ISSN 22773061

Distributed OPAC System Using Z39.50 Protocol

K. Madhuri[1] Y. Chandramohan Reddy Reddy[2] K.Neeraja[3]

1M-Tech in CSE, CMRTC, Hyderabad, India,

madhuri.it1234@gmail.com

2M-Tech in CSE, CMRTC, Hyderabad, India,

chandram97@gmail.com

3Associate Professor CSE, CMRTC, Hyderabad, India,

kneeraja123@gmail.com

ABSTRACT

Online Public Access Catalogue is playing a vital role in Central Libraries and University/College Libraries. Most of the College libraries are using OPAC for easy search and retrieval of the books and it's Status in a Particular Library. The main limitation in the OPAC we are using in current system is; that we are able to search and retrieve the information about the books of that particular library, but not outside of it.

When we have a scenario like if we have multiple libraries in a campus, if we need to search a book in all the libraries, we have only two possibilities to build OPAC, once to make the single database for all the Libraries or you need to ask the user to search in all OPAC Systems of respective Library manually. But both above solutions are not feasible in real time environment.

So in order to have the above scenario a feasible solution we need to build a distributed environment for the OPAC, which will have all the individual databases connected remotely.

KEY TERMS

Catalogue, Distributed databases, Remote Databases, Z39.50 Protocol,

INTRODUCTION

The Concept of OPAC is already implemented in many Libraries with single databases or multiple databases with in the same server. The main challenge for us is to implement the OPAC in distributed environment. The growth and development of Information Retrieval is rapid in today's world. The key area that we are focusing to improve the OPAC functionality is to implement it in distributed databases environment.

OPAC is a gateway to access library services. OPAC provides the users online access to the library providing an option for searching the exact resource he is looking for and retrieve it upon successful search. Based on the library Management, OPAC can also provide some extra features like checking already borrowed resources, reserving for a particular resources in advance, etc. Several changes have taken place and OPACs have improved significantly since then. The 3rd generation OPACs incorporates features that are characterized by the facilities of World Wide Web (WWW). Now the major Goal is to integrate the above features of OPAC into distributed environment. The major Challenge here is that we need to connect to various databases that are remotely hosted and need to perform the various operations on them. When we go for Certain University, you will have departmental libraries for each department, and central library, if you need to connect all the databases of libraries in the university; you need to follow the principles of distributed databases. In order to use Distributed databases to interact with Client (OPAC interface) and Server (databases) in remote we need to have proper Communication Channel. Here we are going to use Z39.50 Protocol, which is a powerful communication tool based on client-server interaction (search interface to the catalogue and other resources on the net).

OPAC HISTORY

In the early 1980s, catalogues only displayed the bibliographic information for monographs and serial titles physically held in a library (Norgard et al., 1993).

Baker and Lancaster (1991) noted that library catalogue use, as increased and have two important aspects. First, librarians are becoming more concerned with the evaluation of library services in general; they want to know how well the catalogue performs, what are its deficiencies, and how effectiveness can be increased. Secondly, many libraries are replacing traditional card catalogues with OPACs [8].

The computerized catalogue is commonly referred to Online Public Access Catalogue (OPAC) (Chen, 1991). There is no clear definition of the online catalogue [4]. It has been defined in various ways by libraries and there is little consensus about what really constitutes an online catalogue [3].

The Library of Congress has defined the online catalogue as:

An online catalogue is an access tool and resources guide to the collections of a library or libraries, which contains interrelated sets of bibliographic data in machine-readable form and, which can be searched interactively on a terminal by users (Fayen, 1983, p 4).

The National Library of Medicine's definition is as follows:

An online catalogue provides online access to the complete bibliographic record of all of the library's holdings with minimal access points being the same as those available in a card catalogue (Fayen, 1983. p 4).

SEARCH CRITERIA:

Various Search Criteria [6] that we used as keywords for searching the catalogue are as follows:

- **Title Search:**
- **Author Search**
- **Subject Search**
- **Publication Search**

The above Criteria are just like filters used for performing the search action. This reduces the processing time and increases the query execution speed showing the response back to the user in no time.

