

Reto 1: Fundamentos de Sistemas Distribuidos

Juan José Sánchez Cortés
Ingeniería de Sistemas
Universidad EAFIT
Medellín, Colombia
jjsanchezc@eafit.edu.co

Abstract—En este artículo se aborda la aplicación de sistemas distribuidos en el mundo actual como lo son 'Google Stadia' con su arquitectura cliente-servidor y basada en la nube; y 'GTA V Online' con su arquitectura peer-to-peer (P2P). Cada uno se analizará por medio de ciertos puntos los cuales ayudarán a entender por qué estas empresas decidieron la implementación de estas arquitecturas

PalabrasClave—Arquitectura orientada al servicio (SOA), Estilo de arquitectura de microservicios (Microservice architectural style), Arquitectura basada en la nube (Cloud-based architecture), Cloud Computing, Peer-to-peer (P2P), Google Stadia, GTA V

I. INTRODUCTION

A lo largo del tiempo, muchas empresas han tenido que adaptarse a los cambios en las tecnologías, otras, se han creado para intentar ser revolucionarias en sus respectivos campos. En este artículo se busca desarrollar habilidades en el diseño de los sistemas distribuidos a través del estudio de una empresa en la actualidad y los servidores de un videojuego en línea, en este caso hablaremos de Google Stadia y el videojuego GTA V Online (Grand Theft Auto). Se mostrará de forma detallada algunos aspectos tales como:

- Funcionamiento general
- Arquitectura y sus elementos
- Ventajas y desventajas de la solución
- Retos y Oportunidades de mejora

Al final de este artículo se espera que quede claro las diferencias entre los sistemas distribuidos de los que se habló y entender las razones por las cuales cada una de estas empresas escogieron dicha arquitectura.

II. DESARROLLO

A. Funcionamiento Global

Google Stadia:

Fue un servicio de videojuego en la nube, el cual usaba los centros de datos de Google para poder retransmitir videojuegos en diferentes resoluciones y FPS -cada uno con su plan-. Este servicio empezó el 19 de noviembre de 2019 y cerró el 18 de enero del 2023. El sistema funcionaba de tal forma que el usuario simplemente conectara el control e iniciara desde el dispositivo electrónico que tuviese a la aplicación de Google Stadia, sea celular, tv o pc. Una forma muy sencilla de mostrar cómo funcionaba este sistema es que con una buena conexión el usuario se conectaba con el dispositivo y lo que hacía este

era conectarlo otro pc ubicado en el centro de datos de Google, por lo tanto, tu no necesitarías de un hardware muy costoso para correr los juegos, ya que tu estabas conectado a él viendo, por así decirlo, una transmisión y en caso tal que estuvieses en un juego multijugador no tendrías que preocuparte de la latencia entre tu juego y el servidor, ya que estarían en el mismo lugar

GTA Online:

GTA es un videojuego creado por ROCKSTAR GAMES en el año 2013, hasta el momento, este juego ha sido relanzado para 7 plataformas a lo largo de los años. Este cuenta con 2 modos de juego, un jugador y multijugador (en este caso solo hablaremos del multijugador), cuando salió el multijugador, las personas estaban muy sorprendidas debido a que para este modo de juego ROCKSTAR GAMES había usado una arquitectura muy poco usada por las grandes compañías la cual era el P2P (por lo general se usaba la arquitectura cliente-servidor) para el multijugador.

B. Presentación de las aplicaciones e introducción a la problemática

Google Stadia:

Google Stadia se creó con el fin de poder darle -con un bajo precio- una excelente experiencia de juego a las personas, especialmente a las que no se podían permitir un hardware de última generación, este quería mostrarse como una alternativa más económica a las consolas o pc. Stadia tenía como diferenciador que todo el modelo se basaba en la nube. Otra razón por la cual decidieron hacer un sistema distribuido era el tema de la escalabilidad ya que independiente de la concurrencia que hubiese, los datacenters que tenían estaban lo suficientemente optimizados y en caso tal que se vieran en problemas, ya tenían un plan en caso tal que tuviesen que hacer más datacenters

GTA Online:

El multijugador de este juego es un mundo abierto por la ciudad en la que se desarrolla la historia principal (modo un jugador), en esta se pueden hacer diferentes actividades, como tener negocios propios, ir al casino, hacer carreras, entre muchas cosas más. Todo este conjunto de actividades para hacer y la constante actualización del juego ha llevado a las personas a jugarlo desde el día de su salida, y a comprarlo más de 2 veces (por el cambio de generación de las consolas). Por sesión online, puede haber mínimo 1 persona y un máximo de 32 a la vez. En el año 2020 se registró el pico más alto de personas conectadas a la vez (550.000 personas solamente en

pc, no se tiene registro del resto de consolas). Cualquier otro multijugador con arquitectura cliente-servidor hubiese caído ya que por lo general la media de jugadores es mucho más baja y tener servidores con capacidad de ese pico de jugadores sería un desperdicio la mayoría del tiempo.

C. Arquitectura

Google Stadia:

En temas de arquitectura Google Stadia usa dos tipos de arquitectura, las cuales serían: cliente-servidor y arquitectura basada en la nube, cada una cumple un rol específico, por ejemplo:

- **Arquitectura Cliente-Servidor:** Esta arquitectura es usada para dividir el programa en dos principales componentes, los cuales serían el cliente y el servidor. El cliente corre en el dispositivo del usuario, este es responsable de renderizar los gráficos del juego, se encarga de revisar y enviar los movimientos que haga el usuario con el control y tiene la tarea de comunicarse con los servidores de Stadia (en realidad son los de Google). Por otro lado, los servidores se encargan de procesar los movimientos del control, renderizar el juego y mostrar vía stream el video y audio del juego.
- **Arquitectura basada en la nube:** Stadia utiliza los servidores de Google para correr los juegos que pide cada usuario, esto permite que el usuario no tenga la obligación de comprar hardware aparte del control. Gracias a su flexibilidad de la nube en temas que ya el usuario no tiene que preocuparse por nada-solo del internet-, los desarrolladores pueden aprovechar esta arquitectura a su favor ya que no tendrían porqué contenerse en temas gráficos ni en la utilización de recursos del hardware.

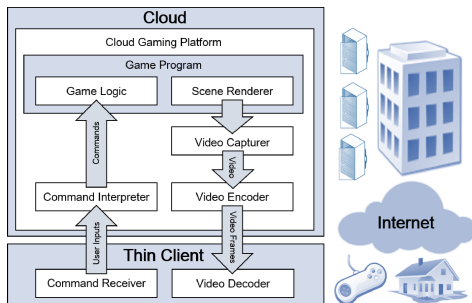


Fig. 1. Modelo de la arquitectura de 'Google Stadia'. Imagen sacada de [13]

GTA Online:

Como se comentó anteriormente en este documento, GTA Online usa una arquitectura Peer-to-Peer (P2P) desestructurado para su multijugador, esta arquitectura solo se compone de nodos, cada nodo tiene la misma responsabilidad de todos, la cual sería compartir recursos, datos y servicios los unos con los otros todo esto sin necesidad de un servidor ya que cada uno de los nodos funciona como cliente y como servidor lo que lleva a que las cargas de trabajo sean repartidas entre todos los nodos. Esta arquitectura tiene diferentes tipos, como lo podrían ser: Estructurado, Híbrido, Desestructurado, entre otros. En este

caso solo se especificará acerca de del desestructurado ya que esta es la que se utiliza en el videojuego ya comentado; A diferencia de los otros tipos de P2P la conexión de cada uno de los nodos no tiene una estructura predefinida -a diferencia del tabla hash distribuida que utiliza el P2P estructurado-, esto genera que a medida que cada nodo llega tiene buscar para encontrar los datos que necesita.

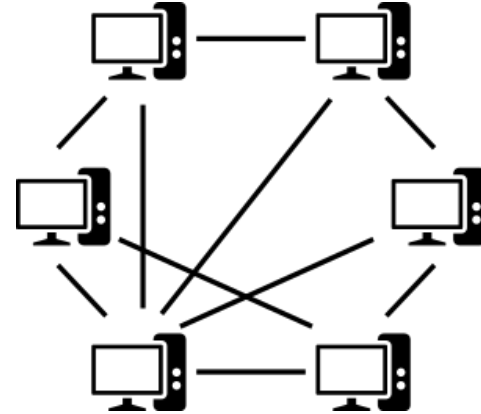


Fig. 2. Ejemplo de arquitectura P2P Desestructurado. Imagen sacada de [14]

