

## Monitoração de Nível de Serviço Baseada em Pesquisa Multidimensional

Andreas Kiefer<sup>1</sup>, Elias Procópio Duarte Jr.<sup>2</sup>, Cristina D. Murta<sup>2</sup>

Centro Internacional de Tecnologia de Software (CITS)  
Rua do Semeador, 702 – 81270-050 Curitiba, PR

<sup>2</sup>Universidade Federal do Paraná (UFPR)  
Departamento de Informática  
Caixa Postal 19081 – 81531-990 Curitiba, PR

andreas@cits.br, {elias, cristina}@inf.ufpr.br

**Resumo.** Este trabalho apresenta uma estratégia para monitoração de nível de serviço que permite a averiguação de múltiplos requisitos quantitativos concorrentes. A estratégia proposta é baseada em árvores multidimensionais de pesquisa. A árvore *K-D* (K-Dimensional tree) é empregada para monitoração online contínua; a árvore *K-D-B* (K-Dimensional B tree) é empregada para a monitoração offline, armazenando um histórico dos requisitos especificados. A estratégia proposta permite que a consulta a padrões de valores de *k* objetos monitorados seja realizada em  $O(\log N)$ , onde *N* é o número de amostragens ou o tamanho do log de monitoração. A instrumentação é feita com SNMP (Simple Network Management Protocol). É possível confirmar o cumprimento de acordos, bem como determinar as porções de tempo em que não foram cumpridos. Uma ferramenta foi implementada e resultados experimentais da monitoração de um conjunto de servidores Web e de um conjunto de servidores de vídeo são reportados e discutidos.

**Abstract.** Internet service providers that need to monitor the service level must deal with a number of multiple concurrent performance requirements. In this paper we present a service level monitoring strategy that allows both online and offline tracking of the performance of multiple concurrent resources. Data is collected with SNMP (Simple Network Management Protocol). The strategy is based on building multidimensional search trees. *K-D* (K-Dimensional) trees are employed for online continuous monitoring, and *K-D-B* trees are employed for offline monitoring, based on logs of monitored data. Searching with the proposed strategy has cost  $O(\log N)$  where *N* is the number of samplings or log size. The strategy allows clients and providers to confirm whether contract specifications were hold or not, and for how long. A practical tool was implemented and experimental results of monitoring several parameters of Web servers and video servers are presented.

### 1. Introdução

A presença ubíqua da Internet no dia-a-dia das pessoas e empresas tem permitido e incentivado a criação de novas aplicações distribuídas tais como comércio eletrônico, sistemas

de indexação e busca na Web, computação remota, entre várias outras. Vários fatores têm contribuído para o uso crescente de computação remota. Um número cada vez maior de empresas está mudando o paradigma de execução de suas tarefas computacionais: em vez de manter um parque computacional para a execução destas tarefas, as empresas contratam serviços de outras empresas, que possuem grandes clusters de servidores de processamento e de aplicações diversas, para executar seus serviços computacionais, que são solicitados via Internet.

A computação remota na Internet tem evoluído rapidamente, o que pode ser comprovado pela multiplicação de plataformas e padrões para computação em grade (*grid computing*) [Livny and Raman 2003] e *utility computing* [Buco et al. 2004]. Especialmente nos sistemas comerciais, um componente essencial é o contrato de serviço. O acesso ao sistema ou a obtenção de serviço são tipicamente pagos e os clientes têm requisitos específicos de qualidade. A definição dos requisitos de serviço é feita por meio de um contrato de nível de serviço (*Service Level Agreement – SLA*) [Bouillet et al. 2002], que é uma parte do contrato de serviço entre o provedor e seus clientes. O SLA define as obrigações de ambas as partes e as expectativas mínimas que o cliente pode ter em relação ao serviço contratado. Os contratos podem estabelecer multas e sanções para casos em que a qualidade do serviço realizado é inferior à contratada.

O provedor de serviço possivelmente atende vários clientes, cada um com requisitos específicos de serviço, e deve assegurar que cada cliente receba, para cada tarefa submetida ao sistema, recursos suficientes para que a tarefa seja completada dentro das especificações. Para isto, o provedor deve ter controle em tempo real do seu parque computacional e de sua rede, para definir, no tempo de chegada de uma tarefa, quando e onde ela será executada. A identificação dos elementos computacionais adequados para a execução de uma tarefa é operação complexa, que depende da prioridade da tarefa, dos recursos que a tarefa demanda no sistema e da carga observada no sistema no momento da decisão.

A implementação da seleção e alocação de recursos requer que parâmetros, medidas de carga e de desempenho de todos os componentes do sistema (computadores e rede) sejam continuamente coletados e organizados, de forma a facilitar a localização de nodos de processamento adequados para cada tipo de tarefa recebida pelo cluster. Assim, uma grande quantidade de informação numérica do sistema é coletada em intervalos de tempo pequenos, e deve ser tratada, indexada e pesquisada com grande frequência, mais especificamente, a cada chegada de tarefa ao sistema servidor.

Este trabalho propõe uma abordagem para monitoração de nível de serviço que permite a especificação de múltiplos requisitos simultâneos. Essas informações de monitoração podem ser usadas para a tomada de decisão na seleção de recursos para a execução de tarefas, com o objetivo de cumprir os requisitos de serviço no ambiente de computação. Um sistema que permite o mapeamento das informações dos sistemas computacionais sobre um espaço  $k$ -dimensional foi implementado e testado. Cada dimensão corresponde a uma característica (parâmetro ou medida) do sistema computacional. Por exemplo, a capacidade de processamento disponível, a memória disponível, a taxa de utilização de cada ponto de rede podem representar dimensões deste espaço.

