**CEPH :**

In computing, Ceph is a free-software storage platform, implements object storage on a single distributed computer cluster, and provides interfaces for object-, block- and file-level storage.

Ceph is an emerging technology, it's providing an integrated, well organized distributed file store.

Performance : Striped data over data servers .

Reliability : There's no single point of failure.

Scalability : it has adaptive metadata clusters.

Ceph storage clusters are designed to run on commodity hardware, using an algorithm called CRUSH (Controlled Replication Under Scalable Hashing) to ensure data is evenly distributed across the cluster and that all cluster nodes can retrieve data quickly without any centralized bottlenecks.

Ceph object storage is accessible through Amazon Simple Storage Service (S3) and OpenStack Swift Representational State Transfer (REST)-based application programming interfaces (APIs), and a native API for integration with software applications.

Ceph components:

● Clients : The clients are all sort of operating on this file system. They

actually stream the data and get the data.

● Metadata Cluster : providing all the directory information. It's mapping the names of our files down into the storage, and identifying where all the objects, what all the parts of the objects are, in our clusters.

● object storage cluster: It's, again, a distributed system, so if we want a larger storage metadata increases as we add to storage cluster.

Ceph storage:

Ceph file storage makes use of the Portable Operating System Interface (POSIX)-compliant Ceph file system (CephFS) to store data in a Ceph Storage Cluster. CephFS uses the same clustered system as Ceph block storage and Ceph object storage.

Dynamic distribution of metadata management system :

● Metadata is split among cluster of servers

● Distribution of metadata changes with the number of requests to even load among metadata servers

● Metadata servers also can quickly recover from failures by taking over neighbors’ data

● Improves performance by leveling metadata load.

**HIVE:**

Apache Hive is a data warehouse system for data summarization, analysis and querying of large data systems in open source Hadoop platform. It is exclusively used to query and analyze huge data sets stored in the Hadoop storage.

HQL is query language for HIVE that is a variant of SQL. It translates queries into map/reduce jobs and this can run on top of Hadoop YARN,Tez, spark.

This is a read only language since update and delete cannot be performed.

The most important file systems that are supported by Hive are:

* Flat Files or text files
* Sequence Files consisting of binary key/value pairs
* RCFiles that stores columns of table in a columnar database.

Some of the major components of Hive architecture are:

* Metastore: It is the repository for metadata. This metadata consists of data for each table like its location and schema. It also holds the information for partition metadata which lets you monitor the various distributed data progress in the cluster. This data is generally present in the relational databases. The metadata keeps track of the data, replicates the data and provides a backup in case of data loss.
* Driver: The driver receives the HiveQL statements and works like a controller. It monitors the progress and life cycle of various executions by creating sessions. When a HiveQL statement is executed the Driver stores the metadata generated while executing the statement. When the reducing operation is completed by the MapReduce job the driver collects the data points and query results.
* Compiler: The Compiler is assigned with the task of converting the HiveQL query into MapReduce input. It includes a method to execute the steps and tasks needed to let the HiveQL output as needed by the MapReduce.
* Optimizer: This performs the various transformation steps for aggregating, pipeline conversion by a single join for multiple joins. It also is assigned with the task of splitting a task while transforming the data before the reduce operations for improved efficiency and scalability.
* Executor: The Executor executes the tasks after the compilation and the optimization steps. The Executor directly interacts with the Hadoop Job Tracker for scheduling of tasks to be run.
* CLI, UI, and Thrift Server: the Command Line Interface and the User Interface submits the queries, process monitoring and instructions so that external users can interact with Hive. Thrift lets other clients to interact with Hive.

Physical layout :

* Warehouse directory in HDFS

E.g., /user/hive/warehouse

* Tables stored in subdirectories of warehouse

Partitions form subdirectories of tables allows us to access pieces of tables that are on diff machines.

* Actual data stored in flat files

Control char-delimited text, or Sequence Files.

Can be customized to use arbitrary format.

**TEZ :**

Apache Tez is an extensible framework for building high performance batch and interactive data processing applications, coordinated by YARN in Apache Hadoop. Tez improves the MapReduce paradigm by dramatically improving its speed, while maintaining MapReduce's ability to scale to petabytes of data.

What Tez Does

Apache Tez provides a developer API and framework to write native YARN applications that bridge the spectrum of interactive and batch workloads. It allows those data access applications to work with petabytes of data over thousands nodes. The Apache Tez component library allows developers to create Hadoop applications that integrate natively with Apache Hadoop YARN and perform well within mixed workload clusters.

Since Tez is extensible and embeddable, it provides the fit-to-purpose freedom to express highly optimized data processing applications, giving them an advantage over end-user-facing engines such as MapReduce and Apache Spark. Tez also offers a customizable execution architecture that allows users to express complex computations as dataflow graphs, permitting dynamic performance optimizations based on real information about the data and the resources required to process it.

Execution Performance

○ Performance gains over Map/Reduce

○ Optimal resource management

○ Plan re-configuration at runtime to get back the maximum number of nodes.

○ Dynamic physical data flow decisions and get the resources, the data, algorithm to the resources.

Empowering end users by:

○ Expressive dataflow definition APIs and providing them an interface on top of it to use it .

○ Flexible Input-Processor-Output runtime model

○ Data type agnostic that allows us to treat all the data types in a similar way

○ Simplifying deployment that is the reason why allocating resources to this is made easier.

**SWIFT:**

The OpenStack Object Store project, known as Swift, offers cloud storage software so that you can store and retrieve lots of data with a simple API. Swift is ideal for storing unstructured data that can grow without bound.

Swift partitions :

It starts with a string, swift takes that string and maps into things called partitions on the disks and then find where those disks are and what needs to be retrieved.

USE CASE :

○ When we wish to store unstructured data like images, binary data we use lot amount of data on disks. So if we move such data into BLOB and access it from there then it would be a better idea.

* Binary Large OBject (BLOB) is collection of binary data stored as single entity in a dbms.

○ This blobs are easily accessible over web and handles to the data, we use pointers to the data, URL in general when we click over the URL we could get the data.

○ Examples include Windows azure blob storage, ambry(Linkedin), facebook’s warm blob storage system, apache open stack blob service(SWIFT).

**AMAZON S3 BLOB STORAGE:**

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% of durability, and stores data for millions of applications for companies all around the world.

S3 - Simple Storage Service :

A bucket (it has flat directory structure) is a container for objects and describes location,

logging, accounting, and access control.

There are several operations corresponding to HTTP actions:

Post: to create a new object or update existing.

Get: to get obj from bucket.

Delete: to remove an obj from bucket.

List: To list keys present in bucket with a filter.

**AMAZON AWS EBS STORAGE:**

Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes offer the consistent and low-latency performance needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes – all while paying a low price for only what you provision.

Amazon EBS is designed for application workloads that benefit from fine tuning for performance, cost and capacity. Typical use cases include Big Data analytics engines (like the Hadoop/HDFS ecosystem and Amazon EMR clusters), relational and NoSQL databases (like Microsoft SQL Server and MySQL or Cassandra and MongoDB), stream and log processing applications (like Kafka and Splunk), and data warehousing applications (like Vertica and Teradata).

**AMAZON S3 GLACIER:**

Amazon Simple Storage Service Glacier, that is Amazon S3 Glacier (Glacier), is a storage service optimized for infrequently used data, or "cold data."

Glacier is an extremely low-cost storage service that provides durable storage with security features for data archiving and backup. With Glacier, customers can store their data cost effectively for months, years, or even decades. Glacier enables customers to offload the administrative burdens of operating and scaling storage to AWS, so they don't have to worry about capacity planning, hardware provisioning, data replication, hardware failure detection and recovery, or time-consuming hardware migrations.

What is Glacier—The rest of this section describes the underlying data model, the operations it supports, and the AWS SDKs that you can use to interact with the service.

Advantages :

* Very low cost, $0.007 per GB per month
* Very durable and reliable
* Each single archive up to 40 TB and these archives are stored in vaults.
* The main access point to glacier is S3
* Typically takes between 3 to 5 hours to prepare a download request
  + After that we have 24 hours to download from the staging location.

**AMAZON ELASTIC FILE SYSTEM:**

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it. Amazon EFS has a simple web services interface that allows you to create and configure file systems quickly and easily. The service manages all the file storage infrastructure for you, meaning that you can avoid the complexity of deploying, patching, and maintaining complex file system configurations.

Amazon EFS supports the Network File System version 4 (NFSv4.1 and NFSv4.0) protocol, so the applications and tools that you use today work seamlessly with Amazon EFS. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance or server.

With Amazon EFS, you pay only for the storage used by your file system and there is no minimum fee or setup cost. Amazon EFS oﬀers two storage classes, Standard and Infrequent Access. The Standard storage class is used to store frequently accessed files. The Infrequent Access (IA) storage class is a lower-cost storage class that's designed for storing long-lived, infrequently accessed ﬁles cost-eﬀectively.

**DROPBOX API:**

The Dropbox Business API is an extension of the Dropbox Platform, which provides thousands of pre-integrated apps for users to create additional productivity and collaboration solutions.

Dropbox is a storage service that syncs files online and across your computers via installed software.

* It uses clouds and offers cloud storage.
* They are easily synced across multiple devices.
* Metadata is stored in dropbox server whereas the actual data is stored in Amazon s3.
* Two levels of API access to DropBox : ***Drop-ins*** 
  + It has cross-platformed components that can be integrated in minutes.
  + It allows instant access to files in Dropbox.