

QUALIDADE DE SERVIÇO EM NUVENS COMPUTACIONAIS: UMA AVALIAÇÃO DE RECURSOS EM AMBIENTES DISTRIBUÍDOS

Fabio L. Licht¹, Bruno R. Schulze² e Luis Carlos E. de Bona³

¹*Departamento de Engenharia e Computação, Universidade Católica de Petrópolis, Petrópolis, 25685-070, RJ, Brasil*

²*Laboratório Nacional de Computação Científica, Petrópolis, 25651-075, RJ, Brasil*

³*Departamento de Informática, Universidade Federal do Paraná, Curitiba, 81531-980, PR, Brasil*

Palavras Chave: nuvem, gerência, telerância a falhas, previsão de falhas, monitoramento, escalabilidade, normalização, QoS.

Resumo. Esse trabalho propõe a criação de um sistema de gerência de recursos computacionais em nuvens e sistemas distribuídos, buscando qualidade de serviço, fornecendo um ambiente de alta disponibilidade com informações relevantes à tomada de decisão em sistemas tolerantes a falhas. A proposta se justifica dado o crescimento do uso da computação na nuvem, e que os recursos disponíveis nessas estruturas passaram a ser de difícil administração, normalmente por estarem geograficamente distribuídos e conseqüentemente prejudicando seu uso em sistemas de computação. O modelo prevê o uso de normalização de valores e criação de rotinas de diagnóstico para previsão de falhas, bem como monitoramento constante dos recursos sujeitos a degradação por meio de uso, falhas ou tentativa de ataque DoS, por exemplo.

*E-mail addresses: fabio.licht@ucp.br, schulze@lncc.br, bona@inf.ufpr.br .

1 INTRODUÇÃO

Dos grandes desafios por que passa a computação em nuvem [3] está a garantia de disponibilidade para seus usuários e o provimento de informações relevantes à aquisição de novos recursos. Casos como o ocorrido no site do Sistema de Seleção Unificada (SiSU) do Ministério da Educação (MEC) em 16/01/2011, quando diversos alunos perderam a inscrição para prova unificada que daria direito à entrada em universidades públicas pela nota desta prova[4]. Ou ainda quando o site da Polícia Federal ficou fora do ar em 19/09/2011, causando impossibilidade de emissão de passaportes[12]. Problemas também ocorrem com frequência no site da receita federal quando dos últimos dias da entrega das declarações do imposto de renda, o site não suporta o número excessivo de conexões, trazendo transtornos e prejuízos ao governo e às pessoas que precisam destes serviços online.

O caso mais grave relacionado a Nuvem foi o ocorrido com a Amazon[14], em abril de 2011, quando um equívoco feito pelos seus técnicos na configuração do sistema, fez com que o fluxo de dados principal foi direcionado para a rede secundária de baixa capacidade e não para a rede e servidor de cópia de segurança primário. Isto causou um efeito em cascata, que comprometeu todo o sistema, causando o que a Amazon chamou de “re-mirroring storm” ocasionando um estado de espera. Na zona afetada os servidores enviavam requisições e fora desta os servidores também enviavam requisições. O resultado foi que 1,4% dos volumes ficaram indisponíveis sendo que deste total a Amazon recuperou a metade (0,7%), os outro 0,7% ficaram perdidos para sempre.

Outro grande caso foi o roubo informações de 77 milhões de usuários na Play Station Network[6]. Uma invasão que levou cerca de 7 dias para ser descoberta, causada pela capacidade do invasor explorar falhas existentes.

Estes fatos registrados na mídia trazem a tona a necessidade de prover um ambiente funcional, confiável e seguro usando o conceito de dependabilidade (do termo em inglês Dependability). Dependabilidade é uma propriedade dos sistemas, composta por atributos como confiabilidade, disponibilidade, segurança física, segurança lógica, confidencialidade, integridade, munutenibilidade e capacidade de sobrevivência. Laprie[8] divide o seu conceito em três partes: Ameaças, Atributos - acima listados - e Meios de obtê-la. Quanto a este último ele é o foco deste trabalho, no que tange o auxílio à tomada de decisão de prevenção, tolerância, previsão e correção de falhas, através do uso de monitoramento contínuo de recursos. Esta entretanto é uma tarefa complexa e que requer alto grau de abstração dada a quantidade de recursos totalmente heterogêneos disponíveis nesse tipo de ambiente computacional.

A proposta deste trabalho foca, nesse sentido, na criação de uma infraestrutura de gerência de recursos distribuídos heterogêneos, em ambientes de nuvem, por meio da seleção e designação de recursos mais confiáveis e adequados a execução dos serviços a eles atribuídos, afim de reduzir a possibilidade de “falta” dos mesmos, obtendo isto através de informações disponibilizadas pelos recursos físicos disponíveis.

