An Efficient Transaction Memory Storage Management Model for Images

P.Sowmya
Department of Computer Science and Engineering
Vignan's Institute of Engineering for Women
Visakhapatnam, India
sowmya.polumahanti@gmail.com

N. Bhagya Lakshmi
Department of Computer Science and Engineering
Vignan's Institute of Engineering for Women
Visakhapatnam, India
blbhagya84@gmail.com

P. Jyothi Priya
Department of Computer Science and Engineering
Vignan's Institute of Engineering for Women
Visakhapatnam, India
jyothipriya2209@gmail.com

Y.Sriya
Department of Computer Science Engineering
Vignan's Institute of Engineering for Women
Visakhapatnam, India
sriyayegi@gmail.com

Dr. P. Vijaya Bharati
Associate Professor
Department of Computer Science and Engineering
Vignan's Institute of Engineering for Women
Visakhapatnam, India
pvijayabharati@gmail.com

Abstract—Transaction Memory Management Model is a structureless virtual file system from which complex databases can be built. This system works only on images which can be further extended to heterogeneous data. Because of its simple flat structure, a more complex storage engines or file systems with an internal structure as required by the user can be developed. It supports full transactions and creates an exact storage that fits the data without wastage of memory.

Keywords—*Virtual file system, Flat structure, Complex storage*

I. INTRODUCTION

The data is increasing data by data. Due to increase in data, data storage Management plays a vital role. There are numerous number of file systems to store data. But, storing of data in an efficient manner is a key challenge. Transaction Memory Storage Management Model is a structureless virtual file system that will store the data in a single master file within the streams.

Transaction Memory Storage Management Model will create an exact storage that fits the data without wastage of memory. This system works only for images with different formats (jpg, bmp, tif and png) and these images are stored in the form of streams in which each stream is referenced with stream Id. It allocates data when the streams are enlarged and deallocates when the streams are shrunk.

II. LITERATURE SURVEY

Many researchers who carried out exploration in storage management field have concentrated more on cluster based architecture while this proposed work being is focused directly on storage system for large complex data which is structureless virtual file system.

There are many distributed file systems like HDFS and GFS. HDFS stands for Hadoop File System, which holds large amount of data. HDFS stores file system metadata and application data separately [6]. HDFS has master/slave architecture [3]. Each file is divided into blocks and replicated among the nodes [2]. HDFS contains two nodes namely DataNode and NameNode [4][5][10]. The Default Block size in HDFS is 128MB [10]. Each NameNode maintains the metadata of data block and the information of the data block on which the primary replica and secondary replica are stored. But, In this file Model if the data to be stored is 1MB then the complete 128MB of the block have to be allocated. GFS stands for Google File System which is a scalable distributed file system developed by Google [1][8].GFS cluster consists of single master and multiple chunk servers[7]. Even GFS contains the files that are divided into chunks which maintain replicas that are present in different chunk servers. The default block size of GFS is 64MB.

III. PROPOSED METHOD

The main purpose of Transaction Memory storage management model for images is to create an exact storage that fits the data without wastage of memory which cannot be achieved through any other file systems. This consists of a master file in which all the images are stored in the form of streams. Each stream can be referenced with a stream Id which is a type of Guid.

A. Segmentation

The master file is divided into variable length segments. Each stream can be composed of one or more segments that are chained together. Segments that mark the free spaces are also chained in a stream called free space stream.

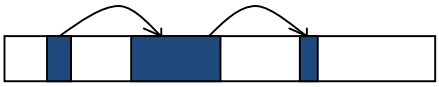

Fig. 1. Representation of a master file with variable length segment

Each segment has a segment metadata which is written at the beginning and holds the following information:

- Size of the segment(Int 64)
- Location of next segment(Int 64)
- Metadata Checksum(Int)

To prevent high fragmentation, the segment size is always a multiple of block size which is fixed to 512 bytes.

B. Space allocation and Deallocation

- When new empty storage is created free space stream is created which is made up of one segment occupying all of the virtual space available in the master file.
- When allocating and deallocating space for stream space, only two operations are performed they are split and merge.
- When the streams are enlarged segments from free space stream are taken in whole or split and added to stream segment chain.
- When segments are added to the segment chain all adjacent segments are merged into bigger segment to prevent excessive segmentation.

C. Stream table

The Meta data of all the streams is stored inside stream table. Stream table itself is stored in one of the streams and the location of it will be kept at the beginning of the stream. Stream metadata holds the following information:

- StreamId(GUID)
- Stream Length (Int 64)
- Initialized stream length(Int 64)
- Position of first segment in chain(Int 64)
- Tag(Int)

D. Transactions

- Transaction Memory storage supports Full transactions.
- During a transaction on every write to the master file transaction memory storage copies the about to be over written data into transaction log file.
- The transaction log file is a separate file located next to the master file and is cleared when transaction is committed.
- Transaction can be rolled back when an exception is thrown. When a system crash occurs the data can be retrieved.

IV. IMPLEMENTATION AND RESULTS

The project is implemented in .NET platform using C# language. The Transaction Memory Storage Management model is added into NoSQL category because it is meant to serve as a base on top of which custom storages or databases can be built.

The storage is created as per the users wish and also each image which is stored in streams can be stored in multiple segments. Those multiple segments are chained together. The images can be stored with different formats namely png, jpg, bmp, tiff, which can also be stored in different sizes. In this Transaction memory Storage Management Model millions of images can be stored and various transactions can be performed.

