# Resource Management in Distributed System with Different Approach

[1]Nalin Chaudhary, [2]Manoj Kumar
[1]Asstt. Professor, [2] Asstt. Professor
[1]Computer Science and Engineering, [2]Computer Science and Engineering,
[1]Bhagwant University, Ajmer, India

***Abstract:*** A distributed system is a collection of autonomous computers with the aim of enhancing resource sharing. In this paper we present an overview of a survey of resource management in distributed system with different approach. The cooperation among all agents achieves the management of the whole system and is the foundation for a fair and efficient resource mapping for all applications, regardless of whether the applications are executed in parallel or not. Whenever a client issues a request for a resource it is firstly checked onto itself and if the requested resource is not available there then it makes a request to its immediate server. If it also fails at its immediate server then with the help of information server the request is broadcasted in the network. If this request reaches a server who is possessing the resource then an acknowledgement will be passed to the server in the network which was responsible for generating the request. Our main aim is to have efficient resource discovery in a decentralised manner for the distributed architecture.

***Index Terms*** **- Resource Management, Agents, Client, Server, Collection.**

## I. INTRODUCTION

Every node consists of finite number of resources that are to be distributed among a number of competing processes occurring at that very node. Memory space, CPU cycles, files and I/O devices are the example of resources being used by any process going within a node[6]. As the number of resources available to a system are limited so we can make use of distributed architecture which allows remote sharing of the various resources. Thus a system present at one location can access the resources of a system present at another location provided both the systems are a part of distributed architecture. Distributed system is a collection of loosely coupled processors interconnected by a communication network where each processor is addressed as a machine or a node or a host or a computer. The main goal of a distributed computing system is to connect users and resources in a transparent, open, and scalable way. Ideally this arrangement is drastically more fault tolerant and more powerful than many combinations of stand-alone computer systems. "RESOURCE MANAGEMENT IN DISTRIBUTED SYSTEMS" is an application specifically designed to handle resource discovery efficiently in a decentralised way in a distributed system. We have implemented a main information server which maintains information about all the servers in the network because in order to be the part of network all the servers have to get themselves registered with the information server. Each server keeps a record of all the resources present among its clients. Whenever a client issues a request for a resource it is firstly checked onto itself and if the requested resource is not available there then it makes a request to its immediate server. If it also fails at its immediate server then with the help of information server the request is broadcasted in the network. If this request reaches a server who is possessing the resource then an acknowledgement will be passed to the server in the network which was responsible for generating the request[6].

A Client is a node or a simple part of the local network that can be part of the distributed system by connecting it to a server. This node is equipped with the privileges of adding, deleting, modifying a resource present with it. In order to access the resources of the network connected using distributed architecture the client makes request to its server which provides it with requested resource[6]. A Server is a node or a simple part of the local network that can be part of the distributed system by connecting it to a main information server. Thus it is a node that is particularly used to connect several clients (or nodes) together and is particularly use to deal with resource requests of its clients[6]. A Main Information Server is a node that is used to keep track of all the servers within the distributed architecture. Thus all the nodes which want to act as a server within the distributed architecture must be registered with this main information server. The main information server is used to route a resource request to all the servers in the distributed architecture if a particular server fails to handle the resource request of its client [6].

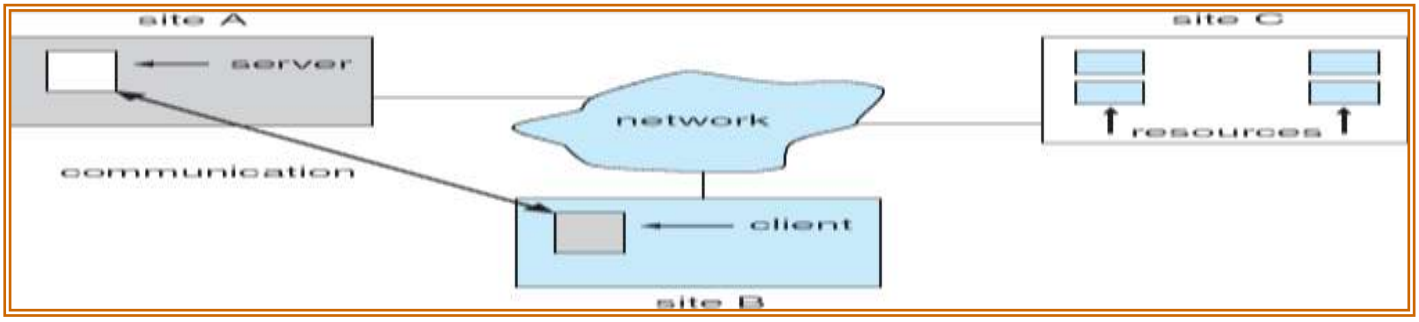


**Fig. 1.1** Distributed System Architecture

## II. Architectures Or Categories

Client-server, 3-tier architecture, N-tier architecture, Distributed objects, loose coupling, or tight coupling.

