

Implementación de un entorno de alta disponibilidad en un Sistema de Información Clínica



Gonzalez Bernaldo de Quirós F., Gómez A., Martínez M., Luna D., Severino J., Conosciuto H., Agata A., Riquelme N.

Departamento de Información Hospitalaria, Hospital Italiano de Buenos Aires, Argentina

Resumen

Asegurar la menor cantidad de tiempo fuera de línea de un sistema informático en la capa clínica o asistencial es una condición indispensable para el éxito de dicho sistema. El Hospital Italiano de Buenos Aires tiene un sistema de información clínica implementado desde el año 1998 y se vio en la necesidad de implementar un sistema de alta disponibilidad para asegurar un servicio 24x7. Se presentan los aspectos metodológicos y prácticos de la implementación de dicho entorno de alta disponibilidad en nuestra institución.

Palabras Clave

Sistemas de Información Clínica, Alta disponibilidad, Servicios basados en web, Java Enterprise Edition

Introducción

Todos los sistemas informáticos son vulnerables a un amplio espectro de amenazas tanto internas como externas a la institución donde están insertos y la caída de los mismos pueden provocar daños importantes a su funcionamiento [1]. Las instituciones financieras han tomado un enfoque más agresivo en cuanto a la instalación y manejo de sus sistemas de información críticos [2]. Si bien el tiempo fuera de línea de un sistema puede no ser algo dramático para el personal administrativo si lo es para el personal asistencial [3]. El avance en la informatización de la capa clínica de una institución hace que los miembros del equipo de salud dependan en gran parte de la Historia Clínica Electrónica para realizar el proceso de atención y brindar calidad en este servicio, las caídas de sistema o “*down-time*” generan que el médico no tenga disponible los datos médicos del paciente en el momento que la necesita teniendo que recurrir a la información que el mismo pueda citar, obteniendo datos incompletos y fragmentados. El proceso es mucho más complejo, cuando se trata de áreas críticas o internación, donde las realidades y complejidades médicas son mayores. Dependiendo del nivel de informatización de capa clínica de las instituciones de salud, en donde a mayor automatización mayor dependencia, el impacto tanto organizacional como económico de las caídas de sistema son cada vez más claros [4].

El *Hospital Italiano de Buenos Aires* es un hospital universitario de alta complejidad que cuenta con 550 camas de internación y más de 400 consultorios ambulatorios en 20 centros de atención distribuidos en la Capital Federal y el Gran Buenos Aires. Desde el año 1998

está implementando un sistema de información clínico en sus diferentes niveles de atención que se dio a llamar como “Proyecto ITALICA” [5, 6]. Dicho sistema está basado en estándares y permitió la integración de múltiples aplicaciones preexistentes [7-10]

Debido a que tanto la rapidez de respuesta del sistema como el tiempo en el cual este no esta disponible son tomados como factores determinantes (entre otros) de éxito en la implementación de sistemas de información hospitalarios [11], una vez que el proyecto ITALICA estuvo en marcha e implementado en su totalidad nuestro nuevo desafío fue montar un sistema que pueda brindar servicio interrumpidamente. El presenta trabajo versa sobre los aspectos metodológicos y prácticos de la implementación de un entorno de alta disponibilidad en el sistema de información clínico del Hospital Italiano de Buenos Aires.

Metodología

El desarrollo del modelo tecnológico esta basado en la implementación de la arquitectura J2EE bajo un entorno de alta disponibilidad de múltiples capas. Para crear un sistema de alta disponibilidad utilizamos una arquitectura basada en *Clustering* la cual involucra a un grupo de servidores que representan nodos independientes pero trabajan juntos como si fueran un solo sistema. El método de Cluster agrupa *instancias replicadas* que tienen acceso a los mismos servicios, estos servicios requieren una infraestructura que permita dar soporte concurrente a múltiples usuarios respondiendo a las peticiones en unos pocos segundos. Los sistemas basados en cluster están basados en múltiples servidores por lo que permite escalar en forma horizontal, es decir que cuando el sistema necesita mas recursos simplemente se agregan nuevos nodos (servidores) al esquema provocando un aumento de procesamiento simétrico en el sistema.

Uno de los puntos mas importantes que hacen que una aplicación tenga buena performance y disponibilidad es el direccionamiento del trafico a través de distintos servers, el balanceo de carga distribuye los pedidos de servicio sobre un set de servidores disponibles mejorando la utilización de los recursos de Hardware, debido a la división de las peticiones por la cantidad de proveedores de servicios disponibles.

La capa del balanceo de carga se puede realizar con múltiples metodologías. La solución implementada en nuestro proyecto fue la instalación del Oracle 10g Web Cache.

El Oracle 10g Web Cache se ubica delante de los servidores de aplicación e intercepta todos los pedidos HTTP enviados al Server y responde con una validación de la versión cache disponible aumentando la velocidad por no tener que ejecutar la petición en el servidor de aplicaciones o base de datos, dado que otro usuario ya ejecutó la misma función y el cache la tiene almacenada, esto disminuye además la utilización de los recursos del Hardware, destinando los recursos a peticiones que no están siendo ejecutadas en ese momento.

La distribución de estas peticiones ocurre de acuerdo a la capacidad relativa de cada servidor, analizando previamente el porcentaje de utilización de los servicios y derivando al de mayor disponibilidad de recursos.

El manejo del tráfico es una solución que ofrece mucho más que la distribución de las peticiones, dado que si uno de los servidores falla, el balanceador detecta la falta de

recursos direccionando él tráfico a los servidores disponibles provocando la continuidad de los servicios y siendo totalmente transparente a la aplicación y al usuario.

En la **capa de aplicaciones** utilizamos dos servidores de aplicaciones Oracle IAS 10g. Oracle 10g es un contenedor J2EE compuesto por un grupo de servidores replicados que proveen alta disponibilidad y escalabilidad de forma transparente.

La metodología utilizada por los servidores de aplicación para brindar permanente servicio esta basada en la replicación de las sesiones en las distintas Java Virtual Machines que dan soporte a las aplicaciones.

A su vez las Java Virtual Machine replican sus aplicaciones en distintas islas internas de procesamiento dividiendo la petición en múltiples hilos de procesamiento, aumentando la performance y la disponibilidad en caso de fallas de la JVM.

Esta metodología permite que el usuario no tenga demoras o cortes de servicios en el momento que se produce una falla, donde no solo puede ser producto del hardware, sino que también contempla fallas de software de aplicación.

Cuando la aplicación cliente solicita una petición de servicio ya sea la ejecución de un servlet o EJB y el servicio falla, el manejador de procesos redirecciona la petición a otro contenedor preservando el estado de la aplicación y creando una reconstrucción del servicio totalmente transparente al usuario (Figura 1).

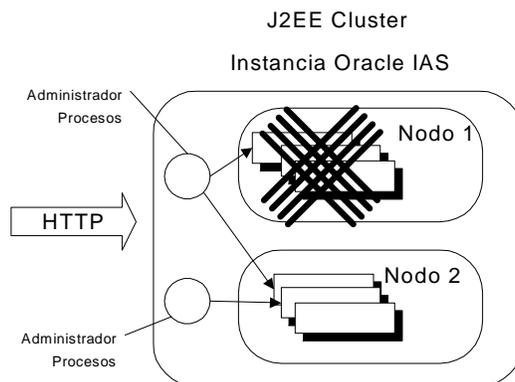


Figura 1: redirección de peticiones

El servidor de aplicaciones también utiliza la tecnología cache almacenando en memoria las páginas web que son frecuentemente instanciadas utilizando pocos recursos del server y provocando un alto crecimiento de performance en la aplicación, además reduce el riesgo de ataque externos con tecnologías que producen “denial-of-service” y mejoran la disponibilidad del contenido estático y dinámico de la aplicación, debido a que el servidor de aplicaciones genera múltiples instancias cache que trabajan juntas como si fuera solo un simple cache lógico.

Además, las funciones más solicitadas son almacenadas en caches múltiples comunicándose unas a otras detectando cambios y fallas.

En la **capa de base de datos** implementamos un cluster de 2 servidores utilizando ORACLE RAC 9i.

Real Application Cluster, administra las instancias de base de datos como si fueran solo una, utilizando tecnología interconectada para monitorear las instancias de los distintos servidores, almacenando los datos en un Share File System.

RAC explota la redundancia utilizando n-1 nodos y n-nodos por cluster, de esta forma provee disponibilidad a grandes volúmenes de datos sin interrupciones en el servicio, generando una rápida reconstrucción de transacciones truncadas.

El entorno es disponible y redundante porque todos los nodos acceden a todos los datos y no afecta a los otros nodos en su capacidad de procesar las transacciones en el cluster.

Modelo Tecnológico

El sistema está montado en una arquitectura de alta disponibilidad de múltiples capas que da soporte a aplicaciones desarrolladas en J2EE basadas en Web (Figura 2).

El cliente de las aplicaciones es ultra fino utilizando como única interfaz un navegador.

En la **capa de balanceo de carga** se implementaron servidores Linux Intel con Suse enterprise edition, montados en el ORACLE WebCache 10g.

La **capa de aplicaciones** se implementó sobre servidores Linux Intel multiprocesados con Suse enterprise edition, montados en el ORACLE IAS 10g y un servidor Linux Intel de infraestructura que da soporte a la administración del cluster de aplicaciones con ORACLE IAS 10g Infrastructure.

La **capa de base de datos** se implementó sobre servidores SUN Enterprise 280 multiprocesados, Sun Cluster 3.0 y ORACLE Real Application Cluster 9i.

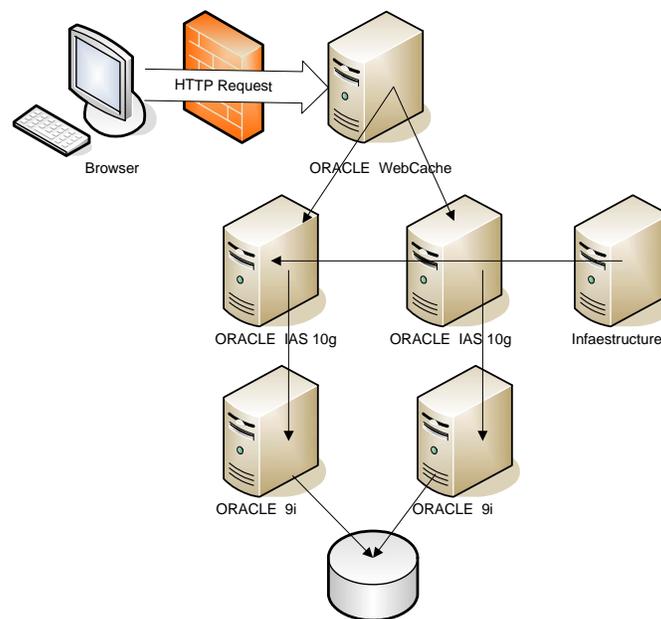


Figura 2: modelo tecnológico de la arquitectura de alta disponibilidad en múltiples capas

Resultados

En los entornos de computación modernos es sumamente sencillo desde el punto de vista teórico brindar alta disponibilidad.

Los sistemas típicamente tienen redundancia en sus componentes de Hardware y Software que hacen que el sistema este disponible en todo momento.

Cuando un componente falla solo tiene que ponerse en marcha el componente de backup del mismo tipo y ejecutar el proceso que reenvía los requerimientos, reconstruyendo el servicio en un tiempo imperceptible para el usuario.

Pero este NO es el escenario común en un entorno de sistemas médicos, a medida que se avanza con las implementaciones de nuevas herramientas cada vez mas el sistema se convierte de *misión-critical* a un sistemas *business-critical*, donde no solo es importante su disponibilidad, sino que también es importante su performance y escalabilidad, dado que los sistemas en salud están permanentemente en crecimiento implementando nuevas tecnologías y procesos que ayudan al equipo de salud en su proceso asistencial, mejorando la calidad de atención y disminuyendo los errores médicos.

La implementación de productos ORACLE en la capa de balanceo de carga, aplicaciones y base de datos nos dio un completo control de los procesos generando disponibilidad 24x7, obteniendo además control sobre los downtime planeados y errores de usuarios.

Desde el punto de vista de la administración de la base de datos y servidores de aplicación es un proceso sin complicaciones a cargo del área de tecnología de la institución con tareas comunes de administración de base de datos, las herramientas son sumamente intuitivas y completas.

El proceso de migración de datos se completó en dos semanas sin haber presentado complicaciones en el pasaje de la información del antiguo sistema.

El sistema contempla además un sistema de Disaster Recovery, implementando la tecnología DataGuard. Esta tecnología cuenta con un servidor espejado de las instancias de la base de datos en producción ubicado físicamente en un lugar seguro y lejos de la base de datos primaria, que se pone en funcionamiento en el caso de falla del cluster en general.

