apresentadas a continuação.

3.1 Fase 1 – Melhoria da Infraestrutura de *hardware.*

Com o ambiente de *software* equiparado e utilizando a mesma base de dados, foram executadas simulações do processamento dos dados para melhorar o perfil da máquina hospedeira e observar se o *software* teria um melhor desempenho. A partir do experimento realizado foram obtidos os seguintes resultados apresentados na Tabela 3.

Tabela 3. Desempenho do dia 27/04/2015 em ambiente de simulação

Registros	Mensagens	Modelo/Memória/ Processamento	Sist. Operacional	Serv. WEB	Linguagem	Data do teste	Tempo
1.529	842	t2.*micro* / 1 GiB / 1 vCPUs, 2.5 GHz, Intel Xeon Family	Win 2003	IIS	PHP	14/03/2015	1:04:59
1.529	842	t2.*small* / 2 GiB / 1 vCPUs, 2.5 GHz, Intel Xeon Family	Win 2003	IIS	PHP	14/03/2015	0:56:13
1.529	842	t2.*medium* / 4 GiB / 2 vCPUs, 2.5 GHz, Intel Xeon Family	Win 2003	IIS	PHP	15/03/2015	1:01:52

O desempenho da rotina original, sem alterações em sua estrutura, no ambiente de simulação, replicou de forma equiparada ao sexto dia de execução da rotina, utilizando o modelo t2.*micro* do serviço de nuvem da Amazon, que opera com 1Gb de memória e um núcleo de processamento. Quando utilizado o modelo t2.*small* contendo 2Gb de memória e mantendo um núcleo de processamento, ocorreu uma variação de quase 10 minutos no processamento da rotina. Entretanto, ocorreu uma variação menor de desempenho, cerca de três minutos, quando realizado o upgrade para t2.*medium* contendo dois núcleos de processamento e 4Gb de memória. A estrutura de processamento sequencial na qual a rotina está construída, demonstrou que o processamento segue todos os passos para todos os clientes, mesmo não existindo registros para serem enviados no dia para um determinado cliente. Isto sobrecarrega o processador da máquina, considerando que apresenta dois núcleos de processamento.

Uma verificação importante no sistema operacional foi durante a execução realizada com a máquina t2.*medium* (com dois núcleos de processamento), onde a rotina utilizou somente um núcleo deixando o outro ocioso ou livre. Este caso mostrou que a simples melhora do ambiente de *hardware* pode não influenciar no desempenho da rotina se a aplicação não estiver preparada, pois a mesma não consegue tirar o melhor proveito do potencial disponível no ambiente. Este cenário indicou como a aplicação poderia explorar o paralelismo do *hardware* para assim obter um melhor desempenho.

3.2 Fase 2 – Particionamento de dados para execução da rotina de forma paralela por grupo de clientes.

Durante esta fase foram definidas regras que deviam ser utilizadas na aplicação, em relação a estrutura da rotina e da demanda apresentada para o dia na empresa. Foram realizadas algumas simulações para descobrir as variáveis que teriam influência no desempenho e para obter uma proposta de reengenharia da arquitetura do *software.* O objetivo neste caso era processar as informações em cerca de 1 hora. Visto que o *software* trabalha em forma sequencial, foi realizada uma simulação da rotina para

observar como seria o desempenho da mesma com uma abordagem paralela simples. Para isto, foram divididos os clientes em 5 grupos, executando a rotina original sem alterações estruturais e sendo chamada em instâncias concorrentes na mesma máquina.

Nesta primeira solução, foi utilizado o comando *exec* do PHP, linguagem em que já estava desenvolvida a rotina, num arquivo .cmd para chamar as instâncias do *software.* O escalonamento fica a cargo das regras de processamento interno do próprio sistema operacional da máquina. O critério utilizado para estimar a distribuição em cinco grupos de dados, foi que o tempo atual deveria ser de no máximo 20% do tempo total, já que as cargas de dados estariam sendo tratadas em paralelismo de atividades, através das instâncias do *software* dividido em cinco processos. Na primeira simulação (cenário 1) como mostram as Tabelas 4 e 5, queria se verificar se o tamanho dos textos tratados no processamento teria algum peso relevante. Neste cenário todos os clientes são agrupados por tamanho dos registros em bytes, para entender se o volume de texto pode estar influenciando no processamento.

