



Web Service buscador de localizaciones de IP's

*Trabajo Final para la Especialización Servicios
Distribuidos*

Fa.M.A.F.

2011

Director del Trabajo: Ing. Miguel Montes

Co-director: Dr. Gabriel Infante-López

Realizado por: Ing. Pablo Pellegrinet Sánchez

Idea Preliminar

Teniendo en cuenta la gran masificación de los web services, como así también su utilidad, el objetivo del presente proyecto es elaborar un servicio web, teniendo en cuenta su optimización y comparando diferentes plataformas para poder lograr un servicio que pueda atender la mayor cantidad de solicitudes posibles.

Con motivo de realizar lo mencionado anteriormente se presenta la siguiente idea:

Crear un web services SOAP que permita realizar búsquedas de ip's y que devuelva la localización del mismo.

El objetivo de esta aplicación es ofrecer la posibilidad de brindar un servicio de buscador eficiente, similar a <http://iplocationtools.com/samples.html>

Requisitos funcionales:

1. Nunca debe dejar de funcionar.
2. Debe estar preparado para soportar un alto número de transacciones.
3. Debe brindar la información de los resultados de la búsqueda en tiempo real de lo almacenado en la base de datos.

Requisitos no funcionales:

Es requisito esencial que soporte gran número de usuarios y que nunca deje de estar disponible el servicio.

Características:

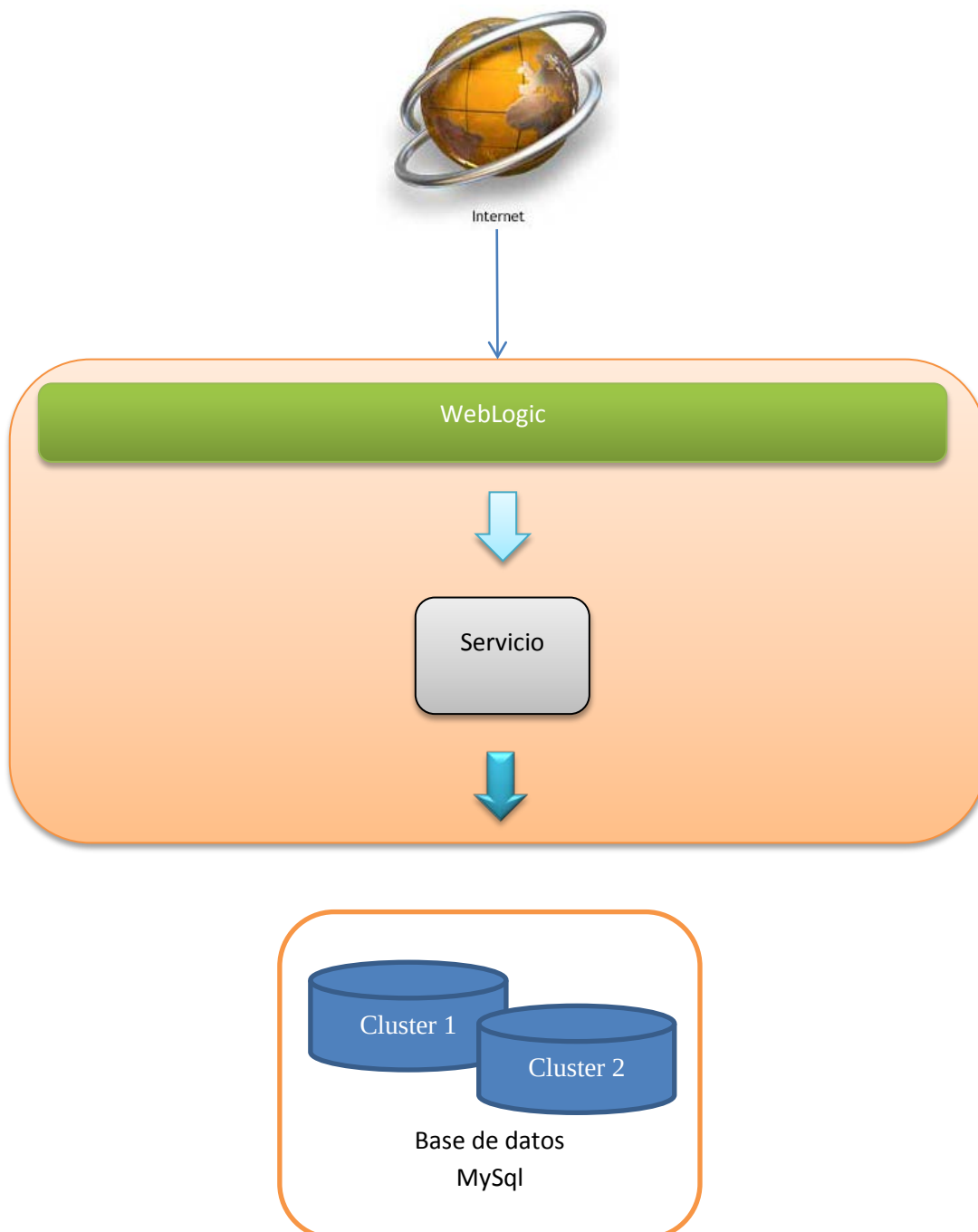
Que el web services sea escalable mediante la plataforma elegida. La plataforma deberá ser configurada para soportar la escalabilidad del servicio de acuerdo a la demanda del mismo.

