UNIVERSIDADE ESTADUAL DE LONDRINA

ERNESTO YUITI SAITO
HAYATO FUJII
HELIO ALBANO DE OLIVEIRA
VINICIUS TADEU GONÇALVES

WEB CACHE DISTRIBUÍDO

LONDRINA - PR

2011

Sumário

1. RESUMO

Um sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens. Essa definição leva às seguintes características dos sistemas distribuídos: concorrência de componentes, falta de um relógio global e falhas de componentes independentes.

O compartilhamento de recursos é um forte motivo para a construção de sistemas distribuídos. Os recursos podem ser gerenciados por servidores e acessados por clientes, ou podem ser encapsulados como objetos e acessados por outros objetos clientes.

Os desafios advindos da construção de sistemas distribuídos são a heterogeneidade de seus componentes, ser um sistema aberto, o que permite que componentes sejam adicionados ou substituídos, a segurança, a escalabilidade – a capacidade de funcionar bem quando o número de usuários aumenta – o tratamento de falhas, a concorrência de componentes e a transparência.

2. INTRODUÇÃO

Redes de telefones móveis, redes corporativas, redes de fábrica, redes em campus, redes domésticas, redes dentro de veículos, todas elas, tanto separadamente como em conjunto, compartilham as características básicas que as tornam assuntos relevantes para estudo sob o título *sistemas distribuídos.*

Definimos um sistema distribuídos como sendo aquele no qual os componentes de hardware ou software, localizados em computadores interligados em rede, se comunicam e coordenam suas ações apenas enviando mensagens entre si. Essa definição simples de sistemas distribuídos tem as seguintes consequências importantes:

- *Concorrência*: em uma rede de computadores, a execução concorrentes de programas é a norma. A capacidade do sistema de manipular recursos compartilhados pode ser ampliada pela adição de mais recursos (por exemplo, computadores) na rede. A coordenação de programas em execução concorrente e que compartilham recursos também é um assunto importante e recorrente.
- *Inexistência de relógio global*: quando os programas precisam cooperar, eles coordenam suas ações trocando mensagens. A coordenação frequentemente depende de uma noção compartilhada do tempo em que as ações dos programas ocorrem. Entretanto, verifica-se que existem limites para a precisão com a qual os computadores podem sincronizar seus relógios em uma rede. Essa é uma consequência direta do fato de que a única comunicação se dá por meio do envio de mensagens em uma rede.
- *Falhas independentes*: nos sistemas distribuídos, as falhas na rede resultam no isolamento dos computadores que estão conectados a ela, mas isso não significa que eles param de funcionar. Na verdade, os programas neles existentes talvez não consigam detectar se a rede falhou ou se tronou demasiadamente lenta. Analogamente, a falha de um computador ou o término inesperado de um programa em algum lugar no sistema não é imediatamente percebida pelos outros componentes com os quais ele se comunica. Cada componente do

sistema pode falhar independentemente, deixando os outros ainda em funcionamento.

A motivação para construir e usar sistemas distribuídos é proveniente do desejo de compartilhar recursos. O termo “recurso” é bastante abstrato, mas caracteriza bem o conjunto de coisas que podem ser compartilhadas de maneira útil em um sistema de computadores interligados em rede. Ele abrange desde componentes de hardware, como discos e impressoras, até entidades definidas pelo software, como arquivos, bancos de dados e objetos de dados de todos os tipos.

3. OBJETIVO DO SISTEMA PROPOSTO

A utilização do Web Cache se justifica pela economia de tempo que se dá através de mecanismos que distribuem cópias das informações disponíveis na Web em vários locais diferentes. Estes mecanismos deixam a informação mais próxima dos usuários finais, facilitando a localização e diminuindo a latência na recuperação dos dados.

São três as principais vantagens em se fazer *caching* do conteúdo Web:

- A latência entre pedido e resposta é reduzida, fazendo com que as páginas sejam carregadas mais rapidamente;
- O consumo de banda de rede é reduzido, diminuindo assim o tráfego e o congestionamento da rede;
- A carga no servidor Web de origem é significantemente reduzida, através da distribuição dos dados entre *proxy* espalhados pela rede.

A primeira das vantagens é a principal e mais citada quando se trata de *caching*. Latência inclui, basicamente, o tempo que um objeto leva para ser transferido do servidor de origem até o *proxy* (latência externa) e o tempo de transferência do objeto do *proxy* até o cliente (latência interna). Uma justificativa para a busca por melhores na latência é que, do ponto de vista dos usuários, um melhor tempo de resposta às suas requisições, aumentam o grau de satisfação.

A segunda vantagem diz respeito à redução do consumo de banda de rede. Reduzindo o consumo de banda não apenas reduz o custo da rede, como também reduz a utilização do link e do servidor de origem, reduzindo a latência externa.

A terceira vantagem se refere à diminuição na carga do servidor Web. Ao reduzir o tráfego entre *proxy* e servidor, o número de requisições ao servidor diminui. Dessa forma, há uma redução na carga do mesmo, melhorando seu desempenho.

