

Universidad de Ciencias Informáticas

Facultad 10



Sistema de réplica para la Intranet Corporativa y el Sitio en Internet de PDVSA

Trabajo de diploma en opción al título Ingeniero en Ciencias Informáticas

Autores:

Kenny Rodríguez García
Eddy Ernesto Baños Fernández

Tutores:

Ing. César González Hernández
Ing. Omar Rey Lazarte

Ciudad de La Habana
2009

"Los locos abren los caminos que más tarde recorren los sabios"

Carlo Dossi (1849-1910)

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de Julio del año 2009.

Kenny Rodríguez García

Firma del Autor

Eddy Ernesto Baños Fernández

Firma del Autor

César González Hernández

Firma del Tutor

Omar Rey Lazarte

Firma del Tutor

Agradecimientos

Resumen

Entre las aplicaciones informáticas desarrolladas en la empresa PDVSA utilizando tecnologías libres se encuentran la Intranet Corporativa y el sitio en Internet de PDVSA ambos basados en diferentes tecnologías. Ambos portales presentan problemas de rendimiento, los tiempos de respuesta son altos y el número de usuarios continúa incrementándose. Estas y otras razones han motivado el desarrollo de nuevos sitios que eliminen estas problemáticas.

Estos nuevos sitios en su capa de acceso a datos necesitan una tecnología capaz de satisfacer las necesidades actuales, garantizando con su implantación, la centralización de los recursos de la información, la disminución del costo de mantenimiento, alta tolerancia a fallos y un alto nivel de accesibilidad para los usuarios finales.

El presente Trabajo de Diploma propone diseñar e implantar una solución capaz de garantizar una alta disponibilidad y un elevado rendimiento en los servidores de bases de datos de la Intranet y el Portal en Internet de PDVSA.

Índice

INTRODUCCIÓN.....1

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA6

1.1 Introducción 6

1.2 Base de datos..... 6

1.3 Base de datos distribuida..... 6

 1.3.1 Fragmentación de datos 10

 1.3.2 Tipos de fragmentación 11

1.4 Replicación de datos 11

 1.4.1 Replicación (Asimétrica) básica 13

 1.4.2 Replicación (Simétrica) Avanzada:..... 16

1.5 Conflictos de replicación..... 22

 1.5.1 Métodos de resolución de conflictos..... 22

1.6 Tendencias y tecnologías actuales 24

 1.6.1 Software Libre 24

 1.6.2 Sistema gestor de base de datos 25

 1.6.3 Debian GNU/Linux..... 27

1.7 Propuesta de desarrollo..... 28

1.8 Conclusiones Parciales 28

CAPITULO 2 DISEÑO DE LA SOLUCIÓN PROPUESTA..... 29

2.1 Introducción 29

2.2 Flujo informativo y la gestión de contenidos en la Intranet Corporativa y PDVSA.com. 29

2.3 Diseño de la distribución..... 31

2.4 Selección de la herramienta de replicación 35
 2.4.1 Cybercluster 38

2.5 Conclusiones parciales 41

CAPÍTULO 3 IMPLANTACIÓN Y PRUEBA DE LA SOLUCIÓN PROPUESTA..... 42

3.1 Introducción 42

3.2 Implantación..... 42
 3.2.1 Instalación y configuración de Cybercluster 42
 3.2.2 Configuración de los servicios mediante un ejemplo de composición del sistema..... 45

3.3 Monitoreo 60

3.4 Pruebas..... 61
 3.4.1 Pruebas de integridad de datos y BD..... 61
 3.4.2 Pruebas de desempeño 61

3.5 Herramientas..... 66

3.6 Entorno de prueba..... 67
 3.6.1 Criterios de aceptación 68
 3.6.2 Ejecución de pruebas de carga..... 69
 3.6.3 Ejecución de pruebas de estrés..... 76
 3.6.4 Ejecución de pruebas de volumen..... 80

3.7 Conclusiones parciales 81

CONCLUSIONES GENERALES..... 82

RECOMENDACIONES..... 83

Introducción

En diciembre de 2002 Petróleos de Venezuela S.A. [PDVSA¹] sufre un sabotaje informático, ejecutado por la empresa INTENSA, encargada de darle soporte tecnológico, que en consecuencia produce un paro petrolero nacional.

Debido al sabotaje informático que hizo tambalear momentáneamente la corporación, mucha de la información que se maneja en la empresa no estuvo disponible, en especial la información oficial interna que se publicaba en la intranet de la corporación. Para resolver este problema en el año 2003 se creó un portal básico que estaba muy lejos de satisfacer las necesidades comunicacionales de la corporación así como aprovechar de manera efectiva y eficiente las características y funcionalidades que ofrecen los ambientes Web como medios de comunicación, estratégicos y masivos.

En el marco de los acuerdos de cooperación tecnológica entre Cuba y Venezuela, firmados en el año 2005, donde ambas partes manifiestan el deseo de desarrollar soluciones tecnológicas, formar al personal, optimizar la gerencia del conocimiento e implantar sistemas de comunicación digital, surge el Proyecto “Misión Portal”, donde la Universidad de las Ciencias Informáticas asume el desarrollo del mismo.

El proyecto “Misión Portal” está concebido en fases, producto del momento histórico que se vivía en Venezuela y a las limitaciones que tenía PDVSA para publicar sus noticias en Internet en la primera fase se implementó el portal pdvsa.com. Luego el desarrollo continuó con la segunda fase “Intranet Corporativa”.

Como resultado de las etapas anteriores PDVSA cuenta con el portal pdvsa.com y la Intranet Corporativa desarrollados en distintas tecnologías. Esta diferencia aumenta el costo de mantenimiento, provoca dispersión de los datos y duplicación del esfuerzo requerido para publicar un mismo contenido en ambos medios.

¹ Petróleos de Venezuela Sociedad Anónima

En la actualidad y a raíz de un conjunto de inconformidades que han surgido debido a la coexistencia de manejadores de contenidos diferentes para cada aplicación, se plantea una nueva estrategia de desarrollo que propone implementar en la presente fase, un ambiente único para la gestión de los contenidos noticiosos que se publican en ambos entornos, para luego continuar en etapas posteriores con la personalización e integración de nuevos servicios a la Intranet Corporativa, dirigidos a potenciar la comunicación y la gestión del conocimiento empresarial.

La Intranet Corporativa y el Sitio en Internet de PDVSA que actualmente se encuentran en funcionamiento contienen un gran volumen de información y cerca de 80 000 usuarios potenciales que pueden acceder y modificar la misma. El número de usuarios continúa incrementándose, cada una de estas aplicaciones utiliza un solo nodo de base de datos (BD), esto ocasiona problemas de **rendimiento**, aumentando los tiempos de respuesta y en ocasiones fallas en los servicios por lo que la información deja de estar **disponible**. Algunos de estos problemas están dados porque los sistemas manejadores de contenido (CMS² siglas en inglés) sobre los que están implementados estos sitios acceden constantemente a sus **bases de datos**. Se cargan excesivamente por el cúmulo de consultas, afectando constantemente el rendimiento de la aplicación, por lo que los tiempos de respuesta tienden a ser más altos. En ocasiones esta situación puede dañar seriamente la aplicación y en caso de ocurrir algún fallo crítico los datos almacenados podrían perderse. Todo lo anteriormente expuesto se identifica como **situación problemática**.

En el presente, con el crecimiento del número de usuarios se evidencia la pérdida de disponibilidad y un bajo rendimiento en los servidores de bases de datos de cada uno de los sitios, por lo que se hace **necesario** que las tecnologías a aplicar en los servidores de bases de datos de la nueva solución satisfagan las necesidades funcionales de la empresa.

Tomando en cuenta la situación actual, surge el siguiente **problema científico**: ¿Cómo garantizar la disponibilidad y el rendimiento de los servidores de bases de datos para la gestión centralizada de los contenidos que se publican en la Intranet y el portal en Internet de PDVSA?

El presente trabajo se propone dar solución al problema existente, permitiendo la actualización de la información íntegra y eficientemente entre los servidores de bases de datos que conforman el sistema.

² Content Management System

Se define como **objeto de estudio** el proceso de réplica para bases de datos distribuidas.

Delimitando así el **campo de acción**, siendo este, el proceso de réplica multimaestro para bases de datos distribuidas en PostgreSQL para la gestión centralizada de los contenidos que se publican en la Intranet y el portal en internet de PDVSA.

El **objetivo general** de la tesis es: Diseñar e implantar una solución capaz de garantizar la disponibilidad y el rendimiento de los servidores bases de datos para la gestión centralizada de los contenidos que se publican en la Intranet y el portal en Internet de PDVSA.

De él se derivan los siguientes **objetivos específicos**:

- Analizar los entornos de réplica existentes.
- Seleccionar una herramienta que permita la replicación datos.
- Diseñar una solución que permita la configuración, gestión y mantenimiento de una política de replicación de datos entre los servidores de bases de datos.
- Implementar y evaluar la Factibilidad desde el punto de vista tecnológico del Sistema mediante la realización de pruebas de carga y estrés a los servidores.

Para dar cumplimiento al objetivo general se han planteado las siguientes **tareas**:

- ✓ Realizar una búsqueda bibliográfica sobre el proceso de réplica para bases de datos distribuidas, así como de los sistemas informáticos existentes.
- ✓ Estudiar el proceso de gestión de los contenidos que se publican en la Intranet Corporativa y el portal en Internet de PDVSA.
- ✓ Estudiar las últimas tendencias y tecnologías que a nivel mundial se utilizan para construir una solución como la que se pretende aplicar.
- ✓ Seleccionar las tecnologías y herramientas a utilizar para el diseño y desarrollo de la solución.

- ✓ Desarrollar una solución que garantice la fiabilidad, rendimiento, disponibilidad y seguridad del flujo informativo en la capa de persistencia de datos mediante una política de réplica de datos en la Intranet Corporativa y el Sitio en Internet de PDVSA.

Los métodos teóricos para dar cumplimiento a estas tareas son **Analítico-Sintético** centrándose en el análisis de documentos, teorías y otros materiales como manuales que permitan elaborar conclusiones.

Se utilizarán métodos **Empíricos** como técnicas para la recopilación de la información del negocio.

Se utilizarán además el método de la **Modelación**.

Método de la modelación

Es justamente el método mediante el cual se crean abstracciones con vistas a explicar la realidad. El modelo como sustituto del objeto de investigación. En el modelo se revela la unidad de los objetivos y lo subjetivo.

La modelación es el método que opera en forma práctica o teórica con un objeto, no en forma directa, sino utilizando cierto sistema intermedio, auxiliar, natural o artificial.

Método Sintético

Es un proceso mediante el cual se relacionan hechos aparentemente aislados y se formula una teoría que unifica los diversos elementos. Consiste en la reunión racional de varios elementos dispersos en una nueva totalidad, este se presenta más en el planteamiento de la hipótesis. El investigador sintetiza las superaciones en la imaginación para establecer una explicación tentativa que someterá a prueba.

Método Analítico

Se distinguen los elementos de un fenómeno y se procede a revisar ordenadamente cada uno de ellos por separado. La física, la química y la biología utilizan este método; a partir de la experimentación y el análisis de gran número de casos se establecen leyes universales. Consiste en la extracción de las partes de un todo, con el objeto de estudiarlas y examinarlas por separado, para ver, por ejemplo las relaciones entre las mismas.

Estas operaciones no existen independientes una de la otra; el análisis de un objeto se realiza a partir de la relación que existe entre los elementos que conforman dicho objeto como un todo; y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis.

El presente trabajo se encuentra dividido en **tres capítulos**.

En el **primero** se tratan aquellos temas que constituyen la fundamentación teórica de la investigación a realizar. Incluye un estado del arte de las tecnologías existentes que de una forma u otra están relacionados con el tema que se aborda, profundizando en ellos.

En el **segundo** capítulo se describe a profundidad el objeto de estudio, así como la modelación de los procesos del negocio y se brinda una descripción de la solución propuesta, definiéndose los requisitos que debe cumplir la misma.

Por último, el **tercer** capítulo constituye la implantación y prueba de la solución propuesta.

Capítulo 1 Fundamentación Teórica

1.1 Introducción

En el presente capítulo se abordará acerca de las tendencias y tecnologías asociadas al dominio del problema. Las bases de datos distribuidas en el contexto actual, así como la descripción detallada de los conceptos de replicación y fragmentación de datos.

Además se introducen las tendencias y tecnologías en las que se apoyará la propuesta de solución.

1.2 Base de datos

Una **Base de Datos** (BD) es una colección de información, de preferencia relacionados y organizada, almacenada con carácter más o menos permanente en la computadora. O sea, que una BD puede considerarse una colección de datos variables en el tiempo. Una base de datos consta de los archivos físicos a crear y modificar en un equipo cuando se instala un software de bases de datos. Por otro lado, un modelo de base de datos es más un concepto que un objeto físico y se utiliza para crear las tablas en su BD. (1)

Por definición, una base de datos es un objeto estructurado. Puede ser un montón de papeles, pero actualmente es más probable encontrarlas en sistemas informáticos. Esa estructura se compone de objetos de datos y metadatos, los metadatos no son más que la estructura. Los datos en una BD almacenan la información descriptiva. Metadatos describe la estructura aplicada por la BD a los datos. Se aplica la estructura y la organización de los datos brutos. (2)

1.3 Base de datos distribuida

Una **Base de Datos Distribuida** (BDD³) es un conjunto de múltiples bases de datos lógicamente relacionadas las cuales se encuentran distribuidas entre diferentes sitios interconectados por una red de comunicaciones, los cuales tienen la capacidad de procesamiento autónomo lo cual indica que puede

³ Base de Datos Distribuida

realizar operaciones locales o distribuidas. La diferencia principal entre los sistemas de base de datos centralizados y distribuidos es que, en los primeros, los datos residen en una sola localidad, mientras que, en los últimos, se encuentran en varias localidades.

Un Sistema de Bases de Datos Distribuida (SBDD⁴) es un sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones de tal forma que, un usuario en cualquier sitio puede acceder a los datos en cualquier parte de la red exactamente como si los datos estuvieran en un mismo servidor de bases de datos. Los procesadores de un sistema distribuido pueden variar en cuanto su tamaño y función. Pueden incluir microcomputadores pequeños, estaciones de trabajo y sistemas de computadores grandes de aplicación general. No comparten la memoria principal ni el reloj. (3)

Los principales factores que distinguen un SBDD de un sistema centralizado son los siguientes:

- Hay múltiples computadores, llamados sitios o nodos.
- Estos sitios deben de estar comunicados por medio de algún tipo de red de comunicaciones para transmitir datos y órdenes entre los nodos.

Los Doce Objetivos de las Bases de Datos Distribuidas:

1. Principio fundamental: Para el usuario un sistema distribuido debe ser igual que uno centralizado.
2. Autonomía Local: Los sitios distribuidos deben ser autónomos, es decir que todas las operaciones en un sitio dado se controlan en ese nodo.
3. No dependencia de un sitio central: No debe de haber dependencia de un sitio central para obtener un servicio.
4. Operación Continua: Nunca debería apagarse para que se pueda realizar alguna función, como añadir un nuevo nodo.

⁴ Sistema de Base de Datos Distribuida

5. Independencia con respecto a la localización: No debe de ser necesario que los usuarios sepan dónde están almacenados físicamente los datos, sino que más el usuario lo debe de ver como si solo existiera un nodo local.
6. Independencia con respecto a la fragmentación: La fragmentación es deseable por razones de desempeño, los datos, pueden almacenarse en la localidad donde se utilizan con mayor frecuencia de manera que la mayor parte de las operaciones sean sólo locales y se reduzca el tráfico en la red.
7. Independencia de réplica: Si una relación dada (es decir, un fragmento dado de una relación) se puede presentar en el nivel físico mediante varias copias almacenadas o réplicas, en muchos sitios distintos.
8. Procesamiento Distribuido de Consultas: El objetivo es convertir transacciones de usuario en instrucciones para manipulación de datos, y así reducir el tráfico en la red implica que el proceso mismo de optimización de consultas debe ser distribuido.
9. Manejo Distribuido de Transacciones: Tiene dos aspectos principales, el control de recuperación y el control de concurrencia, cada uno de los cuales requiere un tratamiento más amplio en el ambiente distribuido.
10. Independencia con respecto al equipo: El SGBDD⁵ debe ser ejecutable en diferentes plataformas de hardware.
11. Independencia con respecto al Sistema Operativo: El sistema debe ser ejecutable en diferentes Sistemas Operativos.
12. Independencia con respecto a la red: El sistema debe poder ejecutarse en diferentes redes.

Todos los usuarios acceden a la BDD a través de un esquema global en forma transparente al usuario. Debe ser posible ejecutar diferentes SGBDD locales que utilicen distintos modelos de datos. (4)

⁵ Sistema Gestor de Base de Datos Distribuidas

Ventajas de la distribución de datos

La principal ventaja de los sistemas distribuidos es la capacidad de compartir y acceder a la información de forma fiable y eficaz.

- Utilización compartida de los datos y distribución del control

La ventaja principal de compartir los datos por medio de la distribución es que cada localidad pueda controlar hasta cierto punto los datos almacenados localmente. En un sistema centralizado, el administrador de base de datos de la localidad central controla la base de datos. En un sistema distribuido existe un administrador global de la base de datos que se encarga de todo el sistema. Parte de esta responsabilidad se delega al administrador de bases de datos de cada localidad. Dependiendo del diseño del sistema distribuido, cada administrador local podrá tener un grado de autonomía diferente, que se conoce como autonomía local. La posibilidad de contar con autonomía local es en muchos casos una ventaja importante de las bases de datos distribuidas.

- Fiabilidad y disponibilidad

Si se produce un fallo en una localidad de un sistema distribuido, es posible que las demás localidades puedan seguir trabajando. En particular, si los datos se repiten en varias localidades, una transacción que requiere un dato específico puede encontrarlo en más de una localidad. Así, el fallo de una localidad no implica necesariamente la desactivación del sistema.

El sistema debe detectar cuando falla una localidad y tomar las medidas necesarias para recuperarse del fallo. El sistema no debe seguir utilizando la localidad que falló. Por último, cuando se recupere o repare esta localidad, debe contarse con mecanismos para reintegrarla al sistema con el mínimo de complicaciones.

La disponibilidad es fundamental para los sistemas de bases de datos que se utilizan en aplicaciones de tiempo real. Por ejemplo, si una línea aérea no puede tener acceso a la información, es posible que pierda clientes a favor de la competencia.

- Agilización del procesamiento de consultas

Si una consulta comprende datos de varias localidades, puede ser posible dividir la consulta en varias subconsultas que se ejecuten en paralelo en distintas localidades. Sin embargo, en un sistema distribuido no se comparte la memoria principal, así que no todas las estrategias de intersección se pueden aplicar en estos sistemas. En los casos en que hay repetición de los datos, el sistema puede pasar la consulta a las localidades más ligeras de carga. (4)

Desventajas de la distribución de los datos

La desventaja principal de los sistemas distribuidos es la mayor complejidad que se requiere para garantizar una coordinación adecuada entre localidades.

El aumento de la complejidad se refleja en:

- Coste del desarrollo de software: es más difícil estructurar un sistema de bases de datos distribuidos, por lo tanto su coste es mayor.
- Mayor posibilidad de errores: puesto que las localidades del sistema distribuido operan en paralelo, es más difícil garantizar que los algoritmos sean correctos.
- Mayor tiempo extra de procesamiento: el intercambio de mensajes y los cálculos adicionales son una forma de tiempo extra que no existe en los sistemas centralizados. (4)

1.3.1 Fragmentación de datos

El problema de fragmentación se refiere al particionamiento de la información para distribuir cada parte a los diferentes sitios de la red. Esto presenta diferentes interrogantes a la hora de aplicar este método ya que se debe definir cuál es la unidad razonable distribución. Se puede considerar que una relación completa es lo adecuado ya que las vistas de usuario son subconjuntos de las relaciones. Sin embargo, el uso completo de relaciones no favorece las cuestiones de eficiencia sobre todo aquellas relacionadas con el procesamiento de consultas. La otra posibilidad es usar fragmentos de relaciones (subrelaciones) lo cual favorece la ejecución concurrente de varias transacciones que acceden porciones diferentes de una relación. En resumen, el objetivo de la fragmentación es encontrar un nivel de particionamiento adecuado en el rango que va desde tuplas o atributos hasta relaciones completas.

1.3.2 Tipos de fragmentación

Dado que una relación se corresponde esencialmente con una tabla y la cuestión consiste en dividirla en fragmentos menores, inmediatamente surgen dos alternativas lógicas para llevar a cabo el proceso: la división horizontal y la división vertical.

La división o fragmentación horizontal trabaja sobre las tuplas, dividiendo la relación en subrelaciones que contienen un subconjunto de las tuplas que alberga la primera.

La fragmentación vertical, en cambio, se basa en los atributos de la relación para efectuar la división. Estos dos tipos de partición podrían considerarse como fundamentales y básicos.

Sin embargo, existen otras alternativas. Fundamentalmente, se habla de fragmentación mixta o híbrida cuando el proceso de partición hace uso de los dos tipos anteriores. La fragmentación mixta puede llevarse a cabo de tres formas diferentes: desarrollando primero la fragmentación vertical y, posteriormente, aplicando la partición horizontal sobre los fragmentos verticales (denominada partición VH), o aplicando primero una división horizontal para luego, sobre los fragmentos generados, desarrollar una fragmentación vertical (llamada partición HV), o bien, de forma directa considerando la semántica de las transacciones.

Otro enfoque distinto y relativamente nuevo, consiste en aplicar sobre una relación, de forma simultánea y no secuencial, la fragmentación horizontal y la fragmentación vertical; en este caso, se generara una rejilla y los fragmentos formaran las celdas de esa rejilla, cada celda será exactamente un fragmento vertical y un fragmento horizontal (nótese que en este caso el grado de fragmentación alcanzado es máximo, y no por ello la descomposición resultará más eficiente). (5)

1.4 Replicación de datos

La replicación es el proceso de copiar y de mantener los objetos de la BD en los múltiples nodos de BD que incorporan un sistema de base de datos distribuida. Las BDD y la replicación de la BD son términos cercanos pero diferentes. En una BD distribuida pura, el sistema maneja una sola copia de toda la información y soporta los objetos de la BD. Mientras que la replicación confía en la tecnología de la BD para funcionar. La replicación de la BD puede ofrecer las ventajas de las aplicaciones que no son posibles dentro de un ambiente de BDD puro. La replicación es útil para mejorar el funcionamiento y para proteger

la disponibilidad de aplicaciones porque existen las opciones alternas del acceso de los datos. Además, la aplicación puede continuar funcionando si el servidor local experimenta una falla porque otros servidores con datos replicados siguen siendo accesibles. (5)

Existen dos modelos de distribución de datos esencialmente aplicados a la replicación de datos:

Asincrónica: a menudo llamada almacena y reenvía, captura cualquier cambio local, los almacena en una cola, y a **intervalos regulares**, propaga y aplica estos cambios en sitios remotos. Con esta forma de réplica, hay un período de tiempo antes de que todos los sitios alcancen la convergencia de datos.