USER INTERACTION WITH OPACS

1. User attitudes and behavior:

The survey show that the most users like the OPAC system, and it is helpful in getting the status of book with no strain. The user's attitude changes based on number of times the user has interacted with various OPAC systems. The major problem that effecting the user's attitude is that they need to search for all the OPACs if they want to search for particular item if there are more databases.

Any user when interacted will expect some features that will make the user to get his information easily. Survey has found that when different levels of user interaction levels are provided like “”expert”, “novice”, most users selected for novice. That makes implication that users need to have a simple and must provide exact results.

There is also a scope that users will think that they know the interactive systems after some successful transaction, but in doesn't mean that their level of expertise has been increased. So when it comes to distributed OPAC the user must be able to understand the system and the features must also be added to make it easy for the user

2. OPAC input from users:

Every OPAC system must have input given by the users. Now its users dependent that what input he is willing to give, it can be either commands, selection, but its mandatory that he has to provide the search string which is helpful in getting the results,

Now the major challenge is that its user has to provide the exact search string so that he can view his exact output. But we cannot assure that user will provide the correct, no grammar mistakes, no spell mistakes in this key words, so now it's our responsibility to build more sophisticated system, which will allow the user to provide the exact input.

The best way to reduce the errors in inputs is, to have touch based input devices, or we can have spell checker based on our database, or we can have auto populate the key string in advance based on most searched string, or string that matches to his input characters.

3. OPAC output:

The Output is basically displayed on VDU, or can be as printed copies as requested. The major constraint here is that what part of data needs to be displayed on the screen, basically these are two type of data to be shown in OPAC, they are brief and less brief. But if we display the complete details on the screen that will increase the length of the output page which will be problematic for the user that he has to scroll the page to view other results, So the out of the search must be precise and important information needed.

NEXT STEP IN OPAC – Distributed OPAC

The major challenging and improvement needed is integrating the OPAC in distributed environment where a single system has to interact with multiple databases which are inter-connected with each other. In distributed environment the search query is passed to master database which will in turn sends it to other slave databases, the result will also sent in the same manner. This will increase the round trip time. Fig 1. Represents the same architecture.

In Order to resolve the round trip time conflict we use Z39.50 Protocol where saves will sends the search results to system directly instead of sending to master database.

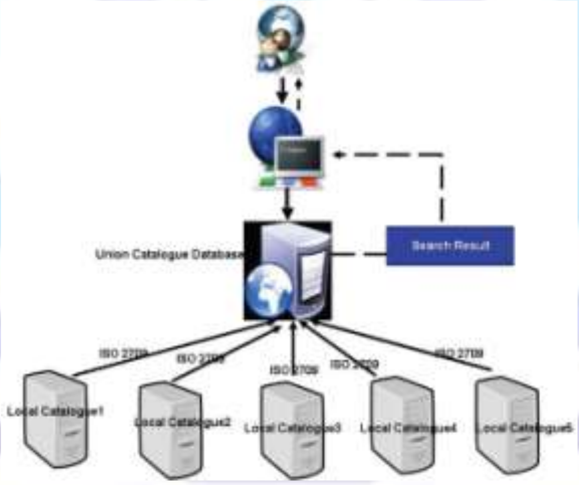


Fig 1. Traditional Distributed Architecture based OPAC System

Distributed OPAC - Z39.50 PROTOCOL:

Z39.50 protocol is the tool used to search the databases over internet; it is the barrier between the client server interactions over internet. The latest version of the Z39.50 standard is Version 3 and it was approved in 1995 by the American National Information Standards Organization (NISO). This version is also being adopted as an international standard, replacing the ISO Search and Retrieve (SR) standard approved by the International Standards Organization in 1991. Z39.50 protocol is basically plays a role of communication channel between OPAC user interface and TCP/IP protocol, which help in showing up the catalogue information after interacting with the database server. Fig 2 represents the Z39.50 based OPAC system.

