Cloud Computing to Support the Evolution of Massive Multiplayer Online Games. Architecture and Challenges

Eric Mesa Aleix Drudis

October 30, 2022

Contents

1 Introduction

As the computer game market has become have gradually become more intensive, growing numbers of games have been released. MMOs is one of the most popular sub-categories. As a result designers and developers are facing the new challenge of creating a mobile background architecture that provides MMO users the same smooth experience enjoyed by traditional PC users.

One advantage of using cloud services for games is that you do not need to invest in physical hardware servers upfront, but only need to pay according to usage or service plans at a later stage. It is one way to help manage the risks involved in developing a new game title.
Another advantage is that your game can tap into vast cloud resources to achieve scalability (effectively manage any sudden spikes in the number of concurrent players, intense real-time game calculations or data requirements). This keeps the performance of your game stable around the clock.

2 Challenges in Cloud Computing on MMOs

2.1 Quality of Service (QoS)

When we speak of QoS in MMOs the technology that is used are the integrated services (IS). We use the term IS for an Internet service model that includes best-effort service, real-time service, and controlled link sharing.The requirements and mechanisms for integrated services have been the subjects of much discussion and research over the past years.

MMO games generally require mutual real-time visibility between users on the same screen. A large volume of movement and combat packets must be broadcast within a certain field of vision. In this case, MMO game servers produce a massive amount of communication packets when many users are playing simultaneously. Therefore, the access layers of MMO game servers require ample network bandwidth and high network packet throughput.

The better the QoS, the easier it becomes to maintain and deploy updates. There can be multiple dimensions to QoS:

- Energy Consumption: Simulation of gaming environments is a daunting task for physical devices, which is why, the server, needs to be smart enough for the game to run smoothly on a device. The rendering time of surroundings, player interaction with it, and the other players need to be done is a very short period. This requires an adequate amount of energy as well as good network strength which is discussed as the second dimension. It has been observed that physical networks consume 12-38% more power as compared to games running on the cloud.
- Network Speed: Latency is the term commonly used to identify the strength of the network, lower the latency, faster the render and response time for the player. What is often referred to as texture glitch is a result of render failure due to service unavailability or network's poor strength? Higher latency can also lead to a frame drop. Anti Aliasing in combination with poor latency can add up to irritable gaming experience. Data consumption also depends upon the type of game and its graphics quality the player selects.

2.2 Elastic Resource Scalability

MMORPG games also constantly launch new servers to attract new players.After a time, players become less active and gradually stop playing, also they have high and low peak periods, so they require server resources that support quick and reliable elastic scaling.

While high-bandwidth, low-latency internet is now becoming ubiquitous, this is not enough to solve the scalability issues that net-VEs are beginning to encounter. These scalability problems arise in part because of the need to maintain consistency between all the players. In the best case, inconsistency may just lead to transient visible artifacts with no long-term consequences. However, in practice, it can easily cause much more serious problems, like objects being lost or duplicated during a financial transaction. In addition to degrading the realism of the virtual world, consistency violations are a major source of security problems in net-VEs . To maintain consistency, all net-VEs have a transaction management layer that employs a commercial database.

3 Architecture

MMO games adopt a typical three-layer architecture (access layer, logic layer, and data layer). The access layer is primarily responsible for game account login authentication, communication packet decryption and client connection management. The logic layer implements the game's primary service logic, such as regional game services, chatting, and rankings. The data layer provides data persistence storage and shared cache. RDS is used to ensure disaster recovery for primary and secondary databases, read/write splitting, distributed databases, etc. At the logic layer, the core scenario servers can flexible extend the service based on the scenario, helping increase the maximum number of simultaneous players for a single zone.

Online play comes in several forms, such as session-based multiplayer matches, massively multiplayer virtual worlds, and intertwined single-player experiences.

In the past, games using a client-server model required the purchase and maintenance of dedicated on-premises or co-located servers to run the online infrastructure, something only large studios and publishers could afford. In addition, extensive projections and capacity planning were required to meet customer demand without overspending on fixed hardware. With today's cloud-based compute resources, game developers and publishers of any size can request and receive any resources on demand, avoiding costly up-front monetary outlays and the dangers of over or under provisioning hardware.

