



# Balanceamento de carga: A evolução para os Application Delivery Controller

**Introdução** Um dos efeitos infelizes da evolução contínua do Balanceador de carga nos Application Delivery Controller (ADC) de hoje é a facilidade em esquecer a razão principal pela qual os Balanceadores de carga foram criados - produzir serviços de aplicativos escalonáveis, previsíveis e com alta disponibilidade. Ficamos perdidos no reino do roteamento inteligente de aplicativos, dos serviços virtualizados de aplicativos e implementações de infra-estrutura compartilhada e acabamos nos esquecendo de que nenhuma destas coisas é possível sem uma base firme na estrutura do balanceamento de carga. O quanto, então, o balanceamento de carga é realmente importante, e como os seus efeitos simplificam a entrega de aplicativos?

## **A necessidade do balanceamento de carga**

O objetivo do balanceamento de carga é criar um sistema que virtualize o trabalho dos servidores físicos que executam aqueles serviços. Uma definição mais básica é a de equilibrar a carga entre vários servidores físicos, fazendo que eles pareçam ser um grande servidor para o mundo externo. Há muitos motivos para fazer isso, mas os principais podem ser resumidos em escalonabilidade, alta disponibilidade e previsibilidade.

A escalonabilidade é a capacidade de adaptação fácil e dinâmica ao aumento da carga, sem impacto sobre o desempenho atual. A virtualização de serviços oferece uma oportunidade interessante para a escalonabilidade; se o serviço no ponto de contato do usuário estivesse separado do servidor, o reescalonamento do aplicativo significaria apenas adicionar mais servidores, que não seriam visíveis ao usuário final. A alta disponibilidade (HA) é a capacidade de um site manter-se disponível e acessível, mesmo em caso de falha de um ou mais sistemas. A virtualização dos serviços também oferece uma oportunidade para a alta disponibilidade; se o ponto de contato do usuário for separado dos servidores, a falha de um servidor individual não deixará todo o aplicativo indisponível.

A previsibilidade é um pouco menos clara, pois representa partes da alta disponibilidade e também algumas lições aprendidas ao longo do caminho. Entretanto, a previsibilidade pode ser descrita como capacidade de ter confiança e controle na maneira como o serviços estão sendo distribuídos, com relação à disponibilidade, ao desempenho e assim por diante.

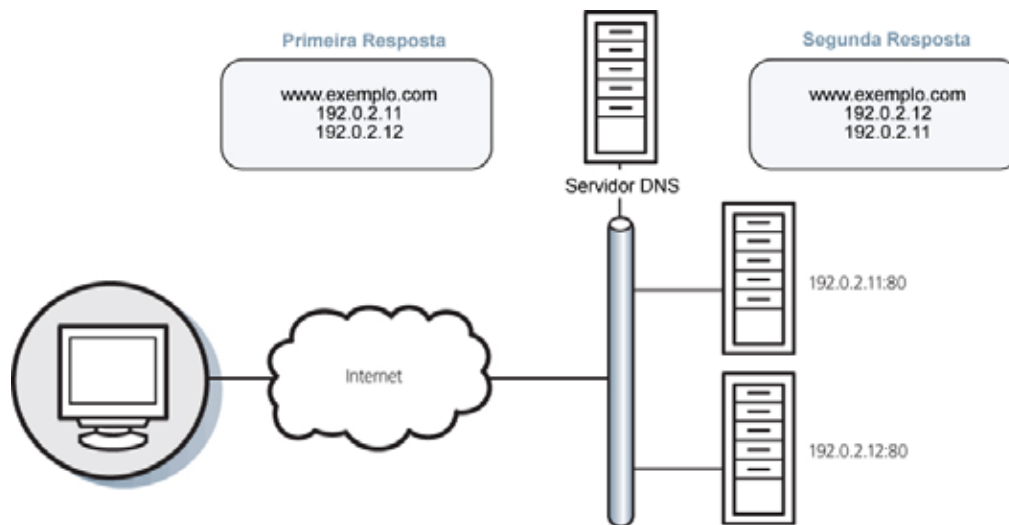
## **Balanceamento de carga: Uma perspectiva**

