Distributed Hash Tables

Danielle Santos da Silva

Marcelo Borghetti Soares

Tiago Alves Macambira

Roteiro

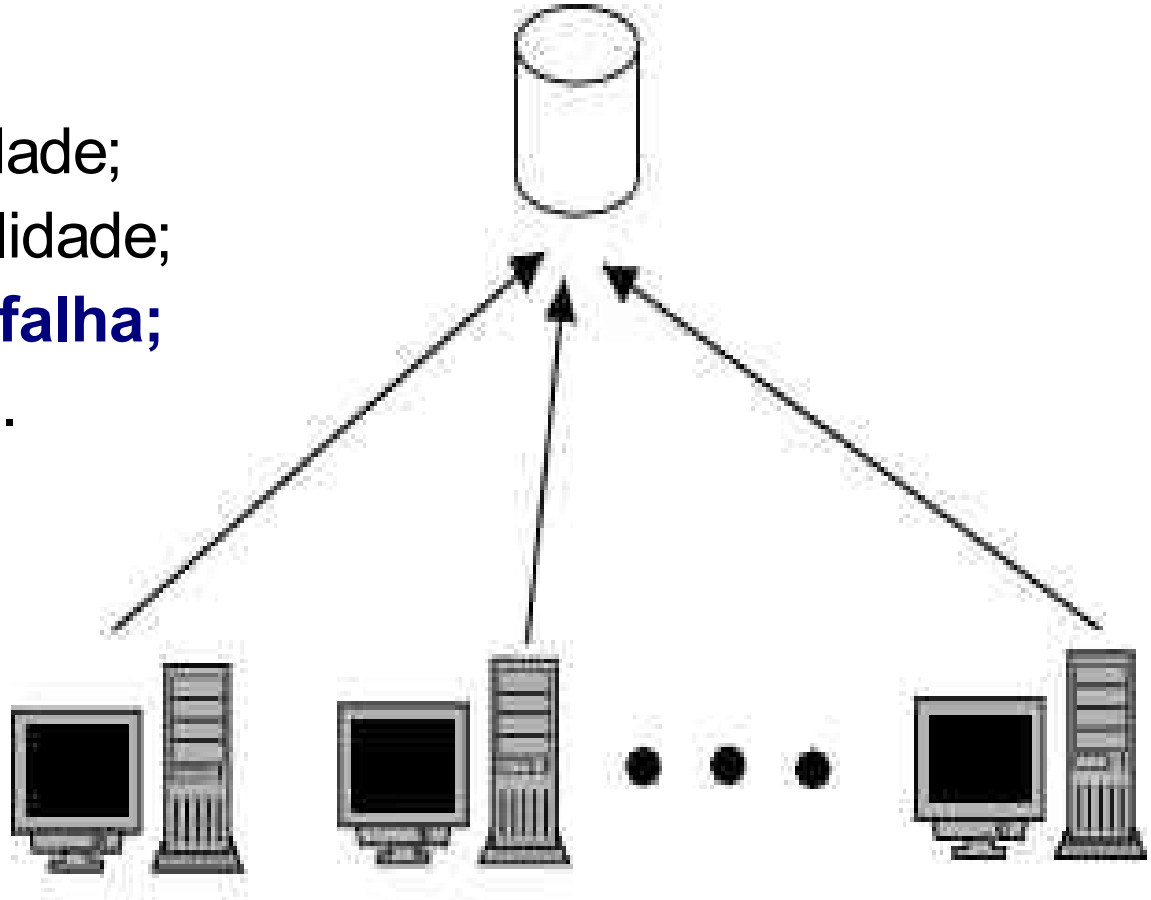
- Introdução
- DHTs
- Chord
- Outros algoritmos para DHT
- Aplicações usando DHT

Introdução

- Lookup Problem: Dado um conjunto de nós dinâmicos, como achar um determinado dado armazenado nesse sistema?
- Como solucionar o problema?
  - Base de Dados Centralizada
  - Sistemas Hierárquicos
  - Sistemas Híbridos
  - DHTs

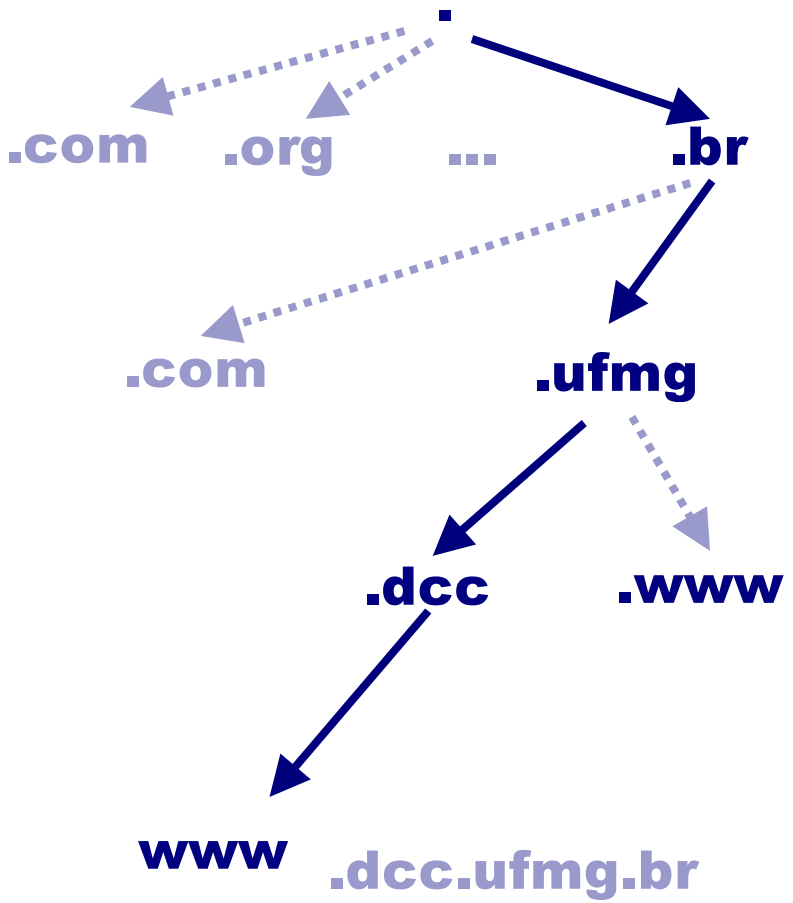
Introdução: Base de Dados Centralizada

- Características:
  - Pouca escalabilidade;
  - Problemas flexibilidade;
  - **Ponto único de falha;**
  - Caro e vulnerável.
- Ex.:
  - Napster;
  - Audiogalaxy;
  - Google.



