```
unit -2 syllabus content
************************
1.web application frame work --ajax(client),phython django (server)
2.web application  sample example-- google app

3.web api --- Google Gadgets ,. GoGrid ,apex


4.protocols --
http xmpp ---communication protocols
  ssl      --security protocol
  openID

5.web services --   json soap rest


6.Cloud storage providers Amazon S3,MobileMe,LiveMesh and Google BigTable DataStore

************************ end ************************
```

UNIT –II

Web Application Framework:

A web framework (WF) or web application framework (WAF) is software that is designed to support the development of dynamic web sites, web applications, web APIs and web services etc.

A **web application** (or web app) is application software that runs on a web server, unlike computer-based software programs that are run locally on the operating system (OS) of the device. Web applications are accessed by the user through a web browser with an active network connection

Example: Gmail, Facebook, shopping carts etc.

Web API is a programming interface/application type that provides communication or interaction between software applications.

Example: YouTube API's, Google Maps API's etc.

A **Web service** is a method of communication between two electronic devices over a network.

Example: JSON, REST, SOAP etc.

Advantages of Web Application Framework:

- No need to write additional code for common activities
- Speed up custom web application development.
- Simplify web application maintenance.
- No need to increase web development cost.
- Work with databases more efficiently.

- Protect websites from targeted security attacks
- Automate common web development tasks
- Perform unit testing efficiently

Types of Web Application Frameworks:

- Server side: Web Application Frameworks
  Ex: Django (Python), ASP.NET (C#), Ruby on Rails (Ruby)
- Client side: Web Application Frameworks
  Ex: Bootstrap, Angular.js

AJAX:

AJAX stands for "Asynchronous JavaScript and XML". It is not exactly a client-side technology, nor a server-side technology: It's both! Ajax is a technique in which websites use JavaScript (client-side) to send data to, and retrieve data from, a server-side script.

How it works:

Ajax is used for creating interactive web applications
By using AJAX, web applications can retrieve data from the server asynchronously.
Because it is being done in the background, it won't interfere with the display and
Behaviour of the current page.

What types of Technologies used in AJAX:

- Extensible Hypertext Mark-up Language (XHTML) and Cascading Style Sheets (CSS) for presentation
- The Document Object Model for dynamic display of and interaction with data
- XML and Extensible Style Sheet Language Transformations (XSLT) for the Interchange and manipulation of data
- The XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

Advantages of AJAX:

- Connections to the server are reduced, because scripts and style sheets need only be Downloaded once. So, it increases the speed.
- Form validations at client slide.
- Bandwidth usage can be reduced.
- Process will be done in the background; it won't interfere with the display and Behaviour of the current page.

Disadvantages of AJAX:

- Dynamically created web pages do not show up in the browser 's history engine, so Clicking on the Back button would not re-create the last seen page.
- It is difficult to bookmark a dynamically created web page.
- If a browser does not support AJAX or if JavaScript is disabled, AJAX functionality Cannot be used.
- There is no standards body behind AJAX, so there is no widely adopted best Practice to test AJAX applications.

Python Django: Example

Django is an open-source web application framework written in Python.

Originally it was created to manage news sites.

Django was developed to ease the creation of database-driven web sites.

It contains number of Reusability components.

Its works on two principles
DRY (Don't Repeat Yourself) and CRUD (create, read, update, and delete)

Features:

• A lightweight, stand-alone web server for development and testing
• A caching framework, which can use any of several cache methods
• An internal dispatcher system that allows an application's components to Communicate using predefined signals
• An internationalization system that translates Django's components into multiple Languages
• A scheme for extending the capabilities of the template engine

Web applications:

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface for performing the tasks.

Web applications examples: Moodle, Gmail, Google talk, Google calendar, face book etc...

