

Una mirada más distribuida a los videojuegos

Miguel Muñoz Molina¹

¹ Universidad de Granada, España

Abstract. En este trabajo vamos a hablar sobre los MMO o Massive Multiplayer Games (juegos de multijugador masivo). Es importante destacar que estos sistemas pueden llegar a ser muy complejos debido a la gran cantidad de jugadores que pueden llegar a albergar.

Vamos a hablar desde juegos online muy básicos hasta algunos juegos que han logrado una gran cantidad de jugadores simultáneos. Tras esa toma de contacto, tocaremos algunas de sus principales características y daremos unas conclusiones acerca de cómo están progresando estos sistemas a través de los años en el sector del videojuego.

Keywords: Sistemas distribuidos, Videojuegos, Cliente-Servidor

1 Introducción

Para poder hablar mejor sobre los videojuegos y los sistemas distribuidos debemos ir un poco atrás en la historia. Nos remontamos al punto en el que empezaron a aparecer los primeros videojuegos para los ordenadores. Es cierto, eran muy divertidos, pero faltaba algo.

Es por eso, que en torno a 1979 un grupo de estudiantes de la Universidad de Essex crean el que sería el primer juego de rol online en un servidor, que sería la base de los actuales MMORPG. Su nombre es MUD, Multi User Dungeon [1].

Este primer juego se basaba en una interfaz sencilla de texto, en el que podríamos crear nuestro personaje y hacerlo volverse más fuerte.

Tras este juego, surgieron muchas variantes de éste, pero siguiendo las mismas premisas. En este punto ya se añadieron interfaces gráficas y comenzaron a surgir los primeros MMORPG como los conocemos hoy en día [2].

Quien haya jugado alguna vez a estos juegos sabe a la perfección lo importante que es la conexión en este tipo de juegos. Una latencia alta puede arruinar el juego completo, y es por eso por lo que los servidores van a ser un pilar fundamental que debemos tener en cuenta en todo momento.

2 Descripción

Lo primero que debemos diferenciar es dos conceptos distintos de hacer estos juegos. Por una parte tenemos los sistemas centralizados y por otra parte los sistemas distribuidos.

Sistemas centralizados:

En estos sistemas encontramos un servidor único al que se conectan todos los clientes. Estos sistemas tienen un gran problema, que se trata de la escalabilidad.

Un ejemplo de este tipo de juegos puede ser EVE Online [3], que contaba con una única copia del juego cargada en el servidor centralizado

Imaginemos un escenario en el que nuestro MMORPG consigue un aumento drástico sobre su número de jugadores, colapsando todo el sistema y generando un gran cuello de botella [4]. Para intentar solventar esto tenemos que dirigirnos a un entorno híbrido, creando una combinación entre servidor centralizado con toques P2P.

Sistemas híbridos:

Tenemos que pensar en estos juegos como una combinación de dos partes distintas, el juego principal y los juegos auxiliares.

El juego principal carga el mundo principal, sus texturas, objetos, NPC's, físicas, etc. Este juego principal se suele cargar en el servidor centralizado, ya que la mayoría de acciones sobre éste son cosas predecibles.

Por ejemplo, si hablas con un personaje controlado por la máquina con un diálogo establecido, ya se sabe qué va a ocurrir y, por tanto, la carga que se genera en este proceso es mucho menor que el resto.

Los juegos auxiliares cargan partes del mapa, que se distribuyen dependiendo de la cantidad de jugadores que cargan una zona. Estos juntan a muchos jugadores con una latencia aceptable para poder conectarse entre ellos. Además, se elige a los jugadores con mejor latencia para hacer de servidores en estas partes.

No obstante, estas arquitecturas tienen un gran problema. Los jugadores humanos son impredecibles y hacen que los servidores se colapsen, ya que no están listos para tanta carga. Para intentar conseguir una escalabilidad infinita se busca cambiar a una arquitectura totalmente distribuida.

