

# Reconfiguração de ambientes virtualizados através do uso de Teste Baseado em Modelos e SLAs

Elder M. Rodrigues, Avelino F. Zorzo, Flávio M. de Oliveira, Leandro T. Costa

<sup>1</sup>Programa de Pós Graduação em Ciência da Computação  
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)  
Porto Alegre - RS - Brasil

{elder.macedo, avelino.zorzo, flavio.oliveira, leandro.costa}@pucrs.br

**Abstract.** *This work proposes a process of reallocation of resources in virtualised environments using Model-based Testing, data mining and decomposition of SLA. Initially, we gather data from performance test on each virtual machine, through a process of performance testing where test scenarios derive from UML models of the application. Data mining techniques and SLA decomposition are applied to suggest, from a current configuration, the best set of parameters of reconfiguration required to modify the environment and provide better overall performance.*<sup>1</sup>

**Resumo.** *Este trabalho propõe um processo de realocação de recursos em ambientes virtualizados utilizando técnicas de testes baseados em modelos UML, mineração de dados e decomposição de SLA. Inicialmente, são obtidos os dados de desempenho sobre cada máquina virtual, através de um processo de testes de performance a partir de modelos UML da aplicação. Sobre estes dados são aplicadas técnicas de mineração de dados e decomposição de SLA. Tal processo pretende sugerir, a partir da configuração corrente, qual o melhor conjunto de parâmetros de reconfiguração exigidos para modificar o ambiente e prover um melhor desempenho global.*

## 1. Introdução

Teste de desempenho é uma metodologia capaz de prover informações altamente relevantes, não somente sobre o sistema em teste (*System Under Test* - SUT), mas também sobre a infra-estrutura na qual o sistema está implantado. Informações obtidas com os testes de desempenho, como por exemplo, o tempo de resposta de uma aplicação, pode ser traduzido no nível de recursos físicos consumidos pela aplicação durante a sua execução, tais como o uso de processador e memória. Estes dados sobre a estrutura física são fundamentais no processo de realocação de recursos em ambientes virtualizados, pois para inserir uma aplicação nesse ambiente, é necessário conhecer o nível de recursos exigidos pelo ambiente para suprir a demanda de recursos da aplicação.

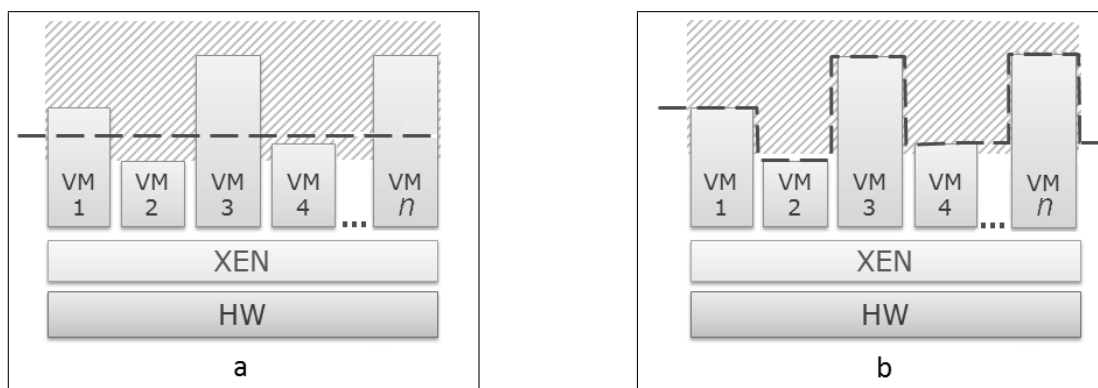
Através dos testes de desempenho é possível obter dados sobre a real capacidade computacional à disposição de cada Máquina Virtual (*Virtual Machine*-VM). Os resultados dos testes servem para predizer os níveis de recursos consumidos por uma aplicação sob determinada carga de trabalho. A partir desses resultados, é proposto um método

---

<sup>1</sup>Este trabalho foi desenvolvido em colaboração com a HP Brasil P&D.

para realocação dinâmica de recursos, onde a sugestão de parâmetros de reconfiguração baseia-se em modelos preditivos gerados a partir de mineração de dados. Além disso, ao utilizar esta abordagem, os testes podem ser realizados sobre diferentes cargas de trabalho, desta forma, possibilitando a predição de recursos para diferentes SLAs (*Service Level Agreement*).

Para atingir um critério de cobertura, capaz de cobrir de maneira satisfatória as possíveis configurações de processador e memória que uma aplicação, implantada em uma determinada estrutura pode necessitar, é preciso uma vasta quantidade de cenários de testes diferentes, tornando a criação destes cenários uma atividade onerosa. Com o objetivo de superar este obstáculo, a geração de cenários de testes a partir de modelos apresenta-se como a alternativa ideal. Os cenários gerados podem ser convertidos em *scripts* de testes de desempenho, possibilitando a sua execução em uma *engine* de teste. Os resultados desses testes tornam possível a predição dos níveis de recursos necessários por determinado SLA.



