₹ 200 ISSN - 2249-555X
Volume : 1 Issue : 10 July 2012

Indian Journal Of Applied Research

Journal for All Subjects

www.ijar.in

Listed in International ISSN Directory, Paris.

ISSN - 2249-555X

Indian Journal of Applied Research

Journal for All Subjects

Editor,
Indian Journal Of Applied Research
8-A, Banans, Opp. SLU Girls College, New Congres Bhavan, Paldi,
Ahmedabad-380006, Gujarat, INDIA
Contact.: +91-9824097643 E-mail : editor@ijar.in

INDEX

Volume : 1 | Issue : 10 | July 2012 ISSN - 2249-555X

Research Paper Engineering

A survey on secure file synchronization in distributed system

* Chhaya Nayak ** Deepak Tomar

* Sanghvi Innovative Academy, Indore

**TIT, Bhopal

ABSTRACT

Distributed software systems have gained increasing popularity in recent years many systems have been designed to address the security problems of modern distributed file systems. However, these systems have suffered either from weak security, poor performance, or both. Distributed file systems are most widely used system today. A secure system is one that can keep information to be secret. Individuals, governments, and institutions such as banks, hospitals, and other commercial enterprises will keep their information secret on computer system if they can be absolutely certain of confidentiality. The problems of maintaining security are compounded because the sharing of secrets is generally desired but only in a tightly controlled manner.

Keywords : Distributed File Systems, Synchronization etc

1. Introduction

Distributed software systems have gained increasing popularity in recent years. Such type of applications are not just client/ server systems, but more and more complex systems consisting of a number of independent components running concurrently on different machines on top of a communication network. Today distributed systems are implemented using middleware technologies, such as Microsoft/COM, OMG/CORBA, or Java RMI. But there exist also a number of proprietary solutions to distributed systems, e.g. implementations of a H.323 video conferencing protocol and others. The Distributed File System (DFS) is used to build a hierarchical view of multiple file servers and shares on the network. Instead of having to think of a specific machine name for each set of files, the user will only have to remember one name; which will be the 'key' to a list of shares found on multiple servers on the network. Permanent Storage is a fundamental abstraction in computing. A permanent storage consists of a named set of objects that (1) come into existence by explicit creation, (2) are immune to temporary failures of the system, and (3) persist until explicitly destroyed.

Synchronization resembles file copying. But synchronization is more complex, and can handle several people changing source files at the same time. File synchronization (or syncing) in computing is the process of ensuring that computer files in two or more locations are updated via certain rules. In one-way file synchronization, also called mirroring, updated files are copied from a 'source' location to one or more 'target' locations, but no files are copied back to the source location. In two-way file synchronization, updated files are copied in both directions, usually with the purpose of keeping the two locations identical to each other.

2. File Systems Background

File systems are useful for long-term and persistent storage of information and are a way to organize information, usually in the form of files and a file hierarchy. A file consists of mapping a name to a block of information accessed as a logical group. Many ways exist to implement file systems and many file systems have been implemented, but relatively few of these are widely used.

File systems serve several main purposes:

1. Storing information
2. Retrieving information
3. Sharing information

There will be some general background about existing file systems.

Distributed File Systems

The most widespread distributed file system in use is Sun's Network File System (NFS) [32]. NFS is extremely mature, commercially supported, and widely available. A machine in NFS can both be a server and a client, as NFS does not have the notion of a dedicated server. The NFS protocol [19] is based on Sun's Remote Procedure Call (RPC).The NFS protocol is as stateless as possible to simplify server implementations

2.1. AFS

The Andrew File System (AFS) is a distributed networked file system which uses a set of trusted servers to present a homogeneous, location-transparent file name space to all the client workstations. It was developed by Carnegie Mellon University as part of the Andrew Project. AFS provides a strong form of security guarantee, based on access control lists at the level of entire directories.

