

Índice

01.- Introducción	pág. 02
02.- Motivación del Proyecto	pág. 03
03.- Tecnología	pág. 04
03.1.- Breve reseña de Clustering	pág. 04
03.2.- JBoss	pág. 06
03.3.- Terracotta	pág. 07
03.3.1.- Tomcat	pág. 08
04.- Aplicación web de ejemplo	pág. 09
05.- Casos de uso de la aplicación web	pág. 10
06.- Calendario	pág. 11
07.- Referencias	Pág. 13

01.- Introducción

El propósito del presente Trabajo Profesional, es analizar y comparar diferentes arquitecturas necesarias para el desarrollo de una Aplicación Web focalizada en brindar escalabilidad, alta disponibilidad y alto desempeño, junto con la estructuración, configuración, administración y despliegue de un entorno productivo con soporte de Clustering. Particularmente, se desean comparar las arquitecturas a desarrollar teniendo en cuenta la utilización de las herramientas Terracotta y JBoss.

Ambos entornos será evaluados y validados mediante la interacción con una aplicación web de prueba, con la que se pretende usufructuar los beneficios de alta disponibilidad, escalabilidad, y alto desempeño que deberían ofrecer ambas herramientas.

La aplicación web será en cuanto a la funcionalidad la misma al probar el entorno montado con JBoss como el entorno montado con Terracotta, con la arquitectura adecuada a cada entorno, el cual es condicionado por cada herramienta.

El proyecto será desarrollado tal como implica la tecnología ya mencionada, con software open source: lenguaje de programación Java, entorno de desarrollo Eclipse, servidor JBoss, Terracotta, entre otros.

02.- Motivación del Proyecto

A medida que van apareciendo más aplicaciones web integrales que resuelven cada vez más y más aspectos del negocio que intentan abarcar, más se va dependiendo de dichas soluciones, y más impacto tienen las caídas de los sistemas. Por lo cual, solamente tener una muy buena aplicación web, algo para nada fácil de lograr por cierto, es solo una parte del desafío si se quiere desarrollar una aplicación realmente exitosa, otra parte del desafío es lograr que esté disponible prácticamente siempre.

Una de las soluciones posibles para intentar que las aplicaciones estén online casi bajo cualquier condición, y particularmente la solución que nos interesa, es implementar Clustering en el entorno productivo de la aplicación. De esta forma, por las características propias de la técnica, se puede de mantener la aplicación online incluso en caso de que alguna de las computadoras que conforman el Cluster deje de funcionar, entre otros inconvenientes que podrían surgir.

Por otro lado, vemos cómo a nivel laboral también aumenta paulatinamente el interés por la técnica en cuestión, lo que nos lleva a concluir que dominar la técnica constituye una manera más de defenderse y destacarse a nivel laboral, en un mercado muy dinámico y competitivo.

Revisando las tecnologías disponibles, vemos que podríamos montar un ambiente de tales características usando herramientas que no nos son enteramente ajenas, como ser el servidor JBoss, o una librería de Java denominada Terracotta, entre otras herramientas finalmente descartadas como el servidor Glassfish. También notamos que las herramientas elegidas proponen dos maneras de abordar la implementación de un ambiente de Clustering.

JBoss propone replicar partes de una aplicación en varios servidores sincronizándolos mediante mecanismos provistos por el mismo JBoss, pero esto requiere efectuar implementaciones específicas requeridas por el servidor de aplicaciones para poder realizar esta sincronización, condicionando el diseño de la aplicación.

Terracotta propone montar una aplicación pensada para un servidor y desplegarla en todos los servidores de manera que no tengan conocimiento de existencia entre si. El encargado de la sincronización es justamente esta misma librería, la cual no requiere implementaciones específicas para esta tarea, por lo que en teoría se puede replicar cualquier aplicación.

Ambas maneras son muy atractivas e interesantes, incluso una es prácticamente opuesta a la otra, y se podrá implementar una u otra según las necesidades del caso en una situación del mundo laboral. Finalmente, elegimos las herramientas mencionadas porque nos permiten abordar el área de interés con variantes, lo que consideramos que favorecerá a la riqueza de nuestra investigación.

En consecuencia, es muy interesante para nosotros poder llegar a dominar la técnica de Clusterización, y su implementación de dos maneras distintas, ya que es una técnica que nos llama la atención, y que podríamos aplicar a nivel laboral, con herramientas de fácil acceso.

03.- Tecnología

03.1.- Breve reseña de Clustering

El término Cluster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.

Hoy en día desempeñan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

La tecnología de Clusters ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con Clusters surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, un Cluster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

Los Clusters son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador típicamente siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

De un Cluster se espera que presente combinaciones de los siguientes servicios:

- 1.- Alto rendimiento
- 2.- Alta disponibilidad
- 3.- Balanceo de carga
- 4.- Escalabilidad

La construcción de los ordenadores del Cluster es más fácil y económica debido a su flexibilidad: pueden tener todos la misma configuración de hardware y sistema operativo (Cluster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (Cluster semi-homogéneo), o tener diferente hardware y sistema operativo (Cluster heterogéneo), lo que hace más fácil y económica su construcción.