Supondo a existência de um algoritmo de engenharia, por exemplo, em fase de teste, onde não se sabe ao certo se a aplicação resolverá o problema, entrará em loop, ou consumirá todo o recurso disponível. Com um monitoramento contínuo, poderia-se avaliar a aplicação e se ela está funcionando sobre uma condição ideal.

Ainda sobre monitoramento, a definição de um mecanismo de controle de aplicação, para saber se um recurso suporta a execução de uma tarefa pode ajudar na aquisição de hardware, prevendo o tipo de recurso ideal para determinada aplicação científica ou comercial, evitando o alto dispêndio financeiro com recursos que não atenderiam às necessidades da aplicação.

Monitorar *job's* sendo executados, entretanto, não é simples. Dror Feitelson e Larry Rudolph [2] demonstram o trabalho de controlar estas tarefas sendo executadas em um ambiente de programação paralelo, calculando o custo da aplicação, mas não mostram a degradação de determinados recursos causados pela aplicação, por exemplo.

2 TECNOLOGIAS ENVOLVIDAS

2.1 COMPUTAÇÃO NA NUVEM

Computação em nuvem [13] [3], do inglês *Cloud Computing* é a tecnologia que tem como objetivo proporcionar serviços de computação sob demanda, com pagamento de acordo com o uso. Pode-se dizer que os precursores da computação na nuvem foram os Clusters¹ e Grades Computacionais².

A diferença entre a nuvem e seus antecessores é que nesta os serviços não estão vinculados a um grupo específico de usuários, nem tem que estar sob o mesmo domínio administrativo. Qualquer computador, tablet ou celular, pode fazer parte da nuvem e interagir com ela. E como estes recursos normalmente são compostos de centenas ou milhares de máquinas, físicas ou virtuais de baixo custo, o crescimento deste tipo de infraestrutura se torna mais fácil.

A computação na nuvem surge de uma necessidade de aumentar os recursos computacionais, sem ter que necessariamente saber onde estes estão, ou mesmo ter que instalar sistemas operacionais, atualizações ou softwares para que funcione. Basta que o serviço seja disponibilizado para dar acesso aos recursos deste e por ser um ambiente heterogêneo, basta ter mais recursos disponíveis que estes podem ser agregados aos já existentes, aumentando a capacidade de processamento ou armazenamento rapidamente.

Uma das propostas da nuvem é que em um curto espaço de tempo, ninguém precise mais ter que montar infraestruturas de hardware e software para testar aplicações ou serviços sobre demanda, que possuam uma curta atividade ou mesmo para verificar a viabilidade do uso. Ao invés disso, aluga-se o serviço disponível, pelo tempo necessário a um custo menor em relação à compra dos mesmos recursos que poderiam não atender às expectativas. Na falta de recurso, por estouro de sua capacidade, acrescenta-se mais recursos aos já disponíveis.

A proposta de ser global e prover serviços para qualquer usuário, que vão desde uma hospedagem de documentos pessoais à terceirização de toda uma infraestrutura de TI, permitindo que quando a capacidade de armazenamento seja degradada por excesso de informação ou danificação dos recursos de armazenamento, basta que se localize a replicação dos dados ou aumente os recursos disponíveis.

2.2 MODELOS DE SERVIÇOS DA NUVEM

Segundo o NIST [1] os modelos de serviços são:

- **Software como serviço (*Software as a Service, SaaS*):** O usuário utiliza aplicações do provedor que rodam em uma infraestrutura de nuvem. Tais aplicações podem ser acessadas a partir de vários dispositivos, através de uma interface como o browser (por exemplo, e-mail baseado na web). O consumidor não gerencia ou controla a infraestrutura de nuvem subjacente.

¹Diversos computadores homogêneos (SOs, hardware, software, ...) interligados, sendo acessados como se fossem apenas um computador.

²Diversos computadores interligados em um ambiente totalmente heterogêneo e normalmente geograficamente distribuído.

Plataforma como serviço (*Platform as a Service, PaaS*): Nessa modalidade, o usuário pode implementar na infraestrutura de nuvem aplicações próprias ou compradas (que controla), desenvolvidas em linguagens de programação e ferramentas suportadas pelo provedor. Mas o usuário não gerencia ou controla a infraestrutura de nuvem envolvida.