3 Características Principales

En este apartado vamos a tocar las características fundamentales de todo sistema distribuido, pero centrándonos en el ámbito de los videojuegos.

3.1 Compartición de recursos

Para la compartición de recursos debemos indicar que estos juegos cuentan con un programa principal, que consiste en un bucle infinito muy simple con unas pocas órdenes que van actualizando el juego. Sobre estas órdenes se monta el juego entero.

Como hemos visto, el tener un sistema centralizado hace que todo se ejecute en el mismo equipo y genera grandes problemas de escalabilidad, por lo que poco a poco y por la necesidad basada en la demanda de los usuarios se está pasando a sistemas más distribuidos, desde servidores en distintas regiones del mundo hasta servidores creados por parte de los jugadores con mejor conexión.

Estamos hablando de que las personas cada vez demandan servidores más grandes y con mayor carga computacional, por lo que estos sistemas son totalmente necesarios para poder cumplir con estos objetivos.

3.2 Sistema abierto

Este es un punto complicado debido a que la mayoría de los videojuegos en línea no permiten el añadir más programas y funcionalidades por parte de desarrolladores. De todas formas, podemos encontrar juegos que permiten ampliar la parte del software.

Una forma pueden ser los mods, que añaden funcionalidades a los juegos base, y están hechos por la comunidad.

3.3 Concurrencia

La concurrencia es algo indispensable si estamos hablando de videojuegos online (quitando algunos casos sobre juegos por turnos), ya que permiten a cada jugador ir de forma independiente. En el caso de, por ejemplo, los MMORPG, estamos hablando de concurrencia de miles de jugadores de forma simultánea para poder ofrecer un servicio satisfactorio.

Es por ello que cada cliente de nuestro servidor se trata de un proceso independiente del resto. Esto hace que, por ejemplo, un jugador con una baja latencia no se vea afectado por otros jugadores con una latencia más alta.

3.4 Escalabilidad

La escalabilidad va a depender del tipo de juego que queramos hacer y de la cantidad de personas que vayan a jugarlo.

Si hablamos de los MMORPG más conocidos actualmente, como World of Warcraft [5], Guild Wars 2 [6], Black Desert Online [7], etc. Vemos que estos juegos manejan un público cada vez más creciente, por lo que tienen que ser capaces de aguantar el aumento de demanda, con un código perfectamente organizado y separado para simplificar esta tarea.

3.5 Tolerancia a fallos

Nunca podemos asegurar que un sistema nunca vaya a fallar, y en los videojuegos online no iba a ser menos. Hay cientos de posibles fallos que se pueden dar en un videojuego y que, a veces, pueden afectar considerablemente a la disponibilidad.

Los juegos más afectados por esto son juegos como los MMORPG, que, al tener gran cantidad de jugadores, hacen más probable el que se encuentre algún tipo de bug o error en los servidores.

Cuando esto sucede hay varias formas de tratarlo, pero la más común son los mantenimientos. Esto no es más que franjas de tiempo donde no es posible conectarse al servidor y jugar, y durante este tiempo se procede a arreglar los fallos producidos.

Antiguamente, un mantenimiento podía durar hasta días, pero actualmente el público se ha vuelto más exigente y, ahora mismo, sería impensable tener un juego en mantenimiento por varios días y que nadie pusiera el grito en el cielo.

Es por ello que hoy en día el hacer nuestro software lo más tolerante a fallos se ha vuelto importantísimo. Para conseguirlo se hacen desde mantenimientos cortos a las horas de menos afluencia de jugadores hasta cambios en caliente sin necesidad de hacer ningún tipo de mantenimiento. Es por eso que ahora entendemos mejor el paso a un sistema más distribuido. Sólo hay que imaginar lo que pasaría si en nuestro videojuego con un sistema centralizado se cayese el servidor. Absolutamente todo el sistema dejaría de ir.

