

# Laboratório 5 parte 1

---

Nome: Breno Rosa Tomé DRE: 113026430

## 1

---

Fornecer um protocolo simples e eficiente para busca de nós em sistemas peer to peer com entrada e saída rápida de nós

## 2

---

Pode ser usado em sistemas distribuídos para resolver problemas de balanceamento de carga, descentralização, escalabilidade, disponibilidade e nomeação flexível, alguns exemplos citados:

- **Cooperative mirroring** : Um grupo de hosts pode hospedar dados dos hosts parceiros, dividindo a carga entre todos nós e ajudando a mitigar situações de pico, assim cada host só precisa disponibilizar recursos para sua carga padrão e não para sua carga máxima.
- **Time shared storage** : Do mesmo modo do cooperative mirroring um grupo de hosts pode usar chord para cooperar, porem com apenas parte dos nós ativos ao mesmo tempo e com o conteúdo disponível através do resto do grupo.
- **Índices distribuídos** : Permite buscar por palavras chave um arquivo e receber o nó que contem ele
- **Busca combinatória de larga escala** : Permite mapear uma chave

## 3

---

No funcionamento simplificado ele testa se o nó atual é o sucessor da chave buscada, caso contrario ele repete o procedimento com o sucessor do nó atual até chegar no nó sucessor da chave

os nós não guardam informações sobre todos outros nós apenas de  $O(\log N)$  nós e uma busca precisa de  $O(\log N)$  mensagens. usa o método de hashing consistente

## 4

---

Os nós não guardam informações sobre todos outros nós apenas de  $O(\log N)$  nós e uma busca precisa de  $O(\log N)$  mensagens. usa o método de hashing consistente

quando um nó entra parte das chaves do seu sucessor é passada para ele, e caso ele saia as chaves voltam para o sucessor

cada nó executa uma rotina de estabilização periodicamente, nessa estabilização cada nó pergunta ao seu sucessor qual seu predecessor, para descobrir se novos nós entraram no sistema, caso um nó tenha entrado entre ele e seu sucessor ele registra o novo sucessor

Também são chamadas rotinas de correção de tabela que é onde novos nós recebem suas tabelas com referências para os nós que conhecem, e chamadas de verificação do predecessor onde são descoberto os nós que falharam ou saíram do sistema

em cada passo dessa sequencia de chamadas todos nós continuam sendo encontráveis, e é possível que ocorram erros na organização caso muitos nós entrem e saiam no meio das chamadas, por isso a rede necessita de manutenção periódica.

## 5

---

simplicidade, corretude provável e performance provável

Algumas comparações foram fornecidas:

- **DNS**: Depende de um diretório raiz especial e de gerenciamento manual de informação de roteamento, e só funciona bem com nomes estruturados.
- **Frenet**: Fornece anonimidade, para isso não dá responsabilidade a nós e busca em cópias cacheadas, oque leva as buscas a custarem mais que no chord.
- **Ohaha**: Faz roteamento de modo similar ao Freenet e compartilha algumas de suas fraquezas.
- **The Globe system**: Organiza os nós numa hierarquia geográfica e distribui os nós raiz, o chord consegue fazer essa distribuição sem uma hierarquia, mas não consegue se aproveitar da localidade de rede.
- **Tapestry**: Similar ao Chord, porém mais complexo, pode garantir que queries nunca viagem uma distancia maior que o nó onde está armazenado o resultado, ele consegue isso guardando nos nós informações sobre a topologia encontrada no caminho de nós percorrido na entrada do nó no sistema.
- **Napster e Gnutella**: fornecem buscas por palavras chave, oque é difícil em todos os sistemas, Napster usa um índice central oque é um ponto único de falha, e Guntella envia cada query para todo sistema, oque sobrecarrega o sistema.