A Survey on Synchronization of Files with Effective Transmission Framework

[1]Puja Shitole, [2]Ashwini Gore, [3]Archana Jadhav, [4]Meghana Surve, [5]Neha Mule
[1,2,4,5]Student, [3]Assistant Professor

Abstract— In recent years, there has been an explosion of interest in mining time series databases. As with most computer science problems, representation of the data is the key to efficient and effective solutions. One of the most commonly used representations is piecewise linear approximation. This representation has been used by various researchers to support clustering, classification, indexing and data sharing approach of time series data. A variety of algorithms have been proposed to obtain this representation, with several algorithms having been independently rediscovered several times.
In this project, we undertake the first extensive review and empirical comparison of all proposed techniques. We show that all these algorithms have fatal flaws from a data sharing perspective. We introduce a novel algorithm that we empirically show to be superior to all others in the literature.

Index Terms—Two-way communication, deletion channel,insertions and deletions, synchronization, edits, coding for synchronization, rsync, practical protocols.

I. INTRODUCTION

A system model is the conceptual model that describes and represents a system. A system comprises multiple views such as planning, requirement (analysis), design, implementation, deployment, structure, behavior, input data, and output data views. A system model is required to describe and represent all these multiple views. The system model describes and represents the multiple views possibly using two different approaches. The first one is the non-architectural approach and the second one is the architectural approach. Popular utility designed to copy files faster and more reliably, providing the user with many features. Tera Copy uses dynamically adjusted buffers to reduce seek times. It can resume broken file transfer, skip bad files during the copying process. Quickly access your favorite folders and files. Jump to any deeply nested folder in a double mouse click. Direct Folder automatically resizes every standard file dialog, so that you can see a larger number of files.

In this work, system deal with efficient load balancing between the different resource nodes that process the client tasks, in a secure way as well as the elimination of possible single point of failure in a semi centralized load balancing architecture. To ensure that the two fundamentals i.e. co-ordination (the right things) and synchronization (the right time) of the processes will be implemented we use synchronization algorithms. With such synchronization algorithms security will be provided to the data while transmission. This leads to less time consumption as the tasks are been executed concurrently. Our System is a mixture of distribution model for P2P network. Data Sharing System, which has attracted the largest number of users, is the main application scheme for P2P file sharing. In broadcasting network, a single file is shared by many users. The global data(files) to be transmitted is divided into Chunks (i.e. breaking the files into pieces) using chunking mechanism. The chunks can be of fixed size or variable size. All the parts connect to a central node called tracker to get a list of parts. Once all the distributed pieces are obtained at single location then whole data is successfully broadcasted to destination path.

II. LITERATURE SURVEY

In paper [1], the problem of synchronizing two files X and Y at two distant nodes A and B that are connected through a two-way communication channel. This file Y is at node B is obtained from file X at node A by inserting and deleting a small fraction of symbols in X. consider the case where X is a non-binary non-uniform string, and deletions and insertions happen uniformly with rates β_d and β_I, respectively. A synchronization protocol between node A and node B that needs to transmit $O(q/H_2(\beta_d+\beta_I)n \log 1/\beta_d+\beta_I)$ bits (where n is the length of X, q is the alphabet size and H_2 is the collision entropy of X) and reconstructs X at node B with error probability exponentially low in n.

Modern Personal Digital Assistant (PDA) architectures often utilize a wholesale data transfer protocol known as “slow sync” for synchronizing PDAs with Personal Computers (PCs). This is inefficient with respect to bandwidth usage, latency, and energy consumption, since the PDA and PC typically share many common records. We propose, analyze, and implement a novel PDA synchronization scheme (CPI sync) predicated upon recent information-theoretic research. The salient property of this scheme is that its communication complexity depends on the number of differences between the PDA and PC, and is essentially independent of the overall number of records. Moreover, our implementation shows that the computational complexity and energy consumption of CPI sync is practical, and that the overall latency is typically much smaller than that of slow sync. [2]

Racetrack memory is a non-volatile memory engineered to provide both high density and low latency, that is subject to synchronization or shift errors. This paper describes a fast coding solution, in which “delimiter bits” assisting identifying the type of shift error, and easily implementable graph-based codes are used to correct the error, once identified. A code that is able to detect and correct double shift errors is described in detail. [3]

The purpose of this survey is to describe recent progress in the study of the binary deletion channel and related channels with synchronization errors, including a clear description of open problems in this area, with the hope of spurring further research. As

an example, while the capacity of the binary symmetric error channel and the binary erasure channel have been known since Shannon, we still do not have a closed-form description of the capacity of the binary deletion channel. We highlight a recent result that shows that the capacity is at least $(1-p)/9$ when each bit is deleted independently with fixed probability p. [4]

This paper constructs a non-binary code correcting a single-burst of insertions or deletions. This paper also proposes a decoding algorithm of this code and evaluates a lower bound of the cardinality of this code. Moreover, we evaluate an asymptotic upper bound on the cardinality of codes which can correct a single burst of insertions or deletions. In this paper, we have constructed a non-binary-burst insertion/deletion correcting code and presented a decoding algorithm for the code. We also have derived a lower bound on the cardinality of the proposed code and an asymptotic upper bound on the cardinality of non-binary burst deletion correcting codes. Our future works are construction of non-binary codes which correct a deletion burst of at most consecutive symbols and deriving non-asymptotic upper bound on the maximum cardinality of any non-binary burst deletion correcting code. [5]

We investigate binary, number-theoretic, bit insertion/deletion correcting codes as pioneered by Lowenstein (1965, 1966, 1989). The weight spectra and Hamming distance properties of single insertion/deletion error-correcting codes are analyzed. These relationships are then extended to investigate codes that can correct multiple random insertions and deletions. From these relationships, new bounds are derived and a general construction for multiple insertion/deletion correcting codes is proposed and evaluated. In some characteristics of binary codes that can correct insertions, deletions, and additive errors are given, as well as two single-error correcting classes of codes. For a code to be able to perform such correction, it was presupposed that the boundaries, i.e., the beginning and end of the codeword being decoded, are known. [6]

III.SYSTEM ARCHITECTURE

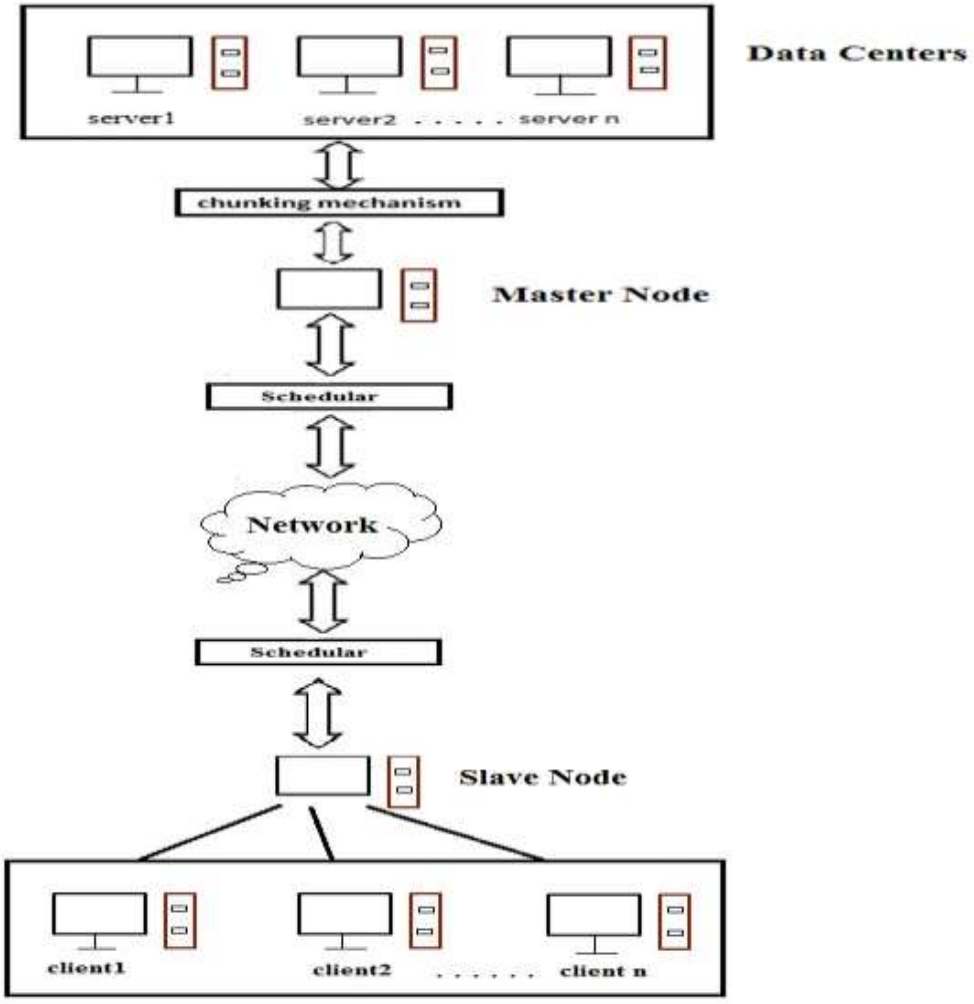

Fig.1: Architecture

Chunking is a process to partition entire file into small pieces of chunks. For any data de-duplication system, chunking is the most time-consuming processes since it has to traverse entire file without any exception. The process time of chunking totally depends on how the chunking algorithms break a file. Moreover, the smaller the size of a chunk has, the better result a de-duplication system has. Increasing the number of chunks, however, results in increasing the processing time.

IV. IMPLEMENTATION METHEDOLOGY

❖ Mathematical Modeling:

S = {I, O, F, P, DD, NDD, Memory shared}

Where,

I: set of inputs,

O: It is the set outputs,

DD: Deterministic data i.e. output data is true,

NDD: Non-deterministic data,

Memory shared: memory blocks that is used for execution

I = {Collection of different kinds of data}

O = {O1; O2; O3}

O1= Profit ratio for computational unit
O2= Profit ratio for memory storage
O3= Profit ratio for memory and bandwidth

F= {F1, F2, F3}

F1 = Data chunk object
F2 = Profit ratio object
F3 = Remote location object

P = {P1, P2}

P1 = Charging cost for inelastic task
P2 = Security mechanism

DD = {output data is true}

NDD = {not fixed}

Memory shared = {memory blocks that is used for execution}

❖ FEATURES

SR.NO	FEATURE	DESCRIPTION
1	PLATFORM INDEPENDENT	Platform independent means that the code remains the same irrespective of the platform involved. Our software system is platform independent which means it will run on any system like Windows, Ubuntu, MAC, etc.
2	LOAD BALANCING	Load balancing distributes workloads across multiple computing resources, such as computers, a computer cluster ,network links, central processing unit or disk drives .load balancing aims to optimize resource use, maximum throughput, minimize response time, and avoid overload of single resource. Our software has a load balancing feature which increases reliability and availability through redundancy.
3	AUDIBLE OPERATIONS	Unlike other software's we cannot access the system if we are not in front of it. we won't get any idea if errors or other processing happening in the system. But in our system, we need not need to be in front of the system but just in the surrounding of the system. The audible operation feature in our application gives audible sound for every processing going on in the system.
4	USER FRIENDLY APPERAENCE	The other software's means those complicated software's have a GUI which is very complicated. Any third person which has no idea about the software cannot

		access the application efficiently, but our application has such an easy GUI that if any third person who is the legal user has no idea about the software can use it easy at his fingertips.
5	FILE TRANSFER	Our application can transfer any kind of file like .exe file, .doc file, .mp3 file, mp4 file and many more. We can also encode and decode the file by just clicking on the Encode and Decode checkboxes. Other systems take lots of time to transfer the file. But Our proposed System transfers file within a fraction of seconds.
6	SPACE UTILIZATION	Space Utilization is an important approach in any computing system. But in other applications space utilization management is not proper which leads to lack of space. Hence due to this we cannot store our desired files, folder, applications, etc. But there is proper space utilization in our proposed system which helps in storing data in a proper way.
7	DIRECTORY TRANSFER	We can transfer the directory in normal systems but we cannot give a security mechanism to it simultaneously during transfer is going on. This is a very time-consuming process. But our system provide mechanism such that we can transfer the directory along with giving security mechanism to it at the same time. This will help in reducing the time.
8	TIME UTILIZATION	Other systems take much more time to transfer any data. But Proposed System transfers data within a fraction of seconds. In advance, it provides the mechanism of transfer of file and security mechanism simultaneously which is very useful approach and time consuming

Table.1: Features

V. CONCLUSION

In this paper, we consider the issue for integrity checking of data sharing approaches with a remote server and will propose an efficient data securely sharing which will be specifically designed to handle a number of deletions linear in the length of the file for different operations where space utilization, security mechanism, splitting and concatenation operations are performed on file information. Our System also consist of verification methodology for integrity for the files stored on remote server, and reduces the storage costs and computation costs of the data. The presented scheme design is based on new lightweight hybrid data structure to support dynamic operations on blocks which incurs minimum computation costs by decreasing the number of nodes shifting. Using our new data structure, the data owner can perform insert, modify or delete operations on file blocks with high efficiency.

VI. REFERENCES

[1] Bart Jacob, "Grid computing: What are the key components? -Taking advantage of Grid Computing for Application Enablement" (June 2003), TSO Redbooks Project Leader.

[2] Ann Chervenak, Ewa Deelman, Carl Kesselman, Bill Allcock, Ian Foster, Veronika Nefedova, Jason Lee, Alex Sim, Arie Shoshani, Bob Drach, Dean Williams, Don Middleton, "High- Performance Remote Access to Climate Simulation Data: A Challenge Problem for Data Grid Technologies" (2001), A technical document, Supercomputing Conference-SC.

[3] Antonio Carzaniga, Matthew J. Rutherford, Alexander L. Wolf, "A Routing Scheme for Content- Based Networking" (June 2003), Software Engineering Research Laboratory Department of Computer Science University of Colorado, Boulder, Colorado, USA, Technical Report CU-CS-953-03 and IEEE INFOCOM.

[4] Baur, C., Moore, R., Rajasekar, A. and Wan, M, "The SDSC Storage Resource Broker 8th Annual IBM Centers for Advanced Studies Conference" (1998), Toronto, Canada.

[5] Beynon, M., Kurc, T., Catalyurek, U., Chang, C., Sussman, A. and Saltz, J, "Distributed Processing of Very Large Datasets with DataCutter", Parallel Computing, 27 (11). 1457-1478. International Journal of Grid Computing and Applications (IJGCA) Vol.2, No.4, December 2011 61.

[6] Mark Braverman and Anup Rao, "Toward Coding for Maximum Errors in Interactive Communication" (2014), IEEE Transactions on Information Theory, Vol. 60, No. 11,November 2014.

[7] Chervenak, A., Foster, I., Wesselmann, C., Salisbury, C. and Tuecke, S , "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets " (2001),J. Network and Computer Applications (23). 187-200.

[8] Foster, I. and Kesselman, "Data Grid Reference Architecture" (2001), Technical Report GriPhyN-2001-12.

[9] Stockinger, H., Samar, A., Allcock, B., Foster, I., Holtman, K and Tierney, B, "File and Object Replication in Data Grids" (2002), Journal of Cluster Computing, 5 (3). 305-314.

[10] Vazhkudai, S., Tuecke, S. and Foster, I, "Replica Selection in the Globus Data Grid. International Workshop on Data Models and Databases on Clusters and the Grid (DataGrid 2001)" (2001), IEEE Press.

[11] Nicolas Bitouze, Frederic Sala, S. M. Sadegh Tabatabaei Yazdi, and Lara Dolecek, "A Practical Framework for Efficient File Synchronization" (2013), Fifty-first Annual Allerton Conference Allerton House, UIUC, Illinois, USA.