2.2. Coda

Another distributed file system is Coda [13], also developed at Carnegie Mellon Coda is a distributed file system with its origin in AFS2. It has many features that are very desirable for network file systems. Coda caches files that may be needed when the system later becomes disconnected. It is freely available under a liberal license it has high performance through client side persistent caching

2.3 Sprite

Sprite [18] is a distributed file system developed at Berkeley, built on the assumption of diskless machines with large main memories. Sprite relies heavily on caching, with caching on both the servers and clients, and with caches stored in local memory. Sprite implements UNIX semantics for file sharing, in what the designers call concurrent write sharing

2.4 xFS

Traditionally, distributed systems rely on central servers to provide all of the services in a file system. These central servers become a single point of failure; so many file systems have the notion of replication. The xFS system is an example

of a server less network file system. It is a log-based system, allowing for checkpoints and the ability to rollback. xFS eliminates central server caching and uses client memory as a large and cooperative cache.

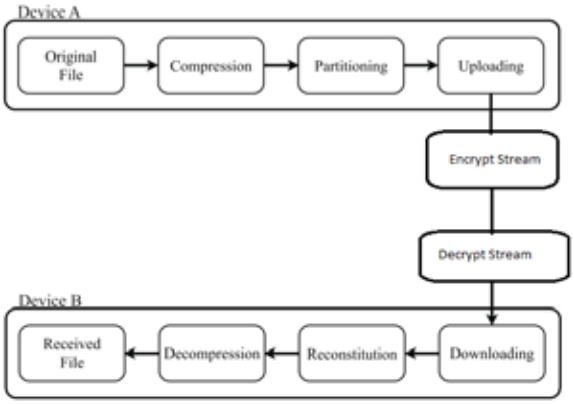
2.5 Ficus
The Ficus system, developed by Jerry Popek's group at UCLA. In Ficus, the model is one of a large-scale file system built of file servers that logically maintain replicas of a single file system image. Ficus is also known for work on stackable file systems, in which a single file system interface is used to provide access to a variety of types of file-like abstractions.

3. Characteristics of Distributed File Systems
Transparency
Transparency is usually built into distributed file systems, so that files accessed over the network can be treated the same as files on local disk by programs and users. The multiplicity and dispersion of servers and storage devices are thus made invisible. It is up to the network file system to locate the files and to arrange for the transport of the data.

Performance
A common performance measurement of a network file system is the amount of time needed to satisfy service requests. In conventional systems, this time consists of a disk-access time and a small amount of CPU-processing time. But in a network file system, a remote access has additional overhead due to the distributed structure. This includes the time to deliver the request to a server, the time to deliver the response to the client, and for each direction, a CPU overhead of running the communication protocol software. The performance of a network file system can be viewed as one dimension of its transparency; to be fully equivalent; it would need to be comparable to that of a local disk.

Concurrent file updates
Concurrency control becomes an issue when more than one person or client is accessing the same file and want to update it. Hence updates to the file from one client should not interfere with access and updates from other clients. Concurrency control or locking may either be built into the file system or provided by an add-on protocol.

4. Architecture to secure file synchronization in distributed System



4.Related Work
Synchronization issues in distributed systems have been and are widely discussed in the literature. In this we provide an overview of the related work that influenced the report on hand by in short explaining relevant issues. As the published work treats different aspects of synchronization_ it is structured here according to these aspects treated.

Many systems have been designed to address the security problems of modern distributed file systems.

However, these systems have suffered either from weak security, poor performance, or both. It is only recently that CPU performance has advanced to the point where strong cryptography can be done quickly with inexpensive processors. This allows its use on low-cost processors that can be associated with each disk in a distributed file system [7]. Most file systems include some measure of security. However, systems such as NFS [13] and xFS [18] pass most of their data in the clear, generally relying on relatively insecure networks and trusted hosts for data protection. Other systems, such as AFS [9] and NASD (Network Attached Secure Disk) [7] use third-party authentication such as Kerberos [19] to provide security. These systems provide stronger security by requiring users to obtain "tickets" from a third party. These systems are considerably stronger than those that rely upon simple authentication, but they still store files in the clear. SCARED [12] and the network-attached disks described by Gobioff [8] also use cryptographic techniques to authenticate users of remote network storage.