Investigar y utilizar las tecnologías y/o herramientas necesarias para lograr el objetivo planteado anteriormente, tanto en lo que se refiere a plataformas como así también en base de datos, necesarias para garantizar la escalabilidad y normal servicio del sitio. Además se profundizara el estudio es la optimización de las tecnologías elegidas.

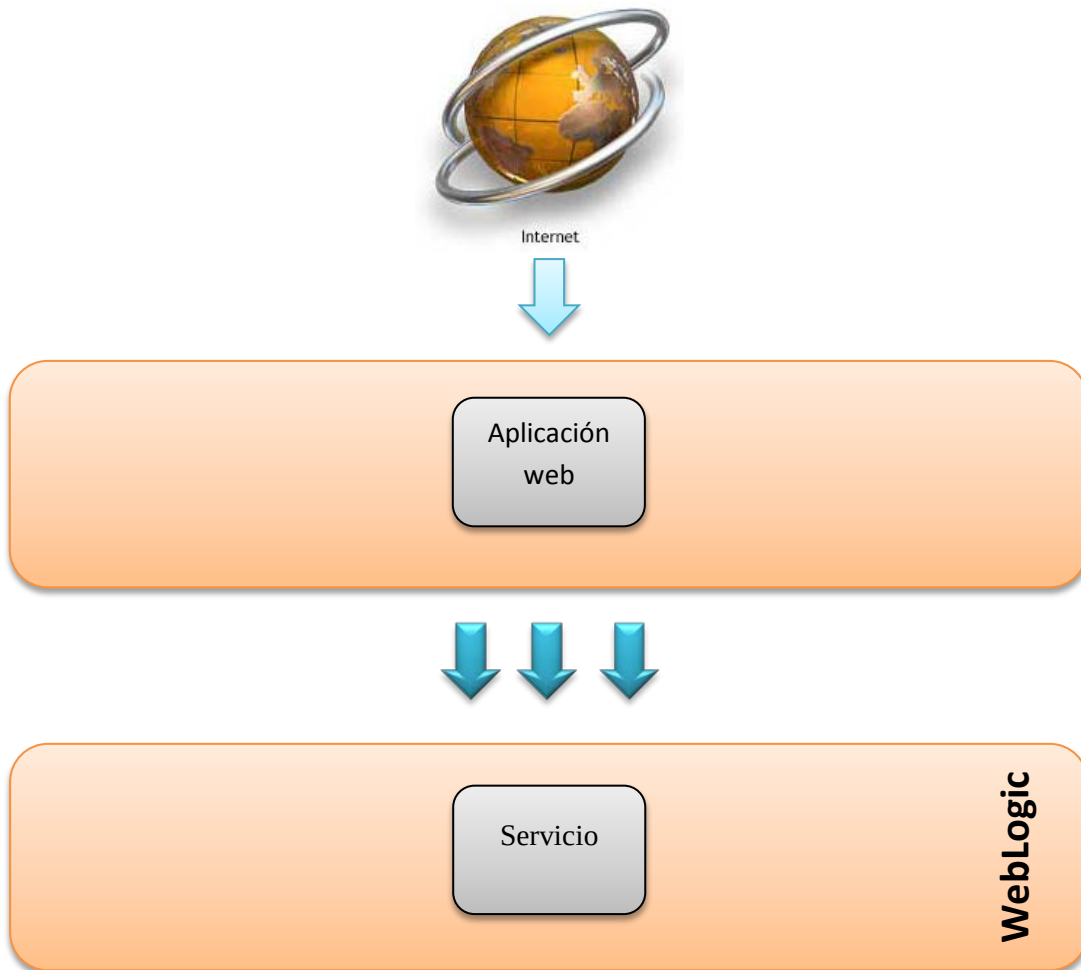
Objetivos

1. Investigar sobre la configuración, implementación y optimización de WebLogic.
2. Desarrollo de un web services en Java y otro en Php.
3. Investigación e Implementación de una base de datos en MySQL que soporte un alto número de transacciones.
4. Mediante una aplicación generar tráfico concurrente que genere solicitudes de servicio al web services.
5. Comparar el rendimiento mediante el seguimiento de las optimizaciones.

Arquitectura sugerida:



Arquitectura de ejecución del test:



Desarrollo del Proyecto:

Como resultado de la investigación realizada se llegó a las siguientes conclusiones:

WebLogic Server 11g Standar Edition:

Es la plataforma de aplicaciones web, con mejor rendimiento y respuesta del mercado, esto se debe a:

- Oracle WebLogic Server 11g brinda el mejor rendimiento comprobado. Independientemente de si se ejecutan una, dos o muchas instancias del servidor de aplicaciones para dar soporte a sus aplicaciones, Oracle WebLogic Server ofrece el rendimiento más rápido. Muchos centros de datos implementan las últimas arquitecturas de hardware, entre ellas, chips multinúcleo, memoria de 64 bits y conexiones de red de alta velocidad.
- También se ofrece una base sólida para SOA utilizando Java Platform, Enterprise Edition 5 (Java EE 5). Oracle WebLogic Server es extremadamente fácil de utilizar y ofrece desempeño, escalabilidad, disponibilidad y confiabilidad de nivel industrial. Los clientes pueden rápidamente realizar actualizaciones sobre sus actuales servicios y administrarlos con poderosas herramientas de configuración, implementación y administración.
- Utiliza Oracle JRockit: la máquina JVM más rápida en la industria. Está adaptado y optimizado para ofrecer rendimiento del lado del servidor en sistemas con memoria de 32 bits y de 64 bits en los tipos de chips más difundidos. Si bien WebLogic Server admite otras JVM, sólo Oracle JRockit logra la velocidad necesaria para imponerse ante las empresas de la competencia. Además, una variante en tiempo real proporciona rendimiento predecible y fluido, que ayuda a garantizar una experiencia uniforme a los usuarios de las aplicaciones.

