

Prototipo de una arquitectura multiservidor basada en el modelo utilizado en los juegos en línea tipo MMO (Massively Multiplayer Online)

Prototype of a multiserver architecture based on the model used in online games like MMO (Massively Multiplayer Online)

Marco Antonio Ramírez Molina

Mtro. de Tecnologías de la Información y la Comunicación
Correspondencia al autor:

ramirezmolina.marco@gmail.com

Recibido: 01 de agosto 2019 | Revisado: 21 de agosto 2019 | Aprobado: 28 de septiembre 2019

Asesorado por: Mtro. Tecnologías de la Información y la Comunicación **Héctor Alberto Heber Mendía Arriola**
hector.mendia@gmail.com

Resumen

El uso de la tecnología como medio de aprendizaje se incrementa en la actualidad de manera rápida y variada a tal punto que el uso de dispositivos móviles es una tendencia marcada en los años recientes, lo cual puede representar una nueva modalidad en el proceso de enseñanza y aprendizaje. En este estudio se detalla un prototipo de arquitectura, que usa como base el modelo de los juegos masivos multijugador en línea MMO, toma en cuenta parte de sus componentes para lograr que funcione y sobre esta arquitectura, se implementa una plataforma educativa para otorgar servicios a miles de usuarios concurrentes.

La investigación es de alcance descriptivo, y pretende demostrar la experiencia del usuario a través de las subvariables cantidad en milisegundos para la carga de la aplicación, usuarios concurrentes y número de partidas efectuadas. Como resultado de lo anterior, se identifica que la arquitectura evaluada responde de manera adecuada, en tiempo de respuesta y la cobertura en cantidad de usuarios, concluyendo así que la arquitectura ofrece eficiencia, disponibilidad y escalabilidad.

Palabras clave

Arquitectura, cluster, multiservidor, balanceo de carga.

Abstract

The use of technology as a means of learning is currently increasing in a fast and varied manner, such as the use of mobile devices a trend that has been marked in recent years, which may represent a new modality in the Teaching and learning process.

In this work a prototype of architecture is detailed, that uses as a base the model of massive multiplayer games in line MMO, takes into account part of its components to achieve and on this architecture, an educational platform is implemented to grant services to thousands of concurrent users.

The investigation of descriptive scope and pretends to demonstrate the user experience through the subvariables amount in milliseconds for the loading of the application, concurrent users and number of games made. As a result of the above, it is identified that the evaluated architecture responds adequately, in response time and coverage in number of users, concluding that the architecture offers efficiency, availability and scalability

Keywords

Architecture, Cluster, Multiserver, Load Balancing.

Introducción

Este estudio muestra un prototipo de arquitectura que ha tomado el modelo de los juegos en línea, Massive Multiplayer Online, un estudio de la arquitectura tradicional, cliente – servidor y la comparación de las capacidades con una arquitectura basada en más de un servidor, que tiene como principal componente el uso de tecnología de balanceo de carga, ya que según Ayala y De la Vega (2010) 6.7 millones de estudiantes han tomado cursos en línea.

Desarrollo del estudio

El prototipo de arquitectura tiene como base la estructura utilizada en los juegos en línea masivos multijugador y toma en cuenta lo mínimo de este modelo.

Se combina software de tipo propietario y de código abierto, de los primeros se cuenta con sistema operativo Windows Server Standard 2012 R2, para gestión de bases de datos, SQL Server Standard 2008 R2, así como los servicios de Internet Information Services, 8.5 y para el balanceo de carga, el servicio NLB (Network Load Balancing). Para el caso del software abierto, la plataforma educativa es Moodle 3.2.2 y PHP 5.6.30 y aplicaciones de acceso libre de ANAYA.

La integración conlleva las configuraciones necesarias para lograr que se trabaje de manera adecuada y destaca el acondicionamiento de los archivos php.ini y config.ini ficheros que gobiernan la funcionalidad del servicio PHP y de la plataforma Moodle, respectivamente.

Se definen parámetros base de comparación, que consisten en escenarios de 50, 100, 150 y 250 usuarios concurrentes que deben ingresar a la plataforma en un tiempo no mayor a los 5,000 milisegundos.

