

MCAP: Multiple Client Access Protocol

Ravindra Singh Rathore
SML College Jhunjhunu, Rajasthan
India
(ravindrathore@gmail.com)

Abstract - We have implicitly assumed that the client-side software of a web-based system consists of a browser that acts as the interface to a user. This assumption is no longer universally true anymore. There is a rapidly growing group of web-based systems that are offering general services to remote application without immediate interaction from end users. This organization leads to the concept of web services.

Keywords – MCAP, ACID, Stateful Server

I. INTRODUCTION

Web service is nothing but a traditional service (e.g. a naming service, a weather-reporting service, an electronic supplier, etc) that is made available over the internet. What makes a Web service special is that it adheres to a collection of standards that will allow it to be discovered and accessed over the Internet by client application that follow those standards as well. It should come as no surprise then, those standards from the core of Web service architectures. The principle of providing and using web service is quite simple. The basic idea is that some application can call upon the service as provided by a server application. Standardization takes place with respect to how those services are described such that they can be looked up by a client application. In addition, we need to ensure that service call proceeds along the rules set by the server application.

Web service architecture is combination four things:

Web: - XML+SOAP+WSDL+UDDI

XML means Extensible Markup Language. It is meta-markup language. This is lot of flexible. It similar like HTML all kind of tag used in this.

WSDL means Web Service Definition Language. Which is a formal language very much the same as the interface definition language used to support RPC based communication.

UDDI means Universal Description, Discovery and Integration. The layout of a database containing service description that will allow web service client to browse for relevant services.

SOAP means Simple Object Access protocol. A core element of web service is the specification of how communication takes place. It makes a framework in which communication between two processes.

But in some condition SOAP cannot work for providing web service like this:

- When relying on HTTP as a transport protocol and not using WS-Addressing the roles of the interacting parties are fixed. Only one party (the client) can use the services of the other.
- It is difficult to get persistence, or for multiple clients to access the same object, rather than multiple instances of the same object. Not supporting distributed objects was something that was decided in the SOAP recommendation [6].

- All SOAP data is serialized and passed by value and currently there is no provision for passing data by reference. This could lead to synchronization problems if multiple copies of the same object are being passed at the same time.

II. METHODOLOGY

If client part is dividing in many numbers of clients or make cluster of clients, the SOAP protocol cannot eligible to communicate, because this protocol can provided service only one client.

We fulfill this condition by using protocol like this:

- For multiple clients or cluster of client, we have used Multiple Client Access Protocol (MCAP) [7].
- For Multiple Client Access Protocol (MCAP) for describe this we work on distributed architecture of clients.

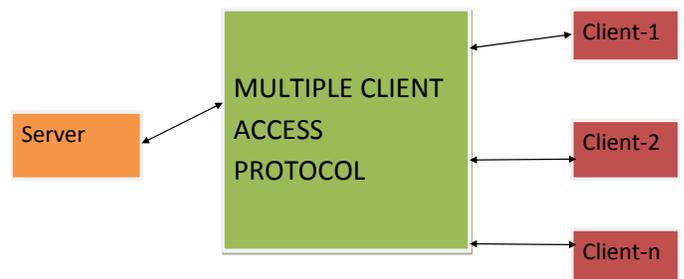


Fig. 1. MCAP

Suppose we design this model multiple clients and single server can communicate by using protocol this name is Multiple Client Access Protocol. But how server can satisfies all clients, in what way it can access it. Some problems arise like:

- How every client shares the single server?
- How the designing protocol MCAP connects with multiple clients?
- How the servers fulfill different client's requests?
- How this protocol maintains clients request and its order?
- After processing request of clients then how distribute its response to clients?
- In which way by using server maintain response to different clients?
- What thing we have used to connects different number of client to this protocol?
- If we use distributed system in this protocol using multiple clients with single server how we improve scalability and availability.
- If we implementing multithreading in server side, but how it communicate different number of clients.

A distributed system is a collection of independent computers that appears to its users as a single coherent system. Transaction

is quite difficult to implement in distributed system. For implementing this type of system we have used IDL (Interface Definition Language).

