

Pesquisa de CDN ver 1.1

André Luiz Cabral Dutra
www.andrecabral.eti.br

SUMÁRIO	3
O QUE É CDN?	3
SISTEMA TRADICIONAL.....	4
QoS.....	5
<i>Técnica de reserva</i>	5
<i>Técnica de diferenciação de classe</i>	5
<i>Técnica CDN</i>	6
O QUE DEVE SER ANALISADO?	7
ESCALABILIDADE	8
BALANCEAMENTO DE CARGA	9
<i>Aumentando o desempenho</i>	10
TOLERÂNCIA À FALHAS	12
BALANCEAMENTO POR DIFUSÃO	12
DCN EM SITE ÚNICO	13
TRIANGULAÇÃO.....	13
OPERAÇÃO DAS CDNS	13
ESQUEMAS ATIVOS PARA ROTEAMENTO EM CDNS.....	14
TABELA DE ROTAS EM CDNS	15
<i>Roteamento estático</i>	15
<i>Roteamento dinâmico</i>	16
COMPARTILHAMENTO DE DADOS EM CDNS.....	17
COMPARTILHAMENTO REMOTO TRANSPARENTE.....	17
COMPARTILHAMENTO POR RÉPLICAS	17
COMPARTILHAMENTO POR CACHE.....	18
CONSISTÊNCIA DE DADOS	19
<i>Sincronização periódica</i>	19
<i>Notificação de atualização</i>	20
<i>Atualização consistente</i>	21
CONEXÃO ENTRE SERVIDORES.....	22
OTIMIZANDO O TCP	22
CONCLUSÃO.....	23
BIBLIOGRAFIA	23

Sumário

Com o crescimento exponencial do uso da rede, diversas corporações vêm tentando se aproveitar disso para desenvolver novos negócios, ou mesmo aprimorar os já existentes. Um caso de grande sucesso é o portal de motor de busca Google, que com recursos básicos e um alto poder de busca descentralizada e personalização por região geográfica ou até por país, mudou o conceito de Internet para todos os usuários da grande rede.

Esta pesquisa fornece dados sobre redes CDNs, como são elaboradas, quais as principais vantagens, desvantagens, obstáculos, principais serviços e aonde já são implantadas.

O que é CDN?

CDN (Content Distribution Network) é um conjunto de tecnologias que são utilizadas para prover usuários e clientes de determinados serviços e/ou aplicações com um tempo menor de resposta, possibilitando melhor uso e melhor satisfação do cliente.

As principais razões de sua utilização são:

- Reduz o congestionamento da rede;
- Menos pontos para negociar tráfego e redução de custos;
- Reduz o tempo de espera do cliente;
- Reduz a carga no servidor principal;
- Diminui carga de processamento.

Seu conceito é que possui diversos servidores, ou grupos de servidores, os quais são conhecidos por clusters, espalhados pela rede do cliente. No caso desta rede ser a Internet ou uma empresa de nível global, estes servidores estarão dispostos de forma geográfica; possibilitando uma melhor escalabilidade, disponibilidade e tempo de resposta. Alguns exemplos reais são os servidores DNS roots da Internet e um grupo de servidores de e-mail mundial os quais são conectados por connectors (caso do MS Exchange).

As redes CDNs também são conhecidas como:

- Redes de Entrega de Conteúdo;
- Infra-Estrutura de Internet Inteligente;

- Serviços Substitutos Avançados.

A seguir será mencionado como são os sistemas sem CDN e uma rápida visão sobre a mesma.

Sistema tradicional

O modelo tradicional de comunicação possui somente quatro componentes que são:

- Terminal Cliente;
- Rede de Acesso;
- Backbone;
- Servidor.

O Terminal cliente é o básico, o usuário comum; já a rede de acesso pode ser desde uma conexão discada até uma conexão T1 ou E1, por exemplo. Se esta interligação com o backbone, o qual pode ser a Internet ou a conexão interna de uma empresa através de WAN estiver degradada, não tem saída, o usuário sentirá a perda de desempenho da rede.

Deve-se ter noção do melhor caso do tempo de resposta para que se possa preparar uma solução para possíveis gargalos, como é o exemplo mostrado acima. Para isso testa-se a aplicação através de uma conexão LAN direta ao servidor e verifica-se este tempo. Deve-se levar em conta que o acesso está sendo exclusivo, o que em ambiente real, dificilmente irá ocorrer.

Para se estabelecer um padrão real, deve-se verificar o throughput real da conexão no ambiente de operação.

Com o intuito de minimizar estes problemas, mas sem o uso de CDNs, costuma-se implementar técnicas de QoS (Quality Of Service) as quais são diferenciadas em 2 tipos:

- Técnicas de reserva;
- Técnica de diferenciação de classes.

QoS

Faz-se necessário o uso de uma técnica para garantir a qualidade do serviço, já que todo o tráfego da rede CDN pode ser concorrente com a rede dos usuários; e não pode haver perda de desempenho em nenhuma das partes ao ponto de degradar o serviço.

