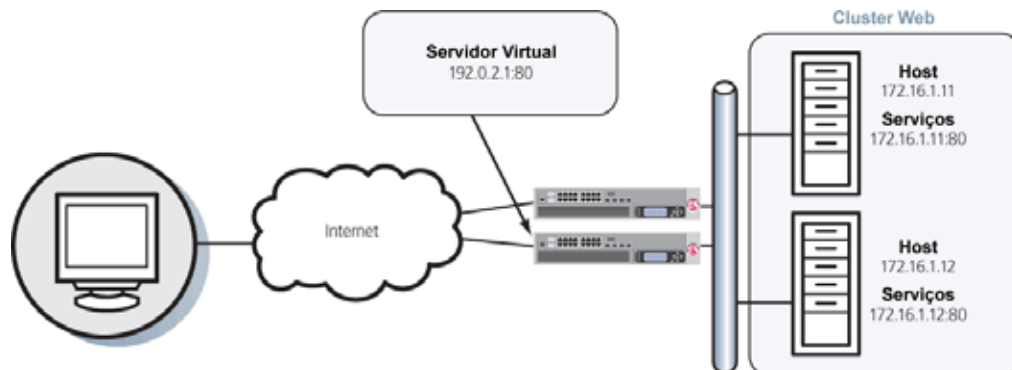


Balanciamento de carga: Conceitos básicos

Introdução A tecnologia de balanceamento de carga está viva e está bem; de fato, ela é a base sobre a qual operam os application delivery controller (ADCs). A disseminação da tecnologia de balanceamento de carga não significa, porém, que ela seja universalmente compreendida nem que seja discutida normalmente, a não ser de um ponto de vista focalizado em redes. Em uma exploração mais aprofundada do assunto, este documento tenta desvendar um pouco do mistério e da magia das práticas básicas de balanceamento de carga.

Hardware de Balanceamento de carga baseado em rede

A segunda geração do balanceamento de carga de finalidade específica (que se seguiu aos sistemas proprietários baseados em aplicativos) surgiu na forma de dispositivos baseados em rede. Esses são os verdadeiros avós dos controladores de distribuição de aplicativos de hoje. Como esses dispositivos eram neutros em relação a aplicativos e residiam fora do servidor de aplicativos, eles podiam fornecer o balanceamento de carga usando técnicas diretas de rede. Em síntese, esses dispositivos ofereciam um endereço de servidor virtual para o mundo exterior e, quando os usuários tentavam se conectar, ele encaminhava a conexão ao servidor verdadeiro mais apropriado, fazendo uma tradução de endereços de rede (NAT) bidirecional.



Terminologia Básica

Tudo seria mais simples se todos usassem o mesmo dicionário; infelizmente, cada fornecedor de dispositivos de balanceamento de carga (e, por sua vez, ADCs) parece usar uma terminologia diferente. Com uma pequena explicação, entretanto, a confusão em torno desse assunto pode ser facilmente esclarecida.

Nó, Host, Membro e Servidor

A maioria dos balanceadores de carga usa o conceito de nó, host, membro ou servidor; alguns usam todos eles, significando coisas diferentes. Há dois conceitos básicos que todos eles tentam expressar. O primeiro, normalmente chamado de nó ou servidor, é a ideia do servidor físico que recebe tráfego do balanceador de carga. Isso equivale ao endereço IP do servidor físico e, na falta de um balanceador, seria o endereço IP para o qual o nome do servidor seria atribuído (por exemplo, www.exemplo.com). No resto desse documento, vamos nos referir a este conceito como "o host". O segundo conceito é o membro (infelizmente, também chamado de nó por alguns fabricantes). Um membro normalmente é mais bem definido do que um servidor ou nó, visto incluir a porta TCP do aplicativo que recebe o tráfego. Por



exemplo, um servidor chamado `www.exemplo.com` pode ser atribuído para o endereço `172.16.1.10`, que representa o servidor/nó, e pode ter um aplicativo (um servidor web) sendo executado na porta TCP 80, formando o endereço do membro `172.16.1.18:80`. Simplificando, um membro inclui a definição da porta do aplicativo, além do endereço IP do servidor físico. No resto desse documento, vamos nos referir a este conceito como "o serviço".