A literatura indica que sistemas comerciais de computação em grade que provêm

serviço de computação remota coletam e armazenam este tipo de informação em sistemas de banco de dados [Buco et al. 2004], [Bouillet et al. 2002] e [Leff et al. 2003]. No entanto, as consultas multidimensionais e por intervalo necessárias neste ambiente são complexas e caras (lentas) em sistemas de banco de dados tradicionais, revelando a necessidade de estruturas de dados específicas para o ambiente e o problema em questão.

A pesquisa por intervalo realizada sobre um espaço multidimensional criado a partir das informações obtidas na monitoração vem facilitar o gerenciamento de nível de serviço. Os recursos monitorados podem ser organizados de duas maneiras, possibilitando a monitoração *online* e *offline*. A árvore K-D (*K Dimensional tree*) [Bentley 1975] é empregada para a monitoração *online* e a árvore K-D-B (*K-Dimensional B tree*) [Robinson 1981] para a monitoração *offline*. As duas estruturas de dados são apropriadas para a realização de pesquisa multidimensional por intervalo dos recursos monitorados. A estratégia de organização dos recursos monitorados na forma de árvores multidimensionais de pesquisa permite que a consulta a padrões de valores de  $k$  objetos monitorados seja realizada em  $O(\log N)$ , onde  $N$  é o número de amostragens ou o tamanho do log de monitoração. Como desvantagem, as árvores K-D possuem custo  $O(N \log N)$  para criação e atualização quando a distribuição dos dados é muito variada. Para reduzir o impacto de criação e atualização das árvores K-D em função da distribuição dos dados, adotou-se a estratégia de normalizar a distribuição dos dados em percentil de sua utilização.

A partir dos resultados da pesquisa multidimensional é possível tomar decisões referentes à alocação de recursos a serviços. Os recursos são permanentemente monitorados e alarmes podem ser configurados informando que determinados recursos estão atingindo limites ou não estão cumprindo metas previamente definidas. Em consequência, pode-se tomar determinadas decisões como, por exemplo, aumentar a capacidade do sistema ou dos recursos que estão atingindo limites estipulados, assegurando a disponibilidade e o desempenho dos serviços, ou alterar a alocação de novas requisições de serviços.

O sistema foi implementado utilizando objetos de gerência SNMP (*Simple Network Management Protocol*) [Harrington et al. 2002], o padrão da Internet para gerência de redes. Estudos de casos são apresentados, incluindo a utilização do sistema implementado para a monitoração de servidores Web e de vídeo.

O restante deste trabalho está organizado da seguinte maneira. A Seção 2 discute sobre projetos de sistemas de gerência de nível de serviço e apresenta os trabalhos relacionados. A Seção 3 apresenta conceitos sobre pesquisa multidimensional, descreve as árvores K-D e K-D-B, e discute os tipos de consultas possíveis nestas estruturas. A estratégia de monitoração proposta é apresentada na Seção 4. A Seção 5 descreve a ferramenta implementada e os resultados experimentais obtidos. O trabalho é finalizado com as conclusões na seção 6.

## 2. Monitoração de Nível de Serviço

A gerência de desempenho é funcionalidade básica esperada de sistemas de gerência de redes [Stallings 1998]. A gerência de desempenho inclui as seguintes etapas [Leinwand and Conroy 1996]: coleta de dados de utilização dos recursos do sistema computacional, análise dos dados coletados para descobrir tendências de utilização dos recursos monitorados, e o ajuste de limiares de utilização dos recursos.

A gerência de nível de serviço surgiu como um passo natural da evolução dos

sistemas de gerência, integrando gerência de redes, gerência de sistemas servidores de processamento e de servidores de aplicações, com o objetivo de assegurar níveis adequados de execução de serviços.

Os níveis de serviços são definidos por contratos de serviços (SLAs - *Service Level Agreements*). Um serviço pode ser definido como uma funcionalidade oferecida em um ambiente computacional, por exemplo, a execução de uma tarefa que requer processamento intensivo, ou uma consulta a um banco de dados hospedado no provedor de serviço.

Um SLA é composto pelos seguintes componentes [Verma 1999]: uma descrição do serviço a ser provido, o desempenho esperado do serviço, um procedimento detalhado descrevendo como agir em caso de problemas, um procedimento para monitoração e aviso do nível de serviço ao consumidor, as penalidades do provedor de serviço, se este não atingir o nível de serviço previamente estabelecido e uma descrição que indica em que circunstâncias a SLA não é aplicável. Também deve fazer parte do SLA a especificação de como as medidas brutas são coletadas no sistema e como são transformadas em medidas qualificadas para a avaliação do SLA [Buco et al. 2004]. Exemplos de métricas normalmente especificadas pelos clientes são tempo de resposta, disponibilidade e taxa de processamento.

A monitoração de contratos de nível de serviço (SLAs) envolve questões fundamentais tais como a especificação do SLA, a definição e coleta de métricas de serviço, a coleta de informações do sistema e a política de alocação de tarefas. Se o conjunto de servidores estiver numa situação de carga elevada, pode ocorrer violação de um acordo. Se houver perspectiva de violação de acordo, o sistema de monitoração de SLAs deve também decidir, em tempo real, que acordo será violado [Leff et al. 2003].

