INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Synchronizing data sharing approach with reducing communication overhead on a multicore system

Pooja Bhosale
bhosalepooja927@gmail.com
Dr. D.Y. Patil Institute of Engineering, Management and Research, Akurdi, Maharashtra

Tejal Fegade
itstejal99@gmail.com
Dr. D.Y. Patil Institute of Engineering, Management and Research, Akurdi, Maharashtra

Shweta Bharambe
shwetabharambe123@gmail.com
Dr. D.Y. Patil Institute of Engineering, Management and Research, Akurdi, Maharashtra

Mily Lal
likemily@gmail.com
Dr. D.Y. Patil Institute of Engineering, Management and Research, Akurdi, Maharashtra

Pratik Doke
pratikdoke95@gmail.com
Dr. D.Y. Patil Institute of Engineering, Management and Research, Akurdi, Maharashtra

ABSTRACT

Synchronization problem can occur when a file has been modified or it can happen when communication has to take place. Due to synchronization problem, general edits may occur in bursts or in isolation. General edits may cause communication overhead and it can create conflicts on the object access. After the modification of file, it is required to maintain the synchronization of files on various location. There can be chances of increasing the communication overhead due to general edits. In this work, we will show that a small amount of interaction can help design low complexity codes with close to optimal communication rate. Insertion on storage media play a major role on any operating system. Insertion on the logical partition cannot be accomplished thought if we have enough space available on our storage media. During the insertion process of the object if the object is broken into a different object then it is possible to store it on multiple logical locations. Our model describes the System which will help to manage synchronization of multiple files and it can have the load balancing of a single file on multiple location. The algorithm work effective when the object has to be stored on single physical storage with multiple logical partition.

Keywords: *Communication, Data Sharing, File Synchronization, File Transfer, Load Balancing.*

1. INTRODUCTION

System model accepts data for storage as a stream of bytes which are collected and stored in a manner efficient for the media. It provides structured, organized access to data and metadata. The system model manages how and where data on a storage media, is stored, accessed and managed. When the person working on the system often required to transfer the file from one location to multiple locations and required to maintain the concurrency in various files. File sharing service provides a way to store and access information, such as documents, data, photos, and video. Copying the file require communication between the storage device and external media. File integrity requires being maintained while sharing file on the multiple locations. File 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 copy and move files at the maximum possible speed and more secure. It can resume split file transfer, skips 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.

Our model is based on the Synchronization mechanism of the file between the multiple parts of the object will be created based upon the future storage requirement. In another way, by copying the file, storing the file is a more important concept. In the process of file storing, the data is saved in files and folders and presented to both the system storing it and the system retrieving it in the same format. It requires Maximum Storage on logical drive equals to the size of Object. Even Storage drive has enough space to store the file sometimes this operation not supported as Maximum Storage needs to be in one logical Drive. The Idea behind the File transfer is concatenated and split the object that can be stored on multiple logical drives with maximum Speed of operation. This operation helps in reducing the overhead on the system and can utilize the memory efficiently. While sharing it on multiple locations it needs to be ensured that the original file has to be obtained when the pieces of the file combined together.

2. EXISTING SYSTEM

In this era of Modern computing all, we required dealing with the files. Every day it is required to transfer a file from one location to another location. The existing system has the great ability to transfer the file to any location, System can store the file on devices with the ease and can have the ease of accessing the file again. General edits on the storage happen when file is transferred from one location to other the [2] The problem comes when the storage device has less logical storage on storage media, Existing system cannot store the file on such a disk instead it creates the prompt low disk storage even though if device has available physical storage on it.

3. PROPOSED SYSTEM

Today's conventional systems do not provide utilization of storage devices which in turn causes waste of resources. The proposed System provides high-performance System which can enhance the performance of the System by keep memory utilization at its full cost. The system can do it by making full utilization of resources while sharing resources on the storage drive. The cause occurs when external devices try to store data on internal storage devices of the System. This System will help to store large files on them by using the mechanism of chunking. It will also possible to reduce the overhead while accessing the same file in multiple locations. The System makes the use of Two Threshold Two divisor algorithm to make the chunking of the file. This is Dynamic chunking algorithm which makes the use of storage services and can help to create the pieces of the file on the basis of available storage. The System will make the proper use of available storage. It is required to specify the minimum and maximum size to create the pieces of the file. It will help to utilize memory resources of the system effectively.

A. Modules in the system

The objects involved in the projects are:

- Authentication Object: Enter the user name and password to get the access to security system.
- Splitting Object: It converts information into no of blocks with the help of Splitter.
- Concatenate Object: It converts Split Blocks into combine block with the help of Concatenate.
- Memory Object: In this, Memory Blocks Management is processed in the multicore system.
- Usability Object: In this, Audio functionality provides for operation status
- Indexing Object: It design to work on data sharing scheme in the distributed mode for the multicore system.

B. Features of Proposed System.

1) Splitting of the file:

We can create a partition of the data for storing the file on multiple locations on the storage media.

2) Reliability:

Grouping certain formation of the file together to maintain file integrity. It provides good storage performance.

3) Volume Stripping:

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) Less time for copying process:

It has the ability to store data randomly at any storage media in short amount of time.

5) Effective 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.

C. Component Diagrams

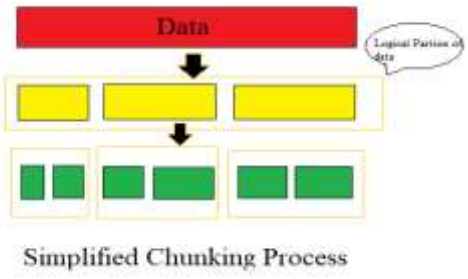

Fig 1.1: Simplified Chunking Process

4. SYSTEM ARCHITECTURE

- In this system, there is a mechanism which helps us to transfer the files with load balancing with another disk.
- This system provides an interface which helps us to select the location of the destination path.
- By using this system, File gets stored on distributed location easily with less time.
- The System has a backup module which can retrieve the file if deleted from the original location.
- Backup Faci;ity like a cloud is maintained at a smaller level, so one can utilize whole memory efficiently and reliable file distribution.

