International Journal of Engineering & Technology, 7 (4.41) (2018) 1-4

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

High Speed Data Backup and Disaster Recovery for Big Data Enterprises

Jayaram M R[1]*, Ani R[2], Sudheesh Narayanan[3], Sayantan Ray[4]

[1 2]Dept. of Computer Science and Applications
Amrita Vishwa Vidyapeetham, Amritapuri, India
[3 4]Knowledge Lens Pvt. Ltd., Bengaluru, India
**Corresponding author E-mail: jayaram@knowledgelens.com*

Abstract

Big data is growing every second without any barriers. The exponential growth rate in the number of devices connected adds the demand for handling of large volumes of structured and unstructured data faster from these devices. Behind all of these is Big Data sitting strong in an authoritative position. Big Data has become more important in nature, as it grows in volume without any bars and also operates in click stream, IoT sensors and web or video content. In this case, obviously the plans used for traditional disaster recovery is outdated and are no longer applicable. As we have found this, it is an important measure to analyze and implement a quick disaster recovery plan that can act as a shield during cyber-attacks, natural disasters or equipment failures. The proposed system initiates protection of big data environment considering new technologies that will keep improving with growth in data, scaling, strong security and compression as well to make sure the critical data is safe and secure from simple and catastrophic equipment failures.

Keywords: *Advanced Encryption; Big Data Backup; Cloud; Disaster Recovery; HDFS; Spark*

1. Introduction

It is well known that our societies and people mostly rely on computer systems and data to complete their works. That means they expect the system to be highly available to them at anytime, anywhere. Business people and IT technicians are well aware that even small periods of down-time may result significant loss in finance. Earlier and even now, business organizations are making use of disaster recovery mechanisms such as backups in tape, transport those tapes to offsite, and recently replication of data. One of the main challenge in providing Disaster Recovery services is to maintain the support of Business Continuity, allowing the applications to quickly recover after a failure occurs. A Disaster Recovery Plan also provide Business Continuity by minimizing the time for recovery or get back to original state of the data lost due to disaster. This paper discusses about how cloud platforms can be used to provide low cost Disaster Recovery solutions with support of High Speed Data Transfer. The formatter will need to create these components, incorporating the applicable criteria that follow.

The idea of cloud platforms is that an organization or user needs to pay only for what they use. All the services provided by the cloud platform providers are always available on demand so that any user can just get in and get the resources required. The memory and network usage are very less in Amazon Web Services (AWS) to synchronize data from primary site to the cloud. It is possible that, at any point of time the disaster happens, the organization can re-quest for full amount of resources which is required to run the application. The processes that we do can be automated through cloud platforms where the resources are allocated automatically once a disaster happens. The automation can thus dramatically reduce the time of recovery after an equipment failure or data loss which ensures the business continuity is followed. The general architecture of the proposed system is shown below in Fig. 1.

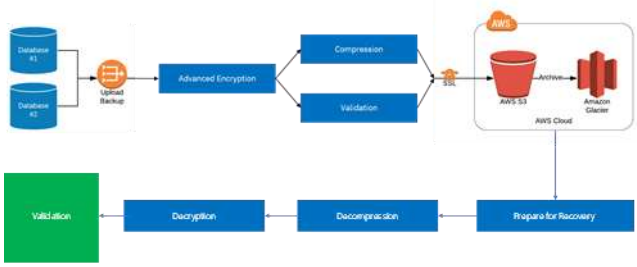

Fig. 1: Architecture of the Backup and Disaster Recovery Process

2. Related Works

This section deals with the various experiments and research carried out on disaster recovery plans on Enterprise.

In [1], the paper talks about the mechanism that Google introduced which later became one of the popular algorithms in managing Big data. The researchers who introduced the MapReduce algorithm to perform distributed processing. Till then the processing was simply linear for each and every piece of data. The solution allowed the distributed computations to make way providing the idea of fault tolerance, data partition and job scheduling.
In [2], the author introduces a solution that supports data backup along with disaster recovery. Here, the best advantage is that the user and service provider are independent each other. For e.g. to

backup a piece of data, the user need not ask or request the service provider as the cloud itself is the one which provides the service. The data backup supports strong protection of data and also simplifies the process of migration between two or more sites.