Teniendo en cuenta las computadoras disponibles, se concluye que el tipo de Cluster que se implementará un Cluster heterogéneo.

Para que un Cluster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del Cluster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un Cluster puede satisfacer estos requerimientos usando los recursos que tiene asociados a él.

La tecnología Cluster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

03.2.- JBoss

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible Java. Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., adquirida por Red Hat en abril del 2006, fundada por Marc Fleury, el creador de la primera versión de JBoss. El proyecto está apoyado por una red mundial de colaboradores. Los ingresos de la empresa están basados en un modelo de negocio de servicios.

JBoss implementa todo el paquete de servicios de J2EE.

JBoss soporta Clusterización de manera nativa. Simplemente, al levantar los servidores que conforman el Cluster, se ejecuta un comando que los sincroniza y logra que se comporten en conjunto como si fueran un solo servidor.

03.3.- Terracotta

Terracotta es un software de código abierto que permite crear aplicaciones Java que pueden escalar a cualquier cantidad de computadoras, sin tener que crear código adicional o usar bases de datos para compartir datos dentro del Cluster.

Terracotta utiliza el concepto de Memoria Adjunta a la Red (NAM - Network Attached Memory). El uso de NAM le permite a Terracotta distribuir en Cluster a Máquinas Virtuales Java (JVM) directamente debajo de las aplicaciones, brindando alta disponibilidad y escalabilidad de forma transparente.

La ventaja de Terracotta es que la aplicación preparada para funcionar en Cluster es exactamente igual a una aplicación Java común. Todos los conceptos y librerías que se usan (POJOs, Spring, Hibernate, threads, sincronización, etc.) funcionan de la misma manera con Terracotta en un entorno distribuido, de la misma manera que funcionan en una única máquina virtual con muchos hilos de ejecución.

Terracotta funciona a nivel de memoria, por lo que no es necesario heredar de ninguna clase ni implementar ninguna interfaz para compatir objetos entre todas las máquinas virtuales del Cluster (ni siquiera es necesario implementar `java.io.Serializable`).

En definitiva, se puede programar la aplicación de manera natural, y se deja a Terracotta que administre todo el trabajo para crear un entorno escalable y de alta disponibilidad.

03.3.1.- Tomcat

Tomcat es un Servlet Container, el cual provee el entorno de ejecución de servlets, mediante los cuales se puede programar una aplicación web.

Los mencionados servlets son simples clases de Java, que reciben pedidos HTTP y envían respuestas que son mostradas en el navegador web del cliente.

Si bien Tomcat y Terracotta son herramientas totalmente diferentes, inclusive de diferentes compañías, éstas se complementan, ya que se puede desarrollar una aplicación web al estilo tradicional utilizando Tomcat y sin tener en cuenta aspectos de la Clusterización, y luego utilizar Terracotta para replicar esta aplicación web y lograr un comportamiento de Cluster.

04.- Aplicación web de ejemplo

La Aplicación Web que se considera para efectuar el análisis propuesto en el presente trabajo, es una aplicación administrativa para recibir trabajos y evaluar el desempeño académico de los alumnos de un instituto educativo de gran tamaño.

El escenario que se considera es el siguiente:

- 1.- Se provee un usuario del sistema a cada uno de los alumnos y a cada uno de los profesores, quienes luego incorporan su contraseña privada de acceso.
- 2.- Todos los alumnos están obligados, periódicamente, a incorporar al sistema todos los trabajos prácticos que van realizando a medida que transcurre el ciclo lectivo, junto con otra información pertinente que permite clasificar los trabajos prácticos de manera correcta.
- 3.- En el instituto hay una gran cantidad de alumnos, y se espera que en la mayor parte de las materias se les solicite muchos trabajos prácticos, los cuales pueden llegar a ser considerablemente grandes. El uso diario del sistema es muy exigente, ya que todos los días existe un muy alto tráfico de trabajos, considerando todas las materias en todos los distintos niveles educativos que ofrece la institución.
- 4.- Los profesores deben estar constantemente evaluando el material cargado, ya que necesitan dicho material para determinar la condición de regularidad de los alumnos, para calificar los trabajos prácticos, y para que a fin de año se puedan obtener estadísticas generales del desempeño del alumnado. Los profesores deben terminar sus evaluaciones, ya sea de los trabajos o la general a fin de año, en determinadas fechas. Por lo cual cabe señalar que todos los días, algunos profesores deben obligatoriamente tener preparadas ciertas evaluaciones.
- 5.- El sistema debe estar siempre disponible, ya que si así no lo hiciera, los alumnos se verían impedidos de entregar a tiempo sus trabajos, lo que pondría en peligro su regularidad, al mismo tiempo que gran cantidad de profesores se verían impedidos de terminar sus evaluaciones a tiempo. El sistema debe ser escalable porque la dirigencia del instituto planea seguir expandiendo sus actividades, lo cual indefectiblemente hará crecer la aplicación y creará más presión sobre los servidores. El sistema debe mostrar un altísimo desempeño, para poder sobrellevar que muchos alumnos incorporen sus trabajos al mismo tiempo, muchos profesores los consulten al mismo tiempo, que mientras que los alumnos cargan, los profesores evalúen, etc.