Tabela 4 – Grupos de distribuição de cliente do cenário 1

Grupo	Clientes	Registros	Tamanho (bytes)
A	1	1.241	3.741.276
B	5	164	712.493
C	12	66	486.535
D	16	58	120.538
E	86	0	0
Total	**120**	**1.529**	**5.060.842**

Neste primeiro cenário identificou-se que o tamanho dos registros em bytes não tem influência sobre o processo, visto que o Grupo E contendo clientes sem registros foi o que consumiu mais tempo. O fato que mereceu destaque foi a simples utilização de 5 grupos de processamento paralelo no modelo de máquina t2.*micro*, que apresentou uma elevação de desempenho de 50% , caindo o tempo de execução em 30 minutos o que representa um resultado considerável para a empresa. A avaliação deste cenário (Figura 1) aponta que o número de clientes tem uma influência considerável sobre o desempenho da rotina. Desta forma, decidiu-se realizar outra simulação, cenário 2, onde o número de clientes fosse constante e tratados somente clientes com registros a serem processados. Neste segundo cenário, apresentado nas Tabelas 6 e 7, são considerados grupos com quantidade de registros e clientes iguais.

Tabela 5 – Resultados de simulação do cenário 1

Registros / Modelo	T2.MICRO		T2.SMALL		T2.MEDIUM	
1.529	**Grupos**	**0:30:20**	**Grupos**	**0:26:03**	**Grupos**	**0:29:51**
1.241	A	0:10:26	A	0:10:15	A	0:08:56
164	B	0:15:15	B	0:14:49	B	0:15:33
66	C	0:07:21	C	0:06:59	C	0:08:52
58	D	0:09:55	D	0:09:58	D	0:11:41
0	E	0:30:20	E	0:26:03	E	0:29:51

Neste novo cenário de equalização das variáveis de número de clientes e número de registros, a simulação apresenta os mesmos modelos trabalhados na primeira fase.

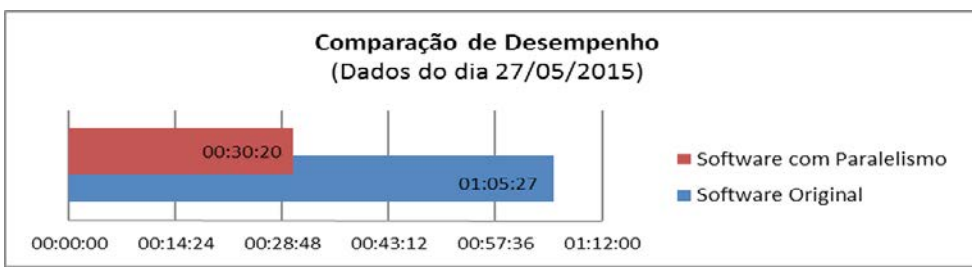

Figura 1 – Gráfico comparativo do primeiro cenário de simulação

Tabela 6 – Grupos de distribuição de cliente do cenário 2

Grupo	Clientes	Registros
A	5	1.598
B	5	1.598
C	5	1.598
D	5	1.598
Total	**20**	**6.392**

Tabela 7 – Resultados de Simulação do cenário 2

Registros / Modelo	t2.MICRO		t2.MEDIUM	
6.392	**Grupos**	**2:36:09**	**Grupos**	**0:22:53**
1.598	**A**	**1:53:28**	**A**	**0:10:51**
1.598	**B**	**1:28:33**	**B**	**0:08:38**
1.598	**C**	**2:36:09**	**C**	**0:22:53**
1.598	**D**	**1:51:47**	**D**	**0:11:19**

Demonstra-se aqui que a t2.*MICRO* sofreu o impacto da elevação do número de registros, extrapolando-se o tempo de 1 hora de execução em todos os grupos. Como a máquina possui somente um núcleo de processamento, tornou inviável a tentativa com o modelo t2.*SMALL*. Já com t2.*MEDIUM* o tempo de execução caiu de forma considerável, o que mostrou que a equalização do número de clientes e o número de registros podia ser utilizado como variável de parametrização para definir o modelo de infraestrutura de máquina empregada em um ambiente que comporte o paralelismo do processamento. Neste cenário foram identificados pontos que podiam ser trabalhados para evitar processamentos desnecessários. Este caso mostrou também que a estratégia de escalonamento, utilizando as variáveis de número de clientes e número de registros iguais, podia definir bem que tipo de infraestrutura de máquina devia ser selecionada de acordo com a demanda do dia, para assim executar o melhor resultado em termos de processamento.

3.3 Fase 3 – Reengenharia da estrutura do *software* buscando o melhor balanceamento das cargas de dados.

Esta fase apresenta uma proposta de reengenharia do *software* para se obter o melhor desempenho dentro do ambiente original. Foi utilizado balanceamento de carga durante o processamento, através das variáveis “número de clientes” e “número de registros”. A Figura 2 mostra o novo fluxo que foi proposto para o *software*.