Sincrónica: también conocida como la réplica en **tiempo real**, aplica cualquier cambio o ejecuta cualquier procedimiento reproducido en todos los sitios que participan en el ambiente de réplica como parte de una sola transacción. Si el procedimiento falla en cualquier sitio, entonces la transacción entera se anula. La réplica sincrónica asegura la consistencia de datos en todos los sitios en tiempo real. (3)

A continuación se muestra una figura que representa los entornos de replicas más utilizados. (Fig. 1)

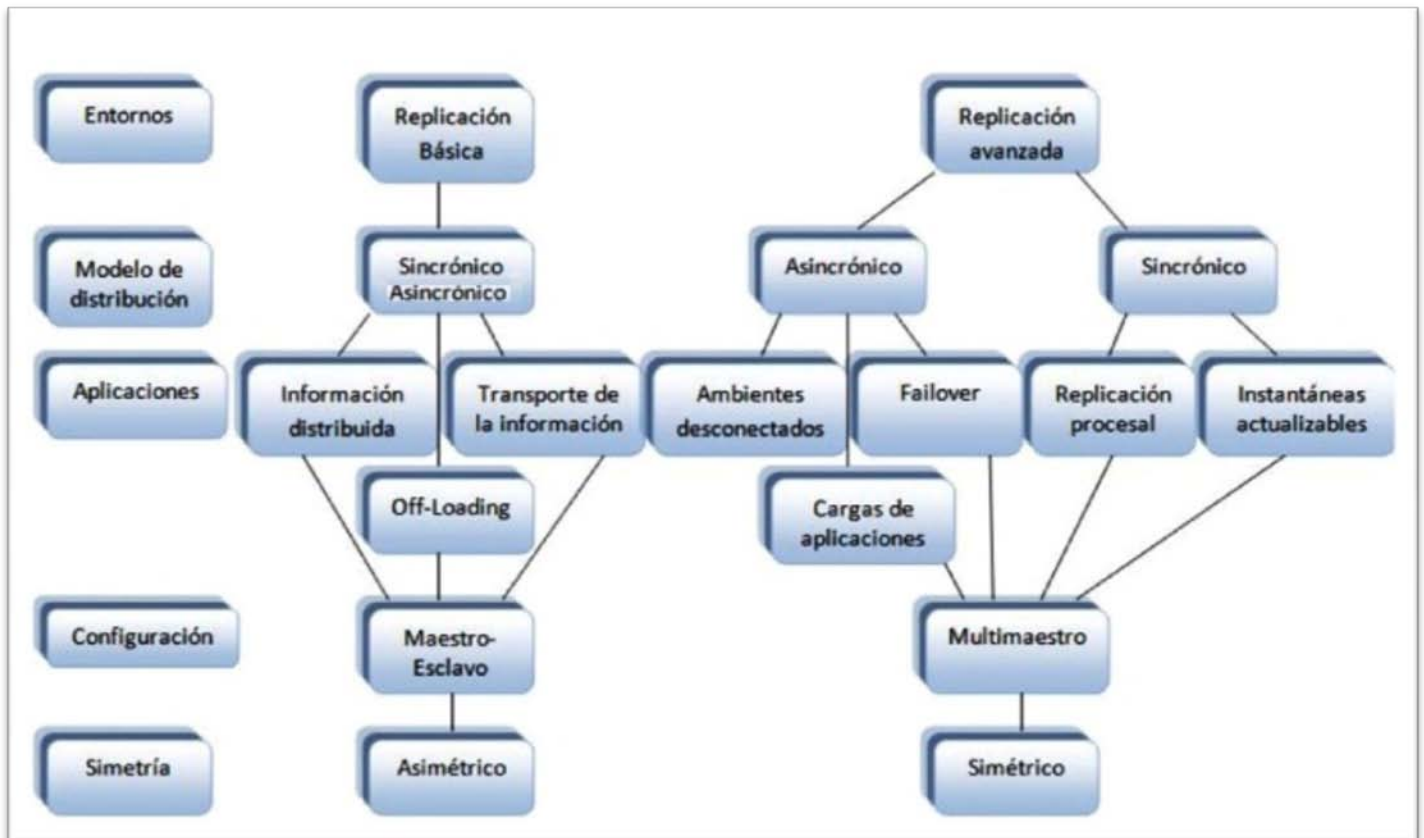


Figura 1. Entornos de réplica más usados.

1.4.1 Replicación (Asimétrica) básica

Con la replicación básica, la replicación de los datos proporciona el acceso de sólo-lectura a los datos de las tablas que provienen de un sitio primario (maestro). Las aplicaciones pueden preguntar datos de las réplicas de datos locales para evitar el acceso a la red, sin importar su disponibilidad. Sin embargo, las aplicaciones a través del sistema deben tener acceso a los datos en el sitio primario cuando las actualizaciones sean necesarias. (6) (Fig. 2)

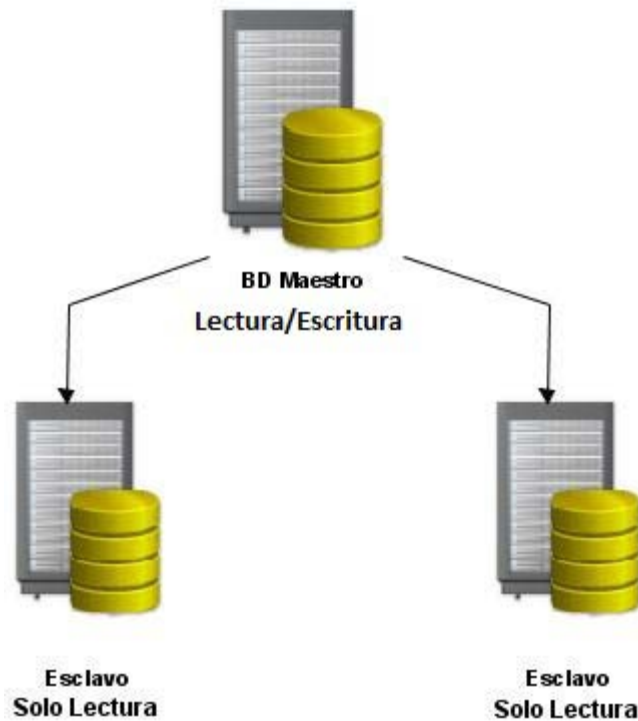


Figura 2 Replicación Básica.

1.4.1.1 Conceptos de la Replicación Básica

Los ambientes de replicación básicos soportan aplicaciones que requieren el acceso de **sólo lectura** a los datos de la tabla que se originan de un sitio primario. A continuación se explican los conceptos fundamentales de los ambientes básicos de la replicación.

- Aplicaciones de la replicación básica.
- Sólo lectura de las Tablas de Instantáneas
- Actualización de Instantáneas.

Aplicaciones de la replicación básica

Distribución de la Información

La replicación básica es útil para la distribución de la información. Por ejemplo, las operaciones de una gran cadena de almacenes de consumo. Con este tipo de negocio, es crítico asegurarse de que la información de precio del producto este siempre disponible, relativamente actual, y constante en todos los distribuidores. Para alcanzar estas metas, cada tienda puede tener sus propios datos del precio del producto que restaure cada noche desde una tabla primaria el precio en las jefaturas corporativas.

Información Off-Loading

La replicación básica es útil como manera de replicar bases de datos enteras o información off-load. Por ejemplo, cuando el funcionamiento de grandes cantidades de transacciones es crítico, puede ser ventajoso mantener una base de datos duplicada para aislar las preguntas exigentes de las aplicaciones de ayuda de decisión. (Fig. 3)



Figura 3 Información Off-Loading.

Transporte de la Información

La replicación básica puede ser útil como mecanismo del transporte de la información. Por ejemplo, la replicación básica puede mover periódicamente datos desde una base de datos de procesamiento transaccional de producción a un almacén de datos. (Fig. 4)



Figura 4. Transporte de la información.

Sólo lectura de las tablas de Instantáneas:

Sólo lectura de las tablas de instantáneas, es una copia local de los datos de la tabla que se originan de unas o más tablas principales remotas. Una aplicación puede preguntar los datos en una instantánea de la tabla, pero no puede insertar, actualizar, o eliminar filas en las instantáneas.

Actualización de Instantáneas:

Los datos que una instantánea presenta, no necesariamente coinciden con los datos actuales de sus tablas principales. Una tabla de instantánea es un reflejo de transacciones consistentes de sus datos principales, como esos datos existieron en un punto específico en el tiempo. Para mantener los datos de una instantánea relativamente actuales con los datos de su maestro, el servidor debe restaurar periódicamente la instantánea. Una actualización de una instantánea es una operación eficiente que hace que esta refleje un estado más actual de su maestro.

1.4.2 Replicación (Simétrica) Avanzada

Las características avanzadas de replicación amplían las capacidades básicas de **sólo lectura** de la replicación, permitiendo que las aplicaciones hagan actualizaciones a las réplicas de las tablas, a través de un sistema replicado de la BD. Con la replicación avanzada, los datos pueden proveer **lectura y acceso a actualizaciones** a los datos de las tablas.

Conceptos avanzados de la Replicación

En ambientes avanzados, las réplicas de los datos pueden proporcionar lectura y acceso a actualizaciones de los datos de una tabla. A continuación se explican los conceptos principales de un sistema avanzado de replicación.

- Aplicaciones para la replicación avanzada.
- Configuraciones Avanzadas de la Replicación.
- Objetos de replicación, Grupos, Sitios y Catálogos.
- Replicación de administradores, propagadores, y receptores.
- Opciones Avanzadas únicas de la Replicación.

1.4.2.1 Aplicaciones para la replicación avanzada

Ambientes Desconectados

La replicación avanzada es útil para el despliegue de las aplicaciones de procesamiento transaccional que funcionan con componentes desconectados (Fig. 5). Por ejemplo, el sistema típico de automatización de la fuerza de ventas para una compañía de seguro de vida. Cada vendedor debe visitar a clientes regularmente con una computadora portátil y grabar órdenes en una base de datos personal, mientras que está desconectado de la red de ordenadores corporativa y del sistema centralizado de la base de datos. Al volver a la oficina, cada vendedor debe remitir todas las órdenes a una base de datos corporativa centralizada. (6)

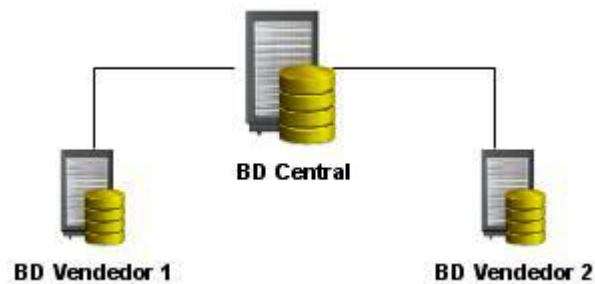


Figura 5 Ambientes Desconectados.

Sitio de Failover

La replicación avanzada puede ser útil para proteger la disponibilidad de una base de datos crítica. Por ejemplo, un sistema avanzado de replicación puede replicar una base de datos entera para establecer un sitio de Failover, si el sitio primario se convierte en inasequible debido a la falla de un sistema o a una interrupción de la red. Tal sitio de Failover puede también servir como base de datos completamente

funcional como alternativa de acceso de la aplicación, cuando el sitio primario es concurrentemente operacional. (Fig. 6).

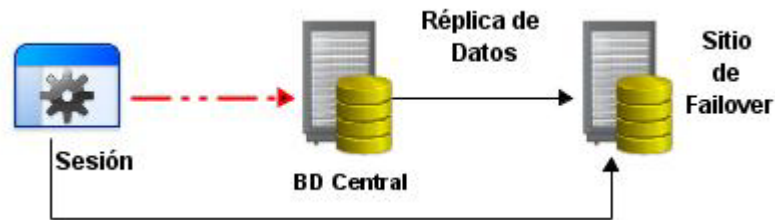


Figura 6 Failover.

Cargas de aplicaciones distribuidas

La replicación avanzada es útil para las aplicaciones de tratamiento transaccional, que requieren puntos múltiples de acceso a la información de la BD con propósitos de distribuir una carga pesada de aplicación, asegurando disponibilidad continua, o proporcionando alternativas de acceso a los datos. Las aplicaciones que tienen tales requisitos comúnmente incluyen servicio de aplicaciones orientadas al cliente. (6) (Fig. 7)

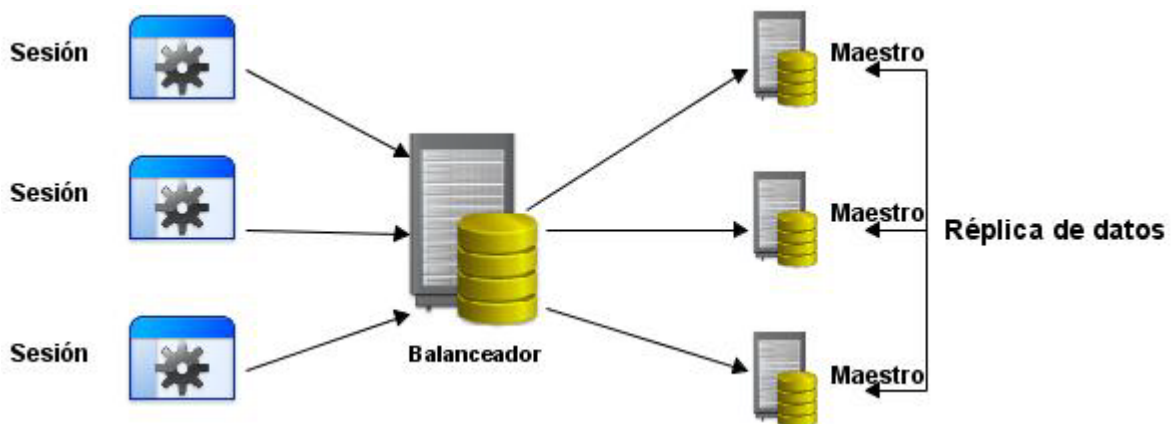


Figura 7 Cargas de aplicaciones distribuidas.

1.4.2.2 Configuraciones avanzadas de replicación

Replicación Multimaestro: La replicación multimaestro permite múltiples sitios actuando como pares iguales, para manejar los grupos de objetos replicados en la base de datos. Las aplicaciones pueden actualizar cualquier tabla replicada en cualquier sitio en una configuración multimaestro. (Fig. 8)



Figura 8 Replicación Multimaestro.

Sitios de instantáneas e instantáneas actualizables

Los sitios principales en un sistema avanzado de replicación, pueden consolidar la información que las aplicaciones actualizan en los sitios remotos de instantáneas. Las facilidades de la replicación avanzada permiten que las aplicaciones inserten, actualicen, y supriman filas de la tabla con instantáneas actualizables. (Fig. 9)

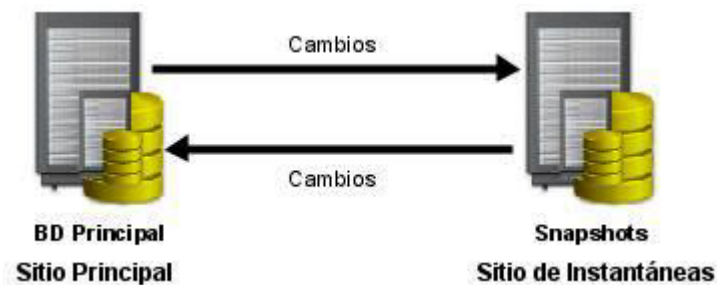


Figura 9. Sitios de instantáneas.

Configuraciones Híbridas

La replicación multimaestro y las de instantáneas actualizables se pueden combinar en configuraciones híbridas para resolver diversos requerimientos de aplicación. Las configuraciones mezcladas pueden tener muchos sitios principales y múltiples instantáneas para cada maestro.

1.4.2.3 Objetos de replicación, grupos, sitios y catálogos

Objetos de Replicación

Un objeto de replicación es un objeto de la base de datos que existe en los servidores múltiples en un sistema de la BDD. Las facilidades avanzadas de la replicación, le permite replicar las tablas y los objetos de soporte tales como vistas, disparadores de la BD, lenguajes procedurales, índices, roles y procedimientos almacenados.

Grupos de Replicación

En un ambiente avanzado, se manejan objetos de replicación usando grupos de replicación. Organizando objetos relacionados de la base de datos dentro de un grupo de replicación, es más fácil administrar muchos objetos juntos. Típicamente, se crea y utiliza un grupo de replicación para organizar el esquema de objetos necesarios para apoyar una aplicación particular de la base de datos. La restricción básica es que un objeto de la replicación puede ser miembro de solamente un grupo.

Sitios de Replicación

Un grupo de replicación puede existir en múltiples sitios de replicación. Los ambientes avanzados de replicación soportan dos tipos básicos de sitios: Sitios maestros y sitios de instantáneas.

Catálogo de Replicación

Cada sitio maestro y de instantáneas en un ambiente avanzado de replicación tiene un catálogo de replicación. El catálogo de replicación de un sitio es un sistema distinto de índice de tablas y vistas que mantienen información administrativa sobre objetos de la replicación y grupos de replicación en el sitio. Cada servidor que participa en un ambiente avanzado de replicación, puede automatizar la replicación de objetos en grupos usando la información en su catálogo de la replicación. (6)

1.4.2.3 Replicación de administradores, propagadores y receptores

Un ambiente avanzado de replicación requiere varias cuentas únicas de usuario de la base de datos para funcionar correctamente, incluyendo administradores de replicación, los propagadores, y receptores.

- Cada sitio en un sistema avanzado de replicación requiere por lo menos a un administrador, un usuario responsable de configurar y de mantener objetos replicados en la base de datos.
- Cada sitio de replicación en un sistema avanzado requiere cuentas especiales de usuario para propagar y aplicar cambios a los datos replicados.

Opciones de Configuración

En la mayoría de las configuraciones avanzadas de replicación, se utiliza apenas una cuenta para todos los propósitos como un administrador de replicación, un propagador de replicación, y un receptor de replicación.

1.4.2.4 Opciones únicas para la replicación avanzada

Algunas aplicaciones tienen requisitos especiales de un sistema avanzado de replicación.

- Replicación Procesal.
- Propagación (en tiempo real) Sincrónica de Datos

Replicación Procesal

Las aplicaciones del procesamiento de lotes, pueden cambiar grandes cantidades de datos dentro de una sola transacción, pudiendo saturar una red. La replicación procesal replica solamente la llamada a un procedimiento almacenado que una aplicación utilice para actualizar una tabla. La replicación procesal no replica modificaciones de los datos.

Para utilizar replicación procesal, en todos los sitios se debe replicar los paquetes que modifican datos en el sistema. Después de replicar un paquete, se debe generar una “envoltura” para este paquete en cada sitio. Cuando una aplicación llama un procedimiento “empaquetado” en el sitio local para modificar datos, la envoltura se asegura de que la llamada esté hecha en última instancia al mismo procedimiento empaquetado en el resto de los sitios en el ambiente replicado.

Propagación Sincrónica de Datos (en tiempo real)

La propagación asincrónica de los datos es la configuración normal para los ambientes avanzados de la replicación. La propagación sincrónica de datos ocurre cuando una aplicación actualiza una replicación

local de una tabla, y dentro de la misma transacción también actualiza el resto de las replicas de la misma tabla.

Por lo tanto, la réplica sincrónica de los datos también se llama réplica en tiempo real de datos. Se utiliza solamente cuando las aplicaciones requieren que los sitios replicados sigan sincronizados continuamente.

1.5 Conflictos de replicación

Los sistemas avanzados de replicación que apoyan un modelo de la actualización de los datos deben tratar la posibilidad de conflictos de replicación, como:

- Conflictos de Unicidad
- Conflictos de Actualización
- Conflictos de Borrado

1.5.1 Métodos de resolución de conflictos

Visión General

Es ampliamente necesario realizar y definir un sistema altamente robusto, para resolver los conflictos de datos que se puedan producir. Como se ha estado explicando anteriormente, los cambios dentro de la BDD se producen y propagan de manera concurrente y asincrónicamente, lo que produce conflictos si dos o más sitios modifican el mismo dato en sitios distintos.

¿Por qué utilizar métodos de resolución de conflictos?

Dichos métodos se usan, principalmente, por dos motivos:

Para asegurar la convergencia de los datos: Esto quiere decir, que los datos no deben ser actualizados inmediatamente, pero si es imprescindible, que en algún tiempo finito se propaguen todos los cambios en todos los repositorios, para asegurar que todo el sistema posee los mismos datos.

Para evitar los errores en cascada: Esto, evita que el sistema caiga en una falla que llevará al sistema a la inestabilidad. El sistema debiese comportarse de manera suave y sin problemas.

Es conveniente considerar lo siguiente, en el diseño de un sistema de resolución de conflictos:

- ✓ Monitorear la ocurrencia de cualquier conflicto sin resolver.

- ✓ Usar un método de notificación, para enviar información a los demás sitios sobre cualquier conflicto inesperado, que sea detectado.

Estos puntos son la base para cualquier sistema que pretenda manejar los conflictos que se producen en la actualización de los datos. En contraste con lo anterior, si todos los sitios propagaran los cambios sincrónicamente y no se tuviesen sitios de “instantáneas” actualizables, no debiesen ocurrir conflictos y no se necesitaría diseñar un método de resolución de conflictos.

Tipos de conflictos

Existen principalmente 3 tipos de conflictos que deben ser detectados por el sistema en cuestión:

Conflictos de Actualización: vale decir, cuando dos sitios intentan actualizar la misma información. En este caso se debe decidir cuál de las dos actualizaciones debe ser hecha primero.

Conflictos de Unicidad: En bases de datos, la unicidad en las llaves primarias es imprescindible, y por lo tanto, no es un problema menor en ambientes distribuidos.

Conflictos de Borrado: se producen conflictos al borrar una determinada fila, sobre todo si un cliente intenta realizar aplicaciones sobre ella y los cambios aún no han sido realizados.

Eligiendo un Sistema de Resolución de Conflictos

Finalmente, la elección de un buen sistema de resolución de conflictos puede tomar tres grandes variantes:

Utilizar un Sistema Propietario o Libre: Existen muchos motores de BD, cada uno de los cuales posee sus propias herramientas para solucionar estos conflictos.

Diseñar un Sistema Propio: También se puede diseñar un sistema propio para tratar de mejor manera los requerimientos específicos para cada caso.

Utilizar un Híbrido entre Ambos: También es posible utilizar el sistema propietario o libre como base, y atacar las debilidades de este con un sistema diseñado propio.

Métodos de Resolución de Conflictos

Los métodos de resolución de conflictos, son funciones o algoritmos que se realizan cuando alguna transacción es realizada, con el objeto de eliminar las inconsistencias que se pudieron haber producido.

Los diferentes sistemas propietarios o libres disponibles poseen sus propias rutinas y métodos para resolver los conflictos. Algunos ejemplos son, comparando el valor que será insertado para ver si es mayor o menor, cuando se trabaja con datos numéricos. También se trabaja con timestamp⁶ de las transacciones y luego se comparan, esto se puede usar, por ejemplo, al trabajar con cadenas de caracteres. (6)

1.6 Tendencias y tecnologías actuales

1.6.1 Software Libre

El Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Este cuenta especialmente con cuatro libertades para los usuarios de software:

- Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.
- Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades, el acceso al código fuente es condición indispensable para esto.
- Libertad 2: la libertad para redistribuir copias y ayudar así a tu vecino.
- Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad, el acceso al código fuente es condición indispensable para esto.

Software Libre es cualquier programa cuyos usuarios gocen de estas libertades. De modo que deberías ser libre de redistribuir copias con o sin modificaciones, de forma gratuita o cobrando por su distribución, a cualquiera y en cualquier lugar. Gozar de esta libertad significa, entre otras cosas, no tener que pedir permiso ni pagar para ello. (7)

⁶ Timestamp: secuencia de caracteres, que denotan la hora y fecha (o alguna de ellas) en la cual ocurrió determinado evento.

1.6.1.2 Software Libre y Software de código abierto

El término «Software Libre» conlleva un problema de ambigüedad para las personas de habla inglesa, un significado indeseado, «software que se puede obtener por un precio cero», que encaja también con el significado deseado: «software que da al usuario ciertas libertades», pero ésta no es una solución perfecta porque no elimina completamente el problema. Un término correcto menos ambiguo sería mejor, si no conlleva otros problemas.

Por desgracia, todas las alternativas en inglés conllevan sus propios problemas. Cualquier sustituto para «software libre» tiene un problema semántico parecido o peor, esto incluye al «software Open Source».

La definición oficial de «software de código fuente abierto», tal y como está publicada por la Open Source Initiative, se acerca mucho a la definición de Software Libre; de todos modos, es algo pobre en algunos aspectos y han aceptado algunas licencias consideradas inaceptablemente restrictivas para los usuarios. De todos modos, el significado obvio para «software de código fuente abierto» es que «puedes ver el código fuente». Este es un criterio más pobre que el del «Software Libre». «Software de código fuente abierto » incluye Software Libre, pero también incluye programas parcialmente libres e incluso algunos programas propietarios. Ese significado obvio para «fuente abierta» no es el significado que sus defensores pretenden. El resultado es que la gente a menudo malinterpreta lo que esos defensores defienden. (7)

1.6.2 Sistema gestor de base de datos

El software que permite la utilización y/o actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina sistema gestor de bases de datos (SGBD⁷).

Es importante diferenciar los términos BD y SGBD.

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

⁷ Sistema Gestor de Base de Datos

Los programas de aplicación operan sobre los datos almacenados en la base utilizando las facilidades que brindan los SGBD, los que, en la mayoría de los casos, poseen lenguajes especiales de manipulación de la información que facilitan el trabajo de los usuarios. (1)

1.6.2.1 PostgreSQL

PostgreSQL es un Sistema de gestión de BD relacional orientada a objetos, es Software Libre y se distribuye bajo licencia BSD.

Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (acrónimo de PostgreSQL Global Development Group: Grupo Global de Desarrollo de PostgreSQL en español). (8)

PostgreSQL es compatible con ANSI SQL⁸, y provee de características que hace posible un diseño de software más complejo. PostgreSQL es extremadamente modular, soporta un gran número de tipos de datos, y es compatible hoy en día con un gran número de interfaces de programación. PostgreSQL está referenciado por los lenguajes de programación más importantes, incluyendo C, Perl, Python, Tel, Java, PHP, ODBC, JDBC, entre otros. PostgreSQL da soporte para la herencia y la seguridad de la capa de dispositivo de transportación de datos (SSL, Secure Sockets Layer). Además cumple completamente con las características ACID (acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español) para realizar transacciones seguras, es multiplataforma, está disponible para 34 plataformas en su última versión estable. (9)

Entre sus principales características se encuentran:

- Alta concurrencia
- El uso de llaves foráneas y de funciones disparadoras
- Vistas
- Integridad transaccional

⁸ Structured Query Language

- Control de versionado concurrente (MVCC en sus siglas en ingles). Estrategia de almacenamiento que permite trabajar con grandes volúmenes de datos.