Infraestrutura como serviço (*Infrastructure as a Service, IaaS*): Nesse modelo, a oferta do provedor inclui processamento, armazenagem, redes e outros recursos computacionais fundamentais com os quais o usuário possa usar e rodar quaisquer softwares, sejam eles sistemas operacionais ou aplicativos. O usuário não gerencia nem controla a infraestrutura de nuvem envolvida, mas controla sistemas operacionais, armazenagem, aplicações implementadas e, possivelmente, componentes selecionados de rede como host firewalls.

2.3 MODELOS DE NUVEM

O NIST [1] apresenta 4 modelos básicos de nuvem a saber:

- **Privada:** A infraestrutura da nuvem é operada exclusivamente por uma empresa/ organização. Ela pode ser gerenciada pelo usuário, ou por terceiros, e pode existir dentro ou fora da empresa/organização.
- **Comunitária:** A infraestrutura da nuvem é compartilhada por diversas entidades e suporta uma comunidade específica que tem objetivos comuns. Pode ser gerenciada pelas organizações ou por terceiros, ser interna ou externa.
- **Pública:** Sua infraestrutura está disponível para o público, ou para um grande grupo empresarial, e sua propriedade pertence ao provedor de serviços em nuvem.
- **Híbrida:** Tem a infraestrutura formada por duas ou mais nuvens (privada, comunitária ou pública) que mantêm suas próprias identidades, mas operam em conjunto através de uma tecnologia padrão ou proprietária que permite portabilidade de dados e aplicativos, como a suspensão da nuvem para balanceamento entre nuvens.

2.4 COLETA DE RECURSOS

Coleta de recursos no contexto desse trabalho é o monitoramento de todo tipo de dados referentes à avaliação da qualidade e disponibilidade dos computadores que disponibilizam serviços na nuvem. Dados como memória, processador, disco e processos podem ser o diferencial quando compara-se a “qualidade” de cada computador. Informações como tipo, velocidade, quantidade total, em uso e livre podem caracterizar e pontuar cada host dentro de uma rede, nesse caso, na nuvem.

Coletar de recursos não é uma tarefa simples e requer acesso à informações de hardware, tarefa complexa, principalmente na linguagem de programação Java, escolhida para esse trabalho, mas que não oferece acesso ao hardware diretamente por estar sendo executado em uma máquina virtual, a JVM. Para conseguir coletar os recursos, de extrema importância para esse trabalho, serão utilizadas consultas ao sistema operacional e a scripts coletores, que retornarão à JVM as informações necessárias.

3 TESTE DE PERFORMANCE

Para que a proposta do trabalho aqui exposto seja viável, é importante que haja um modelo de teste de performance que adeque cada recurso ao seu uso e mantenha a pontuação destes recursos baseada em critérios reais, por exemplo, o uso de teste de performance ou teste de desempenho é fundamental.

O teste de performance é um tipo de teste que tende a determinar a confiabilidade, a capacidade de resposta de um recurso físico, o throughput, a confiabilidade de um recurso ou a escalabilidade de um host, sob uma determinada carga efetuada sobre cada um dos recursos disponíveis, como por exemplo, o cálculo do tempo de escrita e leitura de um arquivo no disco, ou o processamento de uma equação matemática.

As principais funções que serão implementadas neste trabalho, para avaliar o teste de performance são:

1. Avaliar a capacidade do recurso.
2. Definir os critérios de desempenho.
3. Comparar as características de desempenho de todos os hosts monitorados.
4. Ajustar os sistemas.
5. Pontuar os recursos de acordo com a disponibilidade, capacidade e tolerância a falhas.

3.1 OBJETIVOS DO TESTE DE PERFORMANCE

O teste de performance será realizado para identificar gargalos e falhas no sistema, em cada nó do ambiente de nuvem. O principal objetivo é o de estabelecer uma base de dados e informações para testes futuros, afim de averiguar a necessidade de ajustes de desempenho ou determinar o cumprimento das metas de desempenho e exigências, ou ainda coletar outros dados relacionados ao desempenho dos recursos, para a tomada de decisões quanto a qualidade dos recursos disponíveis. Outro ponto que poderá ser administrado são que, através dos resultados dos testes de desempenho, pode-se estimar a configuração de hardware mínima necessária para execução de determinada aplicação.

4 TRABALHOS RELACIONADOS

Nessa seção são apresentados alguns trabalhos que possuem algum grau de similaridade com o trabalho proposto, buscando principalmente os pontos fracos das aplicações em comparação com este trabalho.

4.1 PATTERNS FOR HIGH AVAILABILITY, SCALABILITY, AND COMPUTING POWER WITH WINDOWS AZURE

A proposta da Microsoft resume-se em definir padrões e práticas para o uso da computação em nuvem, mas em um ambiente proprietário, não privilegiando a proposta de software livre. Outro ponto é a necessidade de uso de outras ferramentas Microsoft para o uso da plataforma Azure[10]. Fazendo desta uma solução inviável para o projeto aqui proposto.

