

An Approach to Distributed File System Through Java

Sangeeta Lalwani
Computer Science and Engineering Department
Amity University, Noida, India

Abstract— Despite of having ready implemented distributed file systems for individuals and businesses, still users face problems in using those system and it is still a big challenge for the scientists. The functionalities of a well functioned and an ideal distributed file system through Java platform would to resolve various difficulties that were being occurring in the past and to provide users with the desired functionalities such as file replication with updates, high scalability, cache consistency, provide shared access to the same set of files , naming and transparency of files , less storage space, effective communication, and safe as well as secure access. It would not be very complex to use and also would require minimal human administration. On the other hand Java Distributed File System also provides local independence and migration capabilities for data. The users are able to upload and download the files. It avoids problems such as security problems, bad file replication, poor scalability and slow performance. It also supports effective file replications and mobile computing.

Index Terms— File Replication, Naming, Transparency, Message Passing.

I. INTRODUCTION

These days, more resources are required by new applications than what are actually available on an inexpensive machine and problems are faced with the business processes by the organizations that no longer fit on a single cost effective machine. The security promise that the Distributed File System implemented through Java attempts to provide is confidentiality that is keeping the plaintext secret. This system also helps those applications that are heavily data intensive. As an example Google nowadays is not only a search engine but it supports other functionalities as well such as uploading pictures and videos, services such as Google Earth, Google Maps and also Google News applications. All these services

require huge storage space. Java distributed File system provides these services on a high level.

The system encrypts all the contents before leaving the client, and the key is also not shared with the unauthorized parties. As long as the key that is not shared with anybody till then the data that is there will not be known to anybody. Hence the system provides safe and secured access. The system uses Java as a platform as Java is the language for the advance programming in networks. Through the Java much advancement can be done and many problems can be resolved in systems that have been implemented in the past [1]. Java is the platform that is complete in itself and through it all the problems that have been occurred in the past such as file replication with updates, cache consistency problems, and semantics of sharing etc can be resolved.

Though distributed file system supports network wide sharing of files and devices but there are various issues such as file naming and transparency, cache consistency problems, sharing of file semantics, file replication with updates and lack of performance level. Java distributed file system resolves all these problems and that also with guarantee of integrity, confidentiality, access and availability and the most important one is the safe and secure access since the data is being distributed among the multiple clients, so if a hacker or criminal gets data from one system then he would not be getting from other systems due to which the hacker will not be getting the complete file and thus the software or the system provides security.

This is what a distributed file system implemented by using java provides us. Its main aim is to provide security to systems which is very essential for all. It is a system through which a hacker gets in trouble to collect all the files because the files are distributed and that also in encrypted form, so it is very confusing for a hacker or criminal to collect all the files since

he don't know the key and that is only shared between the client and server so he will not be knowing to get the key ever. This is the way through which a distributed file system and that implemented by using java can protect the files from hackers as well as from any unauthorized persons.

A file system is the method of permanent storage of files and organizing those files so that it is easy to find those files and access them. A distributed file system is conversely a network of file systems, which provides the storage of the files among the commodity of connected computers. A typical configuration of Java Distributed File System is a collection of workstations and mainframes connected by a Local Area Network. A Distributed File System is implemented as part of the operating system of each of the connected computers. One of the main drawbacks found in the prevailing file systems is the lack of the scalability. Security is quickly becoming a mandatory feature of the data storage systems. A secure storage system should protect the confidentiality and the integrity of the stored data. An unauthorized access may occur from other nodes on the network. Data that resides on the storage device, an attacker may physically access the storage device or send appropriate commands over the network. If the network is not secure, these commands may also be initiated by clients that are authorized to access other parts of the networked storage system. Therefore, some of the key features that need to be implemented are manipulation of files using framework, improved scalability, improved reliability, security, authentication and high performance. Either at home or business computers, there is certain amount of data which is stored that needs to be kept private as the data could be sensitive.