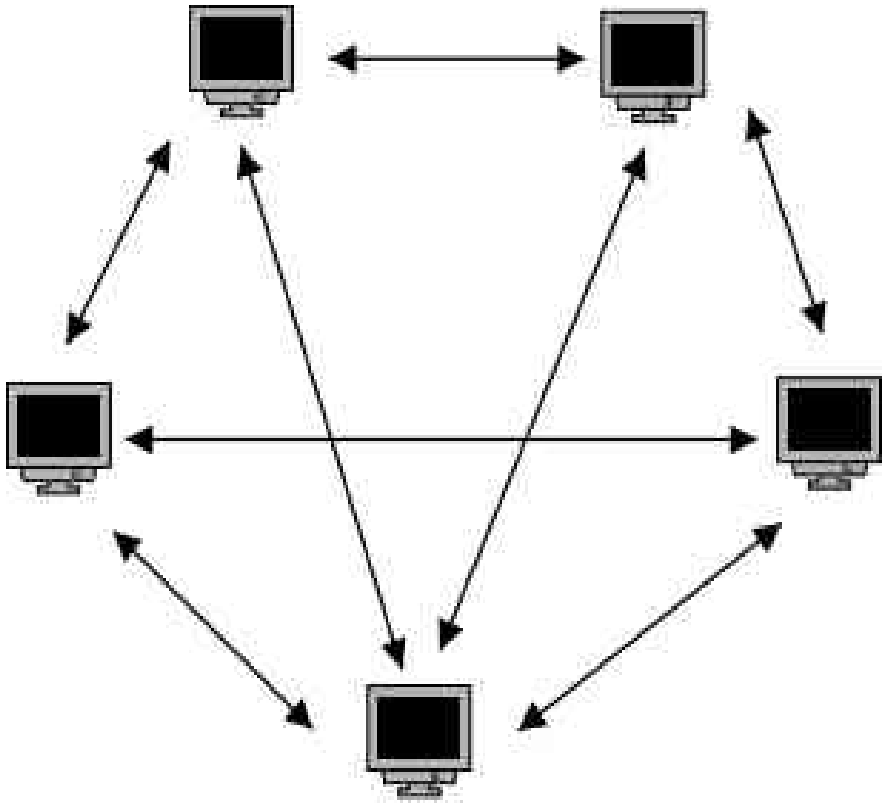
Introdução: Sistema Hierárquico

- Características:
  - Escalabilidade;
  - Balanceamento de carga desigual;
  - Pontos críticos de falha: dependência dos elementos próximos à raiz.
- Ex.: DNS.



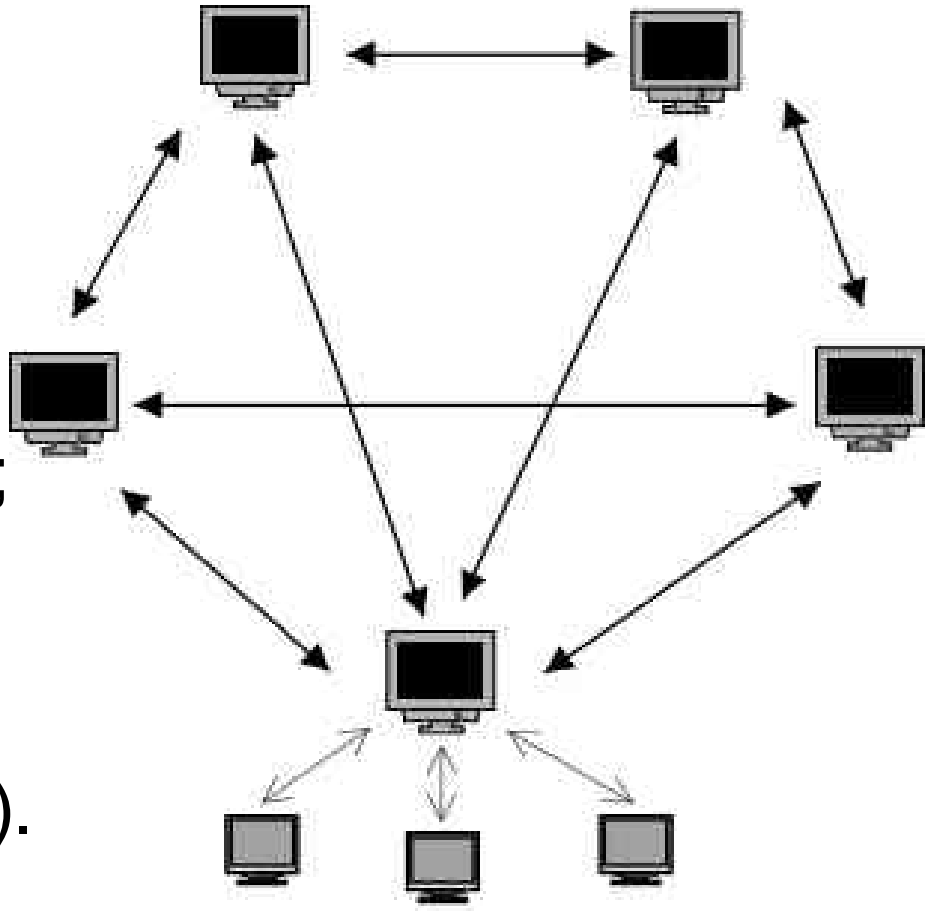
Introdução: Sistema Simétricos

- Características:
  - Rede auto-organizável em uma estrutura de malha eficiente;
  - Balanceamento do custo da busca;
  - Abordagem de busca tipo “Broadcast”: sem escalabilidade.
- Ex.: Gnutella.