Técnica de reserva

São técnicas que possuem implementação nativa sobre arquitetura ATM onde o tipo de serviço e a quantidade de recursos necessários são informados por meio de sinalização, a qual pode ocorrer antes ou durante a transmissão da informação.

Esse processo gera problemas em redes que possuem segmentos não confiáveis ou com muita troca de rotas; pois perde-se a confiança nos sinalizadores, sem contar que redes com muitas rotas distintas e com custos diferenciados e que são alterados constantemente geram retardo devido a processamento da análise do tráfego.

Técnica de diferenciação de classe

É um sistema equivalente a utilização do campo ToS do pacote IP, sendo que todos os equipamentos da rede devem ter a capacidade de interpretar tal classificação.

Uma forma de fazê-lo sem a necessidade de toda a rede tê-lo é de que somente o roteador de borda para iria classificar e identificar tais pacote. Esta solução iria economizar nos equipamentos periféricos, porém não iria garantir a continuidade do QoS na rede interna.

Possui a vantagem de ser mais simples e operar sobre protocolos que são padronizados, ao contrário da célula ATM, onde existem diversos fabricantes de equipamentos, porém nenhum padrão estabelecido. Mas possui a desvantagem de não conseguir fornecer as garantias que um sistema ATM consegue, pois não tem como avaliar o fluxo além do que o rodeia e caso hajam muitas classes definidas a possibilidade de problemas é grande, já que os elementos ativos de rede terão que efetuar cálculos constantes, podendo, com isso, degradar a rede como um todo pela latência de processamento.

Técnica CDN

Ela foi desenvolvida para eliminar os problemas anteriores, e com isso gerou novos, os quais veremos adiante.

Seu conceito é utilizar-se de vários servidores, sejam eles primários e secundários ou mesmo conglomerados de clusters de alta performance e alta disponibilidade, para prover serviços para um determinado número de usuários com a mesma informação, sendo que esta pode estar sendo solicitada a um servidor na China e outra no Brasil, simultaneamente.

Um dos primeiros problemas que surgem com isso é a replicação. Para este problema surge uma solução, a qual gera um novo problema que é o seguinte: a atualização pode ser feita de forma dinâmica, ou seja, se o servidor secundário não possuir a informação, ele irá ao principal buscá-la. Com isso gera-se o tráfego pelo backbone que cria o tão repudiado gargalo, fazendo com que a aplicação responda lentamente.

Para este problema foi dada uma solução a qual resolve os problemas: nos servidores principais e secundários, coloca-se o conteúdo estático, seja este um site WEB, sistema de correio, DNS, por exemplo; e toda a consulta dinâmica é feita a um único servidor.

Para melhor entendimento, entenda um site de WEB. Este possui textos e figuras estáticos, arquivos com extensão jpg, gif, doc, html. Mas o acesso a uma consulta deve ser feito ao servidor de base de dados principal. Com isso, o tráfego pesado da rede será fornecido direto pela rede CDN, e o tráfego eventual de consultas, o qual é muito leve, será feito sobre demanda.

Outro exemplo: um sistema internacional de e-mail sobre o MS Exchange. Cria-se diversos routing groups e é feita a configuração, conforme política interna de que o acesso, no caso encaminhamento de mensagens para fora da filial, só será feito em determinados, com isso, libera-se o tráfego para outros fins enquanto que a rede que distribui os e-mails não afeta a corporação. E caso uma conexão fique limitada, esta aplicação de correio entende isso e encaminha o tráfego por outra rota.

Um caso particular existe quando tratamos de aplicações legadas as quais foram desenvolvidas sem levar em conta facilidades como a CDN. Para estes casos, pode-se fazer uso de um servidor (de substituição) secundário reversa,

que é um servidor que controla o serviço com interatividade, ou modificar a aplicação.

Levando-se o exposto até o momento, pode-se concluir que as aplicações que mais tiram proveito das tecnologias CDNs são as multimídias e as páginas WEB.

Já serviços como chat não podem utilizar estas técnicas diretamente ao produto da conversação, seja ela por meio de voz ou não, porém pode-se aliviar a carga nos servidores por meio de CDN, onde cada grupo de salas ficará operacional e um determinado servidor e outro grupo e outro e assim por diante.

Outro péssimo candidato a redes CDNs são sites que necessitam de alta segurança, pois o custo desta implementação e da manutenção ficará muito elevado, já que toda a segurança aplicada ao site principal, deverá ser aplicado, também, aos secundários.

O que deve ser analisado?

Antes de sair implementando uma rede CDN, deve ser analisado alguns pontos de viabilidade e compatibilidade com a tecnologia. Estes são:

1. Analisar o custo x benefício do projeto; em outras palavras, comparar os gastos de implementação e manutenção contra os de aumentar o parque atual do servidor e do link;
2. A aplicação que será colocada na CDN aceita escalabilidade? Com esta pergunta, pode-se determinar se o parque será de terminais isolados ou de clusters;
3. Existe possibilidade de roteamento para os sites remotos? Como isto pode ser feito? É viável a criação de rotas estáticas nos servidores e elementos ativos?
4. Quais serão às técnicas de cache implementadas, replicação e armazenamento nos servidores secundários?