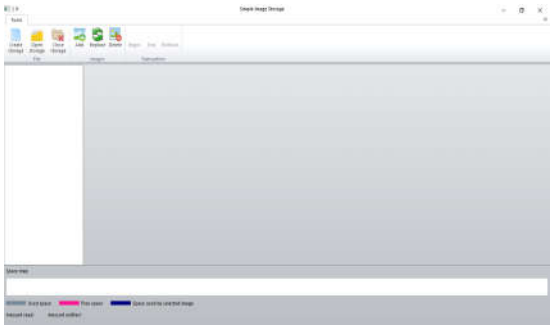

Fig. 2. Output of the project

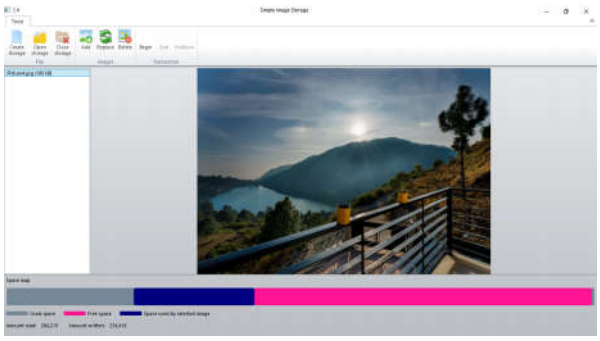

Fig. 3. After adding and deleting an image

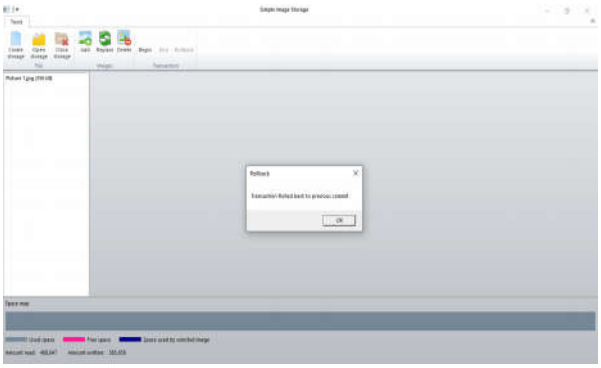

Fig. 4. After performing a rollback operation

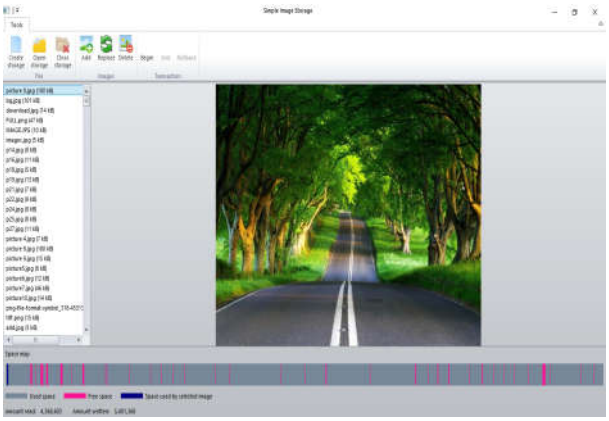

Fig. 5. After adding and deleting many images

As shown in the above picture the storage is created and the images can be added, replaced and deleted from the storage. The transactions can be committed and rolled back. When a system crash occurs the data can be retrieved.

V. CONCLUSION

Transaction Memory storage Management model is a structureless virtual file system which is an efficient model for storing the images without wastage of memory. It is a flat structure and works only on the images which can be extended to heterogeneous types.

REFERENCES

[1] B.Li,M.Wang, Y.Zhao,G.Pu,H.Zhu and F.Song,Modeling andVerifying Google File System, 2015 IEEE 16th International Symposium on High Assurance Systems Engineering, Daytona Beach Shores, FL,2015,pp. 207-214.

[2] International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2018 IJSRCSEIT|Volume 3 | Issue 3 | ISSN: 2456-3307.

[3] Apache Hadoop Project, http://hadoop.apache.org/, 2013.

[4] Hadoop, https://developer.yahoo.com/hadoop/tutorial/module2.html

[5] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chanskler Yahoo! Sunnyvale, California USA {Shv, Hairong} @yahoo-Inc.com The Hadoop Distributed File System, 2010 IEEE 26th symposium on Mass Storage Systems and Technologies(MSST)

[6] A. Sunita.B, Ather and Anita,R. Kulkarni "Hadoop MapReduce: A programming Model for large Data processing", American Journal of computer Science and Engineering Survey, AJCS, pp. 001-010, 2014

[7] Sanjay Ghemawat,Howard Gobloff,and Shun-Tak Leung Google "The Google File system".

[8] Wang,M.,Li, B, Zhao,Y., & Pu, G.(2014),Formalizing Google File System. 2014 IEEE 20th Pacific Rim International Symposium on dependable Computing.

[9] Chandramohan A. Thekkath, Timothy Mann, and Edward K.Lee.Frangipani: A scalable distributed file system.In proceedings of the 16th ACM Symposium on Operating System Principles,pages 224-237,Saint-Malo, France,October 1997.

[10] Google File System, https://www.sciencedirect.com