

# Interfacing the Cloud Using Diff-Utils Algorithm

**<sup>1</sup>Pravin Kandala, <sup>2</sup>Prof. T. Venkat Narayana Rao, <sup>3</sup>Dr. S.Masood Ahamed**

<sup>1</sup>Student, Dept. of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

<sup>2</sup>Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

<sup>3</sup>H.O.D, Dept. of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

## Abstract

Cloud Storage delivers virtualized storage on demand, which eliminates the need to purchase storage and maintaining the data. This is very much helpful for the small and medium size business, where establishing the data center and maintenance of the data center is tedious process. On the other hand backing up of data is essential for recovering our system from the data loss. The cloud provides such as storing large amounts of data, high reliability and data access from any place at any point of time. The main idea of this paper is to propose an interface (plug-in) which facilitates the back up of data stored in local disk on to the cloud. The interface that is to be designed will work on both windows and Linux operating systems. The interface will also provide the user to back up the data on the cloud platforms such as Google App Engine and Amazon S3. The interface allows the user to select any of the backup techniques that is either Full Back up or Incremental Backup.

## Keywords

Full Backup, Incremental Backup, Amazon S3, Google App Engine

## I. Introduction

Cloud computing is a word used to describe a variety of computing concepts which involve a large number of computers connected through a real-time communication network like Internet [1].

In science, cloud computing is a synonym for distributed computing over the network, the ability to run the program or application on many connected computers at the same time.

Cloud can also referred as network-based services, which appear to provide by real server hardware and simulated by software running on one or more than one real machine.

‘The cloud’ is essentially a metaphor for the Internet. [2] Marketers have further popularized the phrase “in the cloud” to refer the software, platforms and infrastructure that can be sold “as a service”, which can be remotely done on the Internet. The seller has actual energy-consuming servers, which host products and services from a remote location, they can simply log on to the network without any installation. The major models of services for cloud computing are known as Software as a Service, Infrastructure as a Service and Platform as a Service. These may be offered in a Public, Private or Hybrid network. [3] Google, Inc. is one of the most well-known cloud vendors. Cloud Interface provides the facility to store, share and backup files on Internet. Cloud Interface can be used in many different ways. For example, local data (such as Laptop) can be moved to the cloud storage for backup. We can also synchronize a virtual disk to the cloud and distributed to the other computers. The difference between the purchase of a dedicated appliance and that of cloud storage is not the functional interface, but nearly the fact that the storage is delivered on demand.

The customer pays either for what they actually use or in other cases, what they have allocated for use. Data Services such as Compression and De-duplication can be used to reduce the actual space consumed.

## A. Existing System

### 1. External Hard Drives

External hard drives are fast backup devices that maintains full backup of large amounts of data, which is performed, on a daily schedule. However, hard drives generally have no support for

encryption and storing in previous versions is cumbersome.

A hard disk drive (HDD) is a data storage device used for storing and retrieving digital information using rapidly rotating disks (platters) coated with magnetic material. Data is read in a random-access manner, which means the individual blocks of data can be stored or retrieved in any order rather than sequentially. They are one of the cheapest backup solutions. An HDD retains its data even when system is off. It consists of one or more rapidly rotating disks with magnetic heads arranged on a moving actuator arm to read and write data to the surfaces.

### 2. Tape Backup

In computers, tape backup is the ability to periodically copy the contents of all or a designated amount of data from its usual storage device to a tape cartridge device so that, in the event of a hard disk crash or failure, the data is not lost. Tape backup can be done manually or with some appropriate software can be programmed to do automatically.

Tape backup systems exist for needs ranging from backing up the hard disk on a personal computer to back up large amounts of storage for archiving and disaster recovery purposes in a large enterprise as part of a storage area network (SAN), usually combining a hardware and software package. Tape backup also includes the ability to restore data that has back up back to hard disk storage devices when needed. Tape backup is a very popular medium for larger amounts of data.

### 3. Network/Direct Attached Storage

Network and disk attached storage are solutions that are more expensive. These solutions are more robust than external hard drives as they ensure that the backup is maintained in another machine across the network. NAS/DAS solutions are generally not encrypted they are designed to operate over a slow internet link so this means a fast network is required to allow the backup to complete in a reasonable time. NAS backup solutions are not designed for offsite backup, so the user need to manually transport data away from the primary storage location.

### 4. Direct Attached Storage

DAS evolved from server where server vendors have traditionally

sold storage as an add-on. DAS is an appropriate choice for Very low-end PC applications, Very high-end high-performance mainframe applications and certain compute intensive and high performance OLTP database applications.

## B. Proposed System

Using the following two cloud systems, the system is made. The following are the two cloud types considered to describe the process, Google App engine and Amazon S3.

### 1. Architecture of Google App Engine

Google App Engine shown in the fig. 1 (often referred to as GAE or simply App Engine) is a Platform as a Service (PaaS), cloud computing platform to develop and host web applications in Google-managed data centers [3]. App Engine offers automatic scaling for web applications as the number of requests increases for different applications; App Engine automatically allocates more resources for the web application that automatically handles the additional demand [4].

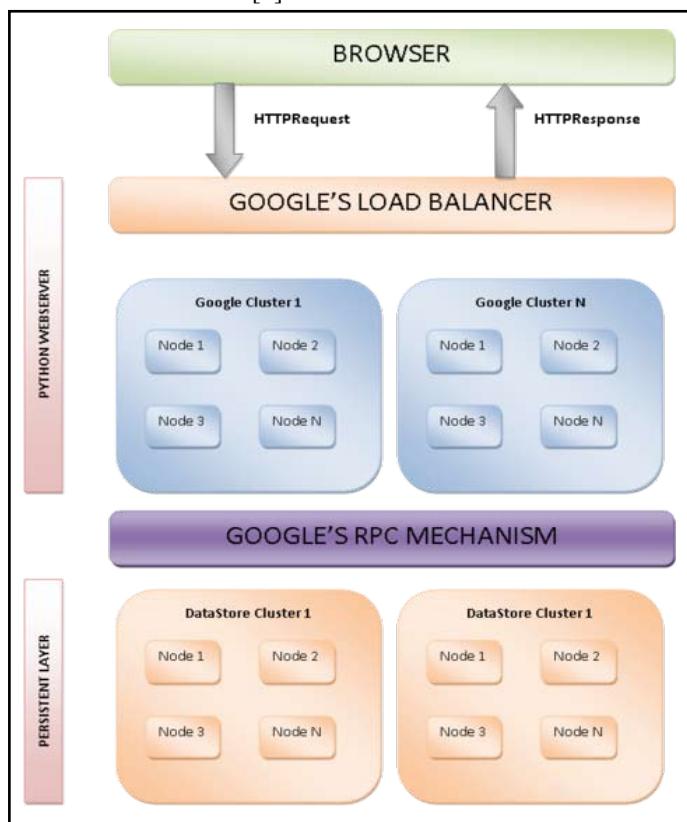


Fig.1: Architecture of Google App Engine

Google App Engine is free for certain level of consumed resources. Fees is charged for additional storage, instance hours or bandwidth required by the application.

### 2. Architecture of Amazon S3

Amazon S3 is storage for the Internet. S3 is designed to make web-scale computing easier for developers. It provides a simple web-services interface that can be used to store and retrieve any amount of data anytime anywhere on the web. S3 gives any developer access to the same highly scalable, reliable, fast, secure, inexpensive infrastructure that Amazon uses to run its own global network of websites. The service always aims to maximize the benefits of scale and to pass those benefits on to developers [5].

### Advantages to Amazon S3

Amazon S3 is intentionally built with a minimal feature set that focuses on simplicity and robustness. The main advantages of the Amazon S3 service are:

Create Buckets – Create and name a bucket that stores data. Buckets are the fundamental container in Amazon S3 for data storage.

Store data in Buckets – Store an infinite amount of data in a bucket. Upload as many objects as you like into an Amazon S3 bucket. Each object can contain up to 5 TB of data. Each object is stored and retrieved using a unique developer-assigned key.

Download data – Download your data any time you like or allow others to do the same.

Permissions – Grant or deny access to others who want to upload or download data into your Amazon S3 bucket. Grant upload and download permissions to three types of users. Authentication mechanisms ensure that data is kept secure from unauthorized access.

Standard interfaces – Use standards-based REST and SOAP interfaces designed to work with any Internet-

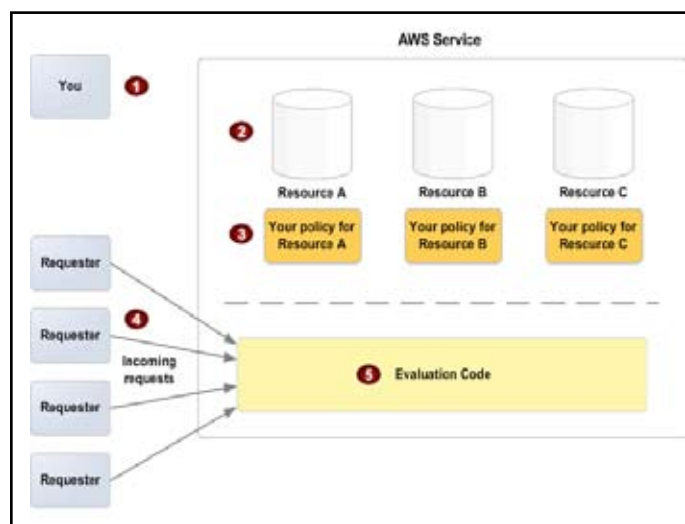


Fig. 2: Architecture of Amazon s3

S3 is designed to provide almost 99.9999% durability and 99.99% availability of the objects over a given year, though there is no SLA for durability.

Amazon Machine Images which are used in the Elastic Compute Cloud - EC2 can be exported to S3 as bundles[7].

Buckets and objects can be listed, created, and retrieved using either a REST-style HTTP interface or a SOAP interface.

Requests are authorized using an access control list associated with each bucket and object. Bucket names are addressable using the url <http://s3.amazonaws.com/bucket/key> and keys by <http://bucket.s3.amazonaws.com/key> <http://bucket/key>.

Because objects are accessible by unmodified HTTP clients, Amazon S3 can be used to replace significant existing (static) web hosting infrastructure[8]. The Amazon AWS Authentication mechanism allows the bucket owner to create an authenticated URL with time-bounded validity.

### II. Methodology

By using these two clouds for sample, this system will connect to these two from single software and has few back up techniques. Using the algorithm of Diff-Utils by “Dmitry Naumenko” to find

the difference between two files and the difference is stored on the cloud which results in saving the cloud storage [7][9].

**A. Full Backup**

The following block diagram fig. 3. Explains the full backup mechanism.

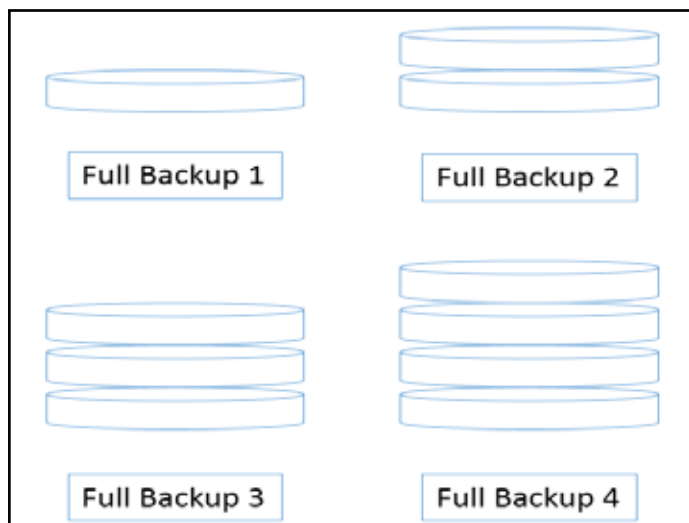


Fig. 3: Illustration of Full Backup Technique

In this mechanism restoring is done swiftly. All the back-up data is stored in a single file (better storage management). But Backing up will be slow compared to other backup types. It requires large network bandwidth.

**B. Incremental Backup**

Store backups by organizing the data into increments of change between points in time. It backups faster since it has only increments. Requires Less storage space compared to other types. Each backup increment will store a different version for a file/ folder [8][5].

Full restore will be slow compared to other backup types. (You will need the first full backup and all increments since then). To restore a latest version of an individual file the increment that contains it must be found first. The fig. 4 explains the incremental backup.

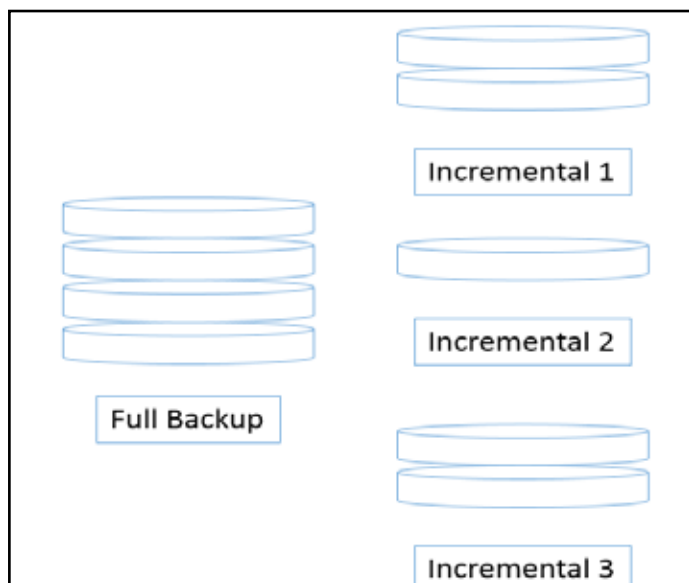


Fig. 4: Illustration of Incremental Backup Technique

**C. System Perspective**

- Creating a user profile.
- Creating buckets in the cloud to hold the objects.
- Providing full back up/ incremental backup option to the user.
- Converting the files in local disk to objects that are stored in those buckets.
- Creating a new version of the object if it already exists in that bucket.
- Restoring the objects as files in local disk.

**1. Creating User Profile**

We have used Jets3t Java toolkit to communicate with cloud. We need login credentials (access key & secret key) to be connected to the cloud. These credentials are stored as AwsCredentials objects / googleCredentials objects[6].

```
String awsAccessKey = "YOUR_AWS_ACCESS_KEY";
String awsSecretKey = "YOUR_AWS_SECRET_KEY";
AWSCredentials awsCredentials =
    new AWSCredentials(aws_AccessKey, aws_SecretKey);
```

```
To communicate with S3, create a class that implements an
S3Service
S3Service s3Service = new
    RestS3Service(awsCredentials);
```

**2. Creating a Bucket**

To store data in S3 we must first create a bucket, a container for objects.

```
S3Bucket testBucket = s3Service.createBucket("test-bucket");
We use S3Object/GoogleObject class to represent data objects
in S3/Google.
S3Object object = new S3Object(filename);
```

**3. Storing and Retrieving Objects**

```
putObject method is used to upload objects into the bucket :
s3Service.putObject(testBucket, object);
SimpleThreadedStorageService is used to restore multiple objects
as files to our local disk
SimpleThreadedStorageService simpleMulti = new
    SimpleThreadedStorageService (s3Service);
```

**4. Downloading Objects**

```
DownloadPackage is created for each object, to associate the
object with an output file.
DownloadPackage[] downloadPackages = new
DownloadPackage[k];
downloadPackages[i] = new DownloadPackage (objects[i],
new File(objects[i].getKey()));
Then objects are downloaded using these download packages:
simpleMulti.downloadObjects(bucket,downloadPackages);
```

**5. Deleting the objects and Buckets**

```
Multiple objects and buckets are deleted using delete functions.
simpleMulti.deleteObjects(bucket, objects);
s3Service.deleteBucket(bucket);
```

**6. Creating Versions for Full backup**

A new version of file/folder is created irrespective of changes

occurred in them.

This new file/folder has its version number added to the end of its filename. Even when there is a change only in one file of the folder given for backup, the complete folder will be backed up.

### 7. Creating Version for Incremental backup

Whenever first time a new file/folder is given for backup, a full backup is performed. Whenever an existing file/folder is given for backup, its difference from the previous version file/folder is stored in the cloud. These differences are stored as an object (Patch object). These objects are stored in files and are uploaded on to the cloud .

### 8. Restoring: Full backup

The user selects a file from his backup and restores it. The user can restore any version of the file available

### 9. Restoring: Incremental Backup

The original file and its successive diffs are merged together to obtain a latest version of the file.

## III. Results and Discussion

User first selects the cloud type – Amazon S3 or Google App Engine. Depending upon the backing type user selects, the data is stored on the select cloud. The later can be retrieved from anywhere with these key and bucket specification. This can also be implemented for finding the difference between two files of different sizes of same but different versions. The different one is stored on the cloud when incremental backup option is selected. System displays the files, which are already on the cloud, and user can select the file or files and download them. This process is different for different cloud. Mixing of cloud files is also possible according to the convenience of the user [10] [11].

Finally, single system or software designed can connect to two different clouds with useful features for the convenience of the user. User might be small company or an individual who wants to store the data on cloud for safety and remote availability and cheaper when incremental is used. Sometimes incremental shows the difference for version to version difference work processed on the file. Diff-Utils created file is non-human readable content thus, it is secured sometimes and only this system can revert and connect the versions and show in normal form.

## IV. Conclusion

The implementation of this software can help users to backup their data with less consumption of storage and in secured way. Remote access of data anytime with version-to-version files. This would help connecting different clouds and enable user to maintain connection with different circles. The paper concludes how user can download the files from any remote place, irrespective of type of file with no limitation on space.

## References

- [1] Mariana Carroll, Paula Kotzé, Alta van der Merwe (2012). "Securing Virtual and Cloud Environments". *Research and Innovations in the Service Economy*. Springer Science+Business Media.
- [2] "Cloud Computing entry". *NetLingo*. Retrieved 15 January 2014.
- [3] "Python Runtime Environment - Google App Engine - Google Code". *Code.google.com*. 1999-02-22. Retrieved 2012-02-

14

- [4] Sanderson, Dan (2009). *Programming Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure*.
- [5] Amazon web services <http://aws.amazon.com/s3/>
- [6] Amazon S3 Protecting Your Data.
- [7] Starting Websphere in Cloud and saving the data in S3
- [8] How to use Amazon S3 for Web Hosting.
- [9] Dmitry Naumenko – java-diff-utils.
- [10] Leon Brocard <acme@astray.com> and unknown Amazon Digital Services programmers.
- [11] Brad Fitzpatrick <brad@danga.com> - return values, Bucket object