

Laboratório
Concorrente

de

Programação

Lab5



Objetivo

Neste laboratório, exploraremos uma construção essencial de golang para concorrência: **select statements**

Como motivação, iremos construir uma aplicação distribuída. Essa aplicação funciona como um buscador de arquivos para filesharing (pense em Bittorrent). Então, um cliente deve procurar que máquinas na rede armazenam o arquivo buscado. Como chave de busca, ele deve passar o hash do arquivo (calcular tal como o **sum** dos labs passados). Considere que haverá, pelo menos, um grupo de quatro máquinas.

O design da solução é aberto. Mas, alguns direcionamentos podem ser úteis:

1. O time precisará pensar em um modelo de comunicação. Sockets (como mostramos em sala de aula <https://github.com/thiagomanel/fpc/blob/master/go/clock.go>) funcionam. Outros esquemas de comunicação podem ser até mais interessantes (<https://grpc.io/docs/languages/go/quickstart/>)
2. Você precisará pensar em um esquema de organização. Completamente P2P ou cliente-servidor?
3. Você precisa pensar em um esquema de descoberta. Quais as máquinas que fazem parte do sistema.
4. Desempenho continua sendo importante. Pense em minimizar o tempo total, do ponto de vista de um cliente, para obter a lista de máquinas (seus IPs). Embora a lista de otimizações possíveis seja enorme, primeiro **FAZ FUNCIONAR!**
5. Considere que os arquivos buscados estão em um diretório no /tmp. Por exemplo, /tmp/dataset

extra-super-cúbico

1. Você vai escanear todos os arquivos ao receber uma nova requisição? Não deveria manter estado sobre isso?

2. E se, ao invés da lista de máquinas, o programa fizesse download do arquivo?
 - a. E se, ao invés de baixar de uma única máquina o programa baixar partes do arquivo de máquinas diferentes.
3. Um sistema distribuído é um sistema que falha porque uma máquina que você não faz ideia que existe falhou. E se uma máquina com a qual você estiver se comunicando falhar durante a comunicação?
4. E se o sistema permitir entradas de novas máquinas dinamicamente (depois de estar funcionando)? Como descobrir que novas máquinas fazem parte do sistema?

Detalhes da entrega

O código desenvolvido será entregue através de um repositório no github. Para isso, no início da aula, crie um repositório **privado** (repositórios públicos não serão considerados). Adicione os professores como colaboradores (<https://github.com/thiagomanel/> e <https://github.com/giovannifs>).

Além disso, é muito importante que você faça commits frequentes. Qualquer mudança relevante em uma função, deve ser comitada.

Considere que o programa deve ser usado assim:

```
$ ./bodocongo_bay search $file_hash
ip.machine.3
ip.machine.1
```

Ou seja, ao passar o hash de um arquivo que se busca, retorna uma lista de IPs de máquinas que armazenam o arquivo.

Prazo

17/set/24 23:59

Appendix

<https://gobyexample.com>
<https://go.dev>
<https://go.dev/tour/list>