A quantidade de informação que um sistema provedor de serviço baseado em grade deve gerenciar é imensa. Informações sobre as aplicações disponíveis no sistema, os usuários e seus contratos de serviço, e os recursos disponíveis no sistema devem ser compostas de maneira que possam ser consultadas rapidamente, para tomada de decisões de escalonamento de tarefas em tempo real.

Alguns *frameworks* e modelos de sistemas de monitoração e controle de acordos de serviço têm sido propostos na literatura. Uma arquitetura para monitoração de SLAs é apresentada em [Molina-Jimenez et al. 2004]. A arquitetura trata da especificação dos SLAs, da localização de servidores, da coleta de métricas e medição do serviço e do serviço de detecção de violação de contratos. *Frameworks* com funcionalidades similares são descritos em [Buco et al. 2004, Bouillet et al. 2002, Leff et al. 2003]. Na proposta descrita em [Leff et al. 2003], os contratos são especificados por meio da linguagem XML para facilitar a definição dos acordos e evitar ambiguidades que possam surgir na linguagem natural. O sistema provê grande controle da carga para evitar situações de sobrecarga, ao mesmo tempo que procura manter em níveis mais altos a utilização do parque computacional, para tornar o negócio mais lucrativo.

Em todas estas propostas os autores reconhecem a necessidade e a importância da coleta, do armazenamento, e da recuperação eficiente das informações de controle do sistema. Todas prevêem unidades ou componentes para executar esta tarefa. No entanto, os autores não especificam claramente como isto será feito, isto é, não definem

uma estrutura de dados ou um formato para organizar e recuperar os dados. As propostas [Buco et al. 2004], [Bouillet et al. 2002] e [Leff et al. 2003] mencionam de forma genérica a utilização de um sistema de banco de dados.

No entanto, a quantidade de dados a ser armazenada, conjugada com a complexidade das consultas necessárias para localizar recursos e gerenciar o sistema, e a necessidade de obter os resultados das consultas para tomada de decisões em tempo real pode sobrecarregar o sistema de banco de dados, tornando inviável sua utilização. Este problema é discutido em [Dinda and Lu 2003], onde os autores apontam o elevado custo das consultas SQL a bancos de dados neste tipo de ambiente, e propõem mecanismos para diminuir estes custos.

### 3. Pesquisa Multidimensional

A manipulação de grandes conjuntos de dados multidimensionais é necessária em diversas aplicações. Estes dados incluem pontos, linhas, retângulos, áreas, volumes, entre outros. Tais dados podem ser representados por seu conjunto de atributos e por suas posições no espaço Euclidiano. O conjunto de atributos de dados multidimensionais pode ser representado na forma de um vetor de coordenadas. De uma maneira geral, cada dado multidimensional pode ser representado por um ponto no espaço  $k$ -dimensional, onde  $k$  representa seu número de atributos. A Figura 1 ilustra a representação de um ponto usando um vetor de duas coordenadas. O vetor de coordenadas  $(3, 5)$  contém o valor 3 no eixo de coordenadas  $x$  e o valor 5 no eixo de coordenadas  $y$ .

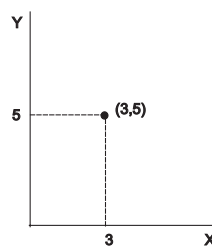
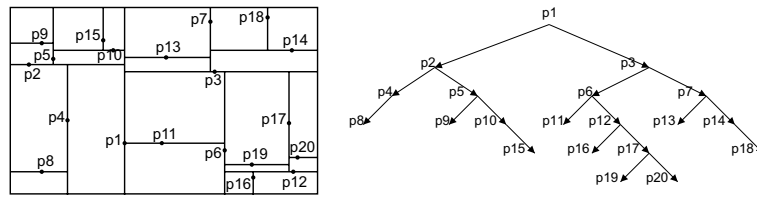


Figura 1. Representação de um ponto no espaço bidimensional.

Nos últimos vinte anos diversas estruturas de dados para manipulação eficiente de dados multidimensionais foram propostas. Muitas destas estruturas de dados são variações da árvore multidimensional de pesquisa K-D (*k-dimensional tree*) [Bentley 1975, Bentley 1979]. A árvore K-D é uma generalização da árvore binária de pesquisa. A árvore K-D utiliza um conjunto de pontos multidimensionais como entrada. Estes pontos estão imersos em um espaço multidimensional. A árvore K-D faz divisões recursivas deste espaço em sub-espacos, por meio de hiperplanos de divisão com  $k$ -dimensões. Por exemplo, para  $k=3$ , os hiperplanos de divisão se alternam perpendicularmente entre os eixos  $x$ ,  $y$  e  $z$ .

Os hiperplanos de divisão são obtidos pelo cálculo da mediana dos valores dos pontos em cada dimensão, que servirá como um valor discriminante. Assim, todos os pontos cujo atributo chave na dada dimensão forem inferiores ou iguais ao discriminante daquela dimensão serão armazenados na sub-árvore à esquerda da raiz. Da mesma maneira, os pontos cujo atributo chave forem superiores ao discriminante serão armazenados



**Figura 2.** Ilustração de uma árvore K-D com uma representação planar (esquerda) e representação em forma de árvore (direita).

na sub-árvore à direita da raiz. A cada dimensão um novo discriminante é encontrado, e o algoritmo de construção da árvore é executado de maneira recursiva até que todos os pontos do espaço multidimensional sejam armazenados na árvore.