- **Client-server** — Smart client code contacts the server for data, then formats and displays it to the user. Input at the client is committed back to the server when it represents a permanent change[1].
- **3-tier architecture** — Three tier systems move the client intelligence to a middle tier so that stateless clients can be used. This simplifies application deployment. Most web applications are 3-Tier[1].
- **N-tier architecture** — N-Tier refers typically to web applications which further forward their requests to other enterprise services. This type of application is the one most responsible for the success of application servers[1].
- **Tightly coupled** (clustered) — refers typically to a cluster of machines that closely work together, running a shared process in parallel. The task is subdivided in parts that are made individually by each one and then put back together to make the final result[1].
- **Peer-to-peer** — an architecture where there is no special machine or machines that provide a service or manage the network resources. Instead all responsibilities are uniformly divided among all machines, known as peers. Peers can serve both as clients and servers[1].
- **Space based** — refers to an infrastructure that creates the illusion (virtualization) of one single address-space. Data are transparently replicated according to application needs. Decoupling in time, space and reference is achieved[1].
- **Grid approach** — A grid uses the resources of many separate computers, loosely connected by a network (usually the Internet), to solve large-scale computation problems. Public grids may use idle time on many thousands of computers throughout the world. Such arrangements permit handling of data that would otherwise require the power of expensive supercomputers or would have been impossible to analyze[1].

## III. Resource Management

Every node consists of finite number of resources that are to be distributed among a number of competing processes occurring at that very node. Memory space, CPU cycles, files and I/O devices are the example of resources being used by any process going within a node[6]. As the number of resources available to a system are limited so we can make use of distributed architecture which allows remote sharing of the various resources. Thus a system present at one location can access the resources of a system present at another location provided both the systems are a part of distributed architecture.

The sequence in which a resource is utilised by any process going within the system is listed as follow :-

### 3.1 Resource Request

Inorder that any process going within a system can access any resource of the system the very first phase for using of the resource is resource request. After making a request for the resource the process must wait till the time the resource can be allocated to the process. As here we are considering the distributed architecture so any process within the system can even make a request to the resource which is not present with that very system. Inorder to make request for a remote resource present within the distributed system the very first phase is resource discovery . The project undertaken by us is going to give an idea of how a resource can be discovered in a distributed environment in an efficient manner such that it leads to management of resources in a decentralised way[6].

### 3.2 Resource Use

Once a resource has been discovered the very next phase is the allocation of the resource to the requested processso that the requested process can make use of the resource according to its respective job[6].

### 3.3 Resource Release

After making use of the requested resource as per its needs the process must release the resource so thst it can be used by anyother process[6].

## IV. CLIENT - SERVER ARCHITECTURE

The client-server software architecture model distinguishes client systems from server systems, which communicate over a computer network. A client-server application is a distributed system comprising both client and server software. A client software process may initiate a communication session, while the server waits for requests from any client. Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server. Standard networked functions such as email exchange, web access and database access, are based on the client/server model. For example, a web browser is a client program at the user computer that may access information at any web server in the world. To check your bank account from your computer, a web browser client program in your computer forwards your request to a web server program at the bank. That program may in turn forward the request to its own database client program that sends a request to a database server at another bank computer to retrieve your account balance. The balance is returned to the bank database client, which in turn serves it back to the web browser client in your personal computer, which displays the information for you. The client/server model has become one of the central ideas of network computing. Most business applications being written today use the client/server model. So do the Internet's main application protocols, such as HTTP, SMTP, Telnet, DNS, etc. In marketing, the term has been used to distinguish distributed computing by smaller dispersed computers from the "monolithic" centralized computing of mainframe computers. But this distinction has largely disappeared as mainframes and their applications have also turned to the client/server model and become part of network computing. Each instance of the client software can send data requests to one or more connected *server*s. In turn, the servers can accept these requests, process them, and return the requested information to the client. Although this concept can be applied for a variety of reasons to many different kinds of applications, the architecture remains fundamentally the same. The most basic type of client-server architecture employs only two types of hosts: clients and servers. This type of architecture is sometimes referred to as *two-tier*. It allows devices to share files and resources. The two tier architecture means that the client acts as one tier and application in combination with server acts as another tier[6]. These days, clients are most often web browsers, although that has not always been the case. Servers typically include web servers, database servers and mail servers. Online gaming is usually client-server too.

