Processamento de Logs de Servidores Web Utilizando Grade Computacional*

Juliano Freitas da Silva, Gerson Geraldo H. Cavalheiro

Universidade do Vale do Rio dos Sinos – UNISINOS
Faculdade de Ciência da Computação
São Leopoldo – RS – Brasil
Av. Unisinos, 950. Fone/Fax: 590-8161
julianofs@pop.com.br, gersonc@exatas.unisinos.br

Introdução

Atualmente temos presenciado a popularização de várias tecnologias relacionadas à área de redes de computadores, que têm propiciado vários avanços no campo da comunicação de dados. Também acompanhamos um aumento constante no poder de processamento dos computadores atuais. Em contrapartida, as aplicações utilizadas têm se tornado cada vez mais críticas e importantes ao trabalho das organizações, exigindo maior necessidade de processamento, de alta disponibilidade e de tolerância a falhas. Naturalmente, tal aumento de criticidade das aplicações geraria a necessidade das empresas investirem na aquisição de computadores mais potentes, juntamente a toda a infraestrutura necessária para o funcionamento confiável dos mesmos. Como uma alternativa à esta tendência pode-se considerar a utilização de **Grades Computacionais** (*Grids*). Dessa forma, Grades Computacionais foram desenvolvidas com o objetivo de criar um ambiente com grande poder de processamento, escalável e de baixo custo [CIR 03].

Grades computacionais são plataformas de suporte a execução de aplicações paralelas em equipamentos heterogêneos distribuídos [CIR 03]. Esta abordagem é baseada no compartilhamento de recursos, no sentido que os computadores que compõem a Grade podem ser utilizados para outras finalidades, sendo que muitas vezes pode-se também fazer uso de tempo ocioso de processamento.

Atualmente existem várias ferramentas de suporte à implementação de Grades Computacionais, das quais podemos citar o Globus Toolkit [FOS 98], o MyGrid [MYG 03] e o Condor [LIT 88].

Neste trabalho objetiva-se utilizar grades computacionais para o processamento de logs de servidores Web, de forma a se explorar o paralelismo de dados desta classe de aplicações.

Processamento de Logs de Servidores Web

Atualmente grande parte das organizações possui um ou mais sites na Internet, com as mais variadas finalidades, dos quais podemos destacar sites institucionais e de comércio eletrônico. Neste contexto é importante que as organizações conheçam o perfil de utilização do seu Web site de forma a dimensionar melhor os recursos computacionais, adaptar a navegabilidade do site analisando as áreas mais acessadas e as menos acessadas, entender o perfil do público que acessa o site, entre muitas outras finalidades. Para obter tais informações torna-se necessário processar os arquivos de log gerados pelos servidores Web como o Apache [APA 03] e o MS-Internet Information Server [IIS 03]. Sendo que cada acesso ao site é gerado um novo registro, que é adicionado ao arquivo de log de utilização do servidor Web.

Neste trabalho estamos utilizando os arquivos de log gerados por servidores IIS 5.0. Nos arquivos de log utilizados encontramos as seguintes informações: Data e Hora do acesso, Ip do cliente, usuário que realizou o acesso, ip destino, porta destino, método de requisição utilizado (exemplo: GET, POST, etc), página acessada, a consulta URI realizada pelo cliente (quando ocorrer), Status da requisição e Browser utilizado. A seguir vemos um exemplo de registro de log neste tipo de arquivo.

