Applications Sharing using Binding Server for Distributed Environment

Kavita Kimmatkar[1], Urmila Shrawankar[2]

[1] G.H. Raisoni College of Engineering, Nagpur, Maharashtra, India

[2] G.H. Raisoni College of Engineering, Nagpur, Maharashtra, India

Abstract

Today's environment is a fast growing environment and data is distributed for the expansion of computation. Distributed computing has a major issue such as how to share a data, how to locate the services and how to make it scalable. When a limited number of nodes are available and load of running application is more as well as overhead are also major then sharing of applications among the node will definitely improve the performance of system. Locating the registered nodes to the binding server is located by making remote procedure call. Using appropriate load balancing technique threshold value will be calculated and sharing of the application on node are performed and utilization of processor as well as resources can be balanced accordingly.

Keywords: *remote procedure call, binding server, distributed computing.*

1. Introduction

A major challenge in distributed environments is how to share a data efficiently. Now a day's data become more and more distributed and it is difficult to manage that data [1]. In proposed system application is shared by using remote procedure call and binding server. Remote procedure call is a protocol which is used for communication in distributed environment. In proposed system Remote procedure call can work same as its basic working shown in Fig.1. It provides a facility to communicate between client and server [2].Here we desire to develop a binding server who keeps a track of all record about application which is mounting on different node.

When any node requires particular application it communicates with binding server through RPC. It makes a subroutine call to the bind server and bind server provide that node to other node who consist that application and maintain a communication between these two node. After the execution of application calling node retrieve the result of subroutine.

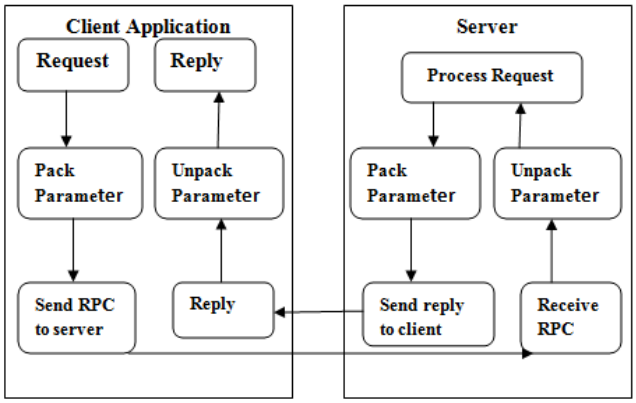

Fig.1 Client Server RPC Procedure

For a client to bind a server there are multiple mechanisms, the simplest one is to use a hard coded server name, but this method is inflexible, because if any change occurred on the server name, version or the location, the client code need to be upgraded, this problem was solved by using the directory service[3],[4],[5].

This paper is organized as follows. We compare a basic protocols used for remote procedure call, different load balancing algorithm which is used for distributed environment and scheduling policies in section II. Our proposed work is explained in section III, and last conclusions are given in section IV.

2. Existing Approaches

2.1 Comparison of protocol

Depending on different system several communication protocol are present for remote procedure call such as Request protocol (R protocol), Request/Reply Protocol (RR Protocol) and Request/Reply/Acknowledgement Protocol (RRA Protocol) [8], [9].

2.1.1 Request (R) Protocol

Request protocols are used when caller party does not required any confirmation about data receive or not, procedure executed or not. Client sends a request message for procedure execution to server. Sever execute that procedure but does not send in reply message.

2.1.2 Request/ Reply (RR) Protocol

Request/Reply protocols are used when caller party require confirmation about data and result of procedure execution. Client sends a request message for procedure execution to server. Sever execute that procedure and send Reply message to the client. This protocol solves the limitation of R protocol as the reply is send to client about execution of procedure.

2.1.3 Request/Reply/Acknowledgement(RRA) Protocol

Request/Reply/ Acknowledgement protocols are used when caller party requires confirmation about data and result of procedure execution and client also send the acknowledgement. Client sends a request message for procedure execution to server. Sever execute that procedure and send Reply message to the client, client send the acknowledgement message to server that they received the result. This protocol solves the limitation of R protocol as well as RR protocol. Because the reply is send to client about execution of procedure and client also send the acknowledgement about result. Comparison of these protocols is shown in Table 1.

