SÃO
PAULO
TECH
SCHOOL

Computação e sistemas distribuídos

Balanceamento de carga

Eduardo Verri

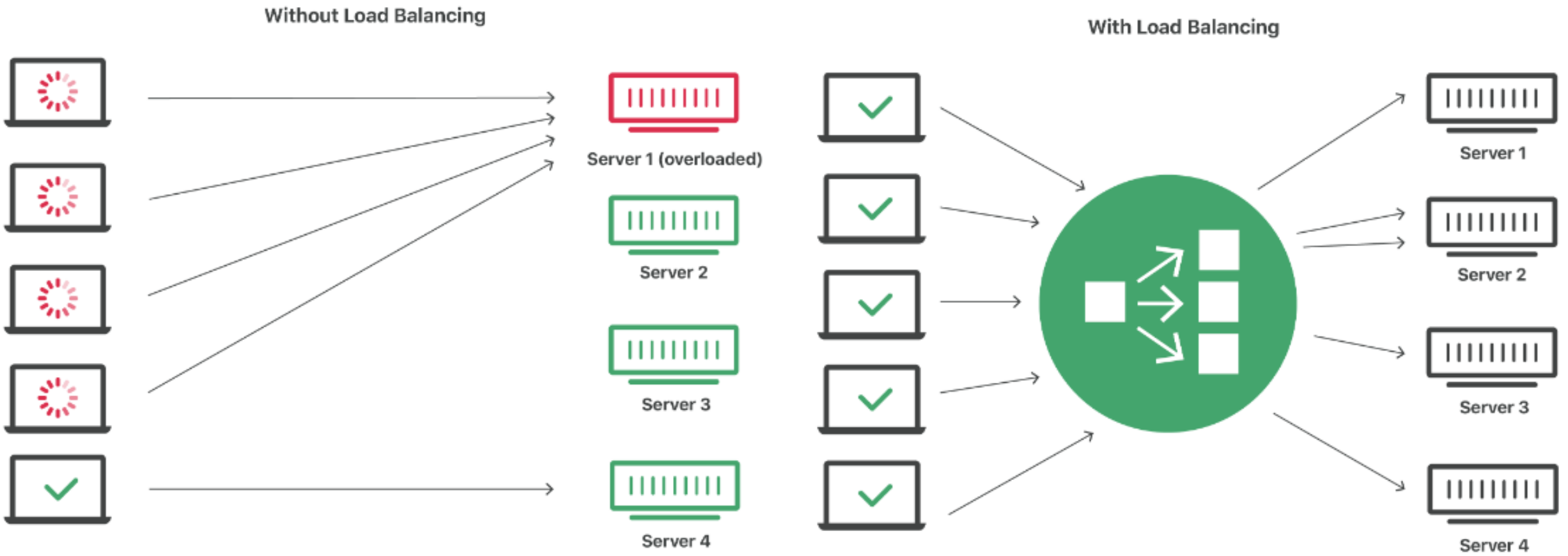
eduardo.verri@sptech.school

Pequena definição

Balanceamento de carga é o processo de distribuição eficiente do tráfego de rede entre vários servidores para otimizar a disponibilidade de aplicativos e garantir uma experiência positiva para o usuário final.

Para lidar com volumes tão altos de tráfego, a maioria das aplicações tem muitos servidores de recursos com dados duplicados entre eles. Um balanceador de carga é um dispositivo que fica entre o usuário e o grupo de servidores e atua como um facilitador invisível, garantindo que todos os servidores de recursos sejam usados igualmente.

Sem load balance / Com load balance



Como funciona o balanceamento?

O balanceamento de carga pode ser implementado de algumas maneiras.

Balanceadores de carga de hardware são dispositivos físicos instalados e mantidos no local.

Balanceadores de carga de software são aplicativos instalados em servidores de propriedade privada ou entregues como um serviço de nuvem gerenciada (balanceamento de carga na nuvem).

Como funciona o balanceamento?

Os balanceadores de carga trabalham mediando as solicitações de clientes recebidos em tempo real e determinando quais servidores de backend podem processar essas solicitações da melhor maneira possível.

Para evitar que um único servidor seja sobrecarregado, o balanceador de carga encaminha as solicitações para vários servidores disponíveis nas instalações ou hospedados em server farms ou data centers em nuvem.

Após o servidor atribuído receber a solicitação, ele responde ao cliente por meio do balanceador de carga. O balanceador de carga então conclui a conexão servidor para cliente, combinando o endereço IP do cliente com o do servidor selecionado.

O cliente e o servidor poderão se comunicar e realizar as tarefas solicitadas até que a sessão seja concluída.

