

1.5.4 Escalabilidade

Os sistemas distribuídos funcionam de forma efetiva e eficaz em muitas escalas diferentes, variando desde uma pequena intranet até a Internet. Um sistema é descrito como escalável se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários. O número de computadores e de serviços na Internet aumentou substancialmente. A Figura 1.6 mostra o aumento no número de computadores e servidores Web durante os 12 anos de história da Web até 2005. É interessante notar o aumento significativo tanto no número de computadores como no serviços Web nesse período, mas também a porcentagem relativa, que cresce rapidamente, uma tendência explicada pelo crescimento da computação pessoal fixa e móvel. Um único servidor Web também pode, cada vez mais, ser hospedado em vários computadores.

O projeto de sistemas distribuídos escaláveis apresenta os seguintes desafios:

Controlar o custo dos recursos físicos: à medida que a demanda por um recurso aumenta, deve ser possível, a um custo razoável, ampliar o sistema para atendê-la. Por exemplo, a frequência com que os arquivos são acessados em uma intranet provavelmente vai crescer à medida que o número de usuários e de computadores aumentar. Deve ser possível adicionar servidores de arquivos de forma a evitar o gargalo de desempenho que haveria caso um único servidor de arquivos tivesse de tratar todos os pedidos de acesso a arquivos. Em geral, para que um sistema com n usuários seja escalável, a quantidade de recursos físicos exigida para suportá-los deve ser, no máximo, $O(n)$ – isto é, proporcional a n . Por exemplo, se um único servidor de arquivos pode suportar 20 usuários, então dois servidores deverão suportar 40 usuários. Embora isso pareça um objetivo óbvio, não é necessariamente fácil atingi-lo, como mostraremos no Capítulo 12.

Controlar a perda de desempenho: considere o gerenciamento de um conjunto de dados, cujo tamanho é proporcional ao número de usuários ou recursos presentes no sistema; por exemplo, a tabela de correspondência entre os nomes de domínio dos computadores e seus endereços IP mantidos pelo Domain Name System (DNS), que é usado principalmente para pesquisar nomes, como www.amazon.com. Os algoritmos que utilizam estruturas hierárquicas têm melhor escalabilidade do que os que usam estruturas lineares. Porém, mesmo com as estruturas hierárquicas, um aumento no tamanho resulta em alguma perda de desempenho: o tempo que leva para acessar dados hierarquicamente estruturados é $O(\log n)$, onde n é o tamanho do conjunto de dados. Para que um sistema seja escalável, a perda de desempenho máxima não deve ser maior do que isso.

Impedir que os recursos de software se esgotem: um exemplo de falta de escalabilidade é mostrado pelos números usados como endereços IP (endereços de computador na Internet). No final dos anos 70, decidiu-se usar 32 bits para esse propósito, mas, conforme será explicado no Capítulo 3, a quantidade de endereços IP disponíveis está se esgotando. Por isso, uma nova versão do protocolo, com endereços IP em 128 bits, está sendo adotada e isso exige modificações em muitos componentes de software. Para sermos justos com os primeiros projetistas da Internet, não há uma solução correta para esse problema. É difícil prever, com anos de antecedência, a demanda que será imposta sobre um sistema. Além disso, superestimar o crescimento futuro pode ser pior do que se adaptar para uma mudança quando formos obrigados a isso – por exemplo, endereços IP maiores ocupam espaço extra no armazenamento de mensagens e no computador.

Evitar gargalos de desempenho: em geral, os algoritmos devem ser descentralizados para evitar a existência de gargalos de desempenho. Ilustramos esse ponto com referência ao predecessor do Domain Name System, no qual a tabela de correspondência entre endereços IP e nomes era mantida em um único arquivo central, cujo download podia ser feito em qualquer computador que precisasse dela. Isso funcionava bem quando havia apenas algumas centenas de computadores na Internet, mas logo se tornou um sério gargalo de desempenho e de administração. Com milhares de computadores na Internet, imagine o tamanho e o acesso a esse arquivo central. O Domain Name System eliminou esse gargalo, particionando a tabela de correspondência de nomes entre diversos servidores localizados em toda a Internet e administrados de forma local – veja os Capítulos 3 e 13. Alguns recursos compartilhados são acessados com muita frequência; por exemplo, muitos usuários podem acessar uma mesma página Web, causando uma queda no desempenho. No Capítulo 2, veremos que o uso de cache e de replicação melhora o desempenho de recursos que são pesadamente utilizados.

De preferência, o software de sistema e de aplicativo não deve mudar quando a escala do sistema aumentar, mas isso é difícil de conseguir. O problema da escala é um tema central no desenvolvimento de sistemas distribuídos. As técnicas que têm obtido sucesso serão discutidas extensivamente neste livro. Elas incluem o uso de dados replicados (Capítulo 18), a técnica associada de uso de cache (Capítulos 2 e 12) e a distribuição de vários servidores para manipular as tarefas comumente executadas, permitindo que várias tarefas semelhantes sejam executadas concorrentemente.