How a web application works:

1. User triggers a request to the web server over the Internet, either through a web browser or the application's user interface
2. Web server forwards this request to the appropriate web application server
3. Web application server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data
4. Web application server sends results to the web server with the requested information or processed data
5. Web server responds back to the client with the requested information that then appears on the user's display

Sample Application:

Google Apps launched as free service in august 2006

It includes:

- Gmail webmail services
- Google Calendar shared calendaring
- Google Talk instant messaging and Voice Over IP
- Start Page for creating a customizable home page on a specific domain
- Google Docs and Spreadsheets
- Gmail for mobile devices

Google Apps Premier Edition has the following unique features:

- **Per-user storage of 10GBs** offers about 100 times the storage of the average Corporate mailbox, eliminating the need to frequently delete email.
- **APIs for business integration** APIs for data migration, user provisioning, single Sign-on and mail gateways enable businesses to further customize the service for Unique environments.
- **Uptime of 99.9 percent** Service Level Agreements for high availability of Gmail, With Google monitoring and crediting customers if service levels are not met.
- **Support for critical issues 24/7** Includes extended business hours telephoneSupport for administrators.

- **Advertising optional** Advertising is turned off by default, but businesses can Choose to include Google's relevant target-based ads if desired.
- **Low fee** Simple and affordable annual fee (US$50 per user account per year)
- Makes it practical to offer these applications to everyone in the organization.

WEB API

Abbreviation: Application Programming Interface

Definition: An application programming interface (API) is a set of programming instructions and standards for accessing a web-based program

APIs allow one program to speak with another.

Software companies release their APIs to the public so that other software developers can design products that are powered by its service.

Example: when you buy something at Amazon and enter your credit card information, Amazon uses an API to send your credit card information to a remote application that verifies whether your information is correct

Who uses API?

These services can be accessed by different kind of users like:

- Web Browsers
- Mobile applications
- Desktop applications
- IOTs (Internet of Things)

How APIs Work:

An API works in between two pieces of software to exchange information.

1. User triggers a request to the recipient web server over the Internet, either through a web browser or the application's user interface
2. Web API forwards this request to the appropriate web server

3. Web server forwards this request to the appropriate web application server for processing the client data.
4. Web application server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data
5. Web application server sends results to the web server with the requested information or processed data
6. Web server responds back to the client via **WEBAPI** with the requested information that then appears on the user's display

What types of Technologies used in WEBAPI:

- **XML:** XML is a general-purpose markup language. . It describes structured data in a way that both humans and computers can read and write.

- **SOAP (Simple Object Access Protocol)** SOAP encodes XML messages so that they can be received and understood by any operating system over any type of network protocol.

- **UDDI (Universal Description, Discovery, and Integration)** UDDI is an XMLbased directory that allows businesses to list themselves, find each other, and collaborate using web services

- **WSDL (Web Services Description Language)** WSDL is the SOAP of UDDI. WSDL is the XML-based language that businesses use to describe their services in the UDDI

API Creators

There are many different APIs you can use to link your organization with your cloud applications.

1. Google Gadgets:

Google Gadgets are a desktop search application that enables users to search their email, files, web history, and chats. Called Google Desktop Search, this new application makes it possible for users to find information on their computers as fast and easily as they can search the Web with Google.

The Google Gadgets API is composed of three languages:

XML: This is the language you use to write gadget specifications

HTML: HTML is used to format web documents,

JavaScript: JavaScript is the scripting language you can use to add dynamic behavior to your gadgets.

Google Desktop Search is a lightweight, free, downloadable application that brings Google search to information on your computer. The application operates locally on the user's computer, where it provides the following capabilities:

System-wide search Users can search across their email and a wide range of files and information such as email in Microsoft Outlook and Outlook Express; files in Microsoft Word, Microsoft Excel, Microsoft PowerPoint, and text; web site history in Internet Explorer; and instant message chats in AOL Instant Messenger.

High search speed Google.com can search billions of web pages in a fraction of a second. Google Desktop Search is built with the same technology, and it can search a single hard drive in even less time.

Easy access to desktop results via Google.com Google Desktop Search enables users to search both their computer and the Web simultaneously.

Dynamic results Unlike traditional computer search software that updates once a day, Google Desktop Search updates continuously for most file types. When a user downloads a new email in Outlook, for example, it can be found within seconds using Google Desktop Search.

Examples of Google Data APIs:

Google Calendar Data API, Google Spreadsheets Data API, YouTube Data API etc..

2. GoGrid:

GoGrid's API that allows developers to control their interaction with GoGrid's cloud hosting infrastructure. The GoGrid API provides two-way communication for controlling GoGrid's control panel functionality.