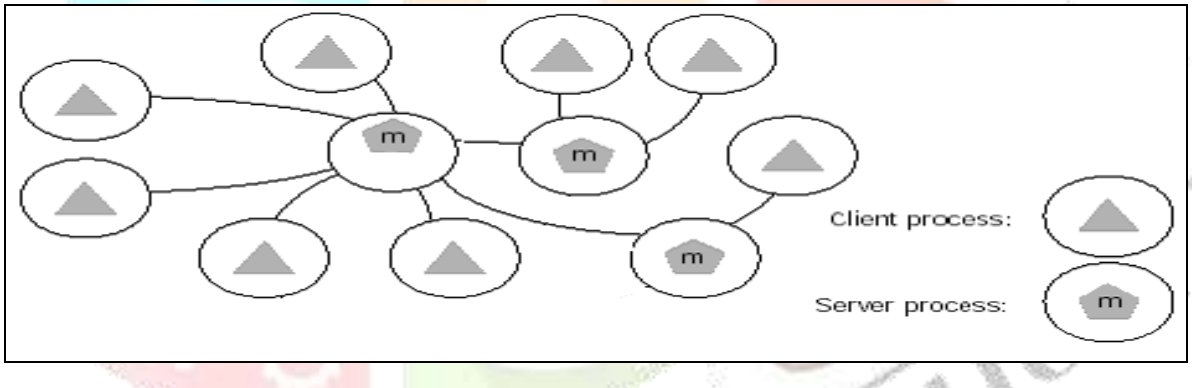


**Fig. 2.2 :** Two Tier Client – Server Architecture

The interaction between client and server is often described using sequence diagrams. Sequence diagrams are standardized in the Unified Modeling Language.

## V. BASIC APPROACH

In order to go for checking for the presence of a particular node in the distributed environment following approach is used :-

1. Whenever a client requires a resource it first of all makes a request. After making the request the client searches its own database for the presence of the resource and if the client is able to fulfill it request by itself only then it sets the availability status of the requested resource false and also communicates this change to its immediate server and quits the resource discovery process.
2. In case client fails to fulfills its request it sends its request to its immediate server and as we have considered multiple client-server architecture so server looks for the presence of the requested resource in its database which has all the enteries corresponding to the details of the various resources present at the various clients of the server and if the server is able to find the required resource and also if that resource has its availability status true then the server acknowledges the requested client with the presence of resource and once the resource is allocated to the requested client the server reports the false availability status of the used resource to the respective client of the resource and also updates its database.
3. In case server fails to find the requested resource in its network it forwards the request to the main information server which has record of all the servers in the entire distributed architecture and this server further floods the request to all the servers of the network and in case it receives a positive response from any of its server which search their own database as per step 2 then it acknowledges the requested server which further reports the resource presence to its respective client and once this resource has been allocated to the

requested client the falsity of the availability status is reported to the server & its respective client from which the request was fulfilled via the main information server. Figure 2.4 illustrates how the information server floods servers P1, P2, P3 and P4 with resource request.

4. In case step 3 also fails then the requested client is reported that the requested resource can't be allocated to the client because of its non – presence in the distributed network.