The organizations would not be able to run properly or would face a high loss in terms of cost and effort. Here comes the use of Java Distributed File System, which helps businesses and individuals in keeping the data or the information important to them in a safe and secured way. In today's real life there is certain amount of data which is stored and it needs to be kept private as the data could be sensitive. A small example could be a student's laptop containing his assignments or lecture notes. Loss of such data would result him to redo all the assignments and making the lecture notes from the start. Here comes the use of Java Distributed File System, which helps businesses and individuals in keeping the data or the information important to them in a safe and secured way that prevents it from hackers or criminals.

Today known doesn't know about one of the most popular and most used websites such as Google or YouTube, also the social networking phenomenon called facebook. There are many significant uses of these websites. Google nowadays doesn't only is a search engine but it supports other functionalities as well such as uploading pictures and videos. Other fascinations and attractive features introduced by Google

are Google Earth, Google Maps and also Google News application. All these services are heavily data intensive. Here comes a big question to be asked that how big companies like Google, YouTube or facebook do it. The answer to the question above is that these big phenomenon's use Distributed File System in order to provide their services on a high level. Java Distributed File System i.e., JDFS provides an environment where users can share the files among each other and the files are stored in a safe and secured way .

Java Distributed File System will be written in Java because Java is a powerful platform for the advancement of distributed file systems. Java is one of the most popular programming languages used these days. Java provides us with all the necessary features for the advance networking programming.

II. LITERATURE REVIEW

It is a system that supports the sharing of information through an intranet in the form of files. An extensive review was carried out in order to gain better understanding of the Java Distributed File System and the tools and the technical skills required for the implementation. The literature review which has been carried out covers the key points such as giving a brief description about Java Distributed File System, understanding the actual meaning of Java Distributed File System and discussing the main functionalities and the main areas of operations of such system. Through the review the concepts of some of the terms such as Protocol, File Replication and Namespace have been properly understood. This is done so in order to understand the problems in the systems that have been implemented in the past.

A file service which is very well designed provides access to the stored files at a server with the performance and reliability similar to that of the disk. In other words it can be said that a Distributed File System allows the users to store and access remote files like in the local way, but from any computer in the intranet. It has a capability of distributing the files from one computer to other computers. If it is to be said in scientific terms, a Distributed File System is a distributed implementation of the classical time sharing model which is constructed on a file system where it allows one or many users to share the files and storage resources of view, a Distributed File System is a system which is inherited from the File System and Distributed System. The Distributed Systems which centralized systems from a set of drivers physically dispersed computer systems which are inter connected by communications networks to permit the exchange of information. So therefore, a Distributed System is a resource set of processors that do not share memory or clock. Similar is the case with the File system, a Distributed File System provides the common files operations such as read, delete, rename, share and copy. These operations are not only used in

the certain local sites of Distributed File System but they are also used in other sites through authorization access. A Distributed File System can be seen as a polymorphic subclass which accommodates its processors or the resources distributed mechanism that is derived from Distributed System.

Java Distributed File System is a system that supports the sharing of information through an intranet in the form of files. A file service which is very well designed provides access to the stored files at a server with the performance and reliability similar to that of the disk. In other words it can be said that a Distributed File System allows the users to store and access remote files like in the local way, but from any computer in the intranet. It has a capability of distributing the files from one computer to other computers. If it is to be said in scientific terms, a Distributed File System is a distributed implementation of the classical time sharing model which is constructed on a file system where it allows one or many users to share the files and storage resources. In past systems such as Sun's Network File System and Hadoop Distributed File System have been widely used.

A. Sun's Network File System:

Sun's Network File System is mainly for the local area network. It provides a transparent access to the remote files to the clients. But today implementation of the network file system can be applied to almost all the operating systems even including the non-Unix systems. Some examples of such systems are MS-DOS, Microsoft or Macintosh [2]. Sun developed an open protocol that specified the exact message formats that server and clients would use to communicate. Using this approach different groups develop their own NFS servers. Companies such as IBM or NetApp are selling such NFS servers.

