Distributed Hash-tables and Scalable Content-Addressable Network (CAN)

Ines Abdelghani

Ferienakademie im Sarntal 2008
FAU Erlangen-Nürnberg, TU München, Uni Stuttgart

September 2008

Overview

Overview

1. Introduction
2. Distributed Hash Tables (DHT)
   - Basic Concept of DHT
   - DHT Categories
   - DHT Properties
   - DHT Based Peer-to-Peer Networks

Overview

Overview

Possible Peer-to-Peer Networks Structures

- Client-Server Structure:
  - Information about data locations stored by a central server
  - Limits: scalability, update of data, single-point-failure
  - *Example:* **Napster**
- Flooding:
  - Not all data can be found
  - *Example:* **Gnutella**
- Distributed Hash Tables:
  - Decentralized, distributed system

Hash Tables

- `Definition:` Data structure, array for storing a set of data by mapping every item x to a hash value $h(x)$ with:
  - Universe U
  - Array $T[1, \ldots, m]$, Table size m
  - Hash function $h : U \rightarrow \{0, 1, \ldots, m-1\}$

Properties of hash function h

- Choice of h: Need to meet quite demanding properties
  - High efficiency: $h(V)$ easy to compute
  - Security: h one-way, hard to invert
    $\Rightarrow$ cryptographically secure functions(MD-5, SHA-2,...)
  - Collision-free: for any x, infeasibility of finding another x' such as $h(x) = h(x')$
    $\Rightarrow$ Hard to satisfy: Image set is usually smaller than input set
  - Balanced mapping

  $\rightarrow$ Designing such functions: challenging task

Example for a hash function

- h: Modulo 5 function
- $h(x) = x \bmod 5$
- Let x be the file name '*music.mp3*'
- ASCII code of x = 870.920.545.682.538.843.149
- Hash Value of x:
  $h(\mathit{music.mp3})$ =ASCII-code(*music.mp3*) mod 5 = 4
  $\rightarrow$ Store item x at the position 4 in the array $T[1, \ldots, m]$

Basic Concept of Distributed Hash Tables

- Array T is divided among peers

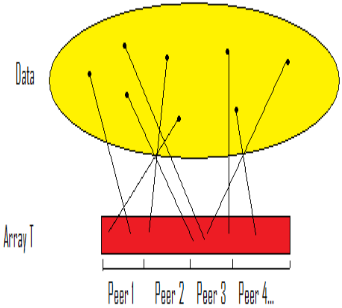
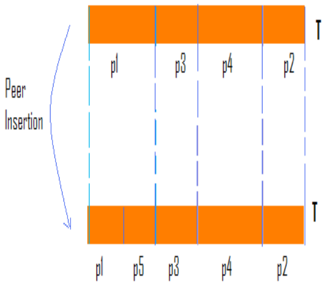


Figure: Distributed Hash Tables

Dynamic Partitioning of the Keys Set Among Nodes

- Node arrival:
  - Division of a certain keys subset between the active node responsible for this subset and the new node
  - Adding the node in the routing structure
  - Updating the routing information



Dynamic Partitioning of Keys'Set Among Nodes

- Node departure:
  - Allocation of associated keys subset to neighboring nodes
  - Data migration to new responsible nodes
- Node failure:
  - Use of redundant routing paths and nodes
  - New allocation of corresponding keys subset to active nodes

DHT Categories

- Deterministic DHTs:
  - First deterministic DHTs: `CAN`, `Chord`, `Pastry`
  - Overlay connection: function of the current set of node IDs
  - Only two sources of uncertainty:
    - Size of the network not known accurately to all participants
    - Mapping of subset of keys to nodes not exactly even
- Randomized DHTs
  - `Viceroy`: first randomized protocol for DHT routing

DHT Properties

- High scalability
- High robustness against frequent peers departures and arrivals
- Self organization: No need for a central server

  $\rightarrow$ No single-point failure problem

  $\rightarrow$ Higher fault tolerance

Why DHT in Peer-to-Peer Networks?