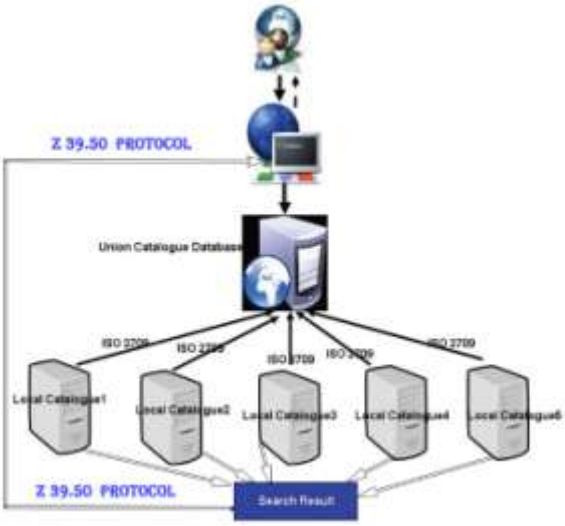


Fig 2. Z39.50 Protocol based OPAC System

Here the Z39.50 protocol will accept the input and it will processes it and interacts with TCP/IP, the TCP/IP in further will interact with servers (database and business logic). Once the data is processed in-turn the TCP/IP will send the data to Z39.50 and then it will display the same in user Interface.

Z39.50 Services:

1. INIT
2. SEARCH
3. SORT
4. PRESENT

INIT: - To establish a Z39.50 association.

SEARCH: - To query the remote database.

SORT: - Sort the retrieved data.

PRESENT: - To Display Sorted data.

Other Z39.50 **Facility** protocols exist to support such features as:

- **Sort** the results as specified by the user.
- **Delete** search results, either entirely or for specified records.
- **Scan (**browse) through index lists of items such as subject terms, titles, author names, and other database fields.
- **Access Control** through authentication and passwords.
- **Resource Control** and termination of Z39.50 search sessions by the client or server.

Z39.50 Distributed search

Usually Z39.50 protocol based OPAC only provides information that resides in their local databases - they do not perform search requests over a distributed database environment. Hence a typical scenario for a Z39.50 based OPAC is distributed database Model Searching. Here a group of Z39.50 servers are connected in parallel, among them some of servers are connected to more than one databases.

For Distributed OPAC we use different servers like UNIverse, Zebra Server. However, there is no specific requirement that a Z39.50 target must always search its local databases only. A UNIverse server, for example, propagates incoming queries to a number of Z39.50 targets, collates the results and presents them as if they were coming from a single database. As a UNIverse server is itself a Z39.50 target, this approach allows for the distributed search hierarchies of arbitrary depth.

ZEBRASRV - ZEBRA SERVER:

For distributed environment of OPAC system we are using Zebra server. Zebra is a high-performance, general-purpose structured text indexing and retrieval engine. It reads structured records in a variety of input formats (e.g. email, XML, MARC) and allows access to them through exact boolean search expressions and relevance-ranked free-text queries.

zebrasrv is the Z39.50 and SRU frontend server for the Zebra search engine and indexer. On UNIX you can run the **zebrasrv** server from the command line - and put it in the background. It may also operate under the inet daemon. On WIN32 you can run the server as a console application or as a WIN32 Service.

Synopsis:zebrasrv [-install] [-installa] [-remove] [-a file] [-v level] [-l file] [-u uid] [-c config] [-f vconfig] [-C fname] [-t minutes] [-k kilobytes] [-d daemon] [-w dir] [-p pidfile] [-ziDST1] [listener-spec...]

Options:

The options for **zebrasrv** are the same as those for YAZ' **yaz-ztest**.