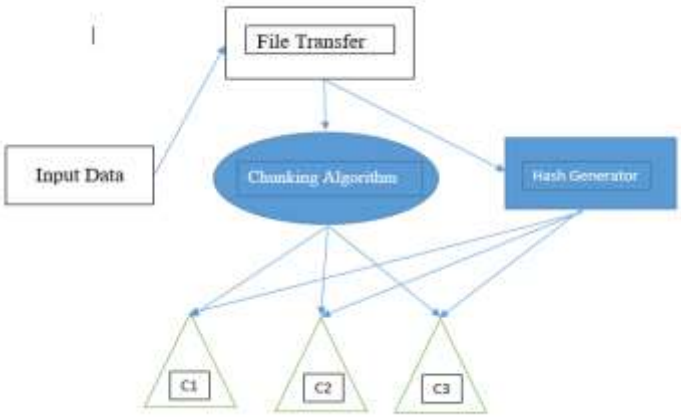

Fig 1.2 System Architecture

5. IMPLEMENTATION

a) Technology

- AWT / Swing :
  - AWT which stands for Abstract Window Toolkit is used by Java programmers for creating graphical user interface (GUI) objects. It is one of the frameworks for creating GUIs.
  - Swing is also used for building Graphical User Interface in java program. It provides platform-independent features for the File system.

- File System :
  - The file system is a method of organizing and retrieving files from a storage media. It usually consists of files separated or split into subparts.
  - The file system defines the way of our files are saved and stored on particular location.

b) Algorithm:

Two Threshold Two Divisior:

Two thresholds two divisor TTTD is a Context-Based Dynamic Chunking Algorithm. The main difference between static chunking and Dynamic chunking is that in Static Chunking it creates part of the file with the fix Size .it add byte by byte and create fixed sized block chunk which is different than previous chunks.

Two thresholds two divisor is a dynamic chunking algorithm where we are required to specify two values Minimum and Maximum size chunk, Algorithm start from the first byte and create minimum size chunk and compute hash value after it calculates hash with main and secondary value. If the result of it is equal to the hash value of the first chunk then new chunk must begin there.

The Two Threshold Two Divisor chunking algorithm needs to compute a hash of every n-byte window over the file. Thus, for maximal performance, it's desirable to choose a hash function that allows the hash value to be efficiently updated as the window is shifted forward by one byte.

Blow fish Algorithm:

Blow Fish Algorithm is used for Encryption of Chunks stored on backup storage. It will help to ensure that only trusted person can access the stored parts of the file.

6. MATHEMATICAL MODEL

S = {S, E, X, Y, NDD, F, Su}

Where,

i. S: Initial State.

Select the file from source path and Select the multiple destinations for the split file.

ii. E: Final State.
Split file copied into the selected location on storage media. When it is necessary to view the file then by using split and join function people can concatenate the entire file.

iii. X: Input Given.

The input to the System is the File and the locations on the Storage device, Minimum and Maximum value of chunk.

iv. Y: Output Obtained.

The file will get stored on distributed Location Easily.

v. P: processing component.

1. File Distributor 2. Backup Module

vi. NDD: Non-Deterministic data.

This is a Size of the Storage media which will be taken based on the storage Available on the Storage Device

vii. Su: Success State.

The splitted file will get stored and retrieved by using join and concatenate function in the system.

viii. F: Failure State.
If the storage required to store the file is not available on the whole disk then only the application will fail.

7. EXPERIMENTAL RESULT

In the Experimental result, Data Transfer speed is increased up to 5 Mbps. The table describes the time required to encode and decode the file at average file size at 100 Mb.

System Details:

	Encode time	Decode time
Docx	**0.4**	**0.7**
Mp4	**1.02**	**1.1**
Exe	**1.05**	**1.08**
Pdf	**1.2**	**1.03**

8. CONCLUSION

The existing System has the drawback of memory utilization while storing which will be effectively reduced by using the proposed system. The system provides file transfer with maintaining the synchronization among multiple chunks of the single file.

9. REFERENCES

[1] R. Venkata Ramanan, H. Zhang, and K. Ramchandran, " Interactive low-complexity codes for synchronization from deletions and insertions," in Proc. IEEE 48th Allerton Conf. Commun., Control, Comput., Monticello, IL, USA, Sep./Oct. 2010, pp. 1412–1419.J.
[2] R. Venkataramanan, V. Narasimha Swamy, and K. Ramchandran, "Efficient interactive algorithms for file synchronization under general edits," in Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton), Monticello, IL, USA, Oct. 2013, pp. 1226–1233.
[3] S. S. Pradhan, K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction", IEEE Trans. Inf. Theory, vol. 49, no. 3, pp. 626-643, 2003
[4] G. Cormode, M. Paterson, S. C. Sahinalp, U. Vishkin, "Communication complexity of document exchange", Proc. ACM-SIAM Symp. On Discrete Algorithms, pp. 197-206, 2000.
[5] Frederic Sala, Clayton Schoeny, Nicolas Bitouzé, Lara Dolecek , " Synchronizing Files From a Large Number of Insertions and Deletions " IEEE Transactions on Communication 6, June 2016. 6. A. Orlitsky, " Interactive communication of balanced distributions and of correlated files ", SIAM J. Discrete Math., vol. 6, no. 4, pp. 548- 564, 1993.