A Figura 2 ilustra a árvore K-D representada de duas formas: a estrutura em um espaço bidimensional e na forma de uma representação abstrata em árvore. Em ambas as representações, o ponto  $p_1$  é escolhido como discriminante, e divide o espaço em dois novos sub-espacos. O ponto  $p_1$  é armazenado na raiz da árvore. O próximo plano da divisão passa pelos pontos  $p_2$  (sub-árvore da esquerda) e  $p_3$  (sub-árvore da direita). O particionamento termina quando todos os pontos estiverem armazenados na árvore.

A árvore K-D é uma estrutura de dados armazenada em memória primária, desta maneira possui bom desempenho nas operações de inserção e busca de dados. Entretanto, se o volume de dados for muito grande e sofrer freqüentes atualizações, o desempenho das operações de inserção e busca de dados na árvore K-D pode ser comprometido, pois a árvore tende a tornar-se desbalanceada. Para resolver o problema do desbalanceamento da árvore K-D, além de permitir a manipulação eficiente de grandes volumes de dados, a árvore K-D-B [Robinson 1981] foi proposta.

A árvore K-D-B é uma variação da árvore K-D, que combina propriedades da árvore K-D para manipulação de dados multidimensionais, e da árvore B para manipulação de dados em memória secundária. A árvore K-D-B também faz divisões recursivas do espaço multidimensional, onde os dados estão imersos em sub-espacos. As divisões do espaço na árvore K-D-B são semelhantes às divisões da árvore K-D, com a associação dos nós internos da árvore a uma região do espaço multidimensional, e dos nós folhas aos dados propriamente ditos.

Os nós internos são denominados páginas, sendo representados pelo par  $\langle \text{discriminante}, \text{ponteiro para um nó} \rangle$ . O discriminante representa um retângulo que contém todos os pontos armazenados na sub-árvore apontada para o nó. Os nós folhas são denominados páginas de pontos e contém os pontos do espaço multidimensional, indicados por suas coordenadas.

A Figura 3 ilustra uma árvore K-D-B. Cada nó interno na representação em árvore corresponde a um retângulo na representação planar, e cada nó folha corresponde a uma partição obtida pela divisão recursiva dos eixos de coordenadas  $x$  e  $y$ , usando linhas paralelas ( $x_1, x_2, x_3$  e  $x_4$ ;  $y_1, y_2$  e  $y_3$ ).

### 3.1. Tipos de Consultas Multidimensionais

Pesquisas são tipicamente iniciadas em resposta a consultas expressas em relação a um conjunto de registros válidos. Um algoritmo de pesquisa encontra, em uma estrutura de

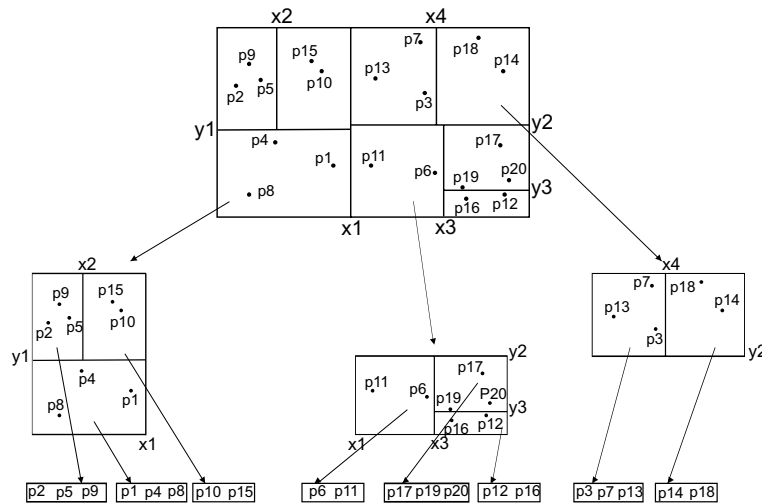


Figura 3. Ilustração de uma árvore K-D-B com representação planar.

dados, os registros especificados por uma consulta. Uma pesquisa multidimensional, algumas vezes referenciada como pesquisa por chave secundária ou simplesmente pesquisa por chave associativa, opera sobre registros que possuem tuplas de chaves ou atributos.

Em termos geométricos, um registro é considerado um ponto com uma tupla de chaves ou atributos, e todos estes atributos são distribuídos (imersos) em um espaço multidimensional, onde cada dimensão do espaço corresponde a um atributo do registro; desta maneira, para  $k$  chaves, teremos um espaço  $k$ -dimensional.

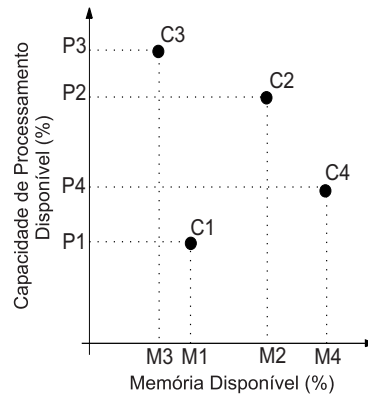
Uma consulta retorna todos os registros que satisfazem certas características. Vários tipos de consulta são possíveis. Diferentes tipos de consultas a registros são classificados de acordo com o espaço vetorial ou vetor de coordenadas. Uma consulta consiste em localizar todos os registros cujas chaves de pesquisa estão inseridos em uma região. Uma região é definida por um conjunto de coordenadas mínimas e máximas dentro de um espaço geométrico.

Alguns tipos de consultas são definidos em [Gaede and Günther 1998]. A consulta *exata*, o tipo de consulta mais simples, procura por um registro específico, definido por suas  $k$  chaves; a consulta *parcial* procura por todos os registros cujas  $k$  chaves combinem com seus valores em uma ou mais chaves de consulta; a consulta por *intervalo* procura por todos os registros que possuem suas  $k$  chaves dentro de um intervalo de chaves de consulta; por fim, a consulta por *proximidade* procura por registros que mais se aproximam de um dado registro, representado por um conjunto de chaves.