Discusión

La implementación de un entorno de alta disponibilidad para asegurar la menor cantidad de tiempo fuera de línea de un sistema informático en la capa clínica o asistencial es una condición indispensable para el éxito de dicho sistema. Desde el punto de vista tecnológico asegurar tal disponibilidad de la información y dar un soporte adecuado al usuario 24x7 son los objetivos más importantes a perseguir.

Referencias

1. Kilbridge, P., *Computer crash--lessons from a system failure*. N Engl J Med, 2003. **348**(10): p. 881-2.
2. Fleegler, E.M., *Computer crash*. N Engl J Med, 2003. **348**(23): p. 2365-6.
3. Mazzoleni, M.C., P. Baiardi, and I. Giorgi, *Lesson learnt from a halt of the hospital information system*. Stud Health Technol Inform, 1999. **68**: p. 102-5.
4. Anderson, M., *The toll of downtime. A study calculates the time and money lost when automated systems go down*. Healthc Inform, 2002. **19**(4): p. 27-30.

5. Luna, D., P. Otero, A. Gomez, M. Martinez, S. García Martí, M. Schpilberg, A. Lopez Osornio, and F.G. Bernaldo de Quiros. *Implementación de una Historia Clínica Electrónica Ambulatoria: "Proyecto ITALICA"*. in *6to Simposio de Informática en Salud - 32 JAIIO*. 2003. Buenos Aires, Argentina: Sociedad Argentina de Informática e Investigación Operativa (SADIO).
6. Gonzalez Bernaldo de Quiros, F., E. Soriano, D. Luna, A. Gomez, M. Martinez, M. Schpilberg, and A. Lopez Osornio. *Desarrollo e implementación de una Historia Clínica Electrónica de Internación en un Hospital de alta complejidad*. in *6to Simposio de Informática en Salud - 32 JAIIO*. 2003. Buenos Aires, Argentina: Sociedad Argentina de Informática e Investigación Operativa (SADIO).
7. Gomez, A., F.G. Bernaldo de Quiros, L. Garfi, D. Luna, G. Sarandria, A. Figar, M. Martinez, F. Campos, and K. D. *Implementación de un sistema de mensajería electrónica -HL7- para la integración de un sistema multiplataforma*. in *4to Simposio de Informática en Salud - 30 JAIIO*. 2001. Buenos Aires, Argentina: Sociedad Argentina de Informática e Investigación Operativa (SADIO).
8. Garfi, L., P. Navajas, A. Gomez, D. Luna, and F.G. Bernaldo de Quiros. *Implementación de un sistema centralizado para la identificación de pacientes en un hospital de alta complejidad*. in *5to Simposio de Informática en Salud - 31 JAIIO*. 2002. Santa Fe, Argentina: Sociedad Argentina de Informática e Investigación Operativa (SADIO).
9. Gomez, A., F.G. Bernaldo de Quiros, F. Campos, M. Martinez, D. Luna, G. Sarandria, D. Calvo, G. Lopez, D. Kaminker, and L. Garfi. *Implementación de mensajería HL7 para la admisión y egresos de pacientes internados*. in *5to Simposio de Informática en Salud - 31 JAIIO*. 2002. Santa Fe, Argentina: Sociedad Argentina de Informática e Investigación Operativa (SADIO).
10. Gomez, A., F. Campos, M. Martinez, D. Luna, G. Staccia, G. Cifarelli, D. Kaminker, D. Calvo, G. Sarandria, and F.G. Bernaldo de Quiros. *Implementación de mensajería HL7 en un sistema de solicitud de exámenes complementarios*. in *5to Simposio de Informática en Salud - 31 JAIIO*. 2002. Santa Fe, Argentina: Sociedad Argentina de Informática e Investigación Operativa (SADIO).
11. Van Der Meijden, M.J., H.J. Tange, J. Troost, and A. Hasman, *Determinants of success of inpatient clinical information systems: a literature review*. *J Am Med Inform Assoc*, 2003. **10**(3): p. 235-43.

Datos de Contacto:

Fernán González Bernaldo de Quiros. Jefe Departamento de Información Hospitalaria. Hospital Italiano de Buenos Aires. Gascón 450. Ciudad Autónoma de Buenos Aires. Argentina. CP 1199.
fernan.quiros@hospitalitaliano.org.ar