In [3], Lili Sun proposed the technique of Cold and Hot Backup Strategy. These are basically the storage policies applied to the system. It may be a partition of both Hot and Cold Backup or completely single policy all the way. The Cold Backup Service Replacement Strategy deals with the recovery process which gets triggered automatically once a failure in service is detected. The Hot Backup Service Replacement Strategy on the latter side is also a recovery process which is focused on service composition in dynamic network. The best part is that; during the implementation no services are expected to go down, which means the services will be in active state and the results will be used to analyze whether the implementation was successful.

3. Methodology

The Disaster Recovery Plan is also known in the term of Business Process Contigency Plan which is a group of pre-defined procedures which are meant to be executed so that the effect created due to a disaster will get minimized and the organization can always maintain the critical operations as per the mission. Disasters are of several forms. Even if the disaster used to happen unexpectedly, it requires a thorough planning on the part of recovery.

A disaster may be a catastrophic failure, accidental deletion by an user, or a device failure. Some of the key objectives of a disaster recovery plan is to ensure there is no downtime for the important clusters, maintain high availability, no data loss, recovery from the point of hardware failure and also support for business continuity.

The application maintains data accuracy, consistency, and versioning with its complex internal mechanisms. The HBase mechanism is used for scenarios like Disaster Recovery that supports replication of data across datacenters. Zookeeper plays an important role in HBase Replication in the case of effective management of the replication activity.

The architecture represents the case of copying two huge datasets that is very much valuable to some enterprise to the Cloud. To ensure the data is secure enough, a layer of AE-256 is performed on top of the data. The encrypted data is then compressed to a lesser volume and pushed to an S3 Bucket. Amazon S3 buckets are basically the folders with unique names created to store the data. The entire environment covered with Apache Spark to accelerate the power of performance.

The difference between normal backup and disaster recovery is that the DR services gives the advantage of setting up each and every setting the enterprise lost during a disaster, and triggers back to setup how the servers, data and configuration were before the occurrence of disaster. It just makes it look like the disaster has not happened before. The data, metadata, settings and server configuration including logs are backed up. The application is programmed in such a way that it can setup the server, recover data with a trigger.

The process of recovery starts with preparing the logs of data ready. The data is compressed and then decrypted. The backed-up data and recovered data are compared using the metadata to check the integrity.

It is important to understand the concepts such as Recovery Time Objective (RTO) and Recovery Point Objective (RPO). Recovery Time Objective is the time set for the recovery of business and IT activities after any disaster. The aim is to calculate how fast the activities get recovered to as usual state. This includes the set of preparations a business takes forward in order to implement the recovery process. Recovery Point Objective on the other side is focused on data loss tolerance. It is calculated by looking at the time between backup of data and the quantity of data that could be lost in between backups. The way both differ is the purpose which they pose to. The RTO deals with the overall business and systems whereas the RPO focuses on data.

The administrators and data-line owners will decide the RTO/RPO of the large datasets in the data lake. There is a concept known as Tiered storage which helps in better storage of data and performance. These are the three policies in the proposed solution.

3.1. Hot Storage Policy

This policy is focused on both data storage and compute. The data that is used for processing most of the time and also popular ones will be considered in this policy. All the replicated data will be stored in disk.

3.2. Cold Storage Policy

Use this policy if your focus is on data storage more than compute. Basically, the Cold policy is only for storage with lesser compute. The data that is being used lesser will be moved to the archive in this case. All the replicated data will be stored in archive in this policy,

3.3. Warm Storage Policy

This policy is basically a mixed version of hot and cold. That means some of the replicated data will be stored in disk and the remaining ones will be stored in the archive. These storage policies are set at the point of file creation or at the lifetime of file.

The proposed solution provides an intuitive user experience which is much integrated suite that enables the power of high-speed backup and recovery with data protection on the Hadoop platform. It enables enterprise users to make replicas of data across datacenters for scenarios like disaster recovery. The files that are usually backed up are the data stored in HDFS, files in local system, HBase, and Hive tables, Impala and their respective metadata. The data stored in HDFS helps to ensure that the critical data inside is available at all times, even in the case of shutdown of a datacenter.

