iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 2 **Issue:** VI **Month of publication:** June 2014

**DOI:**

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

# An Interface between Client and Server for Web Application Database

Richa Sharma[1], Stuti Gupta[2]
*Department of CSE, Maharishi Dayanand University, Rohtak, India*

***Abstract:*** ***In this paper we introduce a solution for the client to access the data in convenient way. In client server model the server is the middle identity that can be installed to serve the basic purpose of the multi client environment. A client can send the request to the server and then wait for the response. The response time depends upon the speed of the server, speed of the network channel and the number of clients attached. This may be a time consuming process if the speed of any of the entities is slow or the number of clients are very large in number. Therefore the basic purpose of the Interface is to create a middle layer for the company's server and its business clients. This interface is used to facilitate the end users which were earlier being facilitated directly by the Main Server. With the help of the Interface one can easily find a fast solution of one's fired request without any intervention of the Main Server whether the connectivity is available or not. This application is a very good solution for all those organizations who want to make their work online without the help of a fast network channel. The web applications are not always be in the fast running mode because of many users. It may be possible that the database capacity is very high and the channel speed is low and in that situation one can easily do his job without the fear of the channel speed. This interface is a very good suited for the far flung areas where it is difficult to provide the connectivity and the company wants to have a client database***

***Keywords: Client, Server, Interface, Database, Distributed database System, Network database system, Distributed application architecture***



## 1. INTRODUCTION

Client Server is an old Technology where many clients jointly send their request to the server and the server can respond them as per the availability of the time frame. Every time when the user requests for the information he/she has to sign up through a UserID and Password. Every time for the security point of view server checks the identification of every user. A server has to do two processes simultaneously; one is the security check and another is the data sharing and these processes require good speed. Now a days we have a very high speed channel (network) but still as the number of users are increasing day by day in some of the areas this speed is not sufficient.

Interface is a process that attempts to make a copy of a company's current data, or selected sets of data, onto other storage devices. Preferably, this should be performed in the background. Thus the applications using the data are not disabled and disrupt. This copy may then be kept on a local storage system (LS), ready to be used in an instant should the primary system fail; alternately, it may reside at a distant location (local distant server), quickly available should a disaster hit the primary data enter. In addition, the copies may be used for advanced backup initiatives, server or storage consolidations, or data migration.

In the market one of the Techniques available is the Data Replication Process, which can solve the problem of data sharing. This is the process of making a replica (a copy) of the information which is available at one area to some distinct area. On the Internet, a Web site that has been replicated in its entirety and put on another site is called a mirror site. Such mirror sites can exist in various locations, which are geographically distant. Using the groupware product, Lotus Notes, replication is the periodic electronic refreshing (copying) of a database from one computer server to another so that all

users in the Notes network constantly share the same level of information.

Database writes are sent to the master database server and are then replicated by the slave database servers. Database reads are divided among the entire database servers, which results in a large performance advantage due to load sharing. Our aim is to give another name to the replicated data to the local database server and that is an interface between the end user and the main server.

In addition, database interface can also improve availability because the slave database servers (Local Server) can be configured to take over the master role if the master database server (Main Server) becomes unavailable.

Thus defining an interface database is pretty easy. The basic idea is that we want to copy data from one place to another. While the concept of interface is simple, several processes are involved in the interface namely collecting the data to be replicated, defining the condition of transferring the data and actually transferring the data. Whichever method of replication is used the processes remains the same.

*Need of Interface*

As mentioned above the database was originally aimed to serve as data backup methodology at local server level. The idea was to keep the data at another storage system to use it at every moment of time whether the system is online or off line. In case of failure of network the offline job will fulfill the user's requirement. This offline database could be used for all other processes as well except latest update record (sometimes Therefore the data as an interface has advantage of providing alternative mechanism of speed up data without the intervention of the server.

*Nature of operations:*

*Database initialization:*

The application can enter a newly published data to the subscriber at a predefined time interval or on continuous synchronous basis. This data is being entered the time when we can start the local server.

*Open database:*

Every time when the user wants to have an interaction with the server he can make an open request to the local server and the moment he can submit the UserID and Password the security measure can be verified by the local server. When the check is over a file is open and maintained by the local server that can provide the opening server mouth for the user. A log file is also created for the user that can have the complete log detail of the user. And if many users are trying to avail the services available, two log registers are created, one for each user and the other one that can maintained the complete detail of the local server's users. It means the main log register that can have the detailed working knowledge of the clients that are attached with it(LS) and this can be done whenever subscriber wants to do the same.

