Título: Un Medidor de Rendimiento de Servidores de Bases de Datos Relacionales

Director: Antonio César Gómez Lora

Alumno: Jose Antonio Jamilena Daza

# Introducción y Motivación

La mayoría de los sistemas gestores de bases de datos actuales disponen de herramientas propias o de terceros destinadas a medir el estado de salud y el rendimiento del propio sistema. Estos sistemas con capaces de proporcionar un extenso y valioso conjunto de información para administradores y auditores. Adicionalmente existen plataformas software destinadas a construir medidores de calidad de servicio, aunque, hoy por hoy, con una orientación muy específica hacia redes y servicios web.

Con este rico abanico de herramientas disponibles, y la necesidad de medir la calidad de los servicios prestados, es cada vez más común encontrar servidores dentro de una empresa u organización destinados a analizar la disponibilidad y rendimiento de los servicios ofertados (entendiendo el acceso a una base de datos como uno de estos servicios). Desgraciadamente estos servidores suelen estar situados estratégicamente en zonas clave de las intranets o de las infraestructuras de comunicación. Cuando los servicios están abiertos al exterior sólo se analiza el entorno próximo del servidor y nunca el entorno del cliente.

Para un usuario simple que accede desde su equipo a un servidor de bases de datos el rendimiento viene marcado por el rendimiento de todos y cada uno de los equipos, dispositivos y enlaces involucrados en el procesamiento y el transporte de la información. En una sesión común esto puede significar la participación de decenas, cientos e incluso miles de elementos hardware y software participantes, la mayoría de los cuales son transparentes al usuario. Y en caso de ser visible alguno de estos participantes generalmente no se tiene la capacidad ni los permisos suficientes para analizar su rendimiento individual.

Se nos plantea pues la necesidad de que un usuario pudiera ejecutar un sencillo programa que le permitiera medir el rendimiento de determinadas tareas en un servidor de bases de datos. Este sistema analizará el rendimiento de estas tareas de forma periódica siguiendo una planificación dada para cumplir su objetivo principal, que es detectar la aparición de posibles problemas. Aunque el diagnóstico del problema queda fuera de los objetivos de esta aplicación, sí es cierto que se añadirá la capacidad de clasificar las diferentes tareas según características particulares para agrupar sus resultados y así asistir al usuario en la interpretación de los datos y estadísticos para que se pueda realizar un diagnóstico. Este programa sencillo tiene un doble destino: primero el ser utilizado por usuarios aislados; y segundo el poder ser instalado en múltiples equipos y permitir recolectar posteriormente sus datos de forma sencilla (preferiblemente en forma de un archivo simple).

# Objetivos

Los objetivos que este proyecto plantea son los que siguen:

* Elaborar un software que ejecute sobre sistemas gestores de bases de datos relacionales, bajo una planificación dada, unos scripts de prueba, para a partir de los tiempos de ejecución generar indicativos sobre su comportamiento.
* Construir una interfaz gráfica que sea capaz de recoger los datos de la herramienta descrita anteriormente y generar unas estadísticas que sean mostradas de forma visual para que el usuario las interprete. Dichas estadísticas se agruparan según los diferentes tipos de scripts lanzados.
* Generar un conjunto base de scripts que puedan ser clasificados en conjuntos no disjuntos de categorías o propiedades. Con esto se consiguen estadísticas según las posibles características evaluar.

# Metodologías y fases de trabajo

En el desarrollo del proyecto se empleará el paradigma de programación orientado a objetos y para modelar el software el Lenguaje Unificado de Modelado o UML, iniciales que corresponden al acrónimo inglés *Unified Modeling Language*.

El modelo de desarrollo será el Desarrollo Iterativo Incremental ].

En cuanto a las fases en las que se dividirá el desarrollo software éstas serán:

* Una prospección inicial del desarrollo en el que barajaremos posibles lenguajes para desarrollar el proyecto, tecnologías a emplear, etc. Así también como la obtención de los objetivos que ha de cumplir el desarrollo software que formará el proyecto. Tras la cual habremos obtenido el análisis de requisitos.
* Se realizará la especificación de los distintos componentes que formarán parte del desarrollo software.
* Una vez que tenemos los componentes que debemos implementar pasaremos a una fase de diseño de la arquitectura de la solución. En esta fase del desarrollo se realizará el diseño de los distintos componentes del sistema software, así como el diseño de la jerarquía de clases de los distintos componentes del desarrollo software.
* Posteriormente pasaremos a la implementación del diseño. En esta fase se puede llegar a tener que hacer cambios en el diseño si la naturaleza del desarrollo así lo exige.
* Finalmente se realizarán las pruebas oportunas para verificar el correcto comportamiento del sistema.

Durante todo el proceso se seguirá el Desarrollo Iterativo Incremental ], se irán obteniendo prototipos de desarrollo sobre los que se irán haciendo pruebas de funcionamiento y de completitud de requisitos.

Durante todas estas fases se llevará un amplio proceso de documentación, tanto para la escritura de la memoria final como para guardar el conocimiento adquirido durante el desarrollo.