* Trabalho realizado na disciplina de Programação Paralela e Distribuída

```
2003-08-30 13:51:39 200.208.30.22 - 145.90.48.70 80 GET /index.htm - 302 Mozilla/4.0
```

Em sites que possuem grande quantidade de acessos o tamanho destes arquivos de log tendem a se tornar bastante grandes. Tomemos como exemplo o servidor IIS 5.0, considerando o tamanho médio de um registro de log como sendo de 130 bytes. Agora tomemos como exemplo o portal *Yahoo!*, que segundo dados de 2001 [WEL 01], possuía em torno de 900 milhões de páginas acessadas por dia. Poderíamos estimar, assim, que um grande portal rodando sobre IIS 5.0 geraria em torno de 10,9 Gb de dados de log por dia. Teríamos em torno de 76,3 Gb por semana e 305.2 Gb por mês. Dessa forma, para realizar o processamento de tais arquivos de log se torna necessária uma quantidade considerável de recursos computacionais.

Implementação da Solução

A ferramenta MyGrid

O MyGrid é uma ferramenta de suporte à implementação de Grades Computacionais, que provê um ambiente global para a execução de aplicações paralelas do tipo *bag-of-tasks* em máquinas nas quais o usuário possui acesso. Aplicações do tipo *bag-of-tasks* são aquelas cujas tarefas são independentes, podendo ser executadas em qualquer ordem [CIR 03].

No MyGrid temos a diferenciação funcional de *máquinas da Grade* e *Máquinas Base*. *Máquina Base* é aquela onde o MyGrid está instalado, que é responsável pelas principais funções da Grade, como: escalonamento de tarefas, gerenciamento de tarefas, monitoramento das máquinas que compõem a grade computacional, etc. Já *máquinas da Grade* são aquelas que compõem o grade do usuário, provendo os recursos computacionais para execução paralela [CIR 03].

Para definir as máquinas que compõem a Grade, é submetido ao MyGrid um arquivo no formato de descrição de Grade [MYG 03]. Neste arquivo devem ser descritas as seguintes informações referentes a cada processador que compõem a grade. Da mesma forma existe o formato de arquivo para descrição de tarefas. Neste arquivo são descritas informações para a execução de cada tarefa a ser submetida à grade.

O MyGrid foi escolhido para a implementação da aplicação em questão devido à sua proposta principal, de ser uma solução pronta para a implementação de grades e ser compatível com a necessidade de processamento de massa de dados a ser realizada.

Descrição da Implementação

A solução implementada para o processamento de logs de servidores web foi baseada no princípio da distribuição de carga. Ou seja, os dados de entrada (um ou mais arquivos de log de servidores Web IIS 5.0) são divididos entre as máquinas que compõem a grade. Assim, cada processador realiza o processamento de uma parte dos dados. Após os logs processados, em máquina da grade é gerado um arquivo contendo a consolidação dos dados processados da parte que lhe foi submetida. Estes arquivos parcialmente consolidados são enviados por fim à máquina base para que esta realize a consolidação dos arquivos parciais de forma a apresentar o resultado final do processamento.

Inicialmente foi montado o ambiente de desenvolvimento com o MyGrid, composto de dois equipamentos: a máquina base e a máquina da Grade. Sendo que na máquina base foi realizada a instalação completa do MyGrid, e na máquina da Grade foi realizada a instalação do Agente do MyGrid (*User Agent*). A instalação desta ferramenta é descrita detalhadamente em [MYG 03].

Após termos o ambiente instalado e operacional, foram desenvolvidos os programas necessários para o processamento nas máquinas da Grade e na máquina base. Foi escolhida para o desenvolvimento a linguagem Java, em função de várias classes que a mesma disponibiliza que facilitaram o desenvolvimento desta aplicação e também pelo fato de ser multiplataforma.

