# Synchronizing Files Operations with Secure Based Memory Scalable Approach

[1]Pradnya More,[2]Neha Prabhune,[3]Nidhi Vyas,[4]Amruta Gadekar
[1,2,3] B.E Student, Dept. of Computer Engineering, D.Y.Patil Institute of Engineering & Technology, Ambi, Pune University ,Maharashtra, India
[4]Professor, Dept. of Computer Engineering, D.Y.Patil Institute of Engineering & Technology, Ambi, Pune University, Maharashtra, India

***Abstract:*** *As* of late, there has been a blast of enthusiasm for mining time series databases. Similarly as with most software engineering issues, portrayal of the information is the way to productive and viable arrangements. A standout amongst the most usually utilized portrayal is piecewise direct estimate. This depiction has been used by various investigators to help bunching, arrangement, ordering and information sharing methodology of time arrangement information. An assortment of calculations have been proposed to get this portrayal, with a few calculations having been freely rediscovered a few times. In this venture, we attempt the primary broad audit and experimental examination of all proposed methods. We demonstrate that every one of these calculations have lethal defects from an information sharing point of view. We present a novel calculation that we experimentally show to be better than all others in the writing.

***IndexTerms*– de-duplication, concatenation, chunks.**

## I. INTRODUCTION

In this work, we 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 ideal time) of the processes will be executed 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 numerous clients. 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 connects 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. MOTIVATION

Privacy is most important factor which is supposed to be obtained nowadays .Also what we would like to provide is global data sharing scheme in distributed network with an efficient increase in data transfer. Security can be provided during the transmission phase instead of being processed after the data transfer is completed. Accuracy of the data being inserted or deleted can be obtained using security algorithm. Synchronization of hardware with software being another aspect can be obtained.

## III. BACKGROUND

**1) "The SDSC Storage Resource Broker. 8th Annual IBM Centers for Advanced Studies Conference"**

Baru, C., Moore, R., Rajasekar, A. and Wan, M. describes the Storage Resource Broker (SRB) middleware infrastructure that provides a uniform, UNIX-style file I/O interface for accessing heterogeneous storage resources distributed over wide area networks. Using its Metadata Catalog (MCAT), SRB provides collection-based access to data based on high-level attributes rather than on physical filenames. SRB also supports automatic replication of files on storage systems controlled by SRB. In contrast to the layered Globus architecture with direct user and application control over replication, SRB uses anintegrated architecture, with all access of data via the SRB interface and MCAT and with SRBcontrol over replication and replica selection.

**2) "Exploration and Visualization of Very Large Datasets with the Active Data Repository"**

Kurc, T., Catalyurek, U., Chang, C., Sussman, A. and Saltz, J. introduced the Active Data Repository and Data Cutter systems support analysis of large datasets, and could be used to achieve high-performance operations. Antonio Carzaniga et al[4] used traditional broadcast protocol combined with a specific content based protocol and used a push pull mechanism for the propagation

of routing information, he evaluated the protocol delivery messages to nodes, prevent unnecessary message traffic and produce a reasonable and stable amount of control traffic.

**3) "Performance Evaluation of Data Transfer Protocol GridFTP For Grid"**

Hiroyuki Ohsaki and Makoto Imase discussed that GridFTP is a data transfer protocol in grid computing which is widely used for transferring a large volume of data efficiently. There is a request manager that depends in particular on two components of the Globus Toolkit: the replica catalog for replica location and GridFTP for secure, efficient transfer. In addition, it uses Message Digest (MDS)-2 for access to Network Weather Service (NWS information), and Grid Security Infrastructure for authentication. This uses of Globus Toolkit components is consistent with proposals for a data grid architecture which includes four levels: fabric, connectivity, resource and collective.

**4) , "Study of Chunking Algorithm in Data De duplication"**

A.Venish and K. Siva Sankar has summed up different algorithms and methods for the de duplication of data. In the data deduplication process, the first and the most important step is the granular division and subdivision of data. For proper granularity, an effective anefficient chunking algorithm is used. If the data is chunked accurately, it increases the throughput and the net deduplication performance.