Benefícios do balanceamento

Disponibilidade

São realizadas verificações de funcionamento nos servidores antes de encaminhar as solicitações para eles.

Se um servidor estiver prestes a falhar, ou estiver offline, a carga de trabalho é redirecionada para um servidor em operação para evitar interrupções de serviço e manter alta disponibilidade.

Escalabilidade

Com uma infraestrutura de alto desempenho sob demanda que pode lidar com as cargas de tráfego de rede mais pesadas ou leves.

Servidores físicos ou virtuais podem ser adicionados ou removidos conforme necessário, tornando a escalabilidade simples e automatizada.

Segurança

Podemos incluir recursos de segurança, como criptografia SSL, firewalls de aplicativos web (WAF) e autenticação multifatorial (MFA).

Ao rotear ou descarregar o tráfego de rede com segurança, o balanceamento de carga pode ajudar a proteger contra riscos de segurança, como ataques de distributed denial-of-service (DDoS).

Algoritmos de balanceamento

O método para rotear uma solicitação para um servidor específico é definido por um algoritmo de balanceamento de carga

- **Round-robin:** Usa o DNS (Domain Name System) para atribuir sequencialmente solicitações a cada servidor em uma rotação contínua. É o método mais básico, pois utiliza apenas o nome de cada servidor para determinar qual deles receberá a próxima solicitação recebida.

- **Round-robin ponderado:** Além de seu nome DNS, cada servidor nesse algoritmo também recebe um "peso". O peso determina quais servidores devem ter prioridade sobre outros para lidar com as solicitações recebidas. Um administrador decide como cada servidor será ponderado com base em sua capacidade e nas necessidades da rede.

Algoritmos de balanceamento

- **Hash de IP:** Combina endereços de IP de origem e de destino do tráfego de entrada e usa uma função matemática para convertê-lo em um hash. Com base no hash, a conexão é atribuída a um servidor específico.

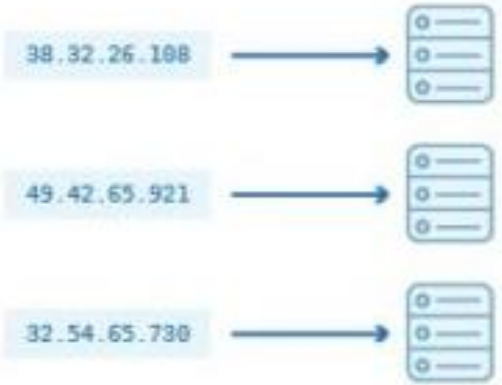
- **Menos conexões (Least connections):** Esse algoritmo dá prioridade ao servidor com as menores conexões ativas quando uma nova solicitação de cliente é recebida. Esse método ajuda a evitar que os servidores fiquem sobrecarregados com conexões e a manter uma carga consistente em todos os servidores o tempo todo.

Algoritmos de balanceamento

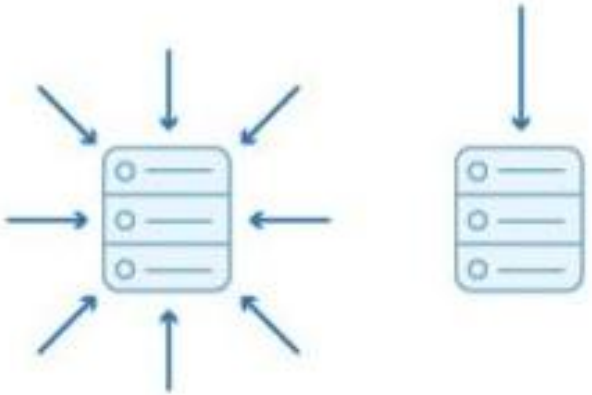
- **Menor tempo de resposta (Lest response time):** Este algoritmo combina o menor método de conexão com o menor tempo médio de resposta do servidor. Tanto o número de conexões quanto o tempo que leva para um servidor realizar solicitações e enviar uma resposta são avaliados. O servidor mais rápido com menos conexões ativas receberá a solicitação recebida.

- **Baseado em recursos:** os balanceadores de carga distribuem o tráfego analisando a carga atual do servidor. Um software especializado chamado agente é executado em cada servidor e calcula o uso de recursos do servidor, como sua capacidade de computação e memória. Em seguida, o balanceador de carga verifica se há recursos livres suficientes no agente antes de distribuir o tráfego para esse servidor.