Por que tanta complicação? A distinção entre o servidor físico e os serviços de aplicativos sendo executados nele permite que o balanceador de carga interaja individualmente com os aplicativos, em vez de interagir com o hardware subjacente. Um host (`172.16.1.10`) pode ter mais de um serviço disponível (HTTP, FTP, DNS e assim por diante). Ao definir cada aplicativo individualmente (`172.16.1.10:80`, `172.16.1.10:21` e `172.16.1.10:53`), o balanceador de carga pode aplicar um balanceamento de carga exclusivo e também o monitoramento de estado (que será discutido mais tarde) baseando-se nos serviços e não no host. Entretanto, ainda há momentos em que interagir com o host (como no monitoramento de estado em baixo nível ou ao desativar um servidor para manutenção) pode ser muito conveniente.

É importante lembrar que a maioria das tecnologias de balanceamento de carga usam algum conceito para representar o host, ou servidor físico, e um segundo conceito para representar os serviços nele disponíveis.

Grupo, Cluster e Fazenda

O balanceamento de carga permite que você distribua o tráfego de entrada entre vários destinos de segundo plano. É necessário, portanto, adotar o conceito da coleção de destinos de segundo plano. Os clusters, como os chamaremos a partir de agora, são coleções de serviços similares disponíveis em um número qualquer de hosts. Por exemplo, todos os serviços que oferecem a página web da companhia seriam reunidos em um cluster chamado "Página web da companhia" e todos os serviços que oferecem comércio eletrônico seriam reunidos ou colecionados em um cluster chamado "eCommerce".

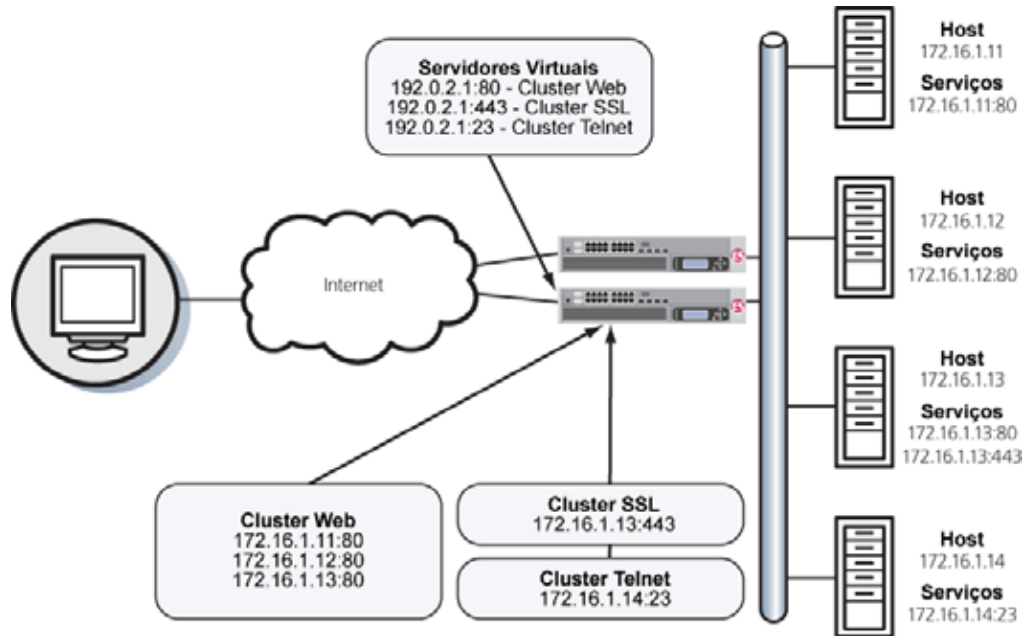
O ponto principal é que todos os sistemas têm um objeto coletivo, que se refere a todos os serviços similares. Isso torna mais fácil trabalhar com eles como uma única entidade. Esse objeto coletivo é quase sempre composto de serviços e não de hosts.

O servidor virtual

Embora não seja sempre o caso, hoje em dia há poucas divergências sobre o termo servidor virtual ou "Virtual". É importante notar que, como definição de serviços, o servidor virtual normalmente inclui a porta do aplicativo e o endereço IP. Como a maioria dos fornecedores usa um servidor virtual, nós continuaremos a usar essa terminologia nesse documento, embora o termo "Serviço virtual" fosse mais adequado à convenção IP:Porta.

Somando tudo

Reunir todos esses conceitos é a idéia principal do balanceamento de carga. O balanceador de carga oferece servidores virtuais para o mundo externo. Cada servidor virtual aponta para um cluster de serviços que reside em um ou mais hosts físicos.