D. Ventajas y desventajas de la solución y arquitectura

Google Stadia:

Como se ha comentado, Stadia tiene muchos puntos positivos a su favor, aparte de la buena planeación y gestión del proyecto que este conlleva, ya que supieron sacarle provecho a la cantidad de servidores que Google posee (2.4 millones) ya que llevar a cabo la arquitectura basada en la nube no es para nada barato, entrando en temas más técnicos, usar la arquitectura basada en la nube es una muy buena idea ya que le permite al usuario no preocuparse del hardware y el servicio como tal solo usa la cantidad necesaria para entregar una buena experiencia sin necesidad de ocupar espacio innecesario que podría estar usando otro usuario por lo tanto, se puede decir que es muy accesible y escalable. Gracias a la cantidad de servidores que tienen a su disposición, le permite ser resiliente y flexible a cambios en temas de: demanda, condiciones de internet y preferencias de usuario. Si seguimos con las ventajas de la arquitectura Cliente-Servidor podremos ver que el sistema es muy seguro, ya que cuenta con características tales como autenticación, previene las trampas y protege los datos del usuario; También reduce la latencia entre el cliente servidor ya que el cliente (en este caso el 'pc') y el servidor están en el mismo lugar. Ahora hablando de que salió mal en Stadia (ya que cerraron, se sabe que diferentes cosas no funcionaron), primero que todo voy a hablar acerca que el modelo de negocio en temas de cómo se conseguían los juegos, en pocas palabras, aparte de pagar la suscripción para poder jugar (en resolución alta o baja), tenías que pagar el juego completo y estar conectado obligatoriamente a internet para poder jugar, a diferencia de otros modelos en donde tú lo comprabas y lo descargabas, independiente si estabas conectado o no a internet. Ahora a lo que nos interesa a nosotros, stadia tenía

un problema, puede que tenga muy poca latencia, ya que la 'consola' en la que se juega y el servidor del juego es la misma, algo que no tuvieron en consideración era el tiempo que la información tardaba en enviar desde el control hasta llegar a la vista del usuario (input lag), donde 0.5 segundos se siente. Por lo tanto, puedo tener muy buena latencia pero estar muy alejado del servidor más cercano, lo que lleva a una mala experiencia ya que independiente si el juego se ve en muy buena definición, si no tengo buena retroalimentación la experiencia se va a ver muy afectada; Otro problema con Stadia era que a veces se borraba el progreso (obviando los errores provocados por usuarios y problemas con la cuenta de estilo suspensión) causado por los errores del servidor como lo sería corrupción de datos o crasheo del servidor; Acabamos de hablar de algunos problemas con la arquitectura basado en la nube ahora seguimos con los problemas de la arquitectura cliente-servidor, algunos de los problemas de esta arquitectura podría ser -como lo comentamos antes en este párrafo- la dependencia a internet y la dependencia del servidor, otros aspectos a tener en cuenta sería el ancho de banda de cada persona y la estabilidad de este, ya que si llegan a haber fluctuaciones en la red, la calidad tiende a intentar buscar la resolución más optima (a veces baja mucho la calidad) pero está a veces tiende a cerrar directamente el juego.

GTA Online:

En este espacio se comentará acerca de las ventajas y desventajas que conlleva usar esta arquitectura en un multijugador tal y como lo es 'GTA V Online', algunas de las ventajas que se pueden ver a primera vista con esta son:

- Precio: Ya que no se necesitan servidores dedicados.
- Disponibilidad: Gracias a que no hay servidores que mantener/arreglar, lo que lleva a que las personas puedan jugar todos los días a todas las horas (menos cuando haya una actualización en el juego).
- Experiencia 'in-game': Usar P2P permite una mejor latencia y mejora el rendimiento del juego ya que los datos se transmiten de jugador a jugador

Por otro lado, siendo ROCKSTAR una de las mayores desarrolladoras y distribuidoras de juegos del mundo ha recibido muchas críticas al respecto de la utilización de la arquitectura peer-to-peer ya que la utilización de esta lleva a diferentes problemas que al final del día les pueden salir más caras que mantener los servidores, algunos de estos problemas pueden ser:

- Seguridad: Ya que en teoría cada jugador estaba conectado directamente con los otros, se puede presentar una brecha de seguridad, un ejemplo de esto es que algún jugador sea capaz de enviar datos modificados a través de dispositivos
- Trampas y exploits: Dado que no hay un servidor central, no hay forma de prever modificaciones a las reglas no autorizadas
- Emparejamiento difícil: Es difícil poder crear una sesión por culpa que no hay un servidor que pueda controlar el tema de los niveles y las personas que entran ya que

simplemente cuando se conectan personas al multijugador los va emparejando sin ningún orden

- Rendimiento y conectividad: Antes en el artículo se comentó que los recursos se dividían en partes iguales entre todas las personas, entonces si juntamos a un poco de personas con mala conexión y las ponemos en el mismo servidor, este tendrá un rendimiento muy malo, en cambio, con otro estilo de arquitectura, prácticamente tu rendimiento depende de tu conexión
- Dependencia: Tu sesión se verá afectada si alguno de esos nodos se va, en algunos casos, mientras estás haciendo una actividad y algunas personas se salen, cabe la posibilidad que te saque de la sesión actual y te mande para una nueva, creando así muchas pérdidas en misiones

E. Retos y oportunidades de mejora

Google Stadia:

Google Stadia tiene mucho rango de mejora, algunas de ellas las diré en pocas palabras ya que estas no son el tema principal del artículo, estos son: la mejora de precios de los videojuegos y en el servicio como tal, ya que las membresías que ofrecía eran muy caras en comparación a la competencia que había empezado a crearse y tener juegos exclusivos para poder traer audiencia. Ahora con los retos y oportunidades que nos incumben serían:

- mejora de seguridad del progreso: Esto hace referencia a cuando los usuarios perdían todo el progreso que llevaban por culpa de un error en el servidor. Esto se puede solucionar por medio de la creación de un sistema de servidores redundante, esto se aplicaría en puntos específicos (por tema de costos) pero haciendo esto reduciría en gran cantidad la probabilidad de pérdida de datos
- Mejora del input lag: Este reto tiene diferentes puntos de donde se puede abordar, algunas de estas soluciones pueden ser: Optimizar los algoritmos de streaming que se usan para poder mostrar el juego en pantalla ya que esto puede ayudar con la reducción de tiempo que le demora a la información que envía el control a la pantalla del usuario.

Otra solución podría ser la creación de más servidores, esta solución es muy fácil de hacer pero en temas de costo se eleva mucho más que otras soluciones, más servidores podrían ayudar con la problemática debido que las personas estarían más cerca de los servidores, pudiendo reducir el tiempo que demora en viajar la información.

Como última opción y en mi opinión la más viable, es permitir jugar en modo local y no solo online, esto hace referencia a que las personas puedan instalar el juego en su dispositivo ya que esto reduciría el tiempo que tarda la información en ir y volver.

GTA Online:

Gracias al punto anterior se evidencia una clara lista de cosas a mejorar para poder darle una mejor experiencia al usuario, algunos de los retos que se les debe de dar más prioridad

debe de ser al tema de la seguridad dado a que afecta de primera mano al usuario, -hasta el momento que se escribe este artículo siguen existiendo estos problemas-, dado que la migración a un modelo cliente-servidor podría ser lo más óptimo, es demasiado costoso, por lo tanto algunas alternativas podrían ser: limitar el acceso a los datos de cada persona, monitoreo y detección de los datos que se envían y se reciben en caso tal que haya alguna vulneración o anomalía. Otro punto de mejora podría ser la mejora en las 'reglas', como ya lo comentamos en el punto anterior, con la falta de un servidor, las personas pueden aprovecharse de ciertos puntos débiles, -dado a que cada persona es un host para la sesión-, haciendo que los otros estén en desventaja contra ellos. Por último, -en este artículo, porque claramente existen más retos-, sería el tema de la conexión. cómo se comentó anteriormente, cada persona es el servidor, por lo tanto, si algunas se salen, algunos jugadores pueden verse afectados.

