

## **Data Consistency Problem in Distributed Environment**

*Anil Kumar Soni*

[\*anilsoni057@gmail.com\*](mailto:anilsoni057@gmail.com)

*Reg No: 11111310*

*Adesh Mishra*

[\*adeshmishra1790@gmail.com\*](mailto:adeshmishra1790@gmail.com)

*Reg No: 11111427*

*Ajay Kumar*

[\*ajay6366@gmail.com\*](mailto:ajay6366@gmail.com)

*Reg No: 11106366*

*School of Computer Application*

*Lovely Professional University*

*Phagwara Punjab*

### **Abstract:**

In this paper we discuss about the Distributed environment. This paper consists of what is distributed system, why we use distributed system, and what are the benefit of this system. The distributed system has two scenarios first one is multiprocessor and second is multi computing. It also contains the tautly coupled multiprocessor and loosely coupled multiprocessor. So in loosely coupled multiprocessor the major problem is data consistency in memory. Now the second scenarios is multi computing this contain the multiple computer communicate with each other for performing the some common goal. The multi computing used for improve the execution performances of the system. In the multi computing distribution system the

major problem is the data consistency. So that main idea of this paper is how to reduce the data consistency problem in the distributed environment. Now till the current times many researchers work on this problem but not find any technique or algorithm to complete reduce the data consistency problem. So in this paper we discuss the alternative solution to how remove the data consistency problem in the distributed environment.

### **Keywords:**

Distribute system, multiprocessor, cache coherence problem, multi computing, share data space, data consistency problem.

## 1. Introduction:

Distributed operating system consist the many computer on the network level. The distributed computing is contains the multiple system to communicate with each other through network. The many computers interact with each other to specify the common goal. The main issues of the distributed system are to solve the computational problem. It is also used for sharing the resources and coordination of distributed activities in networked environments.

The use of concurrent processes that communicate by message-passing has its roots in operating system architectures studied in the 1960s. [1]

The study of distributed computing became its own branch of computer science in the late 1970s and early 1980s. The first conference in the field, Symposium on Principles of Distributed Computing (PODC), dates back to 1982, and its European counterpart International Symposium on Distributed Computing (DISC) was first held in 1985.

Now the distributed system used for the decompose the single task in multiple tasks and perform the execution of each task by using the one or more computer. By use of distribution system it increases the performance of the execution.

The distributed environment contains the two scenarios:

1. Multi processing
2. Multi computing

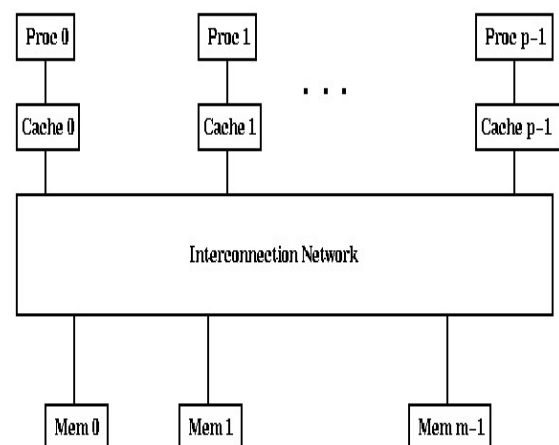
Multi processing is the multiple processes execute the some specific data.

These are the two types, first is tautly coupled multi processes in which each processes contains the its own special data

for the execution but the second is the loosely coupled multi processor in which all the processes execute the some data that means all the processes work on the share data space.

Then in the loosely coupled multi processors the problem is occur which is called the data consistency problem.

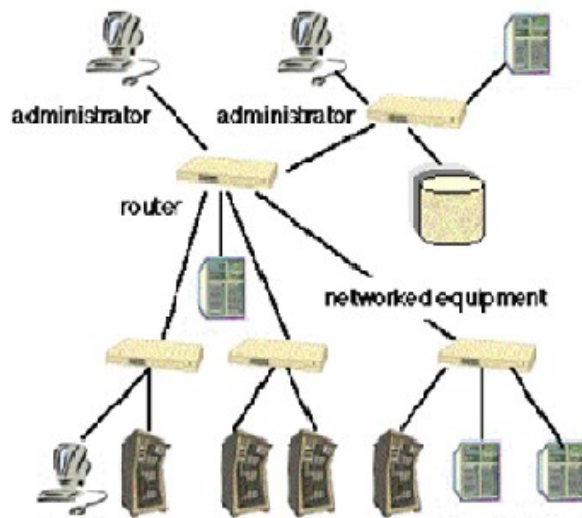
Now the data consistency problem is that all processor have its own cache memory then if the any cache update or modify the own cache data then the another processor not know the information of the data is updated so this problem called the cache coherence problem or the data consistency problem in the multi processor.



Cache coherence arises because multiple processors may be reading and modifying the same memory block within their own cache. Multiple copies of the same data can exist in different caches simultaneously and if processors are allowed to update their own copies freely then the cache coherence problem occur.

Now the second scenario is the multi computing in this scenario many computers interact with each other through network. Sharing of resources and coordination of distributed activities in networked

environments are the main goals in the design of a distributed operating system. Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, [2] which communicate with each other by message passing. [3]



Now this environment the data consistency problem occur because any of the computer update the data in the network level then how another computer gets information to the data is updated. So that many technique available these day to solve the problem by the using of the concept of message passing but the if the we want to implement the message passing technique then we required the mediator which is called the broker. The mediator is nothing but the it is the software which acts the intermediate between the computer and network. Now the main concerns of this paper is the remove the mediator and implement the new technique which is helps to reduce the data consistency problem in the distributed environment.

## 2. Literature Survey:

Distributed computing is the process of aggregating the power of several computing

entities to collaboratively run a computational task in a transparent and coherent way, so that it appears as a single, centralized system.