- Balanced distribution of data among nodes:

  Avoid having nodes overloaded with data

- Minimum disruption by nodes joining, leaving and failing:

  Only a small part of the network concerned by a change in the set of participants

  $\rightarrow$ Consistency

DHT Based Peer-to-Peer Networks

- CAN
- Chord
- Tapestry
- Pastry
- Kademlia
- P-Grid

CAN: Historical Context

- Most of in that period existing Peer-to-Peer designs not scalable: Napster (central structure), Gnutella (unsecured, flooding for data lookup)

  $\rightarrow$ 2001: Content-addressable Network CAN

  Literature: Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp, Scott Schenker: *A Scalable Content-Addressable Network*. SIGCOMM 2001 : 161 – 172

CAN Design

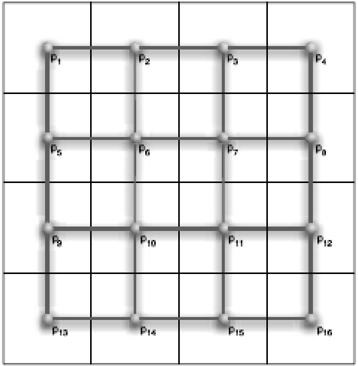


Figure: Ideal structure of a two-dimensional CAN

CAN Design

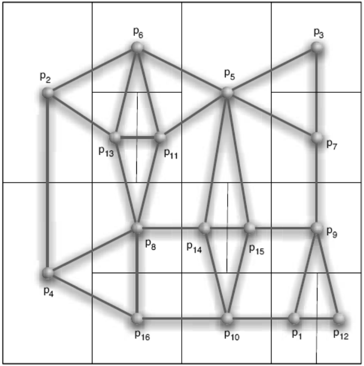


Figure: Typical Structure of a two-dimensional CAN

* Torus Edges not visualized

CAN Construction

- Finding any existing node in CAN
- Finding the corresponding zone to be split
- Notifying the neighbors of the split zone with the new node coordinates and IP adress:

$\longrightarrow$ Allocation of data to all other peers remain unchanged: consistency

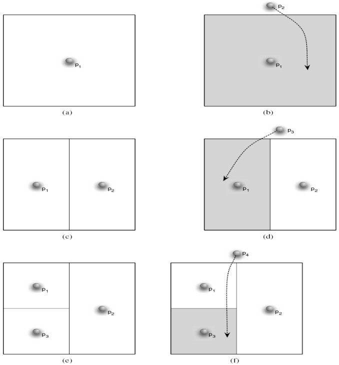


Figure: CAN Construction

Data Distribution Among Nodes in CAN

Question: How uniform is the distribution of the data after peers insertion?

Data Distribution Among Nodes in CAN

Question: How uniform is the distribution of the data after peers insertion?

- Data load proportional to zone area
- Low probability of perfect uniform partitioning

Data Distribution Among Nodes in CAN

- Let p be a peer in CAN
- $R(p)$ rectangle associated to peer p
- $A(p)$ area of the rectangle $R(p)$
- $P_{R,n}$ Probability that the rectangle $R(p)$ is not split after having n peers joining the network
- Lemma 1:

$$P_{R,n} \leq e^{-nA(p)}$$

Proof of Lemma 1

- Let $q = A(p)$
- P_R Probability that $R(p)$ is not split after the insertion of a new peer

- Let $q = A(p)$
- P_R Probability that $R(p)$ is not split after the insertion of a new peer

$$P_R = 1 - q$$

- $P_{R,n}$ Probability that the rectangle $R(p)$ is not split after having n peers joining the network:

$$\Rightarrow P_{R,n} = (P_R)^n = (1-q)^n$$

- For $\forall m \succ 0$: $(1 - \frac{1}{m})^m \leq \frac{1}{e}$
  $\Rightarrow P_{R,n} = (1-q)^n = ((1-q)^{\frac{1}{q}})^{nq} \leq \frac{1}{e}^{nq} = e^{-nq} = e^{-nA(p)}$

Data Distribution Among Nodes in CAN

- Let p be a peer in CAN
- $R(p)$ rectangle associated to peer p
- $A(p)$ area of the rectangle $R(p)$
- c constant

Data Distribution Among Nodes in CAN

- Let p be a peer in CAN
- $R(p)$ rectangle associated to peer p
- $A(p)$ area of the rectangle $R(p)$
- c constant
- Theorem 1: In CAN, after the insertion of n peers, for the probability P_A of having a rectangle $R(p)$ with area $A(p) \geq 2c \cdot \frac{ln(n)}{n}$ we have:

$$P_A \leq n^{-c}$$

Interpretation of Theorem 1

- Theorem 1: In CAN, after the insertion of n peers, the probability P_A of having a rectangle $R(p)$ with area $A(p) \geq 2c \cdot \frac{ln(n)}{n}$ is very low, i.e. $P_A \leq n^{-c}$
- Interpretation: The probability that a zone associated to a given peer is $2c\ln n$ times larger than the average area $\frac{1}{n}$ is smaller than n^{-c}.
- Same interpretation for the amount of data managed by one peer