## IV. CONCEPT OF DATA TRANSFER

Data is transferred by some applications such as electronic mail, file transfer, web documents, so bandwidth and timing are important things for data transfer. Figure 2 refers data transfer;

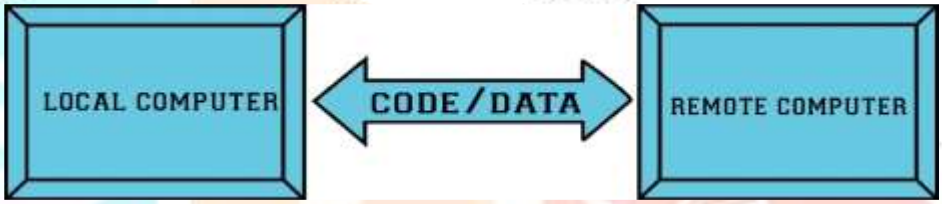



**Fig** :- Data Transfer

If you want to transmit small data, you need small rate bandwidth such as the application of internet telephony encodes voice at 32 kbps. However, if you have huge files and want to transmit them, you need more bandwidth. This is more advantages than small rate bandwidth. Timing is important when you transmit the data. Applications should provide quick data transferring to save time. For example, real-time applications of internet telephony, virtual environments, multiplayer games or etc.

## V. Java Technology

***a) Java***

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform.

Java technology is object oriented programming language as well it provides the JRE platform.

- JDK(Java Development Kit)- JDK consists of JRE and other development tools such as java,javac(compiler),Javadoc etc.
- JRE(Java Runtime Environment)-It provides Java runtime libraries along with JVM.
- JVM(Java Virtual Machine)-It is a machine on which byte code(.class file) executes.JVM converts byte code to machine code.JVM provides memory management,garbage collection.

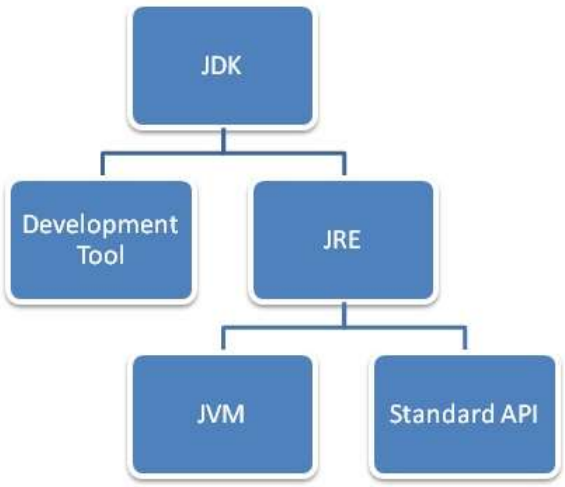



Fig:- Java Technology Hierarchy

***B)File system***

SFTP stands for "Secure File Transfer Protocol". The Secure File Transfer Protocol ensures that data is securely transferred using a private and safe data stream. It is the standard data transmission protocol for use with the SSH2 protocol. WISE-FTP implements a reliable and user-friendly version of the client side of this protocol. The SFTP protocol's main purpose is to transfer data, but it is also used to obtain general access to the FTP server's file system. It encrypts commands and data in terms of avoiding passwords and information which are susceptible.

Before establishing a connection, the SFTP server sends an encrypted fingerprint of its public host keys to ensure that the SFTP connection will be exchanging data with the correct server. The first time the connection is established, this key is not yet known to the client program and must therefore be confirmed by the user before data is exchanged for the first time. Once you have established a connection to an FTP server and are sure that it is really the correct server, you should save the fingerprint information locally.

**Vi. Algorithms:-**
**Chunking Algorithm :**

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.

***a) TTTD***
The Two Threshold Two Divisor (TTTD) chunking method [10] ascertains that chunks smaller than a particular size are not produced., the method ignores the chunk boundaries after a minimum size is reached.
TTTD applies two techniques where two divisors (D, the regular divisor and D0, the backup divisor) are used. By applying these divisors, TTTD guarantees a minimum and maximum chunk size. A minimum and a maximum size limit is used for splitting a file into chunks for searching for duplicates.

***Fixed-Size Chunking.***
Fixed-size chunking method splits files into equally sized chunks. The chunk boundaries are based on offsets like 4, 8, 16 kb, etc. .This method effectively solve issues of the file-level chunking method: If a huge file is altered in only a few bytes, only the changed chunks must be re-indexed and moved to the backup location. However, this method creates more chunks for larger file which requires extra space to store the metadata and the time for lookup of metadata is more.

***Variable Size Chunking***
The files can be broken into multiple chunks ofvariable sizes by breaking them up based on the content rather than on the fixed size of the files. This method resolves the fixed chunk size issue. When working on afixed chunking algorithm, fixed boundaries are defined on the data based on chunksize which do not alter even when the data are changed. However, in the case of a

variable-size algorithm different boundaries are defined, which are based on multipleparameters that can shift when the content is changed or deleted. Hence, onlyless-chunk boundaries need to be altered. The parameter having the highest effecton the performance is the fingerprinting algorithm.