### **Histórica**

Nos primeiros dias da Internet comercial, muitos candidatos a milionários ponto.com descobriram um problema sério nos seus planos. Os mainframes não tinham programas de servidores web (Não até o AS/400e, pelo menos) e, mesmo que tivessem, com seus orçamentos reduzidos, eles não poderiam arcar com os custos. O máximo a que eles tinham acesso era o hardware de servidor padrão, de algum dos fabricantes de PCs comuns. Qual era o problema, segundo eles? Um servidor baseado em um único PC jamais conseguiria gerenciar o tráfego que seria gerado pelas idéias deles e, se ele fosse desativado, eles estariam desconectados e falidos. Felizmente, algumas dessas pessoas tinham planos para fazer muito dinheiro resolvendo exatamente este problema; foi aí que nasceu o mercado do balanceamento de carga.

### No início, era o DNS

Antes que existisse qualquer dispositivo de balanceamento de carga comercialmente disponível, foram feitas muitas tentativas de usar tecnologias existentes para atingir as metas de disponibilidade e escalabilidade. A mais comum, e ainda utilizada, é o DNS round-robin. O Sistema de Nomes de Domínio (DNS) é o serviço que traduz nomes compreensíveis ([www.exemplo.com](http://www.exemplo.com)) em endereços IP utilizados pelas máquinas. O DNS também fornece um modo pelo qual cada pedido de resolução de nome pode ser respondido com múltiplos endereços IP em ordem diferente.



Na primeira vez que um usuário solicita a resolução do endereço [www.exemplo.com](http://www.exemplo.com), o servidor DNS responde com vários endereços (um para cada servidor que hospeda o aplicativo) em ordem, por exemplo, 1,2 e 3. Na próxima solicitação, o servidor DNS informará os mesmos endereços, mas dessa vez como 2,3 e 1. Essa solução é simples e oferece as características básicas procuradas pelos clientes, distribuindo os usuários de maneira seqüencial entre várias máquinas físicas, usando o nome como ponto de virtualização.

Do ponto de vista da escalabilidade, essa solução funcionou muito bem; provavelmente por isso, variações desse método ainda são usadas hoje em dia, particularmente no balanceamento de carga global ou na distribuição da carga para diferentes pontos de serviços no mundo. Conforme o serviço precisava se expandir, tudo o que o proprietário do negócio tinha de fazer era adicionar mais um servidor, incluir seu endereço IP no registro DNS e pronto, capacidade aumentada. No entanto, há um detalhe. As respostas DNS têm um tamanho máximo permitido, portanto, existe a possibilidade de saturar ou escalonar além desta solução.