Typical uses for the API include

- Auto-scaling network servers
- Listing assigned public and private IP addresses
- Deleting servers
- Listing billing details

GoGrid's REST-like API Query interface is designed for individuals who want to Programmatically control their cloud hosting infrastructure over the Internet.

The GoGrid API requires you to be a GoGrid customer and to have technical knowledge and programming skills. The GoGrid API supports these languages:

• Java • PHP • Python • Ruby

3. APEX:

Apex includes an application program interface (API) that developers can use to access user data on Salesforce.com.

The Apex platforms consist of three tools:

• **Apex Builder** An on-demand component allowing easy drag-and-drop Customization with a limited set of features.

• **Apex API** A method of retrieving raw data from Salesforce.com's servers. The API is used by programs that are external to Salesforce.com, like Java applications that need access to information on a client's Salesforce.com account.

• **Apex Code** A programming language that is executed on Salesforce.com's servers. The Apex Code offers flexibility in developing by using the Apex API while reducing the number of calls between the client and server.

Advantages

- No need for buying the infrastructure and hire IT people as all the facilities are provided by Salesforce.
- Cost-effective, as the cost of maintenance of the application, the licensing cost of various software is quite high than buying the monthly subscription of Salesforce.
- Security of application, full Authentication and Authorization is provided to the end-users by Salesforce for accessing the application.
- It is easy to develop an application in the Salesforce.com platform for developers as developers can use the existing applications or make changes in the existing ones according to business needs.
- Apex allows its developers to access the Salesforce.com backend database to third-party Saas applications.

Standards

Protocol:

A **protocol** is a standard set of rules that allow electronic devices to communicate with each other.

There are two imp types of communication protocols are used in cloud computing for accessing cloud data from client machine and vice versa.

COMMUNICATION PROTOCALS: HTTP and XMPP

HTTP:

Hypertext Transfer Protocol

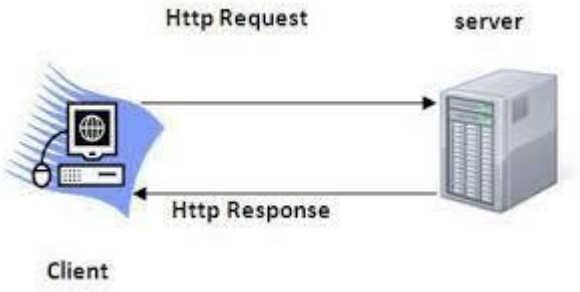
Usage: for accessing cloud data from client machine and vice versa.

HTTP is a stateless protocol.

A stateless protocol does not require the server to retain information or status about each user for the duration of multiple requests.

HTTP/1.0 uses a new connection for each request/response exchange, where as HTTP/1.1 connection may be used for one or more request/response exchanges and transfers all the page's components without hanging up and opening new sessions.

How it works :



Basic Features

HTTP is connectionless: The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server know about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.

HTTP is media independent: It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content.

HTTP is stateless: A stateless protocol does not require the server to retain information or status about each user for the duration of multiple requests.

Request types in HTTP

Request	Description
HEAD	Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is good for retrieving metainformation in the response headers, but without transporting the entire content.
GET	Requests information from a server.
POST	Submits data to be processed to the server. The data is included in the body of the request. The result of the request might be the creation of the resource or updating the existing resource.
PUT	Uploads a representation of the resource.
DELETE	Deletes the specified resource.
TRACE	Echoes the request back to the browser so that the client can see which servers are adding or changing in the request.
OPTIONS	Returns HTTP methods that the server supports for the given URL. This can be used to check the functionality of a web server.
CONNECT	Converts the request connection to a transparent TCP/IP tunnel. It's usually used to facilitate SSL-encrypted communication through an unencrypted HTTP proxy.

TABLE 8-1 The Different Requests in HTTP

XMPP:

The Extensible Messaging and Presence Protocol (XMPP)

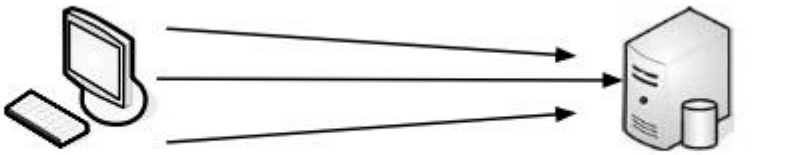
XMPP is a open source protocols

It is also known as jabber/xmpp

XMPP was originally developed in the instant messages services.