#### 4. A Estratégia de Monitoração Proposta

Nesta seção é descrita a estratégia de monitoração concorrente de múltiplos objetos de gerência, que consiste de três fases: amostragem, construção/atualização do espaço multidimensional e avaliação de requisitos.

Considera-se nesta descrição que os objetos monitorados são objetos SNMP. Esses objetos devem especificamente refletir de forma quantitativa características dos recursos computacionais monitorados. Associados a cada objeto podem ser especificados requisitos múltiplos, na forma de funções aritméticas e lógicas que refletem restrições impostas



**Figura 4. Exemplo de mapeamento de objetos e recursos no plano bidimensional.**

aos valores dos objetos.

Para a realização da primeira etapa, amostragem, são definidos os objetos a serem monitorados, os agentes dos quais os objetos são obtidos, além do intervalo de amostragem. Na segunda fase, os valores amostrados são mapeados em um espaço multidimensional, no qual cada dimensão corresponde a um objeto. Um ponto deste espaço corresponde a um recurso computacional gerenciado. A Figura 4 ilustra um exemplo, no qual quatro recursos gerenciados são mapeados em um espaço gerado a partir de duas características destes recursos: a capacidade de processamento disponível e a memória disponível. Nesta figura, C1, C2, C3 e C4 representam os recursos gerenciados. O eixo x representa o objeto *Memória Disponível (%)* e o eixo y representa o objeto *Capacidade de Processamento Disponível (%)*. Os pontos M1, M2, M3 e M4 representam os valores do objeto *Memória Disponível (%)*, para cada um dos recursos computacionais C1, C2, C3 e C4, respectivamente. De maneira análoga, P1, P2, P3 e P4 representam os valores do objeto *Capacidade de Processamento Disponível (%)*, para cada um dos recursos computacionais, C1, C2, C3 e C4, respectivamente.

Ainda nesta segunda fase, o mapeamento dos objetos para o plano multidimensional é feito utilizando árvores multidimensionais de pesquisa. A árvore K-D é empregada para a monitoração *online* e a árvore K-D-B é empregada para a monitoração *offline*. Tanto a árvore K-D como a árvore K-D-B permitem determinar a pesquisa considerando critérios referentes aos recursos computacionais.

Na terceira fase, a estratégia de monitoração proposta utiliza a árvore multidimensional atualizada bem como os requisitos especificados através de limites e funções aritméticas e lógicas que refletem restrições impostas aos valores dos objetos. Além disso, a pesquisa multidimensional é naturalmente aplicada sobre todos os recursos monitorados.

A pesquisa multidimensional permite: (1) localizar os dispositivos considerando critérios de desempenho dos dispositivos monitorados; (2) emitir alarmes quando requisitos não são cumpridos; (3) reportar intervalos de tempo dentro dos quais os recursos atendem ou não atendem certos requisitos. Desta forma, a pesquisa multidimensional permite que diversos parâmetros relacionados à utilização de recursos possam ser localizados em intervalos selecionados, através da pesquisa por intervalo ou parcial. As informações de



gerência coletadas possuem uma estampa de tempo, portanto, é possível analisar o perfil da utilização dos recursos ao longo do tempo.

## 5. Implementação e Resultados Experimentais

Nesta seção é descrita uma implementação da estratégia proposta, bem como resultados experimentais obtidos. A subseção 5.1 descreve a ferramenta implementada, incluindo o módulo de monitoração e de coleta das informações de gerência do SNMP, e o módulo de seleção das informações através de pesquisa multidimensional sobre as árvores de pesquisa K-D e K-D-B. A subseção 5.2 são apresentados resultados experimentais obtidos a partir da monitoração de grupos de servidores Web e de vídeo.

### 5.1. A Ferramenta Implementada

As informações monitoradas e coletadas são obtidas do protocolo de gerência SNMP. O módulo implementado foi desenvolvido na linguagem C++ em ambiente Linux usando a API Net-SNMP versão 5.1 [SNMP 2003]. Alguns objetos SNMP fornecem diretamente a utilização do recurso monitorado. Entretanto, outros objetos fornecem apenas um valor (contador), que é utilizado para o cálculo do recurso em questão. Nestes casos, são necessárias duas observações do recurso monitorado, em instantes diferentes de tempo, para a determinação da utilização do recurso. Como exemplo temos o cálculo da utilização da interface de rede, da utilização da CPU e da utilização de memória. A utilização do processador é calculada a partir dos objetos *ssCpuRawNice*, *ssCpuRawUser*, *ssCpuRawSystem* e *ssCpuRawIdle*, obtidos da UCD-SNMP MIB [SNMP 2003], e representa, respectivamente, o tempo, medido no kernel, que o sistema esteve em modo de “baixa prioridade”, “usuário”, “sistema” e “ocioso”.