Para a execução nas máquinas de grade foi desenvolvida apenas uma classe: estatWeb. Esta classe fará a leitura do arquivo de logs submetido à máquina, armazenando os valores lidos em tabelas *Hash*, uma para cada seção do arquivo de logs. As tabelas Hash foram implementadas a partir da classe *HashMap* do Java. Cada posição da tabela *Hash* possuirá dois atributos: Conteúdo e Contador. Dessa forma, primeiramente é verificado se o valor lido do arquivo de log já existe na *Hash*. Caso exista, um atributo contador é incrementado. Caso o valor a ser inserido não exista no *Hash*, o valor é inserido e o

campo contador é inicializado com o valor 1. A final da leitura do arquivo todas as tabelas *Hash* são lidas e é gerado um arquivo contendo os valores lidos e os contadores indicando o número de vezes de ocorrência do valor no arquivo de log submetido. Abaixo vemos a seção *Páginas acessadas* de um arquivo de saída desta classe. Verificamos o nome da página acessada e em seguida o número de vezes que a mesma foi acessada.

```
/app1/index.htm 650
/app1/ 800
/warning.gif 420
/app2/index.htm 240
```

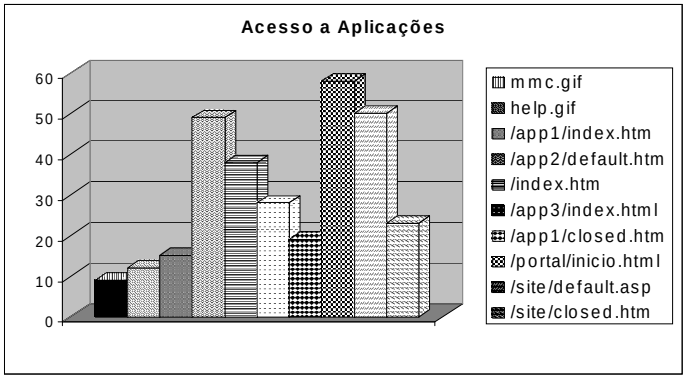
Assim, no arquivo consolidado parcial gerado pela classe estatWeb existirão seções semelhantes a estas para os outros dados constantes no arquivo de log: IPs de estações cliente, Usuários, IPs dos servidores acessados, Portas acessadas, etc. É interessante citarmos que para o *Parse* do arquivo foi utilizada a Classe StringTokenizer do Java.

Em contrapartida foram desenvolvidas duas classes para suporte da Grade na máquina base: Classe para a geração dos dados de controle do MyGrid (GeraControle) e classe para a consolidação final dos arquivos gerados pelas máquinas de Grade (Consolida).

A classe GeraControle possui como finalidade gerar o arquivo de descrição de tarefas a ser submetido ao MyGrid. Ela faz a leitura dos arquivos de log a serem processados e gera o arquivo de tarefas a ser submetido ao MyGrid.

A classe Consolida fará a leitura do conteúdo dos arquivos consolidados parciais gerados pelas máquinas da grade e transferidos para a máquina base. Como saída da execução deste programa temos um arquivo texto contendo os dados consolidados referentes ao processamento dos arquivos gerados pelas máquinas da grade. O funcionamento desta classe é muito semelhante ao da classe estatWeb, baseada também em tabelas *Hash*.

Abaixo vemos um gráfico das dez páginas mais acessadas em determinado dia, de um conjunto de dados fictícios, gerado a partir de um arquivo de dados consolidados.



.

Gráficos semelhantes a este poderão ser gerados para quaisquer seções dos arquivos de log processados.

Experimentos realizados

Como ambiente para a realização dos experimentos foi utilizado um aglomerado de 6 equipamentos: 5 computadores bi-processados Intel Xeon 2.8 Ghz, 1 Gb de memória RAM, 512 Kb de memória Cache e 1 computador IBM PC 300 GL, processador Pentium III 800 Mhz, 128 Mb de memória RAM, 256 Kb de memória Cache, interligados por um Switch Fast-Ethernet. Em todas as máquinas temos o sistema operacional GNU/Linux, kernel 2.4.21.

Neste contexto, o PC 300 GL desempenhava a função de Máquina Base e as outras 5 máquinas desempenhavam função de máquinas do grid.

Foram realizados dois testes principais: um experimento considerando a transferência dos arquivos via rede, e um experimento sem considerar o *overhead* da rede para que se possa verificar o desempenho da solução implementada.

