

Java Based Distributed File System

Sangeeta Lalwani and Subhash Chand Gupta

¹*M.Tech CS&E, Amity University, Noida, India.*

²*CS&E, Amity University, Noida, India.*

Abstract

There are distributed file systems that are being used in businesses and individuals but still users face various challenges and problems in using those systems. The distributed file system based on Java resolve the difficulties that were occurring in the past systems and provides various functionalities to users such as naming and transparency of files, high scalability, file replication with updates, cache consistency, provide shared access to the same set of files , less storage space, secure and safe access and effective communication. It would require minimal human administration and would not be very complex to use. The system uses Java as a platform as Java is the language for the advance programming in networks. Java distributed file system resolves all the problems occurring in systems being implemented in the past 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. Java Based Distributed File System not only avoid problems such as slow performance, bad file replication, security problems , poor scalability etc but also provides migration capabilities for data and location independence. Hence by using Java as a platform, performance of distributed file systems can be improved.

Keywords: GUI, Storyboards, Paper Prototypes, Transparency.

1. Introduction

These days, new applications require more resources as compared to those available on an inexpensive machine and business processes face various problems that pose difficulty on a machine that is cost effective. A permanent storage of files is called a

file system. The files in a file system are organized in a way that they can be easily accessed. However, a Distributed File System provides the file storage across the network. Java Distributed File System represents a typical configuration as it involves a collection of mainframes and workstations connected through Local Area Network.

Java Based Distributed File System attempts to provide the security promise in the form called confidentiality that is it maintains the secrecy of plaintext through encryption of files when uploaded and decrypting them when the request comes from the same client to download them. The system makes the file secure thereby providing safe access to the information contained in them since it encrypts the contents of file as soon as the client uploads the file through selecting the key and does not share the key with the unauthorized parties. The key is only shared between client and server due to which the data or information that the file contains will not be known to anybody. Hence in this way the system provides secured and safe access.

Though previous distributed file system such as Hadoop Distributed File System, Sun's Network File System supports sharing of devices and files across the network but then also there are various issues encountered such as lack of performance level, semantics of file sharing, naming and tolerance, fault occurrence, unavailability of files, lost of data, etc. Java based Distributed file System works to overcome these various problems faced in those systems and also guarantees access and availability, integrity, confidentiality and being the most important promise that the system offers is the secure and safe access to users and business processes.

Java is the platform to be used as it is complete in itself and provides us with all the features that is must for advance network programming. Through the Java we can overcome various problems occurred in distributed file systems used in the past[1] such as file replication without updations, cache inconsistency, ineffective communication, unsecure access and much advancement can also be done. Thus, Java based Distributed File System improves the performance level of Distributed File Systems.

2. Objective

The purpose is to develop a distributed file system, which will be deployed on network of commodity computers. The solution hides the complexities of saving and retrieving files using networked computers [2]. The main aim is to provide a Java Based Distributed File System as a new way or as a new challenge in solving the faults and problems that occur in the distributed systems being implemented. The system provides storage for secure files, high scalability, local independence and naming and transparency that is the migration capabilities for data.

The system avoids problems such as security problems, bad file replication, poor scalability, and slow performance. The users are able to upload and download the files safely and securely. The system provides us with a working prototype by integrating the techniques mentioned above with distributed file system. Finally, the evaluation is done and more future enhancements could be done such as use of more powerful and

advanced algorithms and techniques for effective file replication and mobile computing.

3. Research Methodology

An iterative process of storytelling, prototyping, sketching is involved in the user interface design as a means of design and analysis. The chosen methodology requires a need of constant shift between two types of synthesis, analysis and design activity. When analysis was done, the design was tested to check whether it is meeting goals for usability and during synthesis the design was shaped by drawing on fresh ideas born from user feedback and solutions to similar problems that have been occurring in the past through observation of existing systems and review of literature on existing systems.

The approaches that have been selected for interface designing are storyboard and paper prototyping approach. It includes the process of sketches, storytelling and prototyping as a means of analysis and design. At every stage the designer is able to quickly determine if potential users were able to understand the interface object. The techniques for interface design are described below with the rationale for use. In order to produce user friendly and usable interface following steps are involved:-

3.1 Storyboards

Before prototyping began, storyboards were used to capture more of the scope and flow the design proposal. Storyboards are much like paper prototypes but broader in scope and not generally intended for input from the users.

3.2 Paper Prototypes

Following the storyboarding exercise where the domains were explored, rough paper sketches of several alternatives were created. In some cases, none of the alternatives sketched fully met the needs of the site and game, and so a lot of quick sketches were done utilizing user feedback and ideas born from existing similar systems.

