SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Instagram Database : Cassandra and PostgreSQL

Presentation by -

Batch: A2

Tanvi Natu - 16010421065

Bhavya Nanda - 16010421064

Keyur Patel - 16010421073

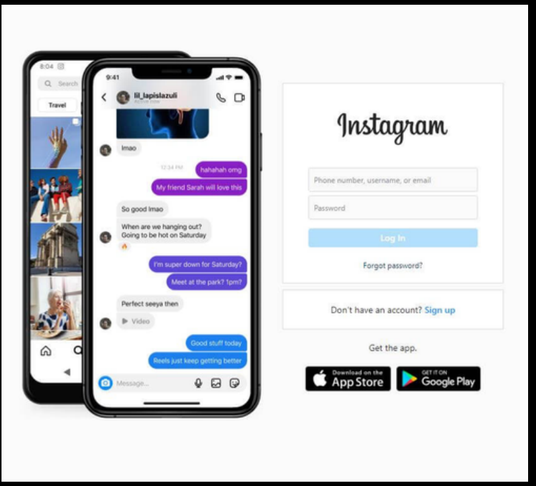
Index

Introduction to Instagram

- Instagram is a popular social media platform that allows users to share photos and videos, apply filters, and engage with others through likes, comments, and direct messaging. It was launched in 2010 and has since grown into one of the largest social media platforms worldwide, with billions of users and millions of posts shared every day.

Huge Data Produced Everyday by Instagram:

Instagram generates a massive amount of data every day due to its immense user base and the constant stream of user-generated content. Here are some examples of the huge data that Instagram produces on a daily basis:

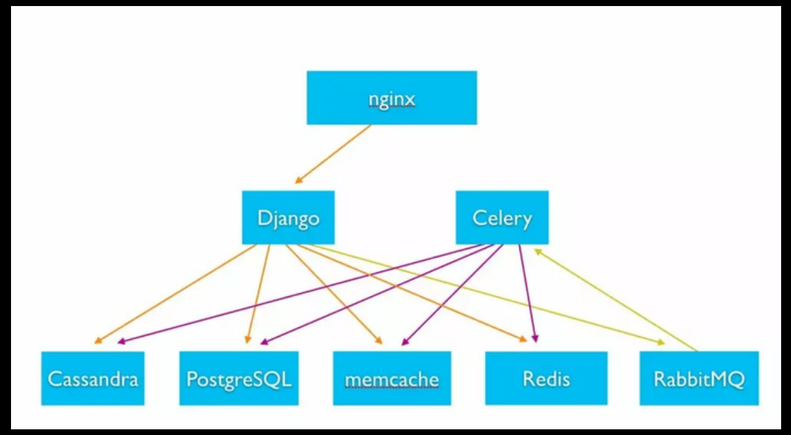


1. Images and Videos
2. Likes, Comments, and Engagement Metrics
3. User Profiles and Metadata
4. System Logs and Operational Data
5. Messaging Data
6. Analytics, Metrics, System Logs and Operational Data

- The sheer volume, velocity, and variety of data generated by Instagram on a daily basis pose significant challenges in terms of data storage, processing, analysis, and management. Instagram's database architecture and data management strategies must be robust, scalable, and efficient to handle this massive data load and provide a seamless user experience

Overview of Instagram's database architecture

- The database architecture of Instagram is a complex system that is designed to handle the massive amount of data generated by millions of users, including images, posts, user profiles, likes, comments, and more. Instagram's database architecture consists of multiple components that work together to store, manage, and retrieve data efficiently. Here's a high-level overview of Instagram's database architecture:



1. **Relational Database Management System (RDBMS):** Instagram uses PostgreSQL, an open-source RDBMS, as one of its primary databases. PostgreSQL is known for its robustness, scalability, and support for complex queries.
2. **NoSQL Database Management System:** Instagram also uses Cassandra, a distributed NoSQL database, as part of its database architecture. Cassandra is designed for high scalability, high availability, and high write performance.

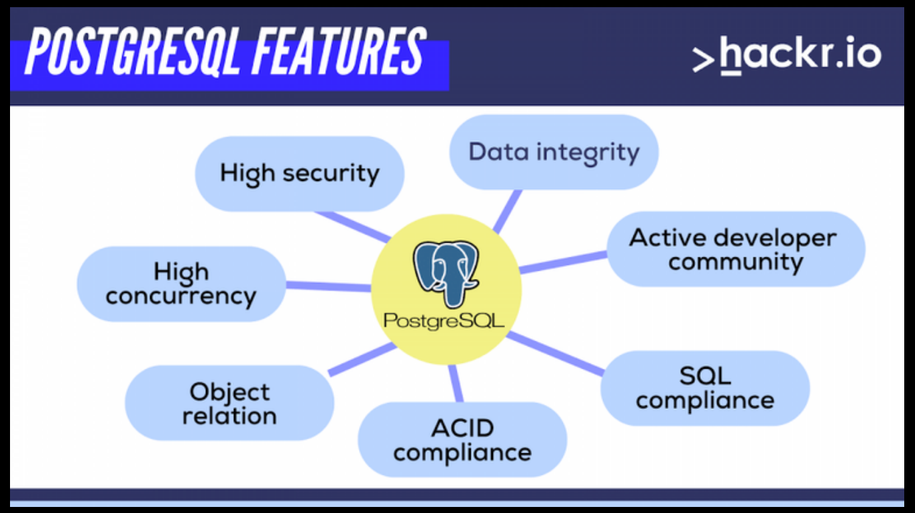
3. Caching: Instagram leverages caching mechanisms to improve the performance of its database operations. Caching involves storing frequently accessed data in memory to reduce the load on the database and decrease response times. Instagram uses caching technologies like Memcached or Redis to cache frequently accessed data, such as user profiles, posts, and metadata.