Este novo fluxo apresentou três pontos fundamentais para a melhoria do desempenho do *software*: 1. Separação das empresas/clientes que possuem e que não possuem registros a serem enviados no dia. Desta forma, as empresas que não possuem registros para o dia não precisam percorrer o processamento de todos os seus usuários, sendo definido um modelo padrão de relatório a ser enviado para todos eles. 2. Separação das empresas/clientes que possuem registros para o dia em grupos de cliente com quantidade de registros equivalentes. Desta forma, é possível montar um paralelismo da rotina e diminuir assim o tempo de processamento geral como apresentado nas simulações realizadas anteriormente. 3. Dentro do processamento das mensagens de cada cliente, separar os usuários que podem ver todas as mensagens do cliente dos que só podem ver as que estão sobre sua responsabilidade. Desta forma, diminuí esforço de montagem de várias mensagens com grande número de registros, montando somente uma mensagem padrão e enviando para todos os usuários envolvidos em uma única mensagem. Para resumir as simulações realizadas no *software,* a Tabela 8 mostra a evolução até chegar na versão atual.

Tabela 8 – Resumo da evolução das versões do *software*

Versão	Descrição	Resultado
1	***Software* Original:** sem alterações na estrutura de execução.	Em alguns momentos não atende o requisito de 1 hora no tempo de processamento das informações para envio das mensagens.
1.5	***Software* com Paralelismo:** sem alterações na estrutura de execução, mas sendo executados em grupos subdivididos de clientes por número de registros, em formato paralelo, dentro da mesma máquina.	Extrema melhoria de desempenho, chegando a diminuir 50% no tempo do processamento do volume atual de informações.
2	***Software* Otimizado e com Paralelismo**: alterações estruturais que realizam o balanceamento de carga em número de clientes e número de registros, além de evitar esforços desnecessários no tratamento das informações.	Extrema melhoria de desempenho, chegando a diminuir 90% no tempo do processamento do volume atual de informações.