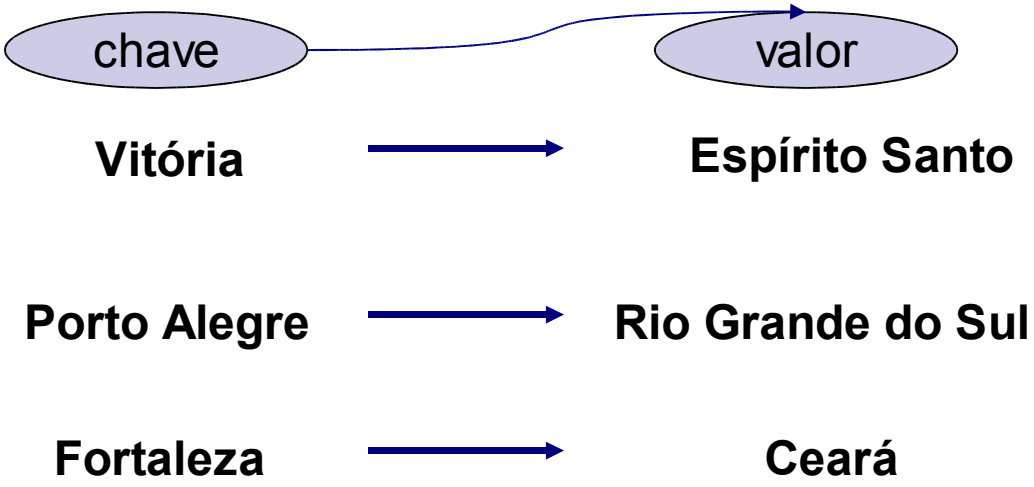
Introdução: Sistema Híbridos

- Características:
  - Escalabilidade;
  - Sem garantias de obtenção de objetos;
- Ex.:
  - KaZaA (Supernode);
  - Gnutella (Superpeer).



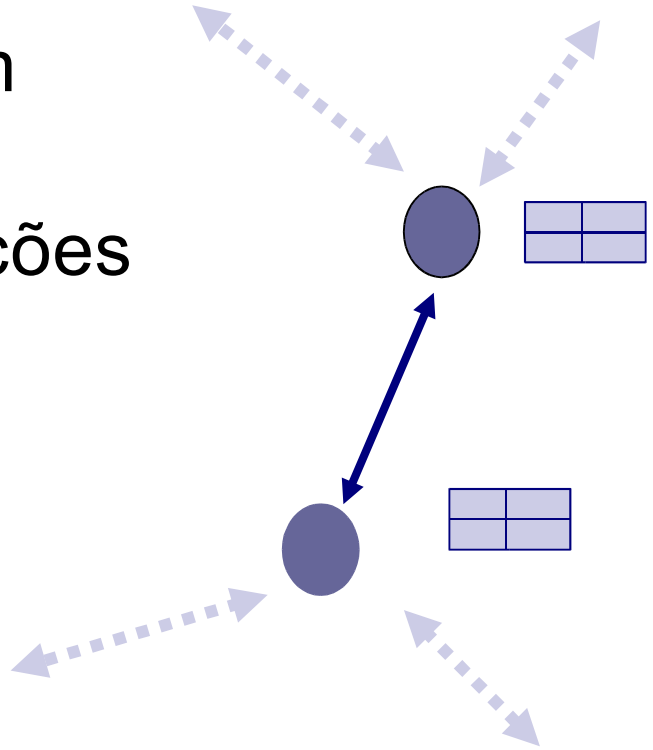
Tabelas Hash: Definição

- Tabelas Hash:
  - Mapeiam chaves em valores (dados)



DHT: Definição

- Distributed Hash Tables
- Dados distribuídos em múltiplos nós
- Nós mantêm informações sobre seus vizinhos
- Operações
  - Pesquisa
  - Inserção
  - Deleção

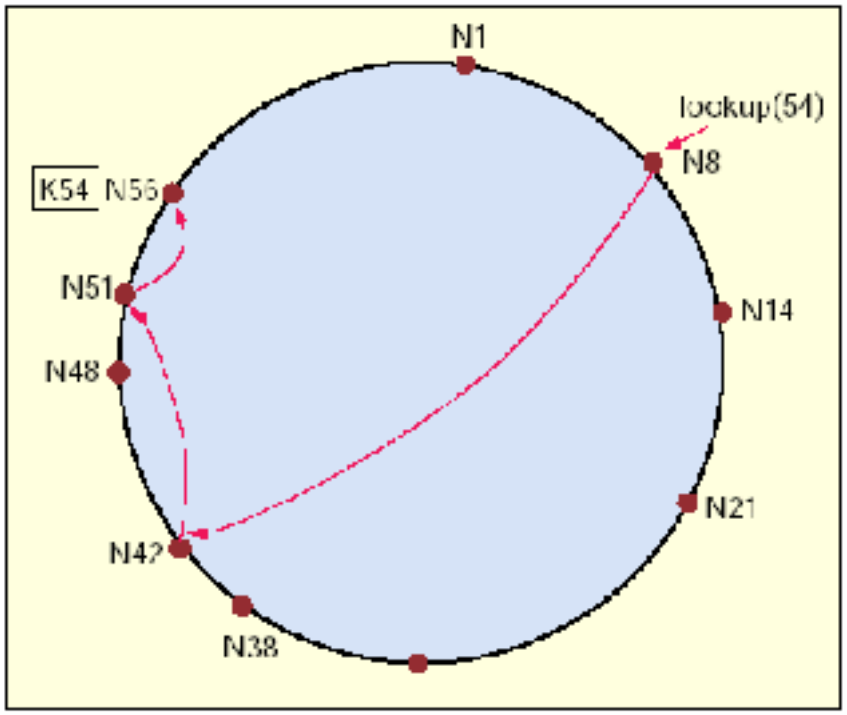


DHT: Definições

- Robusta
  - Trabalham com redes muito grandes
- Escalável
  - Mantêm poucas informações localmente
  - Buscas percorrem nó máximo O(log N) nós
- Carga Balanceada