SOAP and other HTTP-based protocols—are all one-way information exchanges.

XMPP allows for two-way communication and eliminates polling.



HTTP requires multiple polling events to update status from the web browser.

XMPP maintains a connection between the client and the web server.

Customers:
Google, Apple, AOL, IBM, and Live Journal

XMPP offers several key advantages over such services:

Open — the XMPP protocols are free, open, public, and easily understandable

Standard — the Internet Engineering Task Force (IETF)

Proven - there are tens of thousands of XMPP servers running on the Internet today, and millions of people use XMPP for instant messaging through public services such as Google Talk and XMPP deployments at organizations worldwide.
Decentralized — the architecture of the XMPP network is similar to email; as a result, anyone can run their own XMPP server, enabling individuals and organizations to take control of their communications experience.

Secure – It follows end to end encryption techniques.

Extensible — using the power of XML, anyone can build custom functionality on top of the core protocols

Diverse — a wide range of companies and open-source projects use XMPP to build and deploy real-time applications and services

Disadvantages of XMPP

• The main disadvantage of XMPP is the redundancy of the transmitted data. Around 60% of all information transmitted by the protocol is the presence data, which creates the excess traffic.

Secure Sockets Layer (SSL)

SSL is the standard security technology for establishing an encrypted link between a web server and browser. This ensures that data passed between the browser and the web server stays private.

Standards

Secure Sockets Layer (SSL)

SSL is the standard security technology for establishing an encrypted link between a web server and browser. This ensures that data passed between the browser and the web server stays private.
To create an SSL connection on a web server requires an SSL certificate.
The cloud provider's computers then generate two cryptographic keys—a public key and a private key.

SSL Connection
1. The browser checks the web site's certificate to ensure that the site you are connecting to is the real site and not someone else intercepting and spoofing the site.
2. The browser and web site decide on what type of encryption to use.
3. The browser and server send each other unique codes to use when encrypting information to be sent.
4. The browser and server use the encryption to start talking.
5. The browser shows the encrypting icon, and web pages are passed as secured.

Open ID:

OpenID is an open standard and decentralized authentication protocol

Users create accounts by selecting an OpenID identity provider and then use those accounts to sign onto any website that accepts OpenID authentication.

OpenID allows you to use an existing account to sign in to multiple websites, without needing to create new passwords.

It works on OAuth 2.0 protocol.

Please download below topics material from web: (10 features)

****

- HTML features
- CSS features
- DHTML features
- JAVA SCRIPT features
- **XML features (imp)**

Web services

Definition: A web service, as defined by the World Wide Web Consortium (W3C), "is a software system designed to support interoperable machine-to-machine interaction over a network" that may be accessed by other cloud computing components.

Web services components:

- REST
- SOAP
- JSON

JSON :

JSON: **J**ava**S**cript **O**bject **N**otation.

JSON is a syntax for storing and exchanging data.

JSON is text, written with JavaScript object notation.

JSON is a lightweight data-interchange format

JSON is "self-describing" and easy to understand

JSON is language independent

Exchanging Data

When exchanging data between a browser and a server, the data can only be text.

Sending Data

If you have data stored in a JavaScript object, you can convert the object into JSON, and send it to a server:

Example

```
var myObj = {name: "John", age: 31, city: "New York"};
var myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;
```

Receiving Data

If you receive data in JSON format, you can convert it into a JavaScript object:

Example

```
var myJSON = '{"name":"John", "age":31, "city":"New York"}';
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
```

Storing & Retrieving Data

When storing data, the data has to be a certain format, and regardless of where you choose to store it, *text* is always one of the legal formats.

JSON makes it possible to store JavaScript objects as text.