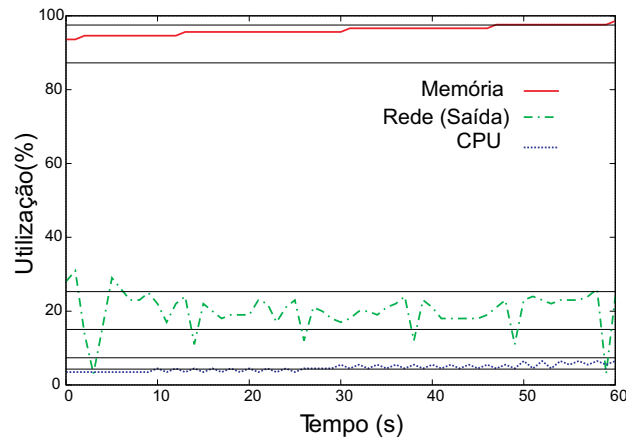
A utilização da memória é calculada a partir dos objetos *hrStorageSize* e *hrStorageUsed* da MIB *Host Resources* [Waldbusser and Grillo 2000]. Estes objetos representam, respectivamente, o total de memória disponível e a utilização de memória do sistema. Todos os valores coletados para a monitoração destes recursos são armazenados em memória secundária antes de serem manipulados pelo módulo de seleção das informações de gerência, descrito a seguir.

As informações obtidas pelo módulo de monitoração de informações de gerência são armazenadas em uma árvore multidimensional de pesquisa, permitindo assim localizar informações de gerência de recursos.

Este módulo foi implementado em linguagem C++ em ambiente Linux. Para a implementação da árvore K-D-B foi empregado o ambiente de software TPIE (*Transparent Parallel I/O Environment*) [TPIE 2003], que facilita a implementação de algoritmos de memória secundária, além de minimizar a comunicação para problemas que manipulam conjuntos grandes de dados. A árvore K-D foi implementada a partir do *container template C++ libkdtree++* [Libkdtree 2004]. Tanto a árvore K-D-B como a árvore K-D provêm um número ilimitado de dimensões, e permitem armazenar qualquer tipo de dado. Por se tratar de estruturas de acesso hierárquico, elas apresentam complexidade logarítmica para o tempo de acesso aos dados nas operações de inserção, remoção e busca.

### 5.2. Resultados Experimentais

Nesta seção são apresentados resultados experimentais obtidos a partir da monitoração de três servidores Web e de três servidores de vídeo. Nos experimentos foram moni-



**Figura 5. Utilização de recursos de um servidor de vídeo.**

torados seis parâmetros dos servidores, que serviram como atributos-chave das árvores multidimensionais de pesquisa: instante de tempo da consulta SNMP, utilização de entrada da interface de rede, utilização de saída da interface de rede, utilização de memória, utilização do processador e número de conexões TCP ativas. A interconexão de rede entre os servidores e os clientes foi feita via uma rede Ethernet de 100Mb/s, não dedicada exclusivamente aos experimentos.

Os registros com as informações obtidas na monitoração foram inseridos em uma árvore K-D, possibilitando assim a pesquisa *online* do desempenho do sistema. Entretanto, os registros poderiam ter sido armazenados também em uma árvore K-D-B, o que permitiria um acompanhamento do histórico da utilização de recursos do sistema monitorado.

Três experimentos foram planejados com objetivos distintos. O primeiro experimento tem o objetivo de comprovar que a ferramenta desenvolvida realiza de forma correta a monitoração das informações de gerência, e que a pesquisa multidimensional aplicada sobre estas informações retorna valores corretos. O segundo experimento foi executado com o objetivo de mostrar que a variação do número de atributos-chave não interfere no desempenho da ferramenta. O terceiro e último experimento mostra a relação do tempo de resposta da pesquisa multidimensional em função do número de registros armazenados na árvore de pesquisa multidimensional K-D, parâmetro importante para garantir a viabilidade da aplicação da ferramenta na monitoração contínua e *online* de sistemas contratados.

No primeiro experimento um servidor de vídeo foi monitorado em um intervalo de 60 segundos. A ferramenta de monitoração realizou a cada segundo a coleta dos dados referentes aos seis parâmetros monitorados. Ao final dos 60 segundos foram coletados 60 amostras de cada um dos seis parâmetros monitorados, refletindo o estado da utilização dos recursos deste servidor dentro do intervalo monitorado. Estas amostras foram armazenadas em arquivo e em seguida, inseridas pela ferramenta de seleção, em uma árvore K-D. A Figura 5 ilustra a utilização dos recursos do servidor de vídeo monitorado.

Sobre esta árvore foi efetuada a seguinte pesquisa multidimensional por intervalo: utilização do processador: [5-7]%, utilização de saída da interface de rede: [15-25]%,

# chaves	# registros inseridos	Tempo para inserção (ms)	# registros obtidos da pesquisa	Tempo para pesquisa (ms)
4	600	20	600	20
5	600	20	600	20
6	600	20	600	20

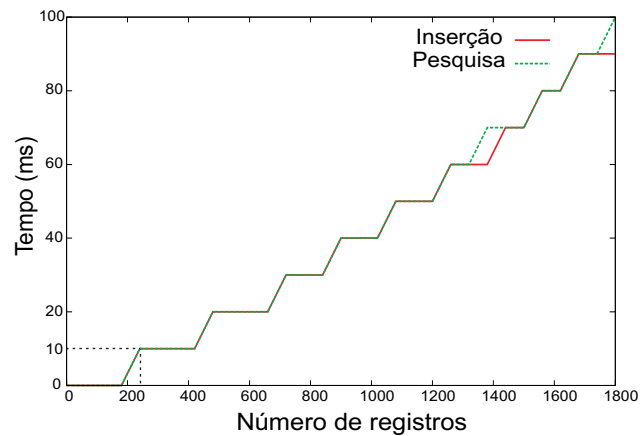
**Tabela 1. Tempo médio para inserção e pesquisa em árvore K-D com 4 a 6 chaves de pesquisa.**

utilização de memória: [85-95]%. Para os três demais parâmetros, os intervalos de pesquisa consideraram o intervalo do domínio dos registros. A pesquisa multidimensional descrita pode ser ilustrada graficamente na Figura 5 através das linhas horizontais que delimitam cada um dos três parâmetros pesquisados. Esta pesquisa multidimensional localizou 38 registros que satisfaziam os critérios acima descritos, o que significa dizer que, durante aproximadamente a metade do tempo da monitoração, os parâmetros selecionados apresentaram seus valores dentro do intervalo especificado pela pesquisa. Este resultado confirma que a ferramenta produz resultados corretos esperados, ou seja permite a monitoração e a pesquisa de parâmetros segundo critérios especificados.