There are some of the advantages of Sun's NFS that comes along with some of the disadvantages as well. Here, are mentioned some. First talking about some of the advantages of Sun's NFS, some of which are Scalability, Access transparency, Hardware and OS heterogeneity and File replication, these are some of the main goals of the Sun's NFS [3]. Despite of having a lot of advantages, it also has some of the disadvantages. Mobile devices are not fully supported by NFS since migration transparency is not fully achieved. It doesn't support the file replication with updates. It does not enforce a single network; therefore each client sees the file system their way. Hence location transparency is affected. NFS uses the underlying UNIX file protection mechanism on servers for access checks. Each RPC request from a client conveys the identity of the user on whose behalf the request being made. The server temporarily assumes this identity, and the file accesses that occur while servicing the request are exactly checked as if the user had logged in directly to the

server. The standard UNIX protection mechanism using user, group and world mode bits used to specify protection policies on individual files and directories.

B. Hadoop Distributed File System

Hadoop Distributed File System provides framework from the analysis and transformation of very large data sets [5]. Executing application computations in parallel close to their data, computation across many hosts and portioning of data are some of the important characteristics of Hadoop. Regulation access to files by clients and management of the file system namespace is managed by the Hadoop Architecture components which are NameNode and a master server. The storage attached to the nodes is managed by Data Nodes. The file which is allowed for the storage by the file system namespace is internally split into one or more blocks which are stored in a set of Data Nodes. Operations such as renaming of files and directories, opening and closing of files are executed by the NameNode. The read and write request from the client is served by the Data Nodes. Data Nodes are also responsible for deletion, creation and replication upon instruction from the NameNode.

There are also some of the advantages as well as disadvantages of Hadoop Distributed File System. Starting with the advantages first, Hadoop Distributed File System is designed for storing large files with streaming data success patterns, running on clusters of commodity hardware. The files with sizes in megabytes, gigabytes or terabytes can be operated. The pattern on which the Hadoop Distributed File System is build is the data processing pattern which is writing-once, read-many time's pattern. Expensive hardware is not required to run it. It also has some disadvantages despite having a number of advantages. It does not cover all the requirements although it implements much of the functionality that our platform requires. It was originally designed for the I/O bounds that are those in which reading, writing and transferring data are the most time consuming operations; hence it is not conservative in CPU and memory usage. The technologies of Hadoop are not very well suited for mobile devices. Also the system can run more than on processes at the same time and can store massive amounts of data with a very high ability to fault tolerance.

III. RESEARCH METHODOLOGY

In the creation of the Java Distributed File System to provide an environment where users can share the files among each other and the files are stored in a safe and secured way, many technical challenges were faced. In order to create a great design of a Java Distributed File System, it is important to be familiar with the present Distributed File Systems and also should have a good understanding of server/ client based architecture. Since Java is the programming language is used

to build this Distributed File System, it is important to know the characteristic or java network and object-orientated characteristic.

The methodology for the implementation of Distributed File System through Java that is Java Distributed File System is as described. Firstly a file has to be shared. On a local machine in JDFS, a file or a directory operation can be executed. The use is for selecting directory to be shared folder of the shared resource on local machine. A file can then be uploaded by the distributed client on the server in the LAN network. The file once uploaded and sent to the server is encrypted by the server using Data Encryption Standard (DES) Algorithm. The uploaded file once gets encrypted is splitted into chunks before being sent to the other clients connected to the server. The server distributes those chunks of encrypted data among the clients. Once the request to download a file is sent by the client who uploaded the file on to the server, the server sends a random key to that client. With the help of that key, the original file can be downloaded. Once the key is selected by the user, the key merges those chunk sizes of data which were distributed among several clients back together and forms the original copy of that file. Once the data is emerged and the file has been re-created, but in encrypted format, the key decrypts the file before sending it to the client who requested for that file download. On a local machine, system feature is located which holds the responsibility for displaying the logical path of remote shared and local shared resources. A system feature is addressed in local machine which is capable of showing the permission of the remote shared file or directory at present.