Table 1: Comparison of RPC Protocol

Parameter	R protocol	RR Protocol	RRA Protocol
Stability	Less	Less	More
Failure Report	No	Yes	Yes
Acknowledgement	No	No	Yes
Efficiency	Less	More	More

Upper layer protocol such as SNMP, IP, FTP, SCTP use IP address directly during communication. Instead of IP address we wish to use MAC Address. For MAC Address we have to use data link layer protocol. Data link layer does not consist any protocol but it supports two protocols.

- DACP (Data Link Switching Client Access Protocol)
- ARP (Address Resolution Protocol)

A. DCAP (Data Link Switching Client Access Protocol)

DCAP is used between workstations and routers for transport SNA traffic over TCP sessions. In a wireless network, each workstation that are connected in a network needs a MAC address to communicate with a FEP. When Data link switching client access protocol is implemented on a workstation, it does not contain MAC address, it only consists of an IP address [8]. If DCAP is implemented on large number of workstation it raises the important issues of scalability and efficiency during communication. As DCAP is a switch-to-switch protocol, it is not efficient when it is implemented on workstations and DCAP protocol has scalability and efficiency problem and it mainly act in logical link layer which consist of data, logical address and control information. MAC address require for communication so ARP Protocol is use in project [9].

B. ARP (Address Resolution Protocol)

ARP Protocol supports data link layer but it's not a data link layer protocol. Address Resolution Protocol (ARP) required a TCP/IP standard defined in RFC 826, ARP resolves IP addresses used by TCP/IP-based software to media access control addresses. ARP has a facility of dynamic mapping, and it dynamically map logical address i.e IP address to Physical address i.e MAC address [8].

2.2 Load Balancing algorithm

Sharing of data required that it can be shared in reliable way, so for this we required a load balancing algorithm. Dynamic load balancing consists of maintaining load equilibrium among all the processors of a system during the execution of an application. For this purpose, a LBA needs to determine the load index that represents the processor load. In general, a LBA considers two policies related with the decision of load transfer and the gathering of load states information. The control policy is responsible for defining when and where to make a load transfer and which processor can make such a decision, only one processor or all of them. Concerning the information policy, it determines how the information will be collected, in a global or partial way. When a global information is maintained, the load index of all processors are sent to one processor, so when a large number of processor are used, possible bottlenecks can occur producing serious limitations on the LBA scalability. An adequate solution to improve scalability is by defining partial information policies where each node only needs the load indexes of some processors [10],[11]. Different load balancing algorithms are existing for distributed system such as Round robin, Random, Local queue, central queue and central manager. Comparison of these load balancing algorithms are given in Table 2.Round Robin and Random algorithm are used decentralized distribution and it also work in static manner. Local queue algorithm and Central queue algorithm can work in dynamic manner but local queue used decentralized distribution [10].

Table 2: Comparison of load balancing Algorithm

Parameter	Round Robin	Random	Local Queue	Central queue	Central Manager
Stability	Large	Large	Small	Small	More
Centralized/ Decentralized	Decentralized	Decentralized	Decentralized	Centralized	Centralized
Dynamic /Static	Static	Static	Dynamic	Dynamic	Static

On the basis of above study of algorithm and comparison on different parameter, we select Central Queue algorithm as it has a centralized control and it works on dynamic distribution for proposed work.