Además, PostgreSQL puede ser personalizado por el usuario en muchas formas, según sus necesidades, por ejemplo, adicionando entre otros, un nuevo:

- Tipo de datos
- Funciones
- Operadores
- Funciones agregadas
- Lenguajes procedurales

1.6.3 Debian GNU/Linux

Debian GNU/Linux es la principal distribución Linux del proyecto Debian, que basa su principio y fin en el Software Libre. Las características de este sistema operativo le permiten su utilización para estaciones de trabajo, servidores y ordenadores personales. (10)

Creada por el proyecto Debian en el año 1993, la organización responsable de la creación y mantenimiento de la misma distribución, centrado en el núcleo de Linux y utilidades GNU. Este también mantiene y desarrolla sistemas GNU⁹ basados en otros núcleos (Debian GNU/Hurd, Debian GNU/NetBSD y Debian GNU/kFreeBSD). Nace como una apuesta por separar en sus versiones el Software Libre del software no libre. El modelo de desarrollo es independiente a empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia. Debian GNU/Linux se caracteriza por estar disponible para muchas arquitecturas de hardware hoy en día.

- Una amplia colección de software disponible. La versión 4.0 viene con 18733 paquetes.

⁹ GNU Is Not Unix

- Un grupo de herramientas para facilitar el proceso de instalación y actualización del software
- Su compromiso con los principios y valores involucrados en el movimiento del Software Libre.
- No tiene marcado ningún entorno gráfico en especial, pudiéndose instalar varios entornos disponibles.

1.7 Propuesta de desarrollo

Las tecnologías descritas anteriormente cumplen con las normas establecidas en el Decreto Ley 3.390 al estar basadas en tecnologías libres, son compatibles con el resto de las aplicaciones usadas actualmente en el desarrollo de los portales de la Intranet Corporativa y PDVSA.com por lo que se propone implantar un sistema de réplica de datos que replicará las bases de datos contenidas en el SGBD PostgreSQL donde estas aplicaciones se ejecutarán en el sistema Debian GNU/Linux.

1.8 Conclusiones Parciales

En el presente capítulo se han introducido conceptos indispensables para la comprensión del proceso de distribución de la información en base de datos, así como las ventajas y desventajas de la misma, con el uso de las tecnologías actuales. Se concluye que la mejor solución al problema de la disponibilidad y rendimiento de los servidores de bases de datos para la nueva solución de la gestión centralizada de los contenidos que se publican en la Intranet Corporativa y PDVSA.com es el empleo de un entorno de réplica multimaestro para garantizar la integridad de los datos en un SBDD (propagación sincrónica de datos en un ambiente de carga distribuido). Por lo que se hace necesaria una herramienta para la utilización de esta técnica en los servidores de bases de datos de la Intranet Corporativa y el Portal en internet de PDVSA.

Se aborda el tema de software libre ya que las necesidades actuales de PDVSA así lo requieren, por lo que es uno de los aspectos más importantes que se ha considerado en el momento de elegir las herramientas a utilizar para realizar la solución que se propone.

Capítulo 2 Diseño de la solución propuesta

2.1 Introducción

En el presente capítulo se abordará acerca del flujo informativo y la gestión de contenidos de los medios digitales de PDVSA lo que permitirá definir el diseño en el cual se basará la distribución de los datos con el fin de implantar la solución propuesta. Además se describirá a profundidad la herramienta de réplica multimaestro sincrónica con la cual se dará solución al problema.

2.2 Flujo informativo y la gestión de contenidos en la Intranet Corporativa y PDVSA.com.

La nueva solución pretende llevar a los usuarios de la Intranet Corporativa y pdvsa.com un medio eficaz y eficiente que cumpla con los objetivos estratégicos de la organización, garantizando con su implantación, la centralización de los recursos de información, así como la posibilidad de reducción de los tiempos de respuesta del personal de Medios digitales de Asuntos Públicos (AAPP) en el desempeño de su función, la disminución del costo de mantenimiento, alta tolerancia a fallas y un alto nivel de accesibilidad para los usuarios finales.

La nueva solución contará con una Sala de Prensa que será el sitio desde el cual se generen y publiquen todos los contenidos informativos. El nuevo sistema Sala de Prensa tiene como objetivo que la información publicada sea accesible desde la Intranet Corporativa y el portal en Internet mediante el uso de servicios web. (Fig. 10)

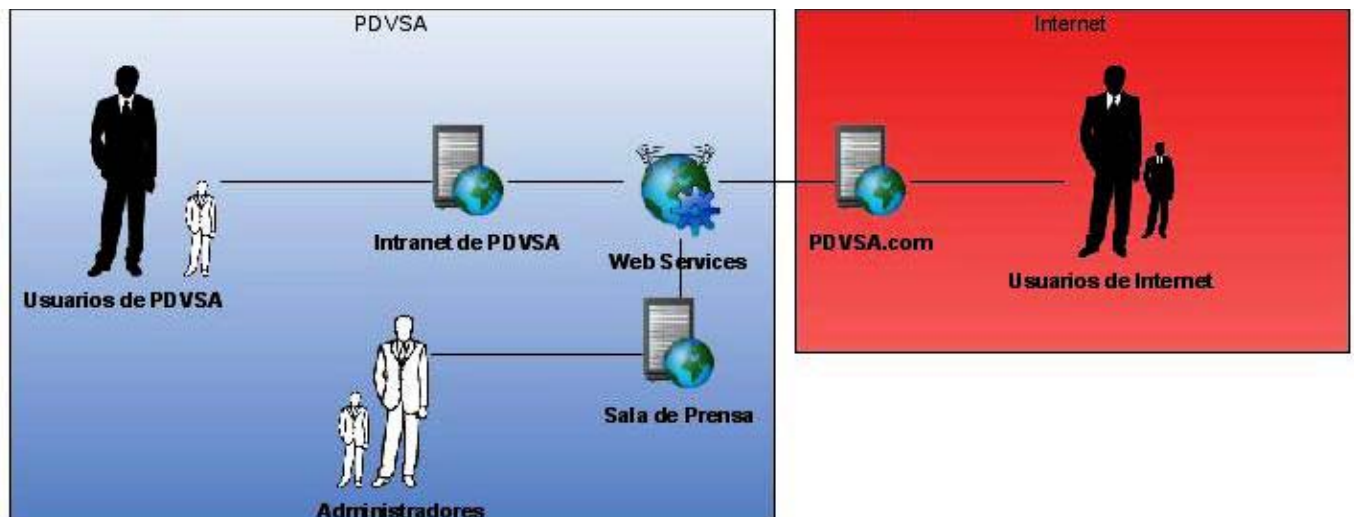


Figura 10. Políticas de acceso.

La Intranet Corporativa es accesible solamente desde la red interna de la corporación y el portal en Internet es de acceso a todos tanto desde la red local así como desde Internet. Debido a estas políticas de acceso y de acuerdo a la confidencialidad de la información publicada en la Intranet Corporativa estos portales se encuentran físicamente separados, en lo referente a redes y a los servidores evitando así que un ataque informático al portal en Internet pueda dañar las aplicaciones de la empresa o acceder la información interna.

La Intranet Corporativa, cuya nueva versión se implementa utilizando el Portal Server Liferay, es el medio en el cual se publican los contenidos noticiosos que se redacten en la Sala de Prensa y estén dirigidos al personal de la Corporación. Es además el Entorno de Trabajo Colaborativo Virtual de PDVSA y el punto de acceso a los servicios que componen la plataforma tecnológica, es decir, a través de la Intranet los usuarios podrán acceder a toda la información y los servicios necesarios para desempeñar su rol en la empresa. Tales razones convierten a la Intranet en un ambiente de alta concurrencia por lo que es de vital importancia garantizar bajos tiempos de respuesta y una alta disponibilidad en los servicios que presta.

El portal de PDVSA en Internet, por su parte, es una puerta informativa que trata de llevar al público externo la realidad de la nueva PDVSA a través de la publicación de los contenidos generados en la Sala de Prensa. Teniendo en cuenta el público potencial, en este caso toda persona con acceso a Internet,

debe ser tratado como un sistema de alta concurrencia. Para la implementación de la nueva versión, el proyecto propone el uso del CMS Drupal en la capa de aplicaciones.

Si se analiza el esquema de la solución propuesta se puede apreciar que los datos necesarios para satisfacer las solicitudes a los servicios de noticias que generan los usuarios internos y externos, se encuentran almacenados en las BD de la Sala de Prensa. Esto trae consigo que la concurrencia en la BD de la Sala de Prensa dependa de la suma de peticiones a los servicios de noticias de los medios digitales mencionados.

El alto número de concurrencia a las BD y la complejidad de cada una de las transacciones pueden provocar la interrupción de los servicios en un servidor de BD centralizada, causado por el gran costo de procesamiento o por un alto consumo de memoria.

2.3 Diseño de la distribución

El diseño de un SBDD implica la toma de decisiones sobre la ubicación de las aplicaciones que accederán a la base de datos y sobre los propios datos que constituyen esta última, a lo largo de los diferentes puestos que configuren una red de computadoras.

Tradicionalmente se ha clasificado la organización de los sistemas de bases de datos distribuidos sobre tres dimensiones:

- El nivel de compartimiento
- Las características de acceso a los datos
- El nivel de conocimiento de esas características de acceso.

El nivel de compartimiento presenta tres alternativas: inexistencia, es decir, cada aplicación y sus datos se ejecutan en una computadora con ausencia total de comunicación con otros programas u otros datos; otra alternativa es compartir sólo los datos y no los programas, en tal caso existe una réplica de las aplicaciones en cada máquina y los datos viajan por la red; y una tercera alternativa donde se reparten datos y programas, dado un programa ubicado en un determinado sitio, éste puede solicitar un servicio a

otro programa localizado en un segundo lugar, el cual podrá acceder a los datos situados en un tercer emplazamiento.

Respecto a las características de acceso a los datos existen dos alternativas principalmente: el modo de acceso a los datos que solicitan los usuarios puede ser estático, es decir, no cambiará a lo largo del tiempo, o bien, dinámico. Se puede comprender la dificultad de encontrar sistemas distribuidos reales que puedan clasificarse como estáticos. Sin embargo, lo realmente importante radica, estableciendo el dinamismo como base, en cuántas variaciones sufre a lo largo del tiempo. Esta dimensión establece la relación entre el diseño de bases de datos distribuidas y el procesamiento de consultas.

La tercera clasificación es el nivel de conocimiento de las características de acceso. Una posibilidad es, evidentemente, que los diseñadores carezcan de información alguna sobre cómo los usuarios acceden a la base de datos. Es una posibilidad teórica, pero sería muy laborioso abordar el diseño de la base de datos con tal ausencia de información. Lo más práctico sería conocer con detenimiento la forma de acceso de los usuarios.

A la hora de abordar el diseño de una BDD se puede optar principalmente por dos tipos de estrategias: la estrategia ascendente y la estrategia descendente.

La estrategia ascendente podría aplicarse en aquel caso donde haya que proceder a un diseño a partir de un número de pequeñas bases de datos existentes, con el fin de integrarlas en una sola. En este caso se partiría de los esquemas conceptuales locales y se trabajaría para llegar a conseguir el esquema conceptual global.

La estrategia descendente debería resultar familiar a la persona que posea conocimientos sobre el diseño de bases de datos, exceptuando la fase del diseño de la distribución. (4)

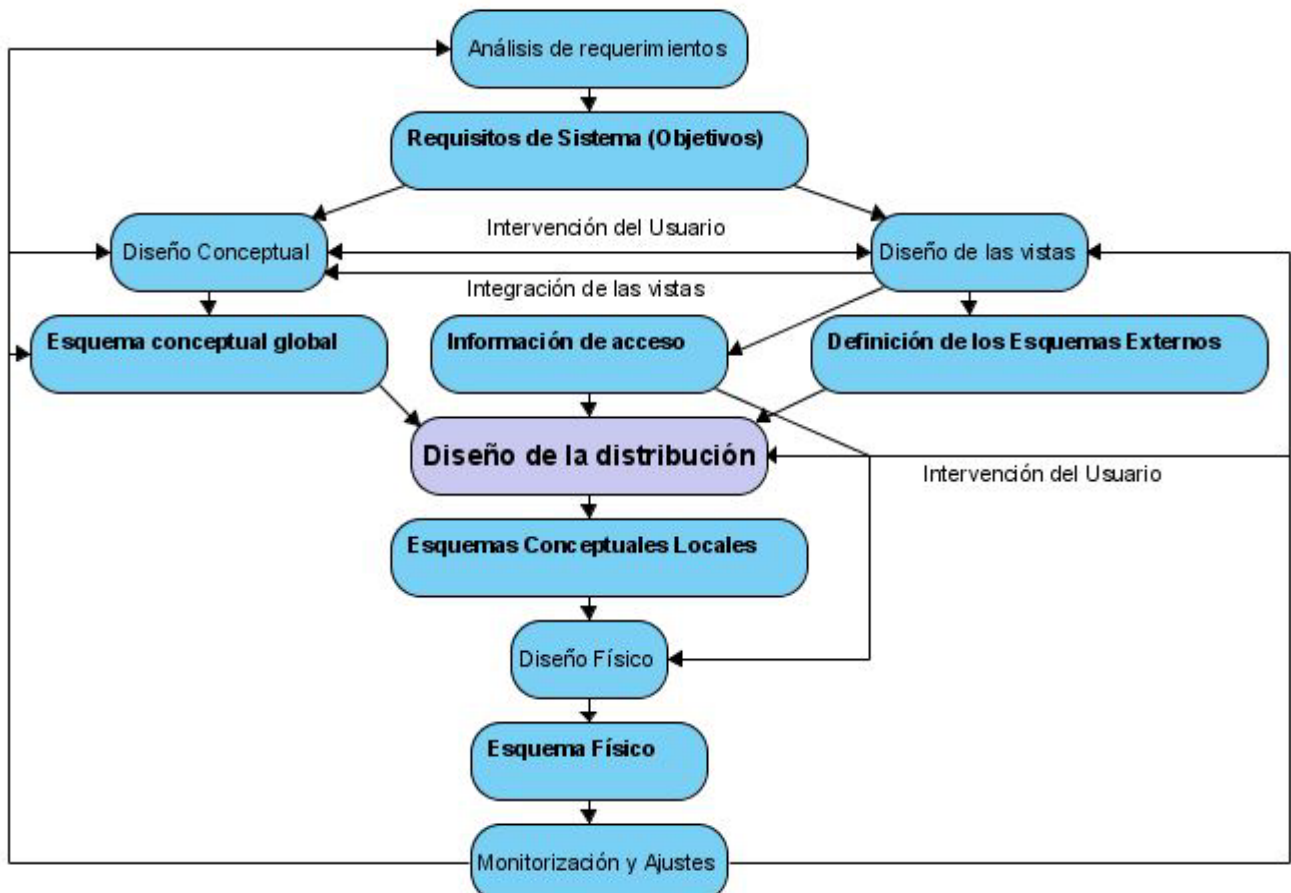


Figura 11. Etapas del diseño en la estrategia descendente.

Las nuevas propuestas de solución están basadas en tecnologías CMS y Portal Server por lo que ya cuentan con su propio diseño de BD por lo que se profundizará en el diseño de la distribución.

El objetivo de esta etapa es la de diseñar esquemas conceptuales locales que se distribuirán a lo largo de todos los puestos del sistema distribuido. Sería posible tratar cada entidad como una unidad de distribución; en el caso del modelo relacional, cada entidad se corresponde con una relación. Resulta bastante frecuente dividir cada relación en subrelaciones menores denominadas fragmentos que luego se ubican en uno u otro sitio. De ahí, que el proceso del diseño de la distribución conste de dos actividades fundamentales: la fragmentación y la asignación.

Diseño de la solución propuesta

El último paso del diseño de la distribución es el diseño físico, el cual proyecta los esquemas conceptuales locales sobre los dispositivos de almacenamiento físico disponibles en los distintos sitios. Las entradas para este paso son los esquemas conceptuales locales y la información de acceso a los fragmentos.

Por último, se sabe que la actividad de desarrollo y diseño es un tipo de proceso que necesita de una monitorización y un ajuste periódicos, para que si se llegan a producir desviaciones, se pueda retornar a alguna de las fases anteriores.

La distribución de las bases de datos en los sistemas distribuidos se hace de 3 maneras: bases de datos particionadas, parcialmente particionadas y de replicación completa. La replicación completa garantiza una alta disponibilidad de datos, calidad del servicio y un gran desempeño. (4)

De acuerdo al flujo informativo y a las políticas de publicación y acceso a la información establecidas en PDVSA se define como unidad de distribución a las BD íntegras de cada localidad y la asignación de las mismas se establece como arquitecturas con políticas de replicación independientes para cada una de las localidades. (Fig. 12)

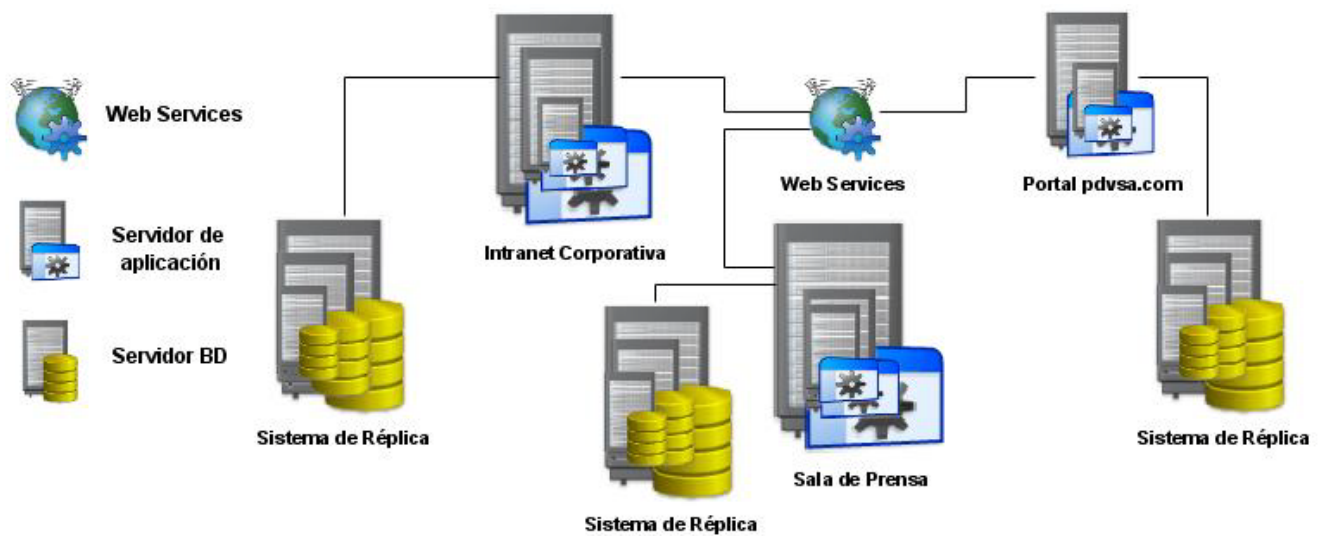


Figura 12. Diseño de la distribución.

2.4 Selección de la herramienta de replicación

Para elegir una herramienta para realizar el proceso de réplica en bases de datos se deben evaluar algunas características importantes que debe manejar la misma.

Disponibilidad: La disponibilidad del sistema se refiere a la accesibilidad de los usuarios a servicios del sistema. Un sistema está disponible si es completamente operacional durante un período de tiempo. A diferencia de la fiabilidad la disponibilidad es instantánea. La disponibilidad se concentra en la duración del tiempo en el cual se espera que el sistema se mantenga en operación continua o en el caso de recuperación confiable. Para los usuarios un sistema posee alta disponibilidad si una denegación de solicitud de servicio es muy rara.

Un sistema fiable no es necesariamente altamente disponible. Por ejemplo, un sistema fiable en el cual ocurran fallos frecuentes pero que siempre se recupere y siempre complete todas las operaciones en progreso, empleará gran cantidad de tiempo en el proceso de recuperación, tiempo en el cual el sistema es inaccesible.

Por otra parte un sistema puede ser altamente disponible pero no tan confiable, por ejemplo un sistema que se reinicia rápidamente sobre fallos sin realizar acciones de recuperación es más disponible que un sistema que si las realiza.

Fiabilidad: La fiabilidad en un sistema es la propiedad que se refiere a la capacidad de tolerar fallos en un largo periodo de tiempo que poseen los componentes que constituyen dicho sistema. Un sistema es perfectamente fiable si este nunca falla. Este hecho está dado si todos los componentes de dicho sistema son fiables por sí mismos y el diseño del sistema no es propenso a fallos o estos componentes y el sistema están diseñados para poder tolerar fallos.

Para aplicar la fiabilidad en un sistema distribuido es necesario el uso de un software tolerante a fallos, este incluye protocolos de detección de fallos, capacidad de adaptarse a estos fallos y la capacidad de prever los fallos mediante configuraciones de dicho sistema.

Integridad: Es la probabilidad de coincidencia de la disponibilidad y fiabilidad de un sistema en un período de tiempo en el cual un servicio inicia y culmina satisfactoriamente. (11)

Escalabilidad de Lectura: Distribuir las consultas, reportes, así como la entrada o salida en caso de operaciones intensas.

Escalabilidad de Escritura: Distribuir las actualizaciones a través de las múltiples bases de datos que conforman el Cluster de BD.

Alta Persistencia: Asegurar que las transacciones de valores críticos como datos financieros o médicos persistan en las múltiples bases de datos evitando pérdidas.

Recuperación ante Desastres: Mantenimiento de los datos y procesamiento de los recursos en una localización remota para asegurar la continuidad del negocio.

Geo-Cluster: Permite a usuarios en diferentes localizaciones geográficas el uso de una base de datos local para procesamiento con la sincronización de otros nodos de bases de datos que conformen el sistema.

A continuación se muestra una tabla comparativa entre las principales herramientas de replicación que soportan PostgreSQL:

Características	pgPool-I	pgPool-II	PGCluster-I	PGCluster-II	Slony-I	Sequoia	Cybercluster
Tipo de replicación	En espera	Multimaestro	Multimaestro	Disco compartido	Maestro Esclavo	Multimaestro	Multimaestro
Comodidad de hardware	Si	Si	Si	No	Si	Si	Si
Modificación de la aplicación	No	No	Si en caso de lectura desde los esclavos	No	Si en caso de lectura desde los esclavos	Necesita actualización de un driver en el cliente	Si
Modificación de la BD	No	No	Si	Si	No	No	No
Soporte de PostgreSQL	>= 7.4 Unix	>=7.4 Unix	7.3.9, 7.4.6, 8.0.1 Unix	8.1	>= 7.3.3	Todas las versiones	>= 8.1...8.3
Pérdida de datos ante fallos	Si	Si	No	No	Si	No	No
Tolerancia a fallos en caso de error en la BD	Si	Si	Si	No si se realiza desde el disco	Si	Si	Si
Tolerancia a fallos transparente	No	No	No	No	No	Si	Si
Recuperación ante desastres	Si	Si	Si	No si se realiza desde el disco	Si	Si	Si
Balance de consultas	No	Si	Si	Si	Si	Si	Si

Diseño de la solución propuesta

Escalabilidad de lectura	Si	Si	Si	Si	Si	Si	Si
Escalabilidad de escritura	No	No	No	Si	No	No	Si
Paralelización de consultas	No	Si	No	No	No	No	No
Réplicas	2	Hasta 128	Balanceadores o límites de replicadores	Límite SAN	Ilimitado	Ilimitado	Ilimitado
Alta persistencia	No	Si	No	Si	No	Si	Si
Añadir un nodo al vuelo	No	Si	Si	Si	Si	Si	Si
Actualizaciones en línea	No	No	Si	No	Si	Si	Si
Cluster heterogéneo	≥ 7.4	≥ 7.4	Si	Si	$\geq 7.3.3$	Si	Si
Soporte de Geo-Cluster	No	No	Posible pero no se usa	No	Si	Si	Si
Simetría	Simétrico	Simétrico	Simétrico	Simétrico	Asimétrico	Asimétrico/ Simétrico	Simétrico

Tabla 1. Comparación entre herramientas libres de replicación.

A partir del análisis de las características expuestas en la tabla anterior, las dos herramientas que resuelven de manera óptima el problema de rendimiento y disponibilidad de los servidores para la gestión centralizada de la información utilizando replicación multimaestro son Sequoia y Cybercluster, las dos poseen licencias libres aunque Sequoia es más restrictiva. Sequoia al estar desarrollado en Java depende para su funcionamiento de la máquina virtual de Java (JVM), esto le añade a los servidores un costo adicional de consumo de memoria, Cybercluster por su parte está desarrollado en C lo que permite un aprovechamiento óptimo de recursos. Aunque ambos no dependen de un hardware específico y permiten múltiples servidores, Sequoia requiere de un método de resolución de conflictos mientras que Cybercluster no lo necesita, esto le ofrece a Cybercluster ventajas en cuanto a fiabilidad.

