

A importância dos sistemas distribuídos nos serviços de internet

Chauã Queirolo

Tanenbaum e Steen [1] definem um sistema distribuído como "uma coleção de elementos de computação autônomos que aparecem para o usuário como um único sistema coerente". Ou seja, existe um ecossistema de múltiplos dispositivos com diferentes características de hardware e software que se comunicam entre si, porém, para o usuário, trata-se de um único sistema. Esses dispositivos, também chamados de nós, operam como um todo, e os usuários não podem ver onde seus processos estão sendo executados.

Desta forma, o sistema funciona de forma transparente para o usuário. O sistema deve suportar vários níveis de transparência. Dentre eles, destacam-se os seguintes aspectos: acesso, localidade, realocação, migração, replicação, simultaneidade e falha. Ao acessar, o usuário não precisa conhecer a representação dos dados transmitidos, sejam eles binários, em modo texto ou dados estruturados. Localmente, o usuário não sabe onde processar. Na realocação, os dados podem mudar de local conforme o usuário os utiliza, enquanto na migração, os dados são movidos de um local para outro.

A replicação de dados ocorre quando os dados são replicados para diferentes locais para garantir um acesso mais rápido (localização geográfica) e/ou para garantir a disponibilidade dos dados mesmo que o repositório falhe. Na concorrência, os dados podem ser acessados ao mesmo tempo de diversas fontes, e finalmente, no caso de falhas, o sistema consegue gerenciar de modo que o usuário não perceba isso.

A busca pela transparência total é uma tarefa difícil de ser alcançada. Isto ocorre porque a medida que controles para garantir a replicação ou a concorrência, por exemplo, ocasiona um aumento da latência da rede. Este aumento da latência expõe para o usuário que algum ponto da rede pode estar com problema. Outro problema é que o tratamento de falhas fica mais difícil. Por exemplo, é difícil verificar se um processo está demorando por causa de erro ou por causa de demora no processamento.

Nos sistemas distribuídos, para que os processos consigam interagir entre si, alguns requisitos são necessários: definição de interfaces de comunicação bem estruturadas, interoperabilidade entre sistemas distintos, portabilidade de aplicações e a facilidade de criação de extensões. Estes requisitos garantem a escalabilidade aos sistemas.

A escalabilidade permite que o sistema otimize o atendimento às requisições de uso conforme a demanda. Por exemplo, uma aplicação web para venda de ingressos tem uma média de 50 acessos por hora em dias normais, no entanto, para um show internacional o número de acessos chega a centenas de acessos por segundo. Um sistema que consegue redimensionar seu tamanho de acordo com a necessidade permite diminuição dos cus-

tos de infraestrutura e manutenção, além de diminuir o consumo de energia elétrica e impactos ambientais.

A escalabilidade pode ocorrer em três níveis: tamanho, geografia e administração. A escalabilidade por tamanho é o gerenciamento do poder computacional conforme a demanda. Os números de nós aumentam ou diminuem de acordo com a necessidade e pode ser obtida através de servidores mais parrudos. A escalabilidade geográfica garante que mesmo que a distância entre o usuário e os nós da rede aumentem, o sistema tenha resposta e desempenho satisfatórios. Finalmente, a escalabilidade administrativa permite que o sistema seja facilmente gerenciado independente do seu tamanho e complexidade.

Uma das maneiras de garantir que diferentes nós trabalhem com diferentes aplicações é usando o recurso de virtualização. Na virtualização uma máquina virtual executa sobre o sistema operacional nativo do nó. As aplicações são projetadas para serem executadas na arquitetura da VM o que isola o hardware da aplicação. Desse modo, uma aplicação pode executar em diferentes hardwares sem precisar de recompilação.

O Java popularizou o conceito de máquina virtual, e tinha como slogan: "write once, run everywhere". Atualmente, a virtualização por contêineres se tornou popular. Um contêiner é uma máquina virtual com uma versão de sistema operacional (normalmente Linux) enxuta que executa uma aplicação específica. Vantagem do uso de contêineres é a aplicação pode ser escrita com qualquer linguagem ou *framework*, quando comparado com o uso das máquinas virtuais Java. O Docker é o sistema utilizado para o gerenciamento de contêineres. Em sistemas grandes e complexos existe a necessidade de orquestração e gerenciamento destes contêineres. O *kubernetes* é a aplicação que realiza esta tarefa.

A comunicação entre aplicações que executam em diferentes nós da rede pode ser realizada através de uma fila de mensageria. O uso das filas garante que um sistema que processe dados muito rapidamente afogue outro que tenha consumo mais lento, ou menor poder de processamento. Assim, todos os processos produtores jogam sua saída na fila e os processos consumidores retiram uma próxima entrada à medida que forem sendo liberados espaços. Alguns exemplos de plataformas de mensagens são Kafka, Rabbitmq, WebSphere MQ, Amazon MQ.

Com relação a persistência de dados, existem diversas questões quanto aos sistemas distribuídos. Por exemplo, como controlar a replicação de dados ou gerenciar um volume grande de escritas ou atualizações em um banco distribuído geograficamente. O Apache Cassandra é um exemplo de banco de dados distribuído que endereça estas questões.

A internet, como um todo, faz uso extensivo de sistemas distribuídos. Deste modo, a criação de aplicações de larga escala requerem soluções elegantes e confiáveis a nível de escalabilidade, processamento e acesso aos dados.