Tiene la particularidad que ajusta automáticamente el número de threads de acuerdo a las necesidades y recursos disponibles.

Permite configurar que porcentaje de sockets permanecen abiertos, por defecto viene sin límites.

Siena Project:

Es una biblioteca que implementa de forma fácil la persistencia en java. Posee las siguientes características:

- No tiene dependencias, ocupa 25Kb y el API está diseñada para ser lo más sencilla posible.
- Es intrusivo. Para persistir objetos es necesario extender de una clase, de modo que el código depende de Siena. El motivo es que se ha utilizado el patrón ActiveRecord, para hacer el API lo más sencilla posible. Otros frameworks de persistencia optan por hacer que las clases persistentes no tengan ninguna dependencia en el framework, pero en la práctica es absolutamente infrecuente que una aplicación cambie de motor de persistencia.

- Esta limitado. El API se ha diseñado para que todas las consultas puedan ser ejecutadas usando sólo un índice. Todas las consultas se realizan sólo sobre una tabla, no se pueden hacer subconsultas ni JOINS y no se puede utilizar OR en una cláusula WHERE. De este modo se consigue que todas las consultas puedan ser realizadas utilizando sólo un índice y así sean lo más rápidas posible. Esta estrategia es la que viene siendo utilizada por las aplicaciones que necesitan ser altamente escalables. Las bases de datos "en la nube" Big Table de Google, SimpleDB de Amazon, utilizan esta aproximación.
- Extensible. La implementación actual utiliza JDBC y ha sido probada en MySQL. No obstante la funcionalidad principal de Siena se basa en dos interfaces que pueden ser implementadas utilizando otros mecanismos de persistencia. En este artículo me centraré en la implementación actual que transforma las consultas a SQL.

MySql:

Es un motor de base de datos, bajo licencia de software libre, entre sus características a destacar para nuestro objetivo es que posee la posibilidad de configuración de query cache, como así también el tamaño de cache donde de almacenaran las query. Es uno de los motores de base de datos más usados con 6 mil millones de usuarios en todo el mundo.

Simulación de consumo de servicio:

Con el objetivo de simular una carga importante de solicitudes al servicio web, se investigó herramientas de análisis de consumo y funcionamiento de servicios web.

SoapUI:

Es una herramienta de Software Libre gráfica, está basada en Java y sirve para el testeo de Web Services y generación de clientes de Web Services.

Permite testear web services de forma fácil, ver los resultados. Además, permite facilitar el uso de herramientas comunes para la generación de clientes, como Axis.

Trabajando con web services, y sin interface gráfica en la aplicación, esta herramienta nos permite automatizar fácilmente las pruebas funcionales y así asegurar la calidad en nuestros proyectos.

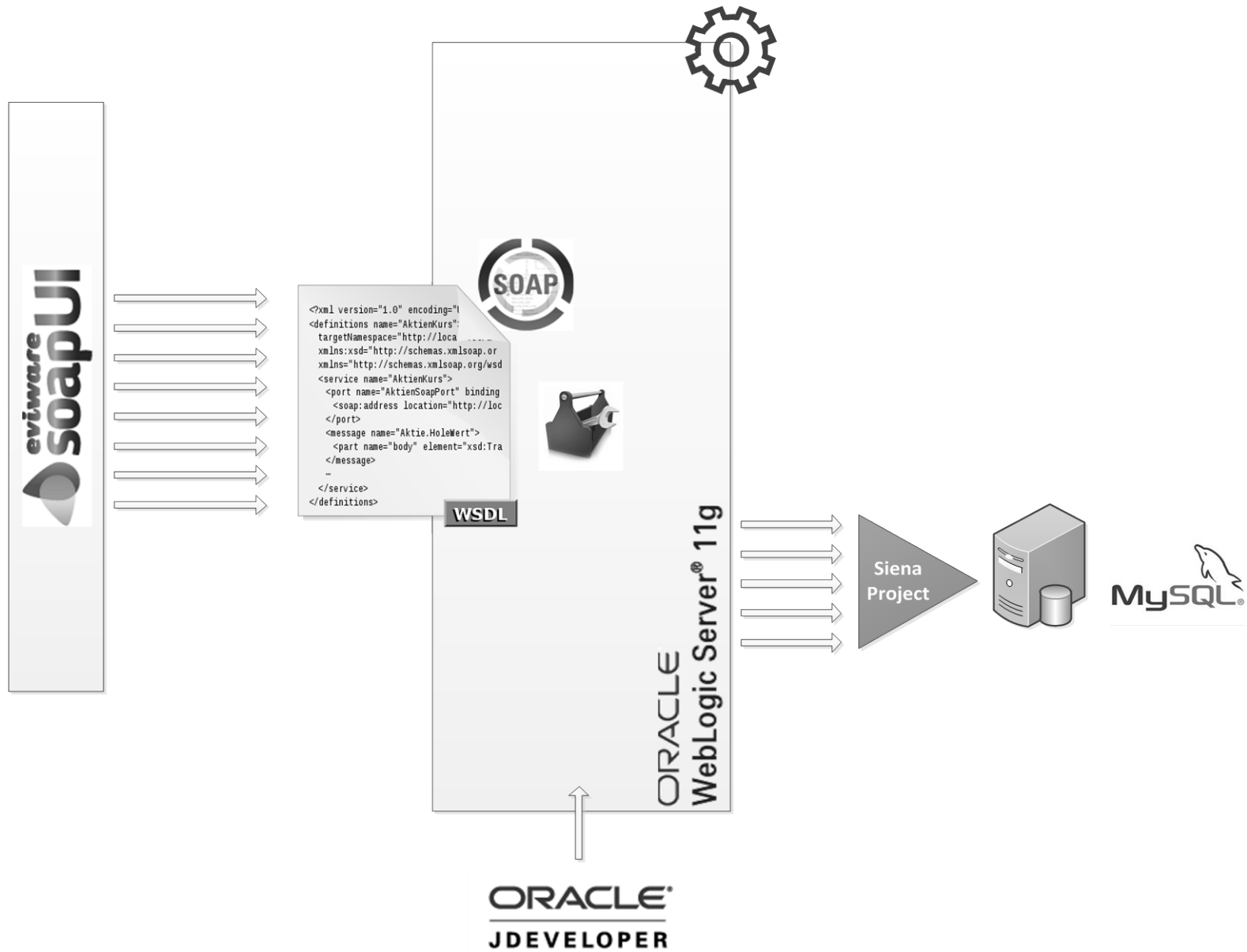
Las pruebas funcionales de los web services podrían usarse para más de un propósito:

- Pruebas unitarias: para validar que cada operación de los servicios funciona como se definió.
- Prueba de aceptación: para validar que el servicio retorna resultados aceptables según los requerimientos.
- Pruebas de proceso: para validar que una sucesión de invocaciones del servicio cumple con el proceso de negocio definido.

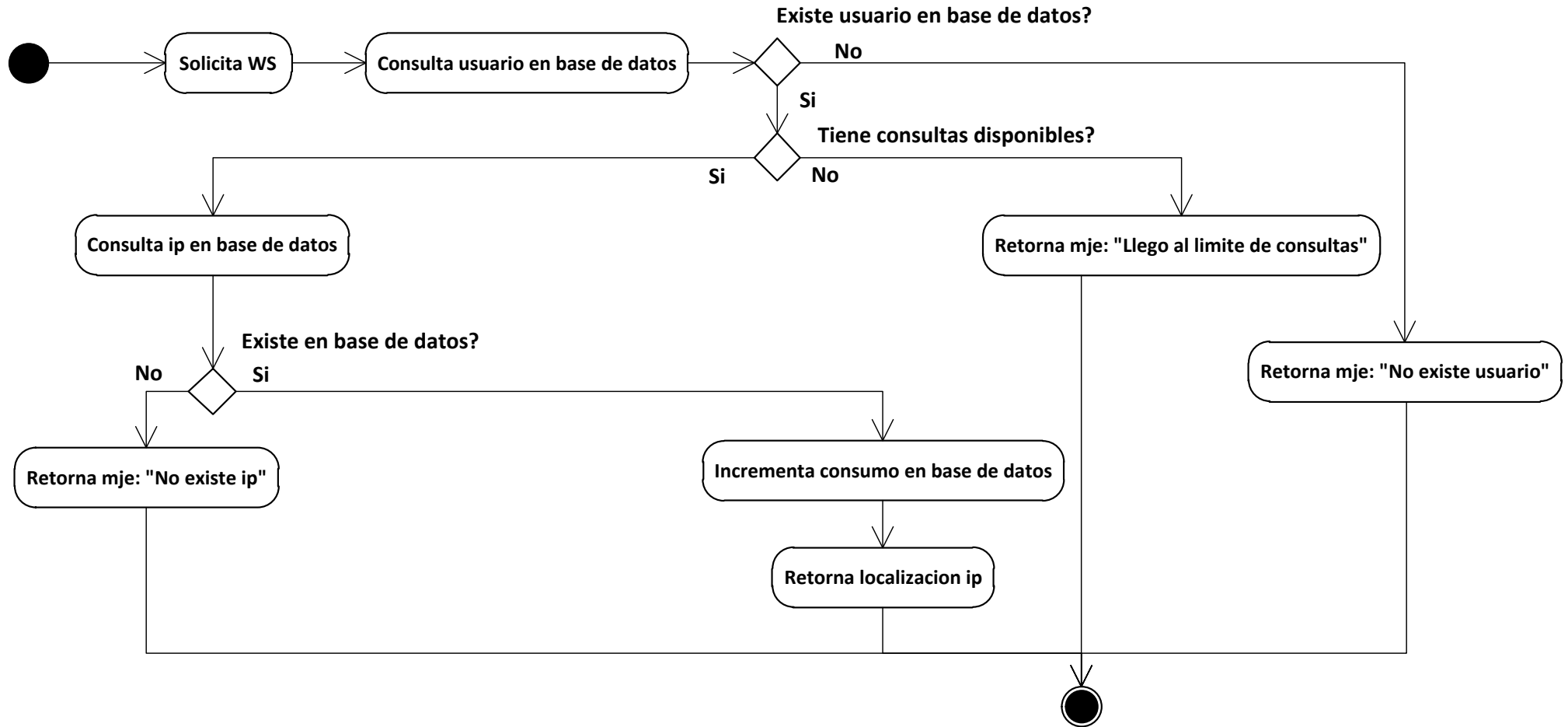
- Pruebas de manejo de datos: para validar el comportamiento con las entradas de datos externos al sistema (bases de datos, otros sistemas, uso de otros web services).
- Pruebas de regresión: para validar el comportamiento post cambios.
- Permite generar un número de threads, configurable, que actúen como usuarios virtuales, consumiendo el servicio en un lapso de tiempo estipulado
- Posibilita establecer casos de test con parámetros de entrada, con los cuales los usuarios virtuales van a enviar solicitudes al web services.
- Ofrece la ventaja de poder, establecer rangos de tiempo de repuestas esperados, como así también validar las repuestas obtenidas.
- Permite observar el resultado del análisis de carga en forma gráfica como así también detalladamente por cada caso de test, en caso de generarse errores posee un log en el cual se puede observar la causa del error.
- Permite generar código de clientes de web services.
- Brinda la posibilidad de verificar los web services si están bien estructurados de acuerdo a la WS-I (Web Services Interoperability).