Todo lo anteriormente expuesto evidencia que la herramienta Cybercluster es la más adecuada para darle solución al problema en cuestión.

2.4.1 Cybercluster

¿Qué es Cybercluster?

Cybercluster es una herramienta de replicación multimaestro sincrónica basado en PostgreSQL. Es el primer software a escala empresarial que proporciona tecnología de replicación de alto rendimiento a las compañías que utilizan PostgreSQL. El uso de esta herramienta es óptimo para aplicaciones web que requieren una alta demanda de lectura.

Cybercluster es software libre. Puede usarse libremente bajo los términos de la licencia de BSD. Esto significa que ningún costo de licencia tiene que ser pagado.

Cybercluster sincroniza escrituras al banco de datos para asegurarse que todos los nodos del banco de datos dentro del racimo contengan los mismos datos. En adición al acceso de lectura se encuentra el balanceador de carga que asegura que todos los nodos de los bancos de datos pueden trabajar eficazmente al mismo tiempo. En la práctica esto significa que el acceso de lectura pueda ser casi infinito.

Características de Cybercluster

Cybercluster proporciona un conjunto de características apreciables. La apreciación global siguiente contiene una lista de las funcionalidades más importantes y da una visión de lo que Cybercluster puede hacer.

Funcionalidad	Balanceador de Carga	Cybercluster proporciona algoritmos de Balanceador de Carga para asegurarse que todos los nodos del banco de datos pueden usarse durante el funcionamiento normal. Esto es sobre todo importante cuando se tiene lectura de alto rendimiento.
	Manejo de errores	Cuando el software del racimo descubre un error automáticamente Cybercluster descarta los nodos rotos del sistema en funcionamiento y continúa su funcionamiento normal en las máquinas de las bases de datos restantes. En cuanto el nodo roto se repare, puede agregarse al banco de datos del sistema de forma segura nuevamente. Ningún tiempo fuera de servicio se necesita para realizar este funcionamiento si por lo menos 3 nodos del banco de datos están disponibles (1 nodo activo, 1 nodo para sincronización, 1 nodo roto).

	Recuperación	Cuando un nodo del sistema se inicia en modo recuperación automáticamente se sincroniza utilizando un maestro activo.
Replicación	COPY()	Soportado
	NOW()	Soportado
	NEXTVAL (), SETVAL ()	Soportado
	Tipos de serie	Soportado
	Procedimientos almacenados	Soportado
	Objetos grandes	Soportado

Tabla 2. Características de Cybercluster.

Cybercluster tiene soporte en otras características que son muy importantes a la hora de diseñar e implementar un sistema de replica puro.

DDL¹⁰: Soportado.

Protocolo de confirmación en dos fases: Soportado.

Restricciones

Al reproducir un objeto grande, necesita ser puesto en un directorio que pueda ser leído por todos los nodos del banco de datos.

Composición del sistema

Cybercluster consiste en Balanceadores de Carga, (Cluster) Racimo de los Nodos de BD y Servidores de la Replicación.

Racimo los nodos de DB

El cluster (nodos del racimo) son las máquinas que procesan las demandas reales que entran. Siempre que una demanda se reciba del cliente un Cluster de BD tiene que determinar si está enfrentando una

¹⁰ DDL- Data Definition Language, Lenguaje de definición de datos

demanda de lectura o escritura. La demanda de lectura es ejecutada directamente por el banco de datos y el resultado se envía al cliente. En caso de que sea una demanda de escritura, el Cluster de BD envía un mensaje al gerente de la Replicación para asegurarse que ese dato se envía a todos los nodos del banco de datos restantes dentro del racimo.

Servidor de la Replicación

El servidor de la Replicación es el componente central del sistema. Toma demandas entrantes solo de los nodos del banco de datos y copias esos cambios a todos los nodos del banco de datos en el sistema. Si el servidor de la replicación descubre un problema en un nodo del banco de datos quitará la máquina de la lista de banco de datos activos y se escribirá al disco un log que contiene una descripción del error detallada. Esto ocurre para asegurar que los nodos del banco de datos nunca puedan estar fuera de sincronización en caso de fallo. Basado en los logs¹¹ el administrador del sistema puede decidir entonces si es posible agregar la máquina de nuevo al racimo.

Servidor de Balanceador de Carga

Cybercluster contiene su propio balanceador de carga que puede usarse para distribuir la carga dentro del racimo. La carga en el sistema es determinada por el número de preguntas activas. La máquina con el número más bajo de preguntas activas será escogida para realizar una nueva demanda.

Si el balanceador de carga detecta un problema en un nodo activo del banco de datos automáticamente descarta este nodo del sistema activo para asegurarse que ninguna otra solicitud se enviará a esta máquina rota.

El Balanceador de Carga usado con Cybercluster es opcional. Si ningún balanceador de carga está disponible, las aplicaciones se pueden conectar a cualquier nodo del banco de datos dentro del sistema. El propio balanceador de carga por sí mismo no está de ninguna manera involucrado en la replicación, simplemente se usa para distribuir la carga.

De acuerdo al diseño de la distribución propuesto y en correspondencia a las características de la herramienta seleccionada se propone que el modelo físico para cada uno de los sistemas de réplica sea:

¹¹ Registros de bitácora

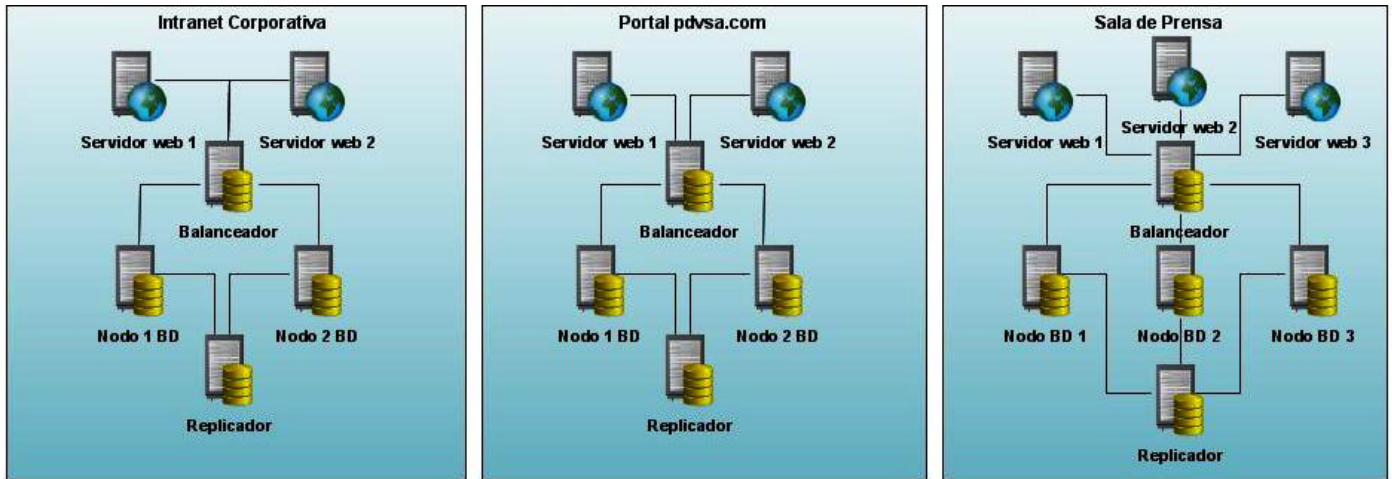


Figura 13. Esquema de entornos de réplicas.

2.5 Conclusiones parciales

En el presente capítulo se abordaron las características del flujo informativo de los medios digitales de PDVSA lo que permitió a través de su análisis definir un diseño de la distribución de las BD a nivel global. Además se le dio cumplimiento a uno de los objetivos al seleccionar y describir una herramienta de replicación de BD conforme a las políticas de replicación establecidas para darle solución al problema y acorde a las características que debe cumplir este tipo de herramienta.

Capítulo 3 Implantación y prueba de la solución propuesta

3.1 Introducción

En el presente capítulo se implantarán los modelos físicos a partir del diseño de la distribución establecido en el capítulo anterior. Además se verificará el correcto funcionamiento del sistema mediante la realización de pruebas utilizando un entorno de desarrollo de pruebas.

3.2 Implantación

Para la correcta implantación del sistema se hace necesario la previa instalación y configuración de manera correcta del mismo.

3.2.1 Instalación y configuración de Cybercluster

3.2.1.1 Sistema Operativo

La instalación de la herramienta partirá de la previa instalación de Debian GNU/Linux en su versión estable. Luego de la instalación se debe actualizar el sistema base a la última rama estable, para esto previamente es obligatorio configurar los repositorios.

3.2.1.2 Instalación

Dos opciones para instalar Cybercluster están disponibles:

Instalación de la fuente: *.tar para que los usuarios puedan construir su propio sistema.

Distribuciones Binarias: Para las plataformas destacadas se proporcionan paquetes binarios. (Debian Sarge (32 bit, 64bit), Debian Etch (32 bit, 64 bit)).

Para iniciar el proceso de instalación se hace necesario iniciar sesión en el sistema con privilegios de administración, es decir como root. Es importante aclarar además que todo el proceso de instalación, configuración y administración será mediante la consola o terminal.

La primera tarea a realizar será descompactar el paquete con las fuentes de Cybercluster en su versión 1.2.1, para esto instalaremos el programa **bzip2**:

```
#apt-get install bzip2
```

Implantación y prueba de la solución propuesta

Descompactando:

```
#tar -jxvf cybercluster-1.2.1.tar.gz2
```

Después de descompactado el fichero se necesitan instalar algunas dependencias antes de poder comenzar con la instalación de Cybercluster v1.2.0. Las **dependencias** son las siguientes:

```
gcc, g++, zlib1g-dev, libreadline-dev, libreadline-dbg, make, flex, bison, openssl
```

Después ejecutar desde la terminal o consola:

```
#apt-get install gcc g++ zlib1g-dev libreadline-dev libreadline-dbg make openssl
```

Una vez descompactado las fuentes de Cybercluster e instalar las dependencias necesarias es preciso acceder al directorio en el cual se encuentran las fuentes descompactadas:

```
#cd cybercluster-1.2.1
```

Ya en el directorio cybercluster-1.2.1, ejecutar la siguiente secuencia de comandos:

```
./configure -with-openssl
```

Revisa y configura las dependencias y archivos necesarios para su posterior uso, la opción **-with-openssl** permite cifrar el canal de comunicación.

make Construye la aplicación para su posterior instalación.

Si todo ha salido bien saldrá un mensaje como este:

```
All of PostgreSQL successfully made. Ready to install.
```

Para la instalación se hace necesario acceder como root ya que en este paso se copian y modifican archivos en el sistema.

```
$su
```

```
#make install
```

Si todo ha salido bien saldrá un mensaje como este:

```
PostgreSQL installation complete.
```

3.2.1.3 Configuración de Cybercluster

Inicializar las BD

Por razones de seguridad es recomendable crear un usuario postgres para la ejecución de los comandos de inicio parada en lo referente al servidor de BD ya que los datos se transmiten desde las aplicaciones

hasta los servidores con los permisos de este usuario y ejecutar este servicio como root ofrece potenciales vulnerabilidades a la seguridad del sistema.

```
#adduser postgres (no ejecutar si tiene un usuario postgres creado anteriormente)
```

Para la correcta ejecución de los servicios es necesario crear un directorio en el cual permanecerán los archivos de configuración de PostgreSQL 8.3

```
#mkdir /usr/local/pgsql/data
```

Se le atribuye control total al usuario postgres sobre la carpeta creada “data”.

```
#chown postgres /usr/local/pgsql/data
```

Una vez creada la carpeta y atribuidos los permisos al usuario “postgres” se accede al sistema con los privilegios de ese usuario, a fin de crear los archivos de configuración de PostgreSQL e iniciar sus servicios.

```
#su - postgres
```

```
$/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
```

El parámetro **-D** le indica al comando que el directorio en el cual son creados los archivos de configuración.

3.2.1.4 Aspectos a considerar al instalar la herramienta:

- **Cybercluster versión 1.2.1** instala por defecto **PostgreSQL 8.3**.
- Una vez instalado **Cybercluster versión 1.2.1** se puede configurar como Servidor de BD, Servidor de Replicación o Balanceador de Carga.
- Un sistema de réplica utilizando la herramienta Cybercluster 1.2.1 puede tener un número ilimitado de servidores ya sea de: Cluster de BD, de Replicación o Balanceador de carga.
- Si existe otra versión de PostgreSQL instalada anteriormente en el servidor o los servidores donde se instalará Cybercluster 1.2.1 se debe realizar alguna de estas operaciones para poder iniciar el servicio de Cluster BD o Balanceador de Carga. Esto no significa que no se pueda instalar Cybercluster 1.2.1 si existe otra versión de PostgreSQL instalado anteriormente.
 - Instalar la herramienta para recibir conexiones por un puerto diferente al **5432**, por el que por defecto se inicia el servidor PostgreSQL.
 - Antes de iniciar algún servicio de los que brinda la herramienta Cybercluster, detener el servicio PostgreSQL existente (Ejecutar como root del sistema “**/etc/init.d/postgresql** versión **stop**”).

Implantación y prueba de la solución propuesta

- Desinstalar la versión existente de PostgreSQL.

3.2.2 Configuración de los servicios mediante un ejemplo de composición del sistema

- **Nombre del Host:** hostname. (Dirección IP o Nombre de Dominio)
- **Puerto de entrada:** receive port.
- **Puerto de recuperación:** recovery port.

Ejemplo de Composición del Sistema: (Fig. 14)

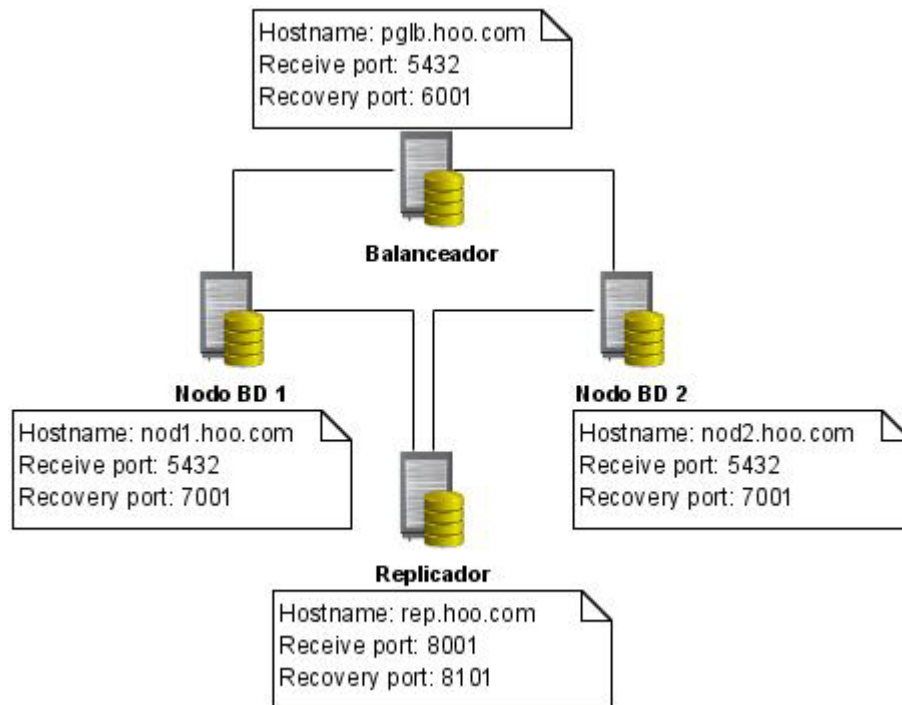


Figura 14. Ejemplo de composición del sistema.

3.2.2.1 Archivos de configuración:

Cada uno de los servidores posee archivos de configuración necesarios para su correcto funcionamiento, estos son: (Fig. 15)

Implantación y prueba de la solución propuesta

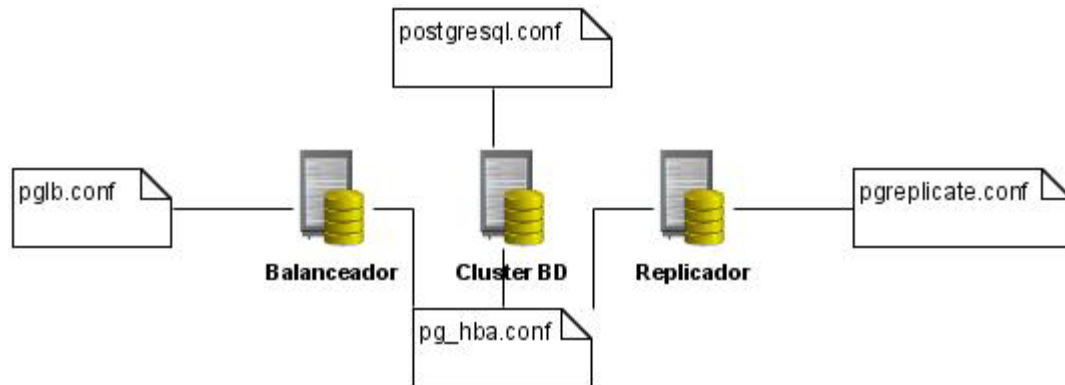


Figura 15. Archivos de configuración.

3.2.2.2 Configuración de los archivos `pg_hba.conf` y `postgresql.conf`.

Archivo `pg_hba.conf`

La autenticación del cliente es controlada por un archivo de configuración, el cual es tradicionalmente llamado `pg_hba.conf` y es almacenado en el directorio de datos del cluster (`/usr/local/pgsql/data`) (HBA stands for host-based authentication). Este archivo es creado por defecto al ejecutar `initdb`. Este archivo controla:

- ✓ A cuales clientes se les permite acceder, el mecanismo de autenticación, que usuarios de la BD tienen acceso y a cuales BD pueden acceder.

El formato general del archivo consiste en un conjunto de opciones de configuración y sus parámetros, cada opción de configuración por línea. Las líneas en blanco son ignoradas, así como cualquier texto detrás del símbolo `#`, lo que indica que es un comentario. Cada parámetro estará separado por espacios o espacios de tabulador.

Cada opción de configuración especifica un tipo de conexión, un rango de dirección IP del cliente (si es relevante para el tipo de conexión), el nombre de la BD, usuario, y el método de autenticación que corresponda al establecido en las opciones de configuración. La primera solicitud en la cual correspondan sus parámetros con los establecidos es la usada para establecer la conexión. Si uno de los parámetros es erróneo la conexión no se establecerá.

La estructura puede tomar cualquiera de las siguientes formas:

```
# local          DATABASE          USER          METHOD[ OPTION ]
# host           DATABASE          USER          CIDR-ADDRESS  METHOD[ OPTION ]
# hostssl        DATABASE          USER          CIDR-ADDRESS  METHOD[ OPTION ]
# hostnossl      DATABASE          USER          CIDR-ADDRESS  METHOD[ OPTION ]
```

Implantación y prueba de la solución propuesta

El primer campo se refiere al tipo de conexión “local”, que es cuando se establece la conexión desde el mismo ordenador, “host” se refiere a un tipo de conexión en texto plano o encriptado usando SSL, “hostssl” encriptado usando SSL y en “hostnossl” establece que la conexión no estará encriptado usando SSL. Para el campo DATABASE se pueden usar los parámetros “all”, “sameuser”, “samerole”, el nombre de la BD o varias BD separadas por comas. El campo USER se puede establecer usando los parámetros “all”, un nombre de usuario, el nombre de un grupo de usuarios precedido por un signo “+” o una lista de usuarios separados por comas. CIDR_ADDRESS se refiere a la dirección o direcciones desde donde se establecerán las conexiones, pueden establecerse usando direcciones IP en formato IPv4 o IPv6, e incluso usando máscaras de subred. El campo METHOD se refiere al mecanismo de autenticación a usar, este puede ser “trust”, “reject”, “md5”, “crypt”, “password”, “krb5”, “ident”, “pam” o “ldap”. Cybercluster necesita de un ambiente de confianza entre los nodos que componen el sistema, o sea, se establecerá como método de autenticación la opción “**trust**”.

El método “**trust**” le permite el acceso a cualquier usuario sin el uso de contraseñas, o sea es una regla de acceso para ambientes de confianza cliente-servidor. Este acceso se restringirá a través de la opción CIDR-ADDRESS especificando las direcciones confiables al servidor.

Archivo **postgresql.conf**:

Es el archivo encargado de establecer la configuración para la ejecución del servidor PostgreSQL. Para un correcto uso de los recursos del sistema así como para el óptimo funcionamiento del servidor existen un conjunto de parámetros descritos en el archivo que deberán ser cambiados. Además se necesitarán realizarles configuraciones adicionales al sistema operativo para que estos parámetros se puedan configurar.

Archivo **postgresql.conf**

```
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
# - Connection Settings -  
listen_addresses = '          # what IP address (es) to listen on;  
                             # comma-separated list of addresses;  
                             # defaults to 'localhost', '*' = all  
                             # (change requires restart)  
port = 5432                  # (change requires restart)  
max_connections = 50        # (change requires restart)
```

Implantación y prueba de la solución propuesta

```
ssl = on#
```

listen_addresses es la opción en la que se especifica al servidor las direcciones desde la cual PostgreSQL aceptará conexiones, se especifican las direcciones de la siguiente forma: “*” para todas las direcciones, una sola dirección o una lista separadas por comas y “localhost” solo para conexión local.

port es el puerto por el cual el servidor PostgreSQL aceptará conexiones.

ssl es la opción que habilita el uso de SSL, esto evita que el enviar sentencias SQL los resultados obtenidos viajen en forma clara y que cualquiera tenga acceso al tráfico de la red pueda ver la información que se está transmitiendo. Esta opción se usa en conjunto con las opciones del archivo **pg_hba.conf**.

max_connections en esta opción se establecerá el número máximo de conexiones permitidas en hilo, esta opción es recomendable configurar teniendo en cuenta el uso de los recursos del sistema y la cantidad de conexiones reales que puedan existir. Se configura a la par de la sección siguiente:

```
# - Memory -  
shared_buffers = 8000kB# min 128kB or max_connections*16kB
```

Es la opción en la cual se establece la memoria física total del sistema asignado para el uso del servidor PostgreSQL, se establece un mínimo de 128kB y un máximo que se calcula de la siguiente manera: $\text{max_connections} * 16\text{kB}$ para su valor mínimo, es recomendable calcularlo de la siguiente manera: RAM disponible dividido por cuatro.

```
temp_buffers = 8Mb
```

Establece el número máximo de buffers temporales usados por cada sesión. Es usado actualmente solo para mantener las tablas temporales en memoria. El valor por defecto recomendado es de 8Mb.

```
max_prepared_transactions
```

Establece el número máximo de transacciones simultáneas preparadas. En caso de que se requieran transacciones preparadas es recomendable establecer este parámetro igual a **max_connections**, de no ser necesario transacciones preparadas debe establecerse igual a cero.

```
work_mem = 2MB# min 64kB
```

Establece la memoria máxima a ser usada para las operaciones de ordenamiento y consultas. Es recomendable calcular este valor como RAM disponible dividido por **max_connections**.

```
maintenance_work_mem = 512MB# min 1MB
```

Establece la memoria a ser usada para las operaciones de mantenimiento, estas incluyen operaciones como VACUUM y CREATE INDEX. Es recomendable calcularlo como RAM disponible dividido por ocho.

```
max_stack_depth = 9MB
```

Establece la máxima profundidad de la pila. Este valor se obtiene auxiliándose del comando `ulimit -s`.

Implantación y prueba de la solución propuesta

```
# - Free Space Map -
```

```
max_fsm_pages
```

Establece el número máximo de páginas de disco en el cuál el espacio libre es referenciado. Es recomendable calcular es valor de la siguiente manera el resultado de la división entre tamaño de la BD y 8, dividido por un valor de 8 a 16, este resultado debe ser mayor que max_fsm_relations multiplicado por dieciséis.

```
max_fsm_relations
```

Establece el número máximo de tablas e índices para las cuales el espacio libre en disco es referenciado. Es recomendable establecer este valor como el total de tablas de la BD más veinte.

```
# WRITE AHEAD LOG
```

```
fsync = on
```

Fuerza la sincronización en disco de las actualizaciones. El servidor usará la llamada al sistema fsync () en varios lugares para asegurarse de que las actualizaciones son escritas físicamente en disco. Esto asegura que el cluster de BD será recuperado a un estado consistente luego de un fallo en el sistema operativo o en el hardware.

```
synchronous_commit = on
```

Esta opción le dice al sistema que se realizará una llamada a fsync () luego de realizar algún cambio. Es recomendable establecer este valor a “on” si se quiere asegurar una mayor integridad de los datos.

```
wal_sync_method = fsync
```

Este parámetro le especifica cual método será usado para forzar las actualizaciones WAL¹² al disco.

```
wal_buffers = 4MB
```

Este parámetro establece el número de buffers de disco en la memoria compartida para WAL. Además establece el número máximo de datos que se perderán en caso de establecer el parámetro synchronous_commit a off.

```
commit_delay
```

Establece la demora en microsegundos entre la culminación de transacciones y el volcado en disco de WAL. El valor recomendado para este parámetro es de 0.

```
commit_siblings
```

Establece el mínimo de transacciones concurrentes activas antes de chequear el parámetro commit_delay. El valor por defecto de este parámetro es de 5 aunque el recomendado es de 1.