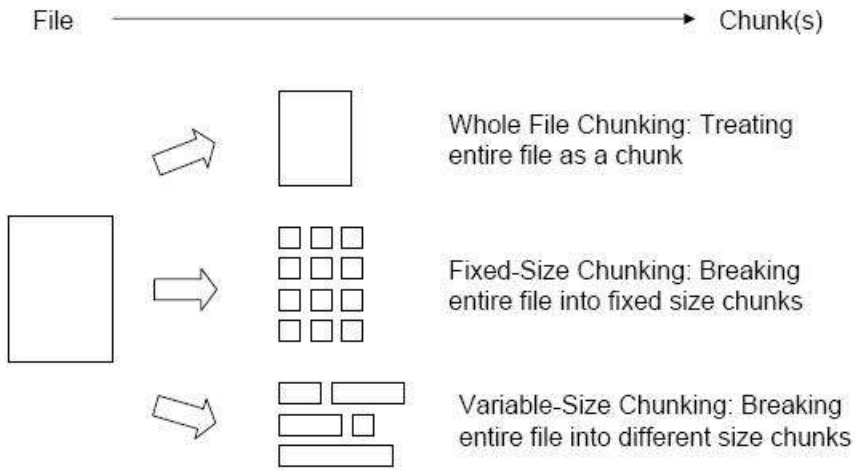



Fig :- Types of Chunking

***b) Blowfish***

Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages. Blowfish is also a block cipher, meaning that it divides a message up into fixed length blocks during encryption and decryption. The block length for Blowfish is 64 bits; messages that aren't a multiple of eight bytes in size must be padded.Blowfish is public domain, and was designed by Bruce Schneier expressly for use in performance-constrained environments such as embedded systems. Blowfish algorithm is divided in 2 phases:-

**1) Generating the Sub keys**

P is an array of eighteen 32-bit integers. S is a two-dimensional array of 32-bit integer of dimension 4x256. Both arrays are initialized with constants, which happen to be hexadecimal digits of π (a pretty decent random number source).

The key is divided up into 32-bit blocks and XORed with the initial elements of the P and S arrays. The results are written back into the array. A message of all zeros is encrypted; the results of the encryption are written back to the P and S arrays. The P and S arrays are now ready for use.

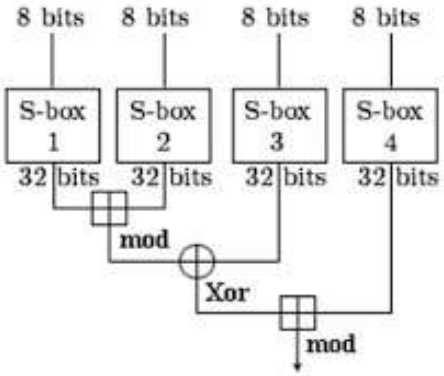
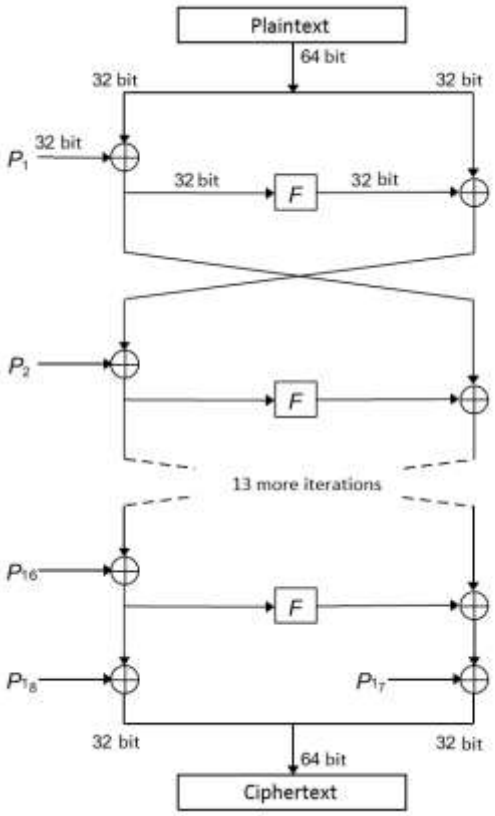



Fig :- Generation of sub keys

**2)Blowfish Encryption**

In this a 64-bit plaintext message is first divided into 32 bits. The "left" 32 bits are XORed with the first element of a P-array to create a value I'll call P', run through a transformation function called F, then XORed with the "right" 32 bits of the message to produce a new value I'll call F'. F' then replaces the "left" half of the message and P' replaces the "right" half, and the process is repeated 15 more times with successive members of the P-array. The resulting P' and F' are then XORed with the last two entries in the P-array (entries 17 and 18), and recombined to produce the 64-bit ciphertext..Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order.




