Bases de datos no relacionales

Práctica Cassandra

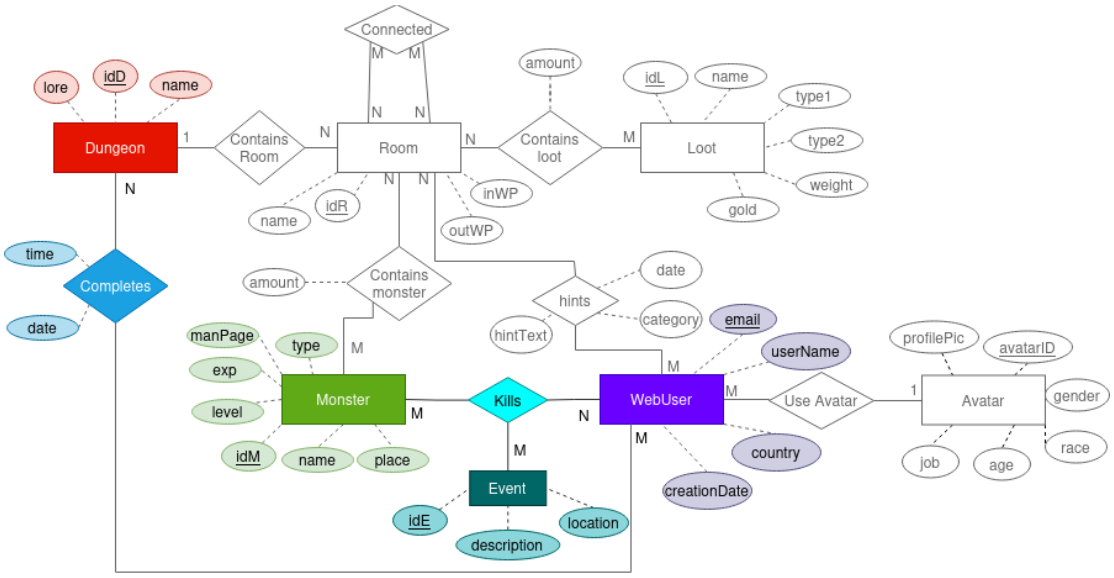
Introducción

En el videojuego “Jotun’s Lair” de la empresa “Norsewind studios”, existen unos leaderboards que muestran los tiempos que han tardado los usuarios en completar las mazmorras. Estos leaderboards solo se pueden consultar desde el campamento de los jugadores. El campamento, es una zona especial del juego dónde los jugadores están a salvo y pueden descansar. Se pensó como una zona de refugio donde los jugadores puedan examinar su equipo, ajustar las habilidades de sus personajes, buscar otros jugadores y mirar las leaderboards. En concreto, los jugadores pueden examinar dos leaderboards: el “hall of fame” de cada país, es decir, para un país concreto se muestran para cada mazmorra del juego el TOP 5 de jugadores más rápidos de ese país, incluyendo sus tiempos; y las estadísticas de un jugador, que muestra los tiempos que ha tardado en completar una mazmorra en particular ordenados de menor a mayor. Los leaderboards se deben ir actualizando en tiempo real, pero hay que tener en cuenta que la consistencia de estos leaderboards es importante y que un pequeño retardo a la hora de actualizar o mostrar estos leaderboards no tiene impacto en el juego ya que en esencia los usuarios están navegando por un menú y no en pleno gameplay.

Por otro lado, se está preparando una nueva feature para el juego, las *“Hordas”*. Una Horda es un evento especial del juego con una duración de unos 30-40 minutos en la que