A distributed computer system is a loosely coupled collection of autonomous computers connected by a network using system software to produce a single integrated computing environment. [4]

In the multi processor concept the **Peter Stenström** definitions: memory coherence: a read shall return the value of the latest write as defined by the partial order of memory operation in a valid execution.” [5]

The word distributed in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. [6]

Other typical properties of distributed systems include the following:

The system has to tolerate failures in individual computers. [7]

The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program. [8]

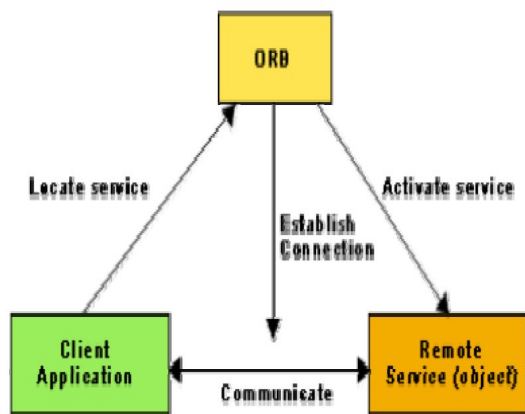
Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input. [9]

In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

A distributed system can be more reliable than a non-distributed system, as there is no single point of failure. Moreover, a distributed system may be easier to expand

and manage than a monolithic uniprocessor system. [10]

Over the years, technologies such as CORBA and DCOM have provided the means to build distributed component-based systems. Such technologies allow systems to Inter operate at the component level, by providing a software layer and protocols that offers the interoperability needed for components developed in different programming languages to exchange messages.



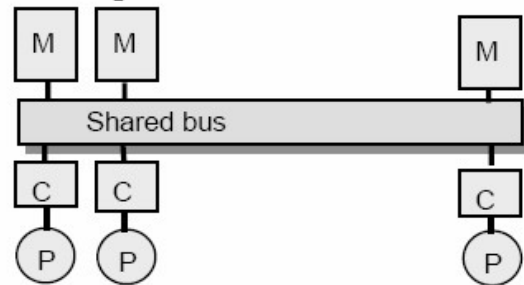
The Object Management Group's Common Object Request Broker (CORBA) is a widely used standard for distributed object systems. Other object management systems include the Open Software Foundation's Distributed Computing Environment (DCE) and Microsoft's Distributed Common Object Manager (DCOM). CORBA specifies a system that provides interoperability among objects in a heterogeneous, distributed environment in a way that is transparent to the programmer. Its design is based on the Object Management Group's object model.[11]

**3. Case Study: (implement the protocol in data consistency problem).**

Now in the first scenario the multi processor problem is the cache coherence problem. This problem is solves by the **Peter Stenström** by giving the two protocol.

- Snoopy Cache Protocols
- Directory Cache Protocols

Snoopy protocol distributes the responsibility for maintaining cache coherence among the entire cache controller in a multiprocessor. When an update action is performed on share cache line, it must be announced to all other cache by a broadcast mechanism.



In the directory cache protocols: The basic concept is that a processor must ask for permission to load an entry from the primary memory to its cache. When an entry is changed the directory must be notified either before the change is initiated or when it is complete. When an entry is changed the directory either updates or invalidates the other caches with that entry.

Now the second scenario is multi computing in this scenario the major problem is data consistency but it can be remove by the using the message passing technique but the sometime message passing technique is failed and its very costly to implements.

So that if we want to complete remove the data consistency problem in the distributed environment so that we change the technique. The idea of the paper is if we modify the protocol then may be this problem is removed.

Another thing is if we use the directory protocol in the distributed computing then cannot be necessary to implement the mediation system between the computer and the network. So that the directory concept better than the message passing technique through the mediation system.

Another solution of the problem is if we change the design of the distributed environment then it definitely complete remove the data consistency.

Now these days many of the researcher work on the how to change the design of the distributed environment for better performance as well as reduce the data consistency problem.

### **Conclusion:**

The conclusion of the paper is that how to remove the data consistency problem in distributed environment. The many solutions given by the researcher like message passing through mediator system like CORBA but it is very costly and sometimes it can be failed. So that if we define the protocol at the server level called directory protocol so that we cannot used the mediator system to reduce the data consistency problem in the distributer environment.

Another solution is that to reduce the data consistency problem is the change design of the distributed environment.

### **References:**

- [1] Andrews, Gregory R. (2000), *Foundations of Multithreaded, Parallel, and Distributed Programming*, Addison–Wesley, ISBN 0-201-35752-6., p. 348.
- [2] Godfrey, Bill (2002). "A primer on distributed computing"  
(<http://www.bacchae.co.uk/docs/dist.html>).

[3] Dolev, Shlomi (2000), *Self-Stabilization*, MIT Press, ISBN 0-262-04178-2., p. 5.

[4] Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, *Applied Operating System Concepts*, 1/e, John Wiley & Sons, Inc, New Delhi.

[5] Peter Stenström, Professor Computer Architecture, Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden

[6] Lynch, Nancy A. (1996), *Distributed Algorithms*, Morgan Kaufmann, ISBN 1-55860-348-4., p. 1.

[7] Ghosh (2007), p. 4, 8. Lynch (1996), p. 2–3. Peleg (2000), p. 4.

[8] Lynch (1996), p. 2. Peleg (2000), p. 1.

[9] Ghosh (2007), p. 7. Lynch (1996), p. xix, 2. Peleg (2000), p. 4.

[10] Elmasri, Ramez; Navathe, Shamkant B. (2000), *Fundamentals of Database Systems* (3rd ed.), Addison–Wesley, ISBN 0-201-54263-3.

[11] A. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*, Prentice Hall, Pearson Education, USA, 2002.