*Data updation or modification:*

The main problem faced when we are working with an on line application is the data consistency but the solution at this level is both the local server and the main server have flag out with a check alert that some modification is being done by some of the client. The check alert is first given by the client that can demand for the data modification. This signal first goes to the local server attached and then from local server to the main server so that the link is provided to the client for the data updation. A lock for that period is being given to that particular database. The moment when updation is complete the server can update all the local server and the process is continued by all the clients. At a first look this process is very tedious but because as the speed of the available network is very high a very small interval of time is required for the same. In Section 2 we have discussed Literature Review, in Section 3 Proposed Work, in Section 4 Implementation of proposed work and finally in Section 5 Paper is concluded.

## 2. LITERATURE REVIEW

*Existing Application Deployment Scenario at an Organization*

An organization always demands for the detailed client structure and the shared environment. As we are growing in the field of IT organization wants to have an online application for its regular working. Now if the work is small there is a need of small speed channel for the transaction of the database [1]. A normal client server modal can take care of the same. The server can easily facilitate by its services to each of its clients. The

client can get the online facility almost at every moment of time. The up speed here is 99%.

On the other hand if we can talk about the big organization then the client can face some problem while providing the online facilities to the end user. Since there are so many clients doing their job and it is very difficult to have an online environment for everyone. They can maintain this facility with the help of the fast channel of internet. Now one of the major issues is there and that is data the security [4]. Many times we have seen that the important data is being lost by some of the reason or is hacked by some illegal user. To maintain the first one the DBA can play an important role.

Interface designed under this can handle the database in such a manner that only the required portion of the database can be shared by the server and the rest of the database is untouched. It is somewhat equivalent to the virtual database that is available for the end user but in a hidden format. Our main aim is to look forward for the same kind of environment and resolve every problem faced by a small as well as big organization.

An Organization has been implementing numerous applications of central and distributed nature. Many of the application users are located in the local areas and other for remote areas [7]. In some case like in a Government Organization, Common Integrated Police Application and Health Scheme application, there are multiple users within a city. These applications are either distributed in nature or a web-based solution.

*Distributed Application Architecture:*

Distributed application works on standalone model where in at every individual level the client application with the required database is installed at a local server level [6][7]. This has advantage of response time, as some times there is no network constraints issues problem. Thus, the need of scaling to a high-end server is also ruled out. After a certain period of time the application's database can get copied to the local server and that is sufficient for that area. Normally for a local area the policies are not changed frequently and that is why this type of interfacing is good enough.

There is a need to consolidate the information on a daily basis to the central database server. This is one of the good solution by which the remote location server can get the database back if in any case they have lost the data [8]. This is also required by senior officials for getting the top-level information for decision making from all the remote sites. The current solution to this is a combination of interfacing as well as manual process of merging of databases.

The database at every remote site is being copied to the central database server daily during the off peak period. The main server thus receives copy of the database from all remote sites through this central database server. This copy of each Local Server database is merged with the Central Database. This process is repeated for the entire Local Servers Databases received at the central node.

The common database so generated is a point-in-time copy of the remote databases thus containing all the records of the database replicated earlier. As the volume of the database increases the volume of the database at the central location gets also increases [5]. This requires higher bandwidth and more merger time. But the interfacing can reduce this problem for the local as well as the for the main server as it can pass on only the relevant database from its area to the central server and if required by any other remote location only then a replica of that is being transferred through the database server via main server.

*Web Based Solution:* The browser based application architecture requires very minimum hardware and software sizing at the remote end (client side) there by reducing the cost and time of implementation to minimum. All updates are done at the central site i.e. on the central master database server. Local application is connected over the Internet to access these databases for retrieving information and updating transactions.[3]

These applications are public domain Organization to Organization based or Organization to Customer based e-governance applications where Internet link break-up and high concurrent users delay the delivery of the services. This causes a huge embarrassment to the user of the application, which in the above applications is Police department and doctors respectively.

## 3.PROPOSED WORK

The client can be connected to the main server through the central database server which is a part of the central administration, in our case which is the Interface.

ISSN: 2321-9653



*Terminology Used:*To describe the proposed interface architecture we need to understand certain terminology and their functioning as they are used in this solution.

*Transaction Table:*

Tables in the database those are used by application programs for recording transactions as per the business logic.