DHT: Funcionamento

- Operação *lookup(key)*
- Repassar uma consulta para
  nó apropriado
- Os nos mantem informações sobre outros nós em tabelas de roteamento.

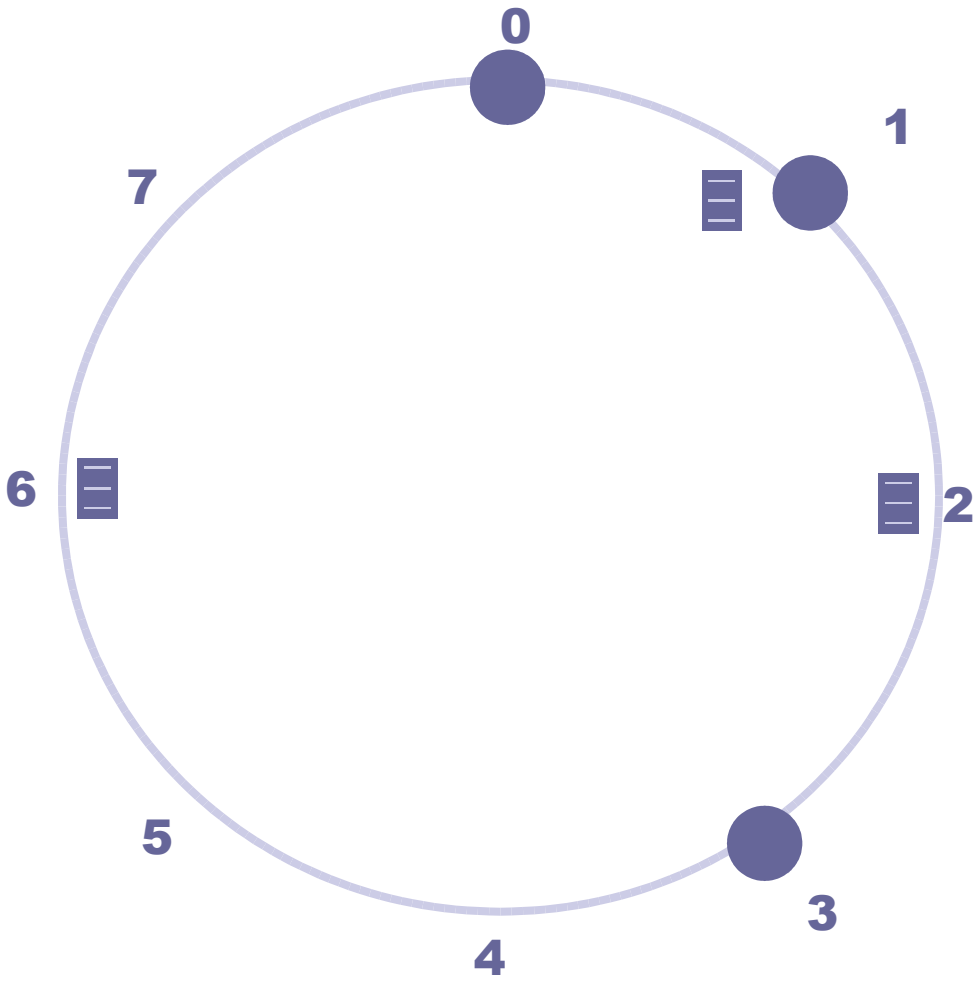


Chord

- Algoritmo para DHT
  - Completamente decentralizado
  - Simétrico
  - Roteamento distribuído
- Custos:
  - Buscas:
    O(log **N**) mensagens
  - Manutenção/Estado:
    O(log **N**) entradas
- Application-Layer Overlay Network

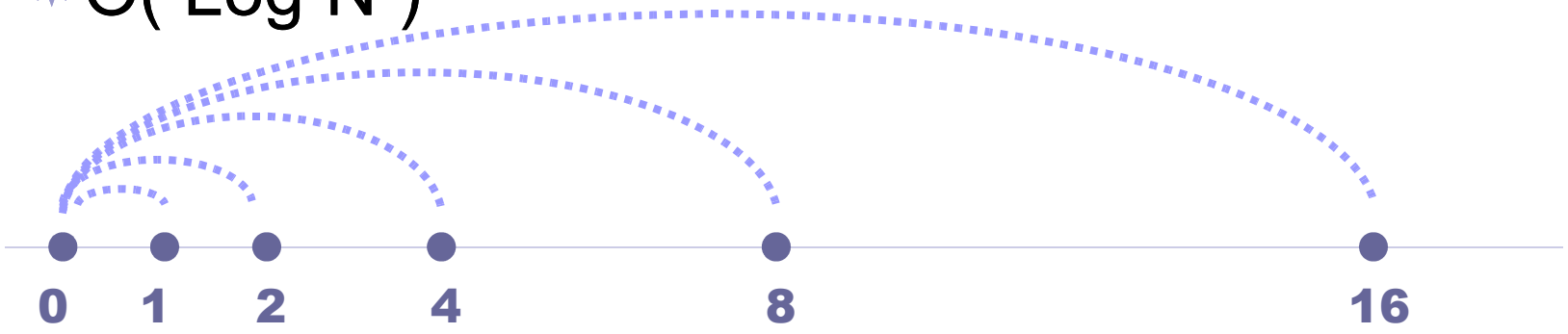
Identificandos Nós e Dados

- Identificadores de *m*-bits
  - Hash
  - SHA-1, MD5
- Espaço de *m*-bits
  - Círculo de Identificadores



