CSCI 5105

Instructor: Abhishek Chandra

Today

- Attribute-based Naming

Attribute-based Naming

- Identify an entity by its description
  - Want an entity of certain kind
  - Name specified by (attribute, value) pairs
  - Examples:
    - A laser printer on 4th floor in CS dept
    - A mail server for cs.umn.edu domain
- Also called Directory Service
- Can be implemented using:
  - Hierarchical naming
  - Decentralized naming

Hierarchical Implementation

- Convert (attr, value) pairs into structured name space
- Use hierarchical naming techniques

LDAP

- Light-Weight Directory Access Protocol
  - Simplified version of X.500 Directory Service
  - Uses a hierarchical naming scheme
- Directory entries or records:
  - Contain (attribute, value) pairs
- Several implementations:
  - MS Active Directory, Novell Directory services, openLDAP

5

Hierarchical Naming

- Directory entries are arranged in a tree structure
  - Each node has its own record
  - Each node also has pointers to its children records
- Distinguished name (DN):
  - Each record has a unique global name
  - Sequence of Attribute-value pairs
  - E.g.: /C=US/O=UMN/OU=CS
- Relative Distinguished name (RDN):
  - Name w.r.t. parent's DN

6

LDAP Implementation

- Directory Service Agents (DSA)
  - Maintain entries for part of the naming tree
  - Similar to DNS name servers
- Directory User Agents (DUA)
  - Client-side name resolvers
- LDAP supports more advanced queries
  - Search operations on attributes
  - Use indexes, filtering, pruning
- LDAP combined with DNS
  - Root directory node of a tree accessible through DNS

7

Decentralized Implementation

- Map (attribute, value) pairs to identifiers
- Two types of queries:
  - Single-value queries: Each attribute has a single desired value
  - Range queries: Each attribute can have a range of values

8

Distributed Indexing Approach

- Assume d attributes
- Divide name space by attribute:
  - Use a set of servers for each attribute
  - Keep range of values at each server
- How do we query for entity with k attributes?
- Limitations?

9

Space-filling Curves

- Consider N-dimensional attribute space
  - Map to a 1-dimensional space
- Use hashing to distribute the 1-d space among index servers
  - (attr, val) pairs that are close in the space mapped to the same index server

10

Hilbert Space-filling Curves

- Derive a curve that fills the space
  - Done by recursively diving the space into small squares
- Properties:
  - Locality: Two indices that are close on the curve are also close in the N-dimensional space
  - Reverse need not be true

11

Name Space Implementation

- Entities are mapped to N-dimensional space
  - Based on N-dimensional (attr,val) tuples
- Each entity maps to an index on the Hilbert space-filling curve
- Indices are maintained by index servers
  - Can use DHT

12

Name Resolution

- Single-value query:
  - Find the index for the entity
  - Locate the index server
- Range query:
  - Find the curve indices corresponding to the range in the N-dimensional space
  - Query the servers corresponding to the indices

13

Attribute-Value Trees

- Hierarchical naming structure
  - Similar to LDAP
- Attribute-value Tree (AVT)
  - Used to represent each entity
  - Links: attributes
  - Nodes: values
  - Path: a sequence of links and nodes from the root

14

DHT Conversion

- Hash each path of record's AVT to a key
  - Servers with corresponding hashes keep the record
- Query:
  - Produce an AVT of the query
  - Hash each path in the AVT
  - Contact the corresponding servers
- Optimizations:
  - Can filter out top parts of the tree
  - Only need to maintain specific parts of name space

15

SWORD

- Used for resource discovery
- Each attribute can take a range of values
  - E.g.: Find a machine s.t. CPU in [500-1000 MHz], RAM in [256-512 MB]
- Convert each (attribute, value) pair to key:
  - Partition key by attribute (n bits) and then by value (m bits)
  - k=h1(attr).h2(val)
  - h1, h2 hash functions
- Server corresponding to key k holds the record for the entity

16

Resolving Range Queries

- Servers partitioned by:
  - Attributes: Those with same h1
- Value ranges partitioned among servers
- Looking for attr a and value range [v1,v2]
  - Among servers hashing to "a":
  - Find subset hashing to [v1,v1'],[v1',v1''],…, […,v2]
- What if searching on multiple attribute-value ranges?
- How many servers need to be updated?

17