Resultados obtenidos

La propuesta se basa en la ejecución de un clúster de servidores gestionados por un servicio de balanceo de carga, para determinar la carga de red en un modelo

distribuido en cuatro capas, los elementos son los siguientes: la primera, dispositivos móviles o de escritorio, que acceden vía web a los servicios, la segunda el servicio de balanceo de carga, que distribuye las peticiones en los nodos integrantes de la arquitectura.

Una tercera capa que aloja los servicios de la plataforma educativa a través de dos servidores web que a su vez son el *front end* de la solución y una cuarta que es el servidor de base de datos.

En la figura 1 se observa cómo se distribuye la arquitectura tomando en cuenta cada una de las capas mencionadas.

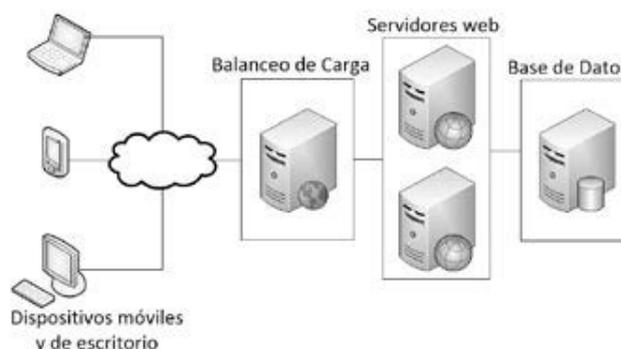


Figura 1.

Diagrama de la arquitectura multiservidor.

La evaluación compara dos arquitecturas: cliente / servidor y cliente / multiservidor, en ambos casos, cuatro grupos de usuarios concurrentes, definidos previamente, para determinar capacidades en el tiempo de respuesta y rendimiento de las arquitecturas evaluadas, el tiempo máximo esperado es de 5,000 milisegundos.

Tabla 1

Resumen de la actividad en cliente – servidor

Usuarios	50	100	150	250
Media (ms)	1,239	1,371	1,736	10,262
CPU (%)	24	41	59	76
RAM (%)	45	49	55	64
Bits	296,914	559,064	812,312	1,056,744

Fuente: elaboración propia.

Para el caso del ambiente cliente / servidor, los resultados se pueden observar en la tabla 1, destaca cómo el tiempo de respuesta, en el escenario de mayor cantidad de usuarios concurrentes, excede en más del 50 % el parámetro definido. En el modelo multiservidor, se obtienen los resultados que se ven en la tabla 2 los cuales corresponden a la actividad de los nodos integrados en la arquitectura, donde W1 y W2 corresponden a los servidores que integran la arquitectura.

En ambos casos, se observa que, en el escenario de 250 usuarios, el resultado está debajo del parámetro definido de 5,000 milisegundos.

Tabla 2
Resumen de la actividad en cliente – multiservidor

Servidor	Media (ms)	CPU	RAM	Bits	Usuarios
W1	1,516	13%	23%	138,424	20
W2	974	11%	24%	191,432	30
W1	1,642	24%	21%	246,112	40
W2	1,103	21%	21%	351,584	60
W1	1,774	41%	24%	402,776	70
W2	1,135	28%	25%	462,648	80
W1	2,835	60%	28%	602,144	110
W2	1,546	48%	31%	771,736	140

Fuente: elaboración propia.

Discusión de resultados

La arquitectura cliente / servidor muestra limitaciones en el último de los escenarios de experimentación, esto supone que la arquitectura se ve comprometida y debido al alto consumo de procesador y memoria, existe la posibilidad para que se generen errores que hagan inestable al equipo y de pauta a que los usuarios deban esperar un tiempo mayor al promedio deseado o se generen pérdidas de peticiones. En cuanto a los resultados del prototipo implementado, se observa una mejora en cada uno de los indicadores evaluados.

A partir de estos resultados se distingue cómo el problema de escalabilidad y eficiencia se hace evidente en una arquitectura de un solo servidor, pues como lo indica Vidal Rojas (s.f.) este modelo llega a representar, en ciertos niveles, un cuello de botella para los usuarios. Se suma el aspecto de latencia que se genera por la transmisión masiva de datos en un solo canal, lo que se puede observar en el comportamiento del adaptador de red, véase tabla 1.