Finger-Table

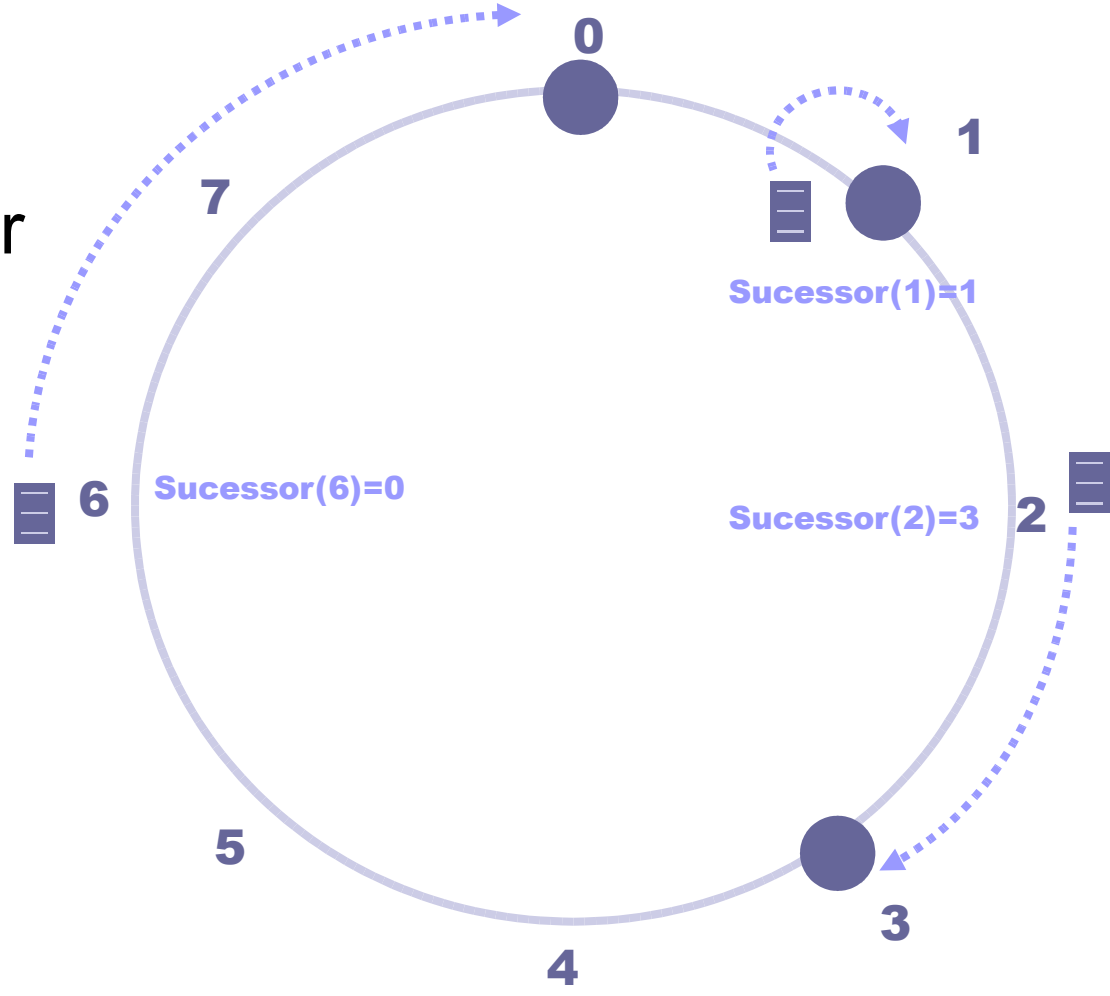
- Tabela de roteamento
- *m*-entradas
  - Espaçamento Exponencial
    - Baseado em Skip Lists
  - O(Log N)