III. CONCLUSIÓN Y DISCUSIÓN

En conclusión podemos ver la importancia que tiene el análisis de los pro y contras de cada una de las arquitecturas para cada uno de los proyectos que se quieran hacer. A veces el factor costo-beneficio es más importante, en mi opinión GTA V Online peca de esto dado que tiene una de las ips (propiedad intelectual) más grandes del mundo, -sino la más grande-, porque prefirió un arquitectura menos costosa que otras, -como lo sería el cliente-servidor-, pero hasta el día de hoy compromete la seguridad de sus usuarios y el balanceo de su propio juego debido a la cantidad de personas que se aprovechan de estas brechas para sacar ventaja. Por otro lado podemos ver que aunque se tenga la mayoría de puntos analizados, estudiados y se tenga toda la infraestructura para llevar a cabo el proyecto, a veces pueden fallar, como lo hizo 'Google Stadia' debido a esos detalles como la experiencia de juego que se veía muy afectada por el retraso en la imagen. Pero en mi opinión, fue un producto revolucionario, ya que le permitía el acceso a los videojuegos casi a cualquier persona que tu tuviese capacidades de una consola o pc capaz de poder ejecutar los juegos.

REFERENCIAS

- [1] IBM. "What is SOA (Service-Oriented Architecture)? — IBM". IBM - Deutschland — IBM. <https://www.ibm.com/topics/soa> (accedido el 20 de febrero de 2023).
- [2] K. P. "Cloud Gaming : Stadia vs Nvidia GeForce Now". Medium. <https://medium.com/@kevinp11/cloud-gaming-stadia-vs-nvidia-geforce-now-2789c4575826> (accedido el 20 de febrero de 2023).
- [3] "What is Cloud Architecture? — VMware Glossary". VMware. <https://www.vmware.com/topics/glossary/content/cloud-architecture.html#:~:text=Cloud%20architecture%20is%20the%20way,used%20to%20access%20the%20cloud> (accedido el 20 de febrero de 2023).
- [4] B. Bellalta y M. Carrascosa, Cloud-gaming: Analysis of Google Stadia traffic. Barcelona. [En línea]. Disponible: https://repositori.upf.edu/bitstream/handle/10230/55530/Carrascosa_com_clou.pdf?sequence=1&isAllowed=y
- [5] "¿Qué es la arquitectura de la nube?" Trend Micro. https://www.trendmicro.com/es_es/what-is/cloud-security/cloud-architecture.html (accedido el 20 de febrero de 2023).
- [6] GazAdventures. STADIA: La consola de Google (que ya no existe). (13 de febrero de 2023). Accedido el 20 de febrero de 2023. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=WVf2WjZ-awg>

- [7] Cristian Meneses. Arquitectura en la Nube. (3 de agosto de 2021). Accedido el 20 de febrero de 2023. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=JH1P3wNOFew>
- [8] R. Leadbetter. "The big interview: Phil Harrison and Majd Bakar on Google Stadia". Eurogamer.net. <https://www.eurogamer.net/digitalfoundry-2019-google-stadia-phil-harrison-majd-bakar-interview> (accedido el 20 de febrero de 2023).
- [9] "Rockstar trabaja en un parche para arreglar los graves problemas de seguridad de GTA Online y espera tenerlo listo muy pronto". Hobbyconsolas. <https://www.hobbyconsolas.com/noticias/problemas-seguridad-gta-online-ya-estan-manos-rockstar-1189904> (accedido el 20 de febrero de 2023).
- [10] "Rockstar trabaja en un parche para arreglar los graves problemas de seguridad de GTA Online y espera tenerlo listo muy pronto". Hobbyconsolas. <https://www.hobbyconsolas.com/noticias/problemas-seguridad-gta-online-ya-estan-manos-rockstar-1189904> (accedido el 20 de febrero de 2023).
- [11] T. Walker, B. Gilhuly, A. Sadeghi, M. Delbosc y S. L. smit. "Predictive Dead Reckoning for Online Peer-to-Peer Games". IEEE Xplore. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10020168> (accedido el 20 de febrero de 2023).
- [12] A. S. Tanenbaum y M. v. Steen, Distributed Systems. Createspace Independent Publishing Platform, 2017.
- [13] <https://disruptiveludens.wordpress.com/2019/03/21/concretando-sobre-stadia/>
- [14] Contributors to Wikimedia projects. "Peer-to-peer - Wikipedia". Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Peer-to-peer:_text=Unstructured%20peer-to-peer