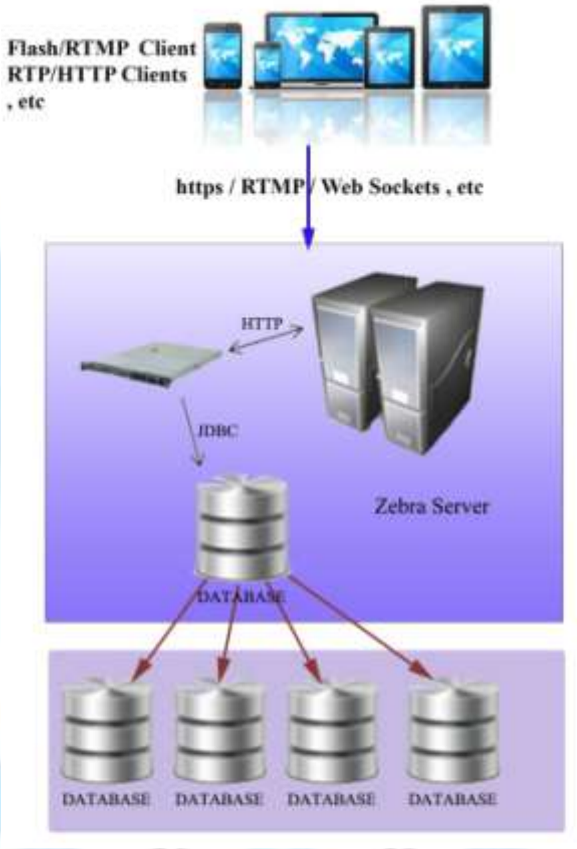
- **–c:** specifies a Zebra configuration file - if omitted zebra.cfg is read.
- **-a file:** specifies a file for dumping PDUs (for diagnostic purposes). The special name - (dash) sends output to stderr.
- **–S:** Don't fork or make threads on connection requests. This is good for debugging, but not recommended for real operation:
- **-1:** Like -S but after one session the server exits.
- **–T:** Operate the server in threaded mode. The server creates a thread for each connection rather than a fork a process.
- **-s:** Use the SR protocol (obsolete).
- **-z:** Use the Z39.50 protocol (default). This option and -s complement each other.
- **-l file:** Specify an output file for the diagnostic messages. The default is to write this information to stderr
- **-c config-file:** Read configuration information from config-file. The default configuration is ./zebra.cfg
- **-f vconfig:** This specifies an XML file that describes one or more YAZ frontend virtual servers.
- **-C fname:** Sets SSL certificate file name for server (PEM).
- **-v level:** The log level. Use a comma-separated list of members of the set.
- **-u uid:** Set user ID. Sets the real UID of the server process to that of the given user. It's useful if you aren't comfortable with having the server run as root, but you need to start it as such to bind a privileged port.
- **-w working-directory:** The server changes to this working directory during before listening on incoming connections. This option is useful when the server is operating from the inetd daemon.
- **-p pidfile: S**pecifies that the server should write its Process ID to file given by pidfile. A typical location would be /var/run/zebrasrv.pid.
- **-i:** Use this to make the the server run from the inetd server (UNIX only). Make sure you use the logfile option -l in conjunction with this mode and specify the -l option before any other options.
- **-D: U**se this to make the server put itself in the background and run as a daemon. If neither -i nor -D is given, the server starts in the foreground.
- **-install:** Use this to install the server as an NT service (Windows NT/2000/XP only). Control the server by going to the Services in the Control Panel.
- **-installa: U**se this to install and activate the server as an NT service (Windows NT/2000/XP only). Control the server by going to the Services in the Control Panel.
- **-remove:** Use this to remove the server from the NT services (Windows NT/2000/XP only).
- **-t minutes:** Idle session timeout, in minutes. Default is 60 minutes.
- **-k size:** Maximum record size/message size, in kilobytes. Default is 1024 KB (1 MB).
- **-d daemon:** Set name of daemon to be used in hosts access file.
- **listener-address** consists of an optional transport mode followed by a colon (:) followed by a listener address. The transport mode is either a file system socket unix, a SSL TCP/IP socket ssl, or a plain TCP/IP socket tcp (default).
- For TCP, an address has the form

hostname | IP-number [: portnumber]

ZEBRA SERVER ARCHITECTURE:

Fig 3: Represents the Zebra Server Architecture. In this architecture we have several Client devices like Flash, RTMP Clients, HTTP Clients, which will interact with Zebra Server using respective protocol.