¹² Write Ahead Log

Implantación y prueba de la solución propuesta

- Checkpoints -

`checkpoint_segments`

Establece la mayor distancia entre segmentos entre puntos de chequeo automáticos WAL. El valor por defecto de este parámetro es de 3 aunque el recomendado es de 16 a 128.

`checkpoint_completion_target`

Establece el tiempo empleado para volcar los datos no necesarios del buffer durante un punto de chequeo. Es recomendable establecer este valor igual a 0.9.

- Planner Cost Constants -

`effective_cache_size`

Este parámetro le especifica al planificador de consultas de PostgreSQL cuanta memoria RAM es estimada para cachear los datos, en ambos buffers de memoria compartida y en la caché del sistema. Este parámetro ayuda a realizar buenas estimaciones de costo, aunque esta memoria no es liberada.

- Other Planner Options -

`default_statistics_target`

El software de BD recolecta estadísticas de cada una de las tablas en su BD para decidir cómo se ejecutarán las consultas sobre ellas. Por defecto, no recolecta demasiada información, y si no se obtienen buenos planes de ejecución particularmente en las más largas (o variadas) tablas se debería incrementar `default_statistics_target` y luego ejecutar ANALYZE a la base de datos nuevamente. El valor por defecto para `default_statistics_target` en hardware más moderno debe ser llevado a 100.

`constraint_exclusion = off`

Este parámetro le habilita al planificador de consultas el uso de limitaciones para optimizar las consultas. Este valor solo se debe establecer a “on” en caso de usar fragmentación de datos.

AUTOVACUUM PARAMETERS

`autovacuum = on`

Este parámetro controla cuando el servidor debe ejecutar el proceso de autovacuum. Está establecido por defecto a “on” aunque el parámetro `track_counts` debe ser cambiado a “on” para que esto pueda funcionar.

```
# LOCK MANAGEMENT
```

```
deadlock_timeout
```

Establece el tiempo de espera de un bloqueo antes de chequear si existe un punto muerto. El valor recomendado para este parámetro es de 5.

```
max_locks_per_transaction
```

Establece el número máximo de bloqueos por transacciones. El valor por defecto de este parámetro es de 64, aunque para BD con esquemas muy complejos o con transacciones muy largas necesitan una cantidad mayor.

3.2.2.3 Cluster de servidores de BD

En un servidor donde se aloja un Nodo BD creado con la herramienta Cybercluster se necesitan editar tres archivos.

Archivo #1: **pg_hba.conf**

Archivo #2: **postgresql.conf**

Archivo #3: **cluster.conf**

Ruta a los archivos: **/usr/local/pgsql/data**

Archivo: **cluster.conf**

Este archivo controla cual(es) nodo(s) es/son usado(s) como replicador(es) y que puerto se usa para establecer la conexión con el/los mismo(s). Además que puerto es usado por el replicador para la solicitud de replicación desde el servidor de replicación y cual comando es usado para la función de recuperación.

Este archivo está dividido en las siguientes secciones:

Replication Server Information: en esta sección se establece la información de los servidores de replicación.

```
#o Host_Name: Dirección del servidor de replicación
#o Port: Puerto para establecer la conexión
#o Recovery_Port: Puerto de conexión para el proceso de recuperación
```

Cluster DB Server Information: en esta sección se establece la información del servidor de BD.

```
#o Host_Name: Nombre del nodo que se conecta al replicador
#o Recovery_Port: Puerto de conexión para el proceso de recuperación
#o Rsync_Path: Ruta al comando rsync
#o Rsync_Option: Opción de transferencia de archivos de rsync
#o Rsync_Compress: Opción de compresión de rsync
```

Implantación y prueba de la solución propuesta

```
#[yes/no]. Por defecto: yes
#o Rsync_Timeout: establece el tiempo límite para la recuperación mediante rsync
#(rango 1s - 1h)
#10s -- 10 segundos
#10min -- 10 minutos
#1h -- 1 horas
#por defecto : 0s (sin tiempo límite)
#o Rsync_Bwlimit: establece el límite del ancho de banda para rsync; KBytes/sec
#(rango 1KB - 10GB)
#10KB -- 10 KBytes/sec
#10MB -- 10 MBytes/sec
#1GB -- 1 GByte/sec
#Por defecto: 0KB (sin límite)
#Si la recuperación falla a menudo en grandes BD,
#pruebe cambiar el valor de ancho de banda.
#Se puede encontrar este valor ejecutando el siguiente comando:
#bwmeasure [dirección del nodo]
#o Pg_Dump_Path: Ruta del comando pg_dump
#o Ping_Path: Ruta del comando ping
#o When_Stand_Alone: En caso de fallar los servidores de replicación,
# se puede establecer 2 tipos de permisos en cuanto el acceso al # nodo de BD, solo lectura o
lectura/escritura, "read_only" o #"read_write".
#o Replication_Timeout: Tiempo límite para cada solicitud de replicación
#o Lifecheck_Timeout: Tiempo límite para la comprobación de respuesta
#o Lifecheck_Interval: Intervalo de tiempo entre las comprobaciones de respuesta
#(rango 1s - 1h)
#10s -- 10 segundos
#10min -- 10 minutos
#1h -- 1 horas
```

Partitional replicate control Information: en esta sección se establece la información que no se quiere replicar.

```
# set DB name and Table name to stop replication
# o DB_Name : Nombre de la BD
# o Table_Name : Nombre de la tabla
```

La configuración para los nodos de BD sería: (Fig. 16, Fig. 17)

Implantación y prueba de la solución propuesta

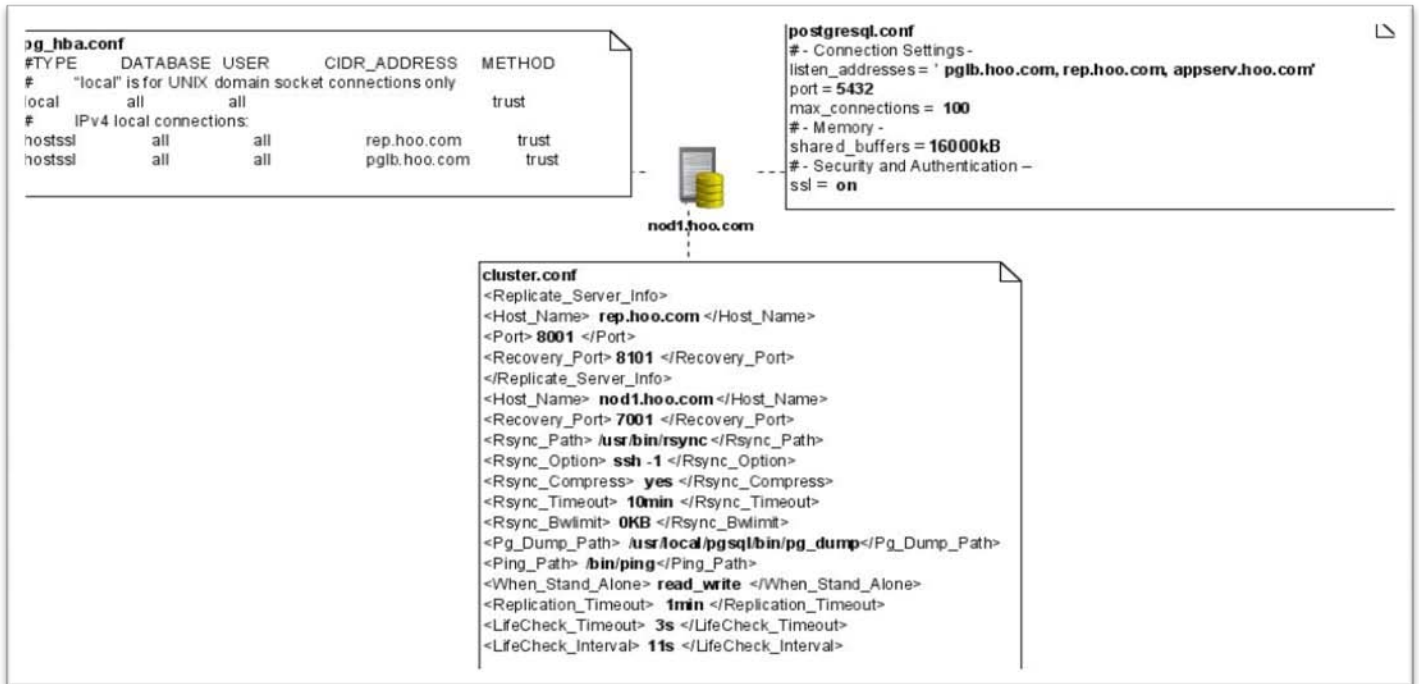


Figura 16. Configuración de archivos para Nodo 1.

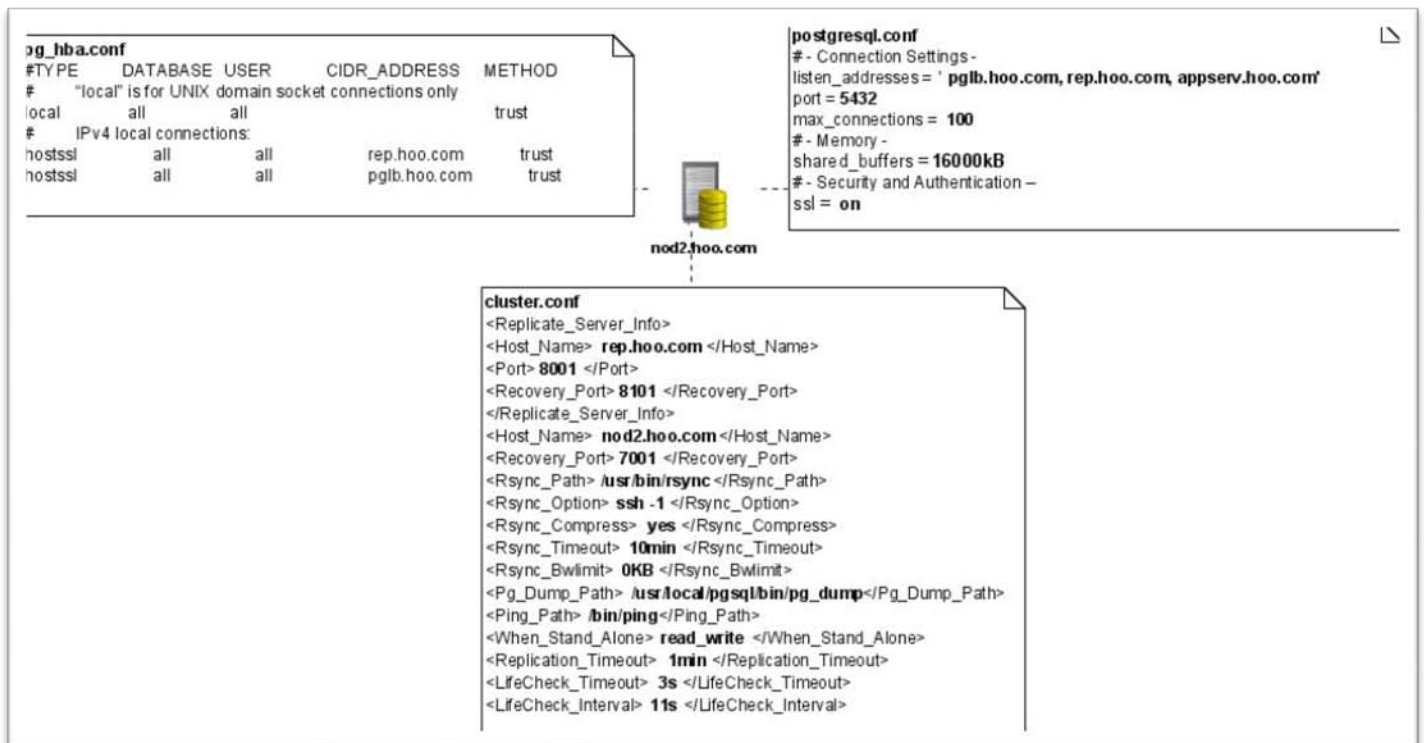


Figura 17. Configuración de archivos para Nodo 2.

3.2.2.4 Balanceador de Carga

En un servidor donde se aloja un Balanceador de Carga creado con la herramienta Cybercluster se necesitan editar dos archivos.

Archivo #1: pg_hba.conf

Archivo #2: pglb.conf

Este archivo por defecto no existe, es necesario crearlo a partir del archivo pglb.example.conf, mediante:

```
#cp /usr/local/pgsql/share/pglb.example.conf /usr/local/pgsql/share/pglb.conf
```

Ruta al archivo pglb.conf: **/usr/local/pgsql/share**

Archivo: **pglb.conf**

Este archivo controla cuales nodos son usados como servidores BD y que puerto se usa para establecer la conexión con los mismos. Además cuantas conexiones son permitidas para cada servidor de BD.

Este archivo está dividido en las siguientes secciones:

Cluster DB Server Information: en esta sección se establece la información del servidor de BD.

```
#o Host_Name : Dirección del nodo de BD
#o Port :Puerto para establecer la conexión
#o Max_Connection :Número máximo de conexiones al servidor de BD
```

Load Balancer Server Information: en esta sección se establece la información del servidor de balanceador de carga.

```
#o Host_Name : Nombre del servidor Balanceador de carga
#o Backend_Socket_Dir :Ruta del socket de dominio UNIX para respuesta
#o Receive_Port :Puerto de conexión para el cliente
#o Recovery_Port :Puerto de conexión para el proceso de recuperación
#o Max_Cluster_Num :Número máximo de servidores de BD que atiende
#o Use_Connection_Pooling : Usar agrupamiento de conexiones [yes/no]
#o Lifecheck_Timeout :Tiempo límite para la comprobación de respuesta
#o Lifecheck_Interval : Intervalo de tiempo entre las comprobaciones de respuesta
#o Connection_Life_Time : Tiempo de vida de la conexión.
#           (0 significa sin tiempo límite)
#(Rango 1s - 1h)
#10s  -- 10 segundos
#10min -- 10 minutos
#1h   -- 1 hora
```

A setup of a log files: en esta sección se establecen las opciones de los log.

```
#o File_Name :Ruta completa al nombre del Log
#o File_Size :Tamaño máximo para cada Log
# Especificar en número y unidades (K o M)
# 10  -- 10 Byte #10K  -- 10 Kbyte # 10M  -- 10 MBytes
#o Rotate :Tiempo de rotación
#           de especificar 0, versiones anteriores serán eliminadas.
```

Implantación y prueba de la solución propuesta

La opción `Use_Connection_Pooling` es recomendable establecerla a `yes` si se espera una concurrencia mayor a los 1000 usuarios o para anticipar concurrencias mayores a los picos de carga.

La configuración del servidor de balance de carga sería: (Fig. 18)

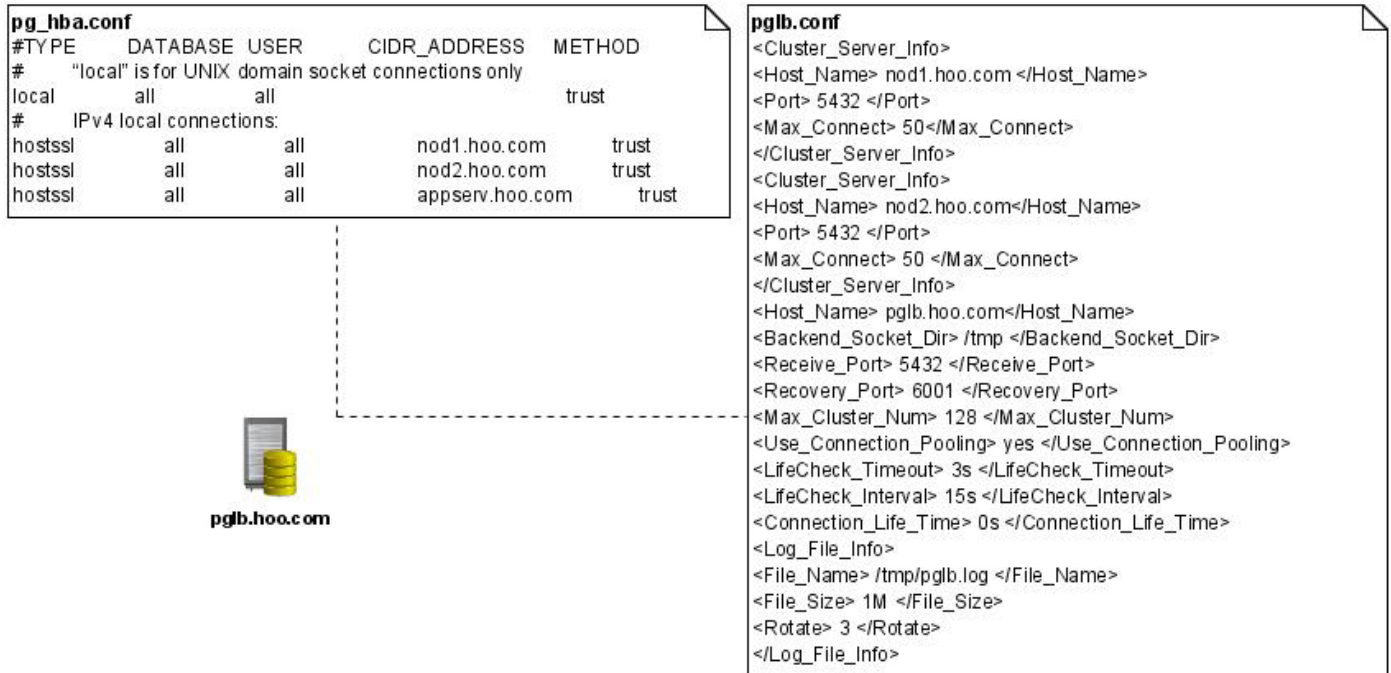


Figura 18. Configuración de archivos para el balanceador

3.2.2.5 Servidor de Replicación

En un servidor de Replicación creado con la herramienta Cybercluster se necesitan editar dos archivos.

Archivo #1: **pg_hba.conf**

Archivo #2: **pgreplcate.conf**

Este archivo por defecto no existe, es necesario crearlo a partir del archivo `pgreplcate.example.conf`, mediante:

```
#cp /usr/local/pgsql/share/pgreplcate.example.conf /usr/local/pgsql/share/pgreplcate.conf
```

Ruta al archivo: **/usr/local/pgsql/share**

Archivo **pgreplcate.conf**: Este archivo controla cuales nodos son usados como servidores BD y que puerto se usa para establecer la conexión con los mismos. Además cual puerto es usado para la solicitud de replicación desde los nodos de BD.

Este archivo está dividido en las siguientes secciones:

A setup of Cluster DB(s): en esta sección se establece la información de los nodos de BD que conforman el cluster.

```
#o Host_Name : Dirección del nodo de BD.
#o Port : Puerto para establecer la conexión.
#o Recovery_Port : Puerto de conexión para el proceso de recuperación.
```

A setup of Load Balance Server: en esta sección se establece la información del balanceador de carga.

```
#o Host_Name : La dirección del Balanceador de carga.
#o Recovery_Port : Puerto de conexión para el proceso de recuperación.
```

A setup of the cascade connection between replication servers: en esta sección se establecen las opciones de configuración para una conexión en cascada entre servidores de replicación.

```
#o Host_Name : La dirección del servidor de replicación maestro.
#o Port : Puerto para establecer la conexión
#o Recovery_Port : Puerto de conexión para el proceso de recuperación.
```

A setup of a replication server: en esta sección se establecen la información del servidor de replicación.

```
#o Host_Name : Dirección del servidor de replicación.
#o Replicate_Port : Puerto de conexión para la replicación.
#o Recovery_Port : Puerto de conexión para el proceso de recuperación
#o RLOG_Port : Puerto para el LOG de replicación
#o Response_mode : Modo de espera de respuesta
# normal -- Devuelve la respuesta de la BD que recibe la consulta
# reliable - Devuelve el resultado antes de esperar por una respuesta de todos
# los nodos de BD.
#o Use_Replication_Log : Usar LOG de replicación
# [yes/no]. por defecto : no
#o Replication_Timeout : Tiempo límite de replicación
#o Lifecheck_Timeout : Tiempo límite para la comprobación de respuesta
#o Lifecheck_Interval : Intervalo de tiempo entre las comprobaciones de respuesta
#(rango 1s - 1h)
#10s-- 10 segundos
#10min-- 10 minutos
#1h-- 1 hora
```

A setup of a log files: en esta sección se establecen las opciones de los log.

```
#o File_Name : Ruta y nombre completo al archivo log
#o File_Size : Tamaño máximo de cada archivo log
# especificar una unidad (K o M)
# 10-- 10 Byte
# 10K-- 10 KByte
# 10M-- 10 MByte
#o Rotate : Tiempo de rotación
# de especificar 0, versiones anteriores serán eliminadas.
```

La configuración para el servidor de replicación de replicación sería: (Fig. 19)

Implantación y prueba de la solución propuesta

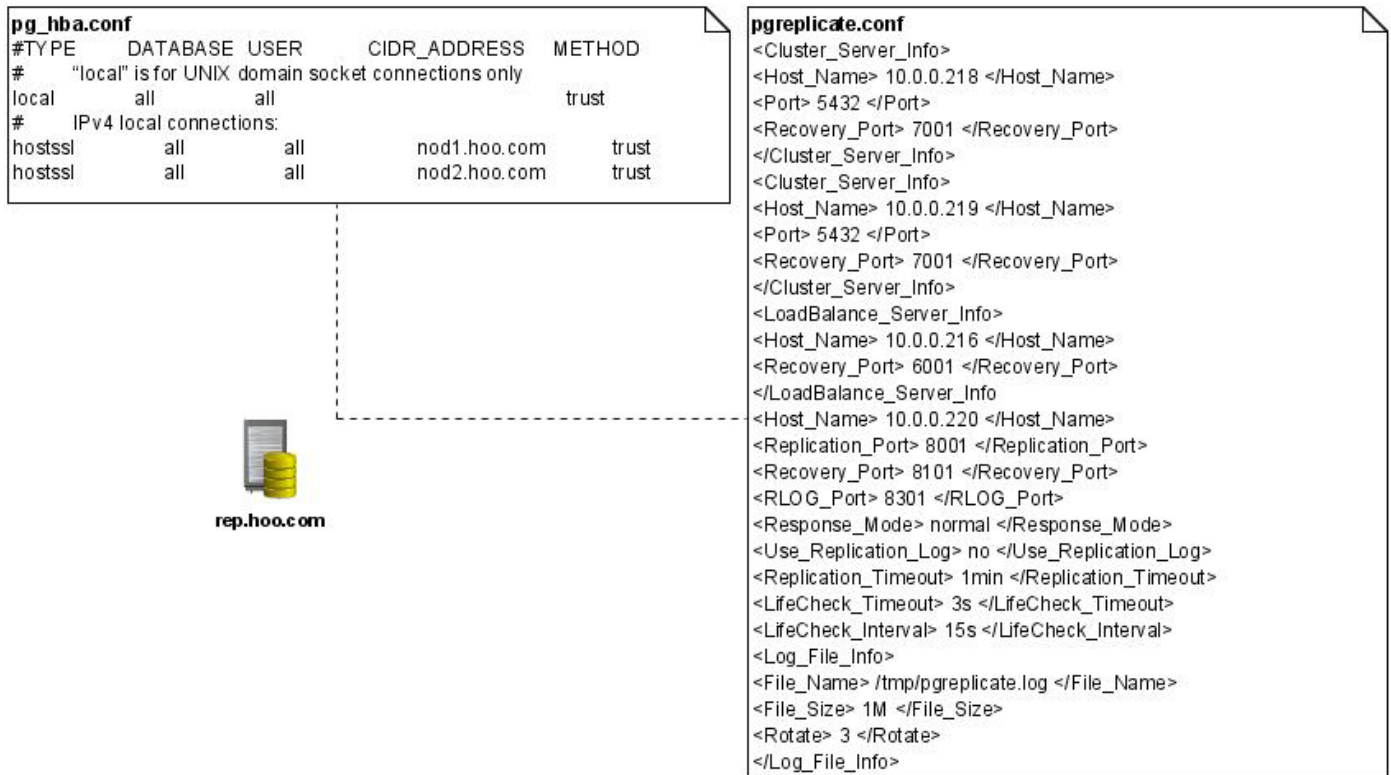


Figura 19. Configuración de archivos para el servidor de replicación.