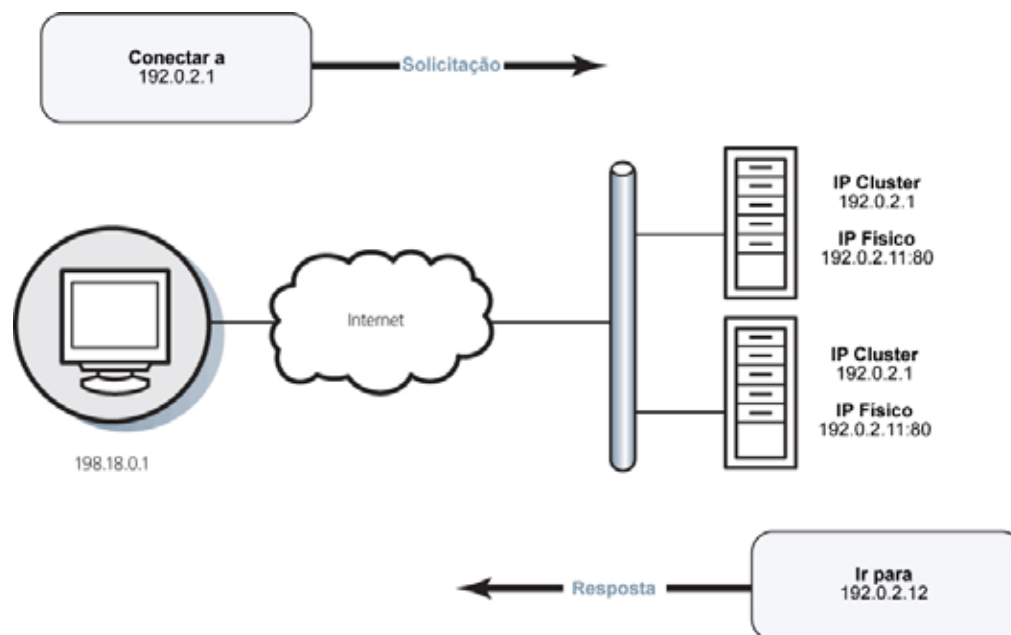
Essa solução não melhorou muito a disponibilidade. Primeiro, porque o DNS não tem a capacidade de saber se os servidores listados estão funcionando ou não, portanto, se um servidor se tornasse indisponível e um usuário tentasse acessá-lo antes que os administradores soubessem do problema e o removessem da lista, o usuário poderia obter um endereço IP para um servidor não funcional. Além disso, os clientes costumam armazenar (no cache) as resoluções de nome. Isso significa que eles nem sempre perguntam por um novo endereço IP e simplesmente voltam para o servidor que usaram antes - esteja ou não funcionando, e sem considerar a intenção de virtualizar e distribuir a carga.

Essa solução também destacou várias necessidades adicionais na área do balanceamento de carga. Conforme mencionado acima, ficou claro que qualquer dispositivo de balanceamento de carga precisaria ter a capacidade de detectar automaticamente as falhas nos servidores físicos, removendo-o dinamicamente da lista de servidores possíveis oferecida aos clientes. Da mesma forma, um bom mecanismo deveria ser capaz de garantir que um cliente não pudesse contornar o balanceamento de carga devido ao caching ou outros meios, a menos que houvesse uma boa razão para isso. Mais importante, os problemas com os servidores DNS intermediários (que não só armazenavam as entradas DNS originais, mas também reordenavam a lista de IPs antes de distribuí-la aos clientes) ressaltaram uma diferença marcante entre distribuição de carga e balanceamento de carga. O round-robin DNS oferecia distribuição descontrolada e um balanceamento ruim. Por último, uma nova necessidade tornou-se aparente - previsibilidade.

A previsibilidade é a capacidade de ter um alto nível de confiança no conhecimento ou previsão do direcionamento do usuário ao servidor. Embora esteja relacionada ao balanceamento de carga em comparação à distribuição não controlada, ela é centrada na idéia de persistência. A persistência é o conceito de garantir que um usuário não seja redirecionado pelo balanceamento de carga para um novo servidor após o início de uma sessão, ou quando o usuário retoma uma sessão previamente suspensa. Essa é uma questão muito importante, que o round-robin DNS não pode resolver.

#### Balanceamento proprietário de carga em software

Uma das primeiras soluções específicas para o problema do balanceamento de carga foi o desenvolvimento de funções de balanceamento de carga integradas diretamente no aplicativo ou no sistema operacional (OS) do servidor de aplicativos. Embora cada companhia desenvolvesse uma implementação diferente, a maioria das soluções era baseada em truques de rede. Por exemplo, uma destas soluções tinha todos servidores listados em um cluster atendendo em um "IP cluster", além do seu próprio endereço IP fixo.



Quando um usuário tentava conectar o serviço, eles conectavam ao IP cluster em vez do IP físico do servidor. Qualquer um dos servidores do cluster que respondesse



primeiro ao pedido iria redirecioná-lo para um endereço IP físico (dele próprio ou de outro sistema no cluster) e a sessão de serviço seria iniciada. Um dos principais benefícios dessa solução é que os desenvolvedores de aplicativos podiam usar várias informações para determinar a qual endereço IP físico o cliente deveria se conectar. Por exemplo, eles podiam fazer que cada servidor no cluster mantivesse uma contagem de quantas sessões cada membro do cluster já estava atendendo, direcionando novos pedidos para o servidor menos utilizado.