Escalabilidade

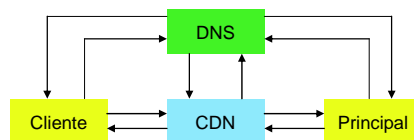
Uma aplicação pode ser escalável sem a necessidade de CDNs, mas isto onera em alto custo de servidores, o link de interconexão é de custo acessível e pode ter sua capacidade aumentada facilmente. Porém mesmo com todos esses recursos, se existir um link que não possua o nível de disponibilidade necessária, é necessário fazer um uso de várias tecnologias e métodos.

Nestes casos, o grande desafio é passar total transparência para o usuário final, de forma que este não perceba qual servidor está sendo usado e nem que há vários deles.

Em cima dessa filosofia, pode-se preparar a aplicação para permitir a interação ou não do cliente. Neste caso, o que se quer dizer sobre interação do cliente é referenciado a aplicação do cliente sem a intervenção do usuário, já que tais ações não são recomendadas devido a falhas de segurança e que um conjunto de escolhas erradas de diversos usuários, pode vir a derrubar um site.

Um exemplo deste tipo de rede é, novamente o sistema de DNS, sendo que para melhor visualização, cita-se o caso de uma empresa que possui vários servidores DNS, como uma zona integrada do Active Directory, onde o servidor mais disponível irá automaticamente responder a solicitação do cliente e os outros serão desconsiderados.

Este é um esquema básico do funcionamento de uma rede CDN com base em DNS.



Outro método é o de acesso central a aplicações integradas. Ele utiliza a tecnologia AIS para coletar dados de diversas fontes, independente de suas tecnologias e publicá-las para os usuários de forma padronizada e inteligível, independentemente do tipo de visualizador que o cliente utilize. Esta ferramenta é ótima para sites de e-commerce, que com isso podem distribuir suas bases de dados sem comprometer a sua utilização.

Balanciamento de carga

É a técnica mais recomendada para aplicações estáticas como páginas WEB, podendo ter outra aplicação servindo a todos os servidores WEB, como, por exemplo, um banco de dados.

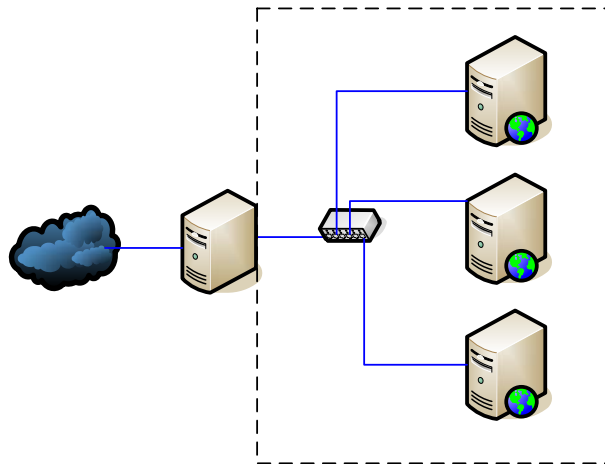


Figura 1

Sendo assim, este modelo pode ser utilizado para qualquer aplicação estática, como para servidores de arquivos que são somente utilizados para consulta por um grande número de usuários, sendo sua alteração possível somente para poucos.

Com isso surge à dúvida de como o front-end irá lidar com as solicitações para não sobrecarregar um determinado servidor. A resposta está na aplicação que está fazendo a gestão do balanceamento as quais são feitas por meio de políticas internas, elas podem ser por capacidade do link, capacidade de processamento dos servidores, tipo de arquivo solicitado dentre outras que podem ser elaboradas. Tudo depende da política de gestão do cluster.

Porém nem tudo é alegria. Esta solução, apesar de muito satisfatória em várias situações, pode ser uma grande dor de cabeça se o front-end vier a falhar, tudo devido ao fato deste ser um ponto de falha central. Para minimizar este problema, implementa-se um cluster de alta disponibilidade entre os front-ends conforme mostrado na Figura 2. Neste caso, pode-se implementar diversos tipos de ferramentas ou scripts. Um exemplo é por meio de PINGs (ICMP), neste caso também conhecido como *heartbeat*, que pode verificar se o

balanceador principal está operacional ou não e efetuar a eventual comutação do serviço. Outra possibilidade é por meio de protocolo de roteamento BGP.

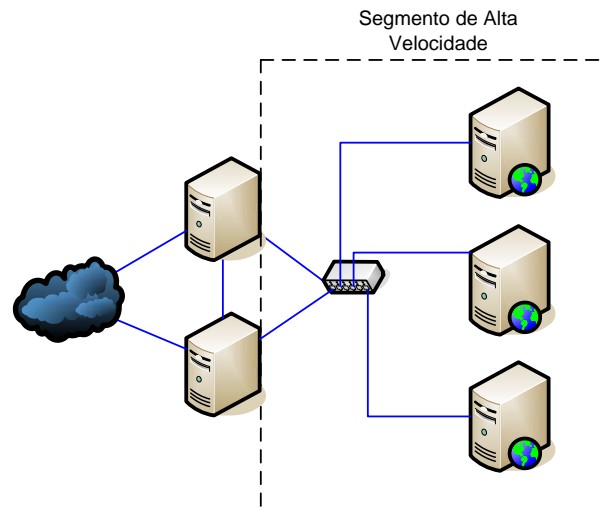


Figura 2

Aumentando o desempenho

Para melhorar o desempenho de um balanceador de carga, utiliza-se o serviço de NAT de forma que este efetue a distribuição para os servidores das solicitações no sentido cliente para o servidor, isto porque estas são de baixa carga possibilitando maior número de conexões; enquanto que a saída para o cliente deve ser feita através de outro servidor NAT ou diretamente do servidor para a rede do solicitante. Um exemplo desta solução é mostrado na Figura 3