los jugadores deben resistir en una fortaleza del mapa oleadas de monstruos que tratan de conquistar la fortaleza. Las Hordas aparecerán distribuidas por el mapa según el estado del juego y pueden ocurrir varias al mismo tiempo. Por ahora, las Hordas no están disponibles para todos los jugadores, aunque se está realizando una beta cerrada con testers profesionales para afinar esta nueva feature del juego. El equipo de “game design” quiere introducir un leaderboard para las Hordas que muestre los N jugadores (aún no se tiene claro cuantos) que más monstruos han matado hasta el momento durante una Horda en concreto. El leaderboard se debe ir actualizando en “tiempo real”, hay que tener en cuenta que cualquier retraso a la hora de enviar/recibir la información en este punto es crítico ya que se ejecuta mientras los usuarios están en pleno gameplay. La consistencia en este leaderboard no es tan importante ya que el equipo de “game design” ha comprobado que, si los rankings bailan un poco, los jugadores se motivan más porque da la sensación de que la competición está más tensa. Sin embargo, los rankings deben de reflejar la realidad, si bailan demasiado los usuarios lo notarán y se disgustarán. Por temas de latencia, los jugadores solo participan en una Horda con otros jugadores del mismo país, por lo que el leaderboard de Horda será local por país.

Hasta el momento, la empresa tiene una base de datos relacional para dar soporte al videojuego. Sin embargo, en los últimos años, el juego ha cogido mucha popularidad y el número de jugadores y países donde se vende el juego creció mucho. El equipo de operaciones ha notado que la base de datos relacional que usan no cumple con los objetivos de rendimiento ni de escalabilidad necesarios para dar soporte a los leaderboards del juego, en especial el de los nuevos eventos de Horda. Por esa razón, se están planteando migrar la base de datos relacional a Cassandra. A continuación, se muestra la base de datos relacional que se usa actualmente para el juego, incluidas las tablas creadas para las pruebas sobre las Hordas.



Esquema E/R de la base de datos relacional que da servicio al videojuego. En color las entidades y relaciones relacionadas con los leaderboards, en blanco el resto de las entidades.

Objetivo

Se desea realizar un prototipo de una base de datos de Cassandra para dar servicio a los tres leaderboards anteriores. El juego hace tres peticiones de lectura al servidor que se resumen en la siguiente tabla. La fila “rank” indica el leaderbord que se puebla con la petición de lectura, la fila "in” indica los parámetros de entrada que recibe la llamada y “out” indica el formato de los datos que devuelve el servidor. Nótese que los leaderboards son locales a cada país:

LECTURAS			
Rank	Hall of fame	User statistics	Top Horde
in:	country: str	user_id: int dungeon_id: int	country: str, event_id: int, K: int,

Out:	data : [{
	dungeon_id: int,
	dungeon_name: str,
	Top_5: [{
	email: str,
	user_name: str,
	time_minutes: float,
	date: str
	},...]
	},..]
		data : [{
		time_minute: float,
		date: str
		}...]
			data : [{
			user_id: int,
			user_name: str,
			email: str,
			n_killed: int
			}...]

* Las fechas deben seguir el formato descrito en el iso 8601			

Además, el juego hace dos peticiones de escritura al servidor que se ejecutan cuando ocurre un evento concreto durante el juego. La siguiente tabla muestra la información de las peticiones de escritura, la fila “When” indica el evento en el juego que dispara la petición de escritura y la fila “data” indica los datos que se envían al servidor:

Escrituras		
when:	User finish dungeon	User kills monster during Horde event
data:	dungeon_id: int, email: str, time_minutes: float, date: str	event_id: int, email: str, monster_id: int
* Las fechas deben seguir el formato descrito en el iso 8601		

Teniendo esto en cuenta resuelve las siguientes tareas:

Tareas

1. Diseña una base de datos Cassandra para dar servicio a las lecturas y escrituras anteriores. Argumenta tus decisiones de diseño.
2. Crea las consultas .sql necesarias para exportar los datos de la base de datos relacional a ficheros .csv. Los ficheros deberán tener un formato acorde al diseño del punto 1.
3. Prepara un cluster local de 3 nodos todos en el mismo rack y datacenter.
4. Haz un fichero .cql que creen tu diseño en Cassandra y cargue los ficheros .csv creados en el paso 2. Se debe utilizar un factor de replicación de 2 y tener en cuenta que se las pruebas se ejecutaran en un cluster local.
5. [OPCIONAL] Si el diseño lo necesita, actualiza la tabla de escrituras para incluir cualquier modificación que sea necesaria en la información que se le debe proporcionar al servidor.
6. Haz un fichero .cql que realice las consultas de escritura y lectura necesarias. Incluye el nivel de consistencia de cada consulta, teniendo en cuenta las características de los diferentes rankings.