Inicialmente, a escalonabilidade dessa solução era óbvia. Tudo o que você tinha de fazer era montar um novo servidor e adicioná-lo ao cluster, aumentando a capacidade do seu aplicativo. Porém, com o tempo, a escalonabilidade do balanceamento de carga baseado em aplicativos começou a gerar dúvidas. Como os membros do cluster precisavam manter contato constante uns com os outros para decidir quem aceitaria a próxima conexão, o tráfego de rede entre os membros do cluster aumentou exponencialmente a cada novo servidor adicionado ao cluster. Depois que o cluster crescia até um certo tamanho (normalmente, de cinco a dez hosts), esse tráfego começou a criar impacto nos usuários finais, bem como na utilização do processador dos próprios servidores. Portanto, a escalonabilidade era ótima, desde que você não precisasse exceder um número pequeno de servidores (menos do que o round-robin DNS poderia suportar).

A alta disponibilidade aumentou dramaticamente com estas soluções. Como os membros do cluster estavam em comunicação constante uns com os outros, e os desenvolvedores de aplicativos podiam usar seu conhecimento para determinar se um servidor estava sendo executado corretamente, isso praticamente eliminou a possibilidade de que um usuário alcançasse um servidor que não pudesse atender ao seu pedido. Devemos notar, entretanto, que cada ciclo de características inteligentes que habilitavam a alta disponibilidade tinham um impacto correspondente na utilização da rede e dos servidores, limitando ainda mais a escalonabilidade. O outro impacto negativo que a alta disponibilidade causou foi sobre a confiabilidade. Muitos dos truques de rede usados para distribuir tráfego nesses sistemas eram complexos e exigiam um monitoramento considerável no nível da rede; conseqüentemente, eles sempre encontravam problemas que afetavam todo o aplicativo e todo o tráfego na rede do aplicativo.

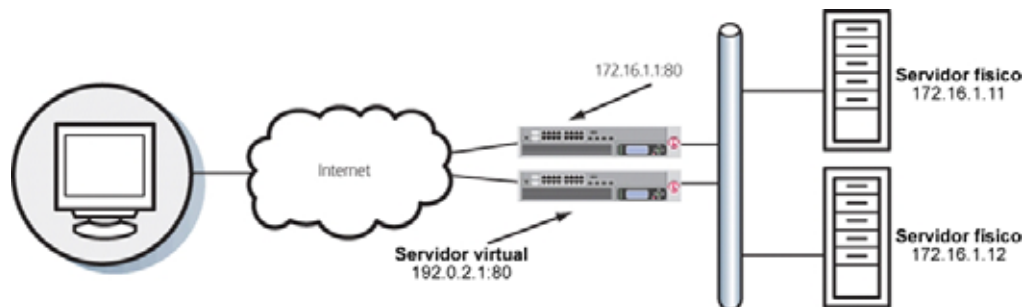
A previsibilidade também foi melhorada por essas soluções. Como os desenvolvedores de aplicativos sabiam quando e como os usuários deveriam ser direcionados ao mesmo servidor em vez de redirecionados pelo balanceamento de carga, eles puderam criar uma lógica para ajudar a garantir a persistência dos usuários por tanto tempo quanto necessário. Eles também usaram a mesma tecnologia de clustering para replicar informações do estado de usuários entre servidores, eliminando várias instâncias que exigiam persistência. Por último, graças ao seu profundo conhecimento de aplicativos, eles puderam desenvolver algoritmos de balanceamento de carga baseados no verdadeiro estado do aplicativo e não em elementos como a conexão, que não eram bons indicadores da carga do servidor.