4.2 ABORDAGENS PRÁTICAS À ALTA DISPONIBILIDADE COM BASE NA NUVEM

M. Tim Jones [7] define abordagens com o uso de virtualização para prover alta disponibilidade em ambientes de nuvem, mas não aborda diretamente sistemas de gerência deste tipo de recursos. Ele aborda ainda o uso de ferramentas e tipos de aplicações em ambientes práticos, mas não propõe uma solução ao uso de gerência dinâmica dos recursos disponíveis.

5 DESCRIÇÃO DA PROPOSTA

A proposta deste trabalho é a de apresentar um modelo de monitoramento de recursos que pode ser utilizado para prover disponibilidade em ambientes distribuídos, além de disponibilizar dados para que a aquisição de hardware possa ser feita de maneira mais precisa. A Figura 1 apresenta o modelo da arquitetura básica proposta por essa aplicação, enquanto que a Figura 2 apresenta a arquitetura detalhada do sistema em uma nuvem computacional.

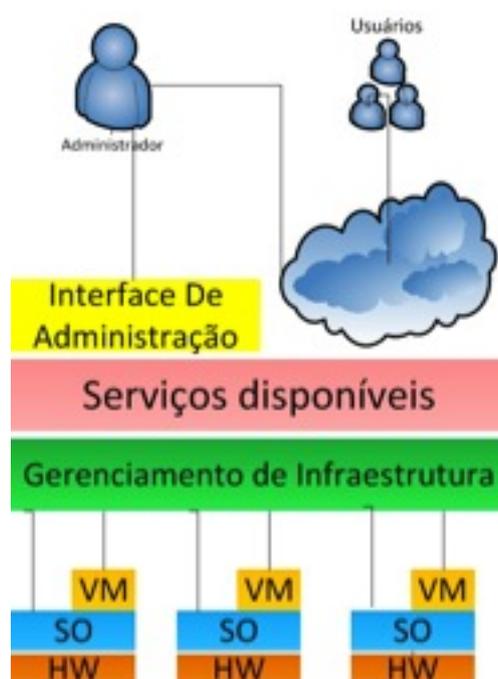


Figura 1: Arquitetura da Aplicação.

Um exemplo de monitoramento seria o de controle de uma aplicação executando em um recurso físico. Poder-se-ia verificar a degradação do recurso pela aplicação em questão e nesse sentido, averiguar se o hardware é suficiente.

Todo o tráfego de rede será implementado usando sistemas de transmissão seguros como descritos por esse autor em [9] com o uso de infraestrutura de chaves públicas auto-assinadas para criptografia do canal de dados, de forma a prover integridade e confidencialidade dos dados transmitidos na rede.

A modelagem de disponibilidade segue a seguinte fórmula:

$$A = \frac{MTBF}{MTBF + MTTR} \quad (1)$$

Onde A é a probabilidade de um sistema estar operando em determinado período de tempo,

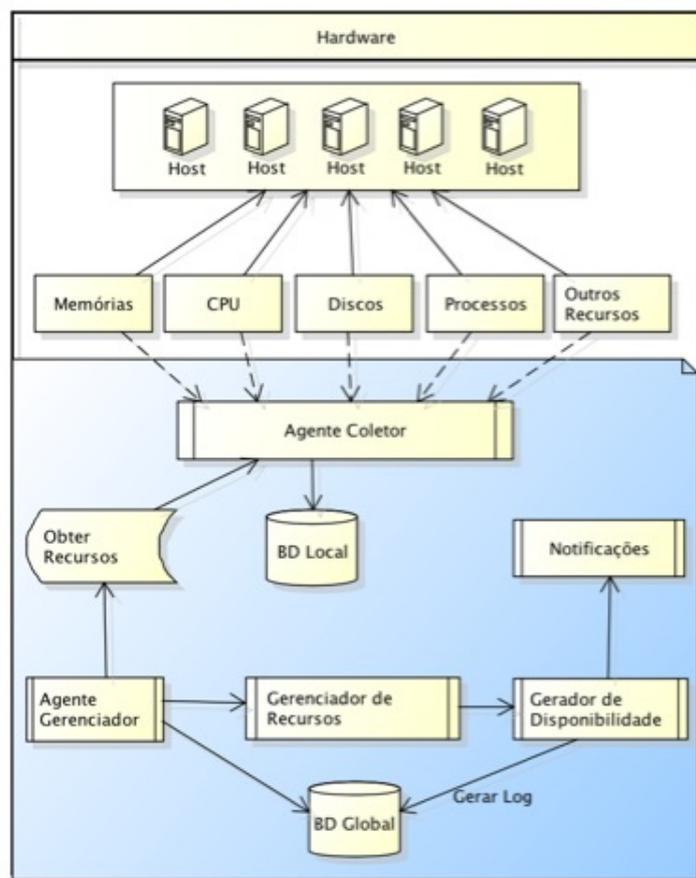


Figura 2: Arquitetura Detalhada.

obtida através do tempo médio entre falhas (MTBF) sobre este valor mais o tempo médio para reparos (MTTR).