**Figura 1. Alocação de recursos no Xen**

A partir dos dados obtidos com esta abordagem, consegue-se prever a quantidade de recursos consumidos por uma aplicação sob determinada carga de trabalho. Através da realocação de recursos, é possível garantir que determinado SLA seja mantido sob diferentes cargas de trabalho. Uma utilização é exemplificada utilizando o paravirtualizador Xen [1]. A Figura 1 apresenta uma típica atuação do *Xen*, onde se tem, no exemplo, quatro VMs (1, 2, 3, 4), sendo que cada uma dessas VMs tem necessidades diferentes de consumo (ex: *CPU* e memória), ilustrado pela altura de cada VM. As linhas tracejadas representam a disponibilidade de recursos para as VMs. A área hachurada equivale aos recursos disponíveis para o ambiente. A Figura 1a representa uma alocação de recursos sem a utilização de SLAs, a qual não se preocupa em definir tais recursos às necessidades de cada VM. Analisando-se a Figura 1a percebe-se que a VM 2 está subutilizando os recursos a ela disponíveis, enquanto que a necessidade de consumo das VMs 1 e 3 é maior do que está à sua disposição, não sendo possível atender a toda sua demanda. Para melhorar o desempenho global do Xen, espera-se que os recursos possam ser adequadamente redistribuídos, respeitando políticas de SLA previamente definidas, e ajustando os recursos de acordo com as mesmas. Este modelo ideal está representado na Figura 1b, onde é possível notar, pela linha tracejada, que os recursos computacionais estão distribuídos proporcionalmente à demanda de cada VM, sem excesso ou escassez,

respeitando a disponibilidade global do ambiente.

Trabalhos recentes têm aplicado técnicas de mineração de dados para análise de desempenho em aplicações multicamadas [2] [3]. Com mineração de dados eles buscam apontar gargalos e identificar, em que situações e com quais configurações pode ocorrer queda de desempenho. A definição de políticas de SLA com a utilização de mineração de dados tem sido foco de estudo em [4] [5]. Em [5], também é citada a utilização do Xen, em um *Data Center*, como cenário. O uso de técnicas de mineração de dados para reconfiguração de recursos em ambientes virtualizados é utilizado em [6]. No trabalho apresentado em [7] SLAs foram utilizados no processo de realocação de recursos em ambientes virtualizados.

Este trabalho apresenta uma abordagem que busca utilizar o conhecimento adquirido com os trabalhos supracitados e deste modo permitir a utilização de teste baseado em modelos e mineração de dados para auxílio a reconfiguração de ambientes virtualizados com SLA. O artigo está estruturado conforme segue. A Seção 2 apresenta o modelo do processo de realocação de recursos proposto, detalhando suas etapas. Na Seção 3 é apresentado a estrutura experimental utilizada na validação do processo proposto. Na Seção 4 são apresentados os resultados obtidos e na seção seguinte são relatadas as conclusões e os trabalhos futuros a serem desenvolvidos.

## 2. Modelo do Processo Preditivo de Realocação de Recursos Proposto

A Figura 2 ilustra o processo proposto para realocação de recursos, o qual abrange um total de 12 passos divididos em quatro etapas.

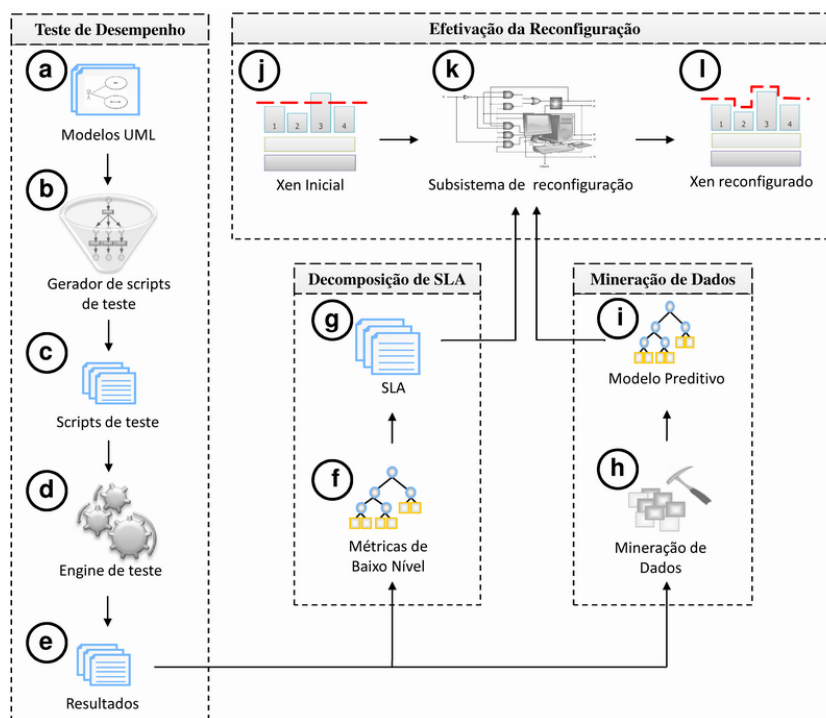


Figura 2. Processo preditivo de realocação de recursos

A primeira etapa consiste na execução de testes de desempenho baseados em modelos UML (*Unified Modeling Language*) sobre a estrutura virtualizada. Nesta etapa, são