Es por todo lo expresado anteriormente que se llegó a la siguiente arquitectura:

Arquitectura definitiva:



Diag. de Estados de Web Services (Java) definitivo:



Proceso de Prueba:

Entorno de Prueba:

Servidor:

Hardware:

Procesador i5, RAM: 4 gb, HD: 500 gb

Software:

Sistema operativo: Windows 7 Home Premium

WebLogic 11g + jDeveloper

MySQL 5.5

Apache 2.2.19

PHP 5.3.6

Librería de soap para php utilizada: nusoap.php

Cliente:

Hardware:

Procesador Pentium IV, RAM: 512 mb, HD: 80 gb

Software:

SoapUI

Parámetros del test:

Objetivos principales del test:

- encontrar la mejor combinación de recursos y configuraciones para brindar u soporte a un alto número de solicitudes concurrentes
- poder establecer con que plataforma el rendimiento es mayor

- establecer el número máximo de solicitudes concurrentes a cuales se puede brindar un buen servicio

Para cada uno de los test, se realizara pruebas con 20, 50, 100, 150 y 200 usuarios virtuales que realizan solicitudes a los servicios web durante 60 segundos, sin delay entre solicitudes para hacer más riguroso el test.

Es importante aclarar que cada test consta de seis solicitudes cada una con ips diferente.

De los resultados obtenidos se obtuvo un promedio de solicitudes ejecutadas y tiempos de ejecución.

Datos a tener presente en la realización del test:

Medición del tiempo por parte de SoapUI: Cabe aclarar que el tiempo que muestra SoapUI es el tiempo comprendido desde que se “escribe” la solicitud, es enviada la misma, el tiempo que espera en cola, tiempo de ejecución hasta el momento que “lee” la respuesta.

Threads en Apache 2.2: Esta plataforma tiene por defecto 64 threads, en la modalidad denominada mpm_winnt para el manejo de threads en sistemas operativos Windows, con un máximo de 1920.

Threads en WebLogic: para el modo desarrollo tiene por defecto 15 threads y en modo producción 25 threads. (**Table B-1** http://download.oracle.com/docs/cd/E11035_01/wls100/perform/appb_queues.html) Además ajusta el número límite de threads de acuerdo a la demanda, es por ello que se obvio del testeo la variación de la cantidad de hilos. Para la realización de este proyecto se trabajó con el modo producción.

MySQL: Este motor posee la posibilidad de establecer de acuerdo a un análisis de recursos de hardware, la configuración óptima para los recursos que dispone, teniendo en cuenta el objetivo de uso del motor. Los valores por defecto del número máximo de conexiones concurrentes permitido es de 100, además query cache en cero. Para este proyecto se optó por la configuración de “Server Machine (OLPT)”

SoapUI: A este software se lo configuro con las siguientes características:

- que al finalizar una solicitud cierre la conexión de la misma
- que se cancelen las solicitudes que no finalicen al terminar el periodo de tiempo.
- Con el objetivo de poder realizar un seguimiento visual mientras se ejecuta, se configuro que actualice las estadísticas cuando cambien.

Se dividió el test en 2 categorías:

- **Tests con servidor Apache + PHP:** se generara pruebas individuales, cambiando el número de threads soportados por Apache. Además se fue ajustando el número de conexiones concurrentes permitidas por MySQL como así también el tamaño de la memoria cache del motor de base de datos

- **Test con servidor Weblogic + Java:** teniendo en cuenta que la cantidad de threads se ajusta automáticamente en weblogic, las pruebas no incluirán cambios de threads como en Apache. Se realizarán cambios en el tamaño de memoria de JVM, además de ir cambiando la cantidad de conexiones concurrentes y el tamaño de cache de MySQL.