Inside the Zebra Servers We have HTTP servers, JDBC or other Database Drivers to communicate to the database servers. Here we have master database server which in turn connected to multiple database servers. These Database servers are again in turn connected to other Zebra Server.



ADVANTAGES OF Z 39.50 PROTOCAL:

- Originally Z39.50 was designed to help with searching very large bibliographic databases like those of OCLC and the Library of Congress.
- Today Z39.50 is used for a wide range of library functions that involve database searching, from cataloging to inter-library loan to reference.
- With the rapid growth of the Internet, the Z39.50 standard has become widely accepted as a solution to the challenge of retrieving multimedia information including text, images, and digitized documents.
- Z39.50 is being used to access, for example, museum data, government information, and geospatial data.
- It can also be used to search the online databases and CD-ROMs that vendor develop according to a variety of design schemes.
- Without having to learn each system, users can search those databases with a single Z39.50 client, even though each uses a different hardware and software configuration, stores different types of data, and has different internal search logic.

CONCLUSION

So the main agenda of the paper is to have the effective usage of OPAC system which is integrated in distributed databases environment. Z39.50 Protocol is the Key component for us to have Distributed OPAC system which is being worked in Zebra Server.

ACKNOWLEDGEMENT

The Successful Completion of any task would be incomplete without expression of simple gratitude to the people who encouraged our work. The words are not enough to express the sense of gratitude towards everyone who directly or indirectly helped in this task.

I thankful to this Organization CMR Technical Campus, which provided good facilities to accomplish my work and would like to sincerely thank to our chairman Gopal Reddy Sir, Director Dr. A. Raji Reddy Sir, Dean Dr. Purna Chandra Rao Sir, and HOD CSE and faculty members for giving great support, valuable suggestions and guidance in every aspect of my work.

REFERENCES

1. Ansari, Mehtab Alam, and Amita. "Awareness and Use of OPACs in Five Delhi Libraries." The Electronic Library 26, no.1 (2008): 111-29.
2. Malak, Piotr. (2008). OPACs' Users' Interface – Do They Need Any Improvements? Discussion on Tools, Technology, and Methodology, available online at: http://edoc.hu-berlin.de/ conferences/bobcatsss2008/ [Accessed on: 25-11-2010].
3. Fayen E G. The online catalog: improving public access to library materials. Knowledge Industry Publications, Inc, 1983.
4. Norgard, Barbara A.et al. 1993. The online catalogue: from technical services to access service. Advances in Librarianship, Vol. 17: 111-148.
5. Client-Server Implications Of The Remote Database Search In The Library Catalogues Using The Z39.50 Standard Dr. Rer. Nat. Friedhelm Rump, Dr. Med. Alexandre Kossovoi
6. Z39.50 — A Primer on the Protocol - NISO Press.
7. The Bradford OPAC 2 -Managing and Displaying Retrievals from a Distributed Search in Z39.50 - F.H.Ayres, L.P.S.Nielsen, M.J.Ridley - University of Bradford
8. Baker and Lancaster (1991), The measurement and evaluation of library services, 102-932-875
9. DESIGNING AN ONLINE PUBLIC ACCESS CATALOGUE - Nathalie Nadia Mitev, Gillian M Venner and Stephen Walker
10. http://www.indexdata.com/zebra/doc/
11. http://manpages.ubuntu.com/manpages/lucid/man8/zebrasrv-2.0.8.html
12. http://dspace.fsktm.um.edu.my/bitstream/1812/185/2/2CHAPTER1.pdf
13. http://www.webpages.uidaho.edu/~mbolin/ruzegea.htm
14. Chen, Shu-Hsien Lai. 1991. A Study of Online Catalogue Searching Behavior of High School Students. Michigan: UMI Dissertation Service.