modelados os casos de teste com o objetivo de mapear o desempenho do ambiente virtualizado pelo Xen (a). Em seguida, estes modelos são inseridos em uma ferramenta de geração automatizada de casos de teste de desempenho baseada em modelos UML (b), obtendo como resultado *scripts* para uma *engine* de teste de desempenho (c). Tais *scripts* são carregados na *engine*, realizando os testes de desempenho nos ambientes virtualizados (d), sendo então extraídos os resultados (e). Na etapa de decomposição de SLA, os resultados são utilizados para a decomposição das métricas de alto nível em métricas de baixo nível (f), mapeando os níveis de recursos necessários para cumprir o SLA (g). Paralelamente, na etapa de mineração de dados, são aplicadas nos resultados dos testes técnicas de mineração de dados (h), gerando um modelo preditivo o qual indica a melhor configuração de recursos para otimizar o desempenho de uma VM (i). A última etapa consiste na efetivação da reconfiguração do ambiente virtualizado, onde o modelo preditivo e o SLA são utilizados em conjunto para que dada uma configuração vigente de uma VM (j) e sendo seguidas as políticas de SLA, o subsistema (k) identifica qual a melhor maneira de reconfigurar os seus recursos(l).

A etapa inicial se baseia na utilização de modelos para a geração automática de casos de teste, pois além dos motivos apresentados por [8], o uso de modelos elimina a necessidade do conhecimento da linguagem de *script*, podendo, através do mesmo modelo gerar *scripts* de teste em linguagens diferentes. A partir dos modelos UML das aplicações [9], pode-se incluir estereótipos do UML *profile* para [10] teste de desempenho, pois o UML *profile* prove a descrição das características necessárias para automatização de teste de desempenho.

Como o desempenho é uma propriedade dinâmica, o uso de cenários é uma regra chave para determinar as características de desempenho. Em UML, os cenários são normalmente modelados usando grafos de colaboração ou de atividades. Sendo assim, é proposto por [11] a simulação de uma Rede de Petri Estocástica Genérica, a partir das informações extraídas dos modelos de casos de uso e de atividades dos sistemas em teste, modelos nos quais estão inseridos alguns dos estereótipos do UML *profile* para desempenho. A partir da extração dos dados dos estereótipos e dos modelos UML, geram-se os *scripts* de teste de forma automatizada. Os elementos utilizados representam uma parcela dos providos pelo UML *profile* para desempenho, sendo encontrados mais detalhes em [12]. Os principais elementos são representados da seguinte forma:

- PApopulation: número de usuários.
- PAoccurrence: taxa de chegada dos usuários.
- PAextDelay: *think time*, o tempo no qual o usuário aguarda pra começar a executar uma atividade.
- PArespTime: tempo de resposta esperado para uma atividade.
- PAprob: probabilidade de determinada atividade ser executada.

Desta forma, o processo do teste de desempenho utilizado neste trabalho ocorre do seguinte modo:

1. Criação dos modelos UML de casos de uso e atividades;
  - (a) Inserção dos estereótipos do UML *Profile* para *performance* nos modelos de casos de uso e atividades. Desta forma, dados relevantes para a realização do teste, são inseridos no modelo.

2. Extração das informações contidas nos estereótipos e, baseado nessas informações, criação automatizada de *scripts* de teste.
3. Execução dos *scripts* na *engine* de teste.

Os resultados extraídos dos testes de desempenho são utilizados, na segunda etapa, para a decomposição de SLA e na terceira etapa, que consiste na mineração de dados.

A decomposição de SLA permite uma definição precisa dos níveis de recursos que uma aplicação necessita para atender as especificações de um SLA [13] [5]. Desse modo, uma métrica de um SLA, como por exemplo, o tempo de resposta de uma aplicação é decomposta em métricas de baixo nível, como largura de banda, utilização do processador e memória disponível. Para cada um desses são definidos valores mínimos a serem atingidos. Mais detalhes sobre decomposição de SLA podem ser encontrados em [7] [14].

Os resultados dos testes de desempenho são também utilizados pela etapa de mineração de dados para indicar qual configuração será aplicada a cada VM do ambiente virtualizado. Os dados produzidos servem para apoiar a tomada de decisão para reconfiguração. Por tratar-se de mineração de dados, faz-se necessário obter uma grande quantidade de dados. Para tanto, é realizado um planejamento para execução dos testes, os quais estão melhor detalhados em [6].

O último passo é composto pelo subsistema de realocação de recursos proposto, que busca otimizar a utilização do *hardware* e possibilitar a utilização de SLA em ambientes virtualizados. Esse processo é realizado através da realocação de recursos entre VMs, tendo como base o processo preditivo. A partir da predição, o subsistema de realocação de recursos configura as máquinas virtuais, levando em consideração a sua configuração e suas necessidades atuais. Estas necessidades são determinadas pelo sistema de monitoramento que analisa o nível de recursos utilizados em cada uma das máquinas virtuais que compartilham a estrutura. No entanto, caso uma máquina virtual necessite de um nível de processamento maior do que os limites já definidos e exista a disponibilidade de recursos, é de responsabilidade do subsistema de realocação distribuir os recursos adequadamente. Mais detalhes sobre o subsistema de realocação de recursos podem ser obtidas em [7] [15].

### 3. Implantação do processo proposto

Para verificar o ganho de desempenho obtido com a proposta descrita neste artigo, o processo supracitado foi dividido em duas estruturas físicas, uma estrutura cliente e outra servidor (Figura 3). A estrutura cliente é composta pela ferramenta de modelagem ArgoUML [16], pelo gerador de *scripts* de testes de desempenho baseados em modelos UML proposto, e pela *engine* de testes de desempenho Apache JMeter [17], responsável por realizar testes de desempenho em aplicações *Web*.

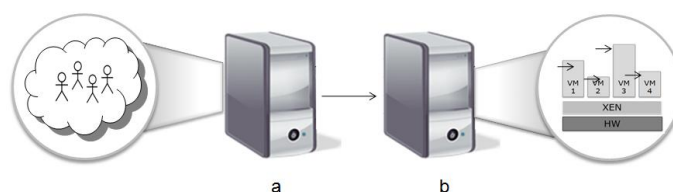


Figura 3. Estrutura utilizada

O ambiente proposto simula um *Data Center* virtualizado que hospeda um conjunto de quatro máquinas virtuais idênticas e suas aplicações. Cada VM hospeda um servidor *web* Apache [18], um servidor Tomcat 5.5 [19] e um servidor de banco de dados MySQL 5.0 [20].

Este ambiente é composto de duas máquinas físicas. Uma é utilizada como hospedeiro das VMs com os serviços, sendo que esta estrutura hospeda quatro Máquinas Virtuais Xen (Figura 3 (b)). A outra máquina (Figura 3 (a)) é utilizada como gerador de *workload* dos clientes. A estrutura descrita é hospedada em dois computadores idênticos com as seguintes configurações: processador *Athlon(tm) 64 X2 Dual Core de 3.8 GHz* com 2.0 GB de memória RAM, com um disco rígido IDE com capacidade de 160 GB. O sistema operacional utilizado foi o Ubuntu 7.04 server i386, Kernel 2.6.19-4-server. A ligação entre os computadores é realizada através de uma rede rápida *Gigabit Ethernet*.

## 4. Resultados

Com base no ambiente de teste descrito na Seção 3, foram implementados diversos casos de teste que buscam retratar diferentes cenários. Os cenários exemplificam situações onde um número variável de usuários acessa diversos conjuntos de aplicações que são hospedados em distintas VMs do mesmo ambiente virtualizado.

### 4.1. Casos de teste e resultados com o TPC-W

O TPC-W [21] foi utilizado para simular o cenário de um *Data Center* que hospeda diversas VMs em uma mesma estrutura física, sendo que uma ou mais dessas VMs sofrem uma sobrecarga no seu número de usuários. Para se comparar o desempenho do subsistema, com três diferentes níveis de recursos, com o escalonador Credit em modo estático, foram realizadas 120 execuções do *benchmark*, totalizando 240 horas de teste.

Para simular a situação de sobrecarga, foi definido em um primeiro momento um SLA que foi aplicado a todas as VMs da estrutura. Esse SLA possui um SLO (*Service Level Objectives*) que define que com uma carga de até 25 usuários simultâneos no sistema o tempo de resposta da aplicação deve ser inferior a 1 segundo. A partir deste ponto, foi realizada a decomposição e configuração dos recursos que são disponibilizados de maneira uniforme para todas as VMs da estrutura. Através da decomposição, o subsistema definiu os limites de recursos alocados para cada VM, sendo 20% do processador e 400 MB de memória RAM. Entretanto, a VM 1 recebeu uma carga de 35 usuários, sendo a mesma superior a aquela utilizada para realizar a decomposição. Esta adição de usuários busca simular uma situação onde uma VM da estrutura recebe uma carga de usuários maior do que a definida no SLA.

Como se pode observar na Figura 4, quando o Xen está utilizando o escalonador Credit, o tempo de resposta das VMs 2, 3 e 4 se encontram dentro dos limites definidos em seus SLAs, no entanto a VM 1 apresenta um tempo de resposta que ultrapassou por larga margem o valor estipulado no SLA. Este resultado se deve ao número de usuários adicionais aos definidos no SLA, pois embora a estrutura que hospeda as VMs possua 20% dos seus recursos livres, o escalonador do Xen (Credit estático) não é capaz de realizar a realocação dos recursos não utilizados em ambientes com SLAs. Com a utilização do subsistema de realocação, esses recursos tornam-se disponíveis para qualquer uma das VMs da estrutura, possibilitando suprir uma demanda maior por recursos pelas mesmas.

Comparando o tempo de resposta da aplicação, quando executada apenas com o escalonador do Credit, com o tempo de resposta de quando se utiliza o subsistema, nota-se uma acentuada redução do mesmo. Porém quando o mesmo teste foi realizado, só que se alterando a quantidade de recursos livres para 10%, os resultados mostraram que, quando comparado com o escalonador Credit estático, apesar do tempo de resposta da VM com sobrecarga apresentar redução, essa redução foi inferior a obtida no teste com 20% de recurso livre, mas foi suficiente para manter o SLA. Outro resultado relevante obtido, é que a utilização do subsistema, com 20% ou 10% de recursos livres, possibilitou um ganho de desempenho global para as VMs da estrutura, sendo que esse ganho é proporcionado pelo fato do subsistema realizar a realocação dos recursos para a VM que necessitar primeiro, neste caso a VM 1.

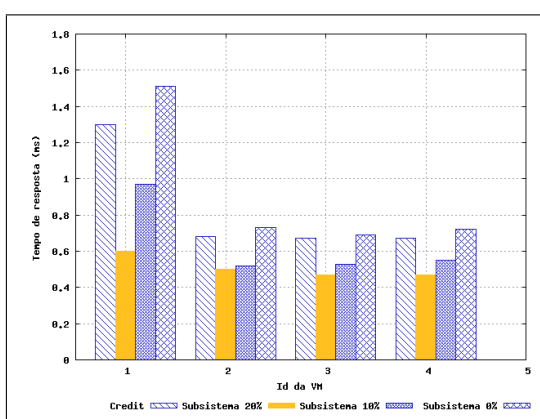


Figura 4. Sobrecarga em 1 VM

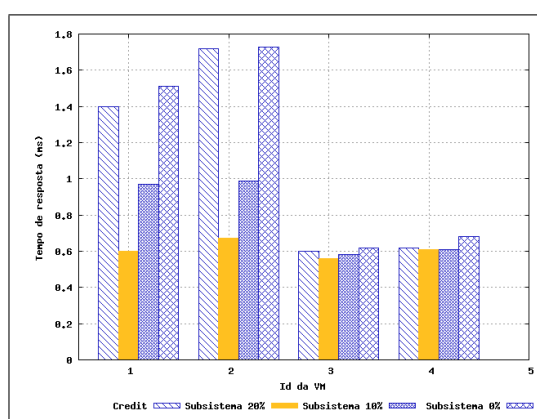


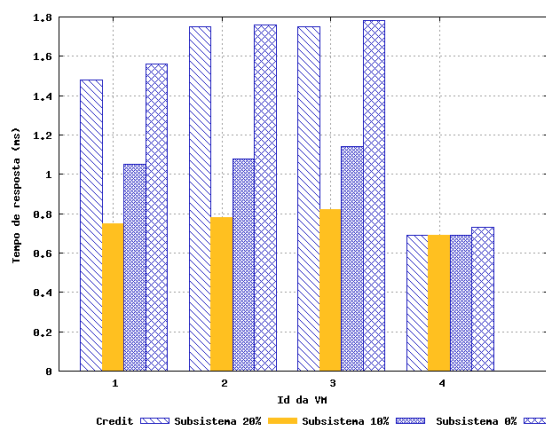
Figura 5. Sobrecarga em 2 VM

No entanto, se o processamento da VM 1 retornar aos níveis descritos em seu SLA, o subsistema retira os recursos adicionais e os realoca para próxima VM que apresentar um pico de processamento. Deste modo, os recursos não são monopolizados pela VM com sobrecarga, e tornam-se disponíveis para todas as VMs da estrutura. Contudo, apesar da utilização do subsistema ter se mostrado benéfica para o desempenho das aplicações hospedadas nas VMs quando o mesmo possuía recursos livres para realocar, este bom desempenho não se repete quando não existem recursos livres para serem realocados, sendo o desempenho do subsistema inferior ao escalonador Credit estático. Este fato ocorre, devido ao custo computacional provocado pela execução do subsistema, que realiza chamadas constantes aos métodos da biblioteca Xenstat para monitorar o nível de recursos livres, e também por tentar em algumas situações utilizar a ferramenta *xm* para realocar os recursos.

No segundo cenário de teste, de modo análogo ao anterior, as aplicações hospedadas na VM 3 e na VM 4 foram submetidas a uma carga que simula o acesso simultâneo de 25 usuários, e as VMs 1 e 2 tiveram o número de usuários simultâneos alterado para 35. O acréscimo de usuários busca retratar um cenário onde o SLA das duas VMs é quebrado temporariamente por parte do cliente e as VMs disputam os recursos livres disponibilizados pelo subsistema.

