

# Computação Distribuída – Guião 04

Diogo Gomes & Mário Antunes

Abril 2019

## 1 Introdução

O objectivo deste guião é desenvolver um sistema de distribuição de carga HTTP (*load balancer*). Um *load balancer* é um componente que actua como *reverse proxy* e distribui tráfego por um grupo de servidores. São usados para aumentar a capacidade (número de utilizadores em paralelo) e a fiabilidade de um serviço. Estes componentes permitem aumentar a performance dos servidores pois diminuem a carga colocada nos mesmos.

Neste guião vamos considerar as seguintes políticas para escolher qual o servidor de *back-end*:

**N to one:** todo o tráfego é enviado para o primeiro servidor (desenvolvido apenas para o guião, não é usado na prática)

**Round Robin:** uma das políticas mais simples. Cada pedido é enviado para o próximo servidor na lista. Quando o último servidor é usado, volta ao primeiro.

**Least Connections:** é escolhido como próximo servidor aquele que tem menos ligações efectuadas até ao momento.

**Least Response Time:** é escolhido como próximo servidor aquele que teve menos tempo a servir clientes

É fornecido um código de exemplo que apenas implementa a primeira política, sendo necessário implementar as restantes. O *load balancer* distribui tráfego por servidores de uma aplicação Web que calcula o valor de  $\pi$  com diferentes níveis de precisão. Quando maior a precisão, mais lento será o

tempo de execução. São também fornecidos dois *scripts* em *bash* para facilitar a execução de código em ambientes UNIX. Para mais detalhes, consultar o README.md no repositório.

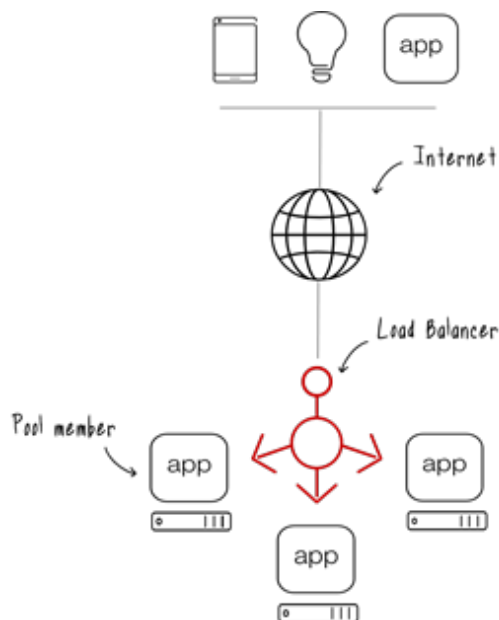


Figura 1: Esquema simplificado de um *load balancer*.

## 2 Objectivos

- Verificar o código de exemplo fornecido [1].
- Implementar as seguintes políticas: Round Robin, Least Connections, Least Response Time.
- Implementar uma cache (limitada a 5 entradas) que armazena as últimas trocas de mensagens e acelera o processo.
- **Desafio:** Implementar um mecanismo de *cache* que permita acelerar a aplicação. A cache deve se encontrar limitada a 5 entradas (por motivos puramente académicos).

### 3 Notas

- Devido a restrições na rede Eduroam [2], o sistema apenas funcionará no seu computador local.
- Como ponto de partida podem usar o código partilhado em [1].
- O código deve ser armazenado/partilhado no GitHub [3] classroom - criar projecto via <https://classroom.github.com/a/ecW1vnum> - (tutorial de git [4])

### Referências

- [1] mariolpantunes. chord. <https://github.com/mariolpantunes/load-balancer>, 2019. [Online; accessed 2019-03-18].
- [2] STIC. Rede wireless – eduroam. <https://www.ua.pt/stic/PageText.aspx?id=15224>, 2019. [Online; accessed 2019-02-24].
- [3] GitHub. Dep. electrónica telecomunicações e informática. <https://github.com/detiuaveiro>, 2019. [Online; accessed 2019-02-08].
- [4] Roger Dudler. git – the simple guide. <http://rogerdudler.github.io/git-guide/>, 2019. [Online; accessed 2019-02-08].