Além dos limites potenciais na verdadeira escalonabilidade e dos problemas de confiabilidade, o balanceamento proprietário de carga também tinha uma desvantagem adicional - ele dependia do fornecedor do aplicativo para o desenvolvimento e manutenção. O problema principal é que nem todos aplicativos ofereciam balanceamento de carga ou clustering, e os que ofereciam não funcionavam com aqueles implementados por outros fornecedores de aplicativos. Embora existissem várias organizações que produziam programas de balanceamento de carga neutros em relação a fornecedores e baseados no sistema operacional, eles sofriam dos mesmos problemas de escalonabilidade. Sem uma

integração entre os aplicativos, estas soluções também experimentaram desafios adicionais de disponibilidade.

### Hardware de balanceamento de carga baseada em rede

A segunda geração do balanceamento de carga de finalidade específica surgiu na forma de dispositivos baseados em rede. Estes são os verdadeiros avós dos controladores Application Delivery atuais. 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, eles encaminhavam a conexão ao servidor verdadeiro mais apropriado, fazendo uma tradução de endereços de rede (NAT) bidirecional.



O Balanceador de carga podia controlar exatamente qual servidor receberia qual conexão e usava monitores de estado de complexidade crescente para garantir que o servidor de aplicativos (um servidor real, físico) estava respondendo conforme necessário; do contrário, ele pararia de enviar tráfego para aquele servidor até que ele produzisse a resposta desejada (indicando que o servidor estava funcionando corretamente). Embora os monitores de estado não fossem tão abrangentes quanto aqueles criados pelos próprios desenvolvedores de aplicativos, a abordagem do hardware baseada em rede podia oferecer ao menos os serviços básicos de balanceamento de carga para praticamente qualquer aplicativo, de uma maneira uniforme e consistente - criando finalmente um ponto de entrada verdadeiramente virtualizado e único para os servidores distribuindo o aplicativo.

Com essa solução, a escalabilidade só era limitada pela capacidade do equipamento de balanceamento de carga e das redes ligadas a ele. O monitoramento de estado, embora ainda tendo um impacto potencial na rede, não mais cresceu exponencialmente e, porque o balanceador de carga precisava somente manter informações sobre o estado de todo o cluster, e não de cada servidor. Isso reduziu a sobrecarga da rede e dos servidores, liberando capacidade adicional. Não era incomum uma companhia substituir o balanceamento de carga via software por uma solução baseada em hardware e notar imediatamente uma grande queda na utilização dos servidores, tornando desnecessária a compra de servidores adicionais a curto prazo e gerando um maior retorno do investimento a longo prazo.

A alta disponibilidade também foi dramaticamente reforçada por uma solução baseada em hardware. É claro que isso exigia que esses sistemas fossem implementados aos pares para oferecer tolerância às suas próprias falhas, mas a simples redução da complexidade da solução, bem como um balanceamento de carga imparcial em relação aos aplicativos ofereceu uma maior confiabilidade e profundidade como solução. O hardware de balanceamento de carga baseado em rede permitiu que as companhias alcançassem altos níveis de disponibilidade em



todos os aplicativos, ao invés dos poucos que possuíam balanceamento de carga integrado.

A previsibilidade era o principal componente adicionado pelo hardware de balanceamento de carga baseado na rede. Como as decisões de balanceamento de carga eram todas deterministas (consistindo em medidas reais da carga da conexão, tempo de resposta e assim por diante), em contraste com a abordagem sintética da maioria das soluções baseadas em aplicativos, era muito mais fácil prever para onde uma nova conexão seria direcionada e muito simples de manipulá-la. Esses dispositivos também eram capazes de oferecer estatísticas reais de uso, fornecendo percepções de capacidade para a equipe de planejamento e ajudando a documentar os resultados das operações de balanceamento de carga. Um ponto interessante é que esta solução reintroduziu o potencial positivo da distribuição de carga em comparação com o balanceamento de carga. O balanceamento de carga é a meta ideal se todos os seus servidores são idênticos; entretanto, conforme um site cresce e amadurece, esse não é sempre o caso. A inteligência agregada para criar a distribuição controlada de carga (em oposição a distribuição não controlada do DNS dinâmico) permitiu que os proprietários dos negócios finalmente usassem a distribuição de carga de uma maneira positiva, enviando mais conexões para os maiores servidores e menos para os menores.

