**Facebook, the well-known application**

**PART 1**

Facebook uses Linux. Facebook uses MySQL. The canters are built on the backs of **three tiers of x86 servers** loaded up with open-source software, some that Facebook has created itself. Facebook uses **PHP** for its front-end, and they also spend a lot of time dispatching to back-end services. Again, their framework is home-grown, but they have even gone so far as to build their own PHP engine and development tools. Social networking sites like Facebook function on the basis of **client- server** architecture. Facebook now has **about 30,000 servers** supporting its operations, hosts 80 billion photos, and serves up more than 600,000 photos to its users every second. With over 1B active users, Facebook has one of the largest data warehouses in the world, storing more than 300 petabytes. The data is used for a wide range of applications, from traditional batch processing to graph analytics, machine learning, and real-time interactive analytics.

In order to do interactive querying at scale, Facebook engineering invented[**Presto**](https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920/?s=keen-io), a custom distributed SQL query engine optimized for ad-hoc analysis. It’s used by over a thousand employees, who run more than 30,000 queries daily across a variety of pluggable backend data stores like Hive, HBase, and Scribe.

t's open source, very customizable, and good for security. **Facebook runs the Linux operating system on Apache HTTP Servers**. Apache is also free and is the most popular open source web server in use.

**Flux** is the application architecture that Facebook uses for building client-side web applications. It complements React's composable view components by utilizing a unidirectional data flow. It's more of a pattern rather than a formal framework, and you can start using Flux immediately without a lot of new code. The social site Facebook, for example, is composed of a **combination of coding frameworks and languages including JavaScript, HTML, CSS, PHP, and ReactJS**. This is Facebook's 'tech stack. ' Developers talk about tech stacks because it makes it easy to communicate lots of information about how an application is built.

Facebook stores majority, if not all, of users' data, such as profiles, friends, posts and comments, inside a single giant **social graph**. There are two elements inside a social graph, nodes and edges. A node represents an entity, such a user, a post, a comment and a location.

Facebook stores majority, if not all, of users’ data, such as profiles, friends, posts and comments, inside a single giant social graph. There are two elements inside a social graph, nodes and edges.

* A node represents an entity, such a user, a post, a comment and a location.
* An edge represents the relationships between the nodes. For example, one edge could mean that a particular user created a particular post.
* The database design for the social graph
* Surprisingly, Facebook uses only two database tables to represent the social graph that captures the activities of its one billion users, object table and association table.
* object and node mean the same thing. So are association and edge.
* Object table
* Object table has a very simple schema. It has 3 columns. The id column stores the unique id of the object. otype stores the object type. Additionally, each object/node can have a list of key-value pairs. otype specifies the possible keys and value type. For instance, otype of User means there could be a key name with value type string. The list of key-value pairs are serialized and stored in the data column.
* *id: int*
* *otype: string*
* *data: byte*
* The highly scalable backends that serve the social graph
* There are two major components in Facebook’s backend system, TAO and the database.

APPLICATION SERVER-----------🡪TAO--------🡪DATABASE

Architecture of Facebook

FRONT-END BIG PIPE HIP HOPE

PHP: CUSTOM COMPILER/CACHE IMPLEMENTATION

LINUX: CUSTOM KERNEL EXTENSION

SCRIBE SERVICE AGGREGATES

THRIFT SERVICE1 SERVICE 1, SERVICE 2, SERVICE 3

DATA STORE

MEMCACHE – WRITE THROUGH CACHE IMPLEMENTATION

CASSANDRA MYSQL HBASE HAYSTACK

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PART2**

**PIPE AND FILTER ARCHITECTURE**

It consists of any number of components (filters) that transform or filter data, before passing it on via connectors (pipes) to other components. The filters are all working *at the same time*. The architecture is often used as a simple sequence, but it may also be used for very complex structures.

PUMP—PIPE-🡪FILTER—PIPE🡪FILTER—PIPE🡪SINK

The **filter** transforms or *filters* the data it receives via the pipes with which it is connected. A filter can have any number of input pipes and any number of output pipes.

The **pipe** is the connector that passes data from one filter to the next. It is a directional stream of data, that is usually implemented by a data buffer to store all data, until the next filter has time to process it.

The **pump** or producer is the data source. It can be a static text file, or a keyboard input device, continously creating new data.

The **sink** or consumer is the data target. It can be a another file, a database, or a computer screen

CLIENT-SERVER:--

The client-server model describes **how a server provides resources and services to one or more clients**. Examples of servers include web servers, mail servers, and file servers. Each of these servers provide resources to client devices, such as desktop computers, laptops, tablets, and smartphones.1

**There are four various types of client-server architecture.**

* 1 Tier Architecture.
* 2 Tier Architecture.
* 3 Tier Architecture.
* N Tier Architecture.

The two-tier architecture is like **client server application**. The direct communication takes place between client and server. There is no intermediate between client and server. So, in client application the client writes the program for saving the record in SQL Server and thereby saving the data in the database. The two-tier architecture is like **client server application**. The direct communication takes place between client and server. There is no intermediate between client and server. So, in client application the client writes the program for saving the record in SQL Server and thereby saving the data in the database.

FIRST TIER TASK/SERVICES

CLIENT \*USER INTERFACE

PRESENTATION SERVICES

APPLICATION SERVICES

SECOND TIER TASK/SERVICES

DATASERVER \*APPLICATION SERVER

BUSINESS SERVICES

DATA SERVER

MICROKERNAL ARCHITECTURE:

A microkernel architecture is called a **plugin architecture** because of the way capabilities and functionality are added to a minimal core OS. They're an efficient way to achieve application customization and can offer software lifecycle benefits. Microkernel has features like “**extensibility, flexibility and isolation**”. There are two types of architecture. They are Plug-in modules and core system. This provides extensibility by dividing the application logic into plug-in modules and core system.

PLUG-IN COMPONENT------CORE------PLUG-IN COMP

PLUG-IN COMPONENT------SYSTEM---PLUG-IN COMP