CSCI 5105

Instructor: Abhishek Chandra

Today

- Naming
  - Basics
  - Flat Naming

Naming

- Names used to refer to entities:
  - Files, hosts, processes, devices
- Name resolution:
  - Find the named entity (access it, find its location)
- Distributed naming:
  - Entities are distributed
  - Naming system itself may be distributed

Naming: Issues

- How are names specified?
- Where are names stored?
- How do we locate an entity given its name?

Types of Names

- Addresses
- Identifiers
- Human-friendly names

5

Addresses

- Name of an entity's access point
  - Provides a means to access the entity
  - E.g.: IP address, port no., etc.
- We could use an address as the name of an entity
  - E.g.: IP address for a machine, http server's TCP-tuple, etc.
  - Problems?

6

Location-Independence

- What happens to the name if:
  - The entity is moved?
  - The entity has multiple addresses?
- Location-independent name:
  - Does not depend on the address/location of the entity

7

Identifiers and Human-friendly Names

- Identifier: Name used to uniquely identify an entity
  - E.g.: inode, unique key
  - Often specified in machine-readable form
  - Context-dependent: IP address can be considered an identifier or an address
- Human-friendly names
  - Typically a sequence of characters
  - E.g.: host name, file name, URL

8

Naming Systems

- Provide name resolution: How do we resolve a name to its address?
- Flat Naming
  - Used for identifiers (names without any implicit structure)
- Structured Naming
  - Used for human friendly names (with structure)
- Attribute-based Naming
  - Used for descriptive names (that describe what the entities are)

9

Flat Naming

- Unstructured names
  - Any entity can have any name
  - All names are semantically equivalent
  - No information about location
  - Typically a sequence of bits. E.g.: IDs, keys

10

Flat Name Resolution

- Broadcasting and Multicasting
- Forwarding Pointers
- Distributed Hash Tables
- Hierarchical Approach

11

Broadcasting and Multicasting

- Two steps:
  - Broadcast the name
  - Named entity responds with address
- Example: ARP (Address Resolution Protocol)
  - Convert IP address to MAC address
- Issues?
- Could use multicast instead of broadcast for focused queries

12

Forwarding Pointers

- Leave a pointer to the new location if the entity moves
- Name resolution:
  - Get initial location of an entity
  - Follow chain of pointers
- Example: SSP chains for RMI with distributed objects
- Issues?

13

Home-based Approach

- Each entity has a home location
  - Keeps track of current address of entity
- Example: Mobile IP
  - Each host has a home IP address
  - Gets care-of address at new location
  - Clients connect to home IP address initially
  - Packets tunneled to C/O address
- Issues?

14

Distributed Hash Tables

- An identifier is a key in a hash space
- Each node is assigned a key
- Entities are also assigned keys from the same space
  - Mapped to "closest"-key nodes
  - E.g.: file placed on node with next highest key

15

Name Resolution

- Use Chord DHT system as example
- Given key k, find succ(k)
  - Succ(k): The node that holds k
  - E.g.: node with smallest id ≥ k
- Name resolution:
  - Each node holds pointer to its successor and predecessor
  - Forwards the key in the appropriate direction

16

Scalable Name Resolution

- Keep shortcuts to distant parts of the id-space
- Finger Table
  - If m-bit hash space, keep table of m entries
  - $FT_p[i] = succ(p+2^{i-1})$
  - i-th entry corresponds to a distance of at least 2^{i-1}
- Lookup k: Forward to entry $FT_p[j]$ s.t.
  - $FT_p[j] \leq k$ and
  - $FT_p[j+1] > k$

17

Handling Node Churn

- Node joining
  - Lookup successor
  - Announce itself to successor and predecessor
  - Initialize finger table entries accordingly
- Keeping finger tables updated
  - Periodically send keep-alive messages to successor and predecessor
  - Check that tables are consistent

18

Increasing Efficiency

- Topology-based key assignment
  - Incorporate network into the hash function
- Proximity neighbor selection
  - If multiple choice of neighbors on join, select the closest neighbor
- Proximity routing
  - Keep multiple choices for each finger table entry
  - Route to the closest node

19

Hierarchical Approach

- Entities assigned names or ids
- Network divided into a set of domains
  - Single top-level (root) domain
  - Each domain divided into sub-domains
  - Leaf domains: contain records for entities
- Directory node (DN): One for each domain
  - Tracks entities in the domain
- Location record (LR) for entity E:
  - Pointer to DN for sub-domain containing E
  - At leaf, contains actual address for E

20

Name Resolution

- Request for entity E starts at a leaf DN
  - Recursively forwarded up the tree until find a DN with a LR for E
  - Then recursively goes down the tree till find leaf DN containing E
- Insertion/deletion happen in a similar fashion
- Replication:
  - A DN may have multiple pointers

21

Scalability

- Domains are partitioned across a set of hosts
- Geographic scalability:
  - By assigning entities to local domains
- Size scalability:
  - By distributing higher-level domains, load balancing

22