A continuación se muestra los resultados obtenidos de todos los test realizados teniendo en cuenta los parámetros establecidos anteriormente:

Resultados Tests con servidor Apache + PHP:

id	ThreadsPerChild	max_con ecctions (MySQL)	query_ cache	solicitudes promedio	tiempo promedio	fallos	200 usuarios	tiempo	250 usuarios	tiempo	300 usuarios	tiempo	comentarios
1	50	100	0	154	7	412	144	11	142	12	204	3	connection refused
2	100	100	0	138	8		134	12	134	12	134	11	
3	150	100	0	160	8		174	13	174	13	174	13	
4	200	100	0	172	9		196	12	198	13	190	15	
9	50	100	50	255	5	415	257	8	251	10	257	2	connection refused
10	100	100	50	244	6		249	8	250	10	241	11	
11	150	100	50	253	6		260	8	249	10	232	10	
12	200	100	50	252	6		253	8	267	11	265	12	
17	50	100	100	259	5	339	242	8	257	10	306	5	connection refused
18	100	100	100	249	6		250	8	243	10	250	12	
19	150	100	100	252	6		262	8	260	8	250	12	
20	200	100	100	256	6		265	9	265	9	251	11	
5	50	200	0	159	6	432	156	11	142	12	214	2	connection refused
6	100	200	0	146	8		146	11	134	13	134	11	
7	150	200	0	141	8		135	12	125	10	126	11	
8	200	200	0	143	8		134	12	134	13	134	12	
13	50	200	50	258	5	406	239	8	249	10	300	3	connection refused
14	100	200	50	240	6		234	8	234	11	234	11	
15	150	200	50	235	6		225	9	236	11	226	11	
16	200	200	50	236	6		232	9	227	12	232	11	
21	50	200	100	256	5	366	245	8	250	10	298	4	connection refused
22	100	200	100	247	6		251	8	245	11	235	11	
23	150	200	100	233	6		225	9	225	11	226	11	
24	200	200	100	227	7		217	9	210	12	208	12	
			maximo:	256	9		265	13	267	13	265	15	
			minimo:	138	6		134	8	125	8	126	10	

Resultados con servidor Weblogic + Java:

id	JVM	max_con ecctions (MySQL)	query_ cache	solicitudes promedio	tiempo promedio	fallos	200 usuarios	tiempo	250 usuarios	tiempo	300 usuarios	tiempo	comentarios
1	256- 512 / 128-512	100	0	254	6	0	267	7	268	9	248	11	
2	256- 512 / 128-512	200	0	266	7	0	261	7	263	10	250	11	
3	256- 512 / 128-512	100	50	1178	1	0	1194	2	1185	2	1029	3	
4	256- 512 / 128-512	200	50	1137	1	0	1191	2	1128	2	1118	3	
5	256- 512 / 128-512	100	100	1194	1	0	1163	2	1157	2	1162	3	
6	256- 512 / 128-512	200	100	1191	1	0	1130	2	1152	2	1130	3	
7	512-1024 / 256-512	100	0	299	5	0	299	7	284	8	296	10	
8	512-1024 / 256-512	200	0	254	7	0	256	7	260	10	210	14	
9	512-1024 / 256-512	100	50	1158	1	0	1162	2	1130	2	1101	3	
10	512-1024 / 256-512	200	50	1192	1	0	1259	2	1225	2	1173	3	
11	512-1024 / 256-512	100	100	1209	1	0	1221	2	1162	2	1142	3	
12	512-1024 / 256-512	200	100	1192	3	0	1215	2	1167	2	1146	3	
			maximo:	1209	7		1259	7	1225	10	1173	14	
			minimo:	254	1		256	2	260	2	210	3	

Conclusiones:

Como se mencionó anteriormente el objetivo de este trabajo es lograr poder dar un buen servicio al mayor número posible de usuarios virtuales concurrentes, es por ello que el rango que nos interesa en las tablas anteriores es de 200, 250, y 300 usuarios concurrentes.

Además se expresa la implementación de web services con una capacidad de procesamiento de un interesante número de clientes concurrentes, en una computadora de uso personal, con recursos limitados, sin disponer de un servidor. Teniendo en cuenta lo expuesto se puede mencionar que el presente trabajo ha demostrado la gran ventaja que nos brinda una correcta optimización de la plataforma como así también del motor de base de datos.

A continuación se mencionan, las acciones realizadas y los resultados obtenidos de las mismas:

Apache:

Podemos ver que la iniciar las pruebas partimos con 134 solicitudes promedio con un tiempo promedio de 11 ms para 300 usuarios virtuales. Con las optimizaciones se logra una combinación de configuraciones que permite alcanzar las 265 solicitudes promedio con un tiempo promedio de 12 ms. Se logró duplicar el número de solicitudes casi manteniendo el mismo tiempo promedio entre la “escritura” de la solicitud y “lectura” de la respuesta. La combinación que permitió alcanzar esto es la de: `ThreadsPerChild = 200`; `max_conecctions (MySQL) = 100`; `query_cache_size = 50M`

Es importante destacar que cuando permitimos 50 threads concurrentes en apache y lo probamos con 300 usuarios virtuales, sin importar la cantidad de conexiones permitidas en mysql ni el tamaño del query cache, logramos lo que se conoce como ataque de denegación de servicio. Apache rechaza todas las solicitudes (connection refused).

WebLogic:

Debido a la característica de ajuste automático de conexiones concurrentes, se obvió este parámetro, ya que se ajusta automáticamente a demanda.

En este caso partimos con una base de 248 solicitudes promedio en un tiempo promedio de 11 ms para 300 usuarios concurrentes. Se logró alcanzar un numero promedio de 1173 solicitudes en 3 ms, gracias a la siguiente combinación: `jvm = 512-1024 / 256-512`; `max_conecctions (MySQL) = 200`; `query_cache_size = 50M`

Podemos ver claramente que el web services que se ejecuta sobre weblogic supero al ejecutado en a apache

En el momento de mejorar el funcionamiento del motor de base de datos, el mismo ofrece configuraciones predeterminadas ejecutando `MySQLInstanceConfig.exe`. Esta aplicación analiza los recursos disponibles de hardware (memoria, HD, y CPU), teniendo en cuenta estos valores y de acuerdo al tipo de instancia elegida (Developer Machine, Server Machine, Dedicated MySQL Server Machine), MySQL configura automáticamente los parámetros del mismo. Es importante aclarar que también ofrece la configuración de forma manual.

