ISSN:2229-6093

RESOURCE MANAGEMENT IN DISTRIBUTED SYSTEM

Shalini Kapoor

CSE Deptt , GNI Mullana , Haryana

get_shalini@rediffmail.com

Poonam Kashyap

CSE Deptt , HITM Dheen , Haryana

p.23nov@gmail.com

***Abstract*: Resource Management in Distributed Environment is concerned with a system in which the main aim is to make sure that a user/client can access the remote resources with as much ease as it can access the local resources. The basis of Resource Management in the distributed system is also resource sharing. The main goal of this paper is to ease the sharing of resources among users. We have implemented a Peer to Peer system where files are our prime resources. A user while sitting on any node can access the files of any other peer provided he is an authenticated user. A graphical user interface is made available at each peer for registering itself to the required peer, for searching and if found, displaying the contents of required file. So it is secure than traditional distributed system.**

Index Terms : **Resource Management , Distributed Environment , Peer-to-Peer**

I. INTRODUCTION

Resource Management in Distributed Environment is a management system of resources like files other data over the distributed system whose main aim is to make sure that a user/client can access the remote resources with as much ease as it can access local resources. The basis of resource management is also resource sharing. Since a computer can request a service or file from another computer by sending an appropriate request to it over the communication network. Hardware and software resources can be shared among autonomous computers[1].This communication can also be referred to as peer-to-peer communication mechanism which is also the basis of distributed system rather than the centralized-server and client mechanism. The peer-to-peer communication mechanism is much more efficient, flexible, convenient and faster than the centralized-server and client's mechanism. In this architecture all the process involved in a task like resource management play similar roles, interacting co-operatively as peers without any distinction between client and server processes or the computers they run on. The aim of the peer-to-peer architecture is to exploit the resources in a large number of participating computers for the fulfilment of a given task. Organizing the interaction between each computer is of prime importance. In order to be able to use the widest possible range and types of computers, the protocol or communication channel should not contain or misuse that may not be misunderstood by certain machines. Special care must also be taken that messages are indeed delivered correctly and that invalid messages are rejected which would otherwise bring down the system and perhaps the rest of the network. Another important factor is the ability to send software to another computer in a portable way so that it may execute and interact with the existing network. This may not always be possible or practical when using different hardware and resources, in which case other methods must be used such as cross-compiling or manually porting this software.

II. ARCHITECTURE OF RESOURCE MANAGEMENT IN DISTRIBUTED ENVIRONMENT

In this architecture all the processes involved in a task like resource management play similar roles interacting co-operatively as peers without any distinction between client and server processes or the computers they run on [2]. The aim of peer to peer architecture is to exploit the resources in a large number of participating computers for the fulfilment of a given task. The following figure shows the Architecture of Resource Management in Distributed Environment.

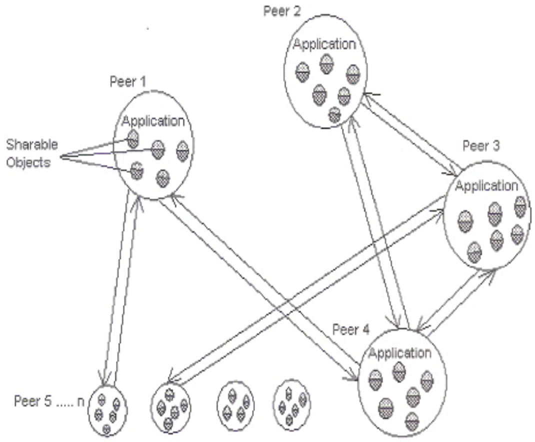


Fig 1 : Architecture of Resource Management in Distributed Environment

The figure shows the Resource Management overview in the form of a peer-to-peer application. Applications are composed of large numbers of peer processes running on separate computers and the pattern of communication between them depends entirely on application requirements. A large number of data objects like files are shared. Each object or file is replicated in several computers to further distribute the load and to provide compensation in the event of disconnection or network failure. The user interface for distributed resource environment needs several kinds of transparency for resources which are distributed on many systems connected to the network of the distributed system. Access transparency and location transparency are general concept for distributed resources. The advantage of the distributed environment is the replication of the resources in any site. Besides the two transparencies, there is semantics transparency in which a user can access resource by semantic name rather than by physical name for the resource.

