

Relatório do Trabalho Prático 3

Replicação de Serviços Web

Identificação

Mateus Carvalho Gonçalves - 201810245

Descrição Geral da Solução

Um dos requisitos principais de um sistema distribuído é a alta disponibilidade, por isso a replicação de serviços é um tema essencial nesse contexto, para garantir que uma falha possa ser atendida por uma réplica e não gere interrupção do serviço.

Além de melhorar a disponibilidade do recurso, podendo chegar a quase 100% de disponibilidade com altos números de réplicas, o desempenho também pode ser otimizado por meio desta técnica. Um dos fatores que podemos citar é a distância física entre cliente e servidor, por exemplo: se um serviço é replicado em cada região brasileira, a latência da requisição diminui, além de diminuir o número de requisições de um servidor.

Para isso, o balanceamento de carga tem papel fundamental, levando em consideração as especificações de cada máquina. Para implementar o balanceamento de carga do presente relatório, foi utilizado o NGINX, um servidor web focado em alta performance, concorrência e utilização de poucos recursos. Ademais, como foi arquitetado como um proxy reverso é possível utilizá-lo como balanceador de cargas.

Foram utilizados três programas escritos em Javascript disponibilizados pelo professor para simular os servidores web, que apenas retornam strings para serem printadas. Para o cliente, foi utilizado o comando *curl* do terminal combinado com comandos de repetição.

Para registrar os servidores para balanceamento de cargas, é necessário alterar o arquivo de configurações do NGINX (*nginx.conf*), como mostrado na figura abaixo.

```
upstream backend_servers {
    server localhost:3001 weight=3;
    server localhost:3002;
    server localhost:3003;
}

server {
    listen      8080;
    server_name localhost;

    location / {
        proxy_pass http://backend_servers;
    }
}
```

É possível perceber que a imagem mostra que foi dado um peso maior ao servidor 1, localizado na porta 3001 do localhost. É utilizado o algoritmo Round-Robin caso nenhum peso seja especificado, e também é possível configurar o NGINX com outros algoritmos de balanceamento.

Resultados

Sem aplicação de pesos aos servidores, obteve-se o resultado ilustrado na figura abaixo. Foram feitas 15 requisições para para o localhost:8080 (NGINX), que fez o balanceamento de carga de forma que cada servidor respondeu a 5 requisições, ou seja, foi feito um balanceamento igualitário.



```
cura@the-mcp: ~  
cura@the-mcp:~$ for i in {1..15}; do curl localhost:8080; done  
response from server - 3.  
response from server - 1.  
response from server - 2.  
response from server - 3.  
response from server - 1.  
response from server - 2.  
response from server - 3.  
response from server - 1.  
response from server - 2.  
response from server - 3.  
response from server - 1.  
response from server - 2.  
response from server - 3.  
response from server - 1.  
response from server - 2.  
cura@the-mcp:~$
```

Já na próxima figura, é possível perceber que o servidor 1 respondeu a 9 requisições e os servidores 2 e 3 a 3 requisições, isso porque foi atribuído um peso ao servidor 1 (*weight=3* presente na primeira figura do relatório).

```
cura@the-mcp: ~  
cura@the-mcp:~$ for i in {1..15}; do curl localhost:8080; done  
response from server - 1.  
response from server - 2.  
response from server - 1.  
response from server - 3.  
response from server - 1.  
response from server - 1.  
response from server - 2.  
response from server - 1.  
response from server - 3.  
response from server - 1.  
response from server - 1.  
response from server - 2.  
response from server - 1.  
response from server - 3.  
response from server - 1.  
cura@the-mcp:~$ ~
```

Dificuldades

A implementação ajudou a simular, em baixa escala, serviços de replicação e balanceamento de carga. Porém, por se tratar de um ambiente local não foi possível simular situações de falhas e outras que acontecem no cenário real de sistemas distribuídos.