Rendimiento en Videojuegos en Línea.

Los videojuegos en línea, al requerir una gran cantidad de reflejos y un rendimiento muy estable para no causar frustración a los jugadores, tienen una serie de medidas específicas, siendo la más característica de estas el tickrate.

Tickrate

El tickrate es la frecuencia con la que un servidor ejecuta pasos de su simulación, siendo un “tick” un número asociado con un paso de simulación que se retransmite a los clientes para ayudarles a sincronizarse con el servidor.

Esta frecuencia se limita a un cierto máximo por tres razones:

Para conservar el ancho de banda del servidor y del cliente.

Para conservar tiempo de CPU.

Para permitir a los clientes saber cuánto tiempo pasa entre cada tick, esto es importante para los juegos en línea, ya que las actualizaciones desde el servidor pueden llegar en intervalos distintos o desordenados.

Cuando el tickrate del cliente y el tickrate del servidor no están sincronizados, por ejemplo, porque la máquina del cliente no es capaz de soportar los requerimientos de red o del juego para conectarse correctamente, se produce el fenómeno conocido como lag, del que hablaremos a continuación.

Lag

El lag en videojuegos online es cuando se produce un retraso visible entre la acción del jugador y la respuesta del servidor.

Este fenómeno suele ser frustrante especialmente en juegos de acción, donde la velocidad y los reflejos son clave.

Se causa ya que mientras un videojuego reproduce una copia local del juego, un videojuego en línea requiere ser replicado a partir de un servidor central para evitar inconsistencias entre clientes, por lo que el cliente no tiene control directo sobre el estado del juego y sólo puede enviar peticiones de cambio al servidor, y así actualizar su copia local recibiendo actualizaciones del servidor, y esta necesidad de comunicación causa un retraso entre los clientes y el servidor.

Hay tres posibles causas de lag:

Rendimiento pobre de la red.

Insuficiente capacidad de procesamiento del cliente.

Insuficiente capacidad de procesamiento del servidor, que es la que nos interesa, ya que estamos estudiandolo desde el punto de vista del servidor y ocurre cuando el usuario realiza un pedido que no se encuentra en la [caché](https://es.wikipedia.org/wiki/Cach%C3%A9_web) y la respuesta requiere una gran cantidad de procesamiento antes de poder ser enviada. Esto afecta generalmente en las horas de mayor actividad donde el servidor se ve exigido por tener que procesar la información de una enorme cantidad de usuarios conectados.

Compensación de lag

Existen diversos métodos para reducir o camuflar el Lag, algunos de estos aplicables desde el punto de vista del cliente y otros desde el lado del servidor.

Desde el punto de vista del cliente:

A los clientes no se les permite cambiar el estado del juego, sino que deben recibirlo del servidor, pero si que puede tratar de mostrar el juego de la forma más precisa posible, y para ello es necesario realizar predicciones, para ello existen dos métodos:

Interpolación, que funciona guardando en buffer el estado del juego, y renderizandolo con un ligero y constante retraso, así cuando un paquete llega del servidor, en lugar de actualizar la posición de un objeto inmediatamente, el cliente interpolará su posición, comenzando por su última posición conocida. Durante ese periodo de interpolación, eI objeto se moverá de forma suave entre las dos posiciones, idealmente este intervalo será el mismo que el retardo entre paquetes, pero este es raramente el caso.

Extrapolación, es un intento de estimar el estado futuro del juego. Tan rápido como llega un paquete nuevo, la posición del objeto se actualiza a la nueva posición, y mientras espera al siguiente, la siguiente posición se extrapola basándose en la posición actual y el movimiento en el momento de la actualización. Esencialmente el cliente asume que un objeto se continuará moviendo en la misma dirección, y cuando un nuevo paquete llega la posición se tendrá que corregir ligeramente, ya que no se puede predecir el movimiento exacto que se dará. Un ejemplo de esto es la opción de predicción de movimiento del videojuego League of Legends.

Desde el punto de vista del servidor:

A diferencia de los clientes, el servidor si conoce el estado exacto del juego, y dicha predicción no es necesaria. El propósito principal de la compensación desde el servidor es proporcionar efectos precisos de las acciones de los clientes. Esto es importante porque entre que un cliente realiza una acción y esta llega pasa un tiempo, y por tanto el juego ya no estará en el estado en el que el jugador efectuó la acción. Un ejemplo de esto es la detección de impacto en los FPS (First Person Shooters) en los que los márgenes son pequeños y causarían errores si no se maneja de forma correcta. Posibles soluciones son:

No hacer nada:

Una posible solución es ignorar el problema, en videojuegos como RTSs, esta solución es posible ya que es posible un retardo entre que el jugador comanda una acción y esta se produce sin resultar frustrante, en FPSs hasta el más mínimo retardo causaría frustración.

Rebobinar el tiempo:

Otra solución sería guardar el estado de la partida cuando el jugador efectuó la acción, y rebobinar al estado en el que se encontraba cuando se procesa esta. Para rebobinar un tiempo apropiado para por ejemplo determinar adónde apuntaba el jugador cuando disparó se usa la latencia del jugador. Aunque esto resultará en que el juego verá al jugador disparando a una posición donde ya no está el objetivo y dando, lo cual puede resultar confuso para el desarrollo de la partida.

Confiar en los clientes:

Es posible permitir a los clientes decir lo que están haciendo y que el servidor confié en esa información que recibe. Aunque este método es susceptible a trampas, ya que los jugadores podría enviar información falsa al servidor, mediante mensajes externos hacia él, con software de trampas.

Aunque esta técnica se utiliza en juegos cuya escala hace técnicas como rebobinado imposibles, como por ejemplo Battlefield 3, donde el sistema sólo hace un test en el que ve si es posible que el disparo fuera acertado según la posición nueva.

Obligar a los clientes a extrapolar:

Una solución menos común sería hacer a los clientes extrapolar para cubrir su latencia, aunque esto produce resultados incorrectos a no ser que los clientes se muevan de forma constante, dando ventaja a aquellos que se muevan de forma inconsistente, ya que el resto de jugadores los verían en posiciones distintas a donde realmente están.