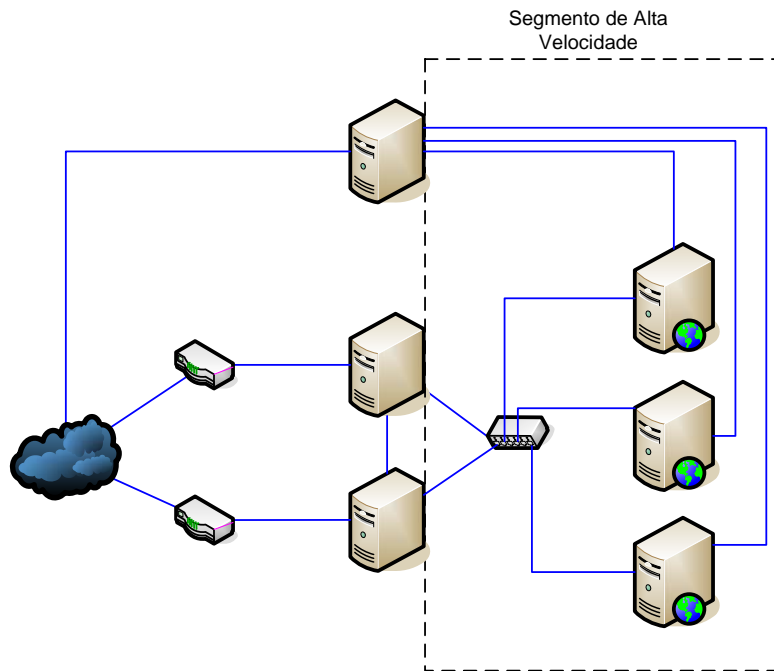


Figura 3

Com isso, surge a necessidade de se implementar uma solução que mantenha o usuário sempre com suas informações de estabelecimento de sessão de browser ou aplicação, segurança e rastreo de atividades dos usuários, caso seja necessário. Assim, faz-se uso de soluções como de cookies, modificação de URL por meio de identificadores e armazenamento compartilhado de sessão.

- Cookie

É um tipo de banco de dados que só possui ponteiros para as informações que estão armazenadas no servidor; isto devido ao fato que o tamanho de cada arquivo de cookie é muito limitado. Ele possibilita que a sessão do usuário seja armazenada por um período de tempo determinado pelo servidor, mesmo se a aplicação for encerrada.

- Identificadores

É uma solução que não depende de cookie, apesar de possuir funcionalidade semelhante. A diferença é que a sessão é curta e se a aplicação for encerrada, o usuário terá que realizar todos os seus processos novamente.

Seu funcionamento não é feito sobre arquivos no cliente. Ele gera um código identificador aleatório para cada sessão estabelecida e modifica a forma que o usuário visualiza a URL.

Nestas condições, se o usuário digitar qualquer URL no browser que está com o ID, a sessão será terminada.

- **Armazenamento Compartilhado**

É semelhante ao cookie, porém todo o controle é feito diretamente em um servidor, o qual irá ser responsável de responder todas as solicitações feitas pelos aplicativos. O problema é a consistência dos dados, pois só poderá existir uma base para todos os servidores e poderá ser armazenado, qualquer tipo de informação; por isso é uma solução mais cara, pois depende de hardwares mais eficientes e redes capazes de suportar o tráfego gerado por vários servidores acessando o mesmo banco de dados e efetuando alterações simultâneas.

Tolerância à falhas

Conforme já mencionado, se o balanceador de carga falhar, toda a estrutura falha. Para resolver este problema, coloca-se redundância nos servidores de front-end e de back-end, caso existirem. Na Figura 3 é mostrado como ficaria o sistema com dois balanceadores de carga. Para que os servidores com tolerância possam saber quando um está com mau funcionamento ou mesmo inoperante, eles precisam ou de uma aplicação externa a eles que monitorem e façam alterações com forme a necessidade.

Pode ser feito, também, uma solução com baixa necessidade de processamento o qual será feito por meio de uma conexão entre os servidores para efetuar comutação como explicado na sessão de balanceamento de carga.

Balanceamento por difusão

Não existe um servidor no front-end; este é substituído por um roteador e todos os back-ends terão o mesmo IP (método ANYCAST, tecnologia IPV6 ou via aplicação). Neste esquema, cada servidor só processa um conjunto de pacotes.