IV. MA MAJOR APPLICATIONS OF JAVA DISTRIBUTED FILE SYSTEM

Java Distributed File System has various applications over other systems that have been implemented in the past such as Hadoop Distributed File System, Sun's Network File System etc. The Distributed File System implemented through Java is found to be advantageous from various perspectives.

A. Processing Application:

Mostly the Distributed File Systems are created on client/server architecture. The client is permitted to access the data/files or process the files which are stored on the server as if they were stored locally. Server, Client and Service are the three main terms. Service can be called as a function which is provided by the server to the clients. Server manages the service software which is running on a specific system/machine that is used to wait for the client in the local or remote computers. Client is the one that makes the request from other programs.

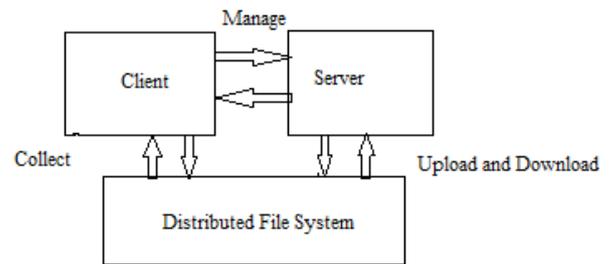


Fig. 1 Java Distributed File System

B. Naming and Transparency:

Talking about the Naming first, it can be called the mapping between the physical and logical objects. A logical data object is represented by the user that has a file name which can be called as the textual name but the data blocks are stored by the system physically which are the numerical identifiers which are mapped to disk blocks in the system-level. This mapping of the two levels provides an abstraction of the file so that the location and how the file is stored get hidden. Then in the abstraction the location of the file is added. The range of the naming mapping is represented as an address in a disk in conventional file system. Abstraction of files hides the details of how and where on the disk the file is actually stored. This is called Multiple Mapping. A set of locations of file replicas is returned by mapping if the client wishes to access a specific file name. The two types of transparency are Location Transparency and Location Independence. When the file name does not give any hint about the location of the storage of the file. This is called the Location Transparency. In other words it means that the path name of the file gives no hint to where the file is located. If the physical storage location of the file is changed but the name doesn't change, then it is called the Location Independence. If to determine that out of the two which is a stronger property, than the answer would be Location Independence is a stronger property than the Location Transparency. The reason is that Java Distributed File System can map the same file at different locations and at two different instances of time.

C. Synchronous and Asynchronous Message Passing:

It is a mode of communication that is enabled in Java Distributed File System. By the medium of this model, we have tried to explain that the objects or the processes can send and receive messages or the other processes. Firstly, talking about the synchronous message passing that is the one who sends the message and the receiver is the one who receives it and have to wait for the transfer to be completed. Buffering is not required in this form of message passing. On the other hand, in asynchronous message passing the sender and the receiver

doesn't have to wait for the file to be transferred completely to the receiver. There is advantage of asynchronous message passing over the synchronous message passing that the sender and the receiver without waiting for each other can overlap communication. Asynchronous communication is more efficient when there is low loss and low error rates over the transmission medium because data is not retransmitted and no time is spent in setting negotiating the connection parameters at the beginning of the transmission.

D. File Replication with Updates:

Java Distributed File System supports duplication of files with uniform updates that helps in improving the availability and performance of file systems and maintains consistency [7]. A system which is supported by replication, the data or the files may be split and represented by several different parts or the copies of the content of that file data are at different locations. File replication is an important concept in all the distributed systems. This splitting or distribution of the file into the copies of its content and creating a redundancy is quite vital. This is so because it helps for accessing wait and delay of the files and sharing of a number of files within the servers. A client can request for a while from any other different server in case a server copy isn't available to the client. Replication is a very powerful feature. It has a policy which makes sure that the files are accessible even if one or more parts or components of the Distributed File System doesn't work or fail for some reason. In order to maintain the consistency of the data, there are introduced different techniques or methodologies of File Replication. Lazy File Replication is one of the three techniques of File Replication. In this technique, the data or the files are copied automatically by the server once the files are written. When the files are sent to the server, then the remote files are brought up to date. It affects the consistency of the state of the file. Second technique is known as Explicit Replication where the client writes the files to more than one server and in order to implement this approach, it requires the support from the client. Group File Replication is the third technique where the requests are written and are sent to a group of servers in order to keep the replicas up to date. Using this technique, the client can read the state of the file from any replica.

