Detecting a Utilization Imbalance Between Dispersed Storage Units

Bin He[1,a], Wu Chuansheng[2,b]

[1]Street Yanshan Harbor District in Hebei Province, Qinhuangdao City No. 331 ,China

[2]Software College, University of Science and Technology Liaoning,China

[a]qhdhebin@gmail.com,[b]gykwcs@163.com

Keywords: utilization imbalance,,between; dispersed; storage network,storage units; detecting

Abstract. A method begins by a processing module of a dispersed storage network (DSN) obtaining utilization information regarding a plurality of storage units of the DSN, where first and second sets of storage units support a first logical storage vault. The method continues with the processing module detecting a utilization imbalance between a first storage unit of the first set of storage units and a second storage unit of the second set of storage units based on the utilization information, where the first and second storage units are not a common storage unit. The method continues with the processing module executing a data storage function regarding the first logical storage vault based on the utilization imbalance.

1. Introduction

Computers are known to communicate, process, and store data. Such computers range from wireless smart phones to data centers that support millions of web searches, stock trades, or on-line purchases every day. In general, a computing system generates data and/or manipulates data from one form into another. For instance, an image sensor of the computing system generates raw picture data and, using an image compression program , the computing system manipulates the raw picture data into a standardized compressed image.[1]

With continued advances in processing speed and communication speed, computers are capable of processing real time multimedia data for applications ranging from simple voice communications to streaming high definition video. As such, general-purpose information appliances are replacing purpose-built communications devices. For example, smart phones can support telephony communications but they are also capable of text messaging and accessing the internet to perform functions including email, web browsing, remote applications access, and media communications.[2]

2. Description of Related Art

A computer's storage system will be compliant with one or more computer storage standards that include, but are not limited to, network file system (NFS), flash file system (FFS), disk file system (DFS), small computer system interface (SCSI), internet small computer system interface (iSCSI), file transfer protocol (FTP), and web-based distributed authoring and versioning (WebDAV). These standards specify the data storage format (e.g., files, data objects, data blocks, directories, etc.) and interfacing between the computer's processing function and its storage system, which is a primary function of the computer's memory controller.[3,4]

Another solution is to utilize multiple levels of redundant disc drives to replicate the data into two or more copies. One such redundant drive approach is called redundant array of independent discs (RAID). In a RAID device, a RAID controller adds parity data to the original data before storing it across the array. The parity data is calculated from the original data such that the failure of a disc will not result in the loss of the original data. For example, RAID 5 uses three discs to protect data from the failure of a single disc. The parity data, and associated redundancy overhead data, reduces the storage capacity of three independent discs by one third (e.g., n-1=capacity). RAID 6 can recover from a loss of two discs and requires a minimum of four discs with a storage capacity

of n-2.

3. Embodiment of Dispersed Storage

As illustrated, the DS unit includes a storage unit control module , a plurality of memories of type A (1 through a), and a plurality of memories of type B (1 through b). The storage unit control module may be implemented with the computing core 26. The memories may be one or more of a magnetic hard disk, NAND flash, read only memory, optical disk, and any other type of read-only or read/write memory. The memories may be implemented as part of or outside of the DS unit . For example, memory A-1 may be implemented in the DS unit and memory A-2 may be implemented in a remote server (e.g., a different DS unit operably coupled to the DS unit via the network). In an example, memory A-1 through memory A-a are implemented with the magnetic hard disk technology and memory B-1 through memory B-b are implemented with the NAND flash technology. FIG. 1 is a schematic block diagram of an embodiment of a dispersed storage (DS) unit.[5]

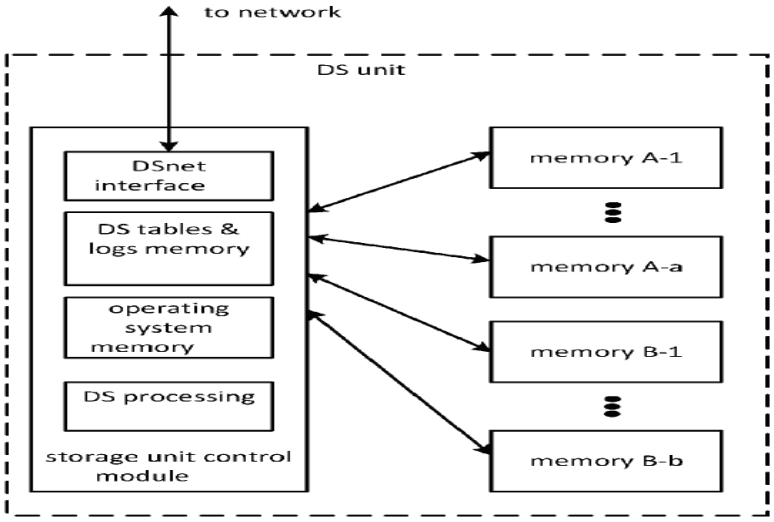

FIG. 1 is a schematic block diagram of an embodiment of a dispersed storage

As illustrated, the storage unit control module includes the DSnet interface , a DS tables and logs memory , an operating system (OS) memory , and the DS processing . The storage unit control module may be operably coupled to the computing system 10 via the DSnet interface by way of the network 24. The storage unit control module receives a store command, metadata, and data to store via the DSnet interface .