Se propone para el presente trabajo profesional, considerar la implementación de una primera versión mínima del sistema explicado, que permita a los alumnos cargar un mínimo de información, que permita a los profesores hacer una evaluación mínima de dicha información, y que permita en el futuro incorporar nuevos módulos como ser el módulo de generación de prioridades para inscripción, nuevas estadísticas generales, etc.. Esta implementación debe necesariamente cumplir con los requisitos no funcionales de escalabilidad, desempeño, y disponibilidad.

05.- Casos de uso de la aplicación web

Elegimos un conjunto de casos de uso que pondrán a prueba las características deseadas de alta disponibilidad y buen desempeño, el cual se logra a través del Clustering.

Asumimos que el sistema debe ser utilizado por toda una universidad con gran cantidad de alumnos, por lo que la carga introducida por cualquiera de los casos mencionados a continuación sera muy grande.

- 1.- Log in: proveer acceso al sistema mediante usuarios que podrán tener acceso a las pantallas de carga o evaluación según corresponda
- 2.- Subir trabajos prácticos: pantallas de carga de los trabajos prácticos de los alumnos.
- 3.- Inscripción: pantalla de inscripción de alumnos
- 4.- Búsqueda de TPs: pantalla de búsqueda de trabajos según cuatrimestre y materia.
- 5.- Visualización de TPs: pantalla de visualización de los trabajos cargados.
- 6.- Evaluación de TPs: pantalla para la calificación de los trabajos cargados por parte de los profesores.
- 7.- Seguimiento de materias en tiempo real: pantalla donde se mostrará dada una materia, la cantidad de TPs cargados, y se graficará la cantidad de TPs según su calificación, actualizando el gráfico cada 5 segundos.
- 8.- Estadística de inscriptos según ciudad: pantalla donde se mostrará la cantidad de alumnos inscriptos por ciudad.

06.- Calendario

Distinguimos en este proyecto las siguientes etapas:

- 1.- Concepción (4 semanas de duración)
 - Investigación y análisis de las herramientas disponibles que proveen Clustering
 - Selección de las herramientas que aseguren diversidad técnica, para enriquecer la investigación ; en este caso como ya se aclaró, JBoss y Terracotta
 - Análisis y elección de en qué tipo de escenario es aplicable la tecnología elegida, lo cual puede verse en la sección "05.- Aplicación web de ejemplo"
- 2.- Elaboración (4 semanas de duración)
 - Creación del repositorio público (<http://code.google.com/p/qin-cluster/>)
 - Confección de la propuesta de trabajo profesional
 - Definición de los casos de uso de la aplicación de prueba
- 3.- Análisis de la arquitectura (12 semanas de duración)
 - Análisis del escenario implementado con Terracotta, denominado "escenario 1"
 - Análisis del escenario implementado con JBoss, denominado "escenario 2"
 - Desarrollo de casos de prueba generales acordes a un entorno que soporte Clustering
 - Desarrollo de los casos de prueba particulares para el escenario 1, si corresponde
 - Desarrollo de los casos de prueba particulares para el escenario 2, si corresponde
 - Entrega de documentación reflejando las características de esta etapa, a incluir en el informe general final
- 4.- Construcción (8 semanas de duración)
 - Implementación del escenario 1
 - Desarrollo de la aplicación de prueba para el escenario 1
 - Implementación del escenario 2
 - Desarrollo de la aplicación de prueba para el escenario 2
 - Ejecución de las pruebas generales, y recolección de información
 - Ejecución de las pruebas particulares del escenario 1, y recolección de información
 - Ejecución de las pruebas particulares del escenario 2, y recolección de información
 - Entrega de documentación reflejando las características de esta etapa, a incluir en el informe general final
- 5.- Finalización (4 semanas de duración)
 - Comparación de los resultados obtenidos
 - Comparación de los resultados obtenidos
 - Documentación de las conclusiones del proyecto
 - Informe general final

Concepción				Elaboración				Análisis de la arquitectura												Construcción								Finalización			
1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	1	2	3	4
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Semanas																															

07.- Referencias

Terracotta:

<http://www.terracotta.org/>

Tomcat:

<http://tomcat.apache.org/>

JBoss:

<http://jboss.org/>

<http://es.wikipedia.org/wiki/JBoss>

Sesiones Web:

<http://www.terracotta.org/web-sessions/?was-expanded=web-sessions-submenu>

Cache distribuida:

<http://www.terracotta.org/ehcache/ehcache-dx>

<http://jboss.org/jbossccache>

Clusterización:

<http://docs.jboss.org/jbossas/jboss4guide/r4/html/cluster.chapt.html>

<http://www.terracotta.org/platform/how-terracotta-works>

<http://www.terracotta.org/platform/platform-concepts>

<http://www.terracotta.org/platform/hello-clustered-world>

http://es.wikipedia.org/wiki/Cluster_%28inform%C3%A1tica%29