Podemos observar que no solo basta con optimizar la plataforma de ejecución sino que es altamente necesario optimizar también los parámetros de procesamiento del motor de base de datos.

Bibliografía

Para la realización del proyecto se tendrá en cuenta las presentaciones de la presente especialización:

- Construcción de aplicaciones sobre infraestructuras distribuidas. Profesor: Mariano Cilia
- Ingeniería de Documentos. XML, WSDL. Profesora: Laura Alonso Alemany
- Proceso de desarrollo de Software: Patricio Maller

Como así también mediante internet se realizaran consultas a las siguientes bibliografías on-line:

Web Services:

Título: Introducción a los Web Services en PHP

Link: <http://www.desarrolloweb.com/articulos/1852.php>

Fecha: 03/03/2005

Autor: Orlando Fabián Brea

Link datos del autor: <http://www.desarrolloweb.com/contacta/133.html>

Título: Web Services: XML-RPC, SOAP, sobre PHP, Perl, y otros conceptos

Link: http://www.programacion.com/articulo/web_services:_xml-rpc-_soap-_sobre_php-_perl-_y_otros_conceptos_211/4

Autor: Martin Valdivia

Escalabilidad:

Título: Evaluación del desempeño en aplicaciones web

Link: <http://www.sabetodo.com/contenidos/EkpEApkIIACvuLULwX.php>

Fecha: 01/2008

Autor: Yoemny González Almaquer

Link datos del autor: <http://www.desarrolloweb.com/contacta/133.html>

Conexiones persistentes:

Título: Conexiones persistentes MySQL Para que ? Que Son ?

Link: http://www.vbhispano.com/foros/f35/conexiones_persistentes_mysql_para_que_que_son-1651/

Fecha: 01/10/2004

Autor: Danny Bembibre Gude

Link datos del autor: <http://www.vbhispano.com/foros/u919.html>

Título: Siena: persistencia en Java sencilla y escalable

Links:

- <http://es.debugmodeon.com/articulo/siena-persistencia-en-java-sencilla-y-escalable>
- <http://code.google.com/p/siena/>
- <http://www.sienaproject.com/index.html>

e-mail autor: gimenete@gmail.com

WebLogic:

Título: Oracle WebLogic 11gR1 on Windows 7

Link: <http://oracledwbi.wordpress.com/2010/04/03/oracle-weblogic-11gr1-on-windows-7/>

Fecha: 03/04/2010

Autor: Suds Kumar (Hemel Hempstead, United Kingdom)

Link datos del autor: <http://oracledwbi.wordpress.com/about/>

Título: BEA WebLogic Server® 9.0 Documentation

Link: http://docs.oracle.com/cd/E13222_01/wls/docs90/index.html

Fecha: 2011

Autor: BEA Systems

Link datos del autor: <http://www.oracle.com/us/corporate/acquisitions/bea/index.html>

Título: Oracle WebLogic Server: Una Base Sólida para la Arquitectura Orientada a Servicios

Link: <http://www.oracle.com/technetwork/es/documentation/317511-esa.pdf>

Fecha: 2008

Autor: Oracle

Link datos del autor: <http://www.oracle.com/lad/corporate/index.html?ssSourceSiteId=otnes>

Configuración Weblogic

Título: A Manual Configuration of WebLogic Server

Link:

http://download.oracle.com/docs/cd/E12058_01/doc/doc.1014/e12026/app_orminweblogic.htm

Fecha: 2008,2009

Autor: Oracle

Link datos del autor: <http://www.oracle.com/lad/corporate/index.html?ssSourceSiteId=otnes>

Título: Automatizar Java Performance Tuning

Link: <http://www.oracle.com/technetwork/es/articles/adf/brahms-tuning-096558-esa.html>

Fecha: 09/2008

Autor: Carl Brahms

Título: Tuning Garbage Collection with the 5.0 Java[tm] Virtual Machine

Link: <http://www.oracle.com/technetwork/java/gc-tuning-5-138395.html>

Fecha: 2003

Autor: Sun Microsystems, Inc.

Link datos del autor: http://en.wikipedia.org/wiki/Sun_Microsystems

Título: Top Tuning Recommendations for WebLogic Server

Link: http://download.oracle.com/docs/cd/E11035_01/wls100/perform/topten.html

Fecha: 2011

Autor: BEA Systems

Link datos del autor: <http://www.oracle.com/us/corporate/acquisitions/bea/index.html>

Título: Writing a WebLogic Startup Script

Link: <http://www.dummies.com/how-to/content/writing-a-weblogic-startup-script.html>

Autor: Jeff Heaton

Link datos del autor: <http://www.jeffheaton.com/>

Título: Configuring WebLogic Server Domain/Machine/Server instances with the JDeveloper 11g ADF installer

Link: <http://chrismuir.sys-con.com/node/952411/>

Fecha: 04/01/2009

Autor: Chris Muir

Link datos del autor: <http://chrismuir.sys-con.com/>

Título: Instalar Weblogic 10.3.3.0 y Desplegar en él una aplicación ADF 11g

Link: <http://fetishcode.wordpress.com/2010/10/17/instalar-weblogic-10-3-3-0-y-desplegar-en-el-una-aplicacion-adf-11g/>

