Rashaad Washington

CSCI 4800

September 18, 2022

Research Paper

A Look into MMORPGs

Video games are on the rise and have been ever since being available to the public. Being able to play alongside a friend seems to be what most video game companies aim for. This abstraction has also proved to increase the number of users playing the game. Massively multiplayer online role-playing games (MMORPGs) have always been notable for their mass number of users. However, a common issue with mass amounts of users, is data management. How do MMORPGs manage and store vast amounts of data from their users?

While a video game may start small in the size of its users, it can change drastically in matters of days, hours, or even seconds. Users also need to be able to process inbound and outbound data while maintaining a reliable connection to the server. To achieve such expectations requires a lot of thought and decision-making about the database architecture. The architecture of the database is crucial to any online video game. Seemingly, MMORPGs spend much time planning the design and architecture of their database in the early stages of development. A preferred and ideal database achieves high scalability, performance, and reliability.

Scalability is needed to adapt and grow with the number of users playing. Second, performance is pivotal in communicating back and forth with the server while being able to update local files, code, and graphics. Lastly, reliability is to ensure that the end user will

experience less interrupts and disconnections from the server. This abstraction layer provides a base for the development of MMORPGs.

Software architecture is the evidently the most important component of all software. This includes video games, web software, desktop applications and more. The architecture as a whole determines every functionality and potential of a MMORPG. Such architectures are categorized by their characteristics. Common architectures of MMORPG's include but are not limited to availability, continuity, performance, recoverability, reliability, robustness, and scalability. While some MMORPG's value some more than another, it is ultimately up to the stakeholders to make those decisions.

Scalability, the ability to adapt and grow in proportion to the number of users, is the most critical characteristic in MMORPGs. It is what defines 'massive' and 'multiplayer' in MMORPGs. When it comes to the architecture structure, a database of some sort is needed. There are two types of database types: ACID and BASE. An ACID (atomicity, consistency, isolation, durability) database attempts to focus on commits and rollbacks to ensure database integrity. In other words, at any given moment, the database is perfect. On the other hand, the database could reflect a BASE type. A BASE (basic availability, soft state, eventual consistency) attempts to prioritize being available over being always perfect. To resolve the scalability constraint, the database needs to reflect a BASE type database.

The difference between ACID and BASE databases has been studied for many years. In short, ACID attempts to keep data safe and consistent whereas BASE aims more for high availability. These two approaches are arguably better for certain databases. For example, ACID is assumed to be an excellent fit for RDBMS while on the other hand, BASE is assumed to be fit for NoSQL databases. A RDBMS provides greater stability, performance, and consistency

whereas BASE provides high availability, scalability, and reliability. This is the ideal approach when choosing what kind of database but one must keep in mind that every architectural decision made about the database is often times a tradeoff. This means that every decision has an advantage and disadvantage.

Database and storage are essential for any online game. It allows users to save their data before exiting in hopes that they will be able to access the same data from when they exited at an earlier time. Although for a mass amounts of data going inbound and outbound, big data managements is needed. This is not a new concept. In fact, there are plenty of microservices today that aims to manage big data. These microservices are NoSQL, Amazon S3, MapReduce, Hadoop Distributed File System, and much more. Most MMORPGs use NoSQL, a SQL native language that are designed to "capture all data without categorizing and parsing it upon entry into the database". Therefore, the data is varied making it a BASE type database.

In fact, scalability is one of the various advantages of NoSQL over RDBMS, otherwise known as relational database management systems. RDBMS use a horizontal scaling that produces overhead. This will likely decrease performance and is not ideal. Additionally, the schema model is not cohesive but decoupled. A decoupled model allows for structures to be independent and less reliable from one another. In an ideal client-server architecture, there are 3 basic servers: the login server, the gateway server, chat server, and the zone server. At the bottom of that architecture is the central database. This database stores almost everything necessary to the game and even meta data about users. It could store game data, world data, state data, log data, and more.

Reliability is the rate at which a user stays connected to the service. In terms of MMORPG's, it translates to the rate at which a user stays connected to the game before being

disconnected or having game crashes. Reliability is implemented through many aspects. These could be networking, data management, and rendering.

At the minimum, networking is involved in all aspects of MMORPG's. A user must always be connected to the server. Also important, the server should remain online so as long as any users are active. This starts with the game company having a network provider to provide low latency internet connection. Not only should this internet connection be low latency, but it should far exceed the internet speeds of household internet, and equipment should be stress tested with realistic incoming traffic. This is to ensure that network issues do not arise when traffic increases. To assist in network traffic, oftentimes a substantial amount of game data is stored locally on the client side. This is typically unchangeable data. Additionally, networking can be improved by networking framework.