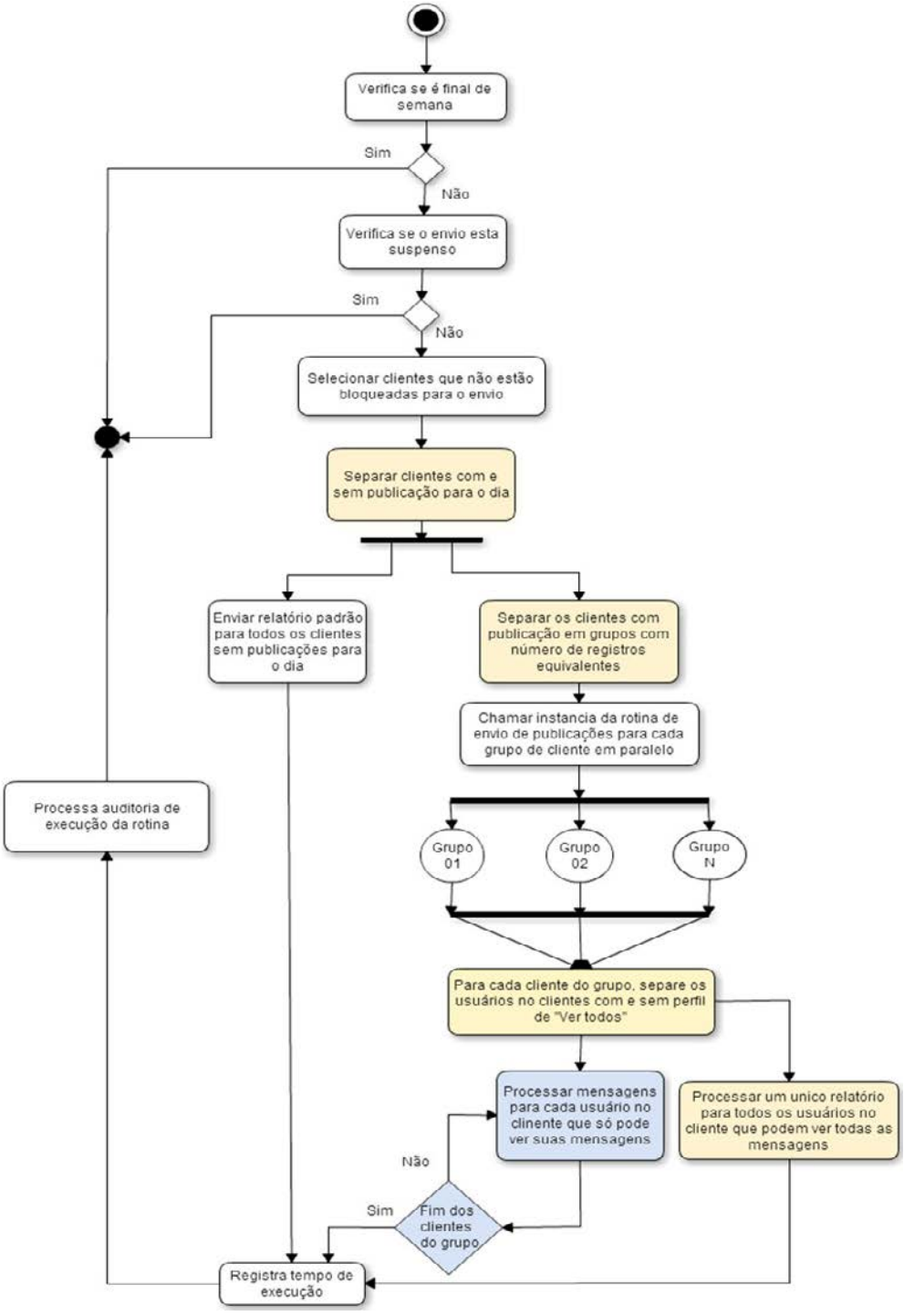

Figura 2 – Proposta de reconfiguração do *software.*

A comparação do desempenho da rotina pura com a rotina otimizada é apresentada na Figura 3. Nesta fase de testes, no ambiente onde o servidor não está dedicado totalmente, já apresenta os resultados com os dados da última semana do mês de maio. Foi concluído que, para o *software* obter resultados com maior exatidão em relação às variáveis que estão sendo trabalhadas, deve ser implantado em um servidor dedicado sem interferências de outras aplicações durante a sua execução.

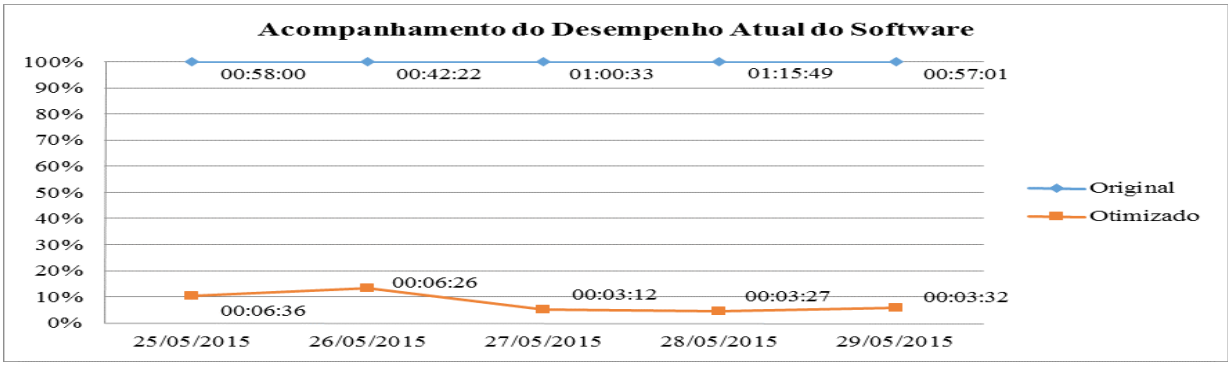

Figura 3 – Gráfico do acompanhamento de desempenho do *software* atual em relação ao *software* otimizado

Entretanto, já se obteve nesta fase um resultado extremamente importante para empresa, com um ganho real que pode atender com maior margem de segurança aos seus atuais clientes. O problema tratado até o momento tentou resolver a situação de otimização do *software* para uma realidade de demanda que pode ser atendida por uma máquina virtual com configuração pré-definida. Agora, e se o crescimento da demanda alcançar o limite de operação da máquina, mesmo com todas as otimizações realizadas no *software*? A resposta conclui a necessidade de uma estratégia de escalonamento *para o software* sobre a infraestrutura considerada, observando questões de memória e processamento.