*Master Log Table:*

Master log table is the one in the database used as look up tables for getting the codes of lists at the remote area site

*Node:* The database instance at any site involved is known as node.

*Central Node (Administrator):*

There is one central database consisting of consolidated information of all the remote databases (called stations in this paper). The central database administrator is responsible for creating and maintaining all the master files. Any updates on these masters are done centrally at this database.

*Local Server Nodes:*

As mentioned above a Local Server is a remote site containing the local copy of the database. The end user application is configured to make changes in the local database. These applications are permitted to change only transaction tables using the application interface woven with business logic.

*Master Server (the Publisher):*

The site that hosts the database and publishes the information is known as Master Server. In the present problem the node that publishes the data is known as central node.

*Local Server (Subscriber):*

The site that subscribes the articles from the master server receives periodically these articles.

*Data Administration.*

The data admin that we have proposed here have the responsibility of taking care of the entire data management connected to the central database server. The moment any part of the system get disturbed by some miss happening the data administrator can make that active and provide entire resources. It can also provide an alternative path to the problematic area so that the nodes that are connected to the zonal server get the connectivity to either the neighboring zonal server or to the central database server. The ultimate aim is to give a healthy environment to the local host so that the client cannot be the sufferer.

## 4. IMPLEMENTING THE PROPOSED WORK

Our system can maintain three different log registers, one at the main server termed as main log resister that is having the complete record of every client connected through the zonal server, second is the zonal log register available at each zonal server termed as zonal log register, the last one is the client log register which is allotted at each client. The client log register can be having the record of the tables of record that are related to the clients. If any new client wants to add himself at the zonal level then its information is added to the zonal server and a log register is created for this new client. Now this log register's entry is added to the zonal register and because the same type of the log register is also there at local, zonal, and main servers this is very difficult to maintain all the log registers. To solve this issue the log register of the clients has been exempted and the responsibility goes to the zonal server.

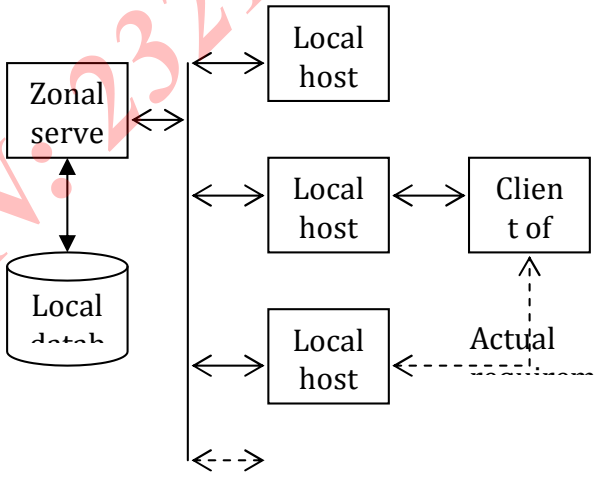
Now when there is any requirement for the client to modify the database the client can send the request to the zonal server to give him permission, this permission can be first checked by the zonal server that whether the similar kind of request is being requested by the other client or not. If no such type of request is being there the zonal register can be activated with a exclusive lock for the requested client. Now this is the time when the requested client can get the facility to update the database. After this process is over the same job is being taken care between the zonal server and the central database and then central database to the main database server. It will be discussed in the latter section. We have to consider the client request raising, request submission, request acknowledgement, and at the end request handling.

ISSN: 2321-9653



*First case*: Every time when the local host has open its account, its first job is to get the log register file from the zonal level server to its log area. With the help of this file the local host can get the details about the existing clients. This log register is a very small information file where the entries are only the client ID and the date of joining. This is for the entire clients at a particular zone. Here we are taking an example of North Zone only. Suppose that a client is coming at north zone. He is interested to join the community for the future work as a member of the organization. In that case the client has to fill a form for the registration. The local host can provide an e-form. The entries that are being filled by the client are being submitted on line and then entered to the record list. This entry is then goes to the central database server to update the local host log register. Now in the future when the same client is coming for the next or other job it can be easily be recognized by he local host because the local host can have that information with it. Now suppose that the local host is offline and the same client is coming for the job but due to off line its information is not available. In that case the local host can check with the duplicate copy of the log register and if that duplicate copy is also not having the record of the current client it can submit its record as a temporary basis so that it gets an easy feel with the system. Now the moment when the local host gets the connectivity with the central database through the zonal server it can be checked immediately the status of the client. If the client's information is not available in the central database then it can be updated on the same time and then generate an client ID and send it back to the local host to update its local log register. The local host can attach the Client Id with all its information and send it to the zonal server. At the end of the day or the set time period the zonal server database can be copied to the central database. The above is the simplest case for the required Interface. But every time this case cannot fulfill the system's requirement. For a client who is the part of one zone wants to get the facilities of the system through another zone. Now in this case the work is little bit difficult or when a client wants to utilize the facilities available at different zone. Some of the issues we will discuss in the sections below.