Besides the distributed architecture of Resource Management, Scheduling is done to make resources available to an application whenever they are needed. Processes need to have resources assigned to them according to priority. A resource Scheduler determines the priority of processes based on certain criteria. Scheduling methods need to be applied to all resources that affect the performance of an application. There are two kinds of scheduling. Fair Scheduling and Real-Time Scheduling. The scheduling also depends upon the types of resources. A resource is any object that a process can request and wait for. A resource can consist of any number of identical units and a process can request any number of units of a resource. The types of resources can be given as:

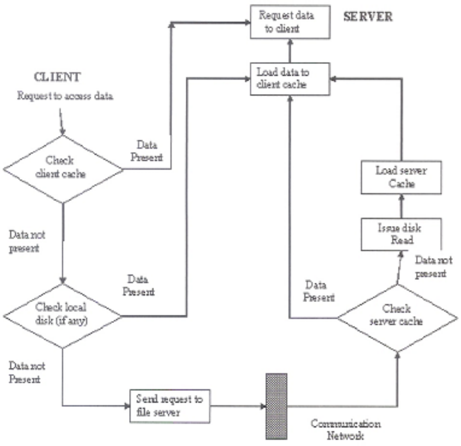
1) **Reusable Resources**: A reusable resource does not vanish as a result of its use but can be used over again and again. In a system, a reusable resource has a fixed number of units and these units can neither be created nor destroyed.
2) **Consumable Resources**: A consumable resource vanishes as a result of its use. When a unit of consumable resource is allocated to a process it as consumed and ceases to exist. There is no fixed number of units of a consumable resource in a system since the resource units can be created and consumed.

In a distributed resource management system, resources can be stored at any machine and the computation can be performed at any machine. When a machine needs to access a file stored on remote machine, the remote machine performs the necessary file access operations and returns the data if a read operation is performed.

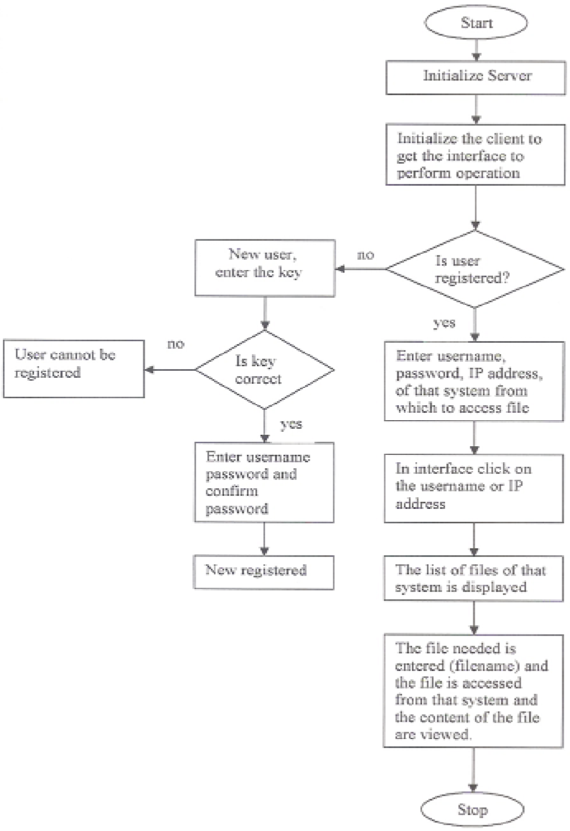
The two most important services present in a distributed resource management system are the name server and cache manager. A name server is a process that maps names specified by clients to stored objects such as files and directories. A cache manager is a process that implements file caching. In file caching, a copy of data stored at file server is brought to the client’s machine when referenced by the client. Subsequent accesses to data are performed locally at the client, there by reducing the access delays due to network latency. Cache managers can be present at both clients and file servers. Cache managers at the server caches files in the main memory to reduce the delays due to disk latency. Typically, accessing remote resources is more expensive than accessing local resources because of the communication delay that occur in the network and the CPU overhead incurred to process communication protocols. The impetus behind the development of Distributed Computing was the availability of powerful processors at low cost and also advances in communication technology. The availability of powerful yet cheap processors led to the development of powerful workstations that satisfy a single user's needs. These powerful stand-alone workstations satisfy user need by providing such things as bit-mapped displays and visual interfaces, which traditional time-sharing systems do not support. When a group of people work together, there is generally a need to communicate with each other, to share data, and to share expensive resources such as printers, disk drives etc. This requires interconnecting computers and resources. Technically, a completely pure peer-to-peer application must implement only peering protocols that do not recognize the concepts of "server" and "client". Such pure peer applications are rare. Most networks and application described as peer-to-peer actually contain or rely on some non-peer.