O roteador consegue acessar cada servidor por meio de rotas estáticas mapeadas pelo MAC, o IP que é publicado é o do cluster. A filtragem de encaminhamento para um ou outro servidor pode ser feita por meio de regras booleanas para distribuição de carga entre os servidores.

Nesta solução, não é necessário o uso de NAT, mas o problema está no aumento de processamento gerado nos servidores, já que o roteador só faz encaminhar os pacotes, o que não ocorre com o uso de um front-end, onde o dado é tratado.

DCN em site único

Recomenda-se o uso de front-end, mas não é o recomendado em redes geograficamente distribuídas devido à que não se tem como assegurar que 100% da rede esteja perfeitamente operacional e nem que os seus meios de conectividade estejam sempre dentro do contratado, mesmo amarrando isso a SLAs específicos, mas isto aumenta em muito o custo; e se um dos meios de acesso caminhos estiverem degradados, poderá afetar todo o serviço.

Para otimizar esta solução, utilização o sistema de roteamento triangular, NAT onde o servidor responde direto ao cliente sem passar pelo servidor de tradução.

Triangulação

O balanceador não possui nenhuma informação do site, exceto a localização lógica dos servidores e por meio de uma estrutura DNS (IP, MAC, Domínio root i.e .br, .tk, .com); com isso ele encaminha a solicitação do cliente para o servidor adequado, mas não é utilizado para a comunicação reversa, ou seja, do servidor para o cliente.

O problema gerado é o aumento do custo e por esse único motivo, deve ser considerado como uma solução emergencial de curto prazo.

Operação das CDNs

Redes CDNs operam, basicamente em dois métodos: ativo e passivo. No modo ativo, o elemento responsável pelo direcionamento das requisições faz

buscas por informações dos clientes por meio de algum agente, normalmente JAVA; já no método passivo a escolha é feita com base em sistema de rotas.

O método ativo gera mais tráfego na rede podendo não ser uma boa escolha se o conteúdo existente nos servidores não for pesado o suficiente para justificar sua implantação.

Já utilizando o esquema passivo, poderá se evitar o acréscimo de tráfego se a tabela de rotas for estática, mas se for dinâmica, todos os elementos terão que aprender as rotas com os outros, o que também irá gerar tráfego extra, mas mesmo assim, este será menor do que o sistema ativo.

Esquemas ativos para roteamento em CDNs

É utilizado, basicamente, o ICMP (PING) somado a outros recursos. O PING irá verificar o tempo de resposta dos servidores, mas como ele não é prioritário na rede, poderia mascarar as informações se fosse usado sozinho; por isso são utilizadas aplicações extras que serão responsáveis por mesurar todos os outros recursos necessários para que a escolha do servidor seja feita de forma correta. Estes podem ser a quantidades de sessões abertas, recursos disponíveis, níveis de threads de um determinado processo ou qualquer outro que o desenvolvedor queira.

Já para o lado cliente, não é recomendado o uso do PING já que isto irá aumentar o retardo e o tráfego na rede; e sendo que acrescentando-se mais uma variável ao processo de escolha do servidor o tempo de resposta irá aumentar, podendo ficar o tempo e carga do processo não satisfatórios.

Porém se pudesse ser utilizado a informação do cliente para dar uma resposta mais precisa sem aumentar a banda utilizada e consequentemente o tempo de resposta, seria uma ótima opção.

Para que isto possa ser efetivamente implantado, faz-se uso de agentes que coletam informações do DNS que provê serviços a estes clientes. Como o DNS só gera tráfego quando sua informação tem a validade expirada, o tráfego extra gerado já é parte integrante da rede de forma nativa.

Existe, porém, um empecilho para as soluções de PING e DNS que é a segurança e a compatibilidade. O uso destas técnicas, em redes abertas, não consegue obter bons resultados já que firewalls irão bloquear PINGs e solicitações de acesso remoto para coleta de informações e quando não

existirem firewalls, fica a dependência dos browsers, pois estes têm que ser compatíveis com a aplicação desenvolvida.

Ou seja, este método só é válido se for possível garantir a compatibilidade entre as versões em uso e o software utilizado e permissões de uso de recursos via rede.

Tabela de rotas em CDNs

Por meio de tabelas de rotas pode-se localizar o melhor servidor para determinado cliente. Bastaria saber qual a localização do cliente e utilizar os dados de IP, domínio ou MAC para determinar o melhor ponto de provimento de serviço; sendo que o controle pode ser feito utilizando a técnica do LRU (*Last Resource Used*).

Roteamento estático

Pode ser utilizado se a topologia usada pelos clientes para acessar a rede CDN for conhecida. Sendo esta condição atendida, bastaria que mapear a rede para que determinasse o servidor mais próximo de cada cliente, o que pode ser feito atribuindo um custo para cada conexão. Com este dado, o melhor servidor poderá ser escolhido.

Já para a Internet que possui muitos trechos desconhecidos fica extremamente complexo utilizar esta solução de forma imediata. Faz-se necessário o uso de métodos diferenciados para a localização e para mapear as rotas.