Através dessas vantagens expostas, há um certo ganho por todas as partes envolvidas no acesso à Web. Os usuários experimentam uma rede mais rápida, devido à redução da latência na transferência de informações. A rede é favorecida devido à diminuição no desperdício de banda com dados

redundantes, deixando largura de banda disponível para outros dados passarem.

Além dessas vantagens que o sistema oferecerá, o sistema proposto terá suporte a algumas propriedades desejáveis aos sistemas de Web Cache, que são:

- *Rapidez no Acesso*: É necessário que um sistema de Web Cache, mesmo com o aumento da latência interna, reduza a latência geral dos acessos. Do ponto de vista do usuário, a latência observada, na média, deve ser menor do que se estivesse sendo usada uma conexão sem *proxy*.
- *Robustez*: Há três aspectos importantes com relação à robustez de sistemas de Web Cache: a queda de um *proxy* não pode conduzir a uma queda no sistema; o sistema de cache deve ser tolerante a faltas; e o sistema de cache deve ser desenvolvido de forma que seja fácil recuperá-lo em caso de falhas.
- *Transparência*: De maneira geral, os caches devem atuar de modo transparente.
- *Escalabilidade*: Os sistemas de cache devem ser escaláveis. É desejável que um sistema de cache se adapte, pelo menos, a problemas como crescimento de usuários e densidade da rede, adaptando seu tamanho e o número de replicações.
- *Eficiência*: O principal ponto a se avaliar quanto a eficiência de um sistema de Web Cache é o *overhead* imposto pelo sistema e o quanto ele aumenta a latência da rede. É necessário que um sistema de cache adicione o mínimo de sobrecarga à rede.
- *Adaptabilidade*: Os sistemas Web Cache deve se adaptar a toda heterogeneidade, como a heterogeneidade de usuários, necessidades e interesses dos usuários e o meios de acesso à rede.
- *Estabilidade*: Os esquemas utilizados em um sistema de Web Cache não podem introduzir instabilidades à rede.
- *Balanceamento de Carga*: É de grande importância que um sistema de cache distribua a carga, se possível, por toda a rede.

- *Simplicidade*: Sistemas simples são mais fáceis de implementar e são mais bem aceitos como padrões internacionais.

Por fim, o sistema proposta deve passar por algumas métricas para analisar o performance do sistema Web Cache implementado.

4. DESCRIÇÃO DO PROJETO

4.1. Funcionalidades do sistema

O projeto do Web cache distribuído consiste implementar funcionalidades para otimizar a comunicação entre o browser e o servidor web.

Basicamente, o cliente manterá os arquivos que foram acessadas previamente em sua cache para que o mesmo e os outros usuários da rede possam acessar esse arquivo de forma mais rápida.

Devido ao design do projeto, mesmo que o arquivo não esteja mais disponível na Internet, os clientes ainda poderão acessá-lo até o momento em que o arquivo expirar.

O Web cache foi projetado considerando a Internet como topologia principal. No entanto, essa limitação não impede que o software seja utilizado em uma rede local, por exemplo.

4.2. Arquitetura

Uma arquitetura de um sistema de Web Cache deve fornecer meios eficientes de comunicação inter-caches, aumentando este grau de cooperação, aumentando assim a sua performance.

4.2.1. Conexão

A ligação do cliente à rede do Web Cache Distribuído se dará através de uma conexão inicial ao servidor central, o qual conterá uma lista de todos os usuários já presentes na rede. Quando um cliente faz essa requisição, ele consultará uma lista salva localmente para identificar a quem deve requisitar a lista de clientes presentes.

Caso o cliente faça um requisição de conexão e o primeiro servidor apontado pela lista local não estiver disponível, o cliente tentará conectar-se a outro servidor, somente se houver outro candidato em sua lista pré-salva.

Após o cliente se conectar ao servidor central, o servidor irá salvá-lo em sua lista de clientes e repassará essa modificação a todos os usuários para que os mesmos atualizem suas listas de clientes ativos.

Caso algum cliente não responda a essa atualização, o servidor irá retirar da lista esse usuário que se desconectou.

No momento que o cliente recebe a lista de participantes, o mesmo efetuará um teste de latência a todos que estiverem na lista. Esse dado será salvo para que seja verificado se o cliente está presente. Este dado também será utilizado como fator de otimização quando um arquivo for requisitado.

Se o usuário encerrar a conexão, será salvo a lista de clientes conectados até o momento. Também será enviada ao servidor uma notificação do encerramento da conexão. Dessa forma, todos os clientes poderão atualizar sua lista.