3.2.2.6 Inicio, reinicio y parada de los servicios de Cybercluster

Cluster BD

Una vez configurados todos los nodos de BD estos pueden ser iniciados por el usuario postgres mediante el comando:

```
$/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data start
```

Para detener el servicio:

```
$/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data stop
```

Para reiniciar el servicio:

```
$/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data reload
```

Implantación y prueba de la solución propuesta

La ejecución de este comando permite al servicio iniciarse verificando si se realizaron cambios en los archivos de configuración.

De producirse algún fallo en alguno de los nodos por ejemplo algún fallo de hardware o error inesperado en el sistema operativo la información en este nodo debe sincronizarse nuevamente al cluster mediante las opciones de recuperación. La recuperación se ejecuta de diversas formas, estas son:

```
$/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data start -o "-R"
```

1. “-R” le dice al Cybercluster que inicie la recuperación basado en Rsync, este modo de recuperación es llamado método de “recuperación en frío” ya que asegura que el nodo desde el cual se sincronizarán los datos será detenido durante la recuperación. La ventaja de este método es que Rsync no copiará datos corruptos o incompletos que se estén actualizando en el momento de la sincronización.

```
$/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data start -o "-u"
```

2. “-u” esta opción fuerza la recuperación basada en Rsync sin realizar salvadas de seguridad del directorio original, esto aumenta considerablemente la velocidad de recuperación pero solo debe ser usado si Rsync está correctamente configurado, por ejemplo que el acceso vía ssh¹³ sea sin contraseña. Este es también un método de “recuperación en frío”.

```
$/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data start -o "-U"
```

3. “-U” provee la opción de usar la recuperación basada en pg_dump. En dependencia del tamaño de la BD este puede demorar largo tiempo la recuperación. Sin embargo, al contrario de los otros métodos de recuperación, este es un método de “recuperación en caliente”, esto significa que el nodo de BD desde el cual se sincronizarán los datos no será detenido durante la recuperación ya que pg_dump se comporta como cualquier cliente de BD. Este es el único método de recuperación que puede ser usado para replicar si Cybercluster se encuentra sobre varias arquitecturas, por ejemplo, en diferentes arquitecturas de procesador.

¹³ SSH (Secure SHell) -intérprete de comandos seguro

Balanceador de Carga

Para iniciar el servicio:

```
# /usr/local/pgsql/bin/pglb -D /usr/local/pgsql/share
```

Para detener el servicio:

```
# /usr/local/pgsql/bin/pglb -D /usr/local/pgsql/share stop
```

Uso: pglb [-D ruta a los archivos de configuración] [-W ruta a los archivos de trabajo] [-n][-v][-h][stop]

-l: imprime los registros de error en los archivos log.

-n: no iniciar en modo demonio.

-v: modo de depuración. Se necesita la opción '-n'

-h: imprime esta ayuda.

stop: detiene pglb

restart: reinicia el servicio

(Ruta por defecto al archivo de configuración: /usr/local/pgsql/share/pglb.conf)

Servidor de la Replicación

Para iniciar el servicio:

```
# /usr/local/pgsql/bin/pgreplicate -D /usr/local/pgsql/share
```

Para detener el servicio:

```
# /usr/local/pgsql/bin/pgreplicate -D /usr/local/pgsql/share stop
```

Uso: pgreplicate [-D ruta a los archivos de configuración] [-W ruta a los archivos de trabajo] [-w tiempo de espera antes de iniciar] [-U usuario de acceso] [-n][-v][-h][stop]

-l: imprime los registros de error en los archivos log.

-n: no iniciar en modo demonio.

-v: modo de depuración. Se necesita la opción '-n'

-h: imprime esta ayuda.

stop: detiene pglb

(Ruta por defecto al archivo de configuración: /usr/local/pgsql/share/pgreplicate.conf)

3.3 Monitoreo

Una vez iniciado el servidor de replicación por defecto este escucha por el puerto 8401, esto permite monitorear el cluster de BD. Este puerto puede ser cambiado en el archivo `pgreplicate.conf` en la opción `<RPLMonitor_Port>####</RPLMonitor_Port>`

Para monitorear el cluster se hará necesaria la ejecución del comando:

```
# /usr/local/pgsql/bin/pgrplmon <host> <puerto>
```

Aplicado al ejemplo de composición del sistema:

```
# /usr/local/pgsql/bin/pgrplmon rep.hoo.com 8401
```

Se obtiene la siguiente salida:

```
RPLMonitor v1.0
DB server #1
Hostname: nod1
Resolved host: 192.168.0.2
Port: 5432
# of started transactions: 0
# of COMMITed transactions: 0
# of ROLLBACKed transactions: 0
# of INSERT statements: 3
# of UPDATE statements: 1
# of DELETE statements: 0
# of DDL statements: 3
DB server #2
Hostname: nod2
Resolved host: 192.168.0.3
Port: 5432
# of started transactions: 0
# of COMMITed transactions: 0
# of ROLLBACKed transactions: 0
# of INSERT statements: 0
# of UPDATE statements: 1
# of DELETE statements: 0
# of DDL statements: 0
2 active cluster members were found
```

Para validar el correcto funcionamiento de la solución propuesta es necesario el montaje de un entorno de desarrollo similar al definido anteriormente, al cual se le realizarán pruebas, las cuales proporcionan datos que permiten obtener una visión del futuro comportamiento del sistema. Para la realización de las pruebas

es imprescindible conocer los procedimientos y conceptos necesarios para la correcta ejecución de las mismas.

3.4 Pruebas

Las pruebas es una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. (12)

3.4.1 Pruebas de integridad de datos y BD

La integridad de una base de datos significa que, la BD ó los programas que generaron su contenido, incorporen métodos que aseguren que el contenido de los datos del sistema no se corrompa así como las reglas del negocio. Tiene como objetivo asegurar que los métodos de acceso y los procesos funcionen apropiadamente y sin corrupción de datos.

Las pruebas de integridad de datos se realizan invocando métodos de acceso a la BD, intentando con datos válidos e inválidos. También es necesario inspeccionar la BD para asegurar que ha sido poblada como se esperaba, que todos los eventos ocurran apropiadamente, o revisar los datos retornados para asegurar que se obtuvieron los datos correctos.

Criterio de cumplimiento

Todos los métodos de acceso a la base de datos y procesos funcionan como fueron diseñados y sin corrupción de datos. Para el caso de las BDD se hace necesario verificar que los datos sean consistentes en las BD de cada nodo.

3.4.2 Pruebas de desempeño

Es un tipo de prueba con el objetivo de determinar la respuesta, el rendimiento, la confiabilidad y la escalabilidad de un sistema bajo una carga de trabajo determinada. Las pruebas de desempeño se llevan a cabo para:

- Evaluar la disposición para la producción
- Evaluar a partir de los criterios de desempeño

Implantación y prueba de la solución propuesta

- Comparar las características de desempeño de múltiples sistemas o configuraciones
- Identificar la fuente de los problemas de desempeño
- Brindar soporte para el ajuste del sistema en función del desempeño
- Encontrar niveles de rendimiento

Las pruebas de desempeño constan de las siguientes actividades:

Actividad 1. Identificar el entorno de pruebas. Identificar el entorno físico de pruebas y el entorno de producción así como las herramientas y recursos disponibles a usar por el equipo de prueba. El entorno físico incluye hardware, software, y las configuraciones de red. Una vez lograda la comprensión total del entorno de pruebas esto permitirá una mayor eficiencia en el diseño y planificación de las pruebas lo que ayudará a identificar los objetivos de las pruebas tempranamente. En algunas situaciones este proceso debe ser revisado periódicamente a través del ciclo de vida del proyecto.

Actividad 2. Identificar criterios de aceptación de desempeño. Identificar el tiempo de respuesta, el rendimiento, y los recursos utilizados esperados, así como sus limitaciones. En general, el tiempo de respuesta es una preocupación de los usuarios, el rendimiento es una preocupación del negocio y la utilización de los recursos es una preocupación del sistema. Adicionalmente, identificar criterios de éxito del proyecto que no deben ser capturados por estas metas y limitaciones; por ejemplo, utilizando pruebas de desempeño para evaluar cual combinación de los parámetros de configuración obteniendo las características de desempeño más eficientes.

Actividad 3. Planificar y diseñar las pruebas: Identificar escenarios claves, determinar la variabilidad entre los usuarios representativos y como simular dicha variabilidad, definir los datos de las pruebas, y establecer las métricas a recolectar. Consolidar esta información en uno o más modelos de uso del sistema a ser implementado, ejecutado y analizado.

Actividad 4. Configurar el entorno de pruebas. Preparar el entorno de pruebas, herramientas, y recursos necesarios para ejecutar cada estrategia, así como que sus características y componentes estén disponibles para las pruebas. Asegurar que el entorno de prueba esté preparado para monitorear los recursos necesarios.

Actividad 5. Implementar el diseño de las pruebas. Desarrollar las pruebas de desempeño de acuerdo con el diseño de las pruebas.

Actividad 6. Ejecutar las pruebas. Ejecutar y monitorear las pruebas. Validar las pruebas, los datos de las pruebas, y la recolección de los resultados. Ejecutar las pruebas validadas para el análisis mientras se monitorean las pruebas y el entorno de pruebas.

Actividad 7. Analizar resultados, reportes y volver a ejecutar las pruebas. Consolidar y compartir los datos resultantes. Analizar los datos individualmente y en equipo. Darle nueva prioridad a las pruebas no ejecutadas y ejecutarlas nuevamente cuantas veces se necesite. Cuando los resultados obtenidos expresados en métricas coinciden con los criterios de aceptación, cuando ninguno de los objetivos han sido violados, y toda la información deseada ha sido recopilada, entonces se han concluido las pruebas para un escenario particular con una configuración específica.

3.4.2.1 Pruebas de carga

El objetivo de las pruebas de carga es verificar el comportamiento de una aplicación bajo condiciones normales de carga y durante los picos de carga, que es cuando la aplicación llega al límite de su funcionamiento normal. Este tipo de prueba es realizada para verificar que la aplicación cumpla con los criterios de aceptación o compromisos de rendimiento. Las pruebas de carga te permiten además medir los tiempos de respuesta, intervalos de rendimiento, los niveles de utilización de los recursos y la identificación del límite de funcionamiento, asumiendo que el punto de ruptura esté por encima del pico de carga. Dentro de las pruebas de carga se realizan además las pruebas de resistencia, una prueba de resistencia es un tipo de prueba que se concentra en determinar o validar las características de desempeño del producto cuando se encuentra bajo modelos de cargas de trabajo y un volumen de carga anticipadas a las condiciones de producción durante un largo período de tiempo. Las pruebas de resistencia son usadas para calcular métricas como MTBF¹⁴ (Mean Time Between Failure) y MTTF¹⁵ (Mean Time To Failure).

¹⁴ Tiempo entre una falla y otra

¹⁵ Tiempo en el cual el sistema falla

Beneficios:

- Determina el rendimiento requerido soportado para anticipar los picos de carga en producción.
- Determina el entorno de hardware adecuado.
- Evalúa el nivel de optimización del balanceador de carga, si está presente.
- Detecta elementos que afecten la concurrencia.
- Detecta errores funcionales bajo carga.
- Recopila datos que ayuden en la escalabilidad y con propósitos para planes de crecimiento.
- Ayuda a determinar cuántos usuarios puede manejar el sistema antes de que el rendimiento sea comprometido.
- Ayuda a determinar cuanta carga puede manejar el hardware antes de que la utilización de los recursos llegue al límite.

No está diseñada para concentrarse en la velocidad de respuesta, los resultados solo deben ser utilizados para compararlos con otras pruebas de carga.

3.4.2.2 Pruebas de estrés

Para determinar o validar el comportamiento de una aplicación cuando es sometida a condiciones de carga superiores a los límites aceptables, sobre condiciones normales y sobre los picos de carga. La meta de las pruebas de estrés es revelar los errores que solo aparecen bajo condiciones extremas de carga. Estos errores incluyen cuestiones de sincronización y fallas de memoria. Las pruebas de estrés ayudan a identificar los puntos débiles del sistema y su comportamiento bajo estas condiciones. Las pruebas de picos es un tipo de prueba de desempeño dentro de las pruebas de estrés cuyo objetivo es determinar o validar las características de desempeño del producto bajo prueba sujeto a modelos de carga de trabajo y volúmenes de carga que repetidamente aumentan más allá de las condiciones esperadas en operaciones de producción, se realizan por cortos intervalos de tiempo.

Beneficios:

- Determina si los datos se corrompen debido a la saturación en consecuencia del nivel de estrés del sistema.
- Provee un estimado de cuán lejos después del límite de carga la aplicación puede soportar la carga antes de ocasionar fallas o errores que ocasionen lentitud al sistema.

Implantación y prueba de la solución propuesta

- Permite establecer actividades de monitoreo a la aplicación basado en reglas que alerten acerca de fallas inminentes.
- Asegura que las vulnerabilidades de seguridad no aparezcan bajo condiciones de estrés.
- Ayuda a determinar qué tipos de fallas son las más comunes para realizar un plan que ayude a mitigarlas.

Debido a que las pruebas de estrés son para condiciones irreales por diseño, algunos clientes no tienen en cuenta sus resultados. En ocasiones es muy difícil saber cuánto estrés es necesario aplicar. Los resultados de las pruebas no están aislados de las fallas externas al entorno de prueba, como fallas de red.

3.4.2.3 Pruebas de volumen

Determina cuantos usuarios y/o transacciones puede realizar el sistema y aún así seguir cumpliendo con los objetivos de desempeño. Las pruebas de volumen se llevan a cabo en relación con la planificación de la capacidad, que se utiliza para planificar un crecimiento futuro, tales como un mayor número usuarios o un mayor volumen de datos. Por ejemplo, para acomodar las cargas futuras, por lo que se necesita saber cuántos recursos adicionales (como la capacidad de procesador, memoria, capacidad de disco, o ancho de banda de red) son necesarios para apoyar los niveles de uso futuro, a fin de determinar si se debe aumentar o disminuir de acuerdo a los resultados.

Beneficios

- Provee información acerca de cómo la carga de trabajo puede ser manejada para cumplir los requerimientos del negocio.
- Provee los datos actuales que los planificadores pueden usar para validar o mejorar sus modelos o predicciones.
- Permite la ejecución de varias pruebas para comparar los modelos de planificación.
- Determina el uso y la capacidad del sistema actual, así como las tendencias que ayuden en la planificación de la capacidad.

Las pruebas de volumen son complejas para crear. No todos los aspectos de las pruebas de volumen pueden ser validados mediante pruebas. (13)

3.5 Herramientas

La realización de las pruebas se llevaran a cabo utilizando las herramientas siguientes:

Apache JMeter: Es una herramienta implementada en Java la cual fue diseñada para hacer pruebas de carga en aplicaciones Web, pero se expandió a otras funciones. Típicamente es usada para medir desempeño y realizar pruebas de carga, es una herramienta Open Source. Soporta HTTP, FTP, SOAP/XMLRPC, JDBC. Es una herramienta GUI¹⁶. Posee muchos GUI Listeners, los cuales son usados para capturar la ejecución de la prueba y mostrar los resultados basados en métricas.

Métricas:

Media (Average): matemáticamente se refiere a al promedio del juego de datos. Es una medida que por sí sola dice muy poco sobre los resultados, pero nos da una idea de cuál es el comportamiento general de la aplicación.

Mediana (Median): Simplemente es el valor medio del juego de datos cuando estos son ordenados de menor a mayor. En los casos en que la cantidad de datos es par, se toman los dos valores centrales y se saca el promedio de estos.

Desviación Estándar (Standard Deviation): es una medida de dispersión para variables de razón y de intervalo, de gran utilidad en la estadística descriptiva. Es una medida (cuadrática) que informa de la media de distancias que tienen los datos respecto de su media aritmética, expresada en las mismas unidades que la variable. Una norma utilizada para esta métrica es: “Los datos con una desviación estándar superior al 68% de su media debe ser tratada como sospechosa. Si es exactamente ese valor, el juego de datos no tiene una distribución normal”.

Rendimiento (Throughput): Es la tasa promedio de mensajes entregados satisfactoriamente. Comúnmente se mide en peticiones/segundos, minutos u horas.

Pgcbench:

Es una herramienta simple para la ejecución de pruebas en detalle o benchmarks sobre PostgreSQL. Este ejecuta en la misma secuencia sentencias SQL una y otra vez, sobre múltiples sesiones concurrentes de bases de datos y entonces calcula el promedio de transacciones por segundo. Por defecto Pgcbench realiza pruebas en un escenario involucrando 5 sentencias mixtas SELECT, UPDATE e INSERT por

¹⁶ Graphic User Interface: Interfaz gráfica de usuario

Implantación y prueba de la solución propuesta

segundo las cuales se envían como una sola transacción. Sin embargo es sencillo realizar otros casos de pruebas al definir scripts propios.

Permite la creación de bases de datos de diferentes tamaños o complejidad sobre las que pueden ejecutarse pruebas, esto se logra definiendo un factor de escala, por ejemplo para una BD de mediana complejidad especificándole un factor de escala –s 100 generará una BD con aproximadamente diez millones de tuplas.

3.6 Entorno de prueba

Para la ejecución de las pruebas se utilizó como entorno el proyecto Informatización de la Universidad de las Ciencias Informáticas (UCI), el cual presenta una situación problemática semejante a la de la Intranet de Corporativa de PDVSA y el Portal en Internet. A partir de la propuesta de solución definida se decide aplicar para el proyecto de Informatización un entorno de réplica único en el cual estarán todas las BD de los servicios que brindan, tales como Intranet, Teleformación, Akademos, entre otros. El sistema de réplica está compuesto por 5 servidores HP Proliant BL 460c G1 con las siguientes características:

- Memoria RAM: 8Gb
- CPU: 2 Quad Core 2.00 GHz
- Sistema Operativo: Centos 5.2, 2.6.18-128.1.6.el5.centos.plus 64 bits
- Configuración de Red: LAN 1Gb

Implantación y prueba de la solución propuesta

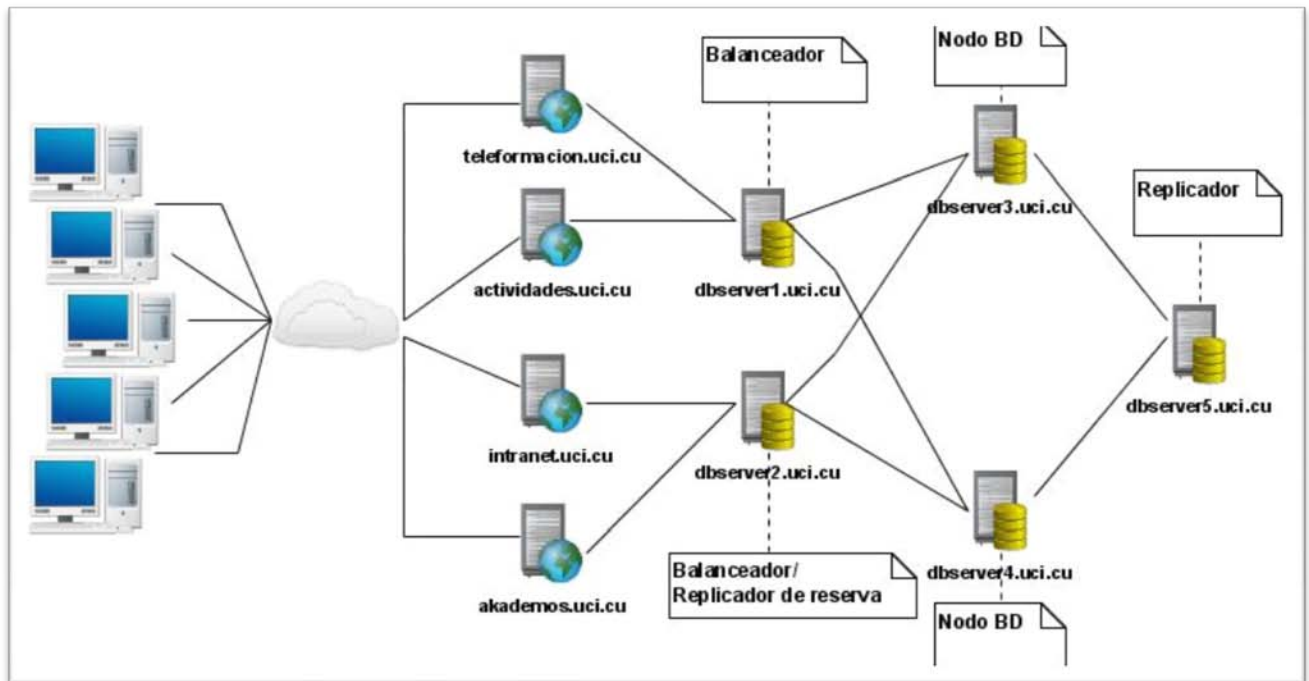


Figura 20. Entorno de réplica UCI.

3.6.1 Criterios de aceptación

Para determinar los criterios de aceptación para el entorno de pruebas utilizado se recolectaron algunos datos:

- ✓ Cantidad de computadoras: 7000.
- ✓ Usuarios potenciales: 17000
- ✓ Picos de carga: 300 usuarios simultáneos para el sitio de la Intranet, 500 usuarios simultáneos para Teleformación.
- ✓ Los tiempos de respuesta para el usuario utilizando una aplicación de acceso concurrente a la BD como la plataforma Moodle están estimados en un tiempo de hasta 20 segundos.
- ✓ Carga Total estimada: 600 usuarios concurrentes (actual).

Según los clientes de las diferentes áreas de la universidad los criterios de aceptación quedan definidos de la siguiente manera:

El sistema debe ser capaz de soportar una carga durante las operaciones normales de 500 usuarios concurrentes a las BD, durante los picos de carga el sistema debe ser capaz de soportar una carga de

Implantación y prueba de la solución propuesta

600 usuarios concurrentes a las BD. En cuanto a los tiempos de respuesta no deben exceder de los cinco segundos para una solicitud en los picos de carga. El sistema debe ser capaz de realizar más de 500 transacciones por segundo para los picos de carga.

Las pruebas se realizarán con varias concurrencias de usuarios simulando distintas cargas en cuanto a transacciones y para diferentes configuraciones del sistema con el fin de seleccionar la configuración más adecuada así como para establecer los límites de carga que el sistema es capaz de soportar para cada configuración.

3.6.2 Ejecución de pruebas de carga

Se simularon pruebas para cargas normales para diferentes configuraciones (ver Anexo 1) con diversas concurrencias de usuarios realizando solicitudes al sistema durante intervalos de tiempo cortos y a BD de diferentes tamaños. Las pruebas de carga fueron ejecutadas utilizando la herramienta Pgbench tanto para el sistema de réplica como para un solo nodo de BD de las cuales se observaron los siguientes resultados teniendo en cuenta las siguientes métricas.

Métricas	
tps	transacciones por segundo
tps (media)	media de las transacciones por segundo
tiempo (s)	tiempo medio en el cual terminaron las transacciones

Tabla 3 Métricas de las pruebas.