Uma forma de descobrir as rotas da Internet é através do BGP que conectam cada um dos sites. Este protocolo de roteamento possui informações sobre cada sub-rede que está conectada a ele e se há domínios administrativos diferentes no caminho. Essas informações fornecem um ponto de início para a elaboração das tabelas. Combinando as informações de cada site e rotear cada sub-rede nas tabelas BGP para o site que utilize o menor número de domínios administrativos diferentes para cada cliente irá fornecer esta facilidade para usuários via Internet.

Porém o BGP só fornece conectividade baseada no melhor caminho (quantidade de saltos) e não no mais rápido. Para mesclar as duas, terá que ser utilizado uma aplicação extra para monitorar o desempenho de certos

pontos chaves da rede e somar estes resultados a tabela BGP para que o servidor escolhido seja o melhor em quantidade de saltos e velocidade de saltos.

Roteamento dinâmico

A principal razão de se utilizar esta solução é pelo fato de que o roteamento estático pode não conseguir enviar uma solicitação para o melhor ponto; isso porque o retardo da rede que é variável, a mudança das tabelas de rotas e o congestionamento são constantes logarítmicas. Já nas tabelas dinâmicas, este problema é minimizado, pois suas rotas são atualizadas automaticamente.

Para gerar as tabelas de roteamento dinâmico é necessário saber de onde as requisições dos clientes são encaminhadas; coletar estatísticas entre cada sub-rede e os sites CDNs; e combinar as estatísticas de desempenho de cada site para gerar a tabela. Com isso, a tabela dinâmica também pode se aproveitar de logs e informações de tráfego de rede que podem ser obtidas de um site on line.

Neste caso, pode-se gerar tabelas nos elementos de forma centralizada, distribuída ou mista. A melhor forma depende da quantidade e da facilidade de gestão.

Considerando um único ponto de roteamento, este seria responsável de mapear e obter todas as estatísticas de cada site para poder gerar a tabela de rotas para cada um dos sites e ele teria que ter acesso direto a todas as informações geradas pelos sites.

Sendo os pontos de roteamento distribuídos, significaria que existe um grande número de sites espalhados geograficamente. Assim, todos os pontos de roteamento têm que poder trocar informações entre si; para possibilitar a troca de informações, pode ser utilizado o método de multicast, caso o broadcast não seja permitido na rede, usa-se o modelo de rede sobre overlay.

No modo misto, cada ponto de roteamento distribuído atua como um centralizado local, mas tem que encaminhar suas rotas para os outros. Esta técnica pode utilizar protocolos com base no estado do link (OSPF) ou sobre distância vetorial (RIP).

Compartilhamento de dados em CDNs

Pode ser feito de várias formas, mas basicamente é dividido em algumas categorias que são compartilhamento remoto transparente; réplicas; e cache.

Em todos os casos, é necessário algum tipo de controle operando sobre a técnica utilizada para o armazenamento. Costuma-se usar sistema de banco de dados para isso, e o LDAP é um bem recomendado pelas suas características. Também é utilizado o sistema de banco de dados relacionais e distribuídos, sendo que no último o problema é com a replicação, já que cada site deverá ter uma cópia dos dados ou só serem utilizados para consulta e todas as alterações devem ser feitas diretamente no servidor principal. O armazenamento de consultas como as feitas sobre ASP também podem necessitar de cache, caso um determinado item tenha muitas consultas iguais e diversos outros métodos e idéias de como e do que armazenar.

Compartilhamento remoto transparente

A base de dados é acessada em um único ponto, o qual normalmente é o site principal e os substitutos o acessam. O acesso é feito por meio de sistemas de RPC (*remote procedure call*). Este método é fácil de ser utilizado pois basta que os servidores substitutos acessem os dados através de drive de rede mapeado ou usar o RPC para esconder o local de armazenamento da aplicação, dando mais segurança ao serviço; costuma-se utilizar a programação em JAVA juntamente com o JDBC para fazer isto.

Fazendo-se uso deste método, elimina-se o problema de consistência da base do banco de dados; mas o problema reside na largura de banda que será consumida.

Para determinar se a latência é aceitável para o método, deve-se mesurar o tempo total e verificar se o tempo de processamento e de resposta do banco de dados é referente a uma parte pequena do tempo total de processamento.

Compartilhamento por réplicas

Cada site possuirá uma cópia idêntica da que existe no site principal. Com isso surge o problema da consistência dos dados, já que cada site tem que poder modificar os dados conforme seja necessário.

Com isso surgem dois tipos de acessos ao sistema de réplicas que são leitura e formatação; e modificação dos dados.

Para o primeiro caso, não há muita alteração, já que baseia-se em leitura e formatação da exibição do conteúdo. Já para o segundo, o sistema é bem mais complicado já que se deve garantir a sincronização entre as atualizações; e garantir a consistência dos dados em todos os sites. Com isso surge a necessidade da elaboração de um esquema de replicação de dados entre os servidores.

Esta solução é ideal para dados estáticos. Caso haja dados maleáveis, pode ser interessante mesclar as duas técnicas citadas anteriormente.