$$f[i] = m + 2^{(i-1)}$$

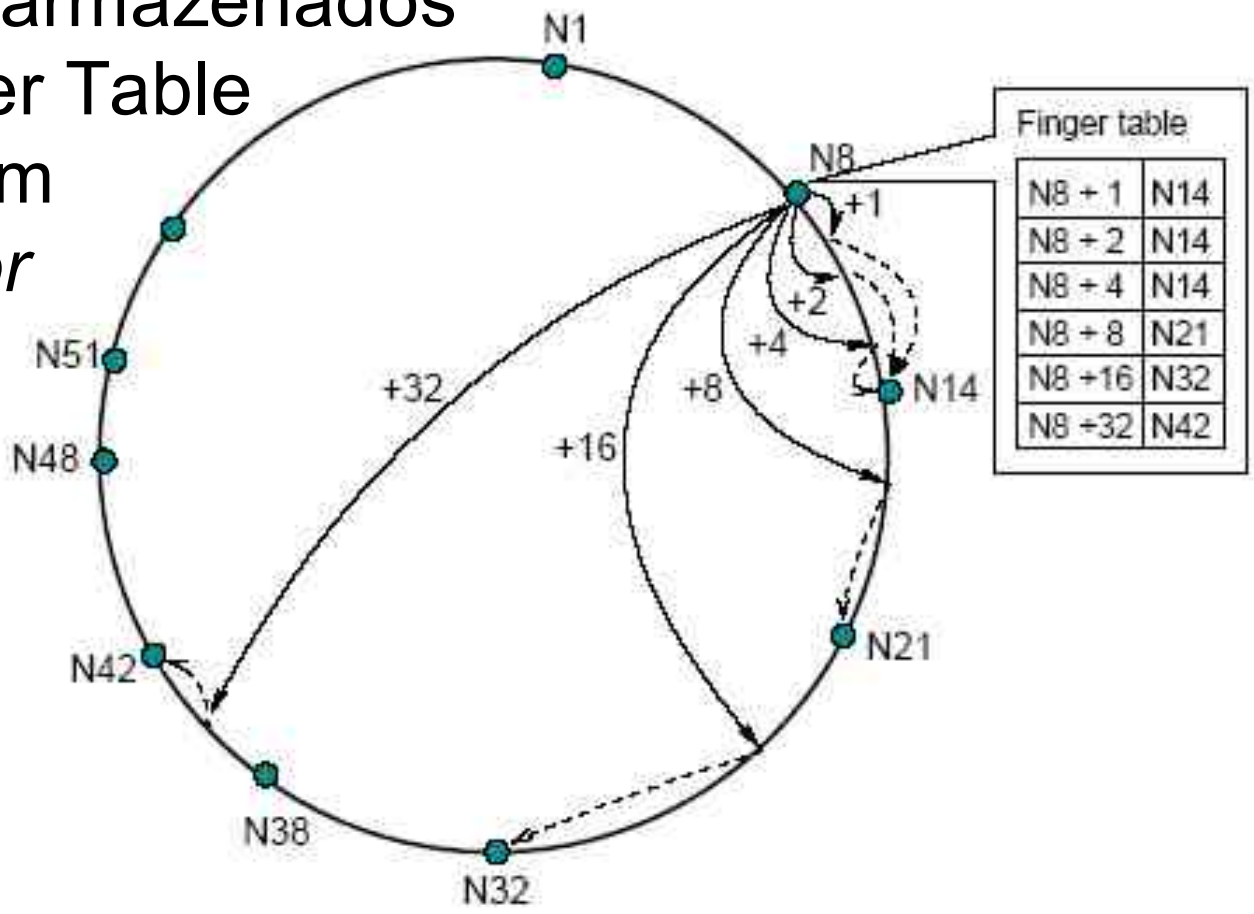
O sucessor

- Nó sucessor (k)
  - Responsável por armazenar a chave K
- Proximo nó no sentido horário



Finger Table e *sucessor*

- Valores armazenados na Finger Table respeitam *sucessor*

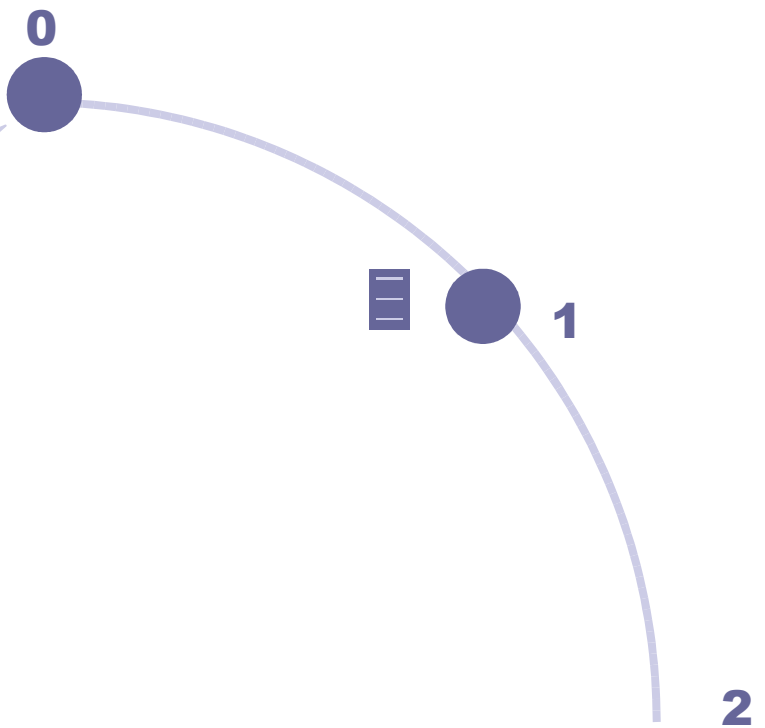


Localizando um dado (cont)

- Busca por *k* consiste em achar o *sucessor (k)*
  - Procura inicia na *Finger-table*