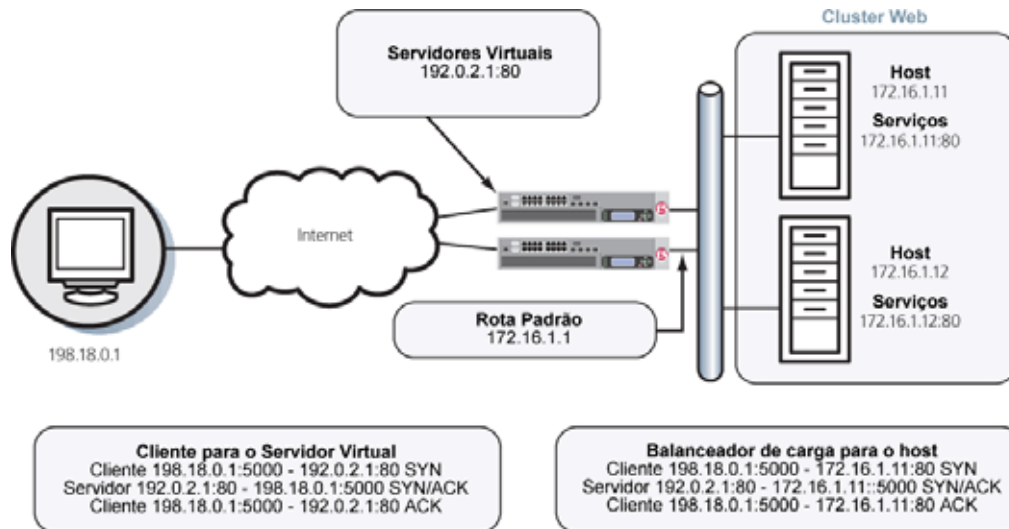
Embora o exemplo demonstrado não represente uma implementação real, ele ilustra a estrutura básica para continuarmos a nossa discussão sobre os conceitos básicos do balanceamento de carga.

Balanceamento de carga - conceito básico

Agora que temos um vocabulário comum, podemos começar a examinar a transação básica de balanceamento de carga. Conforme demonstrado, o balanceador de carga normalmente ficará alinhado entre o cliente e os hosts que fornecem os serviços desejados pelos clientes; como muitas outras coisas no balanceamento de carga, isto não é uma regra, mas uma boa prática em uma implementação típica. Também vamos considerar que o balanceador de carga já está configurado como servidor virtual que aponta para um cluster formado por dois pontos de serviço. Nesse cenário de implementação, também é normal que os hosts tenham uma rota de retorno que aponte de volta para o balanceador, para que o tráfego de retorno seja processado por ele, no caminho de volta para o cliente.

A transação básica de balanceamento de carga funciona assim:

1. O cliente tenta conectar ao serviço no balanceador de carga.
2. O balanceador de carga aceita a conexão e, após decidir qual host deve receber a conexão, muda o IP de destino (e talvez a porta) para combinar com o serviço do host selecionado (note que o IP de origem do cliente não é alterado).



- O host aceita a conexão e responde ao cliente original pela rota padrão, o balanceador de carga.
- O balanceador de carga intercepta o pacote de retorno do host e muda o IP de origem (e provavelmente a porta) para combinar com o IP e a porta do servidor virtual, que encaminha o pacote de volta para o cliente.
- O cliente recebe o pacote de retorno, pensando que ele vem do servidor virtual, e dá continuidade ao processo.

Este exemplo simples é bastante direto, mas há alguns elementos importantes que podemos destacar. Primeiro, para o cliente, a comunicação parece ser direta - ele manda pacotes para o servidor virtual e recebe respostas. Em segundo, o NAT atua na conexão. É aqui que o balanceador de carga substitui o endereço IP de destino enviado pelo cliente (o do servidor virtual) com o IP de destino do host escolhido para balancear a carga do pedido. O terceiro passo é a segunda metade desse processo (a parte que torna o NAT "bidirecional"). O IP de origem do pacote de retorno do host será o IP do host; se esse endereço não for alterado e o pacote for simplesmente encaminhado para o cliente, o cliente iria descartá-lo, pois estaria recebendo um pacote de alguém a quem não solicitou pacotes. Em vez disso, o balanceador de carga, lembrando-se da conexão, altera o pacote para que o IP de origem seja aquele do servidor virtual, resolvendo esse problema.