Assim, $HA = f(MTBF, MTTR)$. Ou seja, $HA = 1$ se $MTTR = 0$. Para aumentar HA , diminui-se $MTTR$ e/ou aumenta-se $MTBF$.

Neste trabalho a proposta é alterar o $MTTR$, fazendo este tender a zero.

5.1 RECURSOS

Como descrito em 2.4, os recursos coletados pela aplicação são a principal tarefa do sistema proposto, entretanto outro ponto a ser tratado é a diferença de grandeza das medidas de desempenho. Não pode-se comparar os dados de processamento como quantidade de cálculos por segundo, geralmente calculados em FLOPS/s, como pode ser visto na Tabela 5.1, com dados de disco, normalmente da ordem de TB (terabytes) ou PT (petabytes), ou ainda comparar com a quantidade de memória de grandeza bem inferior, da ordem de GB (gigabytes).

5.2 NORMALIZAÇÃO DOS DADOS

Afim de pontuar os recursos coletados em um nível onde os valores poderão ser comparados, usou-se a Fórmula 2.

$$R = \frac{C - M2}{M1 - M2}(N1 - N2) + N2 \quad (2)$$

Tabela 1: Tabela da Ordem de Grandeza da Quantidade de Cálculos por Segundo de um Processador.

Ordem de Grandeza	FLOPS
megaflop/s	10.6
gigaflop/s	10.9
teraflop/s	10.12
petaflop/s	10.15
exaflop/s	10.18
zettaflop/s	10.21
yottaflop/s	10.24

Onde:

R : é o valor normalizado de cada entrada.

C : é o valor bruto coletado, sem tratamento.

M1 : é o maior valor encontrado em todas as coletas de todas as máquinas monitoradas.

M2 : é o menor valor encontrado em todas as coletas de todas as máquinas monitoradas.

N1 : é o maior valor da escala que se deseja trabalhar.

N2 : é o menor valor da escala que se deseja trabalhar.

A maior vantagem do uso da Fórmula 2 é poder elevar ou reduzir a pontuação de recursos de acordo com os recursos disponíveis em cada medição e desta forma, já que existe uma comparação entre todos os hosts da nuvem monitorada, a que tiver a maior pontuação, é consequentemente a que tem a melhor qualidade. Repara-se que o valor tratado de cada recurso poderá ser somado, já que após o uso da fórmula estarão na mesma ordem de grandeza. Nesse trabalho, considerar-se-á que os valores terão escala entre 1 e 10, sendo 1 para pior recurso e 10 para melhor.

Outro ponto é que um host pode ter uma melhor qualidade de processamento, mas com baixa qualidade de memória ou disco, e, neste caso, o host perderá no ranking para outras máquinas da nuvem. Quando for avaliada a qualidade, poderá prever-se que um recurso, que possuía anteriormente uma pontuação elevada, se essa pontuação está se degradando, ou novos recursos melhores foram acrescentados, viável de ser averiguado por geração de relatórios, ou o recurso está se degradando, e nesse caso, existe uma probabilidade de que esse host venha a paralisar os serviços que nele estão sendo executados. E antes que isso ocorra, será possível, com a previsão, re-escalonar os processos do recurso degradado, ou ainda acrescentar mais recursos aos já existentes, chamando isso de *Self-Optimization*.

$$P = \frac{\sum_{k=1}^r x_k}{r} \quad (3)$$

Onde:

P é a pontuação obtida por cada host.

r é a quantidade de recursos monitorados.

x_k é cada recurso já normalizado.