PostgreSQL for Instagram

- **PostgreSQL** is a powerful open-source relational database management system (RDBMS) that is known for its robustness, data integrity, and support for advanced querying and transactional capabilities. PostgreSQL uses a structured schema with predefined relationships between tables, making it suitable for storing structured data with complex relationships. Instagram uses PostgreSQL to store critical user information such as user profiles, posts, comments, and likes, where data consistency, integrity, and transactional support are important.

Design and Implementation of PostgreSQL in Instagram:

1. **Data Modeling:** Instagram's data model in PostgreSQL involves the design and creation of database tables to represent different types of data such as user profiles, posts, comments, likes, and messaging data. The data model is carefully crafted to ensure efficient storage, retrieval, and management of data.



2. Schema Design: Instagram uses appropriate schema design techniques in PostgreSQL to optimize performance and ensure data integrity. This includes defining primary and foreign keys, indexes, constraints, and data types to ensure data consistency and query efficiency.

3. Backup and Disaster Recovery: Instagram implements robust backup and disaster recovery strategies using PostgreSQL's features such as point-in-time recovery (PITR), logical replication, and backup tools. This ensures data durability and recoverability in case of data loss or system failures.

Here are some examples of queries that Instagram might run in PostgreSQL:

1. **Retrieving user information:**

```
SELECT * FROM users WHERE username = 'example_user';
```

2. **Retrieving posts for a user:**

```
SELECT * FROM posts WHERE user_id = '1234';
```

3. **Retrieving post information with its comments and likes:**

```
SELECT * FROM posts p
JOIN comments c ON p.post_id = c.post_id
JOIN likes l ON p.post_id = l.post_id
WHERE p.post_id = '5678';
```

4. **Retrieving direct messages between two users:**

```
SELECT * FROM messages
WHERE (sender_id = '1111' AND recipient_id = '2222')
OR (sender_id = '2222' AND recipient_id = '1111')
ORDER BY created_at DESC
LIMIT 10;
```

Cassandra for Instagram

- Instagram uses Cassandra as a part of their database architecture to store user-related data, such as user profiles, followers, and likes. Cassandra's distributed and sharded nature allows it to handle the massive amounts of data generated by Instagram users, while its tunable consistency levels provide the necessary flexibility to balance performance and data consistency. Additionally, Cassandra's support for the column-family data model allows for flexible schema design, enabling Instagram to store data with different structures and attributes.

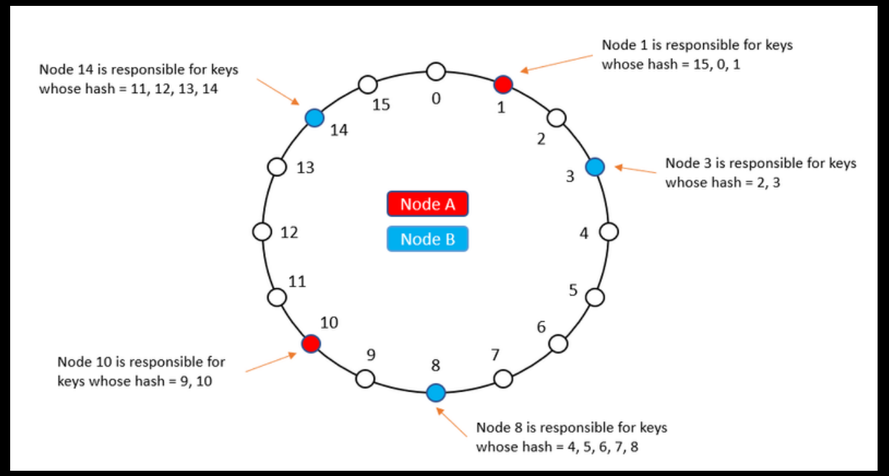
Design

- Instagram's Cassandra deployment consists of multiple clusters across different regions, each containing multiple nodes.
- Each node contains a set of physical disks used for storing data, with the data being distributed across the cluster using a sharding mechanism.
- The sharding mechanism is based on the partition key, which determines the node that a piece of data will be stored on.