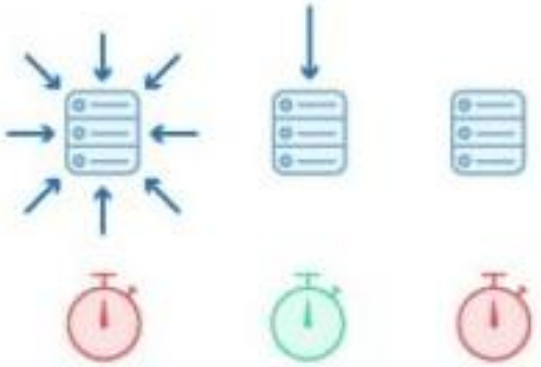
Round Robin



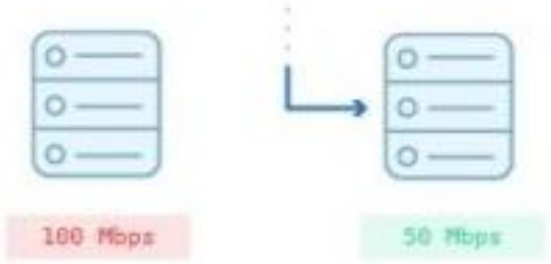
IP Hash



Least Connections



Least Response Time



Least Bandwidth

Tipos de balanceamento de carga

Podemos classificar o balanceamento de carga em três categorias principais, dependendo do que o balanceador de carga verifica na solicitação do cliente para redirecionar o tráfego.

- **Balanceamento de carga de aplicações:** eles examinam o conteúdo da solicitação, como cabeçalhos HTTP ou IDs de sessão SSL, para redirecionar o tráfego.

- **Balanceamento de carga de rede:** eles examinam endereços IP e outras informações de rede para redirecionar o tráfego de maneira ideal. Eles rastreiam a origem do tráfego da aplicação e podem atribuir um endereço IP estático a vários servidores.

- **Balanceamento de carga de DNS:** nele você configura seu domínio para rotear solicitações de rede em um grupo de recursos no seu domínio. Um domínio pode corresponder a um site, um sistema de correio, um servidor de impressão ou outro serviço acessível pela Internet.

E o NGINX?

É possível usar o Nginx como um balanceador de carga HTTP para distribuir o tráfego para vários servidores de aplicativos e melhorar o desempenho, escalabilidade e confiabilidade de aplicações web.

Os seguintes mecanismos (ou métodos) de balanceamento de carga são suportados

round-robin: as solicitações aos servidores de aplicativos são distribuídas em round-robin,

menos conectado: a próxima solicitação é atribuída ao servidor com o menor número de conexões ativas

ip-hash: uma função hash é usada para determinar qual servidor deve ser selecionado para a próxima solicitação (com base no endereço IP do cliente).

Método mais simples

```
http {
        upstream myapp1 {
                server url.server1;
                server url.server2;
                server url.server3;
        }
        server {
                listen 80;
                location / {
                        proxy_pass http://myapp1;
                }
        }
}
```

Por default o Nginx utiliza o método Round-robin de balanceamento de carga.

Least connected

```
http {
        upstream myapp1 {
                least_conn;
                server url.server1;
                server url.server2;
                server url.server3;
        }
        server {
                listen 80;
                location / {
                        proxy_pass http://myapp1;
                }
        }
}
```

Permite controlar a carga nas instâncias do aplicativo de forma mais justa em uma situação em que algumas das solicitações demoram mais para serem concluídas.

O balanceamento de carga menos conectado no nginx é ativado quando a diretiva **`less_conn`** é usada.

```
http {
        upstream myapp1 {
                ip_hash;
                server url.server1;
                server url.server2;
                server url.server3;
        }
        server {
                listen 80;
                location / {
                        proxy_pass http://myapp1;
                }
        }
}
```

Persistência de sessão [ip Hash]

Se houver a necessidade de vincular um cliente a um servidor de aplicação específico (tornar a sessão do cliente "persistente") em termos de sempre tentar selecionar um servidor específico, o **ip-hash** pode ser usado.

Este método garante que as solicitações de um mesmo cliente serão sempre direcionadas para o mesmo servidor, exceto quando este servidor estiver indisponível.

```
http {
        upstream myapp1 {
                server url.server1 weight=3;
                server url.server2;
                server url.server3;
        }
        server {
                listen 80;
                location / {
                        proxy_pass http://myapp1;
                }
        }
}
```

Balanceamento de carga ponderado

Também é possível influenciar ainda mais os algoritmos de balanceamento de carga Nginx usando pesos de servidor.

Com o Round-robin em particular, isso também significa uma distribuição mais ou menos igual de solicitações entre os servidores.

No exemplo a cada 5 novas solicitações: 3 solicitações serão direcionadas para server1, uma solicitação irá para server2 e outra para server3.

Agradeço
a sua atenção!

Eduardo Verri

eduardo.verri@sptech.school

SÃO PAULO TECH SCHOOL