É possível comparar as Tabelas 5.2 e 5.2 onde os mesmos hosts estão com os dados coletados, não tratados, na Tabela 5.2 e já normalizados e somados na Tabela 5.2. Com o resultado da soma dos dados, pode-se definir que um recurso tem melhor qualidade pontual que outro, utilizando somente os valores de disco, memória e bogomips do processador, entretanto a proposta

Tabela 2: Tabela de Dados Reais de um Cluster

Host	Memória MB	Bogomips	Núcleos	Disco GB
192.168.235.1	23019	5333,58	12	107
192.168.235.2	23560	5333,51	24	411
192.168.234.1	19626	4489,39	16	816
192.168.234.2	23639	4533,44	8	121
192.168.234.3	4320	4606,08	16	520
192.168.234.4	11716	4533,47	8	817
192.168.234.5	11708	4533,47	8	616
192.168.234.6	11704	4533,46	8	358
192.168.234.7	11821	4532,68	8	746
192.168.234.201	7094	6227,75	8	742
192.168.234.205	4327	3732,8	2	709

Tabela 3: Tabela de Dados Normalizados

Host	Memória Normal.	Bogomips Normal.	Disco Normal.	Pontuação
192.168.235.1	9,711165174	5,22135375	1	5,310839641
192.168.235.2	9,963196853	10	4,853521127	8,272239327
192.168.234.1	8,130493297	5,805776471	9,987323944	7,974531237
192.168.234.2	10	3,150491162	1,177464789	4,775985317
192.168.234.3	1	5,945178741	6,235211268	4,393463336
192.168.234.4	4,445519954	3,150509082	10	5,865343012
192.168.234.5	4,441793053	3,150509082	7,452112676	5,014804937
192.168.234.6	4,439929603	3,150503108	4,181690141	3,924040951
192.168.234.7	4,49443553	3,1500372	9,1	5,58149091
192.168.234.201	2,292302914	4,162534437	9,049295775	5,168044375
192.168.234.205	1,003261038	1	8,630985915	3,544748985

é estender essa pontuação à quantidade de cálculos por segundo, frequência do processador, da memória e da placa-mãe, quantidade de núcleos, cache e qualquer outra informação relevante à coleta de desempenho dos recursos na nuvem. Ao final, haverá uma pontuação dada pela Fórmula 3.

Com os valores normalizados, serão averiguados continuamente as mudanças de status de cada monitoramento, e em caso de degradação de algum recurso, avaliado pelo próprio host, um alerta é emitido ao gerenciador global e este se encarrega de agregar mais recursos aos já existentes no ambiente.

Para um monitoramento mais preciso, durante um tempo de 3 meses os recursos serão submetidos à exaustão, submetidos à testes de estouro de memória, “fork bombs”³, enchimento do disco e desconexão da rede. Um “mapa” deste perfil mostrará que é possível prever falhas em recursos de software e hardware com o monitoramento contínuo.

Uma vantagem desta abordagem é que não é necessário consumir recursos da rede para troca de informações entre cada host e o gerenciador global continuamente. Cada host que adere à nuvem envia todas as informações para o gerenciador global e mantém uma cópia das informações em uma base de dados embarcada localmente no próprio host, em caso de percepção de mudança dos valores anteriormente gravados no banco de dados, o host envia as variações para o gerenciador global que atualiza o monitoramento e em caso de necessidade, administra o acréscimo de outros hosts.

Durante o monitoramento de recursos remotos, um ou mais hosts podem perder conectividade e deixar de enviar informações ao gerenciador global que poderia interpretar que o host estaria em funcionamento, sem que o mesmo esteja conectado. Afim de contornar tal problema, os hosts que já estão cadastrados no servidor (gerenciador global) são submetidos à “pings” contínuos afim de averiguar se o serviço está respondendo a requisições. A vantagem do uso destes “pings” em contrapartida ao monitoramento é que este consome muito menos banda da rede e atende à necessidade.

É natural que em alguns tipos de aplicação haja uma necessidade de um tipo específico de recurso, por exemplo, mais memória para uma aplicação que demanda pouca quantidade de disco, ou aplicações que usam mais disco e menor quantidade de processamento. No primeiro caso, pontuar a memória (frequência, capacidade, ...) com um peso maior seria mais adequado, enquanto que no segundo, um peso maior para disco (capacidade, quantidade de cache, ...) é mais importante. A Fórmula 4 apresenta o modelo de pontuação baseada em pesos.

$$P = \frac{\sum_{k=1}^r x_k x^p}{\sum_{k=1}^r p} \quad (4)$$

Onde:

P é a pontuação obtida por cada host com os pesos de cada recurso.

r é a quantidade de recursos monitorados.

x_k é cada recurso já normalizado.

p é o peso dado a cada recurso.

Um dos problemas que ocorre quando é necessário fazer o monitoramento de recursos é que não se pode verificar todas as variações no consumo dos mesmos. Por exemplo, quando uma execução de um processo é feita, esse processo pode consumir todo o recurso disponível em um servidor, entretanto o monitoramento pode não detectar esse pico de consumo por ter acontecido

³ Chamadas recursivas de processos até que o processador não consiga mais responder a todas as requisições

no intervalo entre amostras.