4. Benefits of Java Based Distributed File System

There are various benefits of Java Based Distributed File System as described below:

4.1 Transparency and Scalability

A file is abstracted through the mapping among logical and physical objects. This hides the file and its location called as file transparency [8]. Java based Distributed File System has the ability to support high scalability by providing the transparency of files that is not supported in distributed file systems already being implemented.

4.2 Updated Replications

The level of performance and the availability of file systems have been improved by Java Based Distributed File System through updated replication of files. The replication of files supports the users in case of data loss or when coincidentally some

failure occurs. In this way the system does not only provides security to their users but also a safe access through file replication with regular updations.

4.3 Adequate Message Passing

For effective transfer of messages [5] Java based Distributed File system uses a set of protocols that determines the architecture being employed that is either client-server or peer-peer. The protocols being used by the system is also responsible for determining whether the file has been sent, the acknowledgement has been received about the receipt of respective file, the technique for compressing of particular file and error checking mechanism.

4.4 Safe and Secure Access

Today everybody wants to maintain the privacy. Loss of personal and useful data matters for everybody [3]. Here is where Java based Distributed File System plays an important role. The system provides this benefit through distributing the files in a random manner that is important for the protection of files against hackers and also it provides safe access to their users or clients through file replication with regular updations. This is helpful in case of data loss.

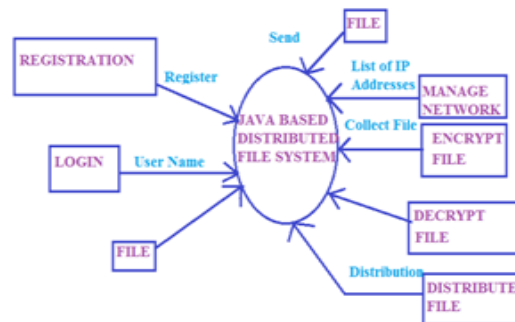


Fig. 1: Java Based Distributed File System.

4.5 Memory Usage

Today many does not know about some of the popular websites such as facebook and Google. These websites are useful for everybody. Google is not only a search engine but it also offers various striking features. All these services require large memory space. More storage space is provided to the files as the system distributes the data for efficient application processing.

5. Conclusion

Java Based Distributed File System is successfully developed. All the major requirements have been gathered and implemented. The system has been the most beneficial and rewarding one. The system has been a way or as a new challenge in solving the problems and the faults that occur in the present distributed file systems [11].

Many different concepts such as transparency, fault tolerance, scalability have been defined and solved using Java based Distributed File System. Alternatives for the methods for providing access to remote files and semantics of sharing have also been provided. Various implementations and design strategies have been demonstrated. New approaches have been provided in implemented systems based on assessment of the other existing systems. A critical examination of the product has been carried out [9] and also possible improvement or future works have been discussed.

References

- [1] G.L.Park, "Performance Evaluation of a List Scheduling In Distributed Memory Multiprocessor System" *International Journal of Future Generation Computer Systems* 20, 2004, 249-256.
- [2] J.Whitten, L.Bentley, and K.Dittman, *Systems Analysis and Design Methods*, 6th Ed, Mc-Graw-Hill, 2004.
- [3] R.J.T Morris Y.T.Wang Load Sharing in Distributed Systems. *IEEE Trans., Comput.* C-34(3), March.1985.
- [4] Neuman C.RFC 1510, "The Kerberos network authentication service (V5)", [S], 1993.
- [5] L. Roychoudhuri, and E. S. Al-Shaer, "Real-Time Packet Loss Prediction Based on End-to-end Delay Variation", *IEEE Transactions On Network And Service Management*, Vol. 2, No. 1, Third Quarter, P.P. 29-38, 2005.
- [6] Mirko Kämpf, and Jan W. Kantelhardt, "Hadoop.TS: Large-Scale Time-Series Processing" *International Journal of Computer Applications*, Vol. 74, 2013.
- [7] R. Gibson, and R. Van Meter, "Network Attached Storage Architecture," in *Communications of the ACM*, vol. 13, Nov. 2000.
- [8] Tankel, D., May 2010, Scalability of Hadoop Distributed File system, Yahoo developer work.
- [9] T.V Suresh Kumar, and S. Jagannatha, "Algorithm of Performance Prediction by Resource Sharing in Distributed Database" *International Journal of Computer Applications*, Vol. 66, 2013.
- [10] Danilo Ardagna, Sara Casolari, Michele Colajanni, Barbara Panicucci, "Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems" *J. Parallel Distrib. Comput.* 72 (2012) 796–808.
- [11] Pallab Dasgupta "Commit Protocols CS60002", *Distributed Systems*, Dept. of Computer Sc. & Engg, IIT Kharagpur.
- [12] D. S. Taubman and M. W. Marcellin, "JPEG2000", *Image Compression, Fundamentals, Standards, and Practice* by, Kluwer Academic Publishers, 2002.