O segundo experimento avaliou se o número de atributos-chave, ou chaves de pesquisa, interfere no tempo de resposta de uma pesquisa multidimensional. Para tanto foram monitorados recursos de servidores Web e de vídeo. Os valores obtidos para os recursos durante o período de coleta de dados foram inseridos, ao final da monitoração, na árvore multidimensional de pesquisa. Foram inseridos 600 registros com 4 chaves, 600 registros com 5 chaves e mais 600 registros com 6 chaves, em árvores distintas. Estas inserções foram repetidas centenas de vezes sempre em novas árvores e os tempos para estas inserções foram anotados.

A Tabela 1 apresenta valores médios obtidos do experimento. Nesta tabela podemos perceber que os tempos de resposta das operações de inserção e pesquisa na árvore multidimensional se mantêm inalterado independente do número de chaves de pesquisa. Este resultado decorre do fato de que a árvore de pesquisa multidimensional é uma generalização da árvore binária de pesquisa, onde, a cada nível da árvore, apenas um atributo chave é empregado como discriminante, e que servirá para guiar o percurso na árvore. Outro ponto importante vem do fato que o tempo para percorrer a árvore multidimensional de pesquisa independe do tipo de informação que é armazenada, ou seja, independe do conteúdo dos registros e, conseqüentemente, independe dos recursos monitorados. No caso da ferramenta desenvolvida, os experimentos acima foram executados tanto sobre dados de monitoração do servidor de vídeo, como dados de monitoração do servidor Web. O resultado deste experimento demonstra que é possível efetuar consultas empregando uma grande quantidade de chaves de pesquisa e, como conseqüência disto, é possível monitorar e selecionar diversos recursos simultaneamente, segundo algum critério, obtendo assim resultados mais precisos.

O terceiro experimento visa relacionar o tempo de resposta da pesquisa multidimensional com a quantidade de registros inseridos e obtidos da pesquisa. Para este experimento os servidores foram monitorados por um período de 30 minutos (1.800 segundos), e os registros de monitoração foram coletados a cada segundo, totalizando 1.800 registros. Estes registros foram inseridos na árvore multidimensional de pesquisa, a cada



**Figura 6. Tempos de resposta para operações de inserção e pesquisa em função do número de registros.**

60 segundos. A inserção dos registros considerou sempre o número total de registros monitorados até o momento de inserção, ou seja, no instante de tempo 60, sessenta registros haviam sido coletados e inseridos na árvore. No instante de tempo 120, cento e vinte registros haviam sido coletados e todos estes inseridos em uma nova árvore. Este processo foi repetido até que todos os 1.800 registros fossem inseridos na árvore de pesquisa multidimensional. Para cada inserção na árvore, o tempo (em ms) da operação de inserção foi medido juntamente com o tempo (em ms) para a pesquisa por intervalo, considerando intervalos que retornassem o conjunto completo de dados inseridos, ou seja, a pesquisa mais abrangente possível.

Como o custo para as operações de inserção e atualização da árvore K-D é o mesmo, e a biblioteca empregada para desenvolver a ferramenta de monitoração *on-line* disponibilizava o algoritmo de inserção de dados em uma árvore K-D, optou-se por criar uma nova árvore K-D a cada novo conjunto de dados.

Os tempos de resposta para a operação de inserção dos conjuntos de registros coletados desde o início da monitoração (gerando um histórico da monitoração), e os tempos de resposta da pesquisa multidimensional sobre a quantidade de registros inseridos na árvore foram plotados e são apresentados na Figura 6.

Neste gráfico podemos perceber que o tempo de resposta tanto da operação de inserção quanto da operação de pesquisa cresce em função do número de registros inseridos e obtidas da pesquisa. Entretanto, se considerarmos um conjunto de aproximadamente 230 registros, o tempo de resposta para a operação de inserção estará próximo de 10ms. Se somarmos o tempo de inserção na árvore com o tempo de uma pesquisa multidimensional por intervalo, nesta árvore, teremos em média um tempo de resposta de 20ms. Neste caso, podemos considerar que um servidor de aplicação em um cluster que utilize a estratégia proposta demoraria em média 20ms para inserir 230 registros em uma árvore multidimensional de pesquisa e obter o resultado de uma pesquisa multidimensional por intervalo.

Cada nova requisição que chega ao cluster precisa ser encaminhada para a unidade computacional adequada, em termos de utilização de recursos, para que possa ser atendida. Para cada nova requisição, uma pesquisa seria realizada, o que custaria 10ms. Para

100 requisições, o tempo de resposta da pesquisa seria de aproximadamente 1 segundo, o que nos leva a uma taxa de aproximadamente 100 requisições atendidas por segundo.