*Second case*: in this case the clients' need the facility which is not available at connects to the local host but that is available at another local host and that one is not the part of the current zone. In the normal case when there is direct link in between the local host and the main server this case does not face any problem but here as we are having an interface it is very difficult to facilitate the client through the local host. Now to solve this issue we have created a new idea. The very first thing is to create a temporary log register for that client where the clients ID is being copied and then send to the central database server through the zonal server. The central database server can check it from the available log register. The moment it can find the entry it can send an acknowledgement with the local host ID to the requested local host. Now the current local host can send the request of the client to the local host whose ID is being given by the central database. This can be looked after by the zonal server because the main responsibility is of the zonal server to take care of all the jobs taken by the local host. The figure shows this case that how one client can get the facilities of its own local host through another local host




*Third case:* This case little bit more complicated then the second case. Here the client of one local host demands the facilities of the services available at some another local host which is a part of another zonal server. In this case there is a very important role of the central database server. As we have discussed above that one client whose entry, which is not available at the current local host, is being transferred to the zonal server and then to the central database for the checking. So in this case whatever be the decision of the central database, it is valid for the clients. If the entry is not available in the central database the client is being rejected by the local host and that client has to register through proper channel. Yes this is possible to enter a client of one zone through another zone. This task is automatically done by the zonal server by providing the client ID by viewing the zone address chose. For example if a client living at north zone wants to register himself by west zone, his zone address which

is currently the north zone, is automatically be added to the north zone. Now in the future he can re-enter at the local client just by giving the client ID. Otherwise the zonal server local host can give an appropriate ID to that client. Now it is the choice of the client to accept the ID provided by the zonal server or to choose the new ID. This is one of the facilities provided by the system to change the ID as per the requirement or choice of the client, because, it may be possible that the client wants to shift its entire database from one area to another one. To do so the system can replicate the client's record from one area to another one. But it does not mean that the database is being transferred but it is just the change the Client ID and the zone address rest everything is same. Since we have the central database system this shifting is just a formality for the server but the client can get the facility as global database accessibility.

## 5. CONCLUSION

The paper deals with all the parameters that are involved in the customer satisfaction level. Our paper can deal with all the areas where the client is involved. If we can see in the detail explanation where we have taken an example of railway reservation system, it becomes clear that for such a big work this paper is good enough and provides vast area connectivity for the reservation system. We do not have such implementation at this moment of time but it can be implemented easily.

The interface that we have discussed is one of the very good results for the current and future work

Now a frequent question rose that how this system can be improved more for the entire database management system. This part can be solved just by adding two technologies here that we have taken on the partial basis

1. Distributed database system and
2. Network database system.

With the addition of these our system becomes more advantageous and affective. As such we have added indirectly both these Distributed and Network database facilities but currently we have not embedded these fully.

## REFERENCES

[1] MacIntyre D., "Meeting great expectations - client/server technology trends - Industry Trend or Event", Software Magazine, Oct 15, 1995.

[2] Pohjolainen T., Niemelia I . "Model Checking a client – server system with a scalable level of concurrency" , Software Magazine , May 1998.

[3] Shopov M., Spasov G. , "Distributed Measurment System Based on Java and Web Technologies", Technical University of Sofia, Feb 2000

[4] Laing A., "The Security Mechanism for IEEE 802.11 Wireless, Networks", November 24, 2001, Version 1.2f

[5] Duchessi P. "Client-server implementation: some management pointers Engineering Management", IEEE Transactions on Volume 47, Issue 1, Feb 2000 Page(s):127 – 145