**Vii. Experimental Results:-**

***System Detail:-***

- Processor:- Intel® Core™ 2 Duo CPU
- RAM:- 3GB
- System Type:- 64 Operating System
- Operating System:- Windows 7 Ultimate
- Internal Memory:- 320 GB

***Data Sets:-***

| Data Type | File Size | | |
|---|---|---|---|
| 1. Audio | 4610 KB | 8020 KB | 3920 KB |
| 2. Video | 1110 KB | 20000 KB | 13000 KB |
| 3. Document | 21 KB | 835 KB | 433 KB |
| 4. Application | 22194 KB | 13588 KB | 982 KB |
| 5. Image | 6.35 KB | 9.09 KB | 707 KB |

***Result Analysis:-***

| File Size | Transfer Time | |
|---|---|---|
| | Character | Buffer |
| 1. 64.7 KB(PDF) | 359 milliseconds | 1.02 milliseconds |

| | | |
|---|---|---|
| 2. 1061KB(mp3) | 9890 milliseconds | 16.0 milliseconds |
| 3. 139KB(.PNG) | 1295 milliseconds | 0.0 milliseconds |
| 4. 1100KB(mp4) | 10702 milliseconds | 17.0 milliseconds |

**Viii. Conclusion:-**

In our paper we have considered the buffered approach to transfer the data from one destination to another. The data can be of any type be it Application files, Audio , Video, Text , Images , Pdf, etc. Since it's a buffered approach the computation time required is reduced to minimum possible milliseconds. We have used splitting and concatenation mechanism which uses chunking algorithm to efficiently solve the problem of space utilization. This obtains all memory blocks available for data sharing and hence data de-duplication is obtained. Using our new data structure, the data owner can perform insert, modify or delete operations on file blocks with high efficiency. Blowfish Algorithm ensures security by providing encryption and decryption facility while performing the transfer which is optional.

The future scope of the project can be enhanced by using our application for space utilization in cloud database management.

**IX. ACKNOWLEDGMENT**

The authors would like to thank the publishers, researchers for making their resources available and teachers for their guidance. We thank the college authority for providing the required infrastructure and technical support. Finally, we extend our heartfelt gratitude to our friends and family members.

**REFERENCES**

**[1]** A. Venish and K. Siva Sankar,(2016) , "Study of Chunking Algorithm in Data Deduplication" , Proceedings of the InternationalConference on Soft Computing Systems, Advances in Intelligent Systemsand Computing 398.

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

**[3]** Ann Chervenak, EwaDeelman, Carl Kesselman,BillAllcock, Ian Foster, Veronika Nefedova, Jason Lee, Alex Sim, Arie Shoshani, BobDrach, Dean Williams, Don Middleton, (2001), "High-Performance Remote Access to Climate Simulation Data: A Challenge Problem for Data Grid Technologies" – A technical document, Supercomputing Conference-SC.

**[4]** Antonio Carzaniga,Matthew J. Rutherford, Alexander L. Wolf, (2004), "A Routing Scheme for Content- Based Networking", Software Engineering Research Laboratory Department of Computer Science University of Colorado, Boulder, Colorado,USA, Technical Report CU-CS-953-03, June 2003 and IEEE INFOCOM.

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

**[6]** Beynon, M., Kurc, T., Catalyurek, U., Chang, C., Sussman, A. and Saltz, J. (2001) , "Distributed Processing of Very Large Datasets with DataCutter". Parallel Computing, 27 (11). 1457-1478. International Journal of Grid Computing & Applications (IJGCA) Vol.2, No.4, December 2011 61

**[7]**Kurc, T., Catalyurek, U., Chang, C., Sussman, A. and Saltz, J. (2001), "Exploration and Visualization of Very Large Datasets with the Active Data Repository". IEEE Computer Graphics & Applications, 21 (4). 24-33.

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

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

**[10]** Hiroyuki Ohsaki, and Makoto Imase, (2009),"Performance Evaluation of Data Transfer Protocol GridFTP For Grid", International Journal of Applied Mathematics And Computer Sciences Vol. 3, No.1,.