Como resultado (Figura 5), podemos observar que as VM 1 e 2 apresentaram um tempo de resposta maior do que o valor definido nos SLOs dos respectivos SLAs,

sendo que quando foram realizados os mesmos testes, mas utilizando o subsistema de realocação com 20% e 10% de recursos livres, foi obtida uma significativa redução do tempo de resposta das VM 1 e 2, quando comparado com o Credit estático. Entretanto nesse cenário houve uma redução do ganho nas demais VMs (VM 3, Figura 5) ou até mesmo não apresentando ganho (VM 4, Figura 5). Este resultado é devido às necessidades de processamento das VM 1 e 2 serem maiores. Deste modo, elas se utilizaram por um intervalo de tempo maior, dos recursos realocados, em detrimento das outras VMs que possuíam uma carga menor. Apesar do bom desempenho do escalonador com 20% e 10% de recursos livres, o desempenho do mesmo sem recursos para realocar (0%), assim como no teste anterior, foi ligeiramente inferior ao do escalonador Credit estático.



**Figura 6. Sobrecarga em 3 VMs**

No terceiro cenário de teste, busca-se retratar uma situação onde três VMs sofrem uma carga de usuários, maior do que aquela definida no SLA. Neste caso, as VMs 1, 2 e 3 sofrem uma carga de 35 usuários e a VM 4 uma carga de 25 usuários (carga prevista no SLA). Como se pode observar na Figura 6 a utilização do subsistema proporcionou uma melhora no desempenho das VMs com sobrecarga quando comparado com o escalonador Credit configurado em modo estático, no entanto, devido ao fato de existirem três VMs disputando os recursos livres, o subsistema acabou não beneficiando a VM sem sobrecarga.

Conforme os resultados apresentados nesta seção, a utilização do processo de realocação de recursos e do subsistema, proporciona uma melhora significativa no desempenho das aplicações hospedadas nos ambientes virtualizados (VMs), quando comparado com o escalonador Credit configurado em modo estático, principalmente em situações onde apenas poucas VMs apresentem sobrecarga. Comparando-se os gráficos das Figuras 4, 5 e 6, pode-se observar que a eficiência do subsistema tende a diminuir quando o número de VMs com sobrecarga aumenta. Esta situação é motivada pelo fato do subsistema compartilhar um conjunto comum de recursos entre as diversas VMs, sendo que quanto maior o número de VMs, menor o tempo de utilização dos recursos. Outro fato mandatário para esta redução de desempenho, é que quando existem diversas VMs com sobrecarga o subsistema necessita realizar um número maior de operações de realocar/desalocar recursos, o que gera certo custo computacional e conseqüentemente uma degradação no desempenho das aplicações. Os resultados também demonstram que a utilização do subsistema, em situações onde não existam recursos livres para serem



realocados, causa degradação no desempenho, motivado pelo custo computacional da execução do subsistema e, principalmente, da utilização do *xm* para realocar recursos.

#### 4.2. Casos de testes e resultados com o Jmeter

Os testes realizados com a ferramenta Jmeter buscam gerar resultados que possibilitem comparar o desempenho do subsistema, quando configurado com diferentes percentagens de recursos livres, com os demais escalonadores do Xen, em situações onde todas as VMs da estrutura recebem a mesma carga de usuários. Esta situação busca avaliar o desempenho do subsistema em uma situação que simule o pior caso possível de uso do mesmo, que é aquele em que todas as VMs da estrutura disputam os recursos livres alocados pelo subsistema.

Para gerar os resultados foram realizadas diversas cargas com a ferramenta de teste de desempenho Jmeter sobre o ambiente, descrito na Seção 3, variando-se de 5 a 75 a quantidade de usuários simultâneos, sendo que cada caso de teste foi realizado repetidamente com o Xen utilizando-se dos seguintes escalonadores em diferentes configurações:

- Scan Earliest Deadline First - SEDF;
- Credit em modo dinâmico;
- Credit em modo estático;
- Credit em modo estático com o subsistema de realocação realocando 20% dos recursos da infra-estrutura;
- Credit em modo estático com o subsistema de realocação realocando 10% dos recursos da infra-estrutura;
- Credit em modo estático com o subsistema de realocação realocando 0% dos recursos da infra-estrutura (sem recurso livre);

Com o objetivo de se obter os dados necessários para se comparar o desempenho do subsistema, com diferentes níveis de recursos, com os escalonadores do Xen configurados em seus distintos modos, foram realizados 840 execuções de *benchmark*, totalizando 210 horas de execução dos testes.

Os resultados obtidos (ver Figura 7) com os testes executados sobre a aplicação demonstram que a utilização do escalonador Credit em conjunto com o subsistema, quando existe 20% de recurso livre, proporciona uma melhora significativa no desempenho das aplicações hospedadas nas VMs, quando comparado com o Credit estático. Nota-se também neste caso que a diferença de desempenho entre o escalonador Credit e o subsistema, tende a ser menor quando poucos usuários estão acessando o servidor ou quando a carga de usuários é muito superior à capacidade do servidor. A primeira situação é motivada pelo fato de que em situações onde o número de usuários simultâneos na aplicação é baixo, o processamento nas VMs não aumenta a ponto do subsistema atuar efetivamente, e nas poucas situações em que aumenta, a quantidade adicional de recurso não se traduz em uma considerável diferença no tempo de resposta, dada a pouca carga na aplicação. Quanto à diferença no desempenho ser menor quando muitos usuários acessam a estrutura, isso se deve ao fato de que quando a capacidade do servidor é ultrapassada, o subsistema não consegue atender às necessidades por mais recursos por parte de todas as VMs, sendo que em muitos casos, determinada VM recebe os recursos e os utiliza durante todo o período do teste.