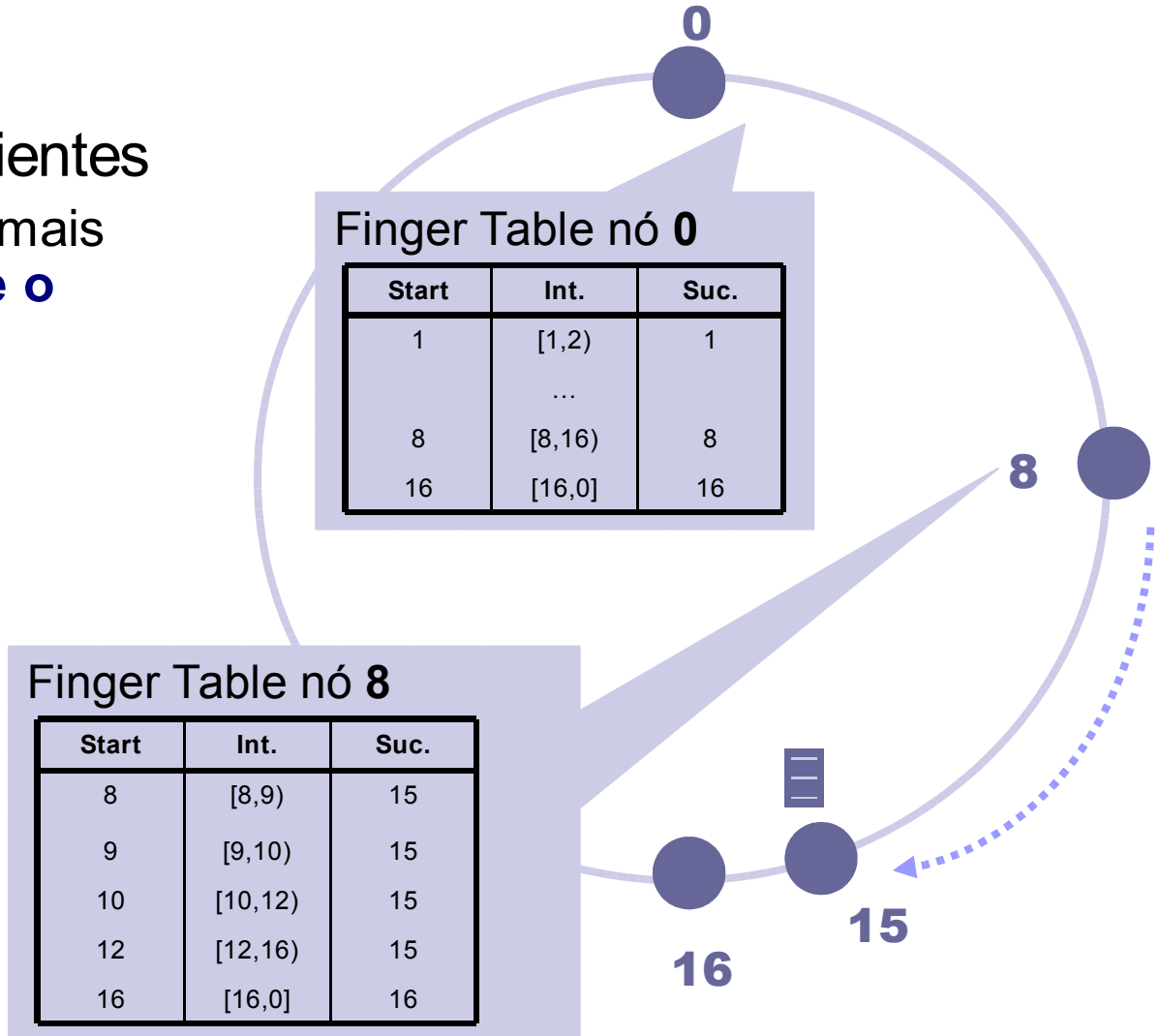
Finger Table

Start	Int.	Suc.
1	[1,2)	1
2	[2,4)	3



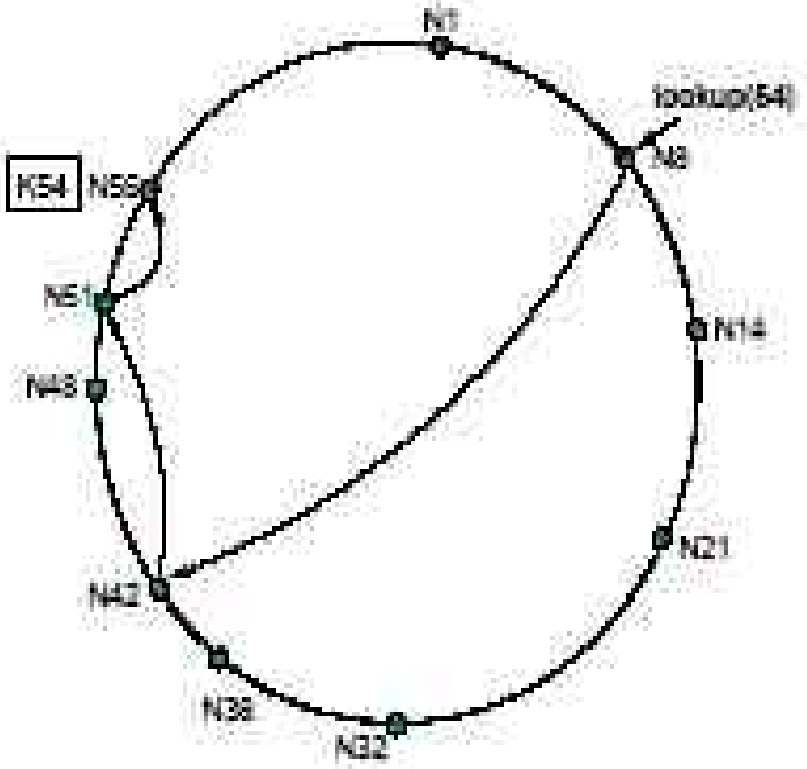
Localizando um dado (cont)

- Se o nó não tem informações suficientes
  - Recorre ao nó *n'* mais próximo de k **que o preceda**
  - *n'* possui informações mais *localizadas* sobre *k*
  - *n' continua a busca …*



Localizando um dado (cont)

- No máximo
  O(Log N) Passos
  - Expaçamento exponencial
  - Distância cai pela metade a cada passo