En cambio, con un sistema distribuido, por ejemplo, en regiones del mundo, puede ser que se caiga el servidor de Europa, pero siga funcionando el de Norteamérica. Incluso en algunos sistemas como el que comentábamos que son los usuarios con menos latencia los que hacen de servidores, es fácil de solventar que uno de estos servidores dejara de funcionar, siendo tan fácil como poner a otro usuario como nuevo servidor.

3.6 Transparencia

La premisa principal es ver el sistema distribuido como un todo, como una única entidad formada por todos. Es por ello que, esto se sigue en los videojuegos para hacer mucho más sencilla su programación. Siendo distribuido podemos encontrar este código para el programa principal de nuestro juego:

“First, you have the “game loop”. This is really just a nearly infinite loop where inside the loop you are checking user input, updating the game state, and rendering the graphics and sound to represent the game state. It looks something like this:

```
while(true) {  
  check_input()  
  update_game_state()  
  render_screen()  
}
```

And that is at a high level what a game is doing. Now, for each of those three things, you’ll get hideous levels of complexity.” [8]

Como vemos aquí, el programa se basa en, mirar qué inputs han sucedido, actualizar el estado de juego y renderizar la pantalla. Esto nunca cambiará aunque tengamos varios servidores, o sólo uno.

También es importante hablar sobre la migración y cómo ésta ayuda a quitar sobrecarga a algunos nodos. Esto se llama balanceo de carga, y la idea es la siguiente: vemos un nodo con mucha carga, por lo que migramos parte de su carga a otro menos cargado. Esto no siempre puede salir rentable, porque puede que el hacer el balanceo de carga requiera muchas migraciones.

4 Conclusiones

Hoy en día, los videojuegos se han convertido en uno de los métodos más importantes de entretenimiento, y eso ha hecho que más y más gente se interese por ellos. Es por eso que debemos responder a esta cantidad en constante aumento de jugadores con un servicio de calidad.

Actualmente los servidores centralizados no nos permiten cumplir estos objetivos. Para conseguirlo, como indica el título de este trabajo, debemos tener una mirada más distribuida a los videojuegos. Esto nos permite poder tener un sistema más escalable y fiable. [9]

La pregunta ahora es, ¿qué me hace pensar eso? Son varios los factores, pero podemos destacar varios.

El más fundamental es la buena maduración del género MMORPG, que ha perdurado en el tiempo y actualmente está más fuerte que nunca. También debemos hacer una mención especial a la moda del género Battle Royale, una modalidad que viene de la antigua película “Battle Royale”, haciendo juegos para más y más jugadores simultáneos, generando una concurrencia mucho más exigente.

Es por todo esto que los sistemas distribuidos actualmente han ganado una gran importancia en el campo de los videojuegos y serán los responsables de que un día, podamos mirar atrás con orgullo, recordando cuando a alguien se le ocurrió utilizar un ordenador para jugar en red.

References

1. MUD Videojuego, [https://es.wikipedia.org/wiki/MUD_\(videojuegos\)](https://es.wikipedia.org/wiki/MUD_(videojuegos))
2. Historia de los videojuegos distribuidos, <https://sistemasdistribuidos.foroactivo.com/t133-videojuegos-como-sistema-distribuido>
3. EVE Online, <https://www.eveonline.com/>
4. Sistemas distribuidos en MMORPG, <https://link.springer.com/article/10.1007/s11042-014-2389-0>
5. World of Warcraft, <https://worldofwarcraft.com/es-es/>
6. Guild Wars 2, <https://www.guildwars2.com/es/>
7. Black Desert Online, <https://www.blackdesertonline.com/>
8. How does any MMO games backedn work?, <https://medium.com/@narengowda/how-does-any-mmo-games-backend-work-df19b44f73a7>
9. Distributed Systems. Concepts and Design, <https://ce.guilan.ac.ir/images/other/soft/distributedsystems.pdf>