3.4 Fase 4: Construção de uma ferramenta para o escalonamento da infraestrutura de nuvem de acordo com a demanda do dia.

A proposta nesta fase foi construir uma ferramenta, que em uma situação de demanda que supera o potencial da máquina padrão, realize uma seleção automática do modelo de máquina para a execução do *software* dentro dos modelos disponíveis da nuvem. Esta ferramenta deve ler as informações necessárias para que o *software* possa apontar qual o melhor modelo de máquina para processar os dados. Desta forma, o escalonamento do *software* deve atender os requisitos de desempenho estabelecidos pela empresa. O diagrama do algoritmo proposto para o escalonamento na nuvem é apresentado na Figura 4, com os passos que necessários para selecionar os recursos. A próxima seção descreve os testes que foram realizados e os resultados obtidos utilizando a estratégia de escalonamento proposta.

4. Ambiente de testes e resultados do escalonamento da aplicação na nuvem

A nuvem da *Amazon* possui uma lista com 39 modelos de máquinas virtuais disponíveis, com uma escala crescente de formatações em termos de memória, processamento e valor por tempo/hora de utilização. O custo da utilização por hora é proporcional ao potencial de processamento de cada modelo. Para as simulações de escalonamento realizadas, foi utilizada como referência os modelos trabalhados nas fases 1 e 2. Para identificar a potência de processamento de registros de cada tipo de máquina, foram realizados testes individuais com cada tipo de máquina, variando as cargas de registros (Tabelas 9 e 10). Com o objetivo de fazer uma simulação completa da arquitetura de

inteligência, para escalonamento e paralelização do processamento utilizando as máquinas no ambiente da Amazon proposto, construímos um cenário controlando o número de registros a serem processados e relatamos o comportamento de cada máquina virtual selecionada. Na construção deste cenário ampliamos o volume de registros para a equivalência de 40 dias, contemplando o período de 20/09/2015 à 30/10/2015.