300 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo(s)	
300	70	MIXTAS	4498 MB	21,965	26,363	26,884	25,071	2,81	Cluster
	100			25,833	26,206	25,724	25,921	3,86	
	200			26,236	25,966	26,241	26,148	7,64	
	300			26,211	25,828	25,312	25,784	11,68	
	500			24,99	24,474	26,033	25,166	19,87	
	600			25,512	25,169	25,485	25,389	23,62	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
300	70	MIXTAS	4498 MB	Falla del sistema			0	0	Nodo
	100						0		
	200						0		
	300						0		

Implantación y prueba de la solución propuesta

300 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
300	70	SELECT	4498 MB	21,934	22,175	22,124	22,078	3,17	Cluster
	100			31,745	30,654	31,726	31,375	3,19	
	200			63,55	63,397	63,385	63,444	3,15	
	300			95,048	95,139	93,409	94,532	3,17	
	500			158,35	153,418	158,773	156,847	3,18	
	600			190,298	190,396	187,994	189,563	3,16	
	1000			317,345	314,417	315,251	315,671	3,16	
	1500			471,017	472,622	469,082	470,907	3,17	
	2000			628,751	601,913	629,348	620,004	3,22	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
300	70	SELECT	4498 MB	21,785	21,633	21,532	21,65	3,23	Nodo
	100			30,717	31,008	30,861	30,862	3,24	
	200			61,311	61,462	61,529	61,434	3,25	
	300			88,3168	88,644	89,149	88,703	3,38	
	500			146,658	150,014	150,216	148,963	3,44	
	600			165,685	170,686	168,255	168,209	3,57	
	1000			264,197	265,886	264,726	264,936	3,78	
	1500			361,777	358,644	370,048	363,49	4,13	
	2000			451,053	461,435	458,853	457,114	4,37	

Tabla 4. Resultados de las pruebas de carga para la configuración 1

300 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo(s)	
300	70	MIXTAS	4498 MB	11,616	12,091	12,249	11,985	5,84	Cluster
	100			12,759	13,432	12,395	12,862	7,77	
	200			13,367	13,475	13,784	13,542	14,76	
	300			14,438	15,215	14,902	14,852	20,2	
	500			14,136	13,973	14,637	14,249	35,1	
	600			13,297	13,471	13,009	13,259	45,26	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
300	70	MIXTAS	4498 MB	Falla del sistema			0	Nodo	
	100						0		
	200						0		

Implantación y prueba de la solución propuesta

300 Usuarios								
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)
	300						0	
	500						0	
	600						0	
300	70	SELECT	4498 MB	22,221	22,275	22,228	22,241	3,14
	100			31,604	31,709	31,745	31,686	3,15
	200			62,955	63,33	63,017	63,101	3,17
	300			94,114	86,907	89,057	90,026	3,33
	500			149,35	150,98	158,73	153,019	3,26
	600			189,34	188,58	186,67	188,197	3,18
	1000			313,03	316,21	310,63	313,289	3,19
	1500			469,53	469,54	473,1	470,722	3,19
	2000			621,22	623,07	625,01	623,101	3,21
300	70	SELECT	4498 MB	21,733	21,738	21,774	21,748	3,22
	100			31,093	31,054	31,071	31,073	3,21
	200			62,043	61,944	59,504	61,164	3,27
	300			91,166	90,158	89,727	90,35	3,31
	500			149,22	152,55	148,62	150,132	3,33
	600			171,93	175,69	176,54	174,716	3,43
	1000			276,19	275,07	280,41	277,221	3,6
	1500			373,75	373,15	386,99	377,965	3,97
	2000			482,77	495,98	494,03	490,926	4,07

Tabla 5. Resultados de las pruebas de carga para la configuración 2

300 Usuarios								
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo(s)
300	70	MIXTAS	4498 MB	15,563	15,006	15,237	15,269	4,585
	100			14,717	15,381	15,429	15,176	6,505
	200			15,713	15,658	15,749	15,707	12,71
	300			16,638	16,66	16,296	16,531	15,61
	500			14,883	14,562	15,425	14,957	33,33
	600			14,701	16,052	15,445	15,399	37,15
300	70	MIXTAS	4498 MB	16,247	16,403	16,524	16,391	4,25
	100			16,684	17,071	15,748	16,501	6,01

Implantación y prueba de la solución propuesta

	200			17,904	18,668	18,269	18,28	10,9	
	300			18,317	18,617	18,938	18,624	16,12	
	500			17,36	17,825	16,935	17,373	28,5	
	600			16,647	16,803	17,113	16,854	35,25	
300 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
300	70	SELECT	4498 MB	21,929	22,208	22,219	22,119	3,15	Cluster
	100			31,727	31,78	31,816	31,774	3,14	
	200			63,189	63,383	63,119	63,23	3,16	
	300			94,805	94,332	94,801	94,646	3,18	
	500			158,29	158,21	156,62	157,704	3,17	
	600			187,9	186,92	186,53	187,118	3,19	
	1000			296,42	315,04	315,93	309,129	3,23	
	1500			451,49	464,67	465,33	460,498	3,22	
	2000			578,92	626,32	623,22	609,485	3,28	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
300	70	SELECT	4498 MB	21,774	21,788	21,811	21,791	3,2	Nodo
	100			31,147	31,238	31,06	31,148	3,21	
	200			61,005	59,316	60,983	60,435	3,3	
	300			88,268	90,236	88,388	88,964	3,36	
	500			147,8	145,6	150,83	148,076	3,37	
	600			171,34	174,62	171,78	172,58	3,47	
	1000			267,08	263,96	263,67	264,902	3,77	
	1500			364,94	360,82	343,45	356,402	4,2	
	2000			458,74	469,93	459,25	462,641	4,32	

Tabla 6. Resultados de las pruebas de carga para la configuración 3.

De los resultados obtenidos se concluye que para las pruebas de carga realizadas a las 3 configuraciones el sistema es capaz de soportar cargas normales, logra completar de manera efectiva sin errores en el procesamiento de los datos más de 30000 consultas por minuto, para consultas de diversa complejidad. Se demuestra la incapacidad de la configuración 1 para un servidor de BD centralizado de satisfacer la demanda actual. Además se evidencia que para concurrencias bajas el comportamiento de los servidores es similar tanto para un sistema distribuido como para uno centralizado por lo que se demuestra que para sistemas con bajas concurrencias no es factible el uso de un sistema distribuido. El uso de los recursos por parte del sistema para las diferentes configuraciones fue mínimo. (Ver Anexo 2 Pruebas de carga)

Implantación y prueba de la solución propuesta

3.6.2.1 Picos de carga

Se realizaron además pruebas de carga simulando el comportamiento del sistema en los picos de carga, de la ejecución de estas se obtuvieron los siguientes resultados.

450 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo(s)	
450	70	MIXTAS	6237 MB	10,274	11,317	12,498	11,363	6,16	Cluster
	100			Falla del sistema			0	0	
	200			Falla del sistema			0		
	300			Falla del sistema			0		
	500			Falla del sistema			0		
	600			Falla del sistema			0		
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	MIXTAS	6237 MB	Falla del sistema			0	0	Nodo
	100			Falla del sistema			0		
	200			Falla del sistema			0		
	300			Falla del sistema			0		
	500			Falla del sistema			0		
	600			Falla del sistema			0		
450 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	SELECT	6237 MB	20,647	21,096	20,843	20,862	3,36	Cluster
	100			29,71	29,707	29,44	29,619	3,38	
	200			57,383	58,932	58,271	58,195	3,41	
	300			87,306	87,331	88,585	87,741	3,44	
	500			143,354	146,522	145,695	145,19	3,44	
	600			173,857	175,179	177,867	175,634	3,41	
	1000			280,479	283,056	276,79	280,108	3,57	
	1500			Falla del sistema			0		
	2000			Falla del sistema			0		
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	SELECT	6237 MB	Falla del sistema			0	0	Nodo
	100			Falla del sistema			0		
	200			Falla del sistema			0		
	300			Falla del sistema			0		
	500			Falla del sistema			0		

Implantación y prueba de la solución propuesta

	600				0		
	1000				0		
	1500				0		
	2000				0		

Tabla 7. Resultados de las pruebas de carga pico para la configuración 1.

450 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	MIXTAS	6732 MB	18,533	19,351	19,94	19,275	3,63	Cluster
	100			20,997	21,685	21,755	21,479	4,62	
	200			22,812	22,532	22,842	22,729	8,79	
	300			22,488	22,857	22,026	22,457	13,36	
	500			22,118	22,594	22,163	22,292	22,42	
	600			22,367	23,158	21,462	22,329	26,9	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	MIXTAS	6732 MB	Falla del sistema			0	0	Nodo
	100						0		
	200						0		
	300						0		
	500						0		
	600						0		
450 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	SELECT	6732 MB	20,603	20,763	20,533	20,633	3,38	Cluster
	100			29,24	29,609	29,61	29,486	3,38	
	200			58,713	58,96	58,647	58,773	3,4	
	300			88,254	88,441	88,445	88,38	3,39	
	500			147,67	146,31	147,64	147,207	3,39	
	600			176,72	175,41	177,41	176,511	3,39	
	1000			294,81	292,85	294,02	293,894	3,4	
	1500			440,13	439,36	440,8	440,099	3,41	
	2000			587,64	585,94	585	586,194	3,41	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	SELECT	6732 MB	20,592	20,636	20,575	20,601	3,39	Nodo
	100			29,478	29,525	29,449	29,484	3,39	
	200			58,748	58,917	58,935	58,867	3,4	
	300			88,793	88,322	88,066	88,394	3,4	

Implantación y prueba de la solución propuesta

	500			146,96	147,05	147,07	147,028	3,4	
	600			176,27	62,264	62,263	100,265	7,26	
	1000			108,71	108,83	108,46	108,667	9,2	
	1500			162,96	163,04	163,28	163,093	9,2	
	2000			217,29	217,4	217,36	217,349	9,2	

Tabla 8. Resultados de las pruebas de carga pico para la configuración 2.

450 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	MIXTAS	6732 MB	19,458	20,575	19,852	19,962	3,48	Cluster
	100			19,546	19,848	19,176	19,523	5,11	
	200			21,028	21,576	22,047	21,55	9,28	
	300			22,993	23,308	22,407	22,903	13,09	
	500			22,641	21,725	21,066	21,811	22,94	
	600			20,295	22,451	21,079	21,275	28,24	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	MIXTAS	6732 MB	9,09	10,573	11,396	10,353	6,82	Nodo
	100			12,126	12,352	12,585	12,354	8,08	
	200			12,899	12,953	12,914	12,922	15,47	
	300			12,948	13,232	13,223	13,134	22,84	
	500			12,543	12,186	12,116	12,282	40,72	
	600			13,027	14,121	12,87	13,339	45,05	
450 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	SELECT	6732 MB	20,681	20,668	20,658	20,669	3,38	Cluster
	100			29,505	29,682	29,47	29,552	3,38	
	200			58,791	59,216	59,189	59,065	3,38	
	300			88,052	87,963	88,913	88,309	3,4	
	500			147,07	148,08	146,42	147,188	3,41	
	600			176,89	178,22	176,95	177,353	3,38	
	1000			292,93	295,32	292	293,416	3,4	
	1500			434,81	439,64	441,32	438,588	3,41	
	2000			579,9	585,11	578,01	581,004	3,44	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
450	70	SELECT	6732 MB	7,459	7,46	7,464	7,461	9,38	Nodo
	100			10,653	10,65	10,653	10,652	9,39	
	200			21,761	21,306	21,314	21,46	9,32	

Implantación y prueba de la solución propuesta

	300			32,617	32,653	31,965	32,412	9,26	
	500			53,104	78,188	53,259	61,517	8,4	
	600			63,833	65,264	63,87	64,322	9,32	
	1000			106,43	108,76	106,48	107,222	9,26	
	1500			234,76	163,14	159,46	185,784	8,31	
	2000			217,23	216,93	217,66	217,271	9,21	

Tabla 9. Resultados de las pruebas de carga pico para la configuración 3

Como resultado de las pruebas para los picos de carga se observa que para las configuraciones 1 y 2 el procesamiento de las transacciones de escritura provoca fallas tanto al sistema como al servidor centralizado. Para la configuración 3 se observan mejores resultados en cuanto a que logran procesar todas las transacciones tanto de escritura como de lectura aunque se registró un aumento en los tiempos de respuesta para las transacciones mixtas. Además de las pruebas para los picos de carga se realizó al sistema pruebas de resistencia desde la aplicación Teleformación, esta funcionó óptimamente mejorando su rendimiento en cuanto a tiempo de respuesta durante las operaciones normales de carga, se realizaron exámenes durante dos días, en el primer día para una carga de 300 usuarios concurrentes realizando más de 200 transacciones tanto de lectura como escritura arrojando resultados satisfactorios en cuanto a los tiempos de respuesta que en comparación a resultados anteriores fueron mejores. (Ver Anexo 2 Picos de carga)

3.6.3 Ejecución de pruebas de estrés

Se simularon pruebas de estrés a las distintas configuraciones de las cuales se registran los siguientes resultados.

600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo(s)	
600	70	MIXTAS	8975 MB	Falla del sistema			0	0	Cluster
	100						0		
	200						0		
	300						0		
	500						0		
	600						0		
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	MIXTAS	8975 MB	Falla del sistema			0	0	Nodo

Implantación y prueba de la solución propuesta

600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
	100						0		
	200						0		
	300						0		
	500						0		
	600						0		
600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	SELECT	8975 MB	Falla del sistema			0	0	Cluster
	100						0		
	200						0		
	300						0		
	500						0		
	600						0		
	1000						0		
	1500						0		
	2000						0		
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	SELECT	8975 MB	Falla del sistema			0	0	Nodo
	100						0		
	200						0		
	300						0		
	500						0		
	600						0		
	1000						0		
	1500						0		
	2000						0		

Tabla 10. Resultados de las pruebas de estrés para la configuración 1.

600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	MIXTAS	8975 MB	11,167	11,522	11,56	11,416	6,13	Cluster
	100			11,78	11,809	11,792	11,794	8,47	
	200			12,317	12,098	11,93	12,115	16,44	
	300			12,77	12,702	13,793	13,088	22,94	
	500			Falla del sistema			0	0	
	600						0		
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	

Implantación y prueba de la solución propuesta

600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	MIXTAS	8975 MB	Falla del sistema			0	0	Nodo
	100						0		
	200						0		
	300						0		
	500						0		
	600						0		
600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	SELECT	8975 MB	20,806	20,112	19,809	20,242	3,44	Cluster
	100			28,238	28,161	28,156	28,185	3,55	
	200			55,703	55,824	54,525	55,351	3,61	
	300			83,638	83,6	81,223	82,82	3,62	
	500			135,28	136,09	135,66	135,679	3,68	
	600			162,94	162,48	162,45	162,619	3,68	
	1000			265,14	270,64	268,38	268,051	3,71	
	1500			379,7	380,48	372,9	377,695	3,97	
	2000			462,02	464,91	473,64	466,856	4,28	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	SELECT	8975 MB	Falla del sistema			0	0	Nodo
	100						0		
	200						0		
	300						0		
	500						0		
	600						0		
	1000						0		
	1500						0		
	2000						0		

Tabla 11. Resultados de las pruebas de estrés para la configuración 2.

600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	MIXTAS	8975 MB	7,504	7,388	7,411	7,434	9,41	Cluster
	100			10,812	10,806	10,801	10,806	9,25	
	200			20,272	20,772	20,515	20,52	9,74	
	300			19,322	19,687	19,385	19,465	15,41	
	500			18,923	19,541	19,617	19,36	25,38	
	600			19,247	19,051	18,862	19,053	32,41	

Implantación y prueba de la solución propuesta

Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	MIXTAS	8975 MB	7,538	7,407	7,54	7,495	9,33	Nodo
	100			10,771	10,763	10,763	10,766	9,28	
	200			19,941	19,644	20,14	19,908	10,04	
	300			18,839	19,53	19,527	19,299	15,54	
	500			18,898	18,536	18,546	18,66	26,79	
	600			Falla del sistema					
600 Usuarios									
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	SELECT	8975 MB	19,269	19,432	19,358	19,353	3,61	Cluster
	100			27,331	27,722	27,64	27,564	3,62	
	200			55,761	55,44	55,428	55,543	3,59	
	300			83,367	82,737	83,367	83,157	3,6	
	500			138,31	138,83	138,57	138,569	3,6	
	600			165,47	166,17	162,58	164,736	3,63	
	1000			277,16	278,02	276,76	277,312	3,56	
	1500			413,2	415,26	416,54	415,002	3,61	
	2000			553,34	549,11	553,54	551,994	3,62	
Escala	Transacciones	Tipo	Tamaño BD	tps 1	tps 2	tps 3	tps (media)	Tiempo (s)	
600	70	SELECT	8975 MB	7,548	7,545	7,559	7,551	9,25	Nodo
	100			10,798	10,802	10,809	10,803	9,25	
	200			21,605	21,6	21,607	21,604	9,25	
	300			32,367	32,429	32,392	32,396	9,26	
	500			53,983	54,016	53,958	53,986	9,26	
	600			64,765	64,813	64,745	64,774	9,26	
	1000			107,99	108,03	107,77	107,928	9,26	
	1500			161,56	161,81	162,04	161,803	9,27	
	2000			215,49	215,52	215,45	215,488	9,28	

Tabla 12. Resultados de las pruebas de estrés para la configuración 3

Se simularon además concurrencias de hasta 700 usuarios por segundo realizando peticiones de baja, media y alta complejidad las cuales generaron un gran volumen de datos sobre una BD de 3240 MB de tamaño utilizando la herramienta JMeter para la configuración 3, de la cual se obtuvieron los siguientes resultados.

Implantación y prueba de la solución propuesta

Label	# Muestras	Media	Mín	Máx	Std. Dev.	% Error	Rendimiento	Kb/sec	Avg. Bytes
Baja_c	854	1846	47	9610	1124,36	0,00%	1,4/sec	0,88	632,0
Media_c	580	4184	0	9016	2359,09	0,00%	57,9/min	147,54	156614,0
Alta_c	151	4154	906	8969	2373,48	0,00%	15,1/min	47,29	192897,0
TOTAL	1585	2921	0	9610	2146,01	0,00%	2,6/sec	195,53	76027,3

Tabla 13. Resultados para la prueba de estrés 4 generadas por la herramienta JMeter.

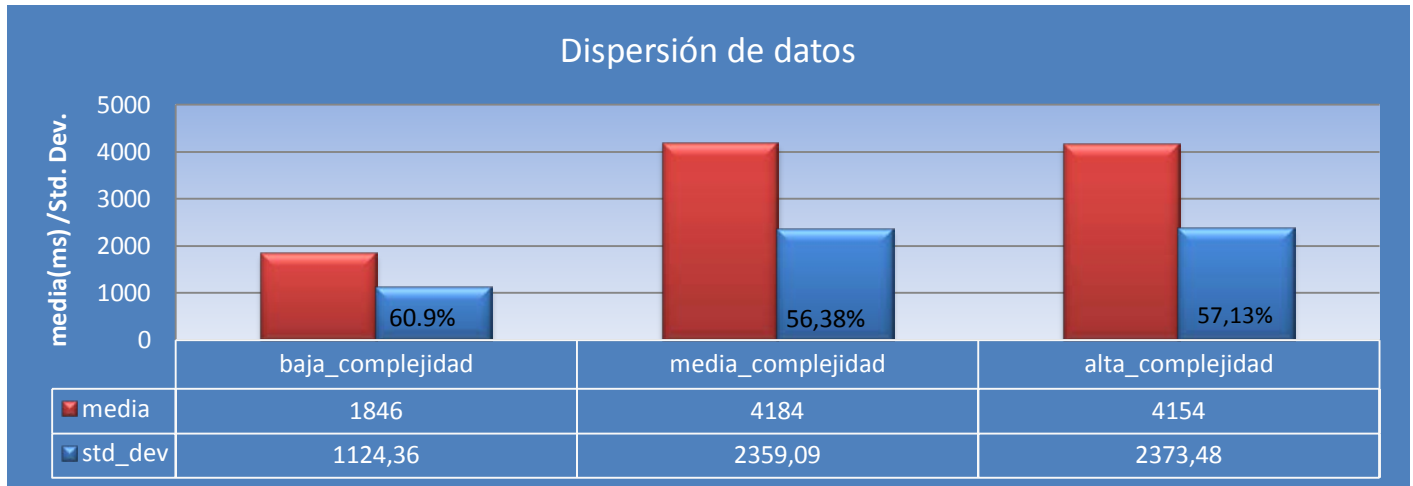


Figura 21. Dispersión de los datos para la prueba de estrés 4.

De las pruebas de estrés ejecutadas se concluye que las configuraciones 1 y 2 no cumplen con los requerimientos del sistema ya que no son capaces de soportar grandes concurrencias lo que provoca que la integridad de los datos se vea comprometida. Para la configuración 3 los resultados obtenidos son aceptables aunque los tiempos de respuesta para grandes volúmenes de transacciones mixtas no son satisfactorios. (Ver Anexo 2 Pruebas de estrés)

3.6.4 Ejecución de pruebas de volumen

Los resultados de pruebas anteriores brindan información suficiente para determinar que el sistema es capaz de asegurar la concurrencia de 700 usuarios realizando peticiones a las diferentes BD que conforman el sistema de réplica. El hardware es adecuado en cuanto a capacidad de procesador, memoria y el ancho de banda de la red para satisfacer el volumen actual, aunque se recomienda para acelerar y asegurar el procesamiento en cuanto a transacciones de lectura y escritura el uso de discos SAN. Se recomienda además para un futuro crecimiento en cuanto a usuarios concurrentes que se le

adicione un nuevo nodo de BD al sistema de réplica con el fin de aumentar la capacidad de procesamiento de dicho sistema.

3.7 Conclusiones parciales

En el presente capítulo se abordaron las características y requerimientos para la realización de pruebas a las BD así como la selección de herramientas para la realización de dichas pruebas. Esto permitió a partir del análisis de los resultados de las pruebas seleccionar la configuración más adecuada para el entorno de réplica de pruebas. Las pruebas aplicadas a las diferentes configuraciones demuestran la incapacidad de procesamiento de un servidor de BD centralizadas frente a un sistema distribuido para satisfacer altas concurrencias. Se evidencia que la configuración más óptima para establecer un entorno de producción a partir del entorno de pruebas es la configuración 3. Se recomienda que las pruebas aplicadas sean ejecutadas nuevamente sobre el hardware que dispone PDVSA para cada uno de los sistemas de réplicas propuestos.

Conclusiones generales

Para el desarrollo del presente trabajo de diploma se realizó un estudio de los procesos y políticas establecidas para el flujo informativo de los medios digitales de PDVSA a utilizar para las nuevas soluciones. Se realizó una extensa búsqueda bibliográfica sobre el proceso de réplica en bases de datos distribuidas.

Se realizó una exhaustiva búsqueda bibliográfica acerca de las últimas tendencias y tecnologías que a nivel internacional se están utilizando en el mundo de la informática, para definir aquellas que mejor respuesta darían al problema al cual se pretende dar solución, partiendo de la previa definición del gestor de base de datos PostgreSQL y haciendo uso de tecnologías libres.

Por tanto se considera que se han cumplido cada uno de los objetivos trazados del presente trabajo, mediante la creación de un sistema que permita la configuración, gestión y mantenimiento de una política de réplica de datos ofreciendo alta disponibilidad y rendimiento para las nuevas soluciones de los medios digitales de PDVSA.

Recomendaciones

El presente trabajo de diploma propone una solución que permite la configuración, gestión y mantenimiento de una política de réplica de BD ofreciendo alta disponibilidad y rendimiento de los servidores de BD para las nuevas soluciones de los medios digitales de PDVSA, por lo que se recomienda:

- ✓ La implementación de una aplicación con interfaz gráfica de usuario que permita la administración y configuración de los servicios de la herramienta Cybercluster.
- ✓ Se recomienda además que se profundice en la teoría expuesta en el documento para uso general en la universidad, tanto para los proyectos productivos como para el uso con fines docentes e investigativos.
- ✓ Realizar un estudio del cual se derive un procedimiento para el diseño e implementación de sistemas de réplica.
- ✓ Implementación de una aplicación con interfaz gráfica de usuario que permita la configuración optimizada de los servidores PostgreSQL para su puesta en producción.

Bibliografía referenciada y consultada

1. **Mato, Lic. Rosa María García.** *Diseño de Bases de Datos.* 1999.
2. **Powell, Gavin.** *Beginning Database Design.* s.l. : Wiley Publishing, Inc., 2006. ISBN-13: 978-0-7645-7490-0..
3. **Vega, Norge Fajardo.** *Sistema de réplica para bases de datos distribuidas en PostgreSQL.* La Habana : s.n., 2007.
4. **C.J.Date.** *Introducción a los Sistemas de Bases de Datos.* s.l. : Prentice Hall, 2001.
5. **Peter Rob, Carlos Coronel.** *Sistemas de bases de datos: Diseño, implementación y administración.* s.l. : Cengage Learning Editores. ISBN 9706862862, 9789706862860..
6. **Monge, Raúl.** *Base de Datos Distribuidas: Replicación.* Valparaíso : Universidad Técnica , 2005.
7. **Stallman, Richard M.** *Software libre para una sociedad libre.* s.l. : Mapas 9, 2004. ISBN: 84-933555-1-8..
8. **Group, PostgreSQL Global Development.** Wikipedia, la enciclopedia libre. [En línea] [Citado el: 3 de Febrero de 2009.] <http://es.wikipedia.org/wiki/PostgreSQL>.
9. **Momjian, Bruce.** *PostgreSQL, Introduction and Concepts.* s.l. : Addison Wesley, 2000. ISBN 0-201-70331-9..
10. **Debian, Proyecto.** Wikipedia, la enciclopedia libre. [En línea] [Citado el: 3 de Febrero de 2009.] http://es.wikipedia.org/wiki/Debian_GNU/Linux.
11. **Helal, Abdelsalam A., Heddaya, Abdelsalam A. y Bhargava, Bharat B.** *Replication Techniques in Distributed Systems.* New York, Boston, Dordrecht, London, Moscow : Kluwer Academic Publishers, 2002. ISBN: 0-306-47796-3, ISBN: 0-7923-9800-9..
12. **IEEE.** *Computer Dictionary. Standard 610.* Nueva York : s.n., 1990.
13. **J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, Dennis Rea.** *Performance Testing Guidance for Web Applications.* s.l. : Microsoft, 2007.

Anexos

Anexo 1 Configuraciones

Configuración 1

Archivo pglb.conf

```
<Cluster_Server_Info>
  <Host_Name>10.0.0.218</Host_Name>
  <Port>5432</Port>
  <Max_Connect>300</Max_Connect>
</Cluster_Server_Info>
<Cluster_Server_Info>
  <Host_Name> 10.0.0.219</Host_Name>
  <Port>5432</Port>
  <Max_Connect>300</Max_Connect>
</Cluster_Server_Info>
<Host_Name> 10.0.0.216 </Host_Name>
<Backend_Socket_Dir> /tmp </Backend_Socket_Dir>
<Receive_Port> 5432 </Receive_Port>
<Recovery_Port> 6001 </Recovery_Port>
<Max_Cluster_Num> 128 </Max_Cluster_Num>
<Use_Connection_Pooling> yes </Use_Connection_Pooling>
<LifeCheck_Timeout> 3s </LifeCheck_Timeout>
<LifeCheck_Interval> 15s </LifeCheck_Interval>
<Connection_Life_Time> 0s </Connection_Life_Time>
```

Archivo postgresql.conf

```
max_connections = 300
shared_buffers = 2097152kB
temp_buffers = 8MB
max_prepared_transactions = 300
work_mem = 2MB
maintenance_work_mem = 512MB
max_stack_depth = 8MB
max_fsm_pages = 271096
max_fsm_relations = 1000
fsync = on
synchronous_commit = on
```

```
wal_sync_method = fsync
wal_buffers = 8MB
commit_delay = 0
commit_siblings = 1
checkpoint_segments = 128
checkpoint_completion_target = 0.9
enable_bitmapscan = on
enable_hashagg = on
enable_hashjoin = on
enable_indexscan = on
enable_mergejoin = on
enable_nestloop = on
enable_seqscan = on
enable_sort = on
enable_tidscan = on
random_page_cost = 4.0
effective_cache_size = 3000MB
default_statistics_target = 200
constraint_exclusion = off
autovacuum = on
autovacuum_naptime = 1140min
deadlock_timeout = 5s
max_locks_per_transaction = 64
```

Configuración 2

Archivo pglb.conf

```
<Cluster_Server_Info>
  <Host_Name>10.0.0.218</Host_Name>
  <Port>5432</Port>
  <Max_Connect> 450 </Max_Connect>
</Cluster_Server_Info>
<Cluster_Server_Info>
  <Host_Name> 10.0.0.219</Host_Name>
  <Port>5432</Port>
  <Max_Connect> 450 </Max_Connect>
</Cluster_Server_Info>
<Host_Name>10.0.0.216 </Host_Name>
<Backend_Socket_Dir>/tmp</Backend_Socket_Dir>
<Receive_Port>5432</Receive_Port>
<Recovery_Port>6001</Recovery_Port>
<Max_Cluster_Num>128</Max_Cluster_Num>
<Use_Connection_Pooling> yes</Use_Connection_Pooling>
<LifeCheck_Timeout>3s</LifeCheck_Timeout>
```

```
<LifeCheck_Interval>15s</LifeCheck_Interval>  
<Connection_Life_Time>0s</Connection_Life_Time>
```

Archivo postgresql.conf

```
max_connections = 450  
shared_buffers = 2097152kB  
temp_buffers = 8MB  
max_prepared_transactions = 450  
work_mem = 2MB  
maintenance_work_mem = 512MB  
max_stack_depth = 8MB  
max_fsm_pages = 271096  
max_fsm_relations = 1000  
fsync = on  
synchronous_commit = on  
wal_sync_method = fsync  
wal_buffers = 8MB  
commit_delay = 0  
commit_siblings = 1  
checkpoint_segments = 128  
checkpoint_completion_target = 0.9  
enable_bitmapscan = on  
enable_hashagg = on  
enable_hashjoin = on  
enable_indexscan = on  
enable_mergejoin = on  
enable_nestloop = on  
enable_seqscan = on  
enable_sort = on  
enable_tidscan = on  
random_page_cost = 4.0  
effective_cache_size = 3000MB  
default_statistics_target = 200  
constraint_exclusion = off  
autovacuum = on  
autovacuum_naptime = 1140min  
deadlock_timeout = 5s  
max_locks_per_transaction = 64
```

Configuración 3

Archivo pglb.conf

```
<Cluster_Server_Info>
```

```
<Host_Name>10.0.0.218</Host_Name>
<Port>5432</Port>
<Max_Connect>600</Max_Connect>
</Cluster_Server_Info>
<Cluster_Server_Info>
  <Host_Name> 10.0.0.219</Host_Name>
  <Port>5432</Port>
  <Max_Connect>600</Max_Connect>
</Cluster_Server_Info>
<Host_Name>10.0.0.216 </Host_Name>
<Backend_Socket_Dir>/tmp</Backend_Socket_Dir>
<Receive_Port>5432</Receive_Port>
<Recovery_Port>6001</Recovery_Port>
<Max_Cluster_Num>128</Max_Cluster_Num>
<Use_Connection_Pooling> yes</Use_Connection_Pooling>
<LifeCheck_Timeout>3s</LifeCheck_Timeout>
<LifeCheck_Interval>15s</LifeCheck_Interval>
<Connection_Life_Time>0s</Connection_Life_Time>
```

Archivo postgresql.conf

```
max_connections = 600
shared_buffers = 2097152kB
temp_buffers = 8MB
max_prepared_transactions = 600
work_mem = 2MB
maintenance_work_mem = 512MB
max_stack_depth = 8MB
max_fsm_pages = 271096
max_fsm_relations = 1000
fsync = on
synchronous_commit = on
wal_sync_method = fsync
wal_buffers = 8MB
commit_delay = 0
commit_siblings = 1
checkpoint_segments = 128
checkpoint_completion_target = 0.9
enable_bitmapscan = on
enable_hashagg = on
enable_hashjoin = on
enable_indexscan = on
enable_mergejoin = on
enable_nestloop = on
enable_seqscan = on
enable_sort = on
```



```
enable_tidscan = on
random_page_cost = 4.0
effective_cache_size = 3000MB
default_statistics_target = 200
constraint_exclusion = off
autovacuum = on
autovacuum_naptime = 1140min
deadlock_timeout = 5s
max_locks_per_transaction = 64
```

Configuración 4

Archivo pglb.conf

```
<Cluster_Server_Info>
  <Host_Name>10.0.0.218</Host_Name>
  <Port>5432</Port>
  <Max_Connect>4000</Max_Connect>
</Cluster_Server_Info>
<Cluster_Server_Info>
  <Host_Name> 10.0.0.219</Host_Name>
  <Port>5432</Port>
  <Max_Connect>4000</Max_Connect>
</Cluster_Server_Info>
<Host_Name>10.0.0.216 </Host_Name>
<Backend_Socket_Dir>/tmp</Backend_Socket_Dir>
<Receive_Port>5432</Receive_Port>
<Recovery_Port>6001</Recovery_Port>
<Max_Cluster_Num>128</Max_Cluster_Num>
<Use_Connection_Pooling> yes</Use_Connection_Pooling>
<LifeCheck_Timeout>3s</LifeCheck_Timeout>
<LifeCheck_Interval>15s</LifeCheck_Interval>
<Connection_Life_Time>0s</Connection_Life_Time>
```

Archivo postgresql.conf

```
max_connections = 4000
shared_buffers = 4194304kB
temp_buffers = 8MB
max_prepared_transactions = 4000
work_mem = 2MB
maintenance_work_mem = 512MB
max_stack_depth = 8MB
max_fsm_pages = 900000
max_fsm_relations = 50000
```

```

fsync = on
synchronous_commit = on
wal_sync_method = fsync
wal_buffers = 4MB
commit_delay = 1000
commit_siblings = 10
checkpoint_segments = 128
checkpoint_completion_target = 0.9
effective_cache_size = 6000MB
default_statistics_target = 200
constraint_exclusion = off
autovacuum = on
autovacuum_vacuum_threshold = 50000
autovacuum_analyze_threshold = 50000
deadlock_timeout = 6s
max_locks_per_transaction = 128

```

Anexo 2 Gráficas comparativas

Pruebas de Carga

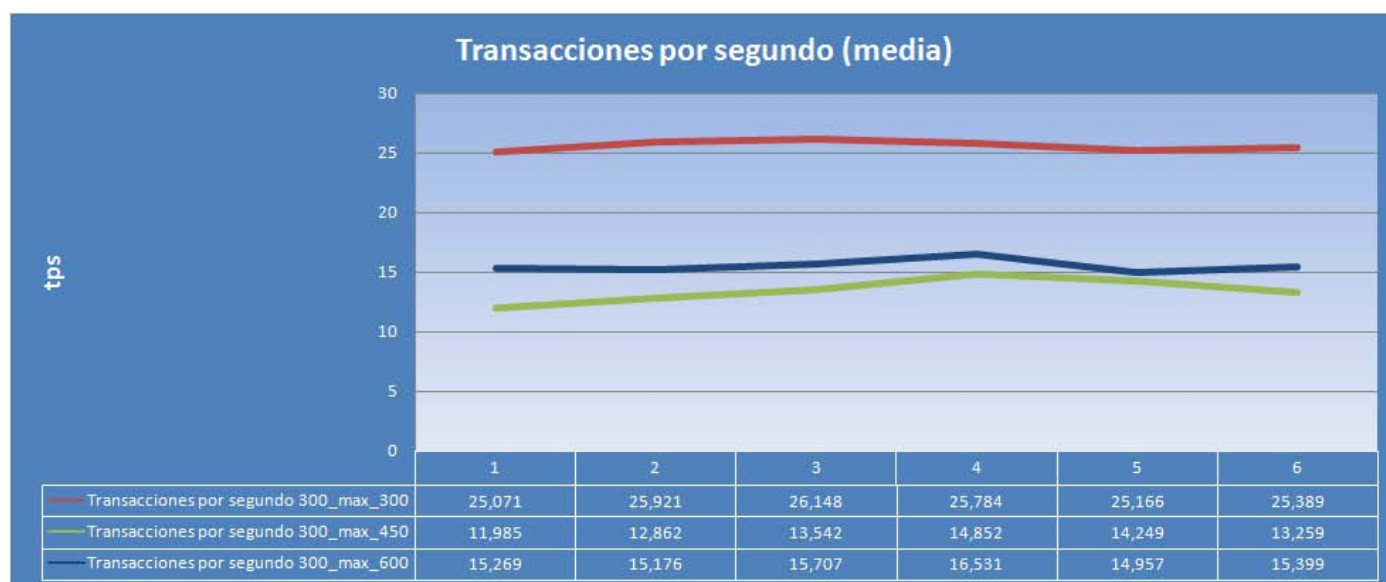


Figura 22. Media de transacciones por segundo para las transacciones mixtas



Figura 23. Media de los tiempos para las transacciones mixtas.



Figura 24. Media de las transacciones por segundo para las consultas SELECT.



Figura 25. Media de los tiempos para las consultas SELECT.

Picos de carga



Figura 26. Media de las transacciones para las transacciones mixtas.



Figura 27. Media de los tiempos para las transacciones mixtas.

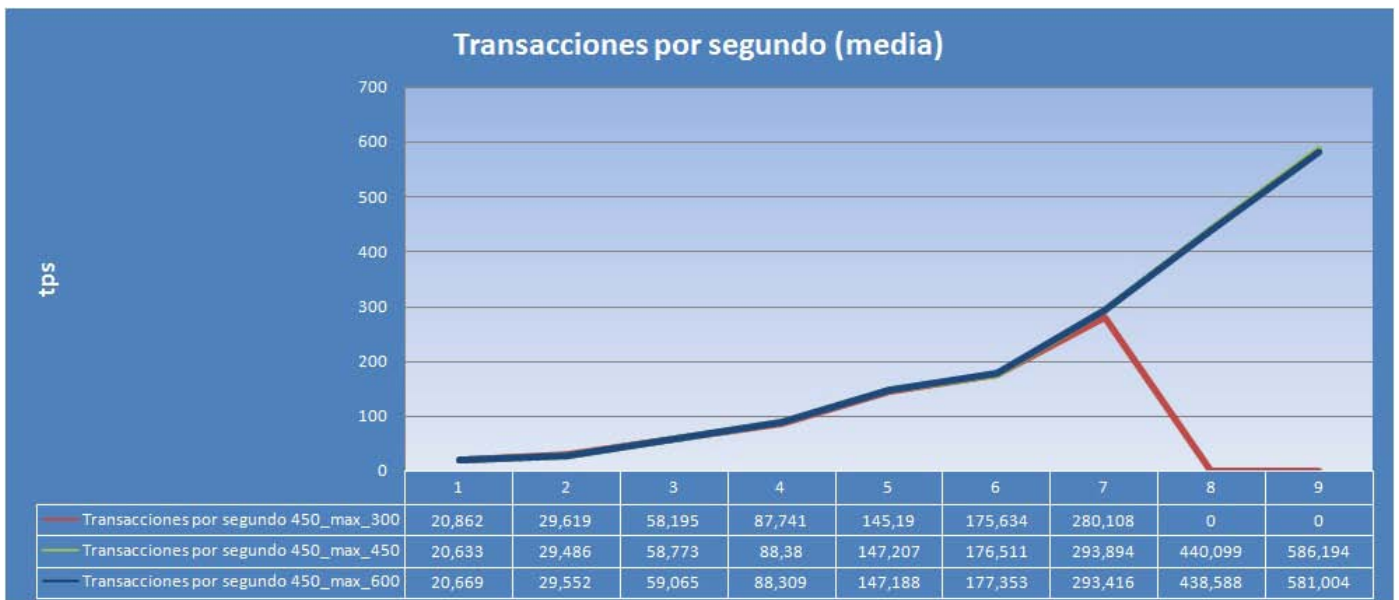


Figura 28. Media de las transacciones por segundo para las consultas SELECT.

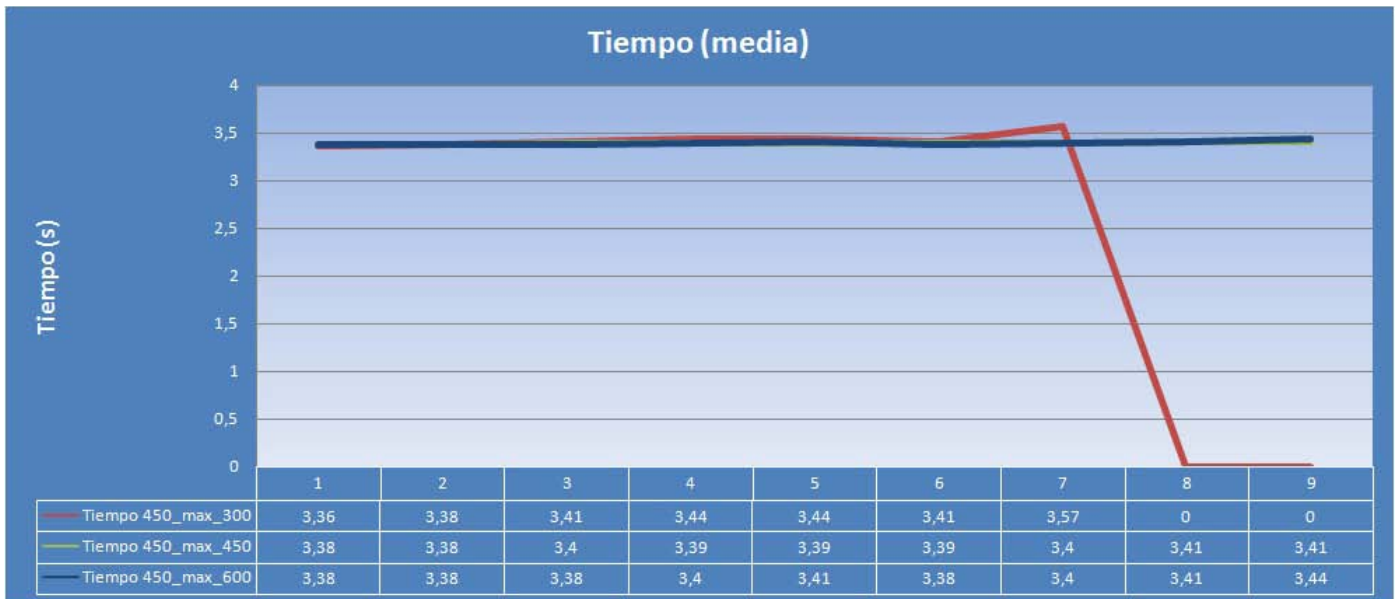


Figura 29. Media de los tiempos para las consultas SELECT.

Pruebas de estrés



Figura 30. Media de las transacciones por segundo para las transacciones mixtas.

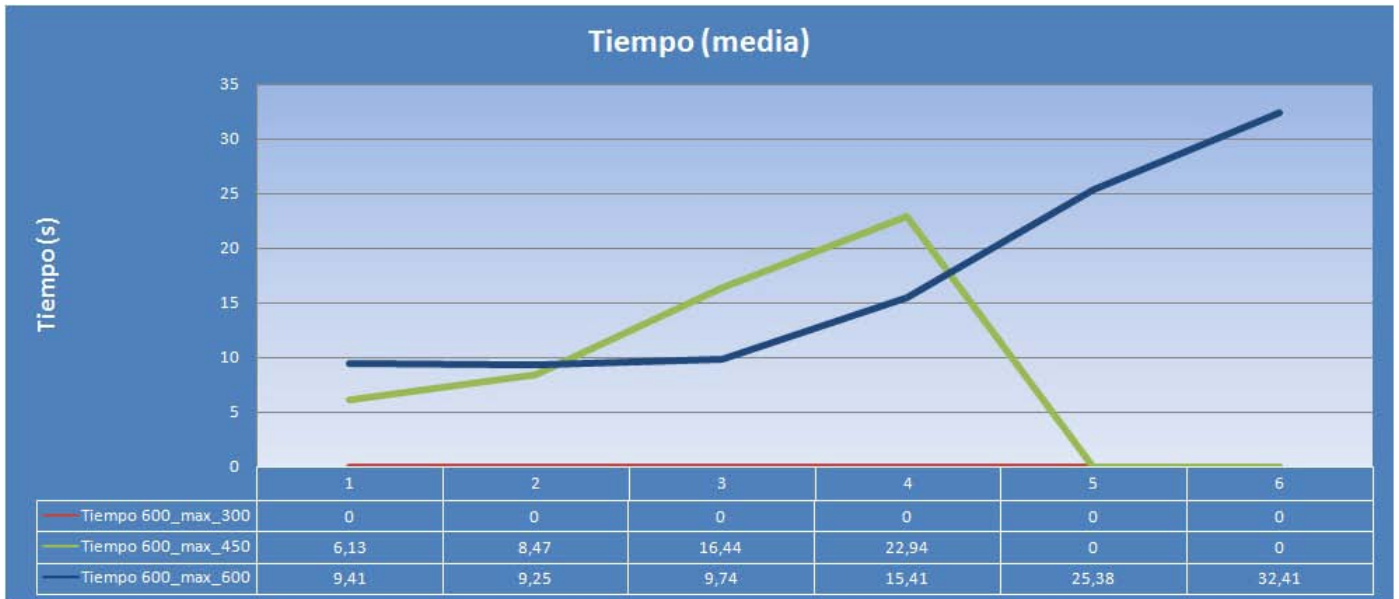


Figura 31. Media de los tiempos para las transacciones mixtas.



Figura 32. Media de las transacciones por segundo para las consultas SELECT.

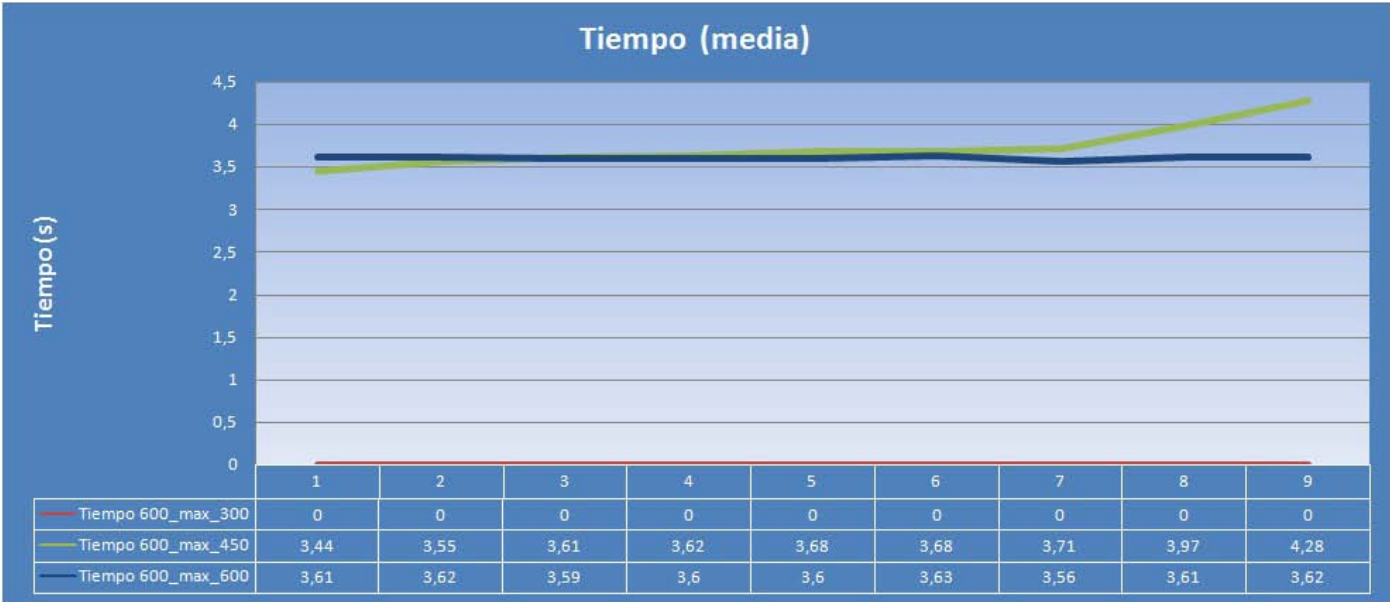


Figura 33. Media de los tiempos para las consultas SELECT.

Glosario de términos

Capítulo 1

PHP

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools).

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

SQL

Lenguaje de consulta estructurado (SQL en inglés **Structured Query Language**), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desde enero de 2008 es una subsidiaria de Sun Microsystems desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Plone

Plone es un Sistema de Gestión de Contenidos o CMS por sus siglas en inglés (Content Management System), basado en Zope (que tiene miles de desarrolladores en todo el mundo) y programado en Python. Es un desarrollo basado en código abierto. Puede utilizarse como servidor intranet o extranet, un Sistema de Publicación de documentos y una herramienta de trabajo en grupo para colaborar entre entidades distantes. Frecuentemente es utilizado también como CRM, aun cuando no fue diseñado como tal.

Failover

Es un modo de operación de backup en el cual las funciones de un componente del sistema son asumidas por un segundo componente del sistema cuando el primero no se encuentra disponible debido a un fallo ó

un tiempo de parada preestablecido. Es usado para hacer a los sistemas más tolerantes a fallos, y de esta forma permanentemente disponible.

Off-Loading

Sitio de descarga, o desconectados para protegerlos de cargas o peticiones extremas

GNU

Sistema operativo libre, su nombre es un acrónimo recursivo que significa **GNU No es Unix** (*GNU is Not Unix*), fue totalmente diseñado para ser compatible con UNIX.

UNIX

Es un Sistema Operativo no libre muy popular, portable, multitarea y multiusuario.

Capítulo 2

bzip2

Es un programa libre desarrollado bajo licencia BSD que comprime y descomprime archivos de extensión .bz2, .tar.bz2, .tbz2, .tb2.

ECM: Enterprise Content Management (*ECM*) son las estrategias, métodos y herramientas usadas para capturar, administrar, almacenar, proteger y entregar contenidos y documentos relacionados a procesos organizacionales.

gcc

Es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un programa ejecutable binario en el lenguaje de la máquina donde ha de correr. La sigla GCC significa "GNU Compiler Collection".

g++

Es un compilador para C++ para todos los sistemas basados en UNIX usualmente ejecutado usualmente a través de líneas de comando. Distribuido en la mayoría de los sistemas basados en UNIX.

zlib1g-dev

Es una librería que implementa el método de descompresión por deflación que se encuentra en gzip y PKZIP. Este paquete incluye los archivos de soporte del desarrollo.

libreadline-dev

(Librerías de desarrollo) **libreadline-dbg** (librerías para depurar): La biblioteca readline de GNU ayuda en la consistencia de la interfaz de usuario a través de programas discretos que necesitan proporcionar una interfaz de línea de comandos.

make

Es una utilidad que permite definir reglas de dependencia entre ficheros. Aunque puede utilizarse para diferentes fines, está especialmente orientado a la compilación de código. El propósito de make es determinar automáticamente qué piezas de un programa necesitan ser recompiladas y lanzar las órdenes necesarias para lograrlo.

flex

Es una herramienta para generar escáneres: programas que reconocen patrones léxicos en un texto.

bison

GNU bison es un programa generador de analizadores sintácticos de propósito general perteneciente al proyecto GNU disponible para prácticamente todos los sistemas operativos, se usa normalmente acompañado de flex aunque los analizadores léxicos se pueden también obtener de otras formas.