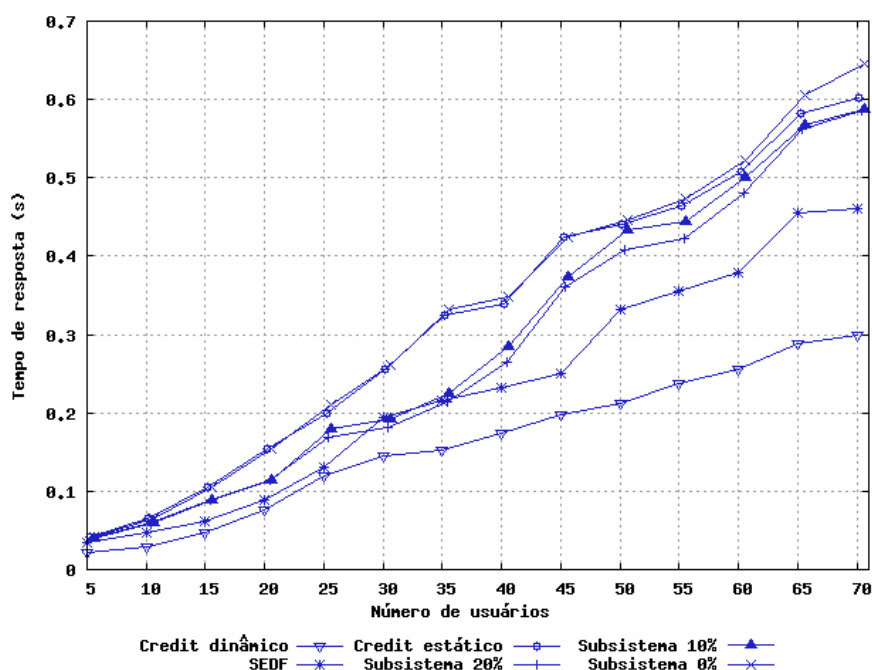


Figura 7. Desempenho do subsistema e dos escalonadores do Xen

Com o objetivo de se conhecer qual é o impacto no desempenho das aplicações quando o subsistema possui uma menor quantidade de recursos livres para realocar, foram realizados os mesmos testes descritos anteriormente, só que com 10% de recursos livres. Os resultados obtidos com esses testes demonstram que a queda no desempenho da aplicação, quando comparado com o teste onde o subsistema possuía 20% de recursos para realocar, foi inexistente em situações onde poucos usuários acessavam a aplicação. No entanto, conforme a carga de usuários sofreu um incremento, a diferença no desempenho tende a tornar-se maior. Outra situação análoga ao teste com 20% de recursos ocorre quando a quantidade de usuários está próxima do limite do que a VM suporta. Nessa situação, o desempenho do subsistema, em seus distintos níveis de recursos, tende a tornar-se bastante próximo. Comparando-se o desempenho do subsistema quando com o escalonador Credit em modo estático, a utilização do subsistema com 20% e 10% de recursos livres proporciona uma melhora no desempenho das aplicações em todos os casos de teste. Sendo que, a utilização do subsistema, apenas em uma configuração, com 0% de recursos para realocar, apresentou um desempenho inferior ao obtido com o escalonador Credit em modo estático. Este fato é motivado pelo próprio custo computacional da execução e monitoração do subsistema, já que apesar de não tentar realocar recursos livres, quando os mesmos não estão disponíveis, o subsistema continua a monitorar as VMs em busca de situações que quebrem o SLA, e a quantidade de recursos livres no sistema. Esta monitoração é necessária para que o subsistema identifique quando algum recurso livre torna-se disponível no ambiente virtualizado, sendo esta situação possível, por exemplo, caso uma VM seja realocada para outra estrutura, ou removida pelo administrador.

Apesar de apenas o escalonador Credit configurado em modo estático possibilitar a utilização de SLAs em ambientes virtualizados e a realocação de recursos através do

subsistema, também foram realizados os testes com os escalonadores SEDF e Credit em modo dinâmico.

Os resultados obtidos com esses escalonadores foram os que apresentaram os melhores resultados, uma vez que o bom desempenho do Credit dinâmico já era esperado, pois o próprio escalonador já possui internamente as funções necessárias para a realocação, não necessitando executar chamadas a bibliotecas de monitoração e a ferramentas de alocação de recursos. Um resultado não esperado foi o desempenho do escalonador SEDF, que apesar de ser inferior ao Credit dinâmico, foi superior aos demais escalonadores e ao subsistema. Esse resultado é relevante para a continuidade no desenvolvimento do mesmo, pois devido ao seu suposto fraco desempenho, não é prevista a inclusão desse escalonador na próxima versão do Xen.