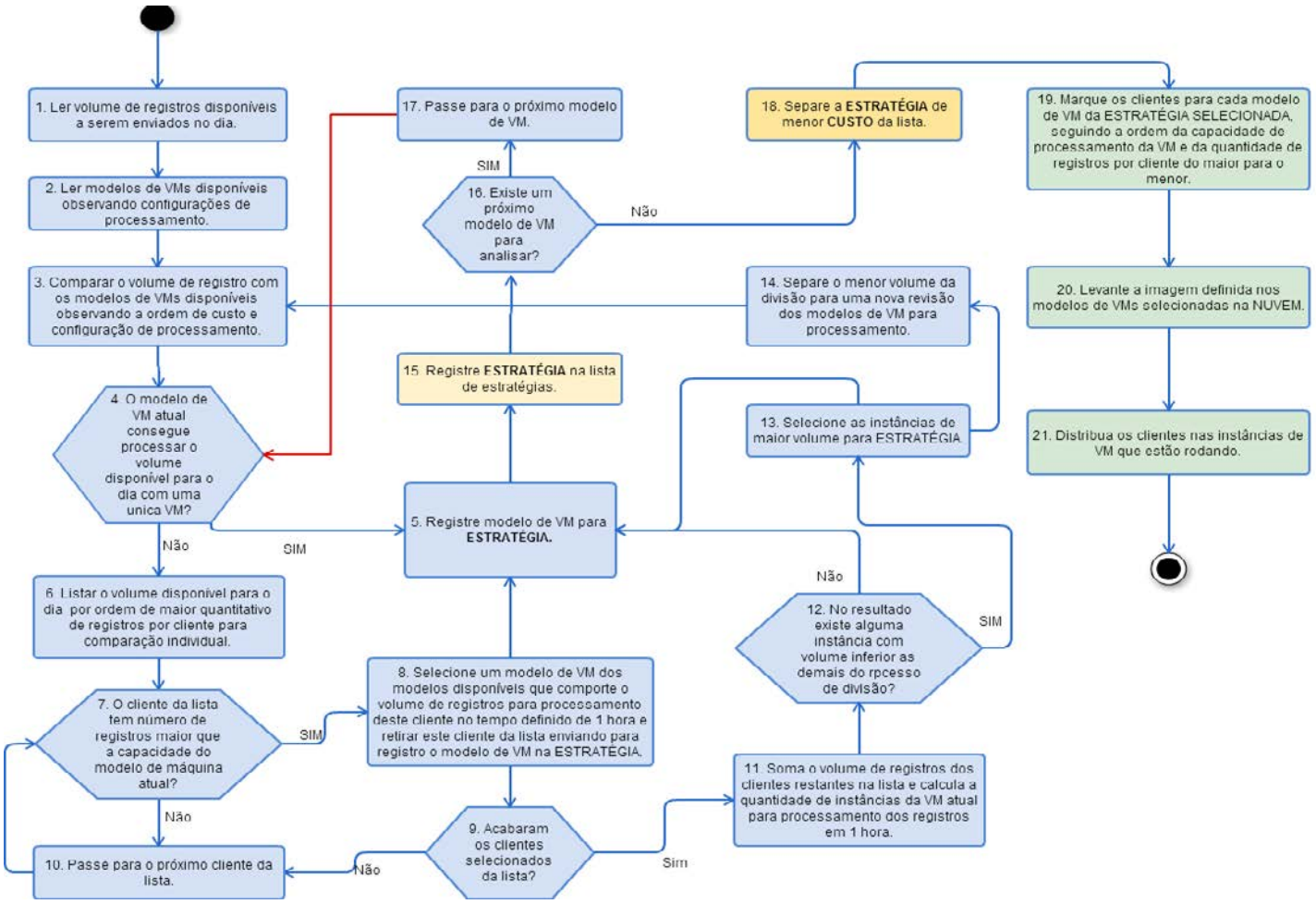

Figura 4 – Algoritmo de escalonamento.

Tabela 9 – Custos e capacidade de processamento por configuração de máquina virtual na *Amazon*.

MODELO	TIPO	CPU	MÉM.	CUSTO HORA (U$)	REGISTROS/HORA
1	t2.*micro*	1	1	0.018	10.000
2	t2.*small*	1	2	0.036	15.000
3	t2.*medium*	2	4	0.072	20.000
4	t2.*large*	2	8	0.134	60.000

Tabela 10 – Opções de estratégia para o cenário na *Amazon*.

Opção	Estratégia	Custo ($)
A	7 máquinas (t2.*micro*) com 10.000 registros e 1 maquina (t2.*small*) com 921 registros	0.162
B	5 máquinas (t2.*small*) com 15.000 registros e 1 maquina (t2.*micro*) com 8.921 registros	0.198
C	4 máquinas (t2.*midium*) com 20.000 registros e 1 maquina (t2.*micro*) com 3.921 registros	0.30
D	1 máquinas (t2.*large*) com 60.000 registros e 1 maquina (t2.*large*) com 23.921 registros	0.26
E	1 máquinas (m4.*large*) com 83.921 registros	0.25

Neste período foram identificados 136 clientes para o volume de registro a ser processado e o algoritmo de escalonamento gerou as seguintes opções para resolução da demanda. A Tabela 11 apresenta o resultado em que a opção de estratégia A tem o menor custo de processamento em relação a escala dos cenários seguintes. Também neste caso t2.*large*, na opção de estratégia D, não consegue processar toda a demanda em uma hora sozinha com custo maior do que o da estratégia A. A observação destes dois cenários iniciais apontam a política de custo da *Amazon*, em que os modelos de máquinas segue uma escala de dobrar o custo diante de cada mudança sequencial de configuração de modelo. Os cenários seguintes sempre serão mais caros que o cenário anterior, pois a capacidade de processamento da máquina não consegue acompanhar a mesma escala de progressão dentro dos cenários de ajustes realizados.

Tabela 11 – Resultado da estratégia selecionada para o cenário na *Amazon*.

Data	Clientes	Registros	Tempo Plan.	Custo Plan.	Custo Total
01/12/2015	136	83.921	60	((8 X $ 0,018) + $ 0,036)	**($) 0,18**
Máquina	**Hora estratégia**	**Início processo**	**Fim processo**	**Tempo real**	**Custo real**
t2.small	12:48:41	12:50:45	13:31:43	00:40:58	($) 0,024
t2.micro	12:48:42	12:50:39	13:21:38	00:30:59	($) 0,0093
t2.micro	12:48:47	12:50:45	13:34:28	00:43:43	($) 0,0129
t2.micro	12:48:50	12:50:41	13:27:13	00:36:32	($) 0,0108
t2.micro	12:48:43	12:50:39	13:36:13	00:45:34	($) 0,0135
t2.micro	12:49:22	12:50:46	13:55:17	01:04:31	($) 0,0192
t2.micro	12:48:45	12:50:37	13:02:17	00:11:40	($) 0,0036
t2.micro	12:48:48	12:50:39	14:05:17	01:14:05	($) 0,0222
t2.micro	12:48:49	12:50:49	14:05:06	**01:14:17**	($) 0,0222
			Custo real total:	**01:14:17**	**($) 0,1377**

Objetivando ter resultados diferenciados para comparação, foi construída uma tabela fictícia (Tabela 12), de um suposto provedor de nuvem em que a escala de custos não siga a mesma regra de valoração da *Amazon*. Nesta nova tabela mantivemos as nomenclaturas e capacidade de processamento da *Amazon*, só que ajustamos a escala de agregação de custos onde o valor seguinte seja 20% menor que o dobro do valor anterior. O cenário da nuvem fictícia tem a mesma ampliação do volume de registros para a equivalência de 40 dias demanda, contemplando o período de 20/09/2015 à 30/10/2015. Neste período foram identificados 136 clientes para o volume de registro a ser processado e o algoritmo de escalonamento as mesmas opções do cenário anterior da *Amazon*, seguindo os novos critérios de custos para seleção da estratégia mais viável.