The centralized single server used for multiple services. The single server can be used for multiple numbers of clients. We use single server managing highly confidential information such as bank accounts.

The Centralized server and client in distributed form this structure similar like TREE in data structures. We design single server and multiple clients using scaling techniques because in distribution the client portion is splitting into many parts in web service.

Replication is a also used for load balancing of server and it main work to move the copy of server from another location but it decrease availability and also affect the scalability.

Some fake assumption about distributed system, we use distributed form of clients that provide certain features like:

- This type of network is reliable, secure, and homogenous.
- Topology does not change and latency is zero.
- Bandwidth is infinite and transport cost is zero
- One administrator domain or server has zero cost in transportation.
- Distributed system need to take heterogeneity into accounts.

Distributing of client also follows the ACID properties. ACID refers:

1. Atomicity. In a transaction involving two or more discrete pieces of information, either all of the pieces are committed or none are.
2. Consistency. A transaction either creates a new and valid state of data, or, if any failure occurs, returns all data to its state before the transaction was started.
3. Isolation. A transaction in process and not yet committed must remain isolated from any other transaction.
4. Durability. Committed data is saved by the system such that, even in the event of a failure and system restart, the data is available in its correct state.

III.WORKING OF PROTOCOL

Thus this transaction model divided into three main parts:

- Server
- MCAP (Multiple Client Access Protocol)
- Clients or cluster of clients

First we work on which type of server is applicable fulfill the condition of this protocol, thus we have used different form or state in distributed system:

- **STATELESS SERVER:** This type of sever actually does maintain information on its client, but crucial for information is lost, it will not lead to a distribution of service offered by server. For example a web generally logs all clients' requests.
- **SOFTSTATE SERVER:** In this case, the server promises to maintain state on behalf limited time. After that time has expired the server falls back to default behavior, thereby discarding any information is kept on account of the associated clients.
- **STATEFULL SERVER:** It generally maintains persistent information on its clients this means that the information needs. This means that the information needs to be explicitly deleted by the server. This type of server would maintain a table containing (client file) entries. Such a table allows the server to keep track of which client currently has the update permission on which file and thus possibly also the most recent version of the file. This approach improves the performance of read and writes operation as perceived by the clients. Performance improvement over stateless server is often an important benefit of state full server.

If the sever crashes, it has to recover its table of client file entries or otherwise it cannot guarantee that it has processed the most recent updates on a file.

We are designing that type of server which provides the better communication in between using protocol (MCAP). This protocol connects with multiple numbers of clients. Thus we designing that type of server can communicate with multiple clients, for developing thread system in server. Thread means it can execute its own piece of code[8][2]. We develop multithreading in server. After developing thread in server can helpful for communicating multiple clients. In sever divided in two levels one is user level and second is kernel level:

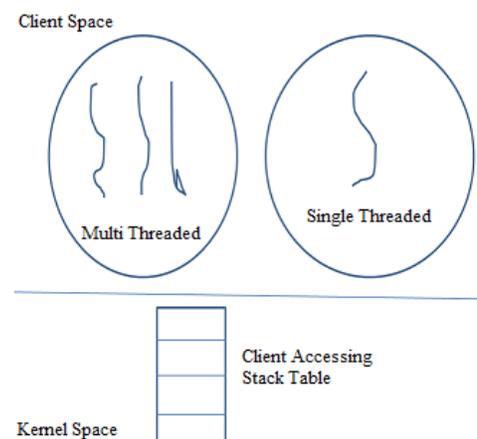


Fig.2. Memory System of Transaction Processing

Two modes of thread used in the server for providing proper communication to different number client that available at the given protocol (MCAP). These two modes are:

- Active mode.
- Passive mode.

Active Mode: when client can send request first its connected to the port or access point after it enter in protocol then request reach at distribution block distribution send a copy to dispatcher with help of replication, the dispatcher maintains the client table. The client table can helpful because it provide facility to other client don't entry, at that time when processing some other client. After sending copy to dispatcher, the server sends the thread that can carry the request of client and distributor send the request with thread. The client request and thread reach on server. After the sever response to client the server again thread can carry the response reach at distribution block. The dispatcher allows the distributor to distribute the response to client in way an active thread can work.