O tempo de resposta da consulta por intervalo pode ser diminuído em função da quantidade de registros armazenados na árvore de pesquisa. Para reduzir o tempo de resposta da pesquisa, aumentando a taxa de tarefas atendidas por segundo, é necessário reduzir a quantidade de registros inseridos na árvore. Isto pode ser feito inserindo na árvore apenas os registros mais recentes, que refletem o estado mais atual do sistema.

Estes resultados permitem concluir que a pesquisa multidimensional pode ser empregada para ambientes computacionais reais, onde existe a necessidade de localizar determinados recursos seguindo critérios complexos de requisitos. Além disso, é possível uma boa taxa de atendimento de requisições considerando que os dados mais recentes são empregados para decidir pelo encaminhamento de requisições para unidades computacionais que as atenderão.

## 6. Conclusão

Este trabalho apresenta uma solução para o problema da monitoração de nível de serviço. Limites para expressões de objetos monitorados podem ser especificados e avaliados continuamente de forma *on-line* ou *off-line*. A estratégia, baseada em pesquisa multidimensional, foi proposta e implementada, utilizando o SNMP para instrumentação. As informações monitoradas são organizadas em uma árvore multidimensional de pesquisa, sendo usada a árvore *K-D* para monitoração contínua *online*, e a árvore *K-D-B* para monitoração *offline*. A estratégia proposta permite que a consulta a padrões de valores de  $k$  objetos monitorados seja realizada em  $O(\log N)$ , onde  $N$  é o número de amostragens ou o tamanho do log de monitoração. Resultados práticos da utilização da ferramenta para a monitoração de grupos de servidores Web e de vídeo foram apresentados, permitindo acompanhar a utilização dos servidores, bem como determinar picos de maior e menor utilização, ao longo do período de observação. Além disso, foi possível apresentar resultados que motivam o uso de estruturas de dados multidimensionais para localizar, dentro de um conjunto de elementos de rede monitorados, aqueles que atendem com requisitos específicos de utilização de recursos. Estes resultados permitem concluir que a pesquisa multidimensional pode ser empregada de forma vantajosa no contexto dos servidores de vídeo e Web testados no experimento.

Trabalhos futuros incluem experimentos com grandes massas de dados. Além disso o desenvolvimento de uma interface Web para a ferramenta, permitindo a configuração, monitoração e consulta multidimensional. Também está previsto o acoplamento de um módulo para geração de gráficos dos recursos monitorados. Por fim, a integração da ferramenta a um sistema de grade comercial com acordos de nível de serviço é também objetivo do trabalho.

## Referências

- Bentley, J. L. (1975). Multidimensional Binary Search Trees Used For Associative Searching. *Communications of the ACM*, 18(9):509–517.
- Bentley, J. L. (1979). Multidimensional Binary Search in Database Applications. *IEEE Transactions on Software Engineering*, 4(5):333–340.

- Bouillet, E., Mitra, D., and G.Ramakrishnan, K. (2002). The Structure and Management of Service Level Agreements in Networks. *IEEE Journal on Selected Areas in Communications*, 20(4):691–699.
- Buco, M. J., Chang, R. N., Luan, L. Z., Ward, C., Wolf, J. L., and Yu, P. S. (2004). Utility Computing SLA Management based upon Business Objectives. *IBM Systems Journal*, 43(1):159–178.
- Dinda, P. A. and Lu, D. (2003). Nondeterministic Queries in a Relational Grid Information Service. In *SC '03: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, pages 12–26, Washington, DC, USA. IEEE Computer Society.
- Gaede, V. and Günther, O. (1998). Survey on Multidimensional Access Methods (Revised Version). *ACM Computing Surveys*, 30(2):170–231.
- Harrington, D., Presuhn, R., and Wijnen, B. (2002). An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. *Request for Comments 3411*.
- Leff, A., Rayfield, J. T., and Dias, D. M. (2003). Service-Level Agreements and Commercial Grids. *IEEE Internet Computing*, pages 44–50.
- Leinwand, A. and Conroy, K. F. (1996). *Network Management a Pratical Perspective*. Addison-Wesley Longman Publishing Co., Inc., 2nd edition.
- Libkdtree (2004). The libkdtree++ Project. <http://freshmeat.net/projects/libkdtree/>, acessado em 12/2005.
- Livny, M. and Raman, R. (2003). Enterprise Resource Management: Applications in Research and Industry. In Foster, I. and Kesselman, C., editors, *The Grid*. Morgan Kaufmann, 2nd edition.
- Molina-Jimenez, C., Shrivastava, S., Crowcroft, J., and Gevros, P. (2004). On the Monitoring of Contractual Service Level Agreements. Technical Report series CS-TR-835 April 2004 School of Computing Science, University of Newcastle upon Tyne.
- Robinson, J. T. (1981). The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes. In Lien, Y. E., editor, *Proceedings of the 1981 ACM SIGMOD International Conference on Management of Data, Ann Arbor, Michigan, April 29 - May 1, 1981*, pages 10–18. ACM Press.
- SNMP (2003). The NET-SNMP Project Home Page. <http://net-snmp.sourceforge.net>, acessado em 12/2004.
- Stallings, W. (1998). *SNMP, SNMPv2, SNMPv3 and RMON1 and 2*. Addison-Wesley Longman Publishing Co., Inc., 3rd edition.
- TPIE (2003). A Transparent Parallel I/O Environment. <http://www.cs.duke.edu/TPIE/>, acessado em 04/2003.
- Verma, D. (1999). *Supporting Service Level Agreements on IP Networks*. Macmillan Technical Publishing.
- Waldbusser, S. and Grillo, P. (2000). Host Resources MIB. *Request for Comments 2790*.