

# Computação Distribuída – Guião 03

Diogo Gomes & Nuno Lau

Março 2020

## 1 Introdução

O objectivo deste guião é desenvolver uma DHT, nomeadamente CHORD [1–3]. Chord é um protocolo e um algoritmo que implementam uma DHT (distributed hash table). Uma DHT (ver Figura 1) armazena pares de chaves-valor distribuídos entre diferentes computadores (chamados nós). Um nó só armazena os valores pelo os quais é responsável. O protocolo Chord define como as chaves e os valores devem ser distribuído pelos nós.

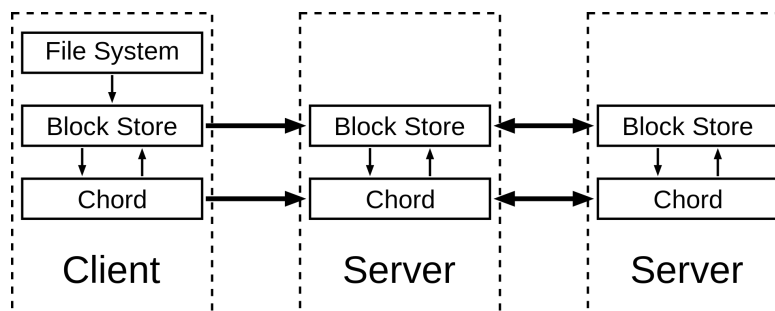


Figura 1: Esquema simplificado de uma DHT. Crédito Wikipédia

Tanto os nós como os objectos são identificados por um ID, gerado através de hash consistente. Tipicamente SHA-1 é usada como algoritmo de hash. A consistência do algoritmo de hash é crucial para maximizar a performance da DHT.

Baseados no valor de ID os nós são organizados em anel ordenado (ver Figura 2). De forma simplificada, a pesquisa no CHORD é efectuada da seguinte forma:

- Cliente contacta um nó do anel e requisita uma determinada chave
- O nó verifica se possui a chave em questão
- Se não possui, propaga a mensagem para o próximo nó no anel
- Se possui, retorna o objecto ao cliente

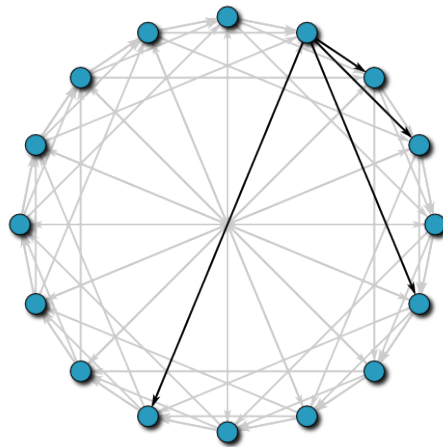


Figura 2: Esquema simplificado do *Chord*. Crédito Wikipédia

## 2 Objectivos

- Estudar o código fornecido.
- O código já implementa os principais métodos usados para a criação do anel e garante a sua correcção
- No entanto as funções que implementam a inserção e pesquisa de objectos não estão completas
- As funções mencionadas anteriormente só verificam se os objectos existem localmente e não propagam para o resto do anel
  - Implementar a pesquisa remota (performance  $O(n)$ )

- Expandir o código para suportar *Finger Tables* (em vez de apenas conhecer um único sucessor)
- Implementar a pesquisa remota que utiliza as *Finger Tables* (performance  $O(\log(n))$ )
- Suportar a adição e remoção de nós em *runtime*.
- Comunicação através de UDP [4]
- As mensagens deverão ser codificadas em Pickle [5].
- Ajustar o número de nós e precisão da hash como necessário.

### 3 Notas

- Devido a restrições na rede Eduroam [6], o sistema apenas funcionará no seu computador local (o código já está preparado para isto).
- Como ponto de partida devem usar o código partilhado em [7].
- O código deve ser armazenado/partilhado no GitHub [8] classroom - criar projecto via <https://classroom.github.com/g/nd9fEmb1> - (tutorial de git [9])

### Referências

- [1] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, August 2001.
- [2] Wikipedia. Chord (peer-to-peer). [https://en.wikipedia.org/wiki/Chord\\_\(peer-to-peer\)](https://en.wikipedia.org/wiki/Chord_(peer-to-peer)), 2019. [Online; accessed 2019-03-18].
- [3] Maarten van Steen and Andrew S. Tanenbaum. *Distributed Systems*, pages 246–251. Maarten van Steen, third edition, 2018.
- [4] Python. Socket programming howto. <https://docs.python.org/3/howto/sockets.html>, 2019. [Online; accessed 2019-02-08].

- [5] Python. pickle – python object serialization. <https://docs.python.org/3/library/pickle.html>, 2019. [Online; accessed 2019-02-24].
- [6] STIC. Rede wireless – eduroam. <https://www.ua.pt/stic/PageText.aspx?id=15224>, 2019. [Online; accessed 2019-02-24].
- [7] mariolpantunes. chord. <https://github.com/mariolpantunes/chord>, 2019. [Online; accessed 2019-03-18].
- [8] GitHub. Dep. electrónica telecomunicações e informática. <https://github.com/detiuaveiro>, 2019. [Online; accessed 2019-02-08].
- [9] Roger Dudler. git – the simple guide. <http://rogerdudler.github.io/git-guide/>, 2019. [Online; accessed 2019-02-08].