ORACLE
NOSQL DATABASE

NoSQL Database for Mobile Social Gaming

Introduction

The explosive growth in smart devices has enabled a new class of collaborative online gaming applications, also known as "Social Gaming." This class of gaming applications allows users to quickly launch a game, connect to a server, and collaborate with other players that are online. For example, consider a game that presents online users a virtual world for exploration where they may encounter other like-minded users along their journey and interact with them. Given the potential scale of such a system (i.e. millions of users with billions of interactions) and the latency requirements of game play, a distributed low latency and linear scale solution is needed.

How it works

In a typical online game, players will acquire "items" as they achieve certain milestones during game play. In addition to item acquisition, social gaming adds another dimension to the game play experience. That is, online players may interact with each other within the virtual world. For example, when a player navigates to a location in the virtual world, the value of the X/Y coordinates on the player's canvas is transmitted to the server. Correspondingly, the client device will request information about other players' avatars that are within a specified pixel range of the player. A list of avatar images and player names that are within pixel range of the user will then be returned to the client application. The player can then choose to interact with the player by either sending the player a chat request or challenging the player to an interactive mini-game.

Hence, developers of such social gaming applications are concerned with the following issues:

- Mass and Scale – Due to the ubiquitous nature of smart devices, the number of online players may be extremely large. Furthermore, game developers require a store that can easily scale out horizontally without a large development investment in database sharding (typically a complex design and implementation effort to partition the RDBMS workload for horizontal scaling).
- Iterative development and "schema-less" store – Delivering new game features at a high rate of velocity requires a flexible store that can easily accommodate changes without requiring a large development effort, especially when these changes must scale out horizontally.
- Write-intensive workload with low latency – Interaction with an online game must not be perceived by the player as sluggish; hence the latency requirements for servicing client events are extremely aggressive (< 30ms). Furthermore, the online gaming workload can be characterized by a very high degree of writes (item acquisition, avatar movement, and player stat updates) with a large majority of reads being cached at game instantiation time.

Figure 1 depicts a simplified architecture for a mobile game solution. Mobile clients will connect to HTTP servers to write X/Y coordinate moves of avatars, update player statistics (coins earned), and retrieve the

Published April 2012

X/Y coordinates of other players' avatars that are currently in game. The Oracle NoSQL database provides an extremely compelling solution in this space due to its following features;

- Schema-less store – A key-value store enables the game developer to move quickly with new functionality without having to expend effort on RDBMS mapping, serialization, and functional sharding.
- Low latency operations – The Oracle NoSQL database delivers extremely low latency write operations while maintaining the ability to horizontally scale to meet throughput requirements.
- Scale without coding – The Oracle NoSQL database enables the game developer to quickly scale out their application without the design and implementation effort to distribute the workload across multiple DBMS instances.

Other data management and processing

Complementing the Oracle NoSQL architecture is the Oracle RDBMS that is critical to the overall solution. Game development organizations must have ad hoc reporting visibility into the health of the business. As depicted in Figure 1, we see the relational database management system serving up real-time, ad hoc reporting. This critical solution component enables the game distributor to ask the following business questions:

- How many unique users per month are interacting with the game?
- How is the game performing from a revenue perspective by geography, quarter over quarter, year to date?
- What is the rate of growth for user acquisition?

In summary, the distributed Key-Value store and the relational database management system work together in order to provide important business value for massive-scale, low-latency, and high-velocity development in the mobile social gaming space.

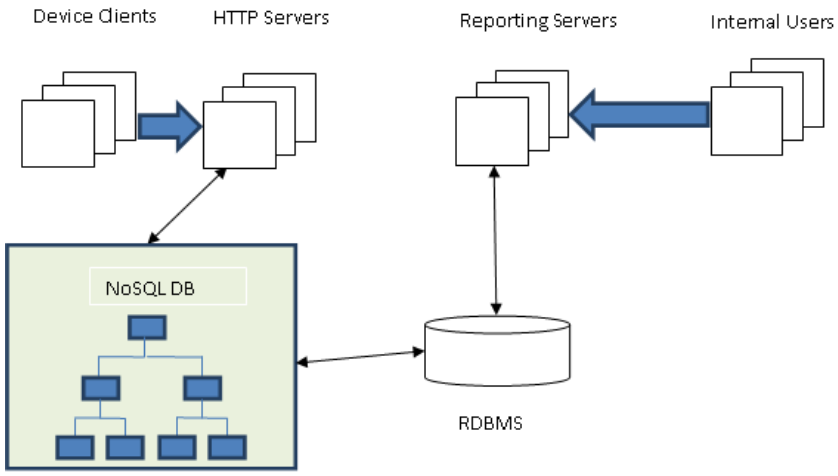

Figure 1: Social Gaming Architecture

Published April 2012