Não é possível coletar amostras a cada um segundo em um servidor, pois isso acarretaria a degradação do recurso disponível com o próprio monitoramento, mas uma coleta a cada 5 minutos, por exemplo, pode não ser ideal para verificar o efeito de um processamento de um código. Para que esse problema seja sanado, a proposta do trabalho é que o monitoramento aconteça de 5 em 5 minutos e quando da necessidade de monitoramento em intervalos menores, o usuário, através do sistema web, poderá solicitar que amostras sejam coletadas em intervalos de tempo menores e assim terá a estatística de degradação de todo o sistema. Outro ponto é que ainda será possível especificar qual os recursos se deseja monitorar: disco, processador, memória, ou outro que estiver disponível nos módulos de coleta.

Todo esse processo de coleta de recursos pretende armazenar no host remoto os efeitos e estatísticas relacionados à coleta de recurso. Também será preciso que essa coleta seja enviada ao “servidor global” responsável pela apresentação dos resultados ao usuário e possível agragação de novos recursos. Afim de que os clientes não fiquem sobrecarregados com arquivos XML, esses deverão ser enviados ao servidor e resetados na origem. É importante que isso seja feito em tempos aleatórios para evitar a sobrecarga da rede do “servidor global”, bem como dos “servidores locais”.

Serão ainda tratados 3 tipos de situação com respeito ao monitoramento de recursos definidas em rotinas de verificação do sistema da seguinte maneira:

1. **Rotinas de nível 1** - Uma rotina mais completa, que verificaria todos os itens do sistema e todas as informações de hardware, como modelo do processador, memória, cache, etc. Essas informações não alteram, a menos que o hardware seja atualizado, e desta forma não há necessidade de ficar comprometendo os recursos com informações deste tipo. Além do que, nesse tipo de verificação, há a necessidade de coletar todas as informações de recursos afim de gerar uma estatística mais completa, que pode ser usada para aferir problemas mais críticos da infraestrutura como um todo.
2. **Rotina de nível 2** - Uma rotina de nível intermediário onde são verificados alguns itens críticos como uso do disco e espaço livre no diretório temporário, por exemplo, (que poderia causar paralização do sistema operacional) outros monitoramentos seriam aqueles considerados importantes para o funcionamento mínimo de um sistema, como, por exemplo, os processos que estão sendo executados e o consumo de recursos destes no ambiente. Um processo como um “forkbomb”⁴[11], uma bomba lógica[5], ou mesmo uma implementação de um loop infinito, poderiam ser detectados antes que corrompessem o sistema.
3. **Rotina de nível 3** - Uma rotina que faria uma varredura dos itens necessários ao funcionamento dos sistema de forma mais sistemática: consumo de memória total, carga do processador, disponibilidade de rede e outras que se fizerem necessárias. É uma abordagem mais simples, que pode ser executada periodicamente em intervalos pequenos para averiguar possíveis anomalias intermitentes.

O principal motivo de dividir o monitoramento em 3 rotinas é o problema que esta abordagem afere, onde, durante a verificação de itens críticos, de nível 1, por exemplo, muitas vezes é

⁴criação de diversos processos paralelos até que todo o recurso (processamento, memória, ou disco) sejam consumidos

necessário interromper o serviço para verificá-lo, daí a motivação para que essas rotinas não sejam executadas constantemente.

Percebe-se nessa abordagem então que uma discrepância numa rotina de nível 3 levaria a uma verificação de nível 2 e em último caso a de nível 1.

6 CONCLUSÕES

Este trabalho apresentou um modelo de monitoramento de recursos adequado ao controle de ambientes distribuídos, em especial, ambientes de nuvem, onde qualquer anomalia poderia ser percebida antes que causasse danos. O modelo possibilita ainda que os dados coletados se tornem a base de informações e uma fonte de consulta para averiguar se determinada aplicação está se comportando como deveria.

Um exemplo de uso da aplicação em questão seria o de verificar todos os dados gerados por uma máquina para avaliar o que causou a falha/desligamento em uma situação de uso inadequada. Neste caso, se um algoritmo causou a “pane”, poderia-se dizer como esse algoritmo executou, como alocou memória, processador, disco, ou qualquer outro tipo de recurso e de que maneira houve a degradação dos mesmos.

De posse dos dados estatísticos das aplicações e máquinas, por um tempo considerável, será possível prever falhas antes que ocorram e presumir que uma aplicação precisaria de uma quantidade ideal de recurso, otimizando inclusive a compra de hardware por empresas e centros de pesquisa, reduzindo custos e aumentando a confiabilidade e disponibilidade, já que se sabe qual a capacidade necessária.

