International Journal of Advance Research in Science and Engineering
Volume No.08, Issue No.06, June 2019
www.ijarse.com
IJARSE
ISSN: 2319-8354

SOFTWARE SOLUTION FOR STORAGE TECHNOLOGY INDEPENDENT DATA MANAGEMENT

Nandini.V[1], Dr. Rajashekar.C.Biradar[2]

[1]*School of ECE, Reva University, (India)*
[2]*School of ECE, Reva University, (India)*

ABSTRACT

The basic objective of the paper is to provide an algorithm by an indigenous solution for storage technology independent data management. It is a software tool which caters to various requirements of file level replication and migration in data center environment. Every data set has its own service level requirements in terms of computing performance, retention and availability. In data center environment ensuring data integrity and availability with optimal utilization of resources in this technically evolving scenario needs a carefully planned data management strategy. Optimized synchronization job execution where only changes are transferred from source to destination. Comprehensive command line interface for administration. Seamless refreshments of underlying hardware without disturbing the ongoing operations.

Keywords: Retention, Availability, Optimized, Interface, Administration

I. INTRODUCTION

Data Synchronization refers to the idea of keeping multiple copies of a dataset in coherence with one another, or to maintain data integrity. Technology obsolescence are quite rapid in the IT domain, something which is in the state of art today may become obsolete tomorrow. Hardware and software ages over time, companies declare end-of-scale and end-of –support to phase out old products and introduce new and improved products. Problems with specialized storage solutions include costly specialized licenses, limited copy options, limited copy constraints, specialized skill sets, in specialized formats, and with various technological constraints. Security, control and legal protection of data are among the most important aspects.

Major features include: Periodic and one time syncing of files from source to multiple replicas constrained by various file attributes, Replicas based on time window and /or storage space usage. Based on the time window expiry and

storage space usage threshold, data moves to replicas. Sync job distribution across multiple hosts for load sharing. Optimized sync job execution where only changes are transferred from source to destinations. Comprehensive command line interface for administration. Aging of hardware and software. End–of–sale and end–of-support of products. Technology Obsolescence. Ensuring data integrity while migrating. Seamless refreshments of underlying hardware without disturbing the ongoing operations. Problems with specialized storage solutions: costly specialized licenses, limited copy options, limited replica constraints, specialized skill sets, in specialized formats, and with various technological constraints. All these motivated to develop a data back up in non-proprietary format, while avoiding lock- in to a specific solution.

1.1 Organization of the paper

Beginning with an overview of Software Solution for Storage Technology Independent Date Management explaining what is the Software solution, why do we need that solution and how to implement. Followed by, Introduction which deals with the major features and what motivated us to do this project. Advantages of the solution which provides a data backup based on policy in non-proprietary format, while avoiding lock- in to a specific solution. Working, Algorithm, Implementation, Case Study and finally explained about Conclusion and the future work.

1.2 Working:

Terminology:

File Store: File system directory/directories grouped together under same context file stores may have multiple copies for administrative reasons.

Sync: Replicating all the file operations from one source location to destination location. File operations include creation, modification and removal. Client: Host where file stores are mounted and subjected to various changes by applications. Active copy: Copy of file-store which is currently accessible to clients. Replica: All other copies of file-store other than active copy are replicas. Complete Replica: A Replica which contains complete data set for the context, starting from the time of inception of the context. Partial Replica: It is defined based on time window and/or space usage. Eg: Last N days/months/years files. Contain all the data till storage space usage reaches a certain level i.e., threshold. Files are discarded from replica in order of their modification time to maintain space usage below configured threshold. Can be defined as combination of both of the above definitions. User: User of the application being developed i.e., storage administrator.

Sync job: One set of job for syncing, encompassing, syncing to one replica from single source.

Window Definitions:

Partial Replicas can be defined based on time window or storage space threshold. A reference time and threshold space usage can be defined for every sync-job configuration.

Eg. Mission start time, launch time etc.

All other time references are relative to reference time.

i. Time Window: Is defined by the window start time and window duration. Window duration of the last replica in partial-partial file-store is always infinity.

ii. Space Window: Is defined by the window start time and window size / threshold. Window size of the last replica in partial-partial file store is always infinity. Window end time is initially equated to window start time and it increases as the window moves. Window end time is assigned the last modified time of the oldest file to fit the window size and is recalculatedevery time when the size of data in the file store exceeds the threshold.

II. ALGORITHM

Solution is based on Client Server architectural pattern. Client Module solution provides comprehensive command line interface for administration, whereas server module solution carries out core responsibilities. Interaction between client and server module is via Socket API's. Client module is an independent executable and can be run from any host which has network reachability to server host. Server module is a daemon and runs in background on identified server host.

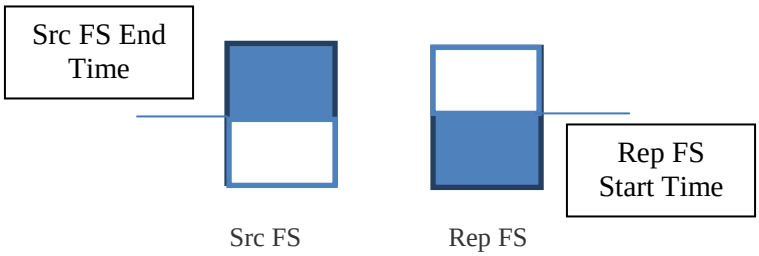

Fig.1 Representation of FS Sync type

Source Exclusive Window Fragment (SEWF):

List all files in this time window at "Src FS", says slist.

List all files in this time window at "Rep FS", says rlist.

Sort slist by last modified and rlist by name.

Keep adding the file sizes until it reaches threshold space.

For each file in slist within threshold limit using binary search check if it is available in "Rep FS"

- If not available, it is No Operation.
- If available it is reported as Warning/Error.

For each file in slist when threshold limit is exceeded using binary search check if it is available in Rep FS before the

Rep FS Start time as shown in the Fig.1

- If not available, the file should be created on Rep FS and deleted on Src FS.
- If available it is reported as Warning/Error.

For each file in RepFS check if the last modified time is after the Rep FS Start time as shown in the Fig.1

- If available it is reported as Warning/Error.

Who can use it?

It is a generic application and can be used by anyone who needs well administered data management. Regular syncing from one storage solution to another for backup.

Eg. Syncing of data from ISILON to SAN.

III. IMPLEMENTATION

Software has been implemented in Java, utilizing various Java SE Technologies. Rsync, well known syncing tools, can be utilized for various types of syncing. When different file opeartions are identified Rsync function is called from the Rsync protocol and that does the appropriate operations required. It also sends a notification or a message saying it has manually checked when the file operation Warning/ Error is reported.

IV. CONCLUSION AND FUTURE WORK

Software Solution has broken down the barrier across storages and simplified the data migration activity. As this solution is generic application, it is planned to extend the use for various applications using space domain approach. There are various scenarios that require well administered data management. Software solution in its current implementation, requires all storages to be mounted on the same system for data migration. Whereas, there are scenarios where storages cannot be mounted on the same system and software solution is required to migrate data over network of systems. Future work will be implementing it using space domain approach for a shared medium and introducing the concept of upadations of data and only the updates are transferred from source to destination. Concept of Defragmention can be introduced in future.