O advento dos Balanceadores de carga baseados em rede inaugurou uma nova era na arquitetura de aplicativos. As discussões sobre a alta disponibilidade, que antes giravam em torno do tempo de operação, passaram rapidamente a se concentrar no significado de "Disponível" (se um usuário tem de esperar 30 segundos por uma resposta, ela está disponível? E se for um minuto?). Eles também trouxeram novos benefícios para a segurança e gerenciamento, como mascarar a verdadeira identidade dos servidores de aplicativos, protegendo-a da Internet, e a capacidade de redirecionar conexões de um servidor para que ele pudesse ser desativado para manutenção, sem impacto sobre os usuários. Esta é a base da qual se originaram os Application Delivery Controller (ADCs).

### **Controladores de Application Delivery**

Em uma definição simples, os ADCs são a evolução dos bons balanceadores de carga. Embora a maioria das conversas sobre ADC raramente mencionem o balanceamento de carga, sem as capacidades do hardware balanceador de carga baseado em rede, eles não poderiam afetar a distribuição de aplicativos. Hoje, nós falamos sobre segurança, disponibilidade e desempenho, mas a tecnologia subjacente do balanceamento de carga é crítica para a execução de todos eles.

Ao discutir a segurança dos ADCs, a virtualização criada pela tecnologia básica de balanceamento de carga é absolutamente crítica. Podemos discutir o alívio da carga de processamento SSL/TLS, autenticação centralizada ou até firewalls com estado, mas o poder dessas soluções reside no fato de que um dispositivo balanceador de carga é o ponto agregado de virtualização de todos os aplicativos. A autenticação centralizada é um exemplo clássico. Os mecanismos tradicionais de autenticação sempre foram criados diretamente no aplicativo. Assim como no balanceamento de carga baseada em aplicativos, cada implementação era única para um aplicativo e dependia da implementação dele, resultando em vários métodos diferentes. Ao aplicar a autenticação em um ponto de entrada virtualizado para todos os aplicativos, um método uniforme e unificado de autenticação pode ser empregado. Isso não apenas simplifica drasticamente o planejamento e gerenciamento do sistema de autenticação como também aumenta o desempenho dos servidores de aplicativos, eliminando a necessidade de executar aquela função. Além disso, ela também



elimina a necessidade, especialmente para aplicativos internos, de gastar tempo e dinheiro desenvolvendo processos de autenticação em cada aplicativo separadamente.

A disponibilidade é o atributo dos ADCs mais fácil de associar ao balanceador de carga original, pois está relacionada a todos os seus atributos básicos: escalonabilidade, alta disponibilidade e previsibilidade. Entretanto, os ADCs ampliam ainda mais esse conceito. Para eles, a disponibilidade também representa conceitos avançados como a dependência de aplicativos e o provisionamento dinâmico. Os ADCs compreendem que, no mundo de hoje, os aplicativos raramente trabalham de maneira isolada; eles normalmente dependem de outros aplicativos para cumprir suas tarefas. Esse conhecimento aumenta a capacidade dos ADCs de fornecer disponibilidade de aplicativos, considerando também estes outros processos. Os ADCs mais inteligentes do mercado também oferecem interfaces programáticas que permitem a mudança dinâmica do modo como eles fornecem serviços, com base em informações externas. Essas interfaces possibilitam coisas como o provisionamento dinâmico e a adição ou subtração de servidores disponíveis, com base na utilização e necessidade.

