International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016

An Optimized Way for Backing up Block Level Data

Syed Tahseen Ahmed
M.Tech 4th Semester
Dept. of Computer Science & Engineering,
SJCE, Mysore, Karnataka, India

Divakara N
Assistant Professor
Dept. of Computer Science & Engineering,
SJCE, Mysore, Karnataka, India

Abstract:

Every day as our need to depend upon technology increases, the amount of user data as well as the accompanying system data being generated is huge. The data could range from as simple as personal photographs and videos to as complex as large user databases managed by big companies and corporations.
Large amount of data also increases the problem of archiving data. As more and more data is generated, the harder the task of backing up data to avoid from data corruption and data loss. There are various techniques to backup data; each new technique trying to be more effective and efficient than the previous one.
In this paper, we propose a technique to backup data, which can reduce the bandwidth utilization of the network, as well as the space utilization in the archiving device.

Keywords – Storage, Block-level, Full Backup, Optimized Backup, Incremental Backup

I. INTRODUCTION

Traditionally, backups were being done at file level. Generally users copy and create a backup of their files on another media, based on the desired criteria on the destination media. The advantage of file level back up is that it is very simple for the user, as the file is accessed through the operating system. The backup is done irrespective of the underlying storage or disk format.
Storage technology used can either be file level storage or block level storage.

A storage area network (SAN) is storage connected in a fabric (usually through a switch) so that there can be easy access to storage from many different servers. From the server application and operating system standpoint, there is no visible difference in the access of data for storage in a SAN or storage that is directly connected. A SAN supports block access to data just like directly attached storage [1].

Network-attached storage (NAS) is really remote file serving. Rather than using the software on your own file system, the file access is redirected using a remote protocol such as CIFS or NFS to another device (which is operating as a server of some type with its own file system) to do the file I/O on your behalf. This enables file sharing and centralization of management for data [1].

So from a system standpoint, the difference between SAN and NAS is that SAN is for block I/O and NAS is for file I/O. One additional thing to remember when comparing SAN vs. NAS is that NAS does eventually turn the file I/O request into a block access for the storage devices attached to it [1].

II. FILE LEVEL STORAGE

This storage technology is most commonly found in hard drives, NAS (Network Attached Storage) systems and so on. The storage disk is configured with a protocol such as NFS or SMB/CIFS and the files are stored and accessed from it in bulk.
Features:

- The File level storage is simple to use and implement.
- It stores files and folders and the visibility is the same to the clients accessing and to the system which stores it.
- This level storage is inexpensive to be maintained, when it is compared to its counterpart i.e. block level storage.
- Network attached storage systems usually depend on this file level storage.
- File level storage can handle access control, integrate integration with corporate directories; and so on [2].

III. BLOCK LEVEL STORAGE

In block level storage, raw volumes of storage are created and each block can be controlled as an individual hard drive. Each block size can be from few KBs to MBs. These Blocks are controlled by server based operating systems and each block can be individually formatted with the required file system.

Features:

- Block level storage is usually deployed in SAN or storage area network environment.
- This level of storage offers boot-up of systems which are connected to them.
- Block level storage can be used to store files and can work as storage for special applications like databases, Virtual machine file systems and so on.
- Block level storage data transportation is much efficient and reliable.
- Block level storage supports individual formatting of file systems like NFS, NTFS or SMB - Server Message Block (Windows) or VMFS (VMware) which are required by the applications.
- Each storage volume can be treated as an independent disk drive and it can be controlled by external server operating system.
- Block level storage uses iSCSI and FCoE protocols for data transfer as SCSI commands, as well as FC protocols, act as communication interface in between the initiator and the target [2].

Here we will be looking at block level storage and backup.