Example

Storing data in local storage

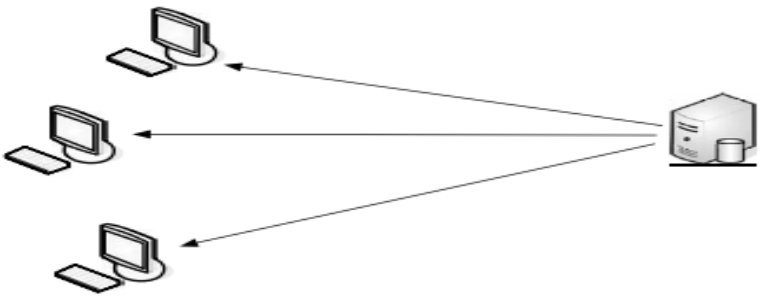
```
// Storing data:
myObj = {name: "John", age: 31, city: "New York"};
myJSON = JSON.stringify(myObj);
localStorage.setItem("testJSON", myJSON);

// Retrieving data:
text = localStorage.getItem("testJSON");
obj = JSON.parse(text);
document.getElementById("demo").innerHTML = obj.name;
```

SOAP

- SOAP stands for Simple Object Access Protocol
- SOAP is an XML based protocol for accessing Web Services.
- SOAP is based on XML
- SOAP is a W3C recommendation

REST

- REST stands for Representational State Transfer.
- REST is an architectural style.
- REST is based on XML
- REST is a W3C recommendation
- REST uses all HTTP Request types (GET, POST etc ...)



Clients send a request to the web server for information, using the same URL. The web site has updated its content, and uses REST to send the information back to the clients.

No.	SOAP	REST
1)	SOAP is a **protocol**.	REST is an **architectural style**.
2)	SOAP stands for **Simple Object Access Protocol**.	REST stands for **REpresentational State Transfer**.
3)	SOAP **can't use REST** because it is a protocol.	REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP.
4)	SOAP **uses services interfaces to expose the business logic**.	REST **uses URI to expose business logic**.
5)	**JAX-WS** is the java API for SOAP web services.	**JAX-RS** is the java API for RESTful web services.
6)	SOAP **defines standards** to be strictly followed.	REST does not define too much standards like SOAP.
7)	SOAP **requires more bandwidth** and resource than REST.	REST **requires less bandwidth** and resource than SOAP.
8)	SOAP **defines its own security**.	RESTful web services **inherits security measures** from the underlying transport.
9)	SOAP **permits XML** data format only.	REST **permits different** data format such as Plain text, HTML, XML, JSON etc.
10)	SOAP is **less preferred** than REST.	REST **more preferred** than SOAP.

Cloud Storage Providers

- Amazon S3**(very imp)**
- Google Big table Data store
- Mobile Me
- Live Mesh

Amazon S3: Amazon Simple Storage Service (Amazon S3)

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 data likes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

Design Requirements in Amazon S3:

- **Scalable**

  Amazon S3 can scale in terms of storage, request rate, and users to support an unlimited number of web-scale applications.

- **Reliable**

  Store data durably, with 99.99 percent availability. Amazon says it does not allow any downtime.

- **Fast**

  Amazon S3 was designed to be fast enough to support high-performance applications. Server-side latency must be insignificant relative to Internet latency. Any performance bottlenecks can be fixed by simply adding nodes to the system.

- **Inexpensive**

  Amazon S3 is built from inexpensive commodity hardware components.

- **Simple**

  Building highly scalable, reliable, fast, and inexpensive storage is difficult. Doing so in a way that makes it easy to use for any application anywhere is more difficult. Amazon S3 must do both.

Design Principles:

• **Decentralization** It uses fully decentralized techniques to remove scaling bottlenecks and single points of failure.

• **Autonomy** The system is designed such that individual components can make decisions based on local information.

• **Local responsibility** each individual component is responsible for achieving its consistency;

• **Controlled concurrency** Operations are designed such that no or limited concurrency control is required.

• **Failure toleration** The system considers the failure of components to be a normal mode of operation and continues operation with no or minimal interruption.

• **Controlled parallelism** parallelism can be used to improve performance and robustness of recovery or the introduction of new nodes.