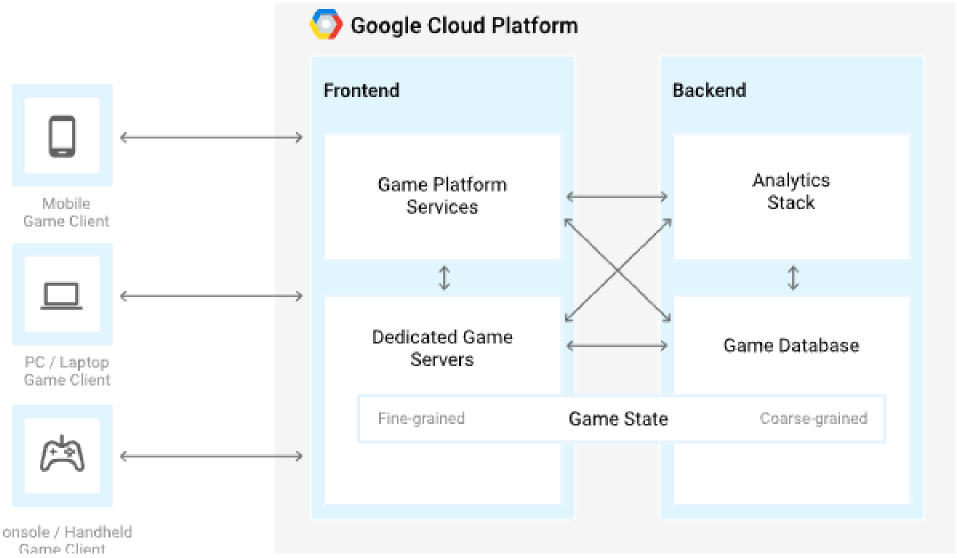

Figure 1: Structure of Google Cloud Platform

3.1 Service-oriented architecture

It's common for the platform your game is running on to provide some or most of these services:

- Leaderboard and match history
- Matchmaking
- Online lobby
- Chat
- Inventory management
- Authorization
- Party/group
- Profile
- Cross-platform unlock
- Feeds
- Analytics

In software engineering, service-oriented architecture (SOA) is an architectural style that focuses on discrete services instead of a monolithic design. A service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently. SOA is also intended to be independent of vendors, products and technologies.

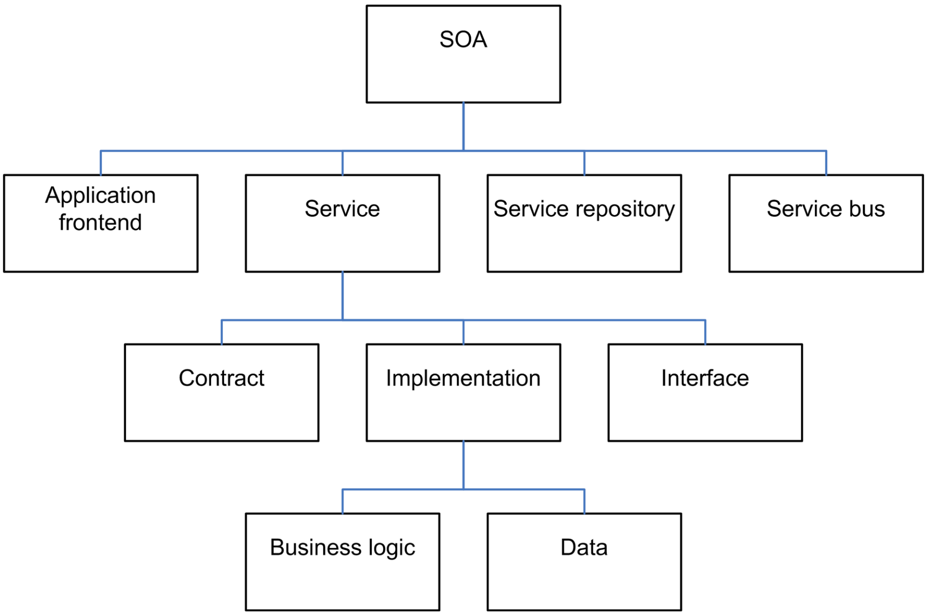

Figure 2: Elements of SOA

4 Bibliography

https://en.wikipedia.org/wiki/Service-oriented_architecture

https://www.alibabacloud.com/es/solutions/gaming/web/MMORPG

https://www.alibabacloud.com/solutions/gaming/computer/MMO

https://www.mdpi.com/2674-113X/1/1/6/htm

https://cloud.google.com/architecture/cloud-game-infrastructure

https://d-nb.info/1247631885/34

https://learn.microsoft.com/en-us/windows/uwp/gaming/cloud-for-games

https://azure.microsoft.com/es-es/solutions/gaming/#solution-architectures