Como ferramenta de monitoramento, será possível controlar todo um parque computacional, bem como recursos distribuídos, tarefa de difícil execução, não tratada diretamente por nenhum trabalho até a data da escrita deste documento.

O modelo de pontuação de recurso também é considerado uma inovação, já que até neste momento não foi encontrado nenhum mecanismo de pontuação que permitisse pontuar recursos heterogêneos, equalizando valores discrepantes de ordens de grandeza diferentes, avaliando não uma única máquina em uma escala pré-definida, mas todos os recursos com pontuações máximas e mínimas de acordo com todos os recursos disponíveis na infraestrutura.

Outro ponto tratado por esse trabalho em termos de qualidade de implementação é que são utilizados canais seguros de transmissão, com o uso de criptografia com infraestrutura de chaves públicas para troca de mensagens, protegendo a integridade e confidencialidade dos dados na rede.

Por fim, o trabalho acrescenta mecanismos facilitadores, já que o projeto integra-se ao esforço dos pesquisadores da UCP, LNCC e UFPR em intensificar as pesquisas na área de Nuvens Computacionais e torná-las um serviço de mais fácil acesso, a pesquisadores e empresas em geral, permitindo fazer auditorias de recursos e segurança em degradações que surgirem por mal uso, uso ilícito ou inadequado do ambiente. Sem que os usuários precisem ter conhecimentos específicos da infraestrutura para utilizá-la.

6.1 TRABALHOS FUTUROS

O desenvolvimento da aplicação proposta por esse trabalho está em fase de coleta de dados estatísticos de recursos distribuídos. Os trabalhos futuros serão o de geração de relatórios e análise de pontos críticos, onde será possível averiguar a causa de panes nos recursos físicos e as possíveis medidas a serem tomadas para que os recursos sendo degradados não tornem o sistema instável.

De posse deste modelo de previsão, outro trabalho futuro será o de continuação da implementação, afim de disponibilizar uma interface mais amigável de análise, controle e monitoramento de todo tipo de recurso em redes locais e geograficamente distribuídas.

REFERÊNCIAS

- [1] Lee Badger, Tim Grance, Robert Patt-Corner, and Jeff Voas. DRAFT Cloud Computing Synopsis and Recommendations. Technical report, U.S. Department of Commerce Gary Locke, Secretary National Institute of Standards and Technology Patrick D. Gallagher, Director.
- [2] Dror G. Feitelson and Larry Rudolph. Towards convergence in job schedulers for parallel supercomputers. In *Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing*, IPPS '96, pages 1–26, London, UK, UK, 1996. Springer-Verlag.
- [3] Borko Furht and Armando Escalante. *Handbook of Cloud Computing*. Springer Publishing Company, Incorporated, 1st edition, 2010.
- [4] G1 Globo. Excesso de procura ao site do sisu causou lentidão no sistema, 2011. [Online; accessed 11-Junho-2012].
- [5] H. Gottlieb, J.E. Mogensen, and A. Zettersten. *Symposium on Lexicography XI: Proceedings of the Eleventh International Symposium on Lexicography May 2-4, 2002 at the University of Copenhagen*. Lexicographica. Series Maior. De Gruyter, 2005.
- [6] Guardian. Site playstation da sony fica fora do ar por causa de invasão, 2011. [Online; accessed 11-Junho-2012].
- [7] IBM. Alta disponibilidade, 2011. [Online; accessed 11-Junho-2012].
- [8] Jean-Claude Laprie. Fundamental concepts of dependability. technical report series university of new-castle upon tyne computing science. *Citeseer ISSN: 13681060*, 1145(010028):7–12, 2001.
- [9] F.L. Licht, B. Schulze, and E. Ishikawa. Fornecimento automatizado de certificados de curta duração para dispositivos móveis em grades computacionais, 2006.
- [10] Microsoft. Plataforma microsoft azure, 2011. [Online; accessed 11-Junho-2012].
- [11] F.P. Miller, A.F. Vandome, and M.B. John. *Fork Bomb*. VDM Verlag Dr. Mueller e.K., 2010.
- [12] Folha Online. Pane afeta emissão de passaportes, 2011. [Online; accessed 11-Junho-2012].
- [13] Leonardo O.; Machado Javam C. Sousa, Flavio R. C.; Moreira. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. *ERCHEMAPI. Todos os direitos reservados a EDUFPI.*, 2009.
- [14] IT Word. Pane no cloud da amazon, 2011. [Online; accessed 11-Junho-2012].