Today, more and more MMORPG's use networking frameworks to manage networking. This is largely due to games not being able to support several hundred simultaneous players. An example of a widely-used networking framework is Project Darkstar. Developed by Sun Microsystems, Darkstar is an open-source Java solution to networking for massive multiplayer online games. Project Darkstar is essentially a server that hosts clients. It is used to send and receive necessary network packets.

In most MMORPG's, to minimize the number of concurrent users playing on one server, players are distributed into different shards. Players are usually connected to shards that are geographically remote to their playing location. Oftentimes, this is to help with reliability issues. For example, a game company which hosts servers in California will likely provide better reliability to users who are geographically remotely located near California. In opposition, users who are geographically located further away from California will experience a lower quality of

service. Therefore, game companies will aim to have servers, or shards, located at various geographical regions to balance the load, thus providing reliability. This is commonly referred to as sharding. Sharding can also prevent the entire video game from being unusable. For example, in the event that an unsharded database fails, the entire video game is unusable. On the other hand, with a sharded database, only parts of the video game is unusable. Sharded databases serve as a backup.

The third step in establishing reliability is by game rendering. Probably one of the most intrinsic characteristics of a video game is the graphics. Many years prior, graphic cards were only a mere abstraction and was not quite an implementation yet. Most video games in the past were played with none to extremely low graphic cards. These graphic cards allowed for video games to extensively draw and animate the screen faster than a CPU might.

Nowadays, modern video games are more essential than ever and require a decent graphics card to render and display the video game. More importantly, establishing reliability is more concerned with notion that a user's personal computer or console should be able to play a video game using the dedicated graphics without crashing. That is not always the case. In most cases, the MMORPG's requires and extreme amount of graphical processing that most personal computers and consoles are not able to render. When that happens, the video game will typically perform much slower than that of the recommended computer or even worse, the video game will crash.

To solve this issue, most MMORPG's have dedicated settings to meet the performance of most graphic cards. For example, World of Warcraft, quite arguably the largest MMORPG by population, has numerous video settings to allow for users to change depending on their individual computer's performance. From shaders to anti-aliasing, these settings will help users

game not crash when playing at certain times during the game. A MMORPG that is widely known for it's heavy graphics is The Elder Scrolls. This game achieves low reliability because it is primarily targeted for systems with an extreme amount of graphical processing power and not low-end graphics.

Performance is another extremely important characteristic when playing MMORPG's. Evidently, a database configured to meet high performance is yet another way of how video game companies manage data in MMORPG's. Factors that play a role high performance are query searching, NoSQL, and Cassandra.

One of the key words that describe the performance of a database is its speed. In databases the speed of queries has a direct relation to performance of the database, and ultimately, the performance of the game. This is more or less known as querying. Querying involves the client requesting a transaction from the server. The time that it takes for the server to return the requested transaction can be evaluated as the speed. Ultimately, slower speeds, output less performance.

Well, how does a database achieve faster query speeds? There are numerous methods to do so but one in common is data partitioning. Data partitioning distributes the processing over a set of data partitions. Data partitioning is different than sharding because data managed in single row is partitioned based on row key horizontally in order to achieve high performance and speeds. The two are also different due to the fact that data partitioning groups sets of data within a single centralized database whereas sharding spreads data across multiple computers. An example of data partitioning is social networking apps like MySpace, Facebook, and even Instragram. In these apps, data is partitioned according to geography.

Depending on the server, speed is measured by calculations. There is a continuous amount of data processing, query optimization, and logic being done. At the very core, the CPU and RAM performs the former. A database with an extremely fast CPU and RAM will have a relatively faster query search speed.

NoSQL, which stands for 'not only SQL', is designed to grasp data without categorizing and parsing. NoSQL databases typically have less schema which is one of the primary reasons that makes it so efficient. In fact, NoSQL came to existence because of the fact that SQL systems had limited performance when scaling. MMORPG's are known to scale unbelievably fast. To counter that issue, NoSQL systems were used.

Some common NoSQL databases are MongoDB, DynamoDB, Neo4j, and Cassandra. These databases are more commonly used because of their BASE properties. In order to compare the differences and similarities of these four, it is worth analyzing their BASE properties.