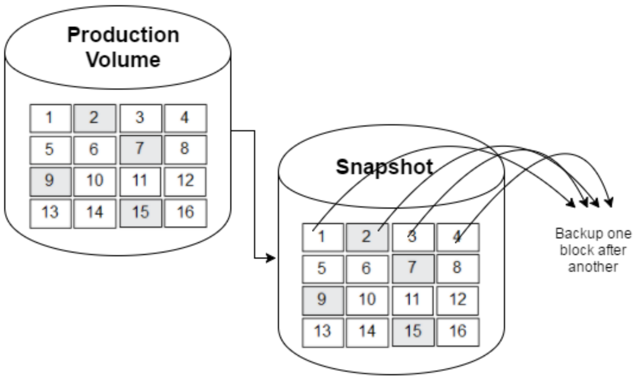
IV. BACKUP

In information technology, a backup, or the process of backing up, refers to the copying and archiving of computer data so it may be used to restore the original after a data loss event. The verb form is to back up in two words, whereas the noun is backup.[American Heritage Dictionary entry for backup, American Heritage Dictionary entry for back up [3].

V. DATA REPOSITORY MODELS

Full Back up:

Full backup is the most basic and complete type of backup operation. In general, it refers to backing up the entire data in the disk or drive. In a file level storage, full backup means taking backup of all the files and directories in the disk irrespective of they are modified or not. In a block level storage, the full back refers to backing up of all the blocks in the disk.



Advantages – There is no overhead for the backup. All blocks of the volume are backed up on the back up device without any extra processing.

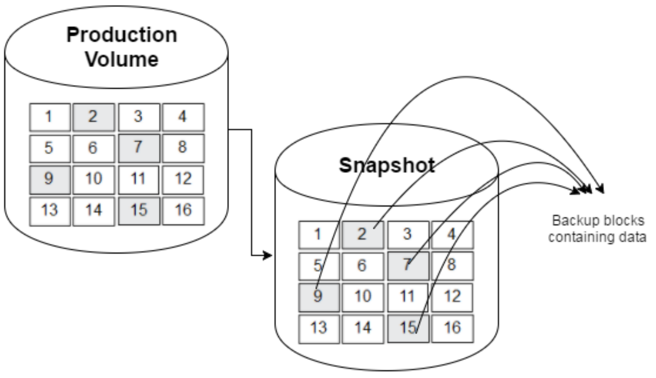
Disadvantages – The time taken for the backup operation will be more, as all the blocks of the volume have to be backed up, irrespective of data being present in them or not. This especially is a waste of bandwidth and space, when:

- The size of the volume is huge compared to the data populated in the volume.
- The backup operation has to be frequently performed on the volume

VI. PROPOSED METHOD:

Optimized backup:

While full backup takes backup of all the blocks, and is very simple to implement, it is not suitable when the data to be backed up is huge and/or when the modifications to data are very frequent. To overcome the issues related with bandwidth needed to transfer such huge amount of data as well as the space needed for the storage of such data, an optimized method can be used to reduce the bandwidth as well as space utilization problems.

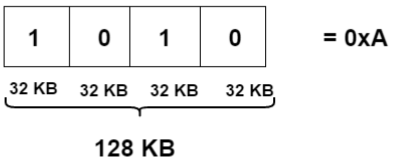


The method used to optimize the backup is – remember the locations or blocks where the data has been written, so that only those blocks can be backed up, without backing up all the blocks on the disk.

A hexa-decimal map is used to achieve this. The entire disk's block allocation can be represented using this map. The disk is represented in terms of the blocks.

Typically, block sizes can be of 32KB. If one block of 32KB size has even one bit of data, that block is considered as occupied. If not a single bit of 32KB block has any data, that can be considered as empty block.
To know if data is present in a block or not, binary numbers can be used.
Binary 1 represents data is present in a particular block.
Binary 0 represents data is not present in a particular block

Four binary digits can be used to represent one hexa-decimal value. That is, one hexa-decimal value would represent four 32 KB of locations of data, viz. 128 KB, as shown in the below figure.



So, if we consider a volume of size 1 GB, dividing it by 128 KB gives us 8192 – which is nothing but the size of the map or the locations represented by the 128 KB block. So there will be 8192 hexa-decimal values in the map, which represents which blocks of the volume are occupied, and which are not.

This map is saved, and referred to during backup operation. During the backup operation, each hexa-decimal digit of the map is analyzed and each individual binary digit is checked if its 1 or 0. Only blocks of binary 1 are read and backed up on the back up device.

Advantages – This greatly reduces the time needed for the backup operation, especially when:

- The size of the volume is huge compared to the data populated in the volume.
- The backup operation has to be frequently performed on the volume.