No primeiro experimento foi utilizado um arquivo de log do IIS com tamanho de 385 Mb. Inicialmente o arquivo citado acima foi dividido em 4 partes iguais, contendo pouco mais de 96 Mb cada. Em seguida, tais arquivos foram submetidos para processamento na Grade, utilizando os programas desenvolvidos, descritos na subseção anterior. O tempo de processamento dos mesmos se deu em 8 minutos e 26 segundos. Nesta execução foram utilizados apenas 3 equipamentos do aglomerado, em conjunto com a máquina base.

No segundo experimento, no qual não estamos considerando o tempo de comunicação entre as máquinas da grade, foi utilizado um arquivo de log do IIS de 1.2 Gb. Este arquivo foi dividido em 5 partes iguais de aproximadamente 246 Mb. Os 5 arquivos criados foram copiados para cada um dos 5 equipamentos que compõe a grade. Após foram submetidas as tarefas ao MyGrid para processar tais arquivos de log. O processamento destes arquivos foi executado em 1 minuto e 6 segundos.

Para fins de comparação, foi executado um programa seqüencial para o processamento do arquivo de 1.2 Gb. Este programa foi executado em uma das máquinas do aglomerado (Intel Xeon bi-processado 2.8 Ghz) em 2 minutos e 29 segundos.

Conclusões e Trabalhos Futuros

A partir dos experimentos realizados pode-se verificar que, funcionalmente, esta abordagem para o processamento de massas de dados é bastante facilitado e acessível com o uso de ferramentas para a implementação de Grades Computacionais, em especial do software MyGrid. Em conjunto, pode-se comparar a execução seqüencial e a execução em grade do problema do processamento de logs, onde a segunda mostrou-se com melhor desempenho, mesmo considerando-se o *overhead* ocasionado pelas funções da grade como gerenciamento de tarefas, submissão de trabalhos à grade computacional, etc.

Para trabalhos futuros, pode-se realizar comparações de desempenho da aplicação descrita neste artigo em relação a outros modelos de programação paralela e distribuída. Também considera-se de extrema valia realizar estudos sobre a granulosidade das tarefas a serem submetidas à grade, de forma a detectar as configurações que ofereçam melhor desempenho.

A proposta desenvolvida neste trabalho pode servir como base para o desenvolvimento de outras aplicações de processamento de logs, como logs de firewall, analisadores de rede, etc. Além disso, o método descrito também poderá ser adaptado para realizar o processamento de outras aplicações que manipulem grandes volumes de dados, constituindo-se numa solução eficiente, escalável e de baixo custo.

Referências

[APA 03] Apache HTTP Server Project. Online em http://httpd.apache.org, setembro 2003.

[BAR 03] Barros, P. et al. Utilização do software MyGrid para adaptar uma aplicação de dinâmica molecular em uma Grade Computacional. **ERAD 2003: 3ª Escola Regional de Alto Desempenho,** p. 225-228, Santa Maria, 2003.

[CIR 03] Cirne, W. Grids Computacionais: Arquiteturas, Tecnologias e Aplicações. **ERAD 2003: 3ª Escola Regional de Alto Desempenho,** p.103-134, Santa Maria, 2003.

[FOS 98] Foster, I.; Kesselman, C. THE GLOBUS PROJECT**:** A STATUS REPORT, 1998. Anais... , 1998.

[IIS 03] Web Hosting with IIS 5.0. Online em http://www.microsoft.com/serviceproviders/whitepapers/, setembro 2003.

[MYG 03] Manual MyGrid versão 1.1. Online em http://www.dsc.ufpb.br/mygrid, setembro 2003.

[WEL 01] Welsh, M.; Culler, D. Virtualization considered harmful: OS design directions for well-conditioned services. **Anais 8º Workshop on Hot Topics in Operating Systems (HotOS VIII)**, Schloss Elmau, Alemanha, 2001.