Compartilhamento por cache

Existem duas formas para que os servidores de uma rede CDN obtenham os dados para armazenamento em cache; estes são os modos de *pull* e *push*. No primeiro, os servidores solicitam os dados para o servidor principal conforme a demanda; no segundo, o servidor mestre envia de forma periódica os dados para os servidores substitutos.

No modo de cache, existe uma vantagem para rede CDNs que vai além dos métodos já comentados que é a velocidade de processamento. Isto é devido a que como só é utilizada a memória principal (RAM), a memória secundária (discos físicos) não precisa ser acessada, aumentando significativamente o tempo de resposta para solicitações.

Para a rede CDN como um todo, o gerenciamento do cache, independente do método, é que será o fator determinante. Como a memória virtual deve ser excluída de uso em quaisquer condições, podem-se escolher os métodos de FIFO (*First in First Out*) ou LRU (*Last Resource Used*); sendo que o último é o mais recomendado, já que irá conseguir determinar quais os endereços de memória são mais utilizados.

O acesso aos dados do cache não é o único método, já que será quase impossível manter todos os dados em memória todo o tempo. Normalmente o conteúdo que é enviado é somente o mais acessado e pode-se determinar que determinadas áreas só poderão ser acessadas através do servidor principal para manter a consistência dos dados.

Um ponto importante é que para determinados tipos de solicitações de clientes, o método de cache de nada servirá, como exemplo temos os arquivos multimídia, já que seu tamanho é muito grande, o que irá sobrescrever toda a memória em cache; porém existe uma outra condição para o conteúdo multimídia utilizar o cache. Este método é o *streaming*.

Realizando o armazenamento de cache em modo streaming, significa dizer que o cliente não irá baixar todo o conteúdo de uma só vez. O início do arquivo solicitado é enviado ao cliente enquanto que os outros quadros serão armazenados em cache e repassados ao cliente conforme os quadros que estão no cliente sejam executados.

Após a execução, o cache pode ser limpo, passado para outro em memória secundária caso seu uso seja freqüente ou manter os primeiros segmentos do arquivo.

Consistência de dados

Uma rede CDN pode trabalhar com qualquer tipo de dado, e como todo dado possui um determinado valor relacionado com seu conteúdo, é necessário que não haja inconsistência, ou se esta ocorrer, que seja dentro de padrões aceitáveis. Para atender a esta necessidade que já fora abordada anteriormente, são utilizadas diversas técnicas para manter os dados sempre atualizados; manter uma inconsistência por um período máximo pré-determinado; ou dados que simplesmente podem coexistir com inconsistência.

Estes esquemas são os seguintes: sincronização periódica; notificação de atualização; atualização consistente.

Sincronização periódica

Sua teoria é bem simples. É baseada em que todas as cópias espalhadas pela rede serão atualizadas em períodos pré-determinados ou na ocorrência de um determinado evento, sendo que, em ambos os casos, podem ocorrer em todos os sites ou somente em um; tudo dependerá da forma que de replicação que fora implantada. Quando da ocorrência de solicitação de alteração de dados, esta é normalmente encaminhado diretamente ao servidor principal, sendo esta alteração repassada aos outros sites.

Para aumentar a confiabilidade da aplicação, pode ser utilizando a técnica de *read-thru* para que o dado que o cliente está solicitando seja comparado com o do servidor principal e com isso sempre manter o dado atualizado. Caso o dado requisitado seja pequeno, não será notado diferença para o cliente, mas se o tamanho deste for grande, o simples ato da comparação irá garantir a informação mais atualizada sem que afete o tempo de resposta.

A ação de comparar os dados é feita com base em relógio, ou seja, é necessário que a hora de todos os servidores esteja sincronizada.

Este método pode ser utilizado com o compartilhamento por réplica e por cache e é recomendado para aplicações que aceitem ficar durante determinado tempo com inconsistência em suas informações.

Notificação de atualização

Diferentemente da atualização periódica, esta técnica divulga na rede, através do servidor principal ou de outro site, uma alteração nos dados e informa qual deles estão com as informações mais atualizadas. Após o recebimento da notificação, os sites que não possuem os dados mais atuais, podem considerar suas informações como não validadas até que estas sejam atualizadas. Porém na própria notificação pode ser encaminhado o dado de atualização, mas este método pode ser uma brecha para expor toda a rede a uma falha de segurança, já que o servidor é confiado e o dado não tem como ser confirmado o que possibilita o tipo de ataque do “homem do meio” e por em risco toda a estrutura. Para este método, também existe o AIS (Application Integration Software). Ele é uma aplicação dedicada que deve ser adquirida e que controla quais dados foram alterados, enviando notificações para os clientes. Em sistemas ponto-a-ponto, não é recomendado que hajam mais de 5 clientes. Para redes geograficamente distribuídas, é necessário um HUB AIS.

Em uma outra situação, pode-se fazer a propagação em estilo multicast, já que existe a possibilidade de que nem todos os servidores da rede utilizem todos os dados, economizando a largura de banda. Este controle pode ser feito de diversas formas, porém se feito com muitas tabelas de acessos, pode piorar o tráfego ao invés de otimizá-lo, já que varias notificações seriam enviadas para os mesmos sites.