MongoDB relies on consistency over availability. It's strong consistency is because consistent data is stored on the primary MongoDB server rather than the replica sets spread across multiple data centers. Reads are only allowed from the primary datacenter. DynamoDB is highly available and has eventual consistency. Almost similar to MongoDB, DynamoDB data is replicated across three datacenters. Neo4j is consistent and can be configured to be highly available. This is done by slave databases given the choice to be replicated into the master database. Lastly, Cassandra is highly available and achieves eventual consistency. Cassandra achieves high availability by a distributed storage, multi-data center, and linear scalability. Of the four databases, research proves that due to its Cassandra's scalability, it is the most likely to resemble a BASE database type.

Based on the NoSQL system and written in Java, Apache's Cassandra database is widely used by many video game companies to manage the data of many users in MMORPG's. Cassandra is an open source, nonrelational, NoSQL database that is able to store large amounts of data. Cassandra is know for its built-in caching. Some popular companies that use Cassandra as its database is Facebook, Instagram, and Netflix.

Cassandra is mostly known for two features its hybrid structure. That is, Cassandra uses a column-orientated and key-value database that features an adjustable consistency. This alone constitutes and abides to the properties of a BASE database.

Cassandra should be used for MMORPG's for multiple reasons. First off, Cassandra does not have any network bottleneck. This is significant because it can ensure that state data can be and will be read or written to the database. Without this feature, some users may experience missing data because their internet was not able read and write data.

Cassandra's column structure is s efficient than a simple key value store. Each row in a Cassandra's column consists of a row key and a dynamic set of super columns. Each super column stores another set number of columns. This allows for databases to manage a single player's data in one row, and partitioned data based on row key across numerous nodes in a cluster. Therefore, there are no more join operation during reading data, and overall, performance will be increased.

A case study was published by Central South University to determine read and write performance of operations in a Cassandra database. With 5 nodes, the average read and write response time was around 15ms. Even further, when stress testing the database with approximately 4500 connections, data implied that there was not much variation in performance.

This study proves that a Cassandra database can support MMORPG's with an increasing number of players.

Another framework that will assist in performance is Hadoop. Hadoop, which stands for highly archived distributed object-oriented programming, is an open source Java framework mainly developed by Goug Cutting and Mike Cafarella for a search engine project. Hadoop aims to store and access chunks of information from data.

By integrating Cassandra and a Hadoop file system, applications can see an increase and improvement in performance. Another case study was published by Central South University that compares the results of the previous case study with the results of a Cassandra-Hadoop integration. The results of this study implied proved that using a Cassandra-Hadoop integration performs much faster than just a Cassandra database alone.

In modern society, MMORPG's are infamous due their addictive nature. As a result, many MMORPG's have an increasing user-base. However, the caveat of having more users is that it usually results in an increased effort to manage those users. More importantly, managing the data.

In conclusion, videogame companies and developers should consider the important characteristics of the MMORPG's database. Most will aim for scalability, performance, and reliability. For determining the database to manage the massive amount of data, MMORPG's should use a BASE approach. In rare cases should one consider using an ACID approach. By using a BASE approach, it provides eventual consistency and prioritizes being available over being always perfect. This is ideal because ACID databases will fail if at any point a state is not complete. An uncomplete state may occur if the user experiences network interruptions or slower internet. NoSQL databases focus on being BASE. A popular and widely-used NoSQL database

is Cassandra. Cassandra is used heavily by most MMORPG's today. Known for its, fault intolerance and having no network bottleneck, Cassandra performs the same even when scaling by increasing users.

Works Cited

Alqwbani, Zhang Zuping, Fares Aqlan, Abdullah. " Big Data Management for MMO Games and Integrated Website Implementation." *Global Journal of Computer Science and Technology* [Online], (0): n. pag. Web. 21 Sep. 2022

Drkušić, Emil. "MMO Games and Database Design." *Vertabelo Data Modeler*, Vertabelo, 11 June 2016, https://vertabelo.com/blog/mmo-games-and-database-design/.

Wadley, Greg, and Jason Sobell. "Using a Simple MMORPG to Teach Multi-User, Client-Server Database Development." *Microsoft Academic Days on Game Development in Computer Science Education,* 22 Feb. 2007, pp. 75–79.

Santos, Alexandre M. M. et al. "A Methodology Proposal for MMORPG Content Expansion Analysis." (2018).

B. Saraladevi, N. Pazhaniraja, P. Victer Paul, M.S. Saleem Basha, P. Dhavachelvan, Big Data and Hadoop-a Study in Security Perspective, Procedia Computer Science, Volume 50, 2015, Pages 596-601, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.091.