III. PROPOSED SOLUTION

The proposed solution for **data access in a distributed resource management system** proceeds as shown in flowchart below:



A request by a process to access a data block is presented to the local cache (client cache) of the machine (client) on which the process is running. If the block is not in the cache, then the local disk, if present, is checked for the presence of the data block. If the block is present, then the request is satisfied and the block is loaded into the client cache. If the block is not stored locally, then the request is passed onto the appropriate file server. The server checks its own cache, for the presence of the data block before issuing a disk input/output request. The data block is transferred to the client cache in any case and loaded to the server cache if it was missing in the server cache. We possess a good understanding of the basic mechanisms required for a provisioned distributed system. Significant challenges remain; however, in understanding how these mechanisms can be effectively combined to create seamless virtualized views of underlying resources and services. Some of these challenges lie strictly within the domain of resource management, for example, robust distributed algorithms for negotiating simultaneous service level agreements across a set of resources. Other issues, such as expression of resource policy for purposes of discovery and enhanced security models that support flexible delegation of resource management to intermediate brokers are closely tied to advances in distributed infrastructure.


Start
Initialize Server
Initialize the client to get the interface to perform operation
Is user registered?
no
New user, enter the key
yes
no
Is key correct
User cannot be registered
yes
Enter username, password, IP address, of that system from which to access file
Enter username password and confirm password
In interface click on the username or IP address
New registered
The list of files of that system is displayed
The file needed is entered (filename) and the file is accessed from that system and the content of the file are viewed.
Stop

(Fig 2. **Functioning of the resource management system**)

Hence, the key to progress in the coming years is to create an extensible and open infrastructure that can incorporate these advances as they become available.

IV. MODULES IN THE SYSTEM

While implementing **Resource Management in Distributed Environment**, we have constructed five modules:

- Server Module
- Server Thread Module
- Client Module
- User Module
- Start Module

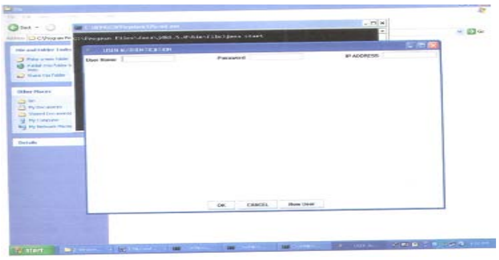
These modules are implemented using different features of java. The web browser, and many of the other applications that sprung up during the early commercialization of the internet, were based around a simple client/server protocol; the client initiates a connection to a well known server, downloads some data, and disconnects.

The description of each module can be given as:

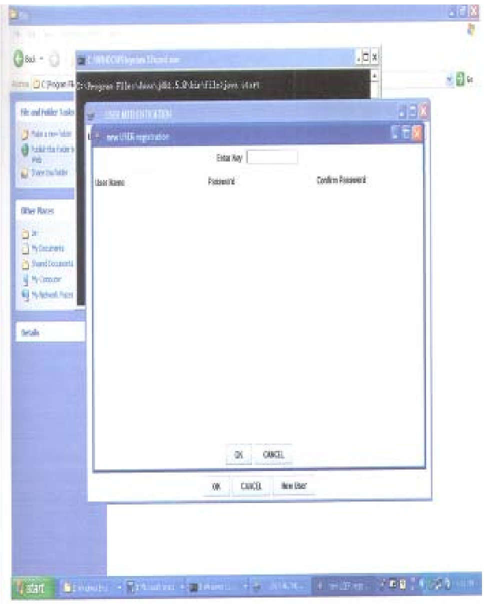
- **Server Module** – The server can also be called as file servers. The server is basically storage of resources and it provides services to the authenticated clients. They are high performance machines, dedicated to storing files and performing storage and retrieval operations. The rest of the machines in the system are meant for computational purpose only.
- **Server Thread Module** – This module is also for the server site. Here the concept of multithreading is used. Because in peer-to-peer distributed environment the operations or the tasks are performed simultaneously on different machines so for that multithreading is very important.
- **Client Module** – The clients are the machines which access the data or the resources from the other computers or servers. The clients are meant for the computational purposes and not for the storage purposes, they simply access files or resources from servers. Some client machines may also be equipped with a local disk storage that can be used for caching remote files, as a swap area, or as a storage area.
- **User Module** - The user module is constructed for the registration of new user or client to be registered. This module is important because new users should undergo the authentication and authorization process. There is such mechanism, which is implemented in our project.
- **Start Module** – The start module is constructed for to initialize the application, so that we can see an interface on the screen, through which only the authenticated and authorized users can login and can access the remote resources from the remote computers.

The term distributed environment is used, because we are working in distributed environment or on distributed systems which is a collection of loosely coupled processes i.e. autonomous computers connected by a communication network with the aim of enhancing resource sharing. The distributed environment comprises distributed computing which involves dividing a large application into many small components which may be running on different machines/platforms. Resource Sharing has the ability to use any hardware, software or data anywhere in the system. Resource Manager controls accesses, provides naming scheme and controls concurrency. Typically, accessing remote resources is more expensive than accessing local resources because of the communication delays.