Passive Mode: this mode similar like an active mode, first client send request and server send response after other client2 can connect with other access point or port2 after then it reach at distribution and same process repeated like an active process, thus the client1 can used the thread this thread is called passive thread. The number client that request is completed and using thread by these client these thread is called passive thread.

Every client can communicate with server with fixed or particular time interval Δt . Δt represents the time interval between active and passive mode, the time interval is predefined that fully depends on time. If sometime the time is short the client can send it request to but time is over that the client request is not completed. Vector time stamp logical clock in distributed system Δt . Vector clock comparing the relationship between two events or clients

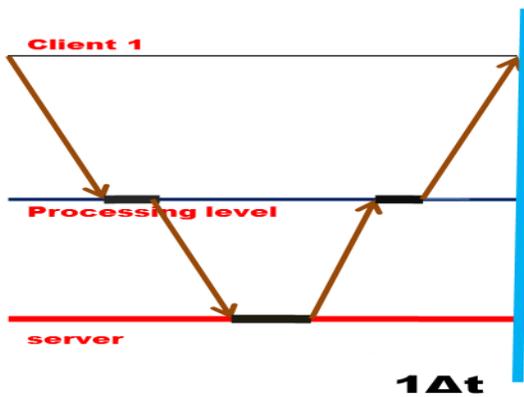


Figure 3a: Processing of Client 1

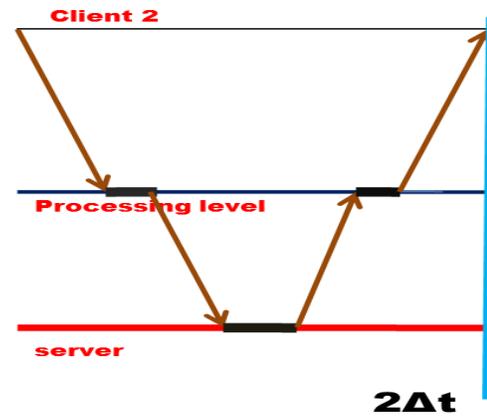


Figure 3b: Processing of Client 2

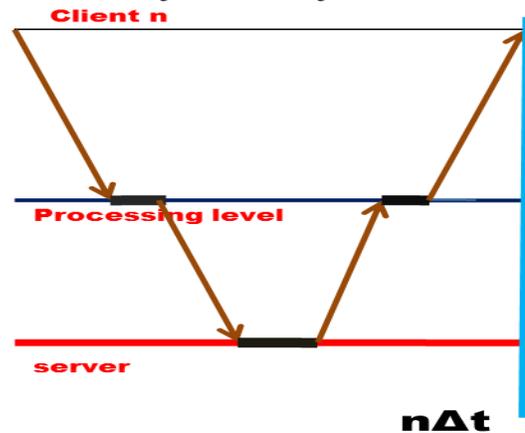


Figure 3c: Processing of Client n

MCAP (Multiple Client Access Protocol)

A complete description of how the Multiple Client Access Protocol (MCAP) can help for communicating multiple clients and single server. In this protocol we provide facilities for different number of clients with single server. In server section defines how it can control the different clients request and response for developing multithreaded system in server. The distributed system carefully studied how we exchange the information. The rules of communicating process must known as protocol or the communicating process is control between different clients and server for using protocol. This protocol divided into four layers. RPC (Remote procedure call)[5] is widely used model for communicating number of clients with single server. An open system is one that is prepared to communicate with one another open system by using standard rules that govern the format, content and meaning of the message sent and received. We have using two types of protocols for communication at web service. Connection oriented and connectionless oriented

Different component of MULTIPLE CLIENT ACCESS PROTOCOL (MCAP):

- PORTS
- DISTRIBUTORS
- DISPATCHER
- CLIENT TABLE

PORTS: The alternative name of PORT like END POINT or ACCESS POINT. The access point or end point available in this protocol for plug-in the different number of clients and it can for easy communicate with server. Client contact a server in all cases client send requests to an end point also called port. Server gives the response on those client requests that uses the port or specific end point. Port or endpoint can indirectly connected with server. In this protocol endpoint or port are attached at only in client side. These endpoints have been assigned by the INTERNET ASSIGNED NUMBERS AUTHORITY (IANA). With assigned endpoint, the client only needs to find the network address, where the server running. Many passive clients, it is often more efficient to a single server provided a specific server with every endpoint.