A decisão do balanceamento de carga

Normalmente, é nesse ponto que surgem duas questões: como o balanceador de carga decide para qual host encaminhar a conexão, e o que acontece se o host selecionado não estiver funcionando?

Vamos discutir a segunda questão primeiro. O que acontece se o host selecionado não estiver funcionando? A resposta mais simples é que ele não responderá ao pedido do cliente e o tempo da conexão se esgotará, fazendo que ela falhe. Obviamente, essa não é a circunstância ideal, pois não garante uma alta disponibilidade. É por isso que a maioria das tecnologias de balanceamento de carga incluem algum tipo de "Monitoramento de estado" que determina se um host está ou não disponível antes de tentar estabelecer conexões com ele. Há vários níveis de monitoramento de estado, cada um deles com granularidade e foco crescentes. Um monitor básico irá apenas executar um PING e aguardar a resposta do host. Se o host não responder ao PING, é razoável assumir que os serviços definidos no host provavelmente não estão ativos e deveriam ser removidos do cluster de serviços



disponíveis. Infelizmente, mesmo que o host responda ao PING, isso não significa necessariamente que o serviço está funcionando. Por isso, a maioria dos dispositivos têm a capacidade de executar algum tipo de "PINGS de serviço", desde conexões TCP simples até a interação com o aplicativo por meio de scripts. Esses monitores de estado de alto nível não só oferecem mais confiança na disponibilidade do serviços (e não do host) como também permitem que o balanceador de carga distinga entre múltiplos serviços em um mesmo host. O balanceador de carga entende que, embora um serviço possa estar indisponível, outros serviços no mesmo host podem estar funcionando perfeitamente e devem ser considerados como destinos válidos para o tráfego de usuários.

Isso nos leva de volta à primeira pergunta: como o balanceador de carga decide para qual host encaminhar o pedido de conexão? Cada servidor virtual tem um cluster dedicado a serviços (listando os hosts que oferecem aquele serviço) formando uma lista de possibilidades. Além disso, o monitoramento de estado discutido antes modifica essa lista para marcar os hosts atualmente disponíveis que fornecem o serviço indicado. Essa lista modificada será usada pelo balanceador de carga para escolher o host que receberá a nova conexão. Decidir qual será o host exato depende do algoritmo de balanceamento de carga associado àquele cluster em particular. O mais comum é o round-robin simples, em que o balanceador de carga percorre a lista de cima para baixo, alocando cada nova conexão para o próximo host; chegando ao fim da lista, ele retorna ao início. Embora isso seja simples e previsível, ele assume que todas as conexões terão uma carga e duração similares no host de segundo plano, o que nem sempre é verdade. Os algoritmos mais avançados usam coisas como contagem de conexões atuais, utilização do host e até tempos de resposta reais do tráfego existente para o host, a fim de escolher o host mais apropriado no cluster de serviços disponíveis.

Os sistemas de balanceamento de carga suficientemente avançados serão capazes de sintetizar as informações do monitor de estado com os algoritmos de balanceamento para incluir uma compreensão da dependência de serviços. Esse é o caso quando um único host tem vários serviços, todos necessários para completar o pedido do usuário. Um exemplo comum seria em situações de comércio eletrônico, em que um único host fornecerá serviços HTTP padrão (porta 80) bem como HTTPS (SSL/TLS na porta 443). Em muitas dessas circunstâncias, você não quer que o usuário vá para um host que tenha um serviço operacional, e o outro não. Em outras palavras, se o serviço HTTPS falhar em um host, você também vai querer que o serviço HTTP daquele Host seja excluído da lista de serviços disponíveis naquele cluster. Essa funcionalidade é cada vez mais importante, visto que serviços baseados em HTTP se tornam mais diferenciados com o uso de XML e scripting.

Balancear ou não balancear a carga?

O balanceamento de carga na escolha de serviços disponíveis quando um cliente inicia um pedido de transação é apenas metade da solução. Depois que a conexão é estabelecida, o balanceador de carga deve manter um registro para decidir se o tráfego daquele usuário deve ser balanceado ou não. De maneira geral, há dois problemas específicos com o gerenciamento do tráfego que teve a carga balanceada: manutenção da conexão e persistência.

Se o usuário estiver tentando utilizar uma conexão TCP de longa vida (telnet, FTP e outras) que não se encerram imediatamente, o balanceador de carga deve se certificar de que os vários pacotes de dados trafegando por aquela conexão não sejam encaminhados para outros hosts de serviços disponíveis. Esse processo é chamado de manutenção da conexão e exige duas capacidades principais: 1) manter um registro das conexões abertas e do host de serviços ao qual elas pertencem; e 2) continuar a monitorar aquela conexão para que a tabela de conexões possa ser atualizada quando a conexão for encerrada. Isso é um procedimento padrão para a maioria dos balanceadores de carga.



Entretanto, é cada vez mais comum o uso de conexões TCP múltiplas de vida curta (por exemplo, o HTTP) pelo cliente para executar uma tarefa única. Em alguns casos, como a navegação comum na web, isso não faz diferença e cada novo pedido pode ser encaminhado a qualquer um dos hosts de serviços de segundo plano; entretanto, há muitas outras instâncias (como XML, carrinhos de compra, HTTPS e afins) em que é muito importante que várias conexões do mesmo usuário sejam encaminhadas ao mesmo host de serviços de segundo plano, em vez de terem a carga balanceada. Esse conceito é chamado de persistência ou afinidade com servidor. Há várias maneiras de cuidar desse problema, dependendo do protocolo e dos resultados desejados. Por exemplo, nas transações HTTP modernas, o servidor pode especificar uma conexão "keep alive" para agrupar as múltiplas conexões de vida curta em uma única conexão de vida longa, que pode ser gerenciada como qualquer outra conexão de vida longa. Infelizmente, isso não é o suficiente para resolver o problema. O mais grave, com o aumento do uso dos serviços web, é manter todas essas conexões abertas por mais tempo do que o necessário, sobrecarregando os recursos de todo o sistema. Nesses casos, a maioria dos balanceadores de carga oferece outros mecanismos para criar uma afinidade artificial com o servidor.

Uma das formas mais básicas de persistência é a afinidade de endereço de origem. Isso envolve o simples registro do IP de origem dos pedidos recebidos e do Host de serviços a que foram encaminhados, fazendo que todas as transações futuras sejam enviadas ao mesmo Host. Essa também é uma forma simples de lidar com dependências de aplicativos, visto que pode ser aplicada em todos os servidores virtuais e serviços. Na prática, entretanto, o uso disseminado de servidores proxy na Internet, bem como em redes corporativas internas, inutilizou esse tipo de persistência; embora ela funcione, os servidores proxy escondem muitos usuários por trás de um único endereço IP, o que faz que o tráfego desses usuários não possa ser balanceado após o pedido do primeiro usuário, basicamente anulando a capacidade de balanceamento de carga. Hoje, a inteligência dos dispositivos de balanceamento de carga permite que você abra os pacotes de dados e crie tabelas persistentes com virtualmente qualquer coisa dentro deles. Dessa forma, você pode usar informações mais exclusivas e identificáveis como o nome de usuário para manter a persistência; entretanto, é necessário ser cuidadoso para garantir que essas informações identificáveis do cliente serão exibidas em todos os pedidos feitos, visto que todos os pacotes sem essa informação não serão considerados persistentes e passarão pelo processo de balanceamento de carga, o que provavelmente causaria erros no aplicativo.

O balanceamento de carga hoje

Esse documento descreveu os conceitos básicos da tecnologia de balanceamento de carga. É importante compreender que essa tecnologia, embora ainda em uso, é considerada hoje como uma função dos ADCs. Os ADCs evoluíram dos primeiros balanceadores de carga e completaram o processo de virtualização de serviços, permitindo não apenas uma melhoria na disponibilidade, mas também no desempenho em segurança dos serviços de aplicativos sendo solicitados. Hoje, a maior parte das organizações compreende que a mera capacidade de alcançar o aplicativo não o torna utilizável, e que aplicativos não-utilizáveis significam tempo e dinheiro desperdiçados para a companhia que os implementou. Os ADCs permitem a consolidação dos serviços baseados na rede, como SSL/TLS, caching, compressão, rate shaping, detecção de intrusos, firewalls e até o acesso remoto a um único ponto que pode ser compartilhado e reutilizado por todos os serviços de aplicativos e todos os hosts, criando uma rede virtualizada de distribuição de aplicativos. Ao mesmo tempo, sem os fundamentos básicos do balanceamento de carga descritos neste documento, nenhuma das funcionalidades aprimoradas dos ADCs seria possível.