• The data which is most important for business operations are always considered first and selected in priority.
• It will be tiring for users to always monitor and manually push tasks. There comes the mechanism of scheduling jobs. The jobs are scheduled for both data replication and snapshots where the processes get triggered automatically at the time scheduled.
• The users of the system would love to see what is going on right now. The console shows a list of jobs that are currently running, those which are queued, completed, or failed. It is very easy and usable for the users to just find the section and monitor the data replication jobs quickly.
• The alerts are triggered when a data replication or snapshot creation job gets failed or interrupted. It is very important to alert the users in that case where the business is totally dependent on schedules.

One of the striking part in Hadoop is that the latest version supports 3X replication, means the files are copied three time to ensure the data is always safe. But organizations considering this are no more interested in getting a disaster recovery plan. The mechanism used for copying data is a tool known as DistCp, which stands for Distributed Copy. MapReduce, the paradigm of DistCp supports distribution and also is fault-tolerant.

The data collisions are avoided by setting the options so that the process of copying multiple sources will get a safer side. All the

operations completed including the job list, skipped files, time required to copy files and other details can be seen in the console. There is a feature called 'Compare', which cross-checks the listing of source and destination after copying of file(s).

3.4. MapReduce Programming model

The MapReduce is the most popular programming model used for distributed processing. It is designed in such a way that the mechanism allows distributed computations which is fault-tolerant. Map and Reduce are the two stages in the algorithm. The first one splits the data, whereas the second collects and joins the results.

A set of (key, value) pair defines the MapReduce model. The data which is processed, steps in between, and the final phase results work on the basis of (key, value) pair. It is impossible to run a MapReduce job without the usage of (key, value) pairs. The MapReduce phases are illustrated below in Fig. 2.

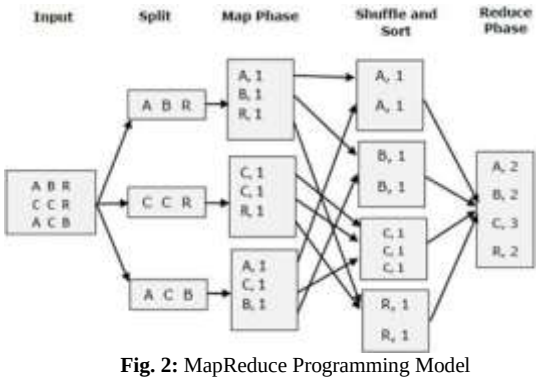

Fig. 2: MapReduce Programming Model

The phases in a typical MapReduce job are of three. The first and foremost thing is to read the source data and push it to MapReduce. The (key, value) pair is then generated. The second part covers the process of shuffling and grouping of output keys by (key, value) pair so as to generate a list of values. The keys are then partitioned and sent to Reducers. The final step is performed by Reducers where they aggregate the lists to build a single value for each and every pair.

3.5. High Speed Backup and Recovery using Spark

The framework called Apache Spark aims at faster performance of distributed computing on Big Data. The platform allows loading programs, and query it iteratively, which makes it a best-suited tool and iterative processing which is very useful in Machine Learning processing. It was discovered by addressing the limitations in the Hadoop MapReduce paradigm which allows dataflow in a linear fashion that makes an intensive disk-usage. Operations performed on Resilient Distributed Datasets schedule tasks automatically into partitions, maintaining the locality of persisted data. They are also fault-tolerant and can be rebuilt at any point of time in case of data loss.

3.6. Backup and Disaster Recovery in HBase

The data stored in HBase can be backed up and later restore at the point of occurrence of disaster. These are the approaches to be considered in backing up and recovering data in HBase:

3.6.1. Snapshots

Snapshots are basically the restore images that we create in operating system. Using the metadata, later the system will be able to restore back to the time the user created that particular snapshot

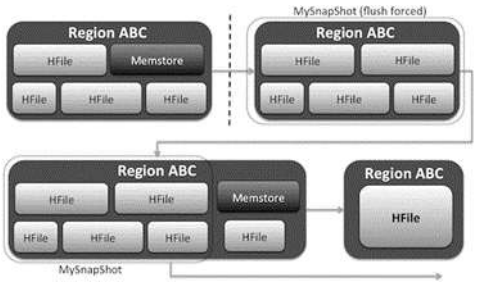

Fig. 3: An example of Snapshots

3.6.2. Replication

This approach is flexible during data ingestion from any data center as it ensures that the data is replicated across other data centers.