Implementation

- Instagram uses Cassandra to store user-related data, such as user profiles, followers, and likes.
- Cassandra is also used for storing metadata related to photos and videos, including their location and tags.
- Instagram uses Cassandra's built-in support for Time-To-Live (TTL) to automatically expire data that is no longer needed, reducing storage requirements and improving performance.

Here are some examples of queries that Instagram might run in Cassandra:



1. **Retrieve a user's profile information by their user ID:**

```
SELECT * FROM user_profiles WHERE user_id = 123456;
```

2. **Retrieve the list of followers for a user**

```
SELECT * FROM user_followers WHERE user_id = 123456;
```

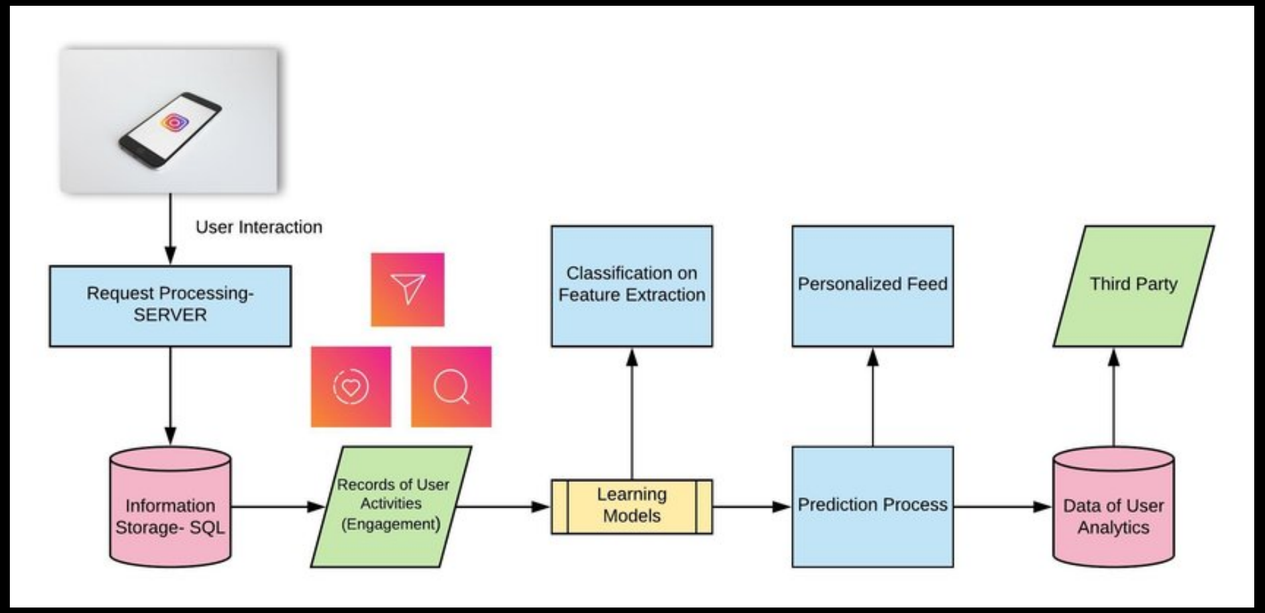
3. **Retrieve the number of likes for a specific photo:**

```
SELECT COUNT(*) FROM photo_likes WHERE photo_id = 'abc123';
```

4. **Retrieve all photos posted by a user in the last 24 hours**

```
SELECT * FROM user_photos WHERE user_id = 123456 AND post_time >=
'2022-04-01 00:00:00' AND post_time <= '2022-05-01 00:00:00';
```

Conclusion



- PostgreSQL, Cassandra, and other databases are crucial to Instagram's operations, and it would be challenging to operate without them. These databases are optimized for handling different types of data and workloads, and their use allows Instagram to efficiently store, retrieve, and process large amounts of user data.
- Other databases such as Redis and Elasticsearch are also important for caching frequently accessed data and providing search functionality, respectively.
- In conclusion, the use of PostgreSQL, Cassandra, Redis, Elasticsearch, and other databases is critical to Instagram's ability to provide a seamless and reliable user experience to its millions of daily users. Without these databases, Instagram would struggle to handle the high volume of data generated by its users, leading to slower processing times, decreased performance, and an increased risk of system failures.

References

- Instagram Engineering Blog. (2014). Scaling Instagram Infrastructure. **https://instagram-engineering.com/scaling-instagram-infrastructure-part-1-3f390c4fba95**
- Instagram Engineering Blog. (2015). Cassandra at Instagram. **https://instagram-engineering.com/cassandra-at-instagram-1b5d5e8cdccc**
- DataStax. (n.d.). Data Modeling in Apache Cassandra: Best Practices. **https://www.datastax.com/resources/data-modeling-in-apache-cassandra-best-practices**
- Apache Cassandra Documentation. (n.d.). CQL for Cassandra 3.11. **https://cassandra.apache.org/doc/latest/cql/**
- PostgreSQL Documentation. (n.d.). SQL Commands. **https://www.postgresql.org/docs/current/sql-commands.html**

Thank You