Fecha: 17/10/2010

Autor: Herman Alonso

Título: JDBC Connection Pool Performance Tuning

Link: <http://weblogicserveradministration.blogspot.com/2010/10/jdbc-connection-pool-performance-tuning.html>

Fecha: 23/10/2010

Autor: Mukesh Negi

Link datos del autor: <http://www.blogger.com/profile/00868823116792559637>

Título: BEA WebLogic: Guía de Administración

Link: http://www.programacion.com/articulo/bea_weblogic:_guia_de_administracion_144/8

Fecha: 23/10/2010

Autor: j calvo

Título: WebLogic 11g Overload Protection in the Cloud

Link: <http://www.munzandmore.com/2010/09/weblogic-11g-overload-protection-in-the-cloud>

Fecha: 14/09/2010

Autor: Dr. Frank Munz

Link datos del autor: <http://www.munzandmore.com/about>

MySQL

Título: MySQL 5.5 Reference Manual

Link: <http://dev.mysql.com/doc/refman/5.5/en/>

Fecha: 09/11/2011

Autor: MySQL

Link datos del autor: <http://dev.mysql.com/doc/index-about.html>

Título: Optimizando el servidor Mysql

Link: <http://www.rinconinformatico.net/optimizando-el-servidor-mysql>

Fecha: 09/01/2010

Autor: Epsilon

Link datos del autor: <http://www.rinconinformatico.net/acerca>

Título: Acelerar el rendimiento de MySQL con la cache

Link: <http://www.codigomaestro.com/mysql/acelerar-el-rendimiento-de-mysql-con-la-cache/>

Fecha: 14/06/2008

Autor: MySQL

Link datos del autor: <http://dev.mysql.com/doc/index-about.html>

Título: MySQL: Optimizar el rendimiento de lectura

Link: <http://www.guatemwireless.org/tecnologia/bases-de-datos/mysql-optimizar-el-rendimiento-de-lectura/>

Fecha: 28/11/2008

Autor: fher98

Link datos del autor: <http://www.guatemwireless.org/>

Título: How to Optimize MySQL server

Link: <http://techgurulive.com/2009/07/14/how-to-optimize-mysql-server/>

Fecha: 14/07/2009

SoapUI:

Título: Pruebas funcionales de servicios web con soapUI

Link: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=pruebas-funcionales-soapui>

Fecha: 28/12/2009

Autor: Iván García Puebla

Link datos del autor: <http://es.linkedin.com/in/ivangarciapuebla>

Título: Creating and Running LoadTests

Link: <http://www.soapui.org/Load-Testing/loadtest-execution.html>

Fecha: 15/10/2009

Autor: Eviware

Link datos del autor: <http://www.eviware.com/>

Título: Simulating different types of Load

Link: <http://www.soapui.org/Load-Testing/simulating-different-types-of-load.html>

Fecha: 15/10/2009

Autor: Eviware

Link datos del autor: <http://www.eviware.com/>

Título: LoadTest Window

Link: <http://www.soapui.org/Load-Testing/load-test-window.html>

Fecha: 13/10/2009

Autor: Eviware

Link datos del autor: <http://www.eviware.com/>

Título: Strategies

Link: <http://www.soapui.org/Load-Testing/strategies.html>

Fecha: 13/10/2009

Autor: Eviware

Link datos del autor: <http://www.eviware.com/>

Apache:

Título: Versión 2.2 de la documentación del Servidor de HTTP Apache

Link: <http://httpd.apache.org/docs/2.2/>

Fecha: 2011

Autor: The Apache Software Foundation

Link datos del autor: <http://www.apache.org/foundation/>

Título: Tunear Apache con cabeza

Link: <http://miguelcarmona.name/notas-y-recortes/tunear-apache-con-cabeza>

Fecha: 14/02/2011

Autor: Miguel Carmona

Link datos del autor: <http://miguelcarmona.name/blog>

Título: Optimizar apache

Link: <http://www.codenb.com/optimizar-apache-16/>

Fecha: 29/08/2007

Autor: The Apache Software Foundation

Link datos del autor: <http://www.apache.org/foundation/>

Título: Apache Performance Tuning

Link: <http://httpd.apache.org/docs/2.2/misc/perf-tuning.html>

Fecha: 2011

Autor: The Apache Software Foundation

Link datos del autor: <http://www.apache.org/foundation/>

Título: Apache MPM winnt

Link: http://httpd.apache.org/docs/2.2/mod/mpm_winnt.html

Fecha: 2011

Autor: The Apache Software Foundation

Link datos del autor: <http://www.apache.org/foundation/>

Título: Configuring Apache for Maximum Performance

Link: <http://bobcares.com/blog/?p=71>

Fecha: 08/01/2006

Autor: Vishnu Ram

Link datos del autor: <http://www.apache.org/foundation/>

Título: Optimizing MySQL and Apache for Low Memory Usage, Part 1

Link: <http://emergent.urbanpug.com/?p=60>

Fecha: 05/10/2006

PHP:

Título: Manual de PHP

Link: <http://www.php.net/manual/es/>

Fecha: 2011

Autor: the PHP Documentation Group

Link datos del autor: <http://www.php.net>

Título: A HOWTO on Optimizing PHP

Link: <http://phplens.com/lens/php-book/optimizing-debugging-php.php>

Fecha: 30/10/2009

Autor: phplens.com

Link datos del autor: phplens.com