Introduction to Name Services

Objective:

- Define and explain name services in distributed systems.

In a distributed system, names are used for recognizing a wide variety of resources such as computers, services, remote objects and files on the network to users. The name service facilitates communication and resource sharing amongst users which require consistent names for resources. The most commonly used naming approaches are Domain Name Service (DNS), the Global Name Service (GNS) and X500. In general, any process that requires access to a specific resource must have a name or an identifier for it. The identifiers are used to refer to names that are interpreted only by programs. Remote object references and NFS file handles are examples of identifiers. There are two naming categories,

namely Pure names, and Non-pure names. The pure names are composed of uninterpreted bit patterns and looked up before they can be used, whereas non-pure names contain information about an object and its location. The addresses are required for accessing objects, but the objects can sometimes be relocated therefore, they are inadequate for identification sometimes.

The most common example of fetching the objects by their names is the DNS system which maps domain names to the attributes of a host computer such as IP address, the X500 directory service that can be used for mapping a person's name onto attributes including their e-mail addresses and telephone numbers. CORBAs Naming and Trading Service are used for mapping the name of a remote object and their properties by their attributes that describe an object in terms that are understandable by human users.

In the DNS system, a client makes a request to resolve the domain name of the server where initially an IP address is looked up to obtain a network address of the destination for executing the client's requests. Later, when a response comes

from the destination, the network address is resolved back to the IP address that is followed by names. The web browsers and e-mail clients make use of the DNS system to interpret the domain flames in URLs and e-mail addresses. These names are needed to refer to entities such as users, computers, and services themselves.

Uniform Resource Identifiers (URIs) are also needed along with the name to ide resources on the Web, and other Internet resources, such as electronic mailboxes where many individual types of resource identifiers are present in the system. Along with URI, the Uniform Resource Locator (URL) is often used for URIs that provides location related information about resources and specify the method for accessing them using the HTTP protocol. Uniform Resource Names (URNs) are URIs that are used as pure resource names rather than locators. For example, the URI:

mid:0F3EB325-5B02-HD9-B185-000A75B543C8@maill.yahoo.com

is a URN that identifies an e-mail containing it in its 'Message-Id' field. There are resolution services that are used for resolving URNs such as DOIs to resource attributes, but they are not used often.

Name Service

A name service stores data about an assortment of textual names, as binding between the names and the attributes of entities such as users, computers, services and objects. The collection is regularly subdivided into at least one naming context, i.e., individual subsets of the binding that are overseen as a unit. The significant activity that is supported by a name service is to determine an attribute by its name. The operations are additionally required for making new bindings, deleting bindings furthermore, listing bound names, and adding or erasing contexts. The name service management has two built-in properties called unification and integration, which are used for the identification of

resources that are managed by different services.

A few general requirements of name service are as follows:

- It should bind names to addresses in a single management domain.
- It must handle an arbitrary number of names and attributes to serve an arbitrary number of an organization.
- It must have a long life to store the names and components, that are used for implementing services.
- It must provide high availability as most other systems are dependent on it. These systems cannot work when the name service is broken.
- It must provide fault isolation for local failures that should not cause the entire service to fail.
- It should provide tolerance of mistrust in a large open system that cannot have any component that is trusted by other clients in the system.

A namespace is the assortment of every single valid name that is perceived by

specific service. The service will endeavor to look into a valid name, despite the fact that the name may prove not to correspond to any object, i.e., to be unbound. Namespaces require a syntactic definition to isolate valid names from invalid names. For instance, '...' is not acceptable as the DNS name of a computer, while www.yahoo.com is legitimate (despite the fact that it is unbound). Names may have an inner structure that represents their position in a hierarchic namespace such as pathnames in a file system, or in an organizational hierarchy such as Internet area names, or they may be chosen from a flat set of numeric or symbolic identifiers. One significant favorable position of a hierarchy is that it makes enormous namespaces that are more manageable.

The DNS namespace has a hierarchic structure, a domain name comprises of at least one string called name components or labels, isolated by the delimiter '.'. There is no delimiter that is present towards the start or end of an area name, despite the fact that the root of the DNS namespace is sometimes referred to as '.' for authoritative purposes. The name parts are invalid if strings do not contain '.'.

As a rule, a prefix of a name is an underlying segment of the name that contains at least zero or more components. For instance, in DNS www and www.yahoo, both are prefixes of www.yahoo.com. DNS names are not case-sensitive, so www.yahoo.com and WWW.YAH00.COM have a similar meaning.

An alias is a name that is characterized to indicate similar information as another name, like a symbolic link between file path names. It enables more convenient names substituted for relatively complicated ones. It enables alternative names that can be taken into use by different people for a similar substance like a short code http://bit.ly/ctsavk refers to open http://yahoo.com/program/add.java.

A naming domain is a namespace for which there exists a single overall administrative authority that is responsible for assigning names inside it. This authority is in general control of which names might be bound inside the domain, and it is allowed to designate these tasks. Domains in DNS are assortments of domain names; syntactically, a domain's name is the suffix of the domain names inside it. Some name services permit distinct namespaces, some of the time

heterogeneous namespaces to be installed into them; and some name services permit the namespace to be modified to suit the necessities of individual groups, users or even procedures.

The Distributed Computing Environment (DCE) namespace permits heterogeneous namespaces to be embedded inside it. DCE names may contain intersections which are similar to mount points in NFS and UNIX, except that they permit heterogeneous namespaces to be mounted.

A naming context either maps a given name onto a set of primitive attributes directly or maps it onto a further naming context and a determined name to be exhibited to that context. It is first introduced to some underlying naming context; resolution iterates as long as further contexts and derived names are outputs.

Any name service that stores an exceptionally large database and is utilized by a huge population will not store the entirety of its naming information on a single

server computer. Such a server may be a bottleneck and a critical point of failure. In this way, a name server may store data for more than one domain. It is commonly consistent with the state that information data is partitioned into servers according to its domain. The partitioning of data infers that the local name server cannot answer all enquiries without the assistance of other name servers.

The process of locating naming data from more than one name server that is used for resolving a name is called navigation. The client name resolution software carries out navigation for the benefit of a client. It communicates with name servers as important to resolve a name. It may be given as a library code and connected to clients.

In DNS and other name services, client name resolution software and servers keep up a cache of the results that are the aftereffects of the past name resolutions. At the point when a customer demands a name lookup, the name resolution software consults its cache. If it holds a recent result from a previous

lookup for the name, it returns it to the client; else, it starts discovering it from a server. The server may return data cached from different servers. Caching is a key to a name service's performance and helps with keeping up the accessibility of both the name service and other services regardless of name server crashes. Its role in enhancing response times by sparing communication with name servers is clear. It can be utilized to eliminate high-level name servers such as the root server, specifically from the navigation path that enables the resolution to continue in spite of any server failure.