2.3 Scheduling Policies

A major challenge in wireless networks is the ability to maximize the throughput. High throughput in a wireless network requires a relatively low complex scheduling policy with a provable efficiency ratio, which is a measure of the performance of the policy in terms of throughput and stability [13]. As we share an application in our proposed system to share a that application load balancing and scheduling is mostly required. Load balancing is already explained in section 2.2. For Operating system different scheduling algorithm had developed some common scheduling algorithms are primitive algorithm, Non primitive algorithm, round robin algorithm, shortest job first algorithm. Author Gustavo Vejarano, Janise McNair explained a new scheduling policy for wireless network which provide stability, Reservation-Based Distributed Scheduling (RBDS). In an RBDS wireless network, the nodes negotiate with their neighbors for the reservation of future data-time-slots of their links. This negotiation is based on a three-way handshake that consists of a request, a grant, and a grant confirmation. Requests, grants, and grant confirmations are transmitted in scheduling packets. The nodes access the control-time-slots for transmitting scheduling packets using an election algorithm. Therefore, in an RBDS wireless network, the nodes access the wireless channel using two different algorithms: the election algorithm and the RBDS algorithm, whose roles are to avoid collisions and waited time slots in the control and data-sub frames respectively.

3. Proposed Work

The proposed system design a remote procedure call network and binding server for communication. The basic idea behind a project is explain in Fig.2.

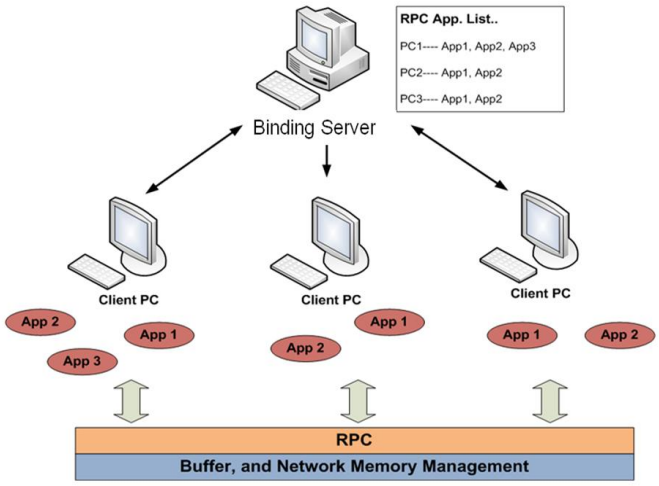

Fig 2. Basic Working idea

As bind server support New Technology File System and have its facility in OS. But in the network it is difficult to know what data shared by whom.

We will use a concept of NTFS in proposed system If NFS server is configured and have some shared document then to use that share one must know what is the location or path of that share and address of the server, after this user has to mount that share at some mount point in the machine. This is very lengthy in the network [14].

To solve this problem we are trying to create one auto server and client. On server side if any machine in the network starts server will detect that is there any share is provided by machine for other and make entry about its share in its database of network shares in the network. On client side client try to get share of other machine that already started from the server and also try to mount that share on local machine for temporary so that it can used by particular user.

In the project first we store all the share detail in the network on the server and then provide that detail to all clients. We will use a basic RPC protocol that is Request Reply Acknowledgement protocol (RRA Protocol) and ARP protocol.

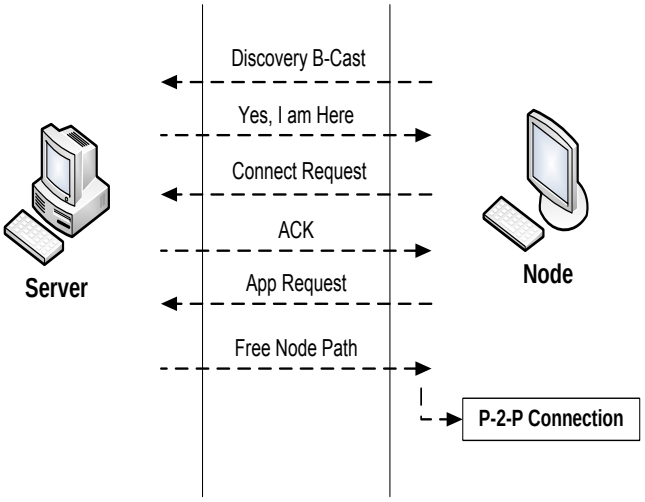

Fig 3. Working of RRA Protocol

In proposed system, client node registers their services with binding server. For register they follow below steps:-

i. Start: - Binding server as well as all the other laptop start for connection.
ii. Request to connect: - All the laptop sends the request to binding server for connection.
iii. Accept Request:-Binding server accepts that request.
iv. Send Application List to share:-after the establishment of connection all the connected laptop send their application list to binding server.
v. Add Library to public List: - Binding server adds them to their public list.