E. Effective Communication

In a Distributed File System, a task is performed by a set of rules that controls the data/files communication. It is responsible for determining whether peer-to-peer or client-server architecture is used. The protocol if agreed-upon format data is transmitted determines the sender indication to finish sending a file, sender's indication to finish receiving the file, the mechanism of error checking and the method for data compression. In case of Java Distributed File System, the client opens the files; the distributed application request is translated into set of protocol messages. In order to fulfil the request by

the client, a response is generated by the server to each protocol. In other simpler words, the protocol provides the communication between the client and the server. Hence Java Distributed File System plays a very vital role in effective communication.

F. Safe and Secure Access

In today's real life there is certain amount of data which is stored and it needs to be kept private as the data could be sensitive. A small example could be a student's laptop containing his assignments or lecture notes. Loss of such data would result him to redo all the assignments and making the lecture notes from the start. Here comes the use of the Java Distributed File System which helps businesses and individuals in keeping the data or the information important to them in a safe and secured way [6]. This is because the system employs the concept of key that is only shared between client and server. The system encrypts all the contents before leaving the client, and the key is also not shared with the unauthorized parties. Hence, the system is protected thereby providing security.

G. More Storage Space

Today known doesn't know about one of the most popular and most used websites such as Google or YouTube, also the social networking phenomenon called facebook. There are many significant uses of these websites. Google nowadays doesn't only is a search engine but it supports other functionalities as well such as uploading pictures and videos. Other fascinations and attractive features introduced by Google are Google Earth, Google Maps and also Google News application [6]. All these services are heavily data intensive. Here comes a big question to be asked that how big companies like Google, YouTube or facebook do it. The answer to the question above is that these big phenomenon's use Java Distributed File System in order to provide their services on a high level. Through distributing the data or information the system provides security as well as provides more storage space for processing of application.

H. High Scalability

The functionalities of a well functioned and an ideal Java Distributed File System would to provide high scalability, provide shared access to the same set of files to all of its users, provide high performance, and provide more storage space and more user friendliness. It enables reliable file sharing across networks and throughput offered is higher. Also Java Distributed File System works well for audio and video applications. It has a pluggable file system, native support for raw tables and Java-like File API and it works well with no modifications. This field would not be very complex to use and also would require minimal human administration.

I. Uploading and Downloading of Files

A file can be uploaded by the distributed client on to the server in the local area network. Once the request to download a file is sent by the client who uploaded the file on to the server, the server sends a random key to that client. With the help of that key, the original file can be downloaded. Thus the file can easily be uploaded and downloaded by using Java through Distributed File Systems .So; through this the security can be established since only the client and server can share the key and nobody else. This is the application of cryptography through symmetric key towards distributed file systems. Thus Distributed File System enables the uploading and downloading of files through Java.

J. Location Independence

Naming is the mapping between the physical and logical objects. A logical data object is represented by the user that has a file name which can be called as the textual name but the data blocks are stored by the system physically which are the numerical identifiers which are mapped to disk blocks in the system-level. This mapping of the two levels provides an abstraction of the file so that the location and how the file is stored get hidden. The two types of transparency are Location Transparency and Location Independence [12]. When the file name does not give any hint about the location of the storage of the file. This is called the Location Transparency. In other words it means that the path name of the file gives no hint to where the file is located. If the physical storage location of the file is changed but the name doesn't change, then it is called the Location Independence which is supported by Java Distributed File System.

V. CONCLUSION

A file system is the method of permanent storage of files and organizing those files so that it is easy to find those files and access them. A distributed file system is conversely a network of file systems, which provides the storage of the files among the commodity of connected computers. As it is being described the applications for Java Distributed File System are successfully gained.