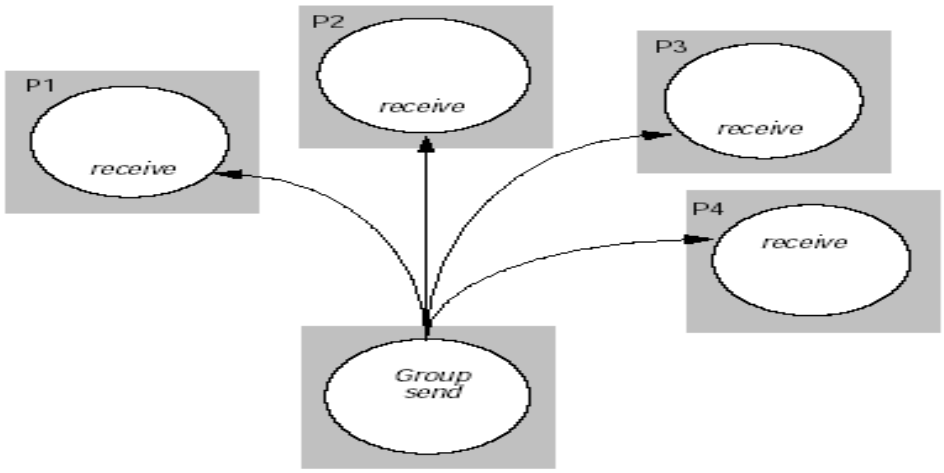


**Fig. 2.3 :** Flooding Approach

## VI. DISCUSSION

### 6.1 Server Registration (Or Deregistration) Window

This is the first page that opens whenever you want to register a server to main information server (Main Information Server runs at the back for registration). This screen prompts the user to enter the IP Address of server as well as main information server machine for the purpose of registration and deregistration.

### 6.2 Registration Of Client With Server

In order to be the part of the distributed network every client must be registered with any of the server which itself has been registered with main information server. This window prompts the user to enter the IP Address of its own machine and also IP Address of that machine to which it wants to act as Client so that a connection can be established between Client and Server.

### 6.3 Options Available To Client

This is the screen which opens when once the client has established connection with the server. This screen shows the various options available at client's disposal through which the client can communicate with the server. Initially client doesn't has any resources with it so its Resource List appears to be empty. Thus whenever client makes any changes with the resources available at its disposal then these changes can occur only if server permits for such changes.

### 6.4 Resource Addition At Client Site

This window prompts the client to add a resource at its site. In order to add a resource the client must add all the relevant data related to resource's detail because else a Dialog Box will appear and will prompt the client to fill in the necessary information. In order to add a resource at client site the resource alongwith its detail must be added at server site because server keeps the record of all the resources of its respective clients.

### 6.5 Resource Deletion At Client Site

This window prompts the client to delete a resource present at its site. The resource that the client wants to delete must have its availability status true because else a Dialog Box will appear and will prompt the client that it cannot delete an allocated resource site so for this the window also shows the resource list of the client alongwith the availability status of each resource. In order to delete a resource from client site the resource alongwith its detail must be deleted from server site because server keeps the record of all the resources of its respective clients.

### 6.6 Resource Modification At Client Site

This window prompts the client to modify a resource present at its site. The resource that the client wants to modify must have its availability status true because else a Dialog Box will appear and will prompt the client that it cannot modify an allocated resource site so for this the window also shows the resource list of the client alongwith the availability status of each resource. In order to modify a resource on client site the resource alongwith its detail must be modified at server site because server keeps the record of all the resources of its respective clients.

### 6.7 Resource Request From Client

This window prompts the client to request a resource present at its site or present at anyother site which is the part of distributed architecture. In case the requested resource is present with the client or with anyother node that is the part of the distributed network

and has its availability status true then resource will be allotted to the respective client else a Dialog box will appear which will show that resource can't be allocated right now!

## VII. CONCLUSION

The paper has a wide scope of enhancing its power of searching the servers for the presence of a resource. Thus instead of going for flooding via the main information server for a query of a Resource's presence any particular Routing Protocol can be implemented based on adaptability for network bandwidth available, network congestion, etc and this routing protocol approach can be compared with the flooding approach. Also trust and security could be added over this method that would imply authentication, integration and privacy. This paper also is a simulation of Resource Discovery in Distributed Systems in decentralised way. The project implements a 3-level architecture consisting of Main Information Server ( MIS) at the topmost level followed by servers at middle level which includes clients in their networks and these clients correspond to the lowest level.

The project introduces the concept of flooding in order to deal with Resource Discovery. Through this approach the main information server will flood all the servers of the network for the requested resource incase the requesting server fails to deal with the resource request in its own network and forwards its request to the main information server.

## REFERENCES

[1] Computer Networks – Andrew S Tanenbaum
[2] Core Java 2 Volume 2-Advanced Features – By Cay S. Horstmann, Gary Cornell Sun Microsystems
[3] The Complete Reference Java 2 – By Herbert Schildt, Tata McGraw-Hill
[4] www.sun.java.com
[5] http://java.sun.com/docs/books/tutorial
[6] An introduction to Database Systems – By C.J. Date
[7] ieeexplore.ieee.org/document/263828.
[8] mazsola.iit.uni-miskolc.hu/tempus/discom/doc/os/res_man
[9] cse.csusb.edu/tongyu/courses/cs660/notes/dresource.php
[10] https://www.slideshare.net/ASHI14march/resource-management
[11] coop.gforge.inria.fr/lib/exe/fetch.php/d2.2.