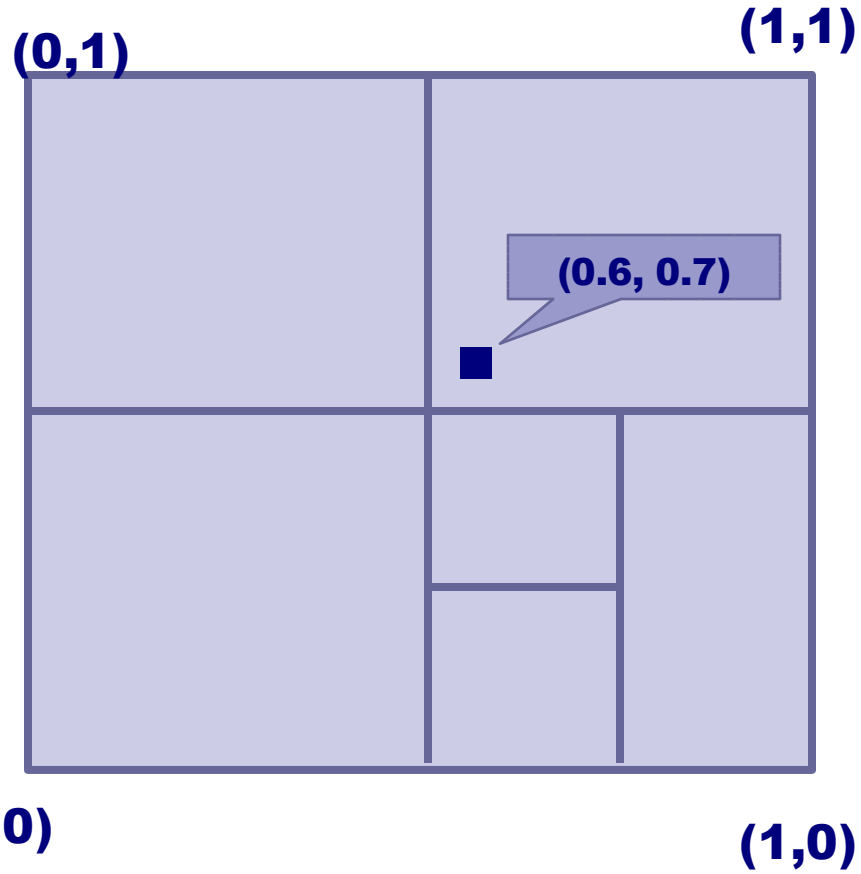


Outros Algoritmos para DHT

- A maior diferêncas entre os diversos algoritmos para DHT
  - estrutura de dados usada para roteamento
  - Chord : Skiplist
- Propriedades são muito semelhantes entre os diversos algoritmos
- Outros exemplos
  - CAN;
  - Pastry;
  - Symphony;
  - Koorde, etc

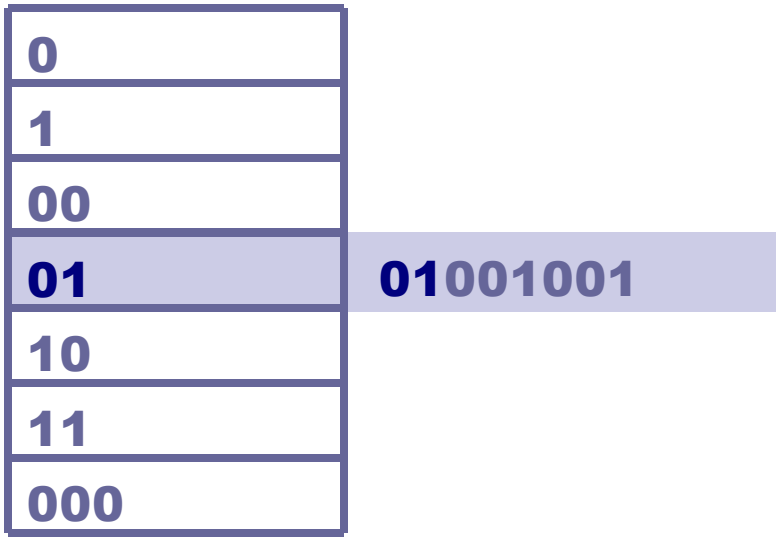
Outros Algoritmos: CAN

- Nodos Mapeados num espaço cartesiano *d*-dimencional
  - Nó responsável por uma *área*
  - Pesquisa repassada para vizinho mais próximo geometricamente do ponto
- Custos
  - Busca: O(**d**$N^{1/d}$)
  - Manutenção/Estado: O(**d**)



Outros Algoritmos: Pastry

- Identificadores de **m**-bits em base 2^b
  - Matriz de de vizinhança
  - Repassa busca para vizinho com maior prefixo
- Custos:
  - Busca: O(log N)
  - Manutenção/Estado O(log N)



Aplicações

- Substitutos para o DNS
  - *Serving DNS using a Peer-to-Peer Lookup Service* , Cox, Muthitachareon and Morris
- Sistemas de arquivos Distribuídos
  - CFS, The Chord Project
- Application-level Multicast
  - *Application-level Multicast using CAN*
- File-sharing
- Etc...

Referências

- Gerais
  - *Looking Up Data in P2P Systems*, Balakrishnan, Kaashoek, Stoica, Karger, Morris. Communications of the ACM – Feb/2003
  - Routing Algorithms for DHTs: Some Open Questions. Ratnasamy, Shenker, Stoica
- Chord
  - http://www.pdos.lcs.mit.edu/chord/
  - A Scalable Peer-to-peer Lookup Service for Internet Applications
  - Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service

Anexo: Skip Lists

- Estrutura de dados probabilística
- Pode ser usada no lugar de árvores balancedas
  - A probabilidade e uma skiplist ser não-balanceada é irrelavante
- Algoritmos para fazer CRUD em Skip Lists são mais simples e eficientes
- Referência
  - *A Probabilistic Alternative to Balanced Trees*, Bill Pugh

FIM

- Dúvidas?!

RESUMÃO

- Introdução
  - Lookup Problem
    - Napster, gnutella, kaaza, DHT
- DHTs
  - O que são
  - caracteristicas
  - Aspectos comuns
  - Aplicações
- Algoritmos para DHT
  - CAN, Pastry
  - Chord
    - Caracterização
      - Limites inferiores para busca
      - Roteamento
      - Estrutura espacial para mapeamento das chaves
- Aplicações
  - Application-layer multicast
  - Etc...
- Questões em aberto...
  - http://citeseer.nj.nec.com/cache/papers/cs/25855/http:zSzzSzwww.cs.rice.eduzSzConferences zSzIPTPS02zSz174.pdf/ratnasamy02routing.pdf