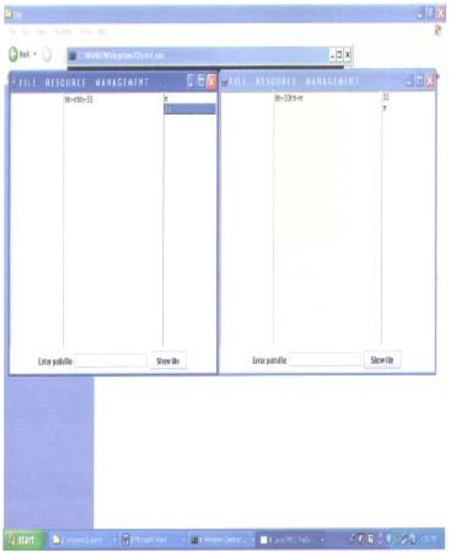
V. OUTPUTS OF THE SYSTEM



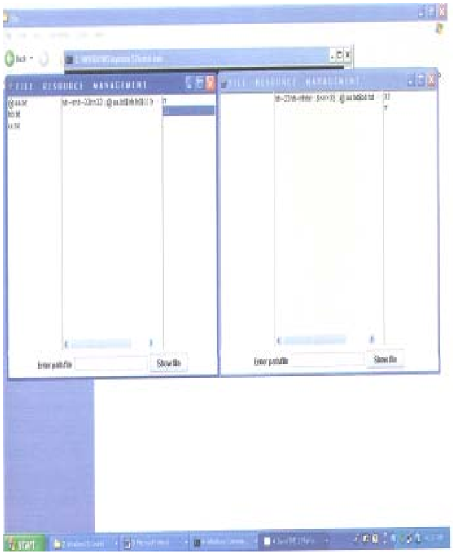
1. User Authentication



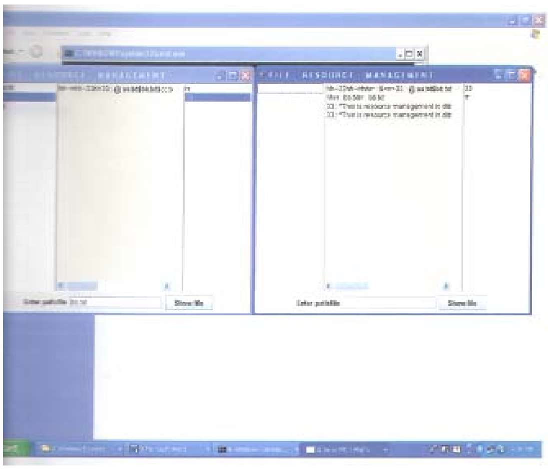
2. New User Registration



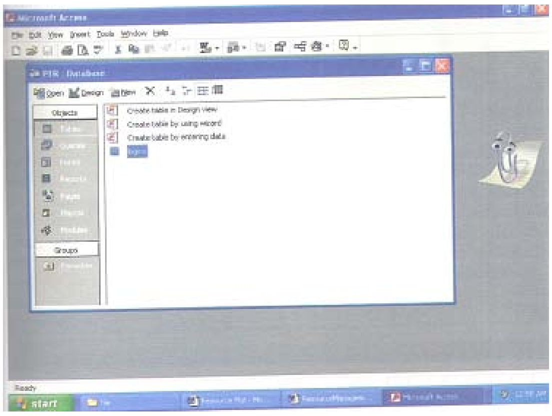
3. Resource Management System Interface after user logs in



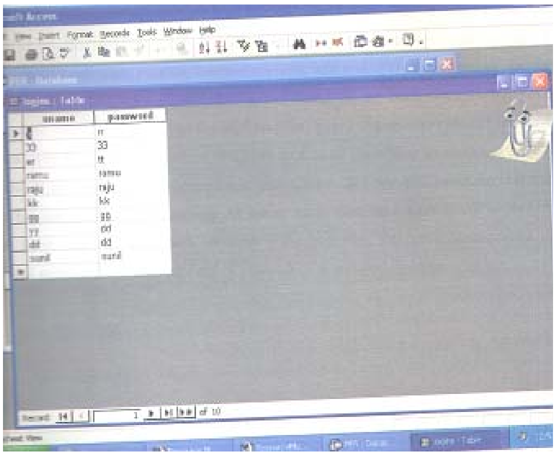
4. Shows the list of files accessed from remote computer



5. Shows the contents of the file accessed



6. Shows the MS-Access Database named - PER



uname	password
[illegible]	rr
33	33
et	tt
ramu	ramu
raju	raju
kk	kk
gg	gg
yy	dd
dd	dd
sunil	sunil

7. Shows the Logins Table in MS-Access

VI. CONCLUSION

The conclusion we draw from this paper- Resource Management In Distributed Environment is that to make accessibility of data easy and manage resources over the network of autonomous computers (i.e. Distributed System) we have to maintain such a system which is efficient, flexible, portable and secure over the network, so that every authorized user over the network is served better. So we devised such a Resource Management System which serves the said purpose. Every organization should maintain such Resource Management System.

VII. REFERENCES

[1]” Ada Reference Manual”, ISO/IEC 8652:2005(E) Ed. 3, Annex E Distributed Systems.

[2]”Distributed System Principles”, @Wolfgang Emmerich,1997,http://www.cs.ucl.a.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf

[3]Andy Oram et al.,”Peer-to-Peer Harnessing the power of Disruptive Technologies”, Oreilly 2001

[4]Ralf Steinmetz, Klaus Wehrle (Eds). “Peer-to-Peer Systems and Applications”, ISBN: 3-540-29192-X, Lecture Notes in Computer Science, Volume3485, Sep 2005

[5]Ghosemajumder, Shuman.”Advanced Peer-Based Technology” Business Models.MIT Sloan School of Management, 2002.

[6]Distributed Systems and Recent Innovations: Challenges and Benefits.

VIII. AUTHORS PROFILE

Shalini Kapoor received the bachelor degree in Computer Science and Engineering from Haryana Engineering College, Jagadhri, India in 2003. She received her Master degree in Information Technology from Karnataka State University, Mysore, India in 2011. She has 2.5 years industrial experience and 4.5 years teaching experience. Presently she is working in Computer Science and Engineering Department of Guru Nanak Institutions Mullana.

Poonam Kashyap received the Masters degree in Computer Science and Engineering from M.M Engineering College,Mullana(Kurukshetra University), India in 2007. She has 1 year industrial experience and 1.6 years teaching experience. Presently she is working in Computer Science and Engineering Department of HITM, Dheen Ambala.