Backup and Recovery for File System and Databases

Malatesh Karibisti
M.Tech (AIT), 6th Sem
Reva University
Rukmini Knowledge Park, Kattigenahalli,
Yelahanka
Bangalore.
Malatesh.karibisti1@gmail.com

Prof. Meenakshi Sundaram
Associate Professor
Reva University
Rukmini Knowledge Park, Kattigenahalli,
Yelahanka
Bangalore.

Abstract: - Information protection is critical to a company's day-to-day operations. In the digital age, data is one of the most important possessions a company owns, and having an efficient and manageable backup and recovery strategy has become an IT imperative. It is very important to safe gaud this data against failure like media failures, mishap data deletes, data corruption etc. It is very important to have well planned backup and recovery mechanism for business continuity, well-designed data protection software is the need of the hour. A well designed protection software try to back up data on a regular schedule chosen by users. At these times they create one or more duplicate or deduplicated copies of the primary data and write it to a new backup file system. In this application we are trying to solve the problem backup of filesystem and databases by developing the web application which will allow the user to schedulers and on demand based robust backup and recovery, User will select applications to be backed up and subscribes to enable backups. It gives options storage repository model to back up the data like native file systems, Hadoop distributed file System HDFS filesystms, any new filesystem can be integrated very easily. In this application we will leverage the ability of HDFS as backup file system.

Keywords: Back up, Recovery, Hadoop distributed file System(HDFS).

I. Introduction:

Backup and data protections are most important for business continuity. Backup and recovery enables companies to safe guard and preserve their information during unfortunate events. A successful backup job starts with selecting and extracting intelligible units of data. Most data on modern computer systems is stored in files. These files are systematized into different file systems. Files that are actively being restructured can be thought of as "moving" and throw a challenge to back up. It is also useful to save metadata that describes the computer or the filesystem being backed up. Deciding what to back up at any given time is a harder process than it seems. By backing up too much redundant data, the data repository will fill up too quickly. Backing up an insufficient amount of data can eventually lead to the loss of critical information.

Backup system contains at least one copy of all data considered worth saving, the data storage requirements can be significant. Organizing this storage space and managing the backup process can be a complicated undertaking. A data repository model may be used to provide structure to the storage. Nowadays, there are many different types of data storage devices that are useful for making backups. Before data are sent to their storage locations, they are selected, extracted, and manipulated. Many different techniques have been developed to optimize the backup procedure. These include optimizations for dealing with open files and live data sources as well as compression, encryption, and de-duplication, among others. Every backup scheme should include dry runs that validate the reliability of the data being backed up. It is important to recognize the limitations and human factors involved in any backup scheme

Backup strategy starts with a concept of a data repository. The backup data needs to be stored, and probably should be organized to a degree. The organization could be as simple as a sheet of paper with a list of all backup media and the dates they were produced. A more sophisticated setup could include a computerized index, catalog, or relational database. Different approaches have different advantages.

In this application we are trying to solve the problem backup of file system and databases by developing the web application which will allow the user to schedule and on demand based robust backup and recovery, User will select applications to be backed up and subscribes to enable backups. It gives options storage repository model to back up the data like native file systems, HDFS file systems, any new file system can be integrated very easily.

II. Data protection.

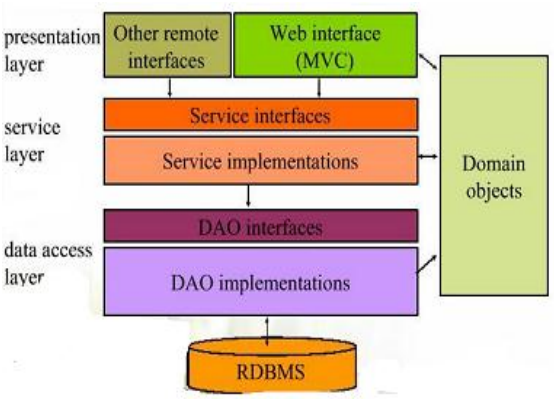
Most of the databases produce the files which are very large, taking backup of large databases which holds very large file sizes has been traditionally done with higher cost solutions like storage array backups.

The storage array backups are viable options for large enterprises but for the businesses which are at middle and lower side cannot afford to have storage arrays for both production and backups. Large file storage traditionally has been was at higher cost but with inventions of distributed file systems like HDFS, Storage of large file has been affordable at very lower cost. HDFS uses the commodity hardware which are arises with lower cost. HDFS cluster can be built with very low cost.

This highly configurable and scalable model of backup and recovery application. Currently the backup has been mostly using storage arrays whose cost of ownership is high and also these technologies has drawback of significant of maintenance cost. HDFS is inexpensive because of two reason the file system relies on commodity storage disks that are much less expensive than the storage media used for enterprise grade storage. Secondly, the file system shares the hardware with the computation framework as well, Also HDFS is open source and does not levy licensing fee on the user. HDFS has been around for more than 7 years and is considered mature technology. There is a large community behind it and a broad range of organizations that are storing petabytes of data on HDFS.

III. System Design

We are trying to leverage the capability of Hadoop distributed file systems (HDFS), by utilizing HDFS for storing the user data to handle the accident damage of data. Hadoop's underlying file system HDFS makes three copies of data by default. These can be used to recover from a number of failure scenarios within the Hadoop cluster. Application will be designed with Java EE layered architecture as shown in the below figure.



Each module will have implementation of service and command layer. Module can be named as.

1. HDFS Module
2. Persistence module
3. File System module
4. Configuration module
5. SSH Module
6. Backup module

IV. Evaluation & Results A. Experimental Environment:

V. Related Work:

VI. Conclusions

HDFS was originally developed for storing large files. Solving the problem backup of file system and databases by developing the web application which will allow the user to schedule and on demand based robust backup and recovery. The application gives options storage repository model to back up the data like native file systems, HDFS, any new file system can be integrated very easily.

VII. Future Work

As for future work, this solution can be enhanced further to provide a more advanced backup and recovery support for various database or any other application which uses very large files like no sql database etc

References

[1] Chandrasekar S, Dakshinamurthy R, Seshakumar P G, Prabavathy B, Chitra Babu "A Novel Indexing Scheme for Efficient Handling of Small Files in Hadoop Distributed File System". In proceedings of Computer
Communication and Informatics (ICCCI), 2013 International Conference, Coimbatore, Jan. 2013, pp. 1 8.
[2] The Hadoop Distributed File System: Architecture and Design, available: http://hadoop.apache.org/
[3] The major issues identified: The small files problem, available: http://www.cloudera.com/blog/2009/02/02/the-small-files-problem, (2010).
[4] A. Chervenak, J. M. Schopf, L. Pearlman, M.-H. Su, S. Bharathi, L. Cinquini, M. D'Arcy, N. Miller and D. Bernholdt, "Monitoring the Earth System Grid with MDS4", Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing. Washington: IEEE Computer Society, (2006).
[5] http://ieeexplore.ieee.org/document/7790278/
An effective merge strategy based hierarchy for improving small file problem on HDFS
[6]http://ieeexplore.ieee.org/document/6394874/
Improving the Efficiency of Storing for Small Files in HDFS
[7]
http://www.oracle.com/technetwork/java/javase/overview/index.html
[8] https://maven.apache.org
[9] http://www.jcraft.com/jsch/
JSch - Java Secure Channel
[10] http://searchoracle.techtarget.com/tip/Oracle-11g-Backup-and-recovery-concepts.