Tabela 12 – Custos e capacidade de processamento por configuração de máquina virtual em uma nuvem fictícia.

Modelo	Tipo	CPU	Mémoria	Custo Hora (U$)	Registros/Hora
1	t2.micro	1	1	0.018	10.000
2	t2.small	1	2	0.028	15.000
3	t2.medium	2	4	0.044	20.000
4	t2.large	2	8	0.070	60.000

A execução do algoritmo de escalonamento para 83.921 registros neste novo contexto de nuvem fictícia, de custo por modelo de máquina, indica a opção de estratégia D como a mais viável, veja Tabela 13. Observe que apresenta custo menor que as opções anteriores e não encontrou nenhuma combinação posterior a disposição, que possa processar a demanda com menor custo de comparação. Apesar do algoritmo de

escalonamento ter gerado as estratégias de forma coerente em relação as questões de demanda e custo preestabelecidos nas regras, os resultados da nuvem com valores fictícios não puderam ser executados, devido a incompatibilidade da tecnologia hoje utilizada pela empresa. A mesma ainda trabalha com sistema operacional *Windows* 2003 *Server* R2 para este serviço; o qual não suporta a configuração de *hardware* superior a 4 Gb de memória e dois núcleos de processamento. Como a partir da t2.*large* a *Amazon* já utiliza 8 Gb de memória, a imagem que é utilizada como padrão, em ambiente *Windows* 2003 *Server* R2, não consegue subir as instâncias de máquinas virtuais para o processamento da demanda.

Tabela 13 – Opções de estratégia para o cenário 2 utilizando a nuvem fictícia.

Opção	Estratégia	Custo ($)
A	7 máquinas (**t2.*micro***) com 10.000 registros e 1 máquina (**t2.*small***) com 921 registros	0.154
B	5 máquinas (**t2.*small***) com 15.000 registros e 1 máquina (**t2.*micro***) com 8.921 registros	0.158
C	4 máquinas (**t2.*midium***) com 20.000 registros e 1 máquina (**t2.*micro***) com 3.921 registros	0.194
D	1 máquinas (**t2.*large***) com 60.000 registros e 1 máquina (**t2.*large***) com 23.921 registros	0.14

Esta situação reforça o contexto teórico tratado no início deste trabalho de que as empresas devem tratar suas aplicações de *software* observando o contexto da nuvem onde pretende operar seus serviços para que possam realmente tirar o melhor proveito dos potenciais de escalabilidade propostos. Segue-se então a recomendação que a empresa atualize a tecnologia de base de seu serviço para tirar melhor proveito dos provedores de nuvem disponíveis hoje no mercado.

5. Trabalhos relacionados

J. NAMJOSHI (2009) aborda a necessidade de empresas de viagens acessar a um grande número de redes de hotéis e serviços das companhias aéreas em todo o mundo, acessando suas tarifas mais recentes e informações de disponibilidade que irão ser atualizadas dinamicamente. As datas de viagem podem mudar a qualquer momento e os usuários podem precisar de ajustes rápidos em itinerários envolvendo passagens aéreas e reservas de hotel. Para atender a demanda, a ferramenta implementada no ambiente escalável da computação em nuvem distingue-se da visão típica da virtualização e destaca o poder que a provisão dos serviços na nuvem pode oferecer aos usuários. O sistema lida com aumento da demanda de clientes e fornecedores de viagens a medida que o negócio cresce, e busca a possibilidade de ajuste dinâmico do uso de infraestrutura de TI em relação a variação das necessidades do negócio. Os serviços de nuvem utilizados visam atender os requisitos de escalabilidade, disponibilidade e desempenho.