Esta técnica reduz o tempo de inconsistência para o patamar de milissegundos, sendo este causado pelo tempo de aviso e transferência de dados atualizados. Porém existem situações que não comportam a latência da rede, como saldos de aplicações bancárias, para estes casos implementa-se o método de atualizações consistentes.

Atualização consistente

É utilizado em situações onde inconsistência não é permitida de forma alguma. A técnica utilizada é de banco de dados distribuído, sendo que é necessário manter a atomicidade, serialização e consistência dos dados. Isto significa que todos os sites devem trabalhar de forma serial e todos devem possuir exatamente a mesma informação, sendo que esta deve ser confirmada ou ocorre o efeito *roll-back*.

Para forçar a serialização, é utilizada a técnica de bloqueio que é como se fosse uma ordenado que irá controlar em que ordem cada servidor irá executar operações de leitura, escrita ou ambos.

Os bloqueios podem ser elaborados por classes de informações, base de dados ou qualquer outro que seja de interesse; quanto maior for o detalhe da elaboração dos bloqueios o sistema fica mais complexo, mas também mais ágil.

Em uma rede CDN, esta técnica também é conhecida por submissão (*commit*) e ocorre em duas fases que são a preparação e a submissão. Na primeira, o site que deseja realizar alguma operação envia uma solicitação em multicast para todos os servidores da rede para que estes obtenham um bloqueio. Todos em sites da rede devem responder com uma confirmação positiva para a solicitação de bloqueio, caso contrário, a operação é cancelada; na segunda, que depende de todos os sites terem dado uma confirmação positiva para a primeira, a alteração é feita pelo site solicitante e a informação é replicada para todos os outros sites. Caso haja falha ou se nenhum ACK tenha sido recebido, todo o processo é cancelado e o *roll-back* é executado em todos os sites.

O grande problema desta solução é o consumo de banda dobrado que faz com que quase 50% da banda sejam perdidas só com confirmações. É um método bem eficaz e confiável.

Conexão entre servidores

É necessário estabelecer formas de conectividade entre os sites de uma rede CDN de modo a não prejudicar o tráfego da rede convencional nem às solicitações feitas por clientes a um determinado recurso provido por um servidor da CDN.

O ideal seria colocar os servidores para se comunicarem através de uma rede paralela a rede de uso normal, mas isto seria muito custoso, seja por meio convencional ou não.

Para aproveitar a rede em uso sem perder o seu desempenho atual ou minimizando os impactos, são utilizadas algumas técnicas que modificam aspectos que são os normais dentro de uma rede convencional, já que muitos destes só pioram o desempenho de uma rede CDN.

Então os sites da CDN estarão utilizando os mesmos recursos que um usuário comum usaria para enviar um arquivo. Porém, para otimizar estas comunicações entre os sites, implementa-se modificações na estrutura de comunicações, faz-se uso de QoS, ou pode-se até modificar a estrutura interna das aplicações.

Como o TCP de uso mundial possui uma estrutura que foi concebida para manter a rede estável e controlar e garantir a qualidade, gera um *overhead* de tráfego que é um diferencial negativo para CDNs, assim como a técnica de janela deslizante e a característica de queda em 50% do tráfego em caso de perda de pacote.

Otimizando o TCP

Para minimizar o problema de comunicações entre os servidores da CDN com o servidor principal, onde para cada solicitação é necessário estabelecer uma comunicação, e isto irá gerar uma latência significativa na rede se forem milhares de comunicações simultâneas, utiliza-se a técnica da comunicação persistente, onde a sessão é estabelecida uma única vez e todas as outras requisições passam por este mesmo canal. Esta técnica ganha mais força se os dados solicitados necessitem de pequena largura de banda.

Já para o caso da janela, pode-se modificar a estrutura da aplicação ou mesmo a do TCP para que não reduza a janela ou mesmo que nem a verifique

e fique sempre enviando os dados em velocidade máxima sem se preocupar com os elementos periféricos. Esta última solução seria agressiva demais podendo comprometer toda a rede; porém se o aumento do tráfego enviado pelos servidores for aumentado juntamente com a aplicação de uma política de QoS externa, como é o caso de aplicações multimídia que usam o RTCP, poderia se obter uma boa melhora de desempenho.

Outra situação seria o aumento de conexões simultâneas sem que as propriedades do TCP fossem alteradas. Como o buffer da conexão é padronizado e não utiliza todos os recursos do hardware, poderiam ser criadas outras conexões físicas onde o seu somatório daria o produto de uma única conexão. Bastaria que fosse utilizada uma determinada aplicação para controlar e montar todos os pacotes.

Conclusão

Toda e qualquer aplicação pode ser utilizada em uma rede CDN, o problema é determinar suas dependências e pré-requisitos que podem gerar falhas ou uma grande necessidade de trabalho administrativo para mantê-las operacionais.

Bibliografia

Dinesh C. Verma, Redes de distribuição de conteúdo; editora CAMPUS

Telematica Instituut, Content Distribution Networks

<http://www.idc.com/>

<http://canguru.ufscar.br>