When computer/laptop that are connected in a network require any application they request to binding server. Binding server can connect that computer to the other computer who consists of that application. In proposed system client pc consist different application and that can be share through binding server. If same application is present on more than one computer at that time binding server analyze both the computer on the basis of their working load.

4. Conclusion

As application and other resources becoming more distributed for enhancing the power of computation and utilization of costly resources, it requires a proper and optimized resource sharing and accessing. This paper will proposed system that share costly resources and application When data is shared in distributed environment it has scalability and efficiency problem in proposed system applications are available among all node which provide a improved computation power.

References

[1] Hao Chen, Lin Shi, Jianhua Sun, Kenli Li, Ligang He,"A Fast RPC system for Virtual Machines", IEEE transactions on parallel and distributed systems 1045-9219/12, IEEE 2012.
[2] Fawaz A. M. Masoud, Qatawneh Mohammad, Wesam Almobaideen, "Interactive RPC Binding Model", European Journal of Scientific Research ISSN 1450-216X Vol.27 No.1, pp.112-119,2009.
[3] Andrew D. Birrell,Bruce Jay Nelson "Implementing Remote Procedure Calls", 0734 2071/84/0200-0039,ACM Transactions on Computer Systems, Vol. 2, No. 1, Pages 39-59, 1984.
[4] Yue-zhuo Zhang , Yong-zhong Huang and Xin Chen, "A Procedure Search Mechanism in OGSA-Based GridRPC Systems" ,Springer Berlin / Heidelberg, pages 400-403, Session 7: Network Communication and Information Retrieval, 2004
[5] Yao Wang,Yongquan Yu "WMIC and the RPC in the enterprise application Integration", 978-1-4244-7874-3/10/2010 IEEE journels,2010.
[6] Li Qilin,Zhou Mintian "The State of the Art in Middleware", 2010 International Forum on Information Technology and Applications 978-0-7695-4115, IEEE,2010
[7] Brian N. Bershad, Thomas N. Anderson, Edward D.,Lazowska,Henry M. Levy, "Lightweight Remote Procedure Call', ACM 0734-2071/90/0200-0037, ACM Transactions on Computer System.2011 s, Vol. 8, No. 1, Pages 37-55, February 1990.
[8] Behrouz A. Forouzan "TCP/IP Protocol suite" (TMH Publication) IIIrd addition.,2005
[9] Kamrul Hasan Talukder and Koichi Harada "Modeling and Verification of Some Communication Protocols" ,ISBN 89-5519-129-4 - 2193 - Feb. 20-22, 2006 IC ACT200G,2006.
[10] S.Sharma, S.Singh, and M.Sharma, "Performance analysis of load balancing algorithms",World Academy of Science, Engineering and Technology,2008.
[11] Juan Santana-Santana, Miguel Alfonso Castro-Garc´ıa, Manuel Aguilar-Cornejo and Graciela Rom´an-Alonso "Load Balancing Algorithms with Partial Information Management for the DLML Library", 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing 1066-6192/10 IEEE,2010.
[12] Rashmi Sharma, Nitin,"Optimal Method For Migration Of Tasks With Duplication", 2012 14th International Conference on Modelling and Simulation, 978-0-7695-4682-7/12, IEEE,2012.
[13] Gustavo Vejarano, Member, IEEE, and Janise McNair, "Stability Analysis of Reservation-Based

Scheduling Policies in Wireless Networks", IEEE Transactions on parallel and distributed systems, vol. 23, no. 4, April 2012.

[14] Harrison Chandler, Haiying Shen,Lianyu Zhao, Jared Stokes, Jin Li, "Toward P2P-Based Multimedia Sharing in User Generated Contents", IEEE Transactions on parallel and distributed systems, vol. 23, no. 5, may 2012.