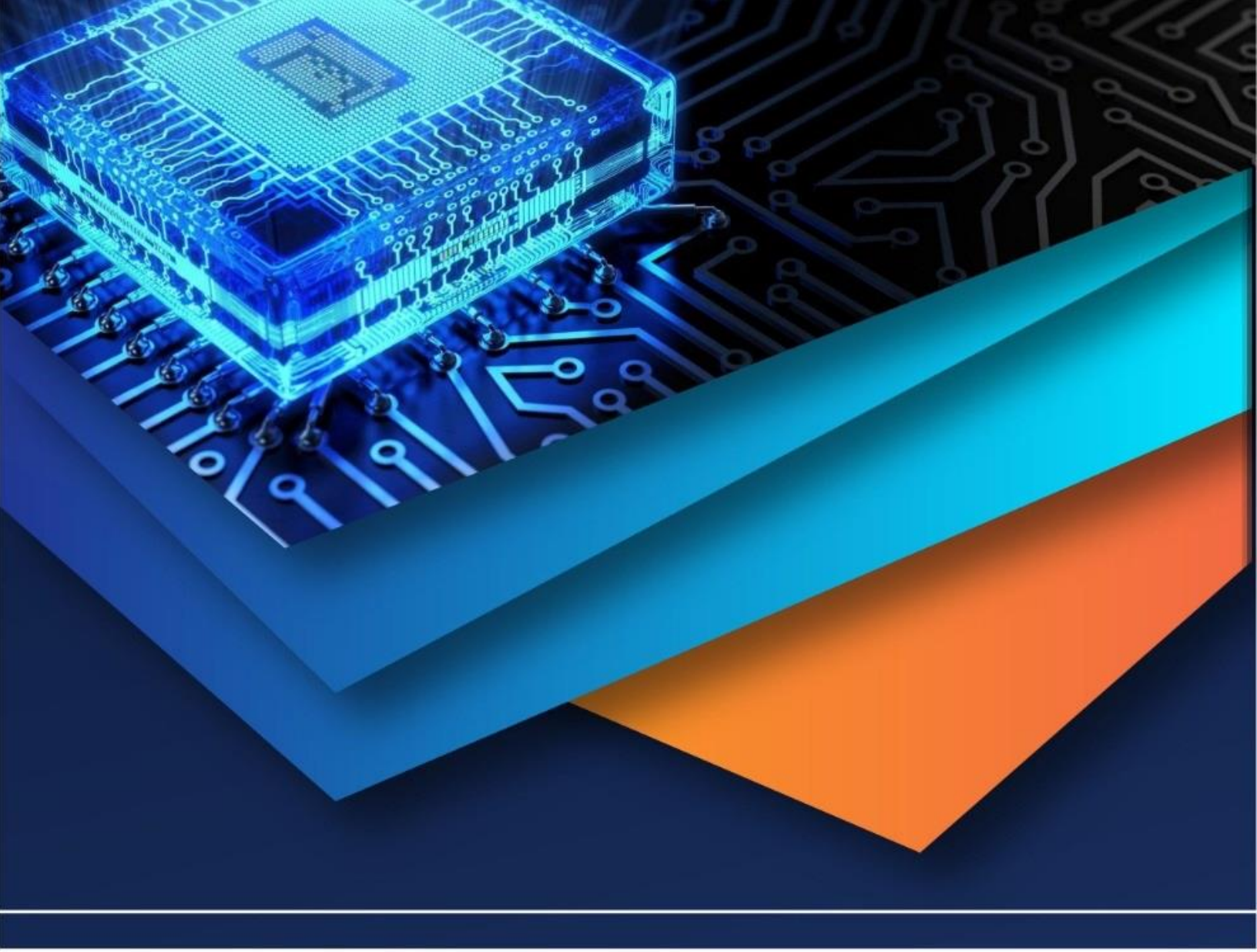
[6] Rodrigues L., Carvalho N., Vilaca, R., Oliveira, R., Guedes S., " GORDA – An Open Architecture for Database Replication" Network Computing and Applications, NCA 2007.

[7] Carter T. Symantec, "Long Distance Replication Technologies", Symantec Veritas Architecture Network, 2007

[8] Adams K., Gracanin D., Hinchey M.G., "Increasing Resiliency through Priority Scheduling of Asynchronous Data Replication" , 11th International Conference on Parallel and Distributed System, 2005, IEEE 2005.

[9] "Client/Server: Past, Present and Future"; http://www.dciexpo.com/geos/dbsejava.html

[10] "Client/Server Software Architecture – Overview", http://web.simmons.edu/~benoit/LIS455/Client-Server1.doc



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY