PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTADE DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE

INGENIERO EN SISTEMAS

"ANALISIS COSTO BENEFICIO DE UNA BASE DE DATOS DISTRIBUIDA"

AUTOR

LUIS BENAVIDES

DIRECTOR:

JAVIER CONDOR

QUITO, ENERO 2015

2015-Luis Alfredo Benavides Jaramillo

Dedicatoria

Dedico este trabajo a mis padres por ser el pilar fundamental en mi vida el apoyo que siempre ha estado presente y quienes siempre me han dado animos para seguir adelante en todos mis proyectos.

A mí prometida Zaira quien es el motor de mi vida quien siempre me ha mostrado su apoyo en todo tipo de condiciones.

A mis hermanos quienes siempre me han dado su apoyo y la energía para seguir adelante.

2015-Luis Alfredo Benavides Jaramillo

Agradecimiento

Agradezco a mi familia por todo lo que me ha brindado para la obtención de este Titulo el cual lo he anhelado mucho.

Agradezco a mis profesores los cuales han abierto un mundo de conocimiento en mí.

A mi director de esta disertación quien me ha dado la guía para llevar a cabo este proyecto el cual a alcanazado mis expectativas de investigación.

ÍNDICE GENERAL

CAPÍTULO I **6**
MARCO TEÓRICO **6**
1.1 Sistema de gestión de bases de datos **6**
1.2 Evolución de los Sistemas de gestión de bases de datos. **7**
1.2.1 Estructura de Base de Datos Distribuidas 11
1.2.2 Consideraciones al distribuir la base de datos 12
1.2.3 Ventajas de la distribución de datos 12
1.2.4 Fiabilidad y disponibilidad 13
1.2.5 Agilización del procesamiento de consultas 13
1.2.6 Desventajas de la distribución de los datos 13
1.2.7 Transparencia y Autonomía 14
1.2.8 Transparencia de la repetición y la fragmentación 15
1.2.9 Transparencia de localización 16
1.2.10 Esquema completo de asignación de nombres 16
1.2.11 Transparencia y actualizaciones 17
1.2.12 El archivo principal de base de datos 30
1.2.13 Una base de datos lógica 32
1.2.13.1 Ejecución 32
1.2.13.2 Ventajas de la Base de datos distribuida: 34
CAPÍTULO II **36**
SELECCIÓN DE UN DBMS QUE SOPORTEN DISTRIBUCIÓN. **36**
2.1 Infraestructura. **36**
2.2 Hardware **37**
2.3 Motores de base de datos que soporten distribución. **38**
2.4 Selección de un motor para manejo de una de base de datos distribuida. **39**
2.4.1 Motores de base de datos que soporten distribución. 42
2.5 Selección de un motor para manejo de una de base de datos distribuida. **45**
2.5.1 Otros motores de búsqueda 45
2.6 Software adicional para la simplificación de la implementación **45**
2.6.1 Diseño de la base de datos 47
CAPÍTULO III **44**
INSTALACIÓN DE MYSQL CLÚSTER **44**
3.1 Introducción **44**
3.2 Infraestructura **46**
3.3 Creación de la máquina virtual en virtual Box. **47**
3.4 Descarga del software que vamos a utilizar en nuestra instalación. **53**
3.5 Instalación del nodo de gestión MGMT. **55**

3.6 Instalación de los Data Nodes. 57
3.7 Instalación de los nodos SQL. 57
3.8 Arranque inicial del Clúster. 58
3.9 Probar creando una tabla. 61
3.10 Carga de datos de ejemplo. 64
3.11 Convertir los procesos en un servicio Windows. 65
3.12 Diseño para mejorar el rendimiento. 67
3.12.1 Carga de datos masiva de datos. 70
CAPITULO IV 72
CONCLUSIONES Y RECOMENDACIONES SOBRE EL 72
ANÁLISIS DEL COSTO BENEFICIO 72
4.1 Análisis costos de recurso humano 72
4.2 Análisis de costo de infraestructura 74
4.3Análisis de los Rubros previamente mencionados 75
4.4 Pruebas de rendimiento 76
4.2.3 Resultados de pruebas 80
4.3 Conclusiones 83
4.4 Recomendaciones 84
Tablas e Ilustraciones. 87
Bibliografía 89

CAPÍTULO I

MARCO TEÓRICO

De acuerdo a la investigación que se realiza sobre el análisis del costo beneficio de implementar una base de datos distribuida, para el caso de estudio de una empresa comercial primero se estima las conceptualizaciones tanto de que es un sistema de gestión de base de datos, su estructura, características, ventajas y desventajas de una red de distribución, fiabilidad, disponibilidad, consultas, transparencia y actualizaciones que existe en el mercado.

1.1 Sistema de gestión de bases de datos

Un sistema de gestión de bases de datos constituye un conjunto de programas que establecen el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.(SGBD)1.Se puede indicar que la mayoría de los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o a través de aplicaciones al efecto los SGBD también determinan métodos que permiten la integridad de los datos, para administrar el acceso de usuarios a los datos y que permiten la recuperación de la información si el sistema se corrompe. Este determina presentar la información de la base de datos en variados formatos. La mayoría de los SGBD contienen un generador de informes. Además incluir un módulo gráfico que permita presentar la información tanto con gráficos y también con tablas.

Existe gran cantidad de SGBD diferentes según manejen los datos y muchos tamaños distintos de acuerdo a su función sobre ordenadores personales y que presentan poca memoria a grandes sistemas que funcionan en mainframes que tienen sistemas de almacenamiento especiales. [1]

1SGBD Un sistema de gestión de bases de datos constituye un conjunto de programas que establecen el almacenamiento…

Cuando se requiere acceder a los datos se lo hace mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. Además estos simplifican la interrogación y la presentación de la información. Se puede indicar que un SGBD sirve para controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Se indica además que las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con mucha atención.

1.2 Evolución de los Sistemas de gestión de bases de datos.

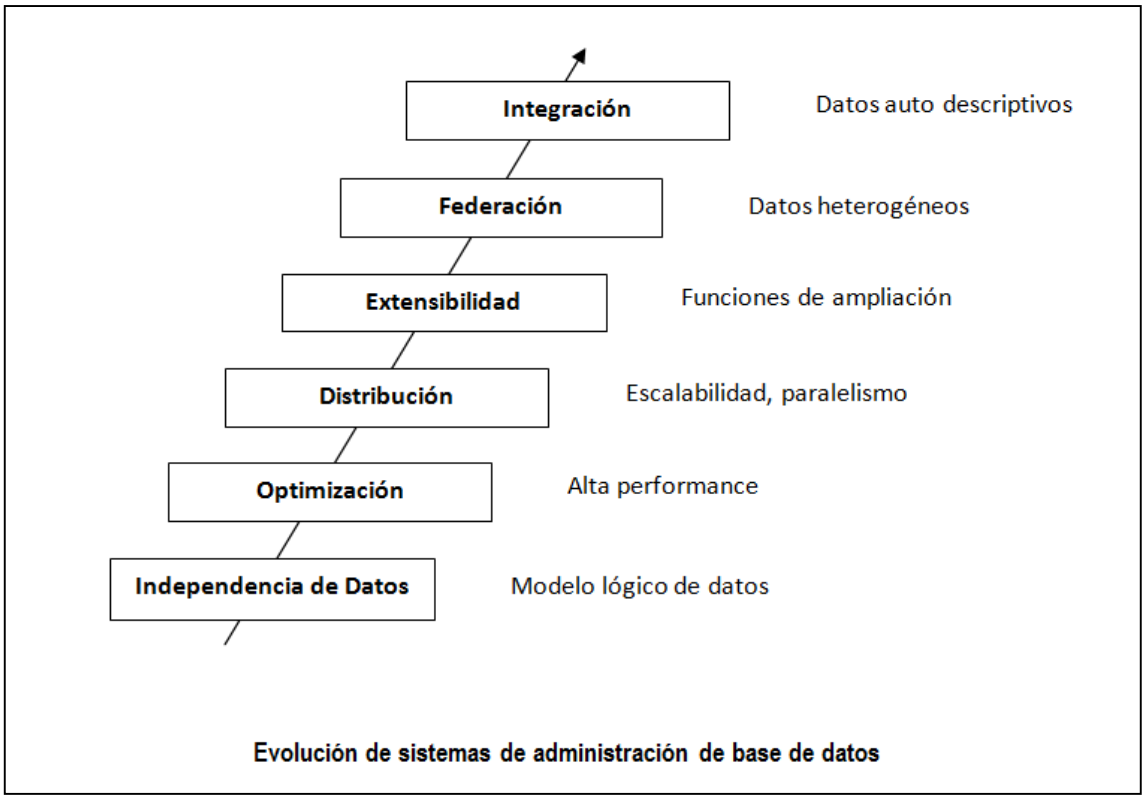


Fig. 1-01[A] 2 Evolución de los Sistemas de gestión de bases de datos

En la figura anterior se puede observar la evolución del sistema de administración de base de datos con el modelo relacional el mismo que permanece la independencia de los datos. El Sistema R de IBM[2] constituye el sistema inicial que implemento y desarrollo las ideas de Codd. Se puede indicar que este sistema R fue la base para SQL/DS[3] que después fue

[2] IBM International Business Machines Empresa Multinacional Estadounidense de tecnología y consultoría…
[3]SQL/DS (Structured Query Language/Data System), released in 1981, was IBM's first commercial relational.

DB2[4]. Además este permitió la implementación del lenguaje usado casi siempre en la actualidad, y esta entrada de la apertura sistemas de administración de base de datos comerciales.

En la actualidad, los sistemas que permiten la administración de base de datos relacionales son el la mayoría DBMS[5] y son desarrollados por varias compañías del software. Se puede indicar que los DBMS relacionales son PostgreSQL, MySQL y dBASE.Oracle, Microsoft SQL Server, DB2, INGRES, etc. [1].

Se determina que cuando las BDD relacionales se hicieron populares, surgió la necesidad de ofrecer consultas de alta desempeño. Siendo el optimizador de consultas uno de los más importantes y que tiene un alto nivel de sofisticación de los DBMS. Se puede indicar que la mayoría de usuarios lo consideran muy especial constituye la caja negra por así decirlo y pasa cualquier consulta SQL a este. SQL calcula la manera más eficaz y veloz de recuperar datos siempre que se tenga en cuenta los factores como la velocidad de su CPU y discos, la cantidad de datos disponible, la situación de los datos, el tipo de datos, la existencia de índices entre los más importantes.

Se puede indicar que existe un rasgo llamado Función de Partición de Base de datos (DPF)[6] el mismo que permite aumentar y extender una base de datos para muchas máquinas las mismas que se basan en una arquitectura que no está compartida , lo que se determina es que cada máquina tiene sus propios CPUs[7] y discos; lo que se consigue con ello es escalar casi linealmente, se puede decir que realizar una consulta en ese entorno permite recuperar en paralelo y que cada máquina recupera una porción del resultado total.

Se puede indicar además que los DBMS evolucionaron, luego, con el concepto de extensibilidad. Tomando en cuenta el lenguaje de Consulta Estructurada (SQL) inventado por IBM en los inicios de 1970 a través de los años ha seguido una constante evolución. Este es un idioma muy poderoso, por lo que los usuarios también están autorizados a desarrollar su propio código que permita extender el SQL[8]. Por ejemplo, se puede crear

[4]DB2 es una marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base...
[5]DBMS.(Data Base Management System). Son las siglas en inglés para los Sistemas de Gestión de Bases de...
[6]DPFFunción de partición de Base de datos
[7]CPUs Unidades de Procesos Computarizados
[8]SQL. (por sus siglas en inglés QueryStructuredLanguage) es un lenguaje declarativo de acceso a bases…

funciones indicadas por el usuario, y guardar procedimientos que den como resultado extender el lenguaje SQL con su propia lógica.

Se puede decir que los DBMS comenzaron a dar soluciones a la problemática de diferentes tipos de datos y de diversas procedencias. En la época actual los servidores de datos permiten almacenar casi todos los tipos de información que incluyen video, audio, datos binarios y así sucesivamente. Es así que, a través del concepto de estandarización al realizar una consulta el usuario deberá basarse en un DBMS como DB2 para acceder a los datos de otros productos de IBM, y de productos que no sean de IBM.[2]

Se puede estimar que el próximo paso evolutivo resalta la integración. En la actualidad la mayoría de negocios necesitan intercambiar la información, y el Lenguaje de Etiquetado extensible (XML)[9] es la tecnología subyacente que se usa para este propósito. XML permite ser extensible, describiéndose a sí mismo como lenguaje. Se puede indicar en esta investigación que su uso ha estado creciendo exponencialmente debido a Web 2.0, así como la arquitectura de servicios. [3]

En la época actual el tema que está vigente es el “Cloud Computing”. Existen muchas, imágenes de DBMS disponible en la Nube Amazon EC2[10], y en el Desarrollo de Negocios Inteligentes y Prueba en la Nube de IBM (también conocido como el Desarrollo y prueba de IBM en la Nube). Se indica que la partición de Base de datos indicada anteriormente calza perfectamente en la nube donde puede solicitar un nodo estándar o un servidor a demanda, y agregar entonces esto a tu nube. Se indica que el equilibrio de los datos es una actividad automática realizada por el DBMS. Lo que representa una gran utilidad en el momento cuando más poder necesita ser dado al servidor de la base de datos para realizar las transacciones de fin de mes o transacciones de fin de año.[3]

[9] XML siglas en inglés de extensibles MarkupLanguage ('lenguaje de marcas extensible'), es un lenguaje…
[10]EC2 Elastic Compute Cloud (EC2) proporciona capacidad informática escalable enla nube de AWS con…

Distribución de bases de datos.

Se puede indicar además que en un sistema de base de datos distribuida, los datos se almacenan en un sinnúmero de computadores. Estos computadores de un sistema distribuido realizan comunicaciones entre sí mediante diversos medios de comunicación, como son los cables de alta velocidad o líneas telefónicas. Además se puede indicar que estos no comparten la memoria principal ni el reloj.

La mayoría de los procesadores de un sistema distribuido pueden tener variaciones que tienen que ver a su tamaño y función. Estos también pueden ser los microcomputadores pequeños, estaciones de trabajo y sistemas de computadores grandes de aplicación general. Los procesadores tienen diferentes nombres, tales como localidades, nodos o computadores.

Es así que un sistema distribuido de bases de datos constituye un conjunto de localidades, a nivel independiente cada uno de las cuales puede participar en la ejecución de transacciones que accedan a datos de una o varias localidades. Se estima que la principal diferencia que se da 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.

Se indica que las Bases de Datos Distribuida constan de muchas ventajas de la computación distribuida al dominio de la gestión de BD. Se determina además que un sistema de cómputo distribuido constituye un conjunto de elementos de procesamiento, que no son iguales, lo cuales se interconectan mediante una red de computadores y que se encuentran cooperando en la ejecución de ciertas tareas asignadas. En definitiva, se puede decir que los sistemas de cómputo distribuido dividen un problema grande y, a veces, inmanejable, en piezas más pequeñas que desarrollan soluciones eficaces. Es así que la viabilidad económica de este enfoque determina algunas razones: Primero se aprovecha más la potencia de la estación de trabajo para resolver tareas complejas .Segundo cada elemento de procesamiento autónomo se puede gestionar independientemente y puede desarrollar mejor sus aplicaciones.

Se puede definir a una BDD desde diferentes puntos de vista, se plantean a continuación algunas definiciones:

Una BDD[11] es una única base de datos lógica que está físicamente separada en varias computadoras que se encuentran conectadas por una red de comunicaciones. Generalmente, la administración de BDD se encuentra centralizada como un recurso corporativo mientras que permite proveer tanto flexibilidad como manejo local. (Burleson 1994)

Una BDD es una colección de múltiples BD, lógicamente interrelacionadas, diseminadas sobre una red de computadoras. Un DBMS distribuido (DDBMS[12]) es un sistema de software que permite la administración de la BDD y hace que esa distribución permanezca transparente al usuario.

Se indica que una BDD no constituye una colección de archivos los cuales pueden ser almacenados individualmente en cada nodo o estación de trabajo de una red. Es así que estos datos deberán formar una BDD mediante estos archivos que necesariamente deben estar interrelacionados, permitiendo un acceso vía una interfaz común.

1.2.1 Estructura de Base de Datos Distribuidas

Además un BDD constituye un verdadero conjunto de localidades, las cuales mantiene un sistema de base de datos local. Es así que cada localidad procesa transacciones locales, o bien transacciones globales entre varias localidades, requiriendo que se comuniquen entre ellas.

Se puede determinar que existen diferencias que se dan en estas configuraciones las cuales son:

Costo de instalación: El coste de conectar físicamente las localidades del sistema

Costo de comunicación: El coste en tiempo y dinero que implica enviar un mensaje desde la localidad A a la B.

[11]BDDes una colección de múltiples BD, lógicamente interrelacionadas, diseminadas sobre una red de…
[12]DDBMS El sistema de administración de base de datos distribuida (DDBMS), está formado por las…

Fiabilidad: La frecuencia con que falla una línea de comunicación o una localidad.

Disponibilidad: La posibilidad de acceder a información a pesar de fallos en algunas localidades o líneas de comunicación.

Cuando existen localidades las cuales están dispersas, por medio de un área geográfica extensa (a lo largo de un país), estas se denominan redes de larga distancia; o pueden estar en un área reducida (en un mismo edificio), denominadas redes de área local. Se indica que cuando existen las primeras se utilizan en la comunicación líneas telefónicas, conexiones de microondas y canales de satélites; en las segundas en cambio se utiliza cables coaxiales de banda base o banda ancha y fibra óptica.[4]

1.2.2 Consideraciones al distribuir la base de datos

Cuando se construyen sistemas distribuido de bases de datos estos responden a varios factores y razones que incluyen compartir la información, fiabilidad y disponibilidad y agilizar el procesamiento de las consultas. Además se puede decir que tienen sus desventajas, como desarrollos de software más costosos, mayor posibilidad de errores y costos extras de procesamiento.

1.2.3 Ventajas de la distribución de datos

Se puede decir que una de las principales ventajas de los sistemas distribuidos consiste en la capacidad de compartir y acceder a la información de la manera más fiable y eficaz.

Utilización compartida de los datos y distribución del control

Se dice que la utilización compartida constituye una de las principales ventajas la cual permite compartir los datos mediante la distribución especifica que cada localidad controle hasta determinado punto los datos almacenados localmente. Es así que se indica un sistema centralizado, el administrador de base de datos de la localidad central controla la base de datos. Mientras que en cambio en un sistema distribuido existe un administrador global de la base de datos que se encarga de todo el sistema.

Se indica también que la responsabilidad está garantizada por el administrador de base de datos de cada localidad. Por lo que cada administrador local tiene un grado de autonomía diferente por cuanto depende del diseño del sistema distribuido, que se conoce como autonomía local. Es así que se dice que la posibilidad de contar con autonomía local es en muchos casos es una ventaja importante de las bases de datos distribuidas.[3]

1.2.4 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.[3]

1.2.5 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.[3]

1.2.6 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:

- Costo del desarrollo de software: es más difícil estructura un sistema de bases de datos distribuidos y por tanto su costo es menor
- 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.

1.2.7 Transparencia y Autonomía

En la sección anterior se vio que una relación puede almacenarse de varias formas en un sistema de base de datos distribuida. Es esencial que el sistema reduzca al mínimo la necesidad de que el usuario se dé cuenta de cómo está almacenada una relación. Como veremos. Un sistema puede ocultar los detalles de la distribución de la información en la red. Esto se denomina transparencia de la red. La transparencia de la red se relaciona, en algún modo, a la autonomía local. La transparencia de la red es el grado hasta el cual los usuarios del sistema pueden ignorar los detalles del diseño distribuido. La autonomía local es el grado hasta el cual el diseñador o administrador de una localidad pueden ser independientes del resto del sistema distribuido. Los temas de transparencia y autonomía serán considerados desde los siguientes puntos de vista:

- Nombre de los datos.
- Repetición de los datos.
- Fragmentación de los datos.
- Localización de los fragmentos y copias.
- Asignación de nombres y autonomía local

Todo elemento de información de una base de datos debe tener un nombre único. Esta propiedad se asegura fácilmente en una base de datos que no esté distribuida.

Sin embargo, en una base de dalos distribuida, las distintas localidades deben asegurarse no utilizar el mismo nombre para dos datos diferentes.

Una solución para este problema es requerir que se registren todos los nombres en un asignador central de nombres. Sin embargo, este enfoque tiene varias desventajas:

- Es posible que el asignador de nombres se convierta en un cuello de botella..
- Si el asignador de nombres se cae, es posible que ninguna de las localidades del sistema distribuido pueda seguir trabajando.
- Se reduce la autonomía local, ya que la asignación de nombres se controla de forma centralizada.

Un enfoque diferente que origina una mayor autonomía local es exigir que cada localidad ponga como prefijo un identificador de localidad a cualquier nombre que genere. Esto garantiza que dos localidades nunca generarán el mismo nombre (ya que cada localidad tiene un identificador único). Además, no se requiere un control central.

Esta solución al problema de asignación de nombres, logra autonomía local, pero no transparencia de la red, ya que se agregan identificadores de localidad a los nombres. Así, la relación depósito podría llamarse localidad17.depósito en vez de depósito simplemente.

Cada copia y fragmento de un elemento de información deben tener un nombre único. Es importante que el sistema pueda determinar qué copias son copias del mismo elemento de información y qué fragmentos son fragmentos del mismo elemento de información. [3]

1.2.8 Transparencia de la repetición y la fragmentación

No es conveniente requerir que los usuarios hagan referencia a una copia específica de un elemento de información. El sistema debe ser el que determine a qué copia debe acceder cuando se le solicite su lectura, y debe modificar todas las copias cuando se produzca una petición de escritura.

Cuando se solicita un dato, no es necesario especificar la copia. El sistema utiliza una tabla-catálogo para determinar cuáles son todas las copias de ese dato.

De manera similar, no debe exigirse a los usuarios que sepan cómo está fragmentado un elemento de información. Es posible que los fragmentos verticales contengan id-tuplas, que

representan direcciones de tuplas. Los fragmentos horizontales pueden haberse obtenido por predicados de selección complejos. Por tanto, un sistema de bases de datos distribuido debe permitir las consultas que se hagan en términos de elementos de información sin fragmentar. Esto no presenta problemas graves, ya que siempre es posible reconstruir el elemento de información original a partir de sus fragmentos. Sin embargo, este proceso puede ser ineficiente. [3]

1.2.9 Transparencia de localización

Si el sistema es transparente en cuanto a repetición y fragmentación, se ocultará al usuario gran parte del esquema de la base de datos distribuida. Sin embargo, el componente de los nombres que identifican a la localidad obliga al usuario a darse cuenta del hecho de que el sistema está distribuido.

La transparencia de localización se logra creando un conjunto de seudónimos o alias para cada usuario. Así, el usuario puede referirse a los datos usando nombres sencillos que el sistema traduce a nombres completos.

Con el uso de seudónimos, no será necesario que el usuario conozca la localización física de un dato. Además, el administrador de la base de datos puede cambiar un dato de una localidad a otra sin afectar a los usuarios.

1.2.10 Esquema completo de asignación de nombres

Ya vimos que un nombre proporcionado por el usuario debe pasar por varios pasos de traducción antes de que pueda servir como referencia a una copia específica de un fragmento determinado en una localidad específica.

Para ilustrar cómo funciona el esquema, consideramos un usuario que se encuentra en la sucursal 1 (L1). Este usuario emplea el seudónimo depósito-local para el fragmento local depósito-F1 de la relación deposito. Cuando este usuario hace referencia a depósito-local, el subsistema de procesamiento de consultas busca depósito-local en la tabla de seudónimos y la sustituye por Ll.depósito.F1. Es posible que L1.depósito.Fl esté repetido.

Si es así, debe consultarse la tabla de copias para elegir una copia. Esta copia podría también estar fragmentada, lo que haría necesario consultar la tabla de fragmentación. En la mayor parte de los casos, sólo es preciso consultar una o dos tablas.

1.2.11 Transparencia y actualizaciones

De alguna forma es más difícil hacer transparente la base de datos para usuarios que la actualizan que para aquellos que sólo leen datos. El problema principal es asegurarse de que se actualizan todas las copias de un dato y también los fragmentos afectados.

En el caso más general, el problema de actualización de información repetida y fragmentada está relacionado con el problema de actualización de vistas. [3]

Ejemplo de Base de datos relacional cliente/servidor de ANSI[13].

En un sistema cliente/servidor, el cliente maneja la aplicación y la presentación de los datos, y el servidor maneja los datos. Clientes y servidores se comunican con cada uno de los otros a través de la interfaz de base de datos.

La base de datos Adaptive Server Anywhere tiene tres componentes:

- El motor de base de datos – el servidor
- Utilidades de base de datos – administrativo y herramientas query.
- Interfaces de base de datos – métodos que usan los clientes conectados a la base de datos.

Replicación:

Además del manejo de base de datos Adaptive Server Anywhere incluye construcción de replicación de datos bidireccional, designado especialmente para usuarios móviles y que se conectan ocasionalmente. Esta funcionalidad está provista por el componente SQL Remoto.

[13]ANSI El Instituto Nacional Estadounidense de Estándares (***ANSI***, por sus siglas en …

EL SERVIDOR ADAPTIVE SERVER ANYWHERE

El servidor responde a los requerimientos de las aplicaciones del cliente para el manejo de datos.

Hay dos versiones de Adaptive Server Anywhere:

El Servidor Personal Standalone.

El Servidor para Red.

Ambas versiones comparten una tecnología común de base de datos mientras ofrecen diferentes capacidades en el trabajo de red para soportar requerimientos específicos en el trabajo de base de datos.

Ambas versiones pueden participar en el sistema de replicación de Adaptive Server Anywhere, en el cual las modificaciones de los datos en una base de datos son aplicadas a datos correspondientes a otras bases de datos.

Servidor Personal Standalone.

El servidor personal opera como una información básica en el computador del usuario. Cuyo pequeño rastro permite que una copia de la base de datos puede fácilmente residir en cada máquina de los usuarios. El motor del servidor personal recibe requerimientos de recuperación, actualización, y otras operaciones en las aplicaciones del cliente en la misma máquina. Estas operaciones son ejecutadas en las tablas de las bases de datos que son almacenadas en uno o más archivos de base de datos que existen en un disco.[3]

Dependiendo de la asignación permitida a su usuario ID, los usuarios personales de base de datos pueden ejecutar todas las operaciones normales de base de datos. Por ejemplo, ellos pueden:

Seleccionar datos.

Insertar, actualizar o borrar datos.

Crear objetos de base de datos tales como tablas, almacenar procedimientos, y crear usuarios.

Respaldar datos de las base de datos.

Asignar y revocar permisos de usuarios.

El motor del servidor personal puede manejar a diez conexiones simultáneas de una o más aplicaciones.

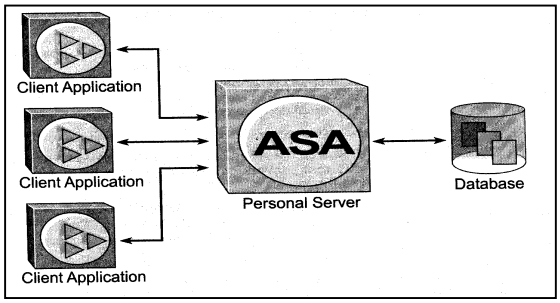


Fig. 1-02[8] Representacion de un servidor stand alone

Ejecutables standalone

El servidor personal standalone es invocado usando estos ejecutables:

Windows 95 y windows NT ----- dbeng6.exe

Unix ----- dbeng6

Windows 3.x ----- dbeng6.exe requiere el subsistema win32s.

Servidor de Red de Base de datos

El servidor de red de Adaptive Server Anywhere, recibe requerimientos para recuperación, actualización y otras operaciones de las aplicaciones del cliente a través de la red. Las aplicaciones del cliente pueden residir en cualquier máquina en la que el servidor de base de datos reside en la red.

Conexiones del Cliente

El servidor de base de datos suporta múltiples conexiones activas de clientes. Una conexión es un cliente con uno o más logins activos.

El servidor de red de base de datos puede soportar comunicaciones ilimitadas de clientes. El número actual de conexiones activas permitidas está determinado por la licencia de Adaptive Server Anywhere y la capacidad de su red y el computador servidor.

Cada uno de los login activos, cada cliente están permitidos por el número ilimitado de conexiones a la base de datos.

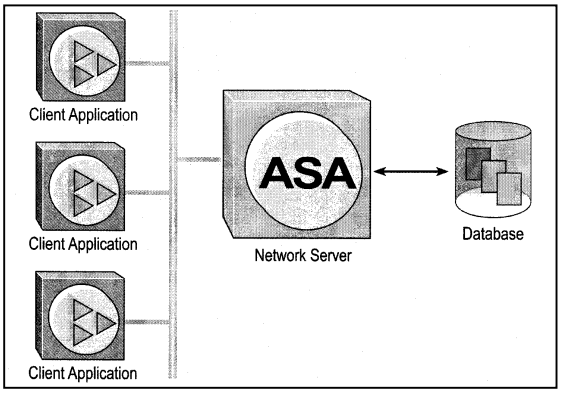


Fig. 1-03[8] Conexión con el servidor

Ejecutables del servidor de Bases de Datos de red

Windows 95 y Windows NT ------- dbsrv6.exe

Novell NetWare ------ dbsrv6.nlm

Unix ----- dbsrv6

REPLICACIÓN CON ADAPTIVE SERVER ANYWHERE

SQL Remote habilita dos o más base de datos Adaptive Server Anywhere, para participar en replicaciones bidireccionales de usuarios conectados continuamente u ocasionalmente.

Las bases de datos Adaptive Server Anywhere también pueden actuar como sitios primarios o de réplica en un sistema SybaseReplication Server.[4]

SQL Remote versus el Servidor de Replicación.

SQL Remote

Sistema de Mensaje-básico diseñado para el uso óptimo con clientes conectados ocasionalmente.

Es Bidireccional por defecto

También soporta Adaptive Server Enterprise.

Servidor de Replicación:

Asume conectividad continua.

La replicación Bidireccional debe personalizarse.

También soporta Adaptive Server Anywhere

Replicación con SQL Remote

SQL Remote replica los datos bidireccionalmente entre una base de datos consolidada y una o más base de datos remota. SQL Remote permite que las bases de datos Adaptive Server Anywhere compartan los datos e intercambia los datos modificados usando transmisiones de mensaje-basado.[4]

Réplica Bidireccional

La réplica entre las bases de datos consolidadas y las bases de datos remotas son de dos maneras. La base de datos consolidada:

Recibe los datos desde las bases de datos remotas.

Los cambios en las Consolidadas con datos ya residen en la base de datos central

Las Copias consolidaron datos a las bases de datos remotas[4]

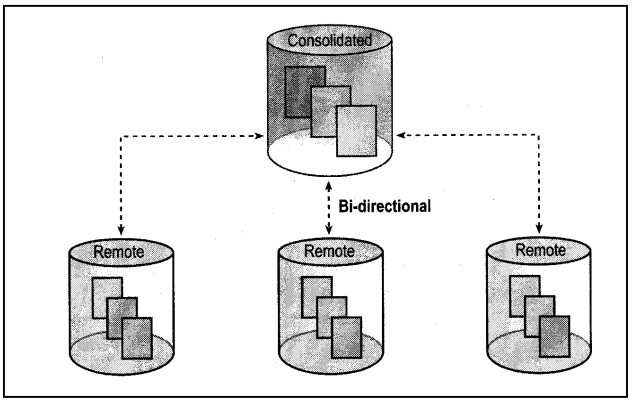


Fig. 1-04[8] Representacion de replicación remota

Datos Consolidados.

Una base de datos remota puede contener todas, alguna, o parte de las tablas que residen en sus bases de datos consolidadas. Una base de datos remota puede también contener tablas que no estén en sus bases de datos consolidadas. Sin embargo, solo las tablas contenidas en ambas; en las bases de datos remotas y consolidadas pueden ser replicadas.

Mensaje-basado en Replicación.

La replicación en Adaptive Server Anywhere es un mensaje-basado. El datos se intercambia entre la base de datos central consolidada y la base de datos remota que usan mensajes. La comunicación mensaje-basado elimina la necesidad de conectividad continua y habilita la replicación entre bases de datos conectadas ocasionalmente.[4]

Jerarquía de la Replicación.

Una base de datos consolidada puede así misma convertirse en una base de datos remota para otra base de datos consolidada en una jerarquía de replicación.

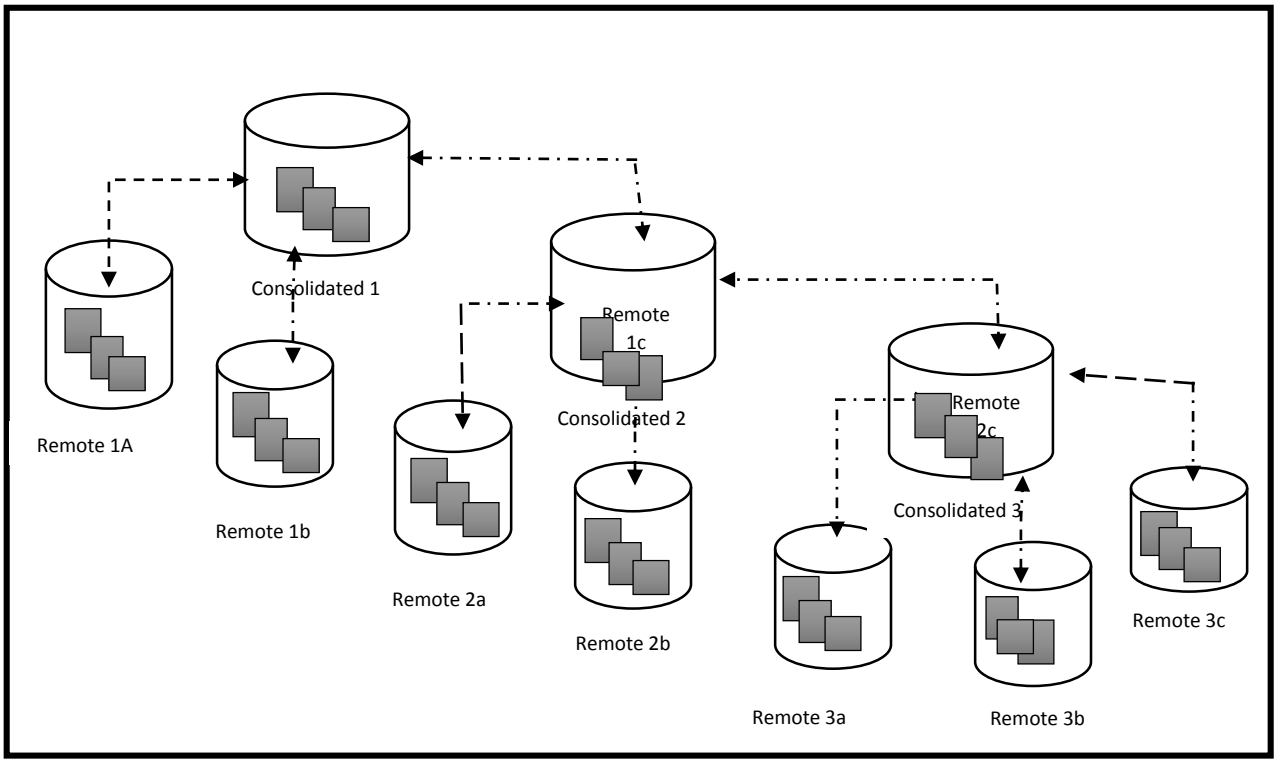


Fig. 1-05[8] Jerarquía de la Replicación.

Utilidades de la Base de Datos Adaptive Server Anywhere

Las utilidades de la Base de Datos soportan la administración de base de datos y la manipulación interactiva de los datos.

Sybase Central

Sybase Central es una herramienta de manejo gráfico de base de datos integrada que provee un completo conjunto de utilidades para el manejo de sus bases de datos y

servidores Adaptive Server Anywhere. Sybase Central no es una herramienta query de base de datos.[4]

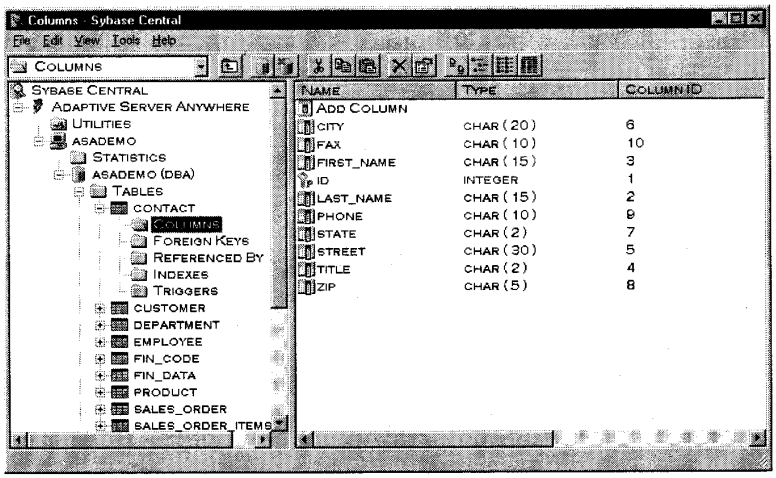


Fig. 1-06[8] Ejemplo en Sybase Central

Sybase Central corre sobre Windows o Windows Server , pero puede manejar servidores que corren en otras plataformas.Interactiva SQL.- SQL Interactivo es una utilidad donde las tenemos instrucciones de declaración para el servidor Adaptive Server Anywhere.

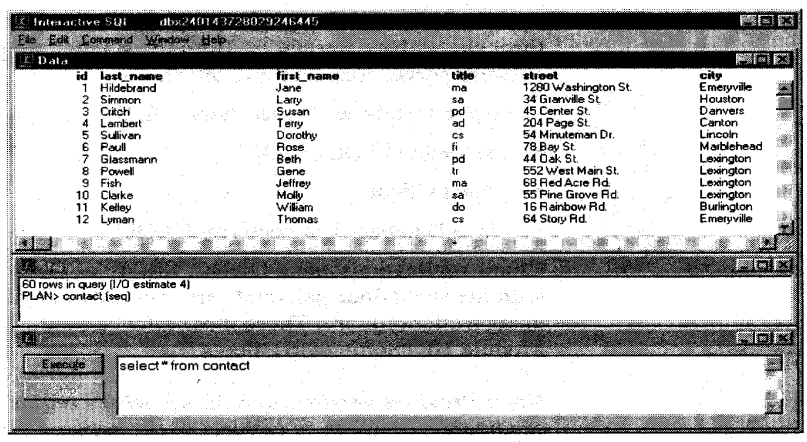


Fig. 1-07[8]Ejemplo de un servidor Adaptive Anywhere en Sybase Central

SQL interactivo también puede ser usado para ejecutar comandos de administración cuando Sybase Central no está disponible. Usted también puede encontrar conveniente escribir los scripts para manejar su base de datos. Estos scripts pueden ser ejecutados usando SQL Interactivo.

Utilidades de línea de Comando

Adaptive Server Anywhere viene con un conjunto de comandos en línea que realizan las mismas funciones como las utilidades de base de datos encontradas en las utilidades de Sybase Centras. Estas utilidades realizan funciones como:

- ✓ Backup
- ✓ Comprimir/Descomprimir
- ✓ Borrar datos de base de datos
- ✓ SQL Interactivo
- ✓ Cargar y descargar base de datos
- ✓ Validar base de datos [5]

Interfaces de Base de datos en Adaptive Server Anywhere

Los programas de clientes tales como herramientas de base de datos y aplicaciones de negocios, comunican con el personal del servidor o del servidor de la base de datos de red a través de las interfaces de las base de datos. Ambos el servidor personal y el servidor de red soportan estas interfaces:

Abrir Conectividad de base de datos (ODBF)

ESQL

Abrir Cliente

Conectividad Java de base de datos (JDBC)

Usando las aplicaciones del Cliente: ODBC, ESQL[14], o Abrir Cliente comunican con el servidor personal o con el servidor de base de datos de red directamente, usando una librería de cliente.

INTERFAZ ODBC

Open DatabaseConnectivity (ODBC[15]) es una interfaz de nivel de llamado standar, desarrollado por Microsoft. Está basado en el Grupo SQL de Access, especificación para acceso de base de datos.

[14]ESQL es una base de datos SQL relacional integrada
[15]ODBS es una interfaz de nivel de llamado estándar, desarrollado por Microsoft.

Soporte Nativo.

En Adaptive Server Anywhere 6.0, soporta el ODBC. Versiones previas de SQL Anywhere requieren un manejador separado de OBDC.

Soporte ODBC 3.0.

Adaptive Server Anywhere soporta ODBC 3.0 para lo siguiente:

Todas las características del nivel-central.

Todas las características del nivel 1, excepto para ejecuciones asincrónicas de funciones ODBC.

Todas las características del nivel 2, excepto:

- Partir la tabla en tres, y ver nombres.
- Ejecuciones Asincrónicas de funciones ODBC para instrucciones especificadas individualmente.
- Habilitar requerimientos fuera del login y pedidos SQL.

Ventajas: ODBC le permite:

- Trabajar con un API
- Desarrollar aplicaciones que pueden ser transportadas a otras fuentes de datos que soportan manejadores ODBC.[5]

Interfaz ESQL:

Es un estándar originalmente definido por IBM. Esta interfaz es de nivel-más bajo para el servidor Adaptive Server Anywhere. ESQL es una interfaz de programación nativa de Adaptive Server Anywhere. Esta soporta toda la funcionalidad de Adaptive Server Anywhere.

Una aplicación de cliente que usa ESQL tiene instrucciones SQL en sus archivos fuentes C o C++.

Open Client:

Las aplicaciones programadas usando el Sybase Open Cliente API pueden acceder directamente a Adaptive Server Anywhere. Estas aplicaciones tienen la ventaja de ser transportables a otras bases de datos SybaseAdaptive Server Anywhere, como Adaptive Server Enterprise. Para que Adaptive Server Anywhere funcione como un Open Server, este debe ser inicializado usando el protocolo de red TCP/IP.

JDBC:

Las aplicaciones Java externas pueden conectar a Adaptive Server Anywhere usando JConnect. JConnect es un producto manejador de SybaseJDBC.

Las aplicaciones internas de Java pueden ser lanzadas por usuarios de base de datos. Los métodos Java pueden ser lanzados como un procedimiento almacenado o pueden ser usados como tipos de datos especializados.

Cómo trabajan juntos los componentes de adaptive server anywhere

Sistema de Base de Datos Standalone.

Las aplicaciones clientes usan ODBC, ESQLu Open Client para requerir un servicio desde el servidor.

El servidor responde a los requerimientos del cliente a través de ODBC, ESQL, u Open Client.

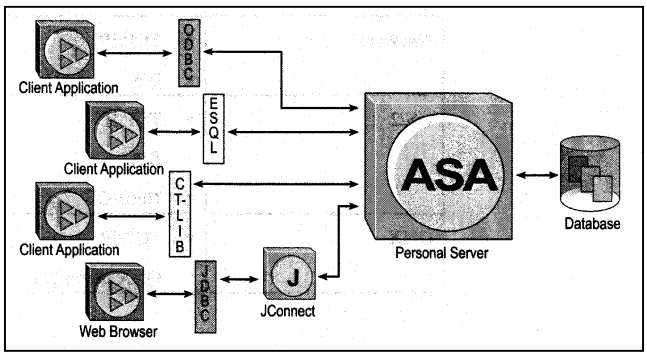


Fig. 1-08[8] Servidor stand alone en red

Sistema de base de datos de Red.

Las aplicaciones cliente usan ODBC, ESQL, u Open Cliente para requerir un servicio directamente desde el servidor de base de datos de la red.

El servidor responde a los requerimientos del cliente a través de ODBC, ESQL, u Open Client.

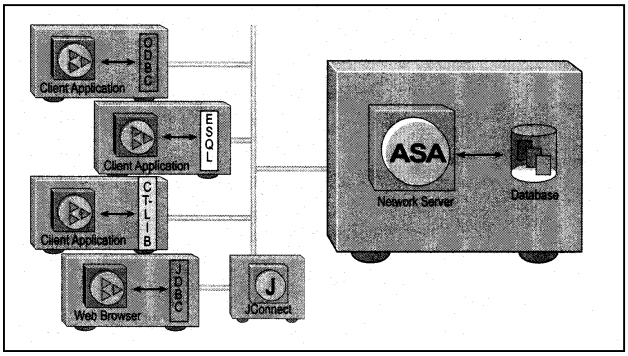


Fig. 1-09[8] Sistema de base de datos en una red

Protocolos Soportados por la Red

Los servidores de base de datos de clientes y red soportados por Adaptive Server Anywhere tienen los siguientes protocolos:

SISTEMA OPERATIVO	PROTOCOLOS SOPORTADOS EN RED
Windows 95, Windows NT	• TCP/IP (requerido si usa Open Server) • IPX • NetBIOS
UNÍS	• TCP/IP
NetWare	• TCP/IP • IPX
OS/2	• TCP/IP • IPX • NetBIOS
QNX	• TCP/IP • MensajesQNX

Ejemplo de configuración de servidor de base de datos:

Este ejemplo ilustra:

- Una aplicación cliente en Nodo 3 al Servidor en Nodo 1
- Una aplicación cliente en Nodo 4 al Servidor en Nodo 1
- Otra aplicación cliente en Nodo 4 conectado al servidor en Nodo 2
- Una aplicación cliente en Nodo 5 conectado al servidor en Nodo 2

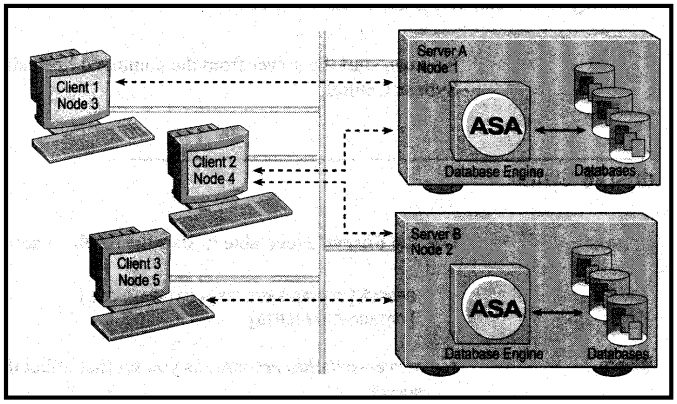


Fig. 1- 10[8] Ejemplo de Replicacion en Red

Iniciando adaptive server anywhere

El servidor personal Standalone o el servidor de base de datos de red debe ser iniciado antes que los clientes pueden empezar con los servicios de requerimientos. [6]

Una vez que los servidores o el motor soniniciado, los clientes pueden conectarse usando cualquiera de las interfaces nativas ya vistas anteriormente. – ODBC, ESQL u Open Client.

Iniciando el servidor de base de datos de red

Ud. Puede iniciar el servidor desde la línea de comando usando dbsrv6 o a través de Sybase Central.

SINTAXIS DBSRV6:

Use el ejecutable dbsrv6 para iniciar el servidor de base de datos.

DBSRV6 [SERVER-SWITCHE5] [DATABASE-FILE]

[DATABASE-SWITCHES]

Donde:

- SERVER-SWITCHE son opciones que usted pone para afectar la conducta del servidor.
- DATABASE-FILE es el nombre del archivo de base de datos que está iniciando, por ejemplo: class.db
- DATABASE-SWITCHES son opciones que ponen para afectar la conducta de la base de datos. [6]

Algunos switches básicos:

- -n {server-name | database-name} Inicia el servidor o base de datos con el nombre especificado.
- -e Habilita la encriptación de la red.
- -c num-MBs, inicia el servidor con la cantidad de caché especificada.
- -gtnum-processors, En una máquina multiprocesadora, inicia el servidor usando el número de procesadores especificados.
- b renombra el viejo login a un nuevo nombre e inicia el servidor con una nuevo login. Usado durante la replicación.
- @filename Especifica un archivo opcional donde usted tiene almacenado su seteos de conexión. [6]

1.2.12 El archivo principal de base de datos

El archivo principal de base de datos contiene tablas de usuarios, índices, views, y tablas del sistema, los cuales contienen un diccionario de datos y la información almacenada.[7]

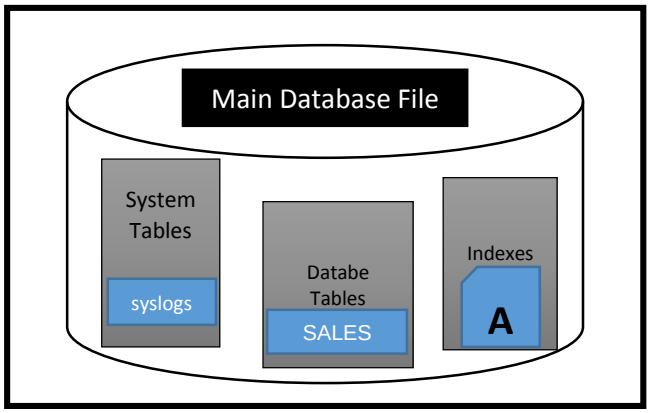


Fig. 1- 11[8] Archivo central de base de datos

El archivo principal de base de datos es creado con la utilidad de inicialización de base de datos, dbinit, o usando el comando CREATEDATABASE. Puede tener acceso a dbinit desde la línea de comando del sistema operativo o empleando automáticamente la utilidad CREATEDATABASE de Sybase Central CreateDatabase.

Inicializando un archivo principal de base de datos automáticamente crea tablas del sistema.

El archivo principal de base de datos puede ser hasta de 2GB de tamaño.[7]

Archivos adicionales de base de datos (DBSPACES)

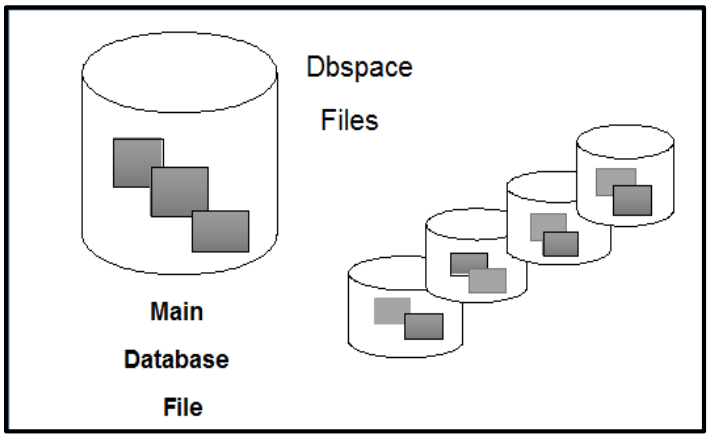


Fig. 1-12[8] Diagrama de archivos adicionales en DBspaces

Inicializando una base de datos crea un archivo de disco con la extensión DB. El archivo principal de base de datos tiene un límite superior de 2 GB.

El tamaño de una base de datos puede ser extendida más allá de su límite con archivos adicionales. Estos archivos son internamente designados como dbspaces. Una base de datos puede constar de un archivo de base de datos y sobre once archivos de dbspace. Estos archivos pueden residir en el mismo o en diferente dispositivo del archivo de la base de datos principal.

1.2.13 Una base de datos lógica

El archivo principal de base de datos unido con los archivos asociados de dbspace, constituyen una base de datos lógica. El archivo de base de datos principal es el rootdatabase file.

El archivo principal de base de datos contiene las tablas del sistema para la base de datos entera. La tabla del sistema SYS.SYSFILE guarda un registro de todos los archivos dbspace asociados con el archivo principal de base de datos.

1.2.13.1 Ejecución

Además de extender el tamaño de la base de datos, los archivos dbspace pueden mejorar la ejecución de las operaciones de la base de datos distribuyendo discos de I/O por múltiples dispositivos y/o controladores.

Se puede variar los nombres para las computadoras que forman parte de un sistema distribuido, como pueden ser los sitios o nodos. Se indica la distribución física de estos sistemas con el término sitio.

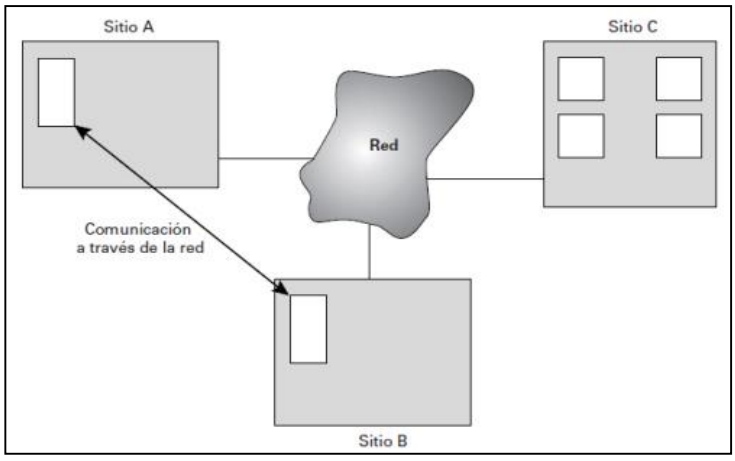


Fig. 1-13[B] ejecución en red de la replicacion

Se puede indicar que las BDD normalmente se encuentran en varios lugares geográficos distintos, se administran de forma separada y poseen una interconexión más lenta.

-Cada sitio es autónomo en sus capacidades de procesamiento

-Capaz de realizar operaciones locales [8]

Componentes:

Software

1.SGBDD local

2.Administración de transacciones distribuidas

3.Sistema manejador de base de datos

Hardware

Se reduce a servidores y la red.

Como transacción local se conoce a aquella que accede a los datos del único sitio en el cual se inició la transacción. Se conoce en cambio una transacción global es aquella que, o bien accede a los datos situados en un sitio diferente de aquel en el que se inició la transacción, o bien accede a datos de varios sitios distintos.[8]

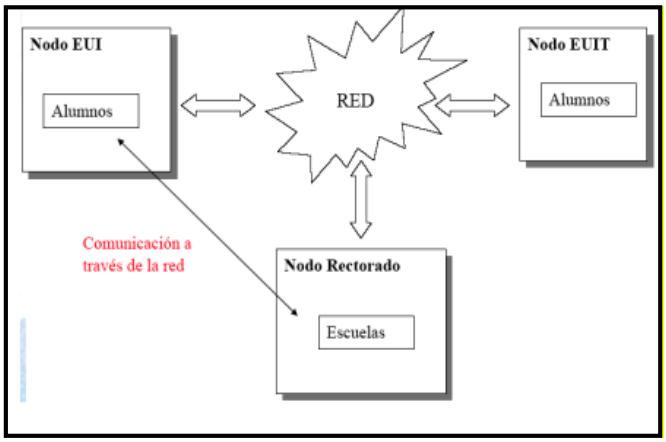


Fig. 1-14[B] Ejemplo de nodos en red

1.2.13.2 Ventajas de la Base de datos distribuida:

Eficiencia en el procesamiento

-Mayor accesibilidad

-Fiabilidad y disponibilidad

-Mejor rendimiento

1.3 Bases de datos de alta disponibilidad.

La utilización de las bases de datos y el aumento de la utilización de Internet han permitido la colaboración y el intercambio de información desde cualquier parte del mundo, incrementando el alcance de las aplicaciones de bases de datos en todas las organizaciones y comunidades. Este alcance nos permite ver la importancia de la alta disponibilidad en soluciones de gestión de datos.

Las empresas del mundo cuentan con usuarios que necesitan acceso a los datos las 24 horas. Esto es de suma importancia porque sin este acceso a los datos las operaciones pueden detenerse.

La importancia de la alta disponibilidad varía dependiendo de las aplicaciones. Pero es una necesidad seguir aumentando los niveles de disponibilidad. Cuando los datos no están disponibles, la aplicación puede dejar de funcionar.

La alta disponibilidad tiene en cuenta criterios como, duración, frecuencia o el impacto cuando algo falla. El objetivo general de la alta disponibilidad es hacer un sistema tolerante a fallas. Para medir la disponibilidad se hace en tres componentes generales:

- El número de fallas y de mantenimientos que ocurren en un sistema en un periodo de tiempo.
- La robustez: es el grado de protección de un sistema o aplicación ante un evento de falla del sistema, permitiéndole continuar disponible cuando se presenta la falla.
- La recuperación: es la velocidad en tiempo que se demora un sistema o aplicación para retornar a su estado normal después de que ha ocurrido un fallo.

Siempre tenemos que prever que nuestra base de datos puede crecer hasta el punto de que tanto la carga del servidor como el espacio puede llegar a ser un problema. Por eso mismo, una configuración en clúster puede ayudarnos a resolver problemas en un futuro. Una de las soluciones para conseguir esto es montar la base de datos en un clúster de tal forma que no haya ningún punto débil en el sistema y si se cae un nodo que el sistema no deje de dar servicio. La disponibilidad juega un papel de suma importancia aquí porque la caída de la base de datos puede llegar a crear pérdidas cuantitativas.

CAPÍTULO II

SELECCIÓN DE UN DBMS QUE SOPORTEN DISTRIBUCIÓN.

2.1 Infraestructura.

La arquitectura de los sistemas de bases de datos establecida por ANSI-SPARC utiliza tres niveles de abstracción: externo, conceptual e interno. En el nivel externo, el esquema consta de las distintas visiones que tienen los usuarios de la base de datos. En el nivel conceptual, el esquema es la visión común de la base de datos. Especifica el contenido de información de la base de datos independientemente de las consideraciones de almacenamiento. En el nivel interno, el esquema es la visión que el ordenador tiene de la base de datos. Específica cómo se representan los datos, en qué orden se almacenan los registros, qué índices y punteros se han creado y qué esquema de dispersión se ha utilizado, si es el caso.(1)

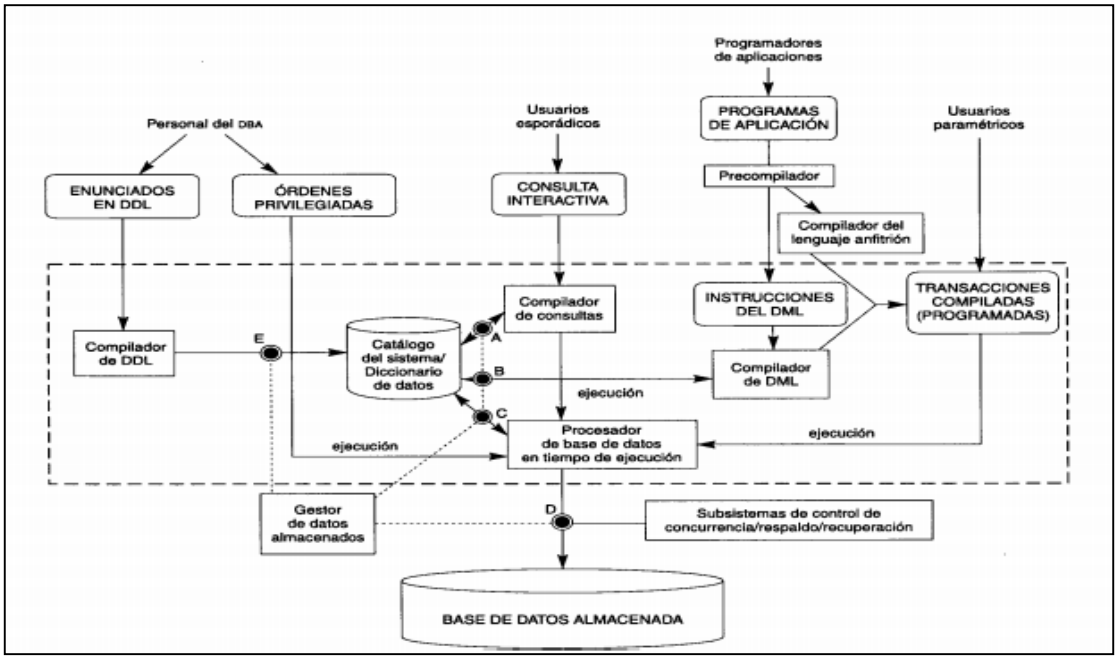


Fig.2-01[6] Infraestructura detallada de la base de datos

2.2 Hardware

Los componentes de hardware del sistema constan de:
Los volúmenes de almacenamiento secundario, como discos magnéticos, que se emplean para contener los datos almacenados, junto con dispositivos asociados de E/S, los controladores de dispositivos, los canales de E/S, entre otros.
Los procesadores de hardware y la memoria principal asociada usados para apoyarla ejecución del software del sistema de base de datos(1)

2.3 Software

El administrador de base de datos o servidor de base de datos conocido como sistema de administración de base de datos (DBMS) maneja todas las solicitudes de acceso a la base de datos ya sea para agregar y eliminar archivos, recuperar y almacenar datos desde y en dichos archivos. Por lo tanto, una función general que ofrece el DBMS consiste en ocultar a los usuarios de la base de datos los detalles al nivel de hardware. Es decir, que el DBMS ofrece a los usuarios una percepción de la base de datos que está en cierto modo, por encima del nivel del hardware y que maneja las operaciones del usuario expresadas en términos de ese nivel más alto de percepción.(1)
El DBMS es el componente de software más importante del sistema en general, aunque no es el único.
Un "sistema de gestión de base de datos" (DBMS) es un conjunto de programas informáticos que proporciona la interfaz entre el usuario y una base de datos o bases de datos. Debido a que están tan estrechamente relacionados, el término "base de datos" cuando se utiliza casualmente a menudo se refiere tanto a un DBMS y los datos que manipula.(1)

Se determinó que una base de datos es una colección organizada de datos . Por lo general los datos suelen estar organizados para modelar los aspectos relevantes de la realidad de una manera que apoye los procesos que requieren esta información. Por ejemplo, el modelado de la disponibilidad de habitaciones en los hoteles de una manera que apoye la búsqueda de un hotel con habitaciones libres.(1)

Se puede determinar que la mayoría de bases de datos tienen software de la aplicación que tiene acceso a la base de datos en nombre de los usuarios finales, sin exponer la interfaz DBMS directamente. Es así que los programadores de aplicaciones pueden usar un protocolo de conexión directamente, o más probablemente a través de una interfaz de programación de aplicaciones . Los diseñadores de bases de datos y administradores de bases de datos interactúan con el DBMS a través de interfaces dedicadas a construir y mantener bases de datos ", y por lo tanto necesita un poco más de conocimiento y la comprensión de cómo funcionan los DBMS y el DBMS 'las aplicaciones interfaces externas y los parámetros de ajuste.(1)

2.3 Motores de base de datos que soporten distribución.

- El Motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos. El Motor de base de datos proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa.
- Use Motor de base de datos para crear bases de datos relacionales para el procesamiento de transacciones en línea o datos de procesamiento analíticos en línea. Se pueden crear tablas para almacenar datos y objetos de base de datos como índices, vistas y procedimientos almacenados para ver, administrar y proteger los datos. Puede usar SQL[16] Server para administrar los objetos de bases de datos

[16] SQL Lenguaje de consulta estructurado

PERVASIVESQL

Motor de base de datos imbebible que sustenta la integridad de los datos, el altorendimiento, flexibilidad, escalabilidad y un bajo coste total de propiedad.(5)

Características

- Rendimiento de Btrieve- API transactional para un acceso más rápido a los datos.
- Optimizador de SQL - proporciona un mayor rendimiento SQL. ODBC, JDBC, ADO / OLE DB y PDAC todos se aprovechan de estas mejoras.
- Turbo WriteAccelerator - maximiza la eficacia de las E/S del disco.
- Compatibilidad retroactiva completa - aplicaciones y bases de datos heredadas.
- Capacidad de incrustación: Biblioteca de desarrollo de clase.
- Caja de herramientas de instalación completa(3)

Operaciones continuas

Tecnología de reconexión automática. Evita problemas menores de red porque es capaz de reconectar perfectamente puestos de trabajo a un motor de servidor cuando se produce un error en la red

Parámetros dinámicos en plataformas cruzadas - no es necesario volver a actualizar el entorno de la base de datos en cada plataforma.

Integridad en plataformas cruzadas - el comportamiento igual en todas las versiones y plataformas simplifica la distribución y administración en plataformas cruzadas.

Independencia de versión de servidor y cliente - Simplifica de forma extraordinaria las tareas de distribución.(4)

2.4 Selección de un motor para manejo de una de base de datos distribuida.

Se puede indicar en base de un sistema distribuido indicando que es el que toma las decisiones sobre la ubicación de datos y aplicaciones a lo largo de una red. Cuando se trata de un DDBMS los factores de distribución son: la distribución del software del DDBMS y la distribución de los programas de aplicación que corren sobre él. Siendo estos aspectos de carácter importante son el eje para el estudio del diseño de una BDD. Es por este motivo que las decisiones de diseño relacionadas con el software del DDBMS y de los programas de aplicación.

Cuando se habla de diseño conceptual de la BD este involucra decisiones de diseño que van más allá de la implementación física que tendrá el modelo. Pueden darse diferentes técnicas para el modelado de dato, pero no se toma en cuenta, al menos en una primera etapa, los detalles físicos de implementación. Ya definido el esquema conceptual global y el patrón de acceso a la información, se desarrolla el diseño de la distribución de la información.

El esquema de distribución homogéneo se caracteriza por poseer un único modelo de dato, esquema conceptual global, y varios esquemas conceptuales locales, los cuales deberán concordar con el modelo global: son solamente una copia parcial de estos. El objetivo del diseño de la BDD es decidir cada uno de estos esquemas locales. Los pasos involucrados en esta etapa son la fragmentación y la replicación de la información. Es de manera especial en cuenta que cada esquema local detalla la información que reside en ese nodo de la red.

Indicándose que, cada nodo tiene una copia del esquema global que le permite conocer el esquema de fragmentación y replicación de información (metadato). (Gray et al.,1996)

Dentro de las ventajas de la base de datos distribuida se puede indicar que esta refleja una estructura organizacional por cuanto los fragmentos de la base de datos se ubican en los departamentos a los que tienen relación. Además presenta una autonomía local: por cuanto un departamento puede controlar los datos que le pertenecen. Presenta además una disponibilidad por cuanto un fallo en una parte del sistema solo afectará a un fragmento, en lugar de a toda la base de datos. La base de datos distribuída presenta además rendimiento

por cuanto los datos se ubican cercanos del sitio con mayor demanda, también los sistemas trabajan en paralelo, lo cual permite balancear la carga en los servidores. Presenta también economía por cuanto es más barato crear una red de muchas computadoras pequeñas, que tener una sola computadora muy poderosa. Esta base de datos distribuida también determina modularidad esto quiere decir que se pueden modificar, agregar o quitar sistemas de la base de datos distribuida sin afectar a los demás sistemas (módulos) (4)

Dentro de las desventajas de la base de datos distribuida se puede determinar en la economía por cuanto la complejidad y la infraestructura necesaria implica que se necesitará una mayor mano de obra. Presenta desventaja también en la seguridad por cuanto se debe trabajar en la seguridad de la infraestructura así como cada uno de los sistemas. Otra de las desventajas se da en la Integridad por cuanto esta se vuelve difícil mantener la integridad, aplicar las reglas de integridad a través de la red puede ser muy caro en términos de transmisión de datos. Se puede determinar también la falta de experiencia en donde las bases de datos distribuidas son un campo relativamente nuevo y poco común por lo cual no existe mucho personal con experiencia o conocimientos adecuados. (4)

Otra desventaja es que el diseño de la base de datos se vuelve más complejo por cuanto además de las dificultades que generalmente se encuentran al diseñar una base de datos, el diseño de una base de datos distribuida debe considerar la fragmentación, replicación y ubicación de los fragmentos en sitios específicos.

Sistema manipulador de base de datos distribuida (DDBMS)

Este sistema está formado por las transacciones y los administradores de la base de datos distribuidos. Un DDBMS implica un conjunto de programas que operan en diversas computadoras, estos programas pueden ser subsistemas de un único DDBMS de un fabricante o podría consistir de una colección de programas de diferentes fuentes. (2)

Administrador de transacciones distribuidas (DTM)

Este es un programa que recibe las solicitudes de procesamiento de los programas de consulta o transacciones y las traduce en acciones para los administradores de la base de datos. Los DTM se encargan de coordinar y controlar estas acciones. Este DTM puede ser propietario o libre. (3)

Sistema manipulador de base de datos (DBMS)

Es un programa que procesa cierta porción de la base de datos distribuida. Se encarga de recuperar y actualizar datos del usuario y generales de acuerdo con los comandos recibidos de los DTM. (3)

Nodo

Un nodo es una computadora que ejecuta un DTM o un DBM o ambos. Un nodo de transacción ejecuta un DTM y un nodo de base de datos ejecuta un DBM. (3)

2.4.1 Motores de base de datos que soporten distribución.

Motores de búsqueda: aspectos generales

Terminología:

-Robots, arañas (spiders),vagabundos (wanders) hormigas web

(WebAnts).

Definición:

- ✓ Agentes que atraviesan de forma automática la WWW con distintos fines:
- ✓ Mantener la estructura hipertextual del Web.
- ✓ Duplicar directorios FTP (mirrors).
- ✓ Realizar análisis estadísticos del Web.
- ✓ Descubrir nuevas aportaciones (What's new) y
- ✓ Crear de forma automática bases de datos textuales a partir de los documentos HTML distribuidos por los distintos servidores.

- El Motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos. El Motor de base de datos proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa.
- Use Motor de base de datos para crear bases de datos relacionales para el procesamiento de transacciones en línea o datos de procesamiento analíticos en línea. Se pueden crear tablas para almacenar datos y objetos de base de datos como índices, vistas y procedimientos almacenados para ver, administrar y proteger los datos. Puede usar SQL[17] Server Management Studio para administrar los objetos de bases de datos y SQL Server Profiler para capturar eventos de servidor. (4)

Motor de Bases de Datos MySQL

Se puede determinar a MySQL[18] como un motor de Base de Datos relacional, multihilo y multiusuario el cual en su mayor parte está desarrollado en ANSI C. Oracle Corporation desde abril de 2009 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.

A diferencia de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius (1)

[17] SQL Lenguaje de consulta estructurado
[18] MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones

	Creador	Fecha de la primera versión pública	Última versión estable	Licencia de software
DB2	IBM	1982	9	Propietario
Informix	Informix Software	1985	10.0	Propietario
HSQLDB	Hsqldb.Org	2001	1.9	Licencia BSD
Microsoft SQL Server	Microsoft	1989	2008	Propietario
MySQL	MySQL AB	Noviembre de 1996	5.0	GPL o propietario
Oracle	Oracle Corporation	1977	11g Release 2	Propietario
PostgreSQL	PostgreSQL Global Development Group	Junio de 1989	9.0	Licencia BSD
SmallSQL	SmallSQL	16 de abril de 2005	0.12	LGPL

Fig.2-02[5] SGBD qye soportan distribución



Fig.2-03[5] Comparacion de fortalezas y debilidades de los SGBD

2.5 Selección de un motor para manejo de una de base de datos distribuida.

De acuerdo a la investigación se determina que el Motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos. Se puede indicar además que el Motor de base de datos proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de la empresa.

Cuando se utiliza un Motor de base de datos para crear bases de datos relacionales para el procesamiento de transacciones en línea o datos de procesamiento analíticos en línea. Se pueden crear tablas para almacenar datos y objetos de base de datos como índices, vistas y procedimientos almacenados para ver, administrar y proteger los datos. Puede usar SQL Server Management Studio para administrar los objetos de bases de datos y SQL Server Profiler para capturar eventos de servidor.

2.5.1 Otros motores de búsqueda

Se puede indicar a YaCy que constituye un motor de búsqueda gratuito que cualquiera puede utilizar para crear un portal de búsqueda para su intranet o para ayudar a buscar en la Internet pública. Al contribuir a la red P2P global, la escala de YaCy sólo está limitada por el número de usuarios en todo el mundo y es posible indexar miles de millones de páginas web. Es totalmente descentralizado, todos los usuarios de la red de motores de búsqueda son iguales, la red no almacena las solicitudes de búsqueda de los usuarios y no es posible que nadie pueda censurar el contenido de los índices compartidos.

2.6 Software adicional para la simplificación de la implementación

Constituye un software que permita interactuar con la información almacenada, ya que el diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. En donde la complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente,

utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

COMPONENTES DE UNA BDD Software: Sistema Manejador de Base de Datos Distribuida (DDBMS) Este sistema está formado por las transacciones y los administradores de la base de datos distribuidos. Un DDBMS implica un conjunto de programas que operan en diversas computadoras, estos programas pueden ser subsistemas de un único DDBMS de un fabricante o podría consistir de una colección de programas de diferentes fuentes. (1)

COMPONENTES DE UNA BBD Software: Administrador de transacciones distribuidas (DTM) Este es un programa que recibe las solicitudes de procesamiento de los programas de consulta o transacciones y las traduce en acciones para los administradores de la base de datos. Los DTM se encargan de coordinar y controlar estas acciones. Este DTM puede ser propietario o desarrollado en casa. (1)

COMPONENTES DE UNA BBD Software: Sistema Manejador de base de datos (DBMS) Es un programa que procesa cierta porción de la base de datos distribuida. Se encarga de recuperar y actualizar datos del usuario y generales de acuerdo con los comandos recibidos de los DTM.

COMPONENTES DE UNA BBD Software Nodo: Un nodo es una computadora que ejecuta un DTM o un DBM o ambos. Un nodo de transacción ejecuta un DTM y un nodo de base de datos ejecuta un DBM (1)

Una base de datos lógica

El archivo principal de base de datos unido con los archivos asociados de dbspace, constituyen una base de datos lógica. El archivo de base de datos principal es el **root database file.**

El archivo principal de base de datos contiene las tablas del sistema para la base de datos entera. La tabla del sistema *SYS.SYSFILE* guarda un registro de todos los archivos dbspace asociados con el archivo principal de base de datos.

2.6.1 Diseño de la base de datos

Consiste en el proceso de crear un diseño de Base de Datos que permite ayudar a las operaciones y objetivos de la empresa. Las metas claves del diseño son:

- Permite representar y relacionar datos requeridos por las áreas en general que pueden ser de aplicaciones y grupos de usuarios.

- Determinar un modelo de datos que ayude a las transacciones requeridas en los datos.

- Aplicar un diseño mínimo el cual, estructurado debidamente, pueda obtener los rendimientos identificados en los requerimientos.

Ejecución

Además de extender el tamaño de la base de datos, los archivos dbspace pueden mejorar la ejecución de las operaciones de la base de datos distribuyendo discos de I/O por múltiples dispositivos y/o controladores

Para este nivel se realizarán los siguientes pasos:

- Se deberá realizar la construcción y validación del modelo de datos lógico local para cada requerimiento de usuario
- Además de se deberá también realizar la construcción y validación del modelo de datos lógico global

Se puede indicar que el objetivo a este nivel es refinar el diseño conceptual para eliminar las características no deseadas con lo que permitirá, transformar este modelo al modelo de datos lógico local.

Ya para este nivel se construye el Diagrama E-R final, además se refina el creado en el diseño conceptual. Se deberá analizar los tipos de relaciones existentes entre las partes relacionadas, de modo de eliminar aquellas que causen problemas, como son las relaciones muchos-a-muchos, atributos multivalóricos, es definitiva, todas aquellas relaciones que producen inconsistencias. Es así que en este nivel se realizan los esquemas para las entidades presentes. Se deben identificar las relaciones lógicas.

Cuando se define el proceso de normalización este incluye:

Se puede indicar que la Primera Forma Normal (1FN) que elimina grupos repitentes.

En la Segunda Forma Normal (2FN) que elimina dependencias parciales en clave primaria

Y en la Tercera Forma Normal (3FN) que elimina las dependencias transitivas en la clave primaria

En este nivel se definen las restricciones de integridad que permiten anular la inconsistencia en la base de datos.

Al Final de este capitulo analizando los motores disponibles he decidido utilizar Mysql con su versión en Cluster 5.5 en virtualización, debido a la versatilidad de este motor de base de datos y la ligueresa de la misma, adicionalmente se utilizara maquitas virtuales de virtualbox las cuales son totalmente portables y podremos movilizarlas de una manera muy eficaz.

CAPÍTULO III

INSTALACIÓN DE MYSQL CLÚSTER

3.1 Introducción

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.
El término clúster (que viene de la palabra en inglés *cluster*, "grupo" o "racimo") se aplica a los conjuntos o conglomerados de ordenadores construidos a través de la utilización de hardwares comunes y que se comportan como si fuesen un único ordenador.

La tecnología de clústeres ha avanzado en apoyo de actividades que pueden ir desde aplicaciones de supercómputo y software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos que pueden tener.
Un clúster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

MySQL clúster es una tecnología que permite el clustering de bases de datos en memoria en un ambiente de no compartición. La arquitectura de no compartición permite que el sistema gestor de base de datos (SGBD) funcione utilizando un hardware que no sea muy costoso y con requerimientos mínimos tanto de software como de hardware.
Como todo sistema de clustering, está diseñado para no tener un sólo punto de falla, cada componente tiene su propia porción de disco y memoria para trabajar. Bajo este esquema no se recomienda el uso de mecanismos de almacenamiento compartido como carpetas compartidas por red, sistemas de archivos de red.
En otras palabras Mysql clúster no es más que un sistema de gestión de base de datos en múltiples ordenadores.

Un Clúster MySQL es un sistema donde los datos introducidos de las bases de datos y de las tablas se quedarán guardados de manera redundante en diferentes nodos del Clúster y si un nodo se cae o resulta con problemas, los demás seguirán realizando el servicio. Un Clúster MySQL está compuesto por tres componentes principales:

1. El Management node que brinda gestión al Clúster MySQL.

2. Los nodos y grupos de nodos donde serán archivados los datos de las tablas.
3. Los nodos SQL en el cual será ejecutado el servidor MySQL, siendo este conectado al Clúster y donde se darán a lugar las consultas a las bases de datos ya propuestas o existentes.

Un clúster MySQL tiene una arquitectura básica la cual está integrada por un servidor MySQL estándar y un motor de almacenamiento en memoria llamado NDB clúster, funcionando en un conjunto de uno o más ordenadores. Cada uno de estos ordenadores se encuentran ejecutando uno o más procesos, que pueden ser procesos de MySQLserver, nodos de almacenamiento de datos, servidor administrador del clúster o programas especializados para acceder a los datos.

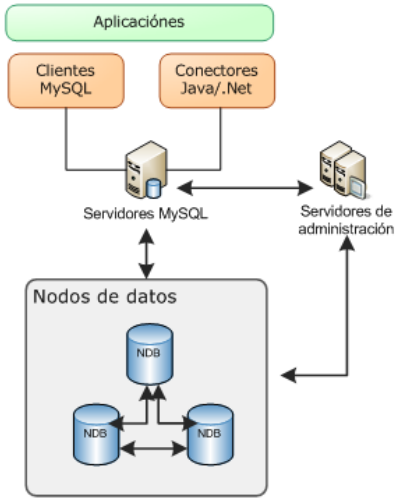


Fig.3-01[D] Representacion de Un Cluster

3.2 Infraestructura

Nuestro entorno implementará las características de MySQL Clúster 7, con una topología como la que presentamos a continuación conformada por tres nodos:

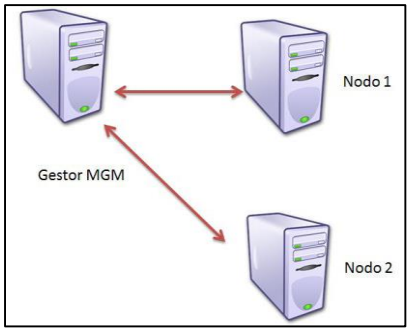


Fig.3-02[D] Representacion de nodos

NDBCLUSTER (o **NDB**) será el motor de almacenamiento más recomendado para la alta disponibilidad y tolerancia a los fallos propuesta por MySQL de tal manera va a ser él que se usará en esta instalación.

En total tendremos 3 nodos:

- Nodo de gestión o MGMT
- Nodo de datos y sql 1
- Nodo de datos y sql 2

El nodo de gestión o MGT cumple con la función de manejar, controlar y coordinar los otros nodos dentro del clúster. Implementa funciones de configuración de datos, iniciar o detener otros nodos dentro del clúster, ejecutar respaldos u otras tareas.

Los nodos de datos como su nombre lo dice son los encargados de almacenar los datos.

Lo más conveniente para un mejor desempeño de nuestro sistema sería ampliar la cantidad de nodos y así llevar a cabo una prueba real. Pudiera haber un inconveniente y es que esto consumiría una cierta cantidad de recursos bastante elevados.

3.3 Creación de la máquina virtual en virtual Box.

Para crear una Máquina Virtual seguiremos los siguientes pasos:

La primera ventana nos pide:

- **Nombre de la Máquina Virtual**: Se recomienda que éste sea significativo.
- **Tipo**: Indicamos el tipo de SO a virtualizar (En nuestro caso será Microsoft Windows).
- **Versión**: Este valor es importante, pues dependiendo de lo que se elija, Virtual Box activará ciertas opciones internas para que la virtualización pueda hacerse correctamente.

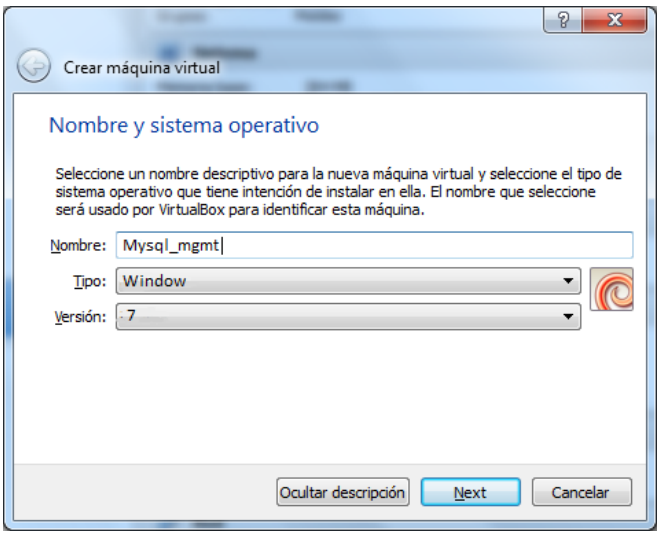


Fig.3-03[D] Instalacion de la Maquina Virtual

En esta ventana veremos el tamaño de memoria RAM que Virtual Box debe reservar de la RAM física para asignársela a la Máquina Virtual cada vez que se ejecute. Es importante tener en cuenta que esta memoria se restará de la RAM que puede utilizar la máquina anfitriona una vez que la Máquina Virtual esté corriendo, y el rendimiento del equipo anfitrión irá bajando a medida que aumente el número de MMVV en ejecución simultánea.

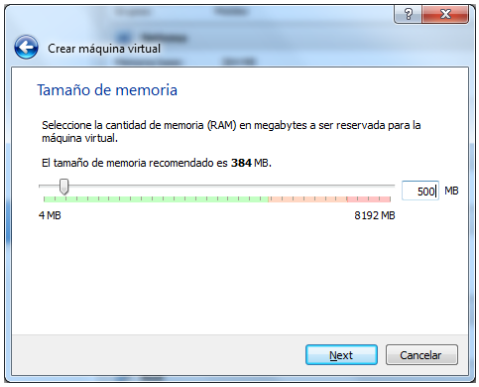


Fig.3-04 [D] Elijiendo el disco duro virtual

La siguiente ventana es sobre la unidad de disco duro que vamos a utilizar con la Máquina Virtual. Nos indica el tamaño mínimo de disco según el SO elegido.

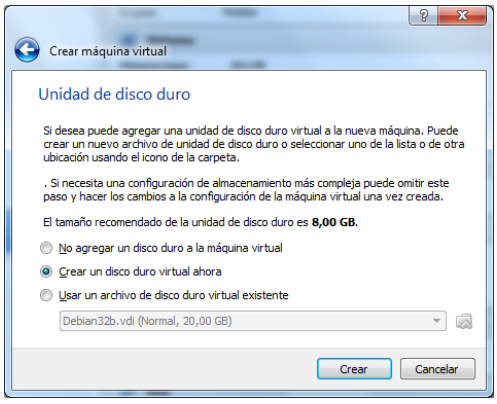


Fig.3-05[D] Tipo de disco duro virtual

A continuación elegimos como tipo de disco virtual el VDI, que es el genuino de Virtual Box.

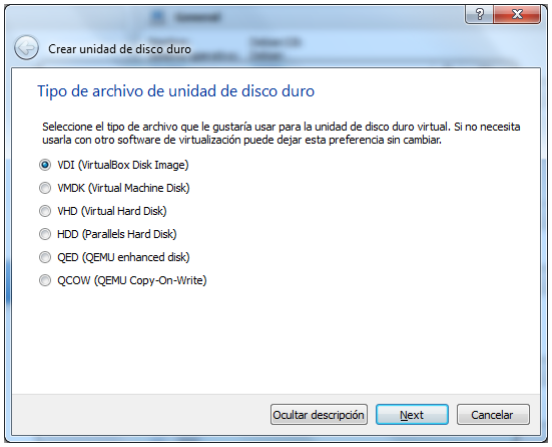


Fig.3-06[D] Eleccion del formato de disco duro virtual

Después seleccionamos que el disco sea dinámico, es decir, el fichero que virtualizará al disco duro de la Máquina Virtual y que se guardará en alguna carpeta del disco de la máquina anfitriona, irá creciendo conforme se vayan grabando cosas en él.

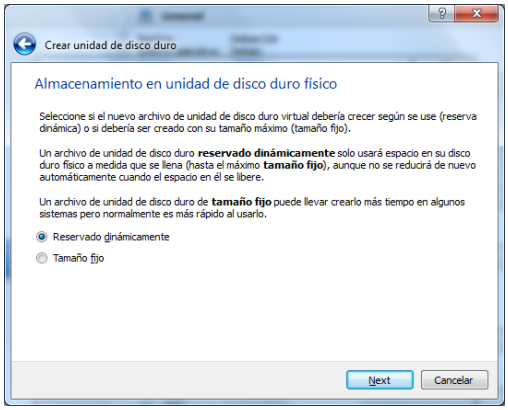


Fig.3-07[D] Eleccion dinámica de disco duro

Por último nos piden:

Nombre del archivo donde se guardará el disco virtual en la máquina anfitriona*:* continuamos con el nombre propuesto, que coincide con el nombre de la Máquina Virtual. También tenemos la opción de cambiarlo a una ubicación distinta a la que se utiliza por defecto.

Tamaño del disco*: Cambiamos el tamaño que se nos da por defecto y lo aumentamos.*

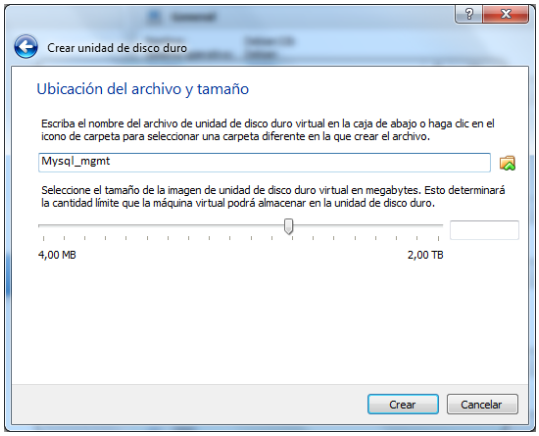


Fig.3-08[D] Selección del tamaño del disco virtual

Una vez creada nuestra primera máquina virtual, entonces pasaremos al siguiente paso, pues tendremos que clonar las dos. Este paso será uno de los más sencillos a realizar.

1. En este caso llamaremos a nuestra primera máquina virtual ¨mysql_nodo1¨.

2. A la segunda la llamaremos ¨mysql_nodo2¨.
3. A la tercera la llamaremos ¨mysql_mgmt¨ ya que esta será la encargada del proceso de gestionamiento.

La clonación de una Máquina Virtual (MV) no es más que hacer una nueva Máquina Virtual igual a la primera pero con identidad diferente, ya que los discos virtuales van a tener números de identificación diferentes, de esta manera la Máquina Virtual original y la clonada se podrían ejecutar en el mismo ordenador. Puede hacerse una Máquina Virtual a partir de la copia del fichero del disco virtual de otra, pero en este caso no podrán ejecutarse las dos Máquinas Virtuales en el mismo ordenador al haber dos discos con el mismo número de identificación.

La clonación tiene un uso muy común, el de hacer una copia de seguridad de una Máquina Virtual recién creada, así, después de la instalación de las Guest Additions, es habitual clonar la Máquina Virtual para disponer de una copia exacta en el momento de su creación, por si se produce algún desastre con la original.

La clonación debe hacerse con la Máquina Virtual apagada.

Máquina. ➡ **Clonar.**

En esta ventana se selecciona el nombre de la máquina virtual.

La opción Reiniciar la dirección MAC de todas las tarjetas de red. Se seleccionará en dependencia de si la Máquina Virtual original y la clonada van a trabajar en la misma red local o no, pues no debe haber dos tarjetas de red con la misma MAC en una misma red local.

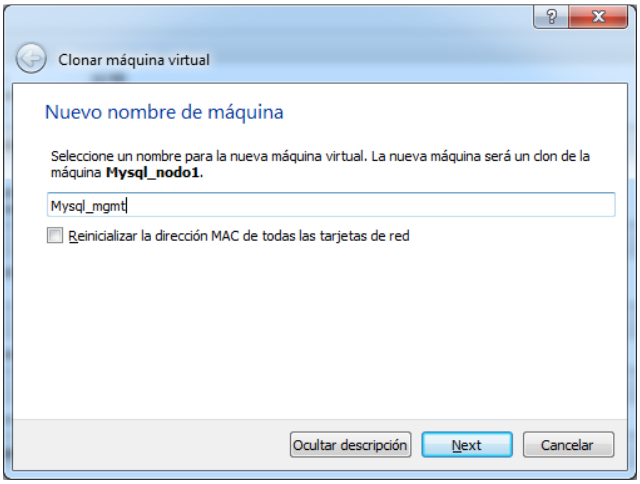


Fig.3-09[D] Clonacion de Maquina Virtual

A continuación elegimos una clonación completa, creamos así Máquinas Virtuales totalmente independientes.

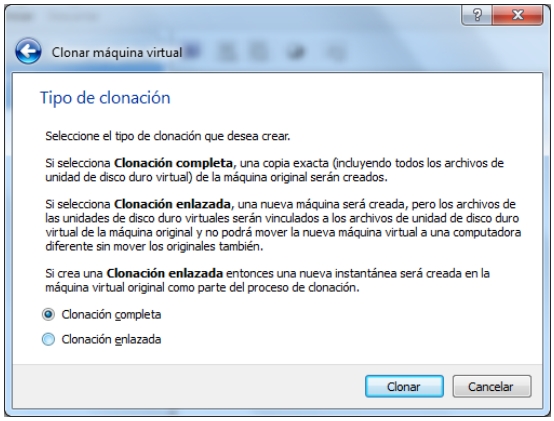


Fig.3-10[D] Clonacion independiente de maquina virtual

Pasaremos a introducir las siguientes direcciones IP para cada una de nuestras máquinas.

- **Nodo gestor:** 192.168.1.141
- **Nodo No. 1:** 192.168.1.142
- **Nodo No. 2:** 192.168.1.143

3.4 Descarga del software que vamos a utilizar en nuestra instalación.

Para este paso lo primero que tendremos que realizar será la descarga del software que se utilizará en la instalación y a continuación instalarlo en ambos nodos.

Accederemos a: www.mysql.org y hacemos clic en el botón: Downloads.

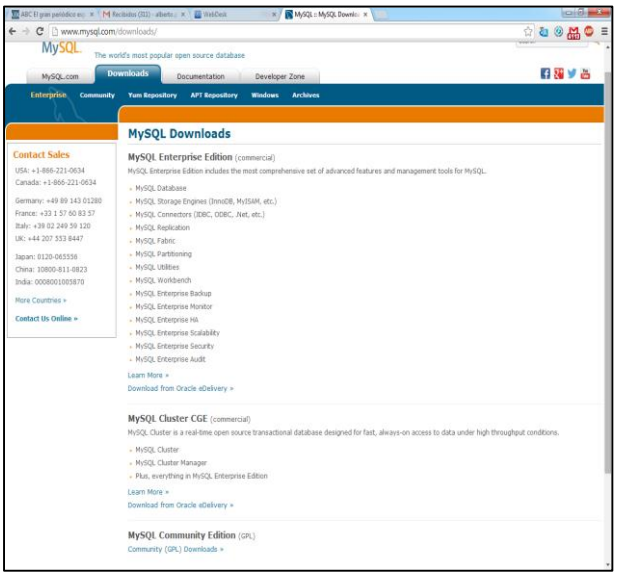


Fig.3-11[D] Pagina Principal de MySql

Accedemos a MySQLCommunityEdition:

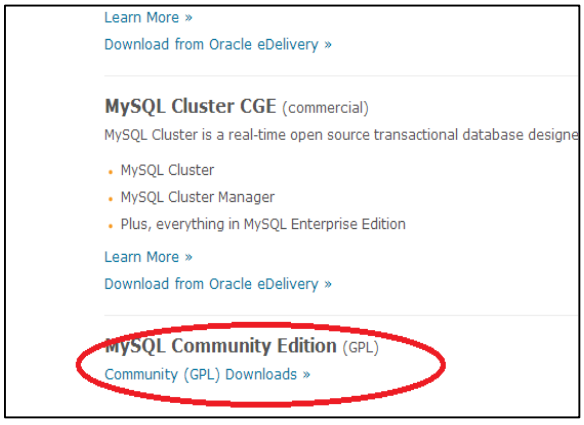


Fig.3-12[D] Descarga de Mysql Cluster

Seleccionamos la siguiente versión:



Fig.3-13[D] Selección de versión de MySql

Una vez dentro pasaremos a seleccionar la versión que corresponda a la versión de Windows que tengamos instalado (Windows xp, Windows Vista, Windows 7, Windows 8). En nuestro caso utilizaremos la versión de 64 bits ya que es la plataforma que tenemos instalada (Windows 7 versión x64 bits) por ser más fluida y presentar mejor aprovechamientos de los recursos de la PC. Será de mayor preferencia que descarguemos él ¨.zip¨ antes que él ¨.msi¨, ya que aunque para la instalación que realizaremos no habrá demasiadas diferencias, el ¨.zip¨ nos brindará una descarga más cómoda al ser un archivo comprimido y ocupará menor cantidad de espacio en nuestro disco duro.

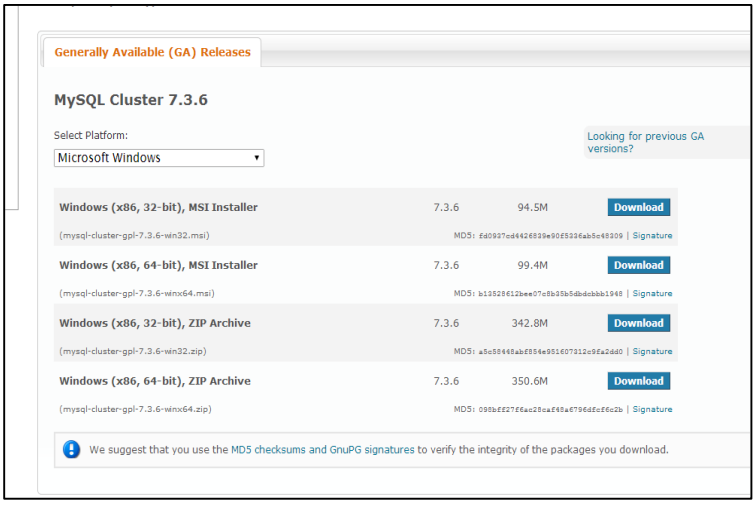


Fig.3-14[D] Descarga del Cluster

3.5 Instalación del nodo de gestión MGMT.

El motor para la Base de Datos que vamos a necesitar será el ¨NDB¨ para soporte en clúster como se ha mencionado anteriormente.

Iniciaremos la máquina virtual identificada como ¨mysql_mgmt¨ que será la que dispondrá del gestor de administración, para este caso la 192.168.1.141

1. Pasaremos a descomprimir el software, la carpeta tendrá el nombre de “mysql”.
2. Una vez descomprimida el archivo ¨.zip¨ accedemos a la carpeta mysql/bin y podremos comprobar que tenemos estos dos ejecutables que serán el servidor y el cliente para el clúster.

   . **ndb_mgmd.exe**
   . **ndb_mgm.exe**

3. Crearemos un fichero de configuración que apunte en dirección al fichero de configuración del clúster dentro del directorio “mysql/bin”.

   Lo llamaremos “my.ini” por ejemplo y añadimos la línea siguiente.

```
[mysql_cluster]
# Apuntamos al fichero de configuración del Cluster
config-file=C:/mysql/bin/config.ini
```

Fig.3-15[D] Instalacion del .Ini

NOTA: En realidad el fichero “config.ini” se puede llamar como queramos, por lo cual no afectará en nada el nombre que se le dé, pues lo llamaremos así por una cuestión de ergonómica.

4. Creamos el segundo fichero "c:\mysql\bin\config.ini" el cual contendrá los parámetros necesarios para arrancar el clúster.

```
[ndbd default]
# procesos ndbd de todos los nodos:
NoOfReplicas=2 #Número de réplicas (solo tenemos dos
nodos)
DataDir=C:/mysql/bin/cluster-data   # Directorio donde
se encuentran los datos en cada nodo

DataMemory=180M   # Memoria para almcenamiento de datos
IndexMemory=50M   # Memoria para almcenamiento de
índices

[ndb mgmd]
# Opciones del Management node, es decir del
coordinador (192.168.1.141)
HostName=192.168.1.141               # Dirección IP
DataDir=C:/mysql/bin/cluster-logs   # DDirección de los
ficheros de log

[ndbd]
```

Fig.3-16[D] Creacion de parametros

```
# Nodo1

HostName=192.168.1.142           # IP

[ndbd]
# Nodo 2
HostName=192.168.1.143           # IP

[mysqld]
# Los dos nodos también será un nodo SQL. en un entorno
de producción real no sería adecuado
HostName=192.168.0.142
HostName=192.168.0.143
```

Fig.3-17[D] Configuracion de IPS de los Nodos

5. También crearemos el directorio “C:/mysql/bin/cluster-logs “ para los logs.

3.6 Instalación de los Data Nodes.

Para cada uno de los dos nodos tendremos que realizar los siguientes pasos:

1. Extraemos el software a la siguiente dirección c:\mysql

2. Accedemos a mysql/bin
3. Comprobamos que tenemos los siguientes ejecutables:

- **ndbd.exe**
- **ndbmtd.exe**

4. Creamos el directorio C:\mysql\bin\cluster-data que es el que indicamos en el fichero de configuración del servidor de gestión.
5. Seguido creamos el directorio “mysql” y un fichero denominado “my.ini” Después añadiremos la siguiente línea apuntando al servidor de gestión.

```
[mysql_cluster]
# Apuntamos al management server:
ndb-connectstring=192.168.1.141
```

Fig.3-18[D] Ingreso de la Ip del Managment Server

3.7 Instalación de los nodos SQL.

Para este caso, los nodos de datos también serán nodos de SQL, aunque estos en realidad deberían de estar separados pero para las pruebas es suficiente. No será necesario extraer el software porque ya lo hemos realizado en pasos anteriores.

Modificamos el fichero “my.ini” para añadirle las siguientes líneas:

```
[mysqld]
ndbcluster                # Ejecutamos el motor NDB
ndb-connectstring=192.168.1.141   # management server
```

Fig.3-19[D] Configuracion final del config.ini

3.8 Arranque inicial del Clúster.

Deberemos de arrancar los nodos en el siguiente orden:

El primero será el Management Server (192.168.1.141).

- Iremos al directorio "mysql/bin" y ejecutamos el comando que indica la ruta donde se encuentra el fichero inicial de nuestra configuración.

```
ndb_mgmd -configdir=c:\mysql\bin
```

Fig.3-20[D] Arranque inicial del Mysql MGMT

- Seguido de ejecutar el comando indicado nos iremos a cada nodo y ejecutaremos el comando "ndbd".

```
C:\mysql\bin> ndbd.exe
Could neither create or open key 'SYSTEM\CurrentControlSet\Services\EventLog\App
lication\MySQL Cluster Data Node Daemon', error: 5
Failed to setup event logging
2014-08-07 14:30:25 [ndbd] INFO     -- Angel connected to '192.168.1.141:1186'
2014-08-07 14:30:25 [ndbd] INFO     -- Angel allocated nodeid: 2
```

Fig.3-21[D] Ejecucion del NDBD en los nodos

Nota: Todavía no arrancamos los nodos SQL.

- Ahora vamos al management server y tendremos que abrir una nueva ventana de MS-DOS. Seguido daremos apertura al cliente del clúster.

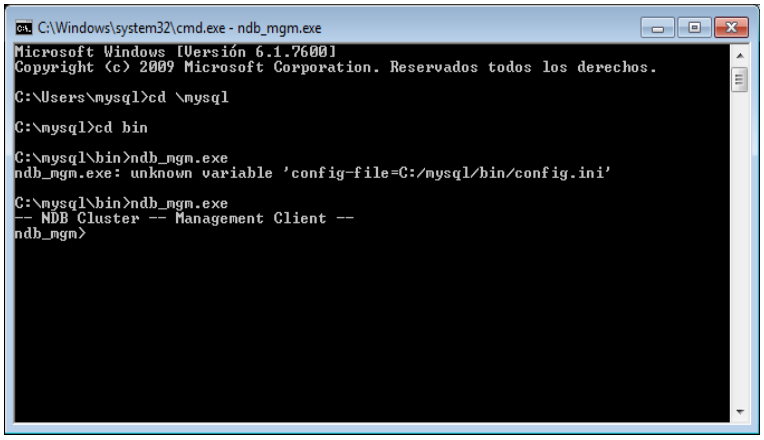


Fig.3-22[D] Ejecucion del Cliente en el cluster

- Comprobamos que los dos nodos de Datos están conectados con el comando ¨show¨.

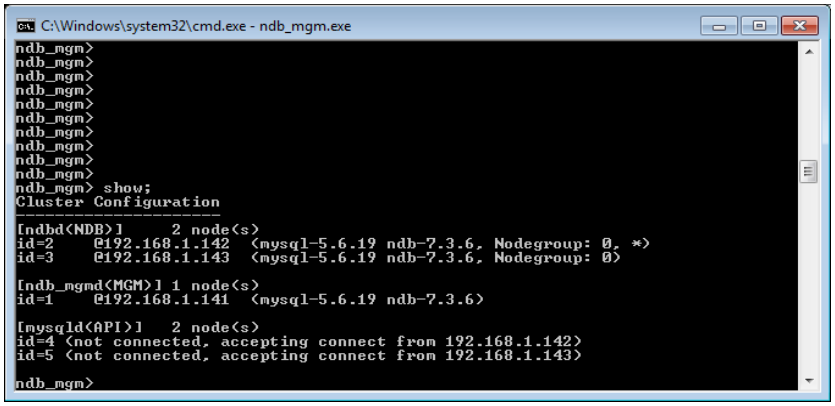


Fig.3-23[D] Ejecucion de comando show en MGT

Accedemos de nuevo a los nodos y arrancamos el mysql. Seguido ejecutamos el comando ¨mysqld –console ¨.

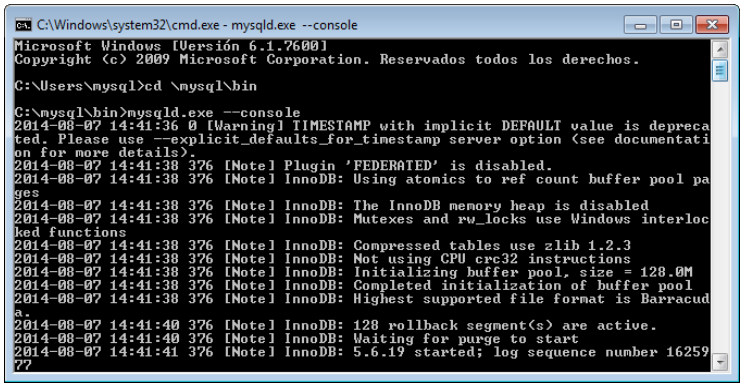


Fig.3-24[D] Ejecucion de Consola en los nodos

- Antes de continuar deberíamos comprobar que en algún momento de la salida del log, el nodo se federa dentro del clúster. En esto caso introduciríamos “nodeid 4”.

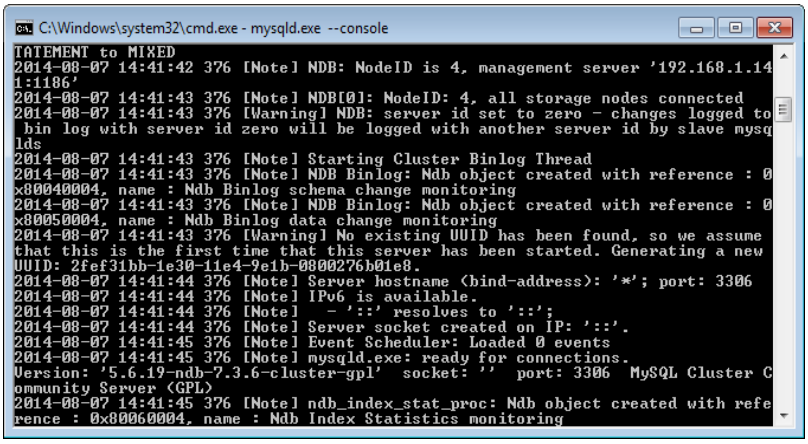


Fig.3-25[D] Carga de Consola en cada nodo

Si regresamos al management server y en cliente ponemos show, se deberán figurar también los dos nodos SQL.

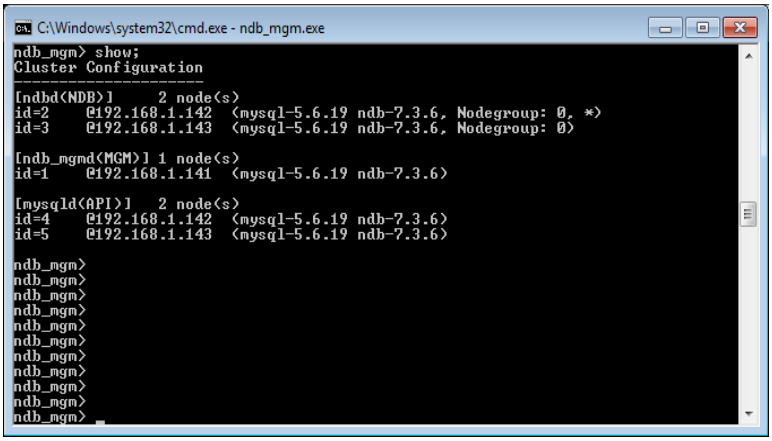


Fig.3-26[D] Ejecucion de comando Show en mgmt

3.9 Probar creando una tabla.

Primero realizaremos una prueba creando una pequeña tabla ya que tenemos el clúster debidamente configurado y arrancado.

- Accedemos desde cualquiera de los dos nodos SQL al mysql.

```
mysql -u root -p
```

Fig.3-27[D] Ingreso a una instancia como root

- Probamos que tenemos acceso a las Bases de datos.

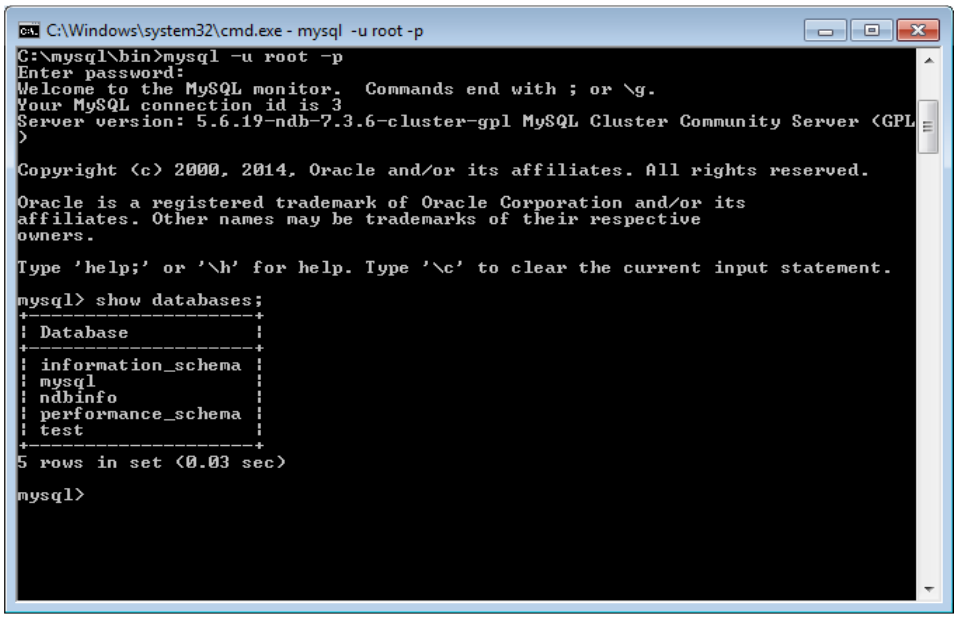


Fig.3-28[D] Test de acceso a la base de datos

- Ahora creamos una nueva Base de datos.

```
mysql> create database prueba character set utf8;
Query OK, 1 row affected (0.14 sec)

mysql>
```

Fig.3-29[D] Creacion de base de datos en un nodo

- Accedemos al otro nodo y comprobamos que también existe:

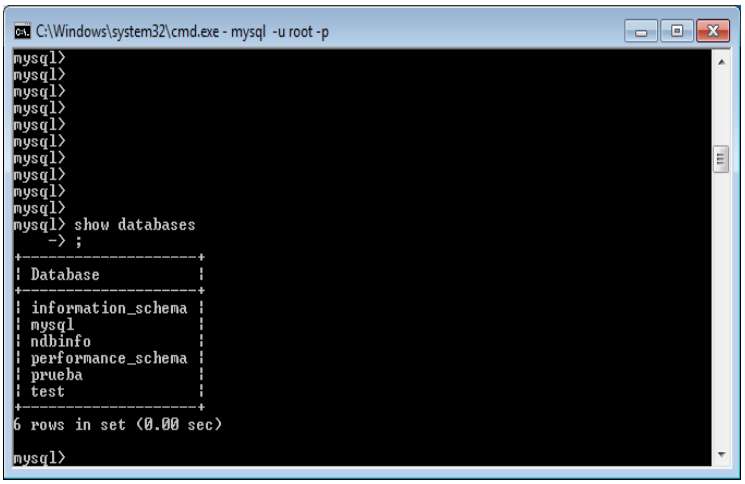


Fig.3-30[D] Test de Existencia de base de datos creada

- Creamos una tabla en la Base de datos desde cualquiera de los nodos.

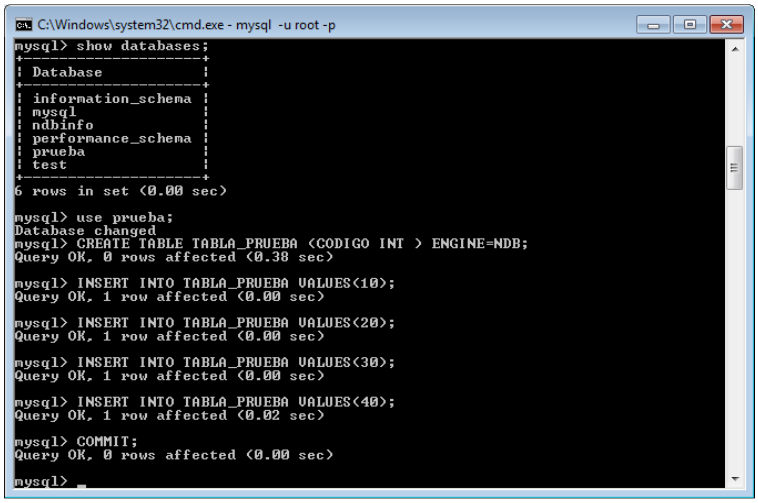


Fig.3-31[D] Creacion de una tabla en un nodo

- Probamos que la tabla existe desde cualquiera de los otros nodos.

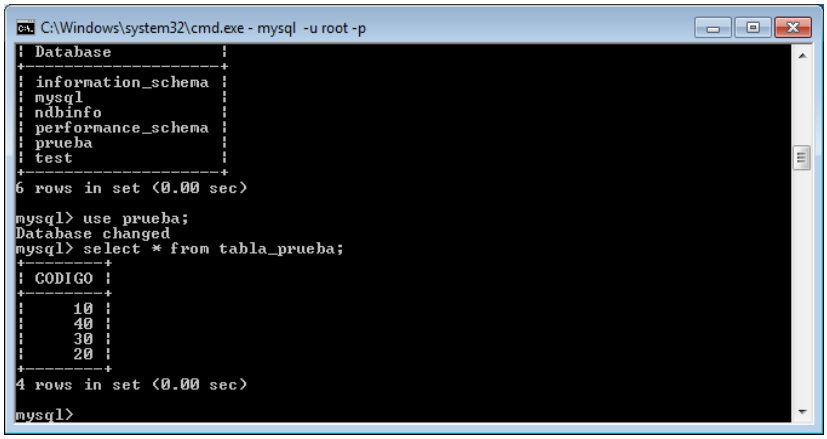


Fig.3-32[D] Test de replicacion

3.10 Carga de datos de ejemplo.

Para siempre trabajar con tablas lo haremos en modo clúster. Por lo cual configuraremos el motor por defecto de Mysql como NBDCLUSTER. Para ello accedemos al fichero de configuración "my.ini" y añadimos la línea al final. El fichero debe quedar de la siguiente manera:

```
[mysql_cluster]
# Apuntamos al management server:
ndb-connectstring=192.168.1.141


[mysqld]

ndbcluster                      # Ejecutamos el motor
NDB
ndb-connectstring=192.168.1.141  # management server
default-storage-engine=NDBCLUSTER
```

Fig.3-33[D] configuración por default en modo Cluster

Nota: Lo haremos en los dos nodos de datos.

Para cargar la Base de datos de Employees que tiene gran cantidad de registros nos iremos a cualquiera de los dos nodos de datos y copiamos el fichero. Lo descomprimimos y lo dejamos por ejemplo en una carpeta denominada “empleados”.

Vamos a ese directorio y abrimos mysql.

Ejecutamos desde mysql el comando.

```
source employees.sql
```

Fig.3- 34[D] Selección de la fuente

Esperamos a que termine el proceso que puede llegar a tardar un poco. Contendrá la siguiente información, la cual será suficiente para realizar pruebas.

- 300.000 empleados.
- 2.8 millones de salarios.

Una vez terminado comprobaremos que tenemos acceso a los datos desde el otro nodo. Debe haber creado una base de datos llamada Employees a la que podemos acceder con:

```
use employees
show tables
```

Fig.3-35[D] Test de Insercion en modo Cluster

3.11 Convertir los procesos en un servicio Windows.

De esta forma no hay que arrancar de forma manual los servicios de las máquinas, sino que lo harán automáticamente.

- Primero configuraremos el Management Server con el siguiente comando:

```
C:\> C:\mysql\bin\ndb_mgmd.exe --install
Installing service 'MySQL_Gestor'
  as '"C:\mysql\bin\ndbd.exe" "--service=ndb_mgmd"'
Service successfully installed.
```

Fig.3-36[D] Transformacion a modo Servicio de windows

- Ya podríamos arrancarlo y a su vez detenerlo con los siguientes comandos:

```
NET START ndb_mgmd

NET STOP dbm_mgmd
```

Fig.3-37[D] Inicio y detención del servicio

- Haremos de igual manera con los servidores de nodo.

```
C:\> C:\mysql\bin\ndbd.exe --install
Installing service 'MySQL Datos' as
'"C:\mysql\bin\ndbd.exe" "--service=ndbd"'
Service successfully installed.
```

Fig.3-38[D] inicio de servicio en el nodo

- Nuevamente hacemos lo mismo con los nodos de SQL.

```
C:\> "C:\mysql\bin\mysqld" --install
```

Fig.3-39[D] instalación del servicio en los nodos

Nota: En este caso serán los mismos comandos que los de datos.

3.12 Diseño para mejorar el rendimiento.

Cuando se trata de mejorar el rendimiento hay que tomar en cuenta los elementos que afectan el rendimiento de la base de datos la cual está en funcionamiento, si se toma en cuenta el rendimiento de MySQL Server se determina que este deberá ser una parte muy importante en el proceso tanto de la aplicación como en el diseño. Para lo cual se estudiara los requisitos del usuario. Además se deberá analizar las expectativas de la mayoría de usuarios para lo cual se ejemplificara con clientes que telefonean a operadoras de centrales los cuales utilizan aplicaciones que pueden ser de entrada de pedidos o que también pueden ser de ofertas de productos.

Se debe conocer también los datos, cuales son los posibles valores, la información que representan así como también como la utilizan.

Se debe diseñar una base de datos que permita la utilización apropiada de los diseños y esquemas relacionados tanto en estrella y de copo de nieve además de desarrollar y probar procedimientos así como de diseñar la estrategia de indización así como de programar un mantenimiento.

Diseñar una base de datos que utilice apropiadamente la normalización y la desnormalización, y aplicar diseños de esquema relacionales, en estrella y de copo de nieve, siempre que sea necesario además de programar un mantenimiento y una monitorización continuada.

Etapas del diseño

La etapa diferenciadora entre el diseño de una base de datos centralizada y una base de datos distribuidos es el “Diseño de la distribución” que consta de dos actividades:

Fragmentación: Decidir “como” dividimos la BD y en “que” partes.

Asignación: Decidir “donde” ubicamos cada parte, así como si tendremos replicación de datos.

Aspectos a considerar en el diseño de una BD distribuida

Fragmentación: Una relación puede ser dividida en un número de sub-relaciones, denominadas fragmentos, los cuales son entonces distribuidos.

Asignación: cada fragmento debe ser almacenado en un sitio en base a una distribución óptima.

Replicación: El SGBBB puede mantener una copia de un fragmento en diferentes sitios.

Existen algunas reglas a considerar adicionalmente para la implementación de la base de datos de manera distribuida.

El principio fundamental de las Bases de Datos Distribuidas o regla cero plantea que:

“Desde el punto de vista del usuario, un sistema distribuido deberá ser idéntico a un sistema no distribuido”

La regla cero conduce a las restantes 12 reglas.

Todas las reglas no son independientes entre sí, ni tienen igual importancia pero son útiles para entender la tecnología distribuida.

1. Autonomía Local: Los sitios de un sistema distribuido deben ser autónomos.

La autonomía Local Implica:

- ✓ Propietario Local
- ✓ Administración Local.
- ✓ Integración Local.
- ✓ Represabilidad local.
- ✓ Integración local.

2. No dependencia de un sitio central

- ✓ No debe existir en único sitio, ya que implicaría:
- ✓ Cuello de botella
- ✓ Vulnerabilidad

3. Operación continúa

- ✓ Adición de elementos
- ✓ Actualización de versiones

4. Independencia de Localización.

- ✓ El usuario desconoce dónde están físicamente los datos.

5. Independencia de fragmentación.

- ✓ Deseable porque simplifica los programas de los usuarios y sus actividades en la terminal.

6. Independencia de réplica.

- ✓ La creación y destrucción de réplicas debe hacerse transparente al usuario

La réplica proporciona:

Ventajas:

- ✓ Mayor Presentación: los datos son locales.
- ✓ Mayor disponibilidad: los datos son accesibles siempre.

Desventajas

Hay que propagar las actualizaciones.

7. Procesamiento distribuido de consultas

- ✓ Los Sistemas relacionales brindan herramientas de consulta muy eficientes.
- ✓ Varias maneras de trasladar los datos

8. Manejo distribuido de transacciones

- ✓ Transacción distribuida: varios agentes de la transacción en varios lugares.
- ✓ Control de recuperación: una transacción atómica. Todos los agentes avanzan o retroceden juntos.
- ✓ Control de recurrencia: Bloqueo mediante pasos de mensaje

9. Independencia con respecto al equipo.

- ✓ El SGBD se ejecutara igual sea cual sea el equipo

10. Independencia con respecto al Sistema Operativo

- ✓ El SGBD debe ser multioperativo sin afectar al usuario

11. Independencia con respecto a la Red.

- ✓ EL SGBD debe ser multioperativo sin afectar al usuario

3.12.1 Carga de datos masiva de datos.

Para realizar las pruebas en la base de datos primero debemos crear el modelo de entidad relación para crear los datos respectivos.

Para este caso hemos diseñado un modelo bastante genérico para una empresa comercial típica que podríamos encontrar en el país

El modelo entidad relación que hemos usado es el siguiente:

