CMPSCI 677 Operating Systems **Spring 2016**

Lecture 11: February 29

Lecturer: Prashant Shenoy *Scribe: Lurdh Pradeep Reddy Ambati*

11.1 Naming System

Names are used to share resources, uniquely identify entities and refer to features. An important issue with naming is that a name can be resolved to the entity it refers to. To resolve names, it is necessary to implement a naming system. Difference in naming in distributed and non-distributed systems is that naming in distributed is distributed itself. Examples of naming systems:

- Distributed Naming
- DNS
- LDAP

11.1.1 Two Approaches to the naming system:

- Hash-based(P2P: Chord, CAN)
- Hierarchical directory structure(DAG): Nodes are responsible for naming resolution. Examples include file names(each file name is a unique path in the DAG) and URLs.

11.2 Structured Naming

- Names are commonly organized into what is called a name space. Name space for structured names can be represented as a labeled, directed graph with two types of nodes- leaf nodes and directory nodes.
- A leaf node represents a named entity and has no outgoing edges.It stores information on the entity it is representing for example its address– so that a client can access it.
- A directory node has a number of outgoing edges each labeled with a name.

11.2.1 Name Resolution

- Name Space offer a convenient mechanism for storing and retrieving information about entities by means of names.
- Given a path name, name space can look up information stored in the node referred to by that name. This process of looking up a name is called name resolution.(Parse the path name and extract the components and then use the hierarchical structure for resolving each component) .
- If name resolution is done within a machine, it is a centralized approach.

11.2.2 Resolving file names across machines

- Name Resolution in case of distributed systems is similar to that of a centralized approach but different machines may be involved in resolving different parts of the name.
- Directory node in the foreign name space (i.e. a remote or different machine's name space) is called mounting point. Normally, the mounting point is the root of a name space.During the name resolution, the mounting point is looked up and resolution proceeds by accessing its directory table.
- If the client machine and the file server are both configured with Network File system and once the directory of file server is mounted on the client machine, client machine can access the files of that directory like local files. In NFS, remote files are accessed using local names(location independence).OS maintains a mount table with the mappings.

  Example : nfs://flits.cs.vu.nl//home/steem, this NFS URL names a file called /home/steem on an NFS file server flits.cs.vu.nl, which can be accessed by means of the NFS protocol.

11.2.3 Name Space Distribution

- Naming in large distributed systems: System may be global in scope (e.g., Internet, WWW). Name space is organized hierarchically.Name space is distributed and has three logical layers - global layer, administration layer and managerial layer.
- The global layer is formed by highest-level nodes(root node and other directory nodes close to it). Represents groups of organization and are characterized by their stability.
- The administration layer is managed by directory nodes that are together within a single organization. Represent group of entities that belong to the same organization. The nodes in these layer are relatively stable
- The managerial nodes consists of the nodes that change frequently.
- The more stable a layer, the longer are the lookups valid (and can be cached longer)

  Example: In the partitioning of the DNS name space, domain names like .com,.edu,.au etc come under global layers, then organization name like sun, yale etc come under administration layer.

11.3 Implementation of Name Space

- Iterative Resolution:In iterative naming system, each component/name is resolved one at a time, going from top level domain and descend down the hierarchy and resolve one at a time.

  Example: ftp://ftp.cs.vu.nl/pub/globe/index.html. In this resolution root server can resolve only the label nl, for which it will return the address of the associated name server. Client the passes the remaining path name to a name server, which resolves the label vu and returns the remaining name and this is passed to a name server and so on until until all the components are resolved.
- Recursive Resolution: Starting from the root, root server will resolve and passes the result to the next name server it finds and so on until all the components are resolved and results will be trickled in opposite direction like in recursion and root name server passes the result to client machine.

  Example: ftp://ftp.cs.vu.nl/pub/globe/index.html. In recursive resolution, root name server resolves only the label nl and passes the remaining path name to a name server, which resolves the label vu and so on until all the components are resolved and results trickle back up to the root name server which pass that to the client.

- Pros and Cons: In iterative resolution, caching can reduce overheads, as it eliminates few look-ups and can reduce burden on root name server. On the other hand, communication costs are quite high in case of iterative resolution compared to the recursive resolution

  In recursive resolution, caching results is more effective compared to the iterative name resolution. But, in recursive resolution burden on root name server is quite high, as every request goes through root server. Communication costs are often cheaper in case of recursive resolution comparatively.

11.4 X.500 Directory Service

- X.500 is a general naming service also referred to as the Directory Service. It generalizes the type of functions that DNS provides
- Directory Service requires us to do other kinds of services. For example: Yellow Pages - look for a plumber. The query might be show me listings of name of plumber. here we do not know the name of the plumber. We are looking for the name of the person with the attribute plumber
- Thus, X.500 provides special kind of naming service, it provides general lookup as well as lookups in a variety of domains.

11.4.1 Lightweight Directory Access Protocol

- X.500 is too complex for many applications. Hence we use a simplified variant of X.500 which is LDAP. For example, People finder on umass web page is running on LDAP service. LDAP can also be used for authentication of a user and to query the resources present in a system etc.
- LDAP is a service which is built on top of a database but specifies a schema.
- Practically, every commercial Operating System supports some kind of directory service which is a form of LDAP.
- Examples of LDAP implementations include
  - Active Directory for Windows OS
  - Novell Directory Services
  - iPlanet Directory Services
  - OpenLDAP for Unix as well as MAC OS
  - Typical uses of these services include storing user profiles, access privileges, resources present in a system etc

11.4.2 The LDAP Name Space

- The schema which LDAP uses may be used to store people's records or machine's records
- Attributes for the schema include Country, Locality, Organization etc. LDAP also keeps track of local machines like mail server, print server etc
- LDAP keeps track of the namespace in a heirarchial form

DNS is used for a specific mechanism on the web to do names to IP address translation. LDAP is useful for other applications within an organization such as to keep track of user profiles, their privileges, their credentials etc

11.5 Canonical Problems in Distributed Systems

These are the common set of problems that arise when building a distributed system:

- Time Ordering and Clock Synchronization
- Leader Election
- Mutual Exclusion
- Distributed Transactions
- Deadlock Detection

11.5.1 Clock Synchronization

- Assume that a machine has a system clock that tells you the current time. Every application can request the current time from the system clock. Time is unambiguous in centralized systems, so there is no synchronization problem in centralized systems.
- However in a distributed system each machine has its own clock. Note that clocks in machines are usually crystal oscillators. They are not very accurate. Thus, the asynchronization among the clocks will cause problems.
- Let's use "make" as an example. Make is an utility that allows you to incrementally compile every source file that was modified since the previous time it was run. The way it works is to compare the time stamps on the source files and the object files. If these files are located in different machines, the clocks on these machines may not be synchronized. Then if you save a source file on a machine with slower clock, its time stamp may be earlier than the object file, and make won't recompile that source file. Many problems like this occur because of lack of synchronization. There is no global clock that all time stamps can derive from. Today we will discuss algorithms that synchronize clocks. Note that all these algorithms have errors, but as long as the errors are small or negligible to the application, we can still use these algorithm.

11.5.2 Physical Clocks: A Primer

There are different kinds of clocks.

- The most accurate clocks are atomic oscillators. They have the accuracy of one part in $10^{1}3$. Most of the time the way you keep real time is to synchronize with a atomic clock. Most countries have a atomic clock, and broadcasts its time with radios or satellites.
- Most of the clocks are less accurate, example mechanical watches
  - Typically in machines, the clocks are crystal oscillators. There are crystals in these clocks.The clocks count the number of oscillations and ticks the clock after a certain number.
  - Thus, clocks might drift because crystal oscillations are not exact.
- What time really means depends on how astronomers tell time. They can use the rotation of earth, the moon, or the sun to tell time.

- There are multiple time zones on earth, and the zones coordinate with universal time(UTC), which is a international standard for time. So the best way to synchronize time is to synchronize your clock with an atomic clock, or some authoritative clock.