

Part I: Describe Possibilities and Your Solution

Problem Specification:

For the given problem statement, there are a few key specifications first. They are,

- The game should support millions of users,
- The system should have scaling support as the database grows with millions of customers.
- The most critical specification is the database should be extremely fast, and the changes should be applied in real-time.
- The database also should show the top 10 players in a region in real-time.

Probable Database System:

Considering the following key specifications, scalability and low latency are the most crucial key requirement. To solve the problem, I would discuss three databases, Azure Cosmos DB, Google BigTable, and Amazon Redshift.

Amazon Redshift is a data warehouse from these three cloud database systems. Because of using sql, it can be considered a relational database. And the rest of the two is distributed NoSQL database. These three databases closely fulfill the given specifications. But I will discuss these three in detail in the next section and describe the one that works the best for the given problem.

Databases:

Azure Cosmos DB: Azure Cosmos DB is a globally distributed, multi-model database service provided by Microsoft Azure. It is designed to support a wide range of workloads, including transactional, analytical, and real-time processing, and can handle millions of requests per second with low latency. Cosmos DB supports multiple data models, including document, key-value, graph, column-family, and multiple APIs. Cosmos DB also has built-in global distribution, multi-region writes, and active-active replication, which allows the user to scale the database across multiple regions and ensure high availability. It also has various consistency levels allowing the user to trade consistency for performance.

Google BigTable: Google BigTable is a distributed, scalable NoSQL database provided by Google Cloud Platform (GCP). It is designed to handle high levels of scale and performance and can handle millions of queries per second and terabytes to petabytes of data. BigTable is a column-family database that stores data in tables with rows and columns. It is designed for large-scale, real-time analytics and supports various workloads, including online and offline batch processing, stream processing, and interactive querying. BigTable has a low-latency, highly-available access pattern that allows users to read and write data with low latencies. It also integrates with other

GCP services such as Cloud Data Fusion, Cloud Data Loss Prevention, and Cloud Dataproc.

Amazon Redshift: Amazon Redshift is a fully managed data warehouse service provided by Amazon Web Services (AWS). It is based on columnar database architecture and uses SQL, making it a relational database. Redshift is designed to handle high levels of scale and performance and can support petabyte-scale data warehouses. It is optimized for fast querying and analysis of data using SQL and can support a wide range of business intelligence and analytics applications. To update data in a Redshift cluster, the user can use the INSERT, UPDATE, or DELETE SQL commands to modify existing data. However, these updates may not be real-time, and the cluster's performance may be affected if users load large amounts of data or make frequent updates.

Difference in Each Database:

All these three databases have their own perks. Let's discuss the differences between these three databases.

Scalability:

- **Cosmos DB:** It is a globally distributed database service that supports automatic scaling. It can handle millions of requests per second with low latency and is designed to support a wide range of workloads, including transactional, analytical, and real-time processing.
- **BigTable:** It is also a distributed, scalable NoSQL database that is designed to handle high levels of scale and performance. It can handle millions of queries per second and terabytes to petabytes of data.
- **Redshift:** It is a fully managed data warehouse service that is designed to handle high levels of scale and performance. It can support petabyte-scale data warehouses and can handle queries from thousands of concurrent users.

In terms of scalability, all three systems are designed to support the storage and management of large amounts of data.

Latency:

- **Cosmos DB:** For the latency, it has multiple consistency levels that allow users to trade off consistency for performance. It also has a low-latency, highly-available access pattern that will enable users to read and write data with low latencies of under 10ms at the 99th percentile.
- **BigTable:** It is designed to support low latency and high throughput. It is optimized for storing large amounts of structured data and is commonly used in applications that require real-time access to large datasets. BigTable uses a distributed, column-oriented storage model and advanced query optimization to enable fast access to data.

- Redshift: It uses a columnar data storage model and is optimized for fast querying and analyzing large datasets using SQL. Redshift uses a distributed columnar storage model and advanced query optimization to enable fast querying and analysis of large datasets. It also offers a range of node types with different performance levels to allow developers to choose the level of performance that best meets their needs.

In terms of latency, Cosmos DB and Bigtable are designed to support low latency and high availability, while Redshift is designed for fast querying and analysis of large datasets.

Query Updates:

- Cosmos DB: Cosmos DB is designed to handle query updates with low latencies and a range of consistency levels to meet the specific needs and requirements of the application or workload.
- BigTable: BigTable is also designed to support low latencies and high throughput. It can also handle query updates with low latencies and a range of consistency options.
- Redshift: Redshift is designed to handle query updates with fast querying and analysis of large datasets, but it is not designed for real-time updates or queries. It is typically used for batch processing and analysis of large datasets rather than real-time processing.

In terms of query updates, Cosmos DB and Bigtable both support real-time updates and a variety of update methods. At the same time, Redshift is not designed for real-time processing and supports batch updates using SQL commands.

Regional Support:

- Cosmos DB: Cosmos DB allows developers to create database accounts in multiple regions worldwide and uses a distributed data model to replicate data across regions. This database enables developers to choose the regions where they want to store their data and provides them with low-latency access to their data from anywhere in the world. In addition to global distribution, Cosmos DB also supports multi-region writes, allowing developers to write data to multiple regions simultaneously. This feature can be helpful for applications that need to write data to various regions for disaster recovery or other reasons.
- BigTable: Bigtable is available in multiple regions worldwide, and developers can choose the regions where they want to store their data. This allows developers to store their data closer to their users, which helps reduce latencies and improve performance.
- Redshift: Redshift supports multiple regions around the world to allow developers to store their data closer to their users and build globally distributed applications that can scale globally and handle high levels of throughput. The specific number

of regions that Redshift supports may vary over time as Amazon adds new regions and updates its service offerings.

In terms of regional support, Cosmos DB and Redshift both offer multiple regional offerings and allow developers to choose the region that best meets their needs. Bigtable also has multiple regional offerings, but it is not available in all regions worldwide.

Most Suitable Database:

After discussing the differences between these three databases, I want to select the Azure Cosmos DB as the most suitable database for the given problem based on its specifications.

If we look closely to the key specification of the problem, we need to have a database that can,

- Handle a vast number of data.
- Auto-Scale itself based on the number of users.
- Reflects the changes in real-time.
- Show the top 10 players of a given region in near real-time. (As there can be players from different regions of the world, I also discussed the regional support of the databases.)

The following discussion above shows that Google Bigtable is not available in all regions, unlike the other two. Similarly, Amazon Redshift does not support real-time processing. And it is also evident that only Azure Cosmos DB only meets all the criteria from the problem specification. Thus Azure Cosmos DB is the best choice for implementing the database of the top mobile game.