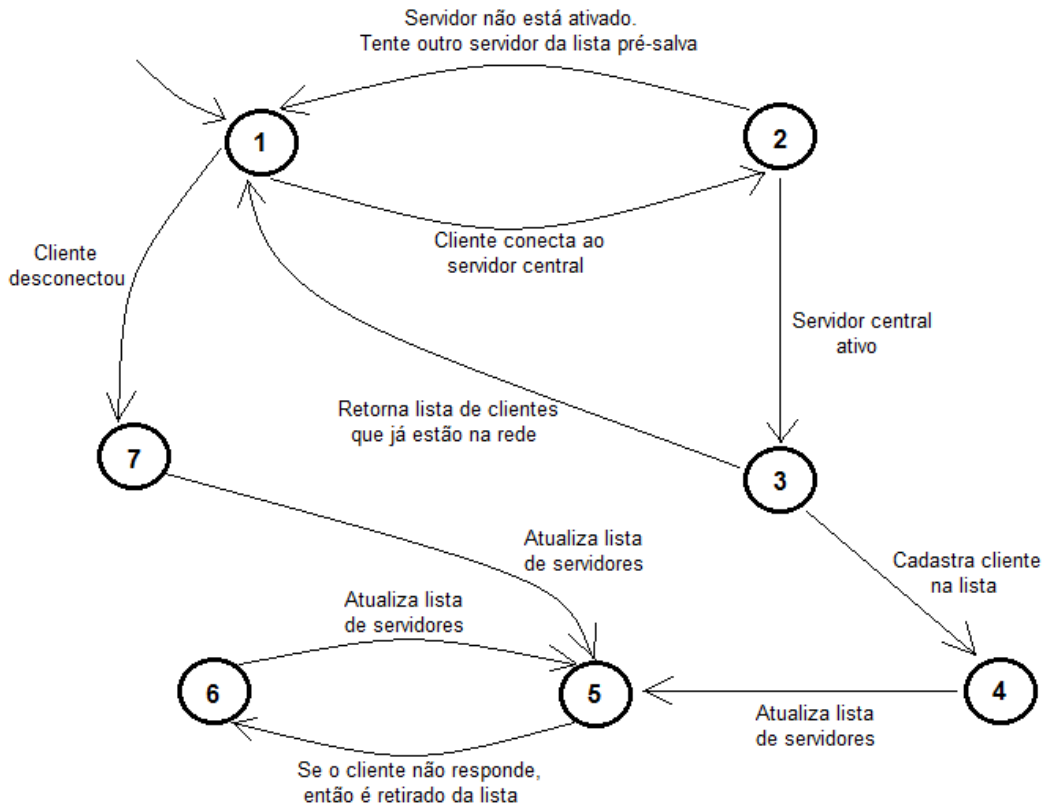

Figura 1 - Fluxo da Conexão

4.3. Mensagens

A grande parte das mensagens trocadas serão mensagens de texto, no qual haverá palavras chaves que identificará o contexto do texto trocados.

4.4. Transferência e Armazenamento dos arquivos

A limitação do espaço disponível para o cache local pode ser definido pelo cliente. A política de substituição a ser utilizado, caso o cache vier a ser preenchido completamente, será o First In, First Out. Para que o cache não tenha vários arquivos antigos, cada um terá um tempo de vida de um minuto.

Para identificar os arquivos na cache, será utilizado uma tabela hash. Para cada arquivo, a tabela terá os endereços de quem realmente possui os dados e seu tempo de vida restante.

Quando um cliente requisitar um arquivo, o mesmo consultará a tabela de dados disponíveis. Se estiver disponível de forma local, o arquivo será enviado para browser. Se estiver disponível na rede de clientes, então uma requisição será enviada para a máquina indicada; no caso de várias tiverem o mesmo arquivo, a primeira requisição será enviada para quem tiver a menor latência. Por fim, caso o arquivo não for encontrado, deverá ser obtido da internet por quem requisitou.

Haverá casos em que a requisição do arquivo remoto não poderá ser completada por problemas de sincronização da tabela hash. Caso um cliente receber uma requisição de um arquivo já apagado, o mesmo irá recuperar o arquivo novamente da internet, guardar na sua cache e aumentar seu tempo de vida.

Quando um novo arquivo não disponível na rede do proxy for obtido, quem fez a requisição buscará na Internet, adicionando uma nova entrada na tabela. A adição de um novo arquivo não afetará lista dos vizinhos de forma imediata; porém, um pacote de 10 atualizações acumuladas serão enviados ou caso a última atualização da listagem foi feita há mais de 60 segundos. O controle de atualização da lista será feito de forma local utilizando semáforos.

5. MECANISMO DE AVALIAÇÃO DOS RESULTADOS

Existem algumas métricas que se relacionam à performance de Web Caches. As mais normalmente adotadas são *hit ratio (HR)* e *byte hit ratio (BHR)*. Em alguns casos se utiliza o tempo de resposta médio como maneira de medir, principalmente para analisar a latência.

Hit Ratio é o percentual de requisições satisfeitas pelos objetos armazenados em cache.

Byte Hit Ratio é o percentual de bytes requisitados pelo cliente que foram enviados pelo cache, sem solicitação ao servidor de origem.