Já Lago (2012) apresenta um escalonador que leva em consideração a quantidade de energia necessária para executar determinada tarefa. Essas informações serão fornecidas ao escalonador EES (*Energy Efficient Scheduler)*, que escalonará o job para o *cluster* que compõem a grade que obtiver o menor gasto de energia (informação fornecida pelo mecanismo que calcula o consumo), utilizando um dos algoritmos de escalonamento disponíveis. O projeto apresenta uma solução de escalonamento em ambiente de nuvem buscando a otimização do tempo com foco na economia de energia das máquinas virtuais na nuvem. Foi desenvolvido um algoritmo de regra de escalonamento de prioridade permitindo a redução do tempo de processamento de cargas com alta prioridade, sem comprometer significativamente o consumo de energia. O objetivo foi a organização das cargas de maior prioridade para seleção das máquinas virtuais mais aptas para seu processamento, sem prejudicar criticamente as cargas com baixa prioridade e praticamente, sem aumento do consumo de energia no escalonamento.

6. Conclusões

As ações exploratórias que foram desenvolvidas nas diferentes fases, geraram uma sequência de resultados evolutivos mostrando que a arquitetura de sistemas em nuvem requer uma visão multidisciplinar e contextualizada da relação entre negócios, *software* e *hardware*. As organizações que pretendem operar seus serviços baseados em *software* no ambiente de nuvem, devem observar se suas aplicações estão preparadas para alcançar os reais benefícios disponíveis neste ambiente. Em relação ao processamento paralelo se observou que o simples fato da paralelização do *software* em grupos de clientes e número de registros, já gerou um ganho real para empresa em termos de desempenho no tempo de execução, mas se a escala de clientes/registros tiver um crescimento considerável, em que a estrutura de máquina padrão não der suporte, será necessário melhorar a infraestrutura da máquina. Uma grande vantagem de se trabalhar com o ambiente de nuvem é a facilidade e agilidade de escalonar a infraestrutura disponível em relação as regras de negócio que se presentarem de forma controlada. O algoritmo de escalonamento criado conseguiu definir a melhor estratégia para a demanda de registros e assim sugerir a infraestrutura virtual de forma automática, selecionando o modelo de máquina que atenda a demanda do momento. O desafio para empresa agora é seguir este modelo de arquitetura, atualizando as tecnologias do serviço atual e adequando os demais serviços de envio de informações automáticas a seus clientes, seguem os mesmos padrões da problemática tratada neste caso. Esta arquitetura de escalonamento pode evoluir para trabalhar o processamento paralelo de forma mais eficiente e também em nível de interoperabilidade entre nuvens, onde o algoritmo de escalonamento pode se utilizar da questão do valor ofertado pelos provedores do serviço de nuvem como parâmetro de seleção das máquinas virtuais a serem utilizadas.

7. Referências

AMAZON WEB SERVICES. < http://aws.amazon.com/> Acesso em: 08 de novembro de 2015.

BORGES, Carlos Augusto Lima. Escalonamento de tarefas em uma infraestrutura de computação em nuvem federada para aplicações em bioinformática. Brasília: UnB, 2011, 115p;

J. NAMJOSHI; A. GUPTE. Service oriented architecture for cloud based travel reservation Software as a Service. In IEEE CLOUD, pages 147-150, 2009;

LAGO, D. G., MADEIRA, E. R. M., and BITTENCOURT, L. F. (2012). Escalonamento com prioridade na alocação ciente de energia de máquinas virtuais em nuvens. In ´ Anais do XXX SBRC. <http://www.lrc.ic.unicamp.br/~bit/arquivos/SBRC2012escalonamento_com_Prioridade_na_Alocacao_Ciente_de_Energia_de_Maquinas_Virtuais_em_Nuvens.pdf> Acesso em 12 de outubro de 2015.

MACIEL, Cristiano; NOGUEIRA, José Luiz T.; GARCIA, Ana Cristina Bicharra. An x-ray of the brazilian e-gov web sites. In:**Human-Computer Interaction-INTERACT 2005**. Springer Berlin Heidelberg, 2005. p. 1138-1141.

MELL, P. and Grance, T. (2009). Draft NIST Working De-finition of Cloud Computing. National Institute of Standards and Technology. <http://csrc.nist.gov/groups/SNS/cloud-computing> Acesso em: 19 de dezembro de 2015.

SHAW, M., GARLAN, D. *Software* Architecture: perspectives on an Emerging Discipline. New Jersey: Prentice-Hall, 1996.TANEMBAUM, A.; WOODHULL, A. Sistemas operacionais, projeto e implementação. Tradução João Tortello. 3. ed. Porto Alegre: Bookman, 2008, 992p.

TURION, Cezar. Cloud Compunting. Computação em nuvem. Transformando o Mundo da Tecnologia da Informação. Rio de Janeiro: Brasport, 2009.

VAQUERO, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. 2008.

VERAS, Manoel. CLOUD COMPUTING: a nova Arquitetura da TI. Brasport - São Paulo, 2012.