REFERENCES

[1] ASHIKALI M. HASAN, "IMPLEMENTING SECURITY LAYERS ON FILE SYSTEM," Journal of Theoretical and Applied Information Technology) John Rushby and Brian Randell, a Distributed Secure IEEE. | [2] Tran Doan Thanh1, Subaji Mohan1, Eunmi Choi1□, SangBum Kim2, Pilsung Kim2, "A Taxonomy and Survey on Distributed File Systems," Fourth International Conference on Networked Computing and Advanced Information Management | [3] ELIEZER LEVY and ABRAHAM SILBERSCHATZ "Distributed File Systems: Concepts and Examples, ACM Computing Surveys, Vol. 22, No. 4, December 1990 | [4] Demissie B. Aredo and Sule Yildirim, "SECURITY ISSUES IN ADAPTIVE DISTRIBUTED SYSTEMS," KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000. (technical report style) | [5] Ethan Miller Darrell Long William Freeman Benjamin Reed, "Strong Security for Distributed File Systems," 2001 IEEE | [6] M. SatyanarayananA Survey of Distributed File Systems Scott | [7] A. Banachowski□, Zachary N. J. Peterson†, Ethan L. Miller‡and Scott A. BrandtIntra-file Security for a Distributed File System. Proceedings of the 10th Goddard Conference on Mass Storage Systems and Technologies | [8] A Methodology to Assess Synchronization Algorithms for Distributed Applications, Christina Class, Burkhard Stiller Computer Engineering and Networks Laboratory (TIK) Swiss Federal Institute of Technology, ETH Z¨urich Gloriastr. 35, 8092 Z¨urich, Switzerland fclassjstillerg@tik.ee.ethz.ch | [9] A Distributed Framework for Integrated Software Process and Deployment Support, July 1998 TIK-Report No. 52, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology | [10] James J. Kistler and M. Satyanarayanan. Disconnected operation inthe coda file system. ACM Transactions on Computer Systems,10:3{25, Feb 1992. | [11] Synchronization Issues in Distributed Applications Definitions, Problems, and Quality of Synchronization, Christina Class December 5 1997 | [12] Intra- and Inter-Stream Synchronization for Stored Multimedia Streams, Ernst Biersack, Werner Geyer, Christoph Bernhardt Institut Eurécom 2229 Route des Crêtes, 06904 Sophia Antipolis, FRANCE {erbi,geyer,bernhard}@eurecom.fr | [13] A Synchronization Scheme for Stored Multimedia Streams, In B. Butscher, E. Moeller, and H. Pusch, editors, Interactive Distributed Multimedia Systems and Services (European Workshop IDMS'96, Berlin, Germany), volume 1045 of LNCS, pages 277–295. Springer Verlag, Heidelberg, Germany, March 1996 | [14] A D A P T I V E SCALABLE INTERNET STREAMING by DMITRI LOGUINOV, A dissertation submitted to the Graduate School Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York 2002 | [15] QoS-Based Synchronization of Multimedia Document Streams M. Farrukh Khan*, Halima Ghafoor**, and Raymond Paul+ *Dept. of Computer Engineering, KFUPM, Dehran, 31261 Saudi Arabia **School of Electrical and Computer Engineering, Purdue University W. Lafayette, Indiana 47907 DoD, OASD/C3I, Washington DC, 20453 | [16] Operating System Issues for Continuous Media Henning Schulzrinne May 15, 1995 | [17] Enforcing multipoint multimedia synchronisation in video conferencing applications Philippe Owezarski LAAS-CNRS 7, Avenue du Colonel Roche F-31077 Toulouse cedex | [18] T. Anderson, M. Dahlin, J. Neefe, D. Patterson, D.Roselli, and R. Wang, "Serverless Network File Systems,"ACM Transactions on Computer Systems , Feb. 1996, pages 41-79. | [19] B. Neuman and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," IEEE | Communications Magazine 32 (9), September 1994, pages 33-38.

Sara Publishing Academy
Indian Journal Of Applied Research
Journal for All Subjects

Editor,
Indian Journal Of Applied Research
8-A, Banans, Opp. SLU Girls College,
New Congres Bhavan, Paldi, Ahmedabad-380006.
Contact.: +91-9824097643 E-mail : editor@ijar.in

Printed at Unique Offset, Novatsing Rupam Estate, Opp. Abhay Estate, Tavdipura, Shahibaug, Ahmedabad