International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 11 ǁ November 2022 ǁ PP. 229-233

File Synchronization within Different Platforms Safeguarding Confidentiality on Cloud

[1]Medha Sinha
[*1]*Medha Sinha, stars.med007@gmail.com, SAP Labs, Bengaluru India*

Abstract
We coexist with a digital life in the information and technological age of today. We nearly exclusively keep digital files in storage. Typically, we make a few small adjustments to the files we create before copying them from one device to another. Every time a person travels, they take photos and films which have some emotional significance as well. These moments are typically recorded on portable devices like mobile phones, tablets, or digital cameras, all of which have a finite amount of internal capacity. Therefore, information must periodically be backed up on a device with more capacity. This is a difficult task. There aren't many file synchronization applications designed specifically to address such problems. However, the most of them are command-line tools, and using them requires some familiarity with programming. Therefore, we have developed a solution that handles the same problems. New files are instantly synchronised amongst all of your devices when they are all linked to the same network. As a result, the user can enjoy life and spend the majority of their time in the real world rather than spending it on repetitive tasks
Keywords:*file synchronization, distributed storage, privacy issues with cloud.*

Date of Submission: 07-11-2022 Date of acceptance: 21-11-2022

I. INTRODUCTION

The project's goal is to offer a file synchronization mechanism that is quicker and uses less bandwidth than its rivals merely in terms of local networks (rather than involving internet so that client privacy is maintained and also to reduce internet traffic). The proposed application leverages distributed file transfers for file synchronization and a two-phase scanning process for potential modifications. It generates a list of the files that need to be synchronized in the first phase by scanning for changes in file and directory structures. The actual file transfer happens in the second stage. The system intelligently copies files, updating only the portions of the file that have been changed. Consequently, efficiently using time and network resources for scanning the updates and synchronizing the changed files.

1.1.1 LITERATURE SURVEY

Title of paper	Description
FileSync/NDN: Peer-to-Peer File Sync over Named Data Networking - Jared Lindblom, Ming-Chun Huang, Jeff Burke and Lixia Zhang. NDN Technical Report NDN-0012, March 2013	Solves the file synchronization problems that can occur when utilising paid services like DropBox or P2P file synchronization. The application aims to utilise intrinsic content signatures and NDN's efficient multicast dissemination. It also emphasises the use of the CCNx Synchronization Protocol to simply incorporate storage at each node while maintaining the consistency of the synchronised files.Compared to IP-based peer-to-peer methods, the application relies on NDN intrinsic content caching and multicast to decrease traffic.
The rsync algorithm Andrew Tridgell and Paul Mackerras Department of Computer Science Australian National University Canberra, ACT 0200, Australia	Provides a method for updating a file on one system to match a file on another machine, assuming particular requirements, such as that the two machines are linked via a low bandwidth, high latency, bidirectional communications channel. Only those sections of the source file that cannot be matched this manner are sent, according to the algorithm, which finds parts of the source file that are identical to specific parts of the destination file.
Pre-Processing Directory Structure for Improved RSYNC Transfer Performance – AlirezaGhobadi, Ehsan Haji Mahdizadeh, Yong Lee Kee, Li Kok Wei, MohamadHosseinGhods ISBN 978-89-5519-155-4 Feb. 13-16, 2011 ICACT2011	Gives an answer to questions like "given two versions of files inside folders on different machines, call outdated hierarchy and a current one, how can we update the outdated version with a minimum communication cost, by exploiting the significant similarity between the versions" and deals with the issue of

	maintaining large hierarchy folder replicated in a distributed environment.
An Algebraic Approach to File Synchronization - Norman Ramsey and ElodCsirmaz Harvard Computer Science Group Technical Report TR-05-01	Contribute an algebra of file system operations, along with algebraic laws that are helpful for reasoning about file synchronization which is used for implementing synchronizers. Also, the paper explains conflict detection and resolution in terms of algebra. These rules show that their technique detects essentially the same conflicts as the state-based technique of Balasubramaniam and Pierce [1998]
Network investigation methodology for BitTorrent Sync: A Peer-toPeer based file synchronization service - Mark Scanlon, Jason Farina, M-TaharKechadi DOI: 10.1016/j.cose.2015.05.003	Develop a framework for protocol inspection and analysis to help manage data flow across security perimeters. Users are becoming more and more dependent on gadgets that consume and produce data in ever-increasing numbers, thus high availability is no longer merely a business continuity issue. Think of BitTorrent Sync, a cloudless synchronization tool that offers data availability and redundancy, as a potential solution.
Synchronizing Files from a Large Number of Insertions and Deletions - Frederic Sala, Clayton Schoeny, Nicolas Bitouz, Lara Dolecek DOI: 10.1109/TCOMM.2016.2552175	Imparts the efficient algorithm to synchronize between different version of files. "Interactive synchronization protocol" which is based on synchronization algorithm, but the synchronization algorithm here is modified in 3 ways. The algorithm outperformed the conventional rsync algorithm.
A practical framework for efficient file synchronization - Nicolas Bitouze, Frederic Sala, S. M. SadeghTabatabaeiYazdi, Lara Dolecek DOI-10.1109/Allerton.2013.6736664	Discussed the practical evaluation of the order-optimal synchronization algorithm. This protocol is fundamentally based on the notion of interactive communication. This algorithm improves in rounds of communication and total number of bits exchanged between users in the worst-case.
Rsync and Rdiff implementation on Moodle's backup and restore feature for course synchronization over the network - FajarPurnama, Tsuyoshi Usagawa, Royyana M Ijtihadie, Linawati DOI: 10.1109/TENCONSpring.2016.7519372	Reduces the amount of network traffic caused by data synchronization. This technique works with all MOODLE (Modular Object Oriented Dynamic Learning) versions and allows the user to manage the synchronization data.
A Server Friendly File Synchronization Mechanism for Cloud Storage - Chao Yang, Ye Tian, Di Ma, ShuoShen, Wei Mao DOI: 10.1109/GreenCom-iThings-CPSCom.2013.70	Focuses on minimizing the energy requirement of cloud by optimizing calculations required by rsync algorithm on cloud.
Efficient interactive algorithms for file synchronization under general edits - RamjiVenkataramanan, VasukiNarasimhaSwamy, KannanRamchandran DOI: 10.1109/Allerton.2013.6736666	Focuses on fixing the problem of missynchronized files from non-local data sources that had previously been synced but had undergone alterations (insertion, deletion, and substitution). In order to synchronise Y to within a target Hamming distance of X, it also provides a 3-factored extension to this approach that use a Hamming-distance estimator.
CloudSync: Multi-nodes Directory Synchronization - Qiang Li, Ligu Zhu, Wenqian Shang, SaifengZeng DOI: 10.1109/ICICEE.2012.386	Proposes the use of the multi-node directory synchronization programme cloud Sync in WAN environments, with an emphasis on optimistic replication through the use of two alternative change detection techniques.

1.1.2 Project Scope

Putting in place a file synchronization tool that can operate on a variety of platforms or operating systems while protecting client privacy, as well as one that can get around the drawbacks of cloud-based synchronization services and services that rely on the internet for file synchronization. So eliminating the limits of cloud storage, including the cost of additional storage and various server difficulties.

1.2 System Design

The following figure gives a brief idea about the system architecture.

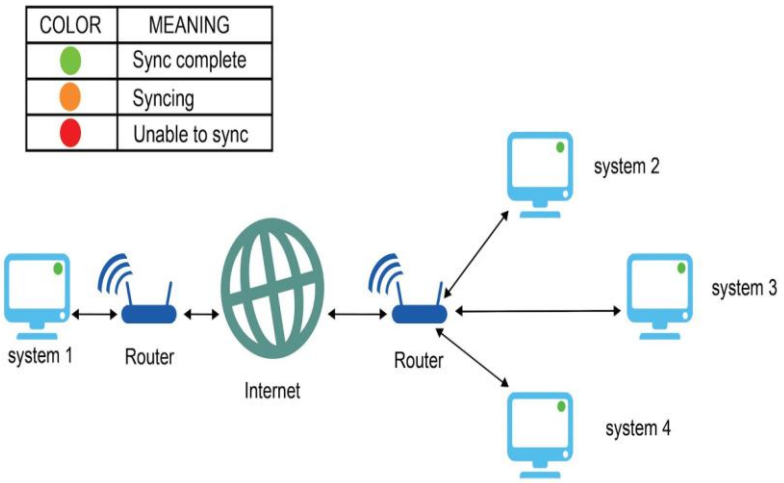

Figure 1:System Architecture

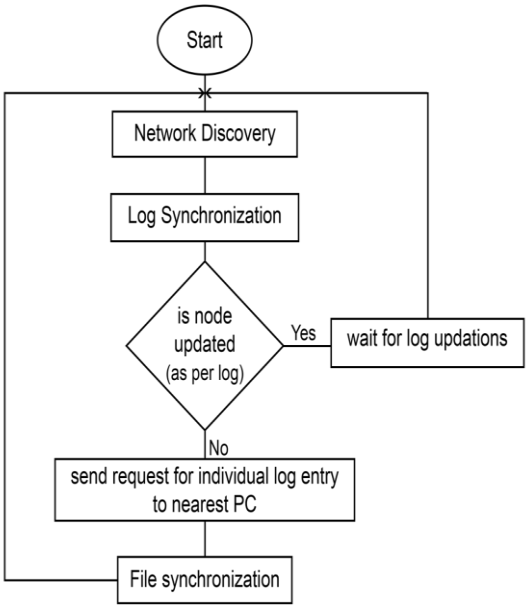

Figure 2:Algorithm

1.2.2 Main Function

1.2.2.1 Network discovery

The tool detects and marks as present nodes in the system any active nodes in the network that have the tool installed on them. Later procedures like log and file synchronisation will only employ these systems. Any node in the network at any given time can operate as a source for all the current nodes.

1.2.2.2 Log synchronization

Prior to actual file synchronisation beginning, a particular file is synced between all active nodes upon network discovery. This unique file keeps track of each and every network activity to ensure consistency between the synced data. According to the differences in its directory structure and log file, each node sends requests across the network.

1.2.2.3 Nearest node discovery

A node will send a request to the closest node in the network to receive each unsynchronized file from the network. The network delay is used to determine which node is closest. Each node maintains a list of the nodes that are closest to it, and each node updates the list on a regular basis.

1.2.2.4 File synchronization

Only when there is a hash mismatch or differences in the directory tree are file actions carried out for synchronisation purposes. In order to benefit from network resources, files are simultaneously copied from multiple nodes. A node may simultaneously request distinct files from two different nodes. Data is securely sent between nodes during file transfers using the SFTP protocol. Additionally, encryption can be applied to the programme to increase security.

1.2.3 Mathematical Model

Input:
I = { machineID[1], fileID[1][], folderID[1][], machineID[2,…n]….. }
Output:
O = { machineID[1,…,n].syncStatus }
Success
S =machineID[1,..,n].syncStatus == 1
Failure
F =machineID[1,..,n].syncStatus == 0

II. CONCLUSION

The results obtained are as discussed belowIn this paper, we suggest a file synchronisation system that can fix problems with existing products on the market as well as classic file synchronisation techniques. The traditional approach has drawbacks like a bottleneck, duplicate files, and slower transfer rates. As the number of sources each cycle rises, the suggested method can deliver substantially faster transfer rates. The system is made more efficient and time is saved by calculating latency characteristics during data transfer between two nodes. By maintaining a straightforward log entry for each network action, file duplication is prevented. This ensures that the data is consistent and keeps every node in the network updated.

REFERENCES

[1]. Jared Lindblom, Ming-Chun Huang, Jeff Burke and LixiaZhang, ”FileSync/NDN: Peer-to-Peer File Sync over Named Data Networking” in NDN Technical Report NDN-0012, March 2013

[2]. Andrew Tridgell and Paul Mackerras, ”The rsync algorithm” at Department of Computer Science Australian National University Canberra, ACT 0200, Australia

[3]. AlirezaGhobadi, Ehsan Haji Mahdizadeh, Yong Lee Kee, Li Kok Wei, MohamadHosseinGhods, ”Pre-Processing Directory Structure For Improved RSYNC Transfer Performance” in ICACT2011 Feb. 13-16, 2011

[4]. Norman Ramsey and ElodCsirmaz, ”An Algebraic Approach to File Synchronization” at Harvard Computer Science Group Technical Report TR-05-01

[5]. Mark Scanlon, Jason Farina, M-TaharKechadi, ”Network investigation methodology for BitTorrent Sync: A Peer-to-Peer based file synchronisation service” in Computers and Security, 2015

[6]. Frederic Sala, Clayton Schoeny, Nicolas Bitouz, Lara Dolecek, ”Synchronizing Files From a Large Number of Insertions and Deletions” in IEEE Transactions on Communications 6, June 2016

[7]. Nicolas Bitouze,FredericSala, S. M. SadeghTabatabaeiYazdi, Lara Dolecek, ”practical framework for effcient file synchronization” in Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference, 2-4 Oct. 2013

[8]. GuiPingWang, ShuYu Chen, MingWei Lin, XiaoWei Liu, ”SBBS: A sliding blocking algorithm with backtracking sub-blocks for duplicate data detection” in Expert Systems with Applications, April 2014

[9]. A Server Friendly File Synchronization Mechanism for Cloud Storage- Chao Yang, Ye Tian, Di Ma, ShuoShen, Wei MaoDOI: 10.1109/GreenCom-iThings-CPSCom.2013.70

[10]. Rsync and Rdiff implementation on Moodle's backup and restore feature for course synchronization over the network- FajarPurnama, Tsuyoshi Usagawa, Royyana M Ijtihadie, LinawatiDOI: 10.1109/TENCONSpring.2016.7519372

[11]. Efficient interactive algorithms for file synchronization under general editsRamjiVenkataramanan, VasukiNarasimhaSwamy, KannanRamchandranDOI: 10.1109/Allerton.2013.6736666

[12]. Synchronization and Deduplication in Coded Distributed Storage Networks- Salim El Rouayheb, SreechakraGoparaju, Han Mao Kiah, OlgicaMilenkovicDOI: 10.1109/TNET.2015.2502274

[13]. Synchronizing Files from a Large Number of Insertions and Deletions- Frederic Sala, Clayton Schoeny, Nicolas Bitouzé, Lara DolecekDOI: 10.1109/TCOMM.2016.2552175

[14]. MetaSync: Coordinating Storage across Multiple File Synchronization Services- Seungyeop Han, HaichenShen, Taesoo Kim, Arvind Krishnamurthy, Thomas Anderson, David WetherallDOI: 10.1109/MIC.2016.44
[15]. SyncDS: A digital safe based file synchronization approach- MayssaJemel, MouniraMsahli, Ahmed Serhrouchni DOI: 10.1109/CCNC.2016.7444796
[16]. BatchFS: Scaling the File System Control Plane with Client-Funded Metadata Servers- Qing Zheng, Kai Ren, Garth Gibson DOI: 10.1109/PDSW.2014.7
[17]. Data synchronization protocol in mobile computing environment using SyncML and Huffman coding- Jiao-Long Li, Jian-Ping Li DOI: 10.1109/ICWAMTIP.2012.6413489
[18]. CloudSync: Multi-nodes Directory Synchronization- Qiang Li, Ligu Zhu, Wenqian Shang, SaifengZeng DOI: 10.1109/ICICEE.2012.386