Por su parte, los resultados para el modelo cliente / multiservidor, son favorables al objetivo de contar con una arquitectura que otorgue rendimiento y escalabilidad, la disminución de latencias y el mejor rendimiento de los procesadores, lo que permite que la experiencia del usuario, sea positiva debido a la función de compartir las peticiones de trabajo y tráfico que ofrece el sistema de balanceo de carga (Díaz, Ruiz y Sinisterra. 2012).

Se puede establecer cómo las interacciones de los usuarios con la plataforma implementada, tienen ventajas sobre la arquitectura tradicional, lo cual evidencia una mejora en la experiencia del usuario toda vez que se puede generar una percepción positiva del uso del sistema propuesto (Cordoba-Cely, 2013) debido a la selección de los elementos, estrategia de implementación y alcance de los requerimientos funcionales, dos de cinco áreas básicas para lograr un efecto positivo en los usuarios que propone Garret (2011).

Conclusiones

1. Se determinan diferencias de rendimiento. Los tiempos de respuesta en el cliente ascienden a 2,835 milisegundos lo cual implica una reducción de 7,427 milisegundos comparado con el modelo tradicional.
2. El esquema propuesto representa la posibilidad de dar cobertura al menos a 26,214 usuarios y si se toma en cuenta las mayores capacidades de los equipos, el número se estima en 419,430;

se identifica una reducción de los costos, en por lo menos un 29 %.

3. Es favorable el uso de Internet Information Server 8.5, PHP 5.6.30, Moodle 3.2.2 y cursos interactivos de naturaleza libre, de Anaya. Todo lo anterior sobre Windows server 2012 Estándar R2 y SQL server 2008 R2
4. Se implementa un sistema de balanceo de carga con la funcionalidad NLB, lo cual permite demostrar el procesamiento horizontal requerido dentro de la arquitectura propuesta; el sistema de balanceo tiene una capacidad máxima de controlar 32 nodos.
5. Se determina la capacidad de la arquitectura prototipo, con datos para cada uno de los parámetros definidos como indicadores de la variable experiencia del usuario con apoyo de las herramientas Webserver stress tool y monitor de rendimiento.

Recomendaciones

1. Indagar sobre aplicaciones que sean capaces de sincronizar altos volúmenes de datos que sean generados por las transacciones realizadas dentro del clúster.
2. Ampliar la investigación en aspectos de seguridad perimetral, sobre todo para determinar que esto no afecta los tiempos de respuesta en las peticiones de los clientes.
3. Es conveniente la investigación acerca de software de plataformas educativas que permitan una interacción del usuario con funcionalidades más avanzadas en relación con juegos por roles.
4. Investigar la implementación de algoritmos que permitan comprobar el avance y logros obtenidos por los estudiantes y cómo resultado de esto, la aplicación educativa, ofrezca reconocimientos virtuales a los educandos.

Referencias

- Aguilar, I. y Ayala, J. (2014). Las propiedades técnicas deseables en las plataformas educativas y herramientas de autor como productoras de contenido estandarizado. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*. Universidad Autónoma De Estado De México.
- Córdoba-Cely, C. (2013). La Experiencia del usuario: de la utilidad al afecto. *Revista Iconofacto* Vol. 9, No. 12 pp. 56 – 70 Medellín, Colombia.
- Díaz, T.; Ruiz, E.; Sinisterra, M. (2012). Clúster de balanceo de carga y alta disponibilidad para servicios web y mail. *Informador Técnico* (Colombia) Edición 76, Enero - Diciembre 2012, p. 93 – 102.
- Garret, J. (2011). *The elements of user experience*. Berkeley, CA: New Riders. p 24.
- Vidal, J. (s.f.). Nuevo modelo de objetos acoplados para interfaces multiusuario. Universidad de Chile.

Información del autor

Licenciado en Administración de Empresas, graduado en la Facultad de Ciencias Económicas de la Universidad San Carlos de Guatemala, 2009.

Maestría en Tecnologías de Información y la Comunicación de la Escuela de Estudios de Postgrado de la Facultad de Ingeniería, Universidad de San Carlos de Guatemala, 2017.