CAN Basic Operations

- Insertion of (key, value) pairs
- Lookup for (key, value) pairs
- Deletion of (key, value) pairs

Insertion of (key, value) pairs

- Key K_1 mapped onto a point O_1 in the coordinate space by a uniform hash function
- Point R_1 in a zone Z_1
- Peer P_1 owns zone Z_1
- Peer P_1 stores the (K_1, V_1) data

Lookup/deletion of (key, value) pairs

- Requesting peer: P_R
- Requested data (K_R, V_R)
- Coordinates of point O_R calculated by N_R
- O_R in Zone Z_R
- Routing from requesting peer to peer managing the zone Z_R

CAN Routing

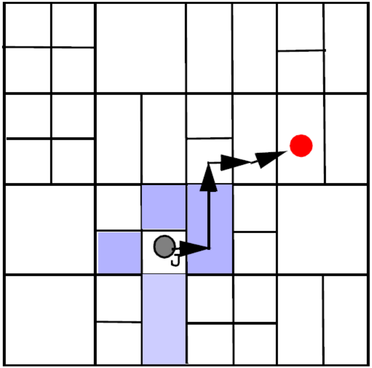


Figure: Request Messages Routing in CAN

CAN Routing

- Coordinate Routing Table in each CAN node:
  - IP address and virtual coordinates of immediate neighbors
- Greedy routing in CAN:
  - Straight line path: $Z_R \rightarrow Z_D$ with:

    P_R: Requesting peer, Z_R : Requesting zone

    P_D: Destination peer, Z_D: Destination zone

CAN Maintenance

- Node departure: Leaving procedure

CAN Maintenance

- Node departure: Leaving procedure
- Node failure:

CAN Maintenance

- Node departure: Leaving procedure
- Node failure: Immediate Takeover Algorithm

CAN Maintenance

- Node departure: Leaving procedure
- Node failure: Immediate Takeover Algorithm

  $\longrightarrow$ Problem: Space fragmentation

CAN Maintenance

- Node departure: Leaving procedure
- Node failure: Immediate Takeover Algorithm

  → Problem: Space fragmentation

  → Solution: Background zone-reassignment algorithm

Node Failure

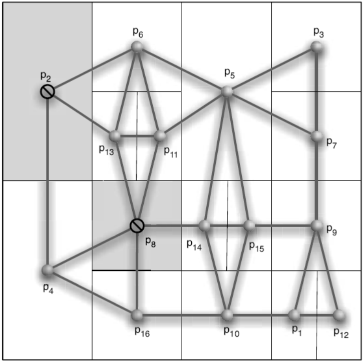


Figure: CAN before node failures

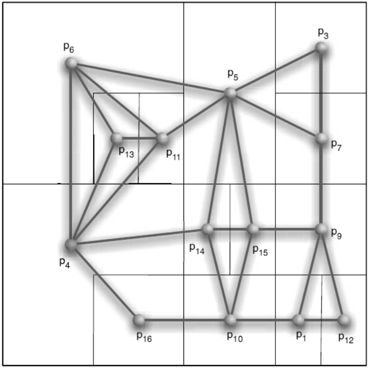


Figure: CAN after node failures

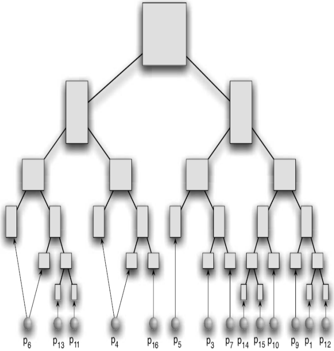


Figure: CAN Tree Presentation

Background zone-reassignment algorithm: simple case

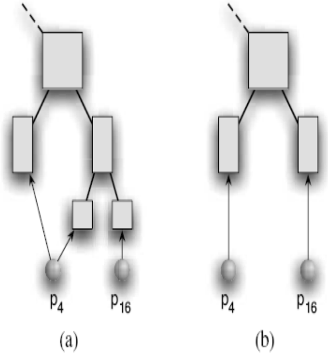


Figure: Zone-reassignment algorithm: simple case

Background zone-reassignment algorithm: complex case

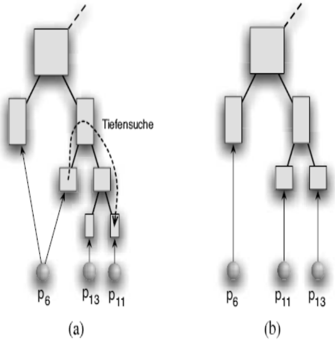


Figure: Zone-reassignment algorithm: complex case

Design Improvements

- Multi-dimensioned coordinate space

Design Improvements

- Multi-dimensioned coordinate space
- Realities: Multiple coordinate spaces

Design Improvements

- Multi-dimensioned coordinate space
- Realities: Multiple coordinate spaces
- Overloading coordinate zones

Design Improvements

- Multi-dimensioned coordinate space
- Realities: Multiple coordinate spaces
- Overloading coordinate zones
- Multiple hash functions

Design Improvements

- Multi-dimensioned coordinate space
- Realities: Multiple coordinate spaces
- Overloading coordinate zones
- Multiple hash functions
- Topologically-sensitive CAN overlay network

Design Improvements

- Multi-dimensioned coordinate space
- Realities: Multiple coordinate spaces
- Overloading coordinate zones
- Multiple hash functions
- Topologically-sensitive CAN overlay network
- Better CAN routing metrics

Design Improvements

- Multi-dimensioned coordinate space
- Realities: Multiple coordinate spaces
- Overloading coordinate zones
- Multiple hash functions
- Topologically-sensitive CAN overlay network
- Better CAN routing metrics
- More uniform partitioning

Design Improvements

- Multi-dimensioned coordinate space
- Realities: Multiple coordinate spaces
- Overloading coordinate zones
- Multiple hash functions
- Topologically-sensitive CAN overlay network
- Better CAN routing metrics
- More uniform partitioning
- Caching and replication techniques

Conclusion

- Distributed Hash Tables:
  - DHT Basic concepts
  - DHT categories
  - DHT properties
- CAN:
  - Basic Design
  - Design Improvements

Proof of Theorem 1

- Let R_i be a rectangle with the area $A(R_i) = 2^{-i}$
- c a constant
- $P_{R_i, c2^i \mathtt{ln} n}$ the probability that the rectangle R_i remains undivided after the insertion of $c2^i \mathtt{ln} n$ peers
- Using Lemma 1:
  $P_{R_i, c2^i \mathtt{ln} n} \leq e^{-A(R_i)c2^i \mathtt{ln} n} = e^{-c \mathtt{ln} n} = n^{-c}$

$$\Rightarrow P_{R_i, c2^i \mathtt{ln} n} \leq n^{-c}$$

Proof of Theorem 1 (Continue)

- We obtained for $P_{R_i,c2^i\ln n}$ the probability that the rectangle R_i remains undivided after the insertion of $c2^i\ln n$ peers:

$$P_{R_i,c2^i\ln n} \leq n^{-c}$$

- Let's consider the insertion of peers stepwise $i = 1, 2, \ldots, \log\frac{n}{2c\ln n}$
- In step i, $c2^i\ln n$ peers are inserted, that divide the rectangle R_i with high probability
- After the $\log\frac{n}{2c\ln n}$-th step, is also the rectangle R_i with area equal to $2c\frac{\ln n}{n}$ high probably divided

- After the $\log \frac{n}{2c\ln n}$-th step, the number of peers N inserted is: $N = \sum_{i=1}^{\log \frac{n}{2c\ln n}} c2^i \ln n = c(\ln n). \sum_{i=1}^{\log \frac{n}{2c\ln n}} 2i$
- $N \leq c(\ln n)2\frac{n}{2c\ln n} = n$
  $\Rightarrow$ The number of peers in CAN in this case is at most equal to n
  $\Rightarrow$ The rectangle with area $2c\frac{\ln n}{n}$ remains undivided after the insertion of n peers with a probability equal to $n^{-c} lnn$

Proof of Theorem 1 (Continue)

- There is at mnost n such rectangles
- The probability that one of this rectangle remains undivided is at most $n.n^{-c}.\texttt{log}n \leq n^{-c+2}$