4. Dispersed Storage Unit

As illustrated, the DS unit includes a DS processing and a memory A-1. In an example, memory A-1 has one billion bytes of storage capacity. In an example of a storage operation, the DS processing receives 900 bytes of data . The DS processing determines an error coded dispersal storage function with a pillar width n=4 and a read threshold=3. The DS processing encodes the data in accordance with the error coded dispersal storage function to produce four data slices where the data slices are each approximately 300 bytes in size. The DS processing determines addressable locations within memory A-1 to store the data slices based on one or more of a lookup of where the last slices were stored, the local virtual DSN address to physical location table, available memory, memory status, a command, memory errors, and the error coded dispersal storage function. In an instance, the DS processing determines to evenly space the pillars apart evenly across the memory A-1. In another instance, the DS processing determines to utilize memory addresses that avoid known issues as indicated by the memory status for memory A-1. Next, the DS processing stores the data slices in the memory A-1 at the addressable locations.

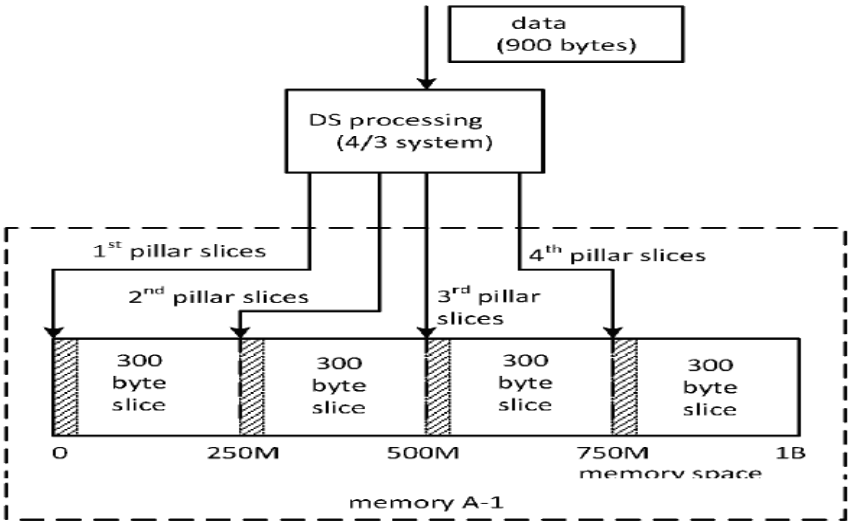

FIG. 2 is a schematic block diagram of another embodiment of a dispersed storage

In an example of a re-commissioning operation, a hard disk drive is utilized for a first time period (e.g., three years) within a non-dispersed storage system. The hard disk drive is re-commissioned in a dispersed storage system at the end of the first time or when a number of disk drive errors exceeds an error threshold. For example, the hard disk drive is removed from the non-dispersed storage system and installed in the dispersed storage system when the hard disk drive is producing too many disk drive errors. A processing module of the DS processing module generates an initial error profile based on the initial error profile. The processing module generates the initial error profile by the generating test data, storing the test data in two or more storage sectors of a set of addressable storage sectors, retrieving the test data from the two or more storage sectors of the set of addressable storage sectors to produce retrieved test data. FIG. 2 is a schematic block diagram of another embodiment of a dispersed storage .

In an example of a rebuilding operation of a single hard drive, a processing module detects a storage error of an encoded data slice of a set of encoded data slices, wherein the set of encoded data slices represents data encoded using an error coding dispersal storage function, wherein the single hard drive is defined to have a set of addressable storage sectors, and wherein encoded data slices of the set of encoded data slices are stored in corresponding addressable storage sectors of the set of addressable storage sectors. In addition, the processing module may detect a plurality of storage errors and determine a rate of increase of the plurality of storage errors. Next, the processing module evaluates the rate of increase of the plurality of storage errors to determine a level of reliability. The processing module determines a second error type when a size of usable storage space is greater than a storage threshold and when the level of reliability compares unfavorably to a reliability threshold.

Conclusion

While RAID addresses the memory device failure issue, it is not without its own failures issues that affect its effectiveness, efficiency and security. For instance, as more discs are added to the array, the probability of a disc failure increases, which increases the demand for maintenance. For example, when a disc fails, it needs to be manually replaced before another disc fails and the data stored in the RAID device is lost. To reduce the risk of data loss, data on a RAID device is typically copied on to one or more other RAID devices. While this addresses the loss of data issue, it raises a security issue since multiple copies of data are available, which increases the chances of unauthorized access. Further, as the amount of data being stored grows, the overhead of RAID devices becomes a non-trivial efficiency issue.

References

[1] Lie Xu,Liangzhong Yao,Christian Sasse."Power Electronics Options for Large Wind Farm Integration:VSC-Based HVDC Transmission,". PSCE06 . 2013

[2] ZHU Shunquan,SUN Weirong,WANG Qian.Review of R&D status of vanadium redox battery. Chemical industry and engineering progress . 2012

[3] P.Karlsson."DC distriibuted power system:analysis,design and control for a renewable energy system"..

[4] Zhang Z R,Yin Z D,Hu F X.Research of multi-farms transmission of distributed generation based on HVDC light. 2006International Conference on Power System Technology . 2010

[5] Fei Lin,Zhiwen Ma,Xiaojie You,Trillion Zheng.The grid connected converter control of multi-terminal DC system for wind farms. Proceeding of the 8th International Conference on Electrical Machine and Drivers (ICEMS 2005) . 2013