3.6.3. Export

The Export tool is a built-in utility in HBase that helps export of data easily from a table in HBase to Sequence files in HDFS.

3.6.4. CopyTable

CopyTable function creates a MR job that utilizes the API to read from the table and creates a replica of the table.

3.6.5. Offline Backup of Raw HDFS Data

The clusters might be shut down for several reasons. But the user should have the provision to still copy all the data in the HDFS.

3.6.6. Kerberos Authentication

The proposed solution supports transfer of data securely from HDFS to Amazon S3 having source end point security as Kerberos and destination end point security as either Kerberos or any other security. That is a major addition to this platform to have different security over the endpoints. Kerberos is one of the best available network authentication protocol which uses secret-key cryptography.

The Kerberos protocol uses highly stronger cryptography so that an organization will be able to prove its identity to a server across an insecure network. The enterprise users will be in complete control of the security — they can manage the encryption keys without depending upon the cloud provider and can take outside which gives the provision for users to activate their authentication to the cloud via the organization's Active Directory.

3.6.7. Compression of huge files using bzip2 and Snappy

Compression of data is an critical part in business organizations where terabytes of data are generated weekly. The data and files should be compressed in order to improve performance in backup and saving space. Compressing files brings two major benefits: it helps in reducing the storage required to store data and it speeds up transfer of data across the network or to or from disk. While dealing with large volumes of data, both of these savings are notable so as we can consider how to use compression in Hadoop. bzip2 and Snappy also provides compression/decompression. It does not provide maximum compression, instead it aims for a very high-speed compression.

4. Results and Analysis

4.1. Data Description

Each and every backup required some quantity of data. We had a set of files of types local, table and few more that would turn up around a total of 137.84 GB (Table I) This was an experimental set of sample data for upload purpose. Out of all, the local files

had a cost around 120 GB. There was table having 17 lakh syslog messages. Those datasets were added to analyze the performance of search query. As of now, these data are stored in a disk.

Table 1: Data Sheet

Types of files	Size
Local files	120 GB
Datasets + Tables	13 GB
Metadata	1.2 GB
Log files	134.6 MB
Other file types	3.63 GB

Performance Impact	Data Footprint	Downtime	Incremental Backups	Ease of Implementation	MTTR	
Snapshots	Minimal	Tiny	Brief (Only on Restore)	No	Easy	Seconds
Replication	Minimal	Large	None	Intrinsic	Medium	Seconds
Export	High	Large	None	Yes	Easy	High
CopyTable	High	Large	None	Yes	Easy	High
API	Medium	Large	None	Yes	Difficult	Up to you
Manual	N/A	Large	Long	No	Medium	High

Fig. 4. Feature List

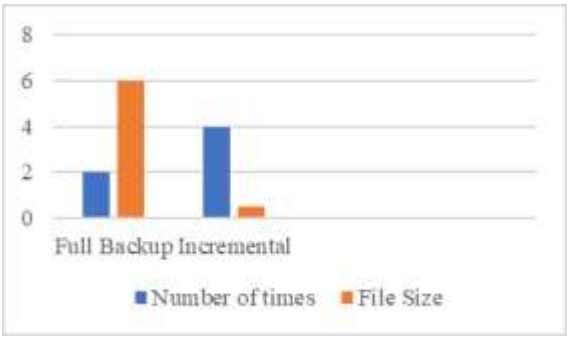

Fig. 5. Daily Statistics of Backup

Table 2. Backup Speed for Block Data

Workload	Time(s)	Length(G)	Speed(MB/s)	TransferTime(s)	TransferSpeed(MB/s)
a	18.85	1.88	99.90	16.50	114.14
b	21.19	2.11	99.42	18.42	114.39
c	15.91	1.51	94.95	13.29	113.66
d	21.32	2.05	96.09	17.91	114.38
e	18.52	1.82	98.24	15.95	114.09
f	18.68	1.82	97.37	15.94	114.08