The software covers all the applications described above. The advantage is that the problems that were suffered by distributed file system such as migration transparency, file replication without updates, naming and transparency problems, sharing semantics problem, cache consistency problems, storage and scalability problems etc are resolved by Java Distributed File System. The system helps in resolving the problems that were occurring in previous systems such as Hadoop Distributed File System, Network File System etc.

There were systems in the past that suffered various problems but this software has concluded to resolve all the problems faced by them. Also the system enables learning abilities. The system also makes us quite familiar with server/client model, file replication techniques and advanced

programming in Java. The system is the most rewarding and beneficial one.

VI. FUTURE ASPECTS

Java Distributed File System also provides various applications that can be used in future. The emphasize on future work is as described .The Java Distributed File System was initially to operate in a network system, but it should also be suitable for multiple workgroups and networks. Hence in future it can be used to involve multiple networks so that data or information can be shared among multiple users [7]. Secondly, The Java Distributed File System might save the space of network by certain compression technique for multiple files transmission. So various algorithms can be implemented on that as compression techniques are the best for reducing the storage space.

The need of compression techniques [11] arises because reducing the memory space is important. Also performance evaluation of compression techniques can be done so as to find out which compression technique performs well among the various techniques implemented in reducing the storage space. The Java Distributed File System should make possible so that multiple user's responds to the request from a specific user in the network, if they have ability to meet the request. The Java Distributed File System will find the nearest and suitable machine to serve the user. Java Distributed File System should also use Data Mining techniques of Artificial Intelligence to capture the shared files by varied permission in a huge shared file warehouse.

REFERENCES

- [1] K.N.Honwadkar and T.R. Sontakke "Improvement in Capacity and Efficiency of Network Storage by Configuring Hard Disk Drives on Nodes of a LAN" International Journals of Computer Applications. 19 (3):15-21, April 2011.
- [2] Sun Microsystems Inc., NFS: Network File System.
- [3] R.Gibson and R.Van Meter, "Network Attached Storage Architecture" in communications of the ACM, vol.13, Nov.2000.
- [4] P.Leach, and D.Naik, "Common Internet File System (CIFS/I.O) Protocol Preliminary Draft," Internet-Draft Dec.1997.
- [5] M.Small, "Applied nonlinear time series analysis: applications in physics, medicine and economics," World Scientific, 2005.
- [6] Towards Secure and Dependable Storage Services in Cloud Computing Cong Wang, Student Member, IEEE, Qian Wang, Student Member , IEEE, Ning Cao, Student Member, IEEE, Kui Ren, Member, IEEE, and Wenjing Lou, Senior Member , IEEE-2011.
- [7] M.sugana and J.Bhavithra, "Maintaining Consistency during Data Replication in Grid Environment" International Journals of Computer Applications. 19 (3):15-21, May 2012.
- [8] R.S.Chang, P.H.Chen, 2007. Complete and Fragmented selection and retrieval in data grids, future generation Computer Systems 25(8):829-839.
- [9] D.Jhalani, D.Palsetia, "Adaptive cache coherence protocol using migratory shared data", 2007.
- [10] J.K.Archibald, "A cache coherence approach for large multiprocessor systems", In Proceedings of the 2nd International Conference on Supercomputing, France , July 1988, pp.pages 337.
- [11] Rafeeq Al-Hashemi and Israa Wahbi Kamal "A New Lossless Image Compression Technique Based on Bose,

Chandhuri and Hocquengham (BCH) Codes”, International Journal of Software Engineering and Its Applications, Vol. 5 No. 3, July, 2011.

- [12] T.Ristaniemi, J.Joutsensalo, “Independent component analysis with code information utilization in DS-CDMA signals separation”, Proceedings of the IEEE Global Telecommunication Conference, Brazil, 1999, pp.320-324.

Author Profile



Sangeeta Lalwani received the B.Tech degree from Moradabad Institute of Technology, Moradabad and pursuing M.Tech degree from Amity School of Engineering and Technology, Amity University, Uttar Pradesh, Noida, India in Computer Science and Engineering. Her areas of interest are Cloud Computing, Java Programming.