

Tópicos Avançados em Arquiteturas Distribuídas de Software

Relatório Trabalho Final

Grupo 8: Lucas Penteado, Michele Argolo e Sofia Silveira

Primeiros Passos

Antes do grupo começar o desenvolvimento efetivo do trabalho, foi decidido que utilizaríamos a cloud da UFSCar ao invés do próprio computador. Inicialmente o grupo imaginou que seria melhor utilizar o próprio computador, pelo fato de o dia 20/06 ser feriado. Com isso, os integrantes voltariam para suas respectivas casas e seria mais difícil dar continuidade ao trabalho até a próxima segunda-feira (dia 24/06), além de ser necessário estar conectado à rede da UFSCar para ter acesso à Dashboard do Prometheus.

Porém, percebemos que o computador não tem memória suficiente, o que não permitiria a realização do trabalho. Além disso, já é possível fazer a primeira aplicação e instalar os BDs das aplicações 2 e 3 antes de ser necessária a utilização da Dashboard do Prometheus. Assim, é possível dar continuidade ao trabalho mesmo ao longo do feriado. Portanto, a decisão final foi a utilização da cloud da UFSCar.

Voltaremos a utilizar a VM de número 8, já que os problemas encontrados nela ao longo do desenvolvimento do Trabalho 1 já foram resolvidos. Para isso, primeiro acessamos a VM Master da cloud da UFSCar:

```
ssh -i cloud_ufscar_rsa.dms ubuntu@200.136.252.136
```

Em seguida, acessamos a VM:

```
$ bash acessar.sh 8
```

Para o banco de dados, o grupo optou por utilizar o MongoDB, pois é um banco de dados conhecido, com bastante informação disponível na Internet, além de já ter sido estudado por alguns integrantes do grupo na disciplina de Processamento Massivo de Dados. Outro motivo que nos influenciou a tomar tal decisão foi a existência de um driver do MongoDB para Python, chamado PyMongo. Com esse driver é possível acessar o banco de dados Mongo, além de enviar *queries* para a inserção e modificação de campos.

Arquitetura

Para este trabalho, serão criadas duas máquinas virtuais, uma delas será o Master e a outra será o Worker, de forma análoga ao Trabalho 1 da disciplina. O Master terá 5 *containers*: um para o Server da aplicação 1 (transferência de arquivos), dois para cada instância diferente do banco de dados MongoDB (sendo um para app2 e um para app3), um para o Server da aplicação 2 (métricas dos *containers*) e um para o Server da aplicação 3 (métricas das VMs). Já a VM Worker, terá apenas três *containers*, sendo um para o Client da aplicação 1 (transferência de arquivos), um para o Client da aplicação 2 (métricas dos *containers*) e um para o Client da aplicação 3 (métricas das VMs).

comentado. Dessa forma, a criação das imagens não está automatizada, já que as flags estavam sendo alteradas constantemente sem fechar o Vagrant.

Ao longo do desenvolvimento, o Server da aplicação 1 foi modificado a fim de aceitar recebimento de mais de um arquivo. Este arquivo também foi alterado para ficar em num loop infinito, até que seja colocada a opção 0. Dessa forma, é possível ficar enviando o mesmo arquivo diversas vezes. Em suma, o Client ficará enviando arquivos para o Server, podendo escolher entre um dos três arquivos de texto existentes ou a opção “Sair” para encerrar o loop while. O Server, por sua vez, abre o socket e fica esperando receber algo, também dentro de um loop while.