Table 3. Backup Speed for Metadata

Workload	Time(s)	Length(KB)	Speed(MB/s)
a	0.118	24.96	0.211
b	0.121	28.33	0.234
c	0.028	19.09	0.682
d	0.118	23.56	0.200
e	0.028	22.62	0.808
f	0.228	25.09	0.110

The Statistical data Fig.4 shows that type of backup performed totally depends on the size of data and business requirement.
Backups are performed daily – both morning and evening most of the days – based on time zones. The statistics shows how fast is the backup and recovery process.

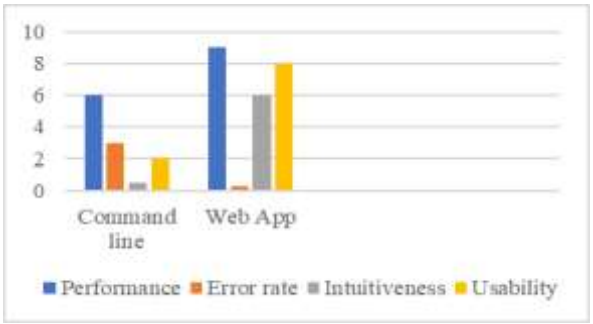

Fig. 6: User Experience and Performance Ratings

The table above shows the user experience and performance metrics. It is identified that users prefer the web app over the command line. As said, the web app offers a lot of features inside a much better intuitive screen with high performance of data ingestion.

5. Conclusion

As with the above result it is concluded that the proposed solution was able to perform backup and recovery with a remarkable speed rate. The concept of a Disaster Recovery Plan as a web app stands out over all metric rates. The application was able to take over humans in setting up broken servers and settings manually as well. The error rate is less compared to the previous techniques, the performance, intuitiveness and usability is higher as the user is now be able to monitor server health and backup status.

Acknowledgement

We sincerely thank our colleagues, work mates and staffs who contributed to and helped in this research work. This would not have been here without them.

References

[1] Jeffrey Dean, Sanjay Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, Google Inc.
[2] Vijaykumar Javaraiah, Brocade, Backup for cloud and Disaster Recovery for Consumers and SMBs - Advanced Networks and Telecommunication systems (ANTS), IEEE 5th International Conference, 2011.
[3] Lili Sun, Jianwei An, Yang Yang, Ming Zeng, Recovery Strategies for Service Composition in Dynamic Network - 2011, International Conference on Cloud and Service Computing
[4] Y. Ueno, N. Miyaho, and S. Suzuki, Disaster Recovery Mechanism using Widely Distributed Networking and Secure Metadata Handling Technology - 2009, Proceedings of the 4th edition of the UPGRADE-CN workshop.
[5] Rabi Prasad Padhy, Big Data Processing with Hadoop MapReduce in Cloud Systems, Senior Software Engineer, Oracle Corp, Bangalore, Karnataka, India.
[6] Anindita Khade, Dr. Subhash K. Shinde, Research Scholar, Map Reduce algorithm for Data Compression - Department of Computer Science Engineering, Lokmanya Tilak College of Engineering, Koparkhairane, Navi Mumbai, (MS), India.
[7] Binu P.K, Akhil V, Mohan V, Smart and secure IoT based child behaviour and health monitoring system using hadoop, Dept of Computer Science and Applications, Amrita School of Engineering, Amritapuri, Amrita Vishwa Vidyapeetham, Amrita University, India
[8] Sai A, Salim S, Binu P.K, Jisha R.C, A hadoop based architecture using recursive expectation maximization algorithm for effective and fool-proof traffic anomaly detection and reporting, Dept of Computer Science and Applications, Amrita School of Engineering, Amritapuri, Amrita Vishwa Vidyapeetham, Amrita University, India
[9] R.G Gaythri, Jyothisha J Nair, MapReduce model for finding closely knit communities in large scale networks, 2017 International Conference on Communication and Signal Processing (ICCSP), Dept of Computer Science and Applications, Amrita School of Engineering, Amritapuri, Amrita Vishwa Vidyapeetham, Amrita University, India