The Access point used two modes: (a) Push mode (b) Pull mode

(a) Push mode: This mode push clients into the protocol for completes its request.

(b) Pull mode: This mode helps the clients can receive the response.

DISTRIBUTORS: it's the second section of this protocol. The clients request can push by access point or port then the port sends the request to distributor. The distributor is important layer of this protocol, this play an important because it can receive the request from client after the receiving it can send a copy of client to the dispatcher. For sending a copy of client by using replication process. The dispatcher can maintain a client table. In client table stored the information of client that can communicate with server. At that time when client can communicate with server. No other client can communicate at that time. Because in distributor can pass only one client at a time. When server sends the response again pass to the distributor. Again the distributor can receive the permission from the dispatcher. The dispatcher can contain the client information from the client table. After dispatcher can give the permission to the distributor. The distributor can collect the permission from dispatcher then it distribute to the client according to dispatcher information.

DISPATCHER: third section of this protocol. In this part we contain the data of client. Which client pass to the distributor to the server side thus the distributor can send a copy of client with help of replication. In client table we store the general information send by distributor. When the server reply than again return at distributor, the dispatcher can permission to dispatcher can match the information of the particular client after match the send to its client.

CLIENT TABLE: the client table is forth section of this protocol. The client table maintains the client information. Client request can enter in the distributor, and then the distributor can send copy of client information. The information like client no. and using port no. can maintain by client table. The information about client in client table can transfer to the DISPATCHER. When request completed the server send the response. The response arrived at distributor. The distributor can contain information the response where it can send. Then the client table can pass the client information dispatcher. The dispatcher transfers the client no and port no. to the distributor. The distributor already contains the response from server after collecting information distributor can dispatch the response to particular port no. and client no.

In this way all section like (PORT, DISTRIBUTOR, DISPATCHER, and CLIENT TABLE) of this protocol can help for communicating multiple clients with single server. Using protocol named is MULTIPLE CLIENT ACCESS PROTOCOL.

In this protocol can communication between multiple clients and single server thus the communication is called PERSISTENT and TRANSIENT communication. In persistent communication a message that has been submitted for transmission is stored by the communication as long as it takes to deliver it to the receiver. In transient communication a message is stored by the communication system only long as the sending and receiving application are executing. This type of communication scheme corresponding with REMOTE PROCEDURE CALL (RPC)

Clients or cluster of clients

This is last part of transaction system. In this part we have used cluster of clients. The main problem is how the multiple clients can communicate with single. Thus this problem is solved by using the protocol of MULTIPLE CLIENT ACCESS PROTOCOL, in this protocol at client side we add ports or access point or end point that can helpful for entering the clients into the protocol. After the client request reach at the distributor. The distributor sends a copy of information about client details. After sending the copy to client table the request of client can distributed to server. At time when any client can communicating that time no other client can communicating with server. Because in sever we have use multithreading, by using multithreading we use two modes of thread can encounters the clients active thread and passive thread.

After completing request the server send the response, after the response reach at distributor again then it collecting information about client can collecting from the dispatcher. The bases of dispatcher information the distributor can dispatcher the response of client to its particular port number then transfer to the specific client.

These three partition of a simple transaction system SERVER, MCAP, and CLIENTS OR CLUSTER OF CLIENTS.