# Herramientas de desarrollo

El desarrollo se hará sobre el lenguaje de programación Java, concretamente haciendo uso de JDK 1.6.0\_14-b08 de Sun Microsystems, la última versión de este leguaje de programación orientado a objetos ].

El interfaz elegido para el desarrollo será Netbeans 6.5, un entorno integrado de desarrollo especializado en Java ].

Todo el proyecto se guardará con el sistema de control de versiones Subversion. Este se encarga de mantener históricos de ficheros durante los desarrollos software ]. Subversión se instala como un software servidor al que se accede mediante un cliente. El cliente elegido es TortoiseSVN, un cliente que se integra tanto en la shell del sistema operativo como en el propio Netbeans ].

Se realizará el diseño de la aplicación sobre Enterprise Architect y sobre un complemento específico de Netbeans para estos menesteres. Enterprise Architect es una herramienta CASE, Computer Aided Software Engineering o Ingeniería de Software Asistida por Ordenador, y en el diseño de este proyecto en cuestión se guardarán los datos de diseño en un sistema de bases de datos MySQL ]]. En cuanto al componente de Netbeans lo almacenará con estructura de archivos de proyecto de Netbeans.

El proyecto tendrá integrada su propia base de datos de estadísticas. Para dicho cometido se empleará SQLite, un sistema gestor de bases de datos empotrado, rápido y de poco peso en memoria ]. Para comunicarse con Java se empleará la librería SQLiteJDBC ]. Para facilitar el diseño en dicho sistema gestor de bases de datos se empleará la herramienta SQLite Management Studio, una herramienta para gestión gráfica de ficheros SQLite ].

Para velar por la integridad y conocer inequívocamente los fallos que pueden acaecer, tanto en la ejecución normal como en las sesiones de pruebas, se empleará Apache Log4J. Apache Log4J es un sistema de diario de ejecución para aplicaciones Java ]. Dicho sistema nos da la capacidad de generar ficheros de diario de ejecución con las trazas de la misma e incluso la capacidad de que llegado el caso, si el sistema al que estamos evaluando se llegara a colapsar, enviar un email advirtiendo este hecho.

Para el componente encargado de mostrar las estadísticas de las pruebas se empleara un interfaz gráfico Java implementado con Swing. Como sistema visualizador de datos y estadísticas emplearemos la biblioteca JFreeChart. JFreeChart es un sistema de generación de graficas para el entorno grafico Swing de Java ].

Las conexiones entre el proyecto y los diferentes gestores de bases de datos se harán mediante la tecnología JDBC (Java Data Base Connectivity). JDBC es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Toda la planificación temporal del proyecto será llevada acabo con la ayuda de OpenProj ]. Y la documentación se generará a partir de Enterprise Architect y se editará el texto con OpenOffice ] y con Microsoft Visio ] los diagramas que fueran necesarios.

# Bibliografía

[1] Wikipedia. Iterative and incremental development. [En línea] http://en.wikipedia.org/wiki/Iterative\_development.

[] Sun Microsystems. Java™ Platform, Standard Edition 6. [En línea] http://java.sun.com/javase/6/docs/api/.

[3] Sun Microsystem. Netbeans. [En línea] http://www.netbeans.org/.

[] Visualsvn. Visualsvn. [En línea] http://www.visualsvn.com/server/.

[5] TortoiseSVN. TortoiseSVN The coolest Interface to (Sub)Version Control. [En línea] http://tortoisesvn.net/.

[6] Sparx Systems. Enterprise Architect - UML for Business, Software and Systems. [En línea] http://www.sparxsystems.com.au/.

[7] Sun Microsystems. MySQL - The world's most popular open source database. [En línea] http://www.mysql.com/.

[8] SQLite. SQLite - Small. Fast. Reliable. [En línea] http://www.sqlite.org/.

[] SQLiteJDBC. SQLiteJDBC. [En línea] www.zentus.com/sqlitejdbc/.

[10] Osenxpsuite. SQLite2009 Pro Enterprise Manager. [En línea] http://link.osenxpsuite.net/?uid=homepage&id=sqlite2009pro.zip.

[11] Apache Software Foundation. Logging Services. [En línea] 1999-2007. http://logging.apache.org/log4j/1.2/index.html.

[] JFree.org. JFreeChart. [En línea] http://www.jfree.org/jfreechart/.

[13] Serena. OpenProj is a free, open source project management solution. [En línea] http://openproj.org/openproj.

[14] Sun Microsystems. OpenOffice - the free and open productivity suite. [En línea] http://www.openoffice.org/.

[] Microsoft Corporation. Microsoft Office Visio 2007. [En línea] http://office.microsoft.com/es-es/visio/FX100487863082.aspx.

[16] Dr. Bert Scalzo, Claudia Fernandez, Donald K. Burleson, Mike Ault, Kevin Kline. Database Benchmarking: Practical Methods for Oracle & SQL Server. s.l. : Rampant TechPress, 2007. ISBN 0977671534, 9780977671533.

[17] Darwin, Ian F. Java cookbook. s.l. : O'Reilly, 2001. ISBN 0596001703, 9780596001704.