Disadvantages – There is an overhead involved in calculating the map before the backup operation can take place. And while doing the backup operation, the map has to be referred to as to which blocks are being backed up. However, this overhead, is very less, when compared to the time saved with respect to full backup.

Incremental Backup

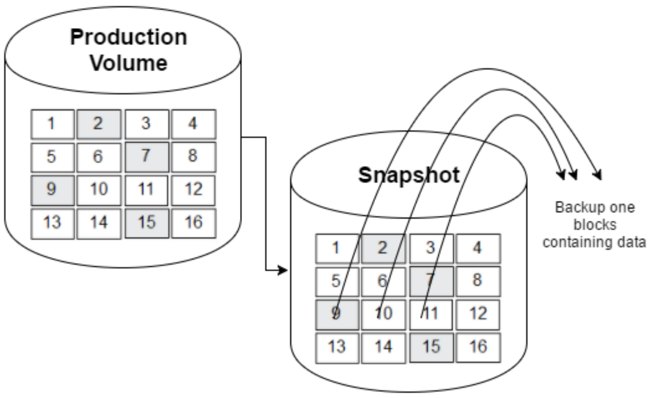
An incremental backup is one that provides a backup of files that have changed or are new since the last incremental backup; it is one that backs up only the data that have changed since the last backup — be it a full or incremental backup. An incremental backup is a backup of latest changes since the last backup (any level) so that when a full recovery is needed the restoration process would need the last full backup plus all the incremental backups until the point-in-time of the restoration

At file level, this means backup only those files which have been changed since the last backup. At block level backup, instead of files, blocks are considered. Only the blocks changed since the last back up are backed up on the backup device.

The map which was created to indicate the blocks, where data was being written for the use of Optimized backup, and the same map will be used for Incremental backup. Whichever blocks have change of data, those blocks are indicated as '1' in the map. The rest all of the blocks will have '0'. This way the map will only portrays the new change in data. This map will be used to backup the blocks as indicated by the map.

Advantages – This greatly reduces the time needed for the backup operation, especially when:

- The size of the volume is huge compared to the data being modified or updated in the volume.
- The backup operation has to be frequently performed on the volume.



VII. CONCLUSION

Recovery Management is an important feature in all data-centric applications – both private as well as enterprise. Backup is only one half of recovery Management; Restore plays an equally important role. The most essential thing in recovery management is to keep the data consistent and free from corruption.

Virtualization plays a crucial role in recovery management, as it helps in abstracting the complexities with respect to servers and storage.

As the complexities of real world increase, more and more data is being saved. This data has to be backed up. And newer and more efficient technologies are needed for their storage and management.

REFERENCES

[1] http://searchstorage.techtarget.com/answer/The-difference-between-SAN-and-NAS

[2] http://www.iscsi.com/resources/File-Level-Storage-vs-Block-Level-Storage.asp

[3] https://en.wikipedia.org/wiki/Backup

[4] R. J. T. Morris and B. J. Truskowski, The evolution of storage systems, IBM Systems Journal, vol. 42, no. 2, pp. 205-217, 2003.

[5] S. Quinlan, S. Dorward, Venti: a new approach to archival storage, in: Proceedings of the First USENIX Conference on File and Storage Technologies, Monterey,CA, 2002, pp. 89–101.

[6] Weijun Xiao, Qing Yang, Jin Ren, Changsheng Xie, and Huaiyang Li, Design and Analysis of Block-Level Snapshots for Data Protection and Recovery, IEEE Transactions on Computers, VOL. 58, 2009

[7] Guiping Wang, Shuyu Chen and Jun Liu, A Network Differential Backup and Restore System based on a Novel Duplicate Data Detection algorithm, WSEAS Transactions on Information Science and Applications, E-ISSN: 2224-3402, Volume 11, 2014

[8] C. H. Qian, Y. Y. Huang, X, F. Zhao, and T. Nakagawa, Optimal backup interval for a database system with full and periodic incremental backup, Journal of Computers, vol. 5, no. 4, pp. 557-564, 2010.

[9] S. Nakamura, C. Qian, S. Fukumoto, and T. Nakagawa, Optimal backup policy for a database system with incremental and full backups, Mathematical and Computer Modelling, vol. 38, no. 11-13, pp. 1373-1379, 2003.