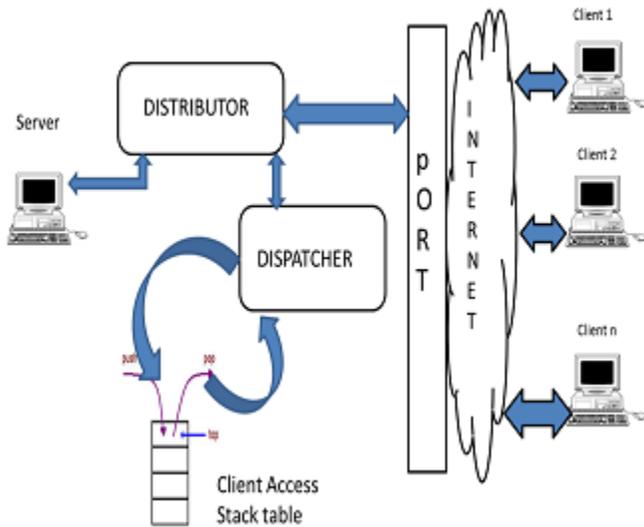


Fig. 4. Simple Transaction Model of Using Multiple Client Access Protocol

Simplified architecture of MCAP it can support three tiered architecture in this architecture programs that form part of processing level. Three –tiered architecture is a good example of transaction processing. In distribution system we have used VERTICAL distribution is a characteristic feature placing logically different component on different machines. Vertical distribution is only way of organizing client server application. Horizontal distribution is used in only P2P system [1].

After developing thread in server can helpful for communicating multiple. A complete description of how the Multiple Client Access Protocol (MCAP) can helpful for communicating multiple clients and single server. In this protocol we provide facilities for different number of clients with single server. In server section defines how it can control the different clients request and response for developing multithreaded system in server.

REFERENCES

- [1] ABERER, K. and HAUSWIRTH, M.: "Peer-to-Peer Systems." In Singh, M. (ed.), *The Practical Handbook of Internet Computing*, chapter 35. Boca Raton, FL: CRC Press, 2005..
- [2] ANDREWS, G.: *Foundations of Multithreaded, Parallel, and Distributed Programming*. Reading, MA: Addison-Wesley, 2000.
- [3] BAL, H.: *The Shared Data-Object Model as a Paradigm for Programming Distributed Systems*. Ph.D. Thesis, Vrije Universities, Amsterdam, 1989.
- [4] BALAKRISHNAN, H., LAKSHMINARA YANAN, K., RATNASAMY, S., SHENKER, S.,STOICA, I., and WALFISH, M.: "A Layered Naming Architecture for the Internet." *Proe.SIGCOMM*, (Portland, OR). New York, NY: ACM Press, 2004.
- [5] BIRRELL, A. and NELSON, B.: "Implementing Remote Procedure Calls." *ACM Trans.Compo Syst.*, (2)1:39-59, Feb. 1984.
- [6] COULOURIS, G., DOLLIMORE, J., and KINDBERG, T.: *Distributed Systems, Concepts and Design*. Reading, MA: Addison-Wesley, 4th ed., 2005.
- [7] HENNING, M.: "The Rise and Fall of CORBA." *ACM Queue*, (4)5, 2006.
- [8] LEWIS. B. and BERG, D. J.: *Multithreaded Programming with Pthreads*. Englewood Cliffs, NJ: Prentice Han, 2nd ed.,] 998.

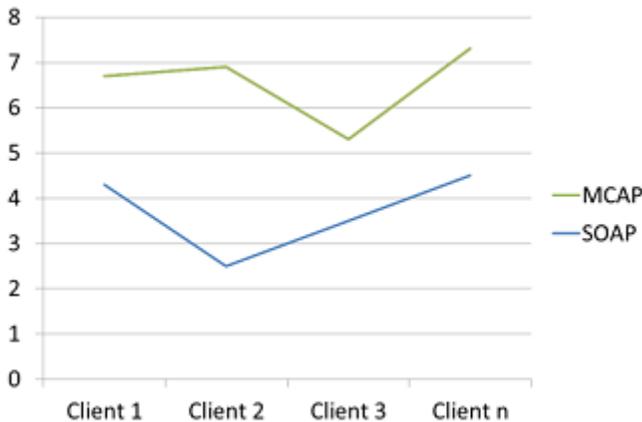


Fig.5. Graph Representation of MCAP Vs. SOAP

IV. CONCLUSION

We are designing this type of server which provides the better communication in between using protocol (MCAP). This protocol connects with multiple numbers of clients. Thus we designing that type of server can communicate with multiple clients, for developing thread system in server. Thread means it can execute its own piece of code. We develop multithreading in server.