## 5. Conclusões e Trabalhos Futuros

Este trabalho apresentou um processo para realocação de recursos em ambientes virtualizados. Tal processo baseia-se em quatro etapas principais. A primeira está em utilizar testes de desempenho baseados em modelos UML para obter as informações necessárias sobre o ambiente. A segunda etapa consiste em realizar a decomposição das métricas de alto nível em métricas de baixo nível para definir os recursos necessários para determinado SLA. A terceira utiliza técnicas de mineração de dados para realizar a predição de qual a melhor configuração de um ambiente virtualizado quando várias VMs são executadas. A última etapa consiste na utilização dos resultados das etapas anteriores pelo sistema de reconfiguração de modo a garantir um SLA em um ambiente virtualizado.

Para efetuar a realocação, foi implementado um subsistema que tem por objetivo otimizar a utilização do *hardware* de modo a manter um SLA contratado. Para realocar os recursos, esse subsistema faz uso da predição para definir os níveis de recursos necessários para manter um SLA.

Os resultados mostraram que, com a utilização do processo, o provedor consegue manter o serviço com os níveis definidos no SLA, inclusive em algumas situações de sobrecarga da VM. Deste modo ele garante recompensas da parte do cliente, pelo fato do serviço superar as expectativas do nível de serviço acordado.

Para aprimorar o subsistema desenvolvido, propõe-se como trabalhos futuros o desenvolvimento de uma ferramenta para realizar o refinamento do modelo preditivo dinamicamente durante a execução da aplicação no ambiente real, tornando a predição mais precisa.

## 6. Agradecimentos

Avelino F. Zorzo possui bolsa de produtividade CNPQ/Brasil.

## Referências

- [1] Barham, P. et al., “Xen and the art of virtualization,” *The ACM SOSP*, pp. 1–14, 2003.
- [2] Jung, G. et al., “Detecting Bottleneck in n-tier IT Applications Through Analysis,” *IFIP/IEEE DSOM*, pp. 149–160, 2006.

- [3] Parekh, J. et al., “Comparison of performance analysis approaches for bottleneck detection in multi-tier enterprise applications,” *IEEE International Workshop on Quality of Service*, pp. 149–160, 2006.
- [4] Udupi, Y. et al., “A classification-based approach to policy refinement,” tech. rep., HP Laboratories Palo Alto, 2007.
- [5] I. Cunha, J. Almeida, V. Almeida, and M. Santos, “Self-adaptive capacity management for multi-tier virtualized environments,” *10th IFIP/IEEE IM*, pp. 129–138, 2007.
- [6] Winck, A. T., Ruiz, D. D., “Processo de KDD para auxílio à reconfiguração de ambientes virtualizados,” in *SBSI*, pp. 211–222, 2008.
- [7] Rodrigues, E. et al., “Uso de Modelos Preditivos e SLAs para Reconfiguração de Ambientes Virtualizados,” *WSO. pages 147-158*, 2008.
- [8] J. D. Larry Apfelbaum, “Model-based testing,” *10th International Software Quality Week*, 1997.
- [9] A. J. Bennett and A. J. Field, “Performance engineering with the UML profile for schedulability, performance and time: A case study,” *12th MASCOTS*, pp. 67–75, 2004.
- [10] OMG. UML Profile for Schedulability, Performance, and Time, v1.1. <http://www.omg.org/docs/formal/05-01-02.pdf>. Acessado em 10 de Fevereiro, 2009.
- [11] Oliveira, F. et al., “Performance Testing from UML Models with Resource Descriptions,” *SAST*, pp. 47–54, 2007.
- [12] M. Marzolla and S. Balsamo, “UML-PSI: The UML Performance Simulator,” *International Conference on Quantitative Evaluation of Systems*, pp. 340–341, 2004.
- [13] Gupta, D. et al., “Enforcing performance isolation across virtual machines in Xen,” tech. rep., HP Laboratories Palo Alto, 2006.
- [14] Y. Chen, S. Iyer, X. Liu, D. Milojicic, and A. Sahai, “Sla decomposition: Translating service level objectives to system level thresholds,” in *ICAC*, (Washington, DC, USA), p. 3, 2007.
- [15] E. M. Rodrigues, “Realocação de recursos em ambientes virtualizados,” Master’s thesis, Pontifícia Universidade Católica do Rio Grande do Sul, RS, Brasil, 2009.
- [16] ArgoUML. <http://argouml.tigris.org/>. Acessado em 10 de Fevereiro, 2009.
- [17] Apache JMeter. <http://jakarta.apache.org/jmeter/>. Acessado em 10 de Fevereiro, 2009.
- [18] Apache HTTP Server Project. <http://www.apache.org/>. Acessado em 10 de Fevereiro, 2009.
- [19] Apache Tomcat. <http://tomcat.apache.org>. Acessado em 10 de Fevereiro, 2009.
- [20] MySQL. <http://www.mysql.com/>. Acessado em 10 de Fevereiro, 2009.
- [21] TPC-W. Transactional Processing Performance Council. <http://www.tpc.org/tpcw/>. Acessado em 10 de Fevereiro, 2009.