A melhoria do desempenho foi outra consequência óbvia do conceito de balanceamento de carga. Os Balanceadores de carga melhoravam o desempenho dos aplicativos garantindo que as conexões fossem distribuídas aos serviços disponíveis (e respondessem em um intervalo aceitável) com o menor número de conexões e/ou utilização do processador. Isso garantia que cada conexão estava sendo atendida pelo sistema mais capacitado para atendê-la. Mais tarde, conforme a transferência de carga SSL/TLS (usando hardware dedicado) se tornou uma função corriqueira nos produtos de balanceamento de carga, ela reduziu a sobrecarga computacional do tráfego criptografado e a carga dos servidores de segundo plano, melhorando também seu desempenho.

Os ADCs de hoje, entretanto, vão ainda mais longe. Esses dispositivos normalmente oferecem caching, compressão e até tecnologia de rate shaping para aumentar o desempenho geral na distribuição de aplicativos. Além disso, ao invés de usar a implementação estática de dispositivos independentes, fornecendo esses serviços, um ADC pode usar sua inteligência de aplicativo para empregar esses serviços somente quando eles vão gerar um benefício de desempenho, otimizando seu uso. Por exemplo, a tecnologia de compressão, ao contrário da crença popular, não é necessariamente benéfica para todos os usuários dos aplicativos. É claro que usuários com pouca banda (como conexões discadas ou dados de pacotes móveis) podem se beneficiar muito de pacotes menores, já que o gargalo é a capacidade real. Mesmo conexões que devem viajar longas distâncias podem se beneficiar, pois pacotes menores significam tempos menores de ida e volta para transportar os dados, diminuindo o impacto da latência da rede. Entretanto, conexões de curta distância (digamos, no mesmo continente) com banda larga na verdade apresentam um desempenho pior com compressão; como a capacidade não é necessariamente o gargalo, a carga adicional da compactação e descompactação aumenta a latência de uma forma que não é compensada pela capacidade adicional, do ponto de vista do desempenho. Em outras palavras, se não for corretamente gerenciada, a tecnologia de compressão pode ser uma solução pior do que o problema original. Porém, aplicando a compressão de maneira inteligente, somente quando ela beneficia o desempenho geral, um ADC otimiza o uso e o custo da tecnologia de compressão, deixando mais ciclos de processador para funções que farão melhor uso deles.



### *O futuro dos ADCs*

Os ADCs são a evolução natural do patrimônio crítico de rede, mantido pelos balanceadores de carga no passado. Embora devam muito a esses dispositivos antigos, eles são uma nova espécie, oferecendo não só disponibilidade, mas também desempenho e segurança. Conforme seu nome sugere, eles cuidam de todos os aspectos da distribuição de aplicativos, da melhor maneira possível.

Assim como os Balanceadores de carga evoluíram para se tornar ADCs, as mudanças do mundo técnico, em constante evolução, vão continuar a moldar os ADCs em algo ainda mais capaz de se adaptar aos requerimentos de disponibilidade, desempenho e segurança na distribuição de aplicativos. As idéias de integração do controle de acesso de rede (NAC genérico), de compressão e caching de aplicativos e até a crescente importância da aplicação de regras comerciais para o gerenciamento e controle da distribuição de aplicativos vão continuar a estender os limites dos benefícios que estes dispositivos podem oferecer às organizações. A pressão crescente para minimizar o número de dispositivos na rede entre usuário e os aplicativos vão continuar a demolir as tecnologias independentes tradicionais (como firewalls, antivírus e IPS), integrando-as ao território do ADC. Conforme novas tecnologias e protocolos são desenvolvidos em uma tentativa de atender a crescente demanda por acesso de qualquer lugar em qualquer dispositivo aos aplicativos de dados, os ADCs de amanhã fornecerão a inteligência para determinar como essas e outras tecnologias serão integradas nas redes existentes, bem como onde e quando elas serão mais eficientes.

Embora ainda não esteja exatamente claro quantas tecnologias serão diretamente substituídas pelos componentes do ADC, é certo que os ADCs vão evoluir e se tornar o ponto primário de integração pelo qual essas tecnologias vão interagir com os aplicativos distribuídos e seus usuários.