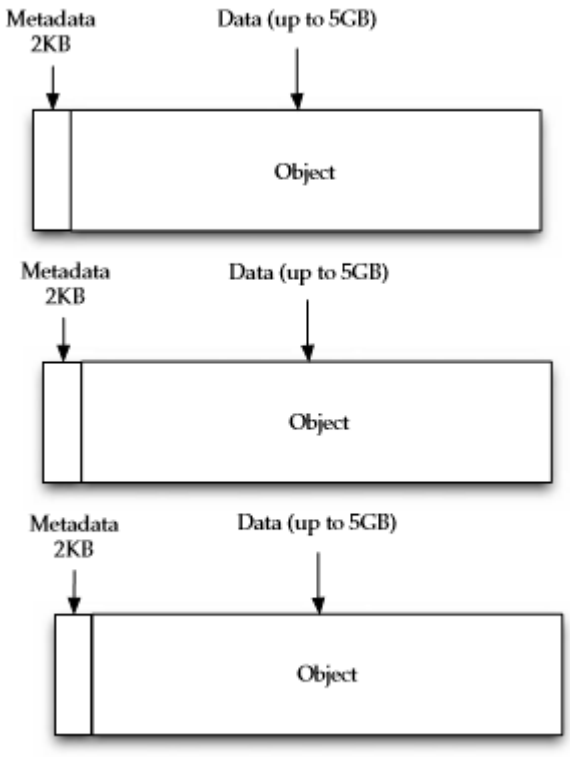
• **Small, well-understood building blocks** Do not try to provide a single service that does everything for everyone, but instead build small components that can be used as building blocks for other services.

• **Symmetry** Nodes in the system are identical in terms of functionality, and require no or minimal node-specific configuration to function.

• **Simplicity** The system should be made as simple as possible.

How S3 Works

Amazon, S3's design aims to provide scalability, high availability, and low latency at commodity costs. S3 stores arbitrary objects at up to 5GB in size, and each is accompanied by up to 2KB of metadata. Objects are organized by buckets. Each bucket is owned by an AWS account and the buckets are identified by a unique, user-assigned key.



Multiple objects are stored in buckets in Amazon S3.

Buckets and objects are created, listed, and retrieved using either a REST-style or SOAP interface. Objects can also be retrieved using the HTTP GET interface or via Bit Torrent. An access control list restricts who can access the data in each bucket.

To upload your data (photos, videos, documents etc.) to Amazon S3, you must first create an S3 bucket in one of the AWS Regions. You can then upload any number of objects to the bucket.

Security:

Amazon S3 supports both server-side encryption (with three key management options (SSE-KMS, SSE-C, SSE-S3) and client-side encryption for data uploads. Amazon S3 offers flexible security features to block unauthorized users from accessing your data.

Cloud Storage Providers
Google Big table Data store
Mobile Me
Live Mesh

Google Bigtable Datastore

Google Bigtable Datastore

- **Cloud Bigtable** is a sparsely populated table
- It can scale upto billions of rows and thousands of columns
- Enables to store TB or even PB of data.
- A single value in each row is indexed
- This value is known as the row key.
- It is possible to store terabytes or even petabytes of data in Google Cloud BigTable

- The row key is the lone index value that appears in every row and is also known as the row value.
- Low-latency storage for massive amounts of single-keyed data is made possible by Google Cloud Bigtable.
- It is the perfect data source for MapReduce processes since it enables great read and write throughput with low latency.
- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
- Applications can access Google Cloud BigTable through a variety of client libraries, including supported Java extension to the Apache HBase library.
- Because of this, it is compatible with the current Apache ecosystem of open-source big data software.
- Applications that require high throughput and scalability for key/value data, where each value is typically no more than 10 MB, should use Google Cloud BigTable.
- Additionally, Google Cloud Bigtable excels as a storage engine for machine learning, stream processing, and batch MapReduce operations.

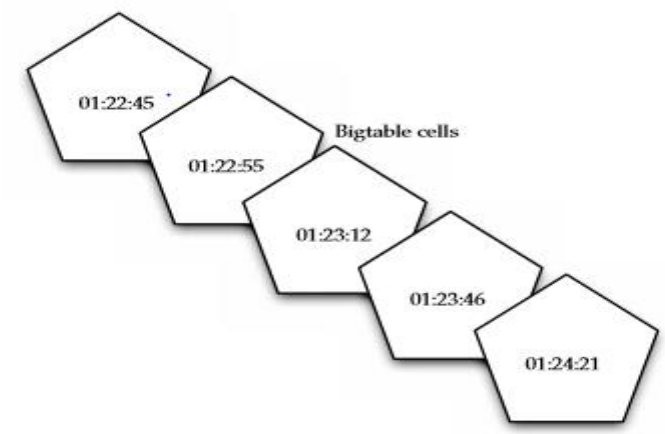
BigTable Storage Concept

- Each massively scalable table in Google Cloud Bigtable is a sorted key/value map that holds the data.
- The table is made up of columns that contain unique values for each row and rows that typically describe a single object.
- A single row key is used to index each row, and a column family is often formed out of related columns.
- The column family and a column qualifier, a distinctive name within the column family, are combined to identify each column.
- Multiple cells may be present at each row/column intersection.
- A distinct timestamped copy of the data for that row and column is present in each cell.
- When many cells are put in a column, a history of the recorded data for that row and column is preserved.
- Cloud by Google Bigtable tables is sparse, taking up no room if a column is not used in a given row.
- Few points to remember Rows of columns could be empty.

Google Bigtable Datastore

All of the following forms of data can be stored in and searched using Google Cloud Bigtable:

- **Time-series information,** such as CPU and memory utilization patterns across various servers.
- **Marketing information,** such as consumer preferences and purchase history.
- Financial information, including stock prices, currency exchange rates, and transaction histories.
- **Internet of Things data,** such as consumption statistics from home appliances and energy meters.
- Graph data, which includes details on the connections between users.



In Google Bigtable, multiple copies of a cell exist, each with a different timestamp.

- Google describes Bigtable as a fast and extremely scalable DBMS.
- This allows Bigtable to scale across thousands of commodity servers that can collectively store petabytes of data.
- Each table in Bigtable is a multidimensional sparse map.
- That is, the table is made up of rows and columns, and each cell has a timestamp.
- Multiple versions of a cell can exist, each with a different timestamp.
- With this stamping, you can select certain versions of a web page, or delete cells that are older than a given date and time.
- A specific row and column contain cells with individual timestamps (t).
- All client queries made through the Google Cloud Bigtable architecture are sent through a frontend server before being forwarded to a Google Cloud Bigtable node.

- The nodes are arranged into a Google Cloud Bigtable cluster, which is a container for the cluster and is part of a Google Cloud Bigtable instance.
- Because the tables are so large, Bigtable splits them at row boundaries and saves them as tablets.
- Each tablet is about 200MB, and each server houses 100 tablets.
- Given this, data from a database is likely to be stored in many different servers—(not in the same geographic location). This architecture also allows for load balancing.
- If one table is getting a lot of queries, it can remove other tablets or move the busy table to another machine that is not as busy.
- Also, if a machine fails, since the tablet is spread to different machines, users may not even notice the outage.
- When a machine fills up, it compresses some tablets using a Google-proprietary technique.
- On a minor scale, only a few tablets are compressed.
- On a large scale, entire tablets are compressed, freeing more drive space.
- Bigtable tablet locations are stored in cells, and looking them up is a three-tiered system.
- Clients point to the META0 table.
- META0 then keeps track of many tables on META1 that contain the locations of the tablets.
- Both META0 and META1 make use of prefetching and caching to minimize system bottlenecks.

Google Bigtable Datastore: Issues

- While Bigtable is a robust tool, developers have been cautious about using it.
- Because it is a proprietary system, they get locked into Google.
- That is also the case with Amazon's Web Services and other cloud providers.
- On the other hand, Google App Engine and Bigtable are affordable, costing about the same as Amazon's S3.
- Costs are as follows:
    - $0.10–$0.12 per CPU core-hour

- $0.15–$0.18 per GB-month of storage
- $0.11–$0.13 per GB of outgoing bandwidth
- $0.09–$0.11 per GB of incoming bandwidth

Mobile Me

Mobile Me

- MobileMe is a set of cloud services and solutions provided by Apple Inc.
- Designed for use with proprietary Apple devices, such as the iPhone.
- MobileMe provides several solutions that are entirely hosted, provisioned and managed via a subscription based billing model from Apple's remote cloud infrastructure.
- Previously known as .Mac and iTools, MobileMe was replaced with iCloud in mid-2011.
- MobileMe is a set of cloud services and solutions Apple's solution that delivers push email, push contacts, and push calendars from the MobileMe service in the cloud to native applications on iPhone, iPod touch, Macs, and PCs.
- MobileMe also provides a suite of ad-free web applications that deliver a desktop like experience through any modern browser.

- MobileMe applications (www.me.com) include Mail, Contacts, and Calendar, as well as Gallery for viewing and sharing photos and iDisk for storing and exchanging documents online.

MobileMe Features

- With a MobileMe email account, all folders, messages, and status indicators look identical whether checking email on iPhone, iPod touch, a Mac, or a PC.
- New email messages are pushed instantly to iPhone Wi-Fi, removing the need to manually check email and wait for downloads.
- Push also keeps contacts and calendars continuously up to date
- Push works with the native applications on iPhone and iPod touch, Microsoft Outlook for the PC, and Mac OS X applications, Mail, Address Book, and iCal, as well as the MobileMe web application suite.
- MobileMe web applications provide a desktop-like experience that allows users to drag and drop, click and drag, and even use keyboard shortcuts.
- MobileMe provides anywhere access to Mail, Contacts, and Calendar, with a unified interface
- Gallery users can upload, rearrange, rotate, and title photos from any browser; post photos directly from an iPhone; allow visitors to download print-quality images; and contribute photos to an album.
- MobileMe iDisk lets users store and manage files online with drag-and-drop filing and makes it easy to share documents too large to email by automatically sending an email with a link for downloading the file.
- MobileMe includes 20GB of online storage that can be used for email, contacts, calendar, photos, movies, and documents

MobileMe

- MobileMe included cloud productivity and synchronization tools, communication services and remote storage.
- MobileMe applications and services included:
    - Find My iPhone: An online tool for iPhone tracking and management
    - Cloud storage: Up to 40 GB
    - Address Book and calendar (iCal): An online contacts and scheduling directory created by synching the iPhone
    - iGallery: Online photo and video storage
    - MobileMe also provided a PC synchronization application, AOL Instant Messenger (AIM) and iWeb for publishing and deploying hosted websites.

Live Mesh

- Live Mesh is Microsoft's "software-plus-services" platform
- Manage, Access, Share files and Applications on the Web and across their world of devices.
- Live Mesh has the following components:
    - A **platform** that defines and models a user's digital relationships among devices, data, applications, and people—made available to developers through an open data model and protocols.
    - A cloud service providing an implementation of the platform hosted in Microsoft datacenters.
    - **Software**, a client implementation of the platform that enables local applications to run offline and interact with the cloud.

- A platform experience that exposes the key benefits of the platform for bringing together a user's devices, files and applications, and social graph, with news feeds across all of these.

- The Live Mesh software, called Mesh Operating Environment (MOE), is available for
  - Windows XP
  - Windows Vista
  - Windows Mobile
  - Mac OS X
- The software is used to create and manage the synchronization relationships between devices and data.
- Live Mesh also incorporates a cloud component, called Live Desktop.
- This is an online storage service that allows synchronized folders to be accessible via a web site.

- It also includes remote desktop software called Live Mesh Remote Desktop, which can be used to remotely connect and manage any of the devices in the synchronization relationship.
- Live Mesh Remote Desktop allows you to control your devices from the Live Mesh application, as well as from any other PC connected to the Internet.

Live Framework

- For developers, there is a development component consisting of a protocol and APIs known as Live Framework.
- Live Framework is a REST-based API for accessing the Live Mesh services over HTTP.
- Live Framework differs from MOE in that MOE simply lets folders be shared.

- The Live Framework APIs can be used to share any data item between devices that recognize the data.
- The API encapsulates the data into a Mesh Object, which is the synchronization unit of Live Mesh.
- It is then tracked for changes and synchronization.
- A Mesh Object consists of data feeds, which can be represented in Atom, RSS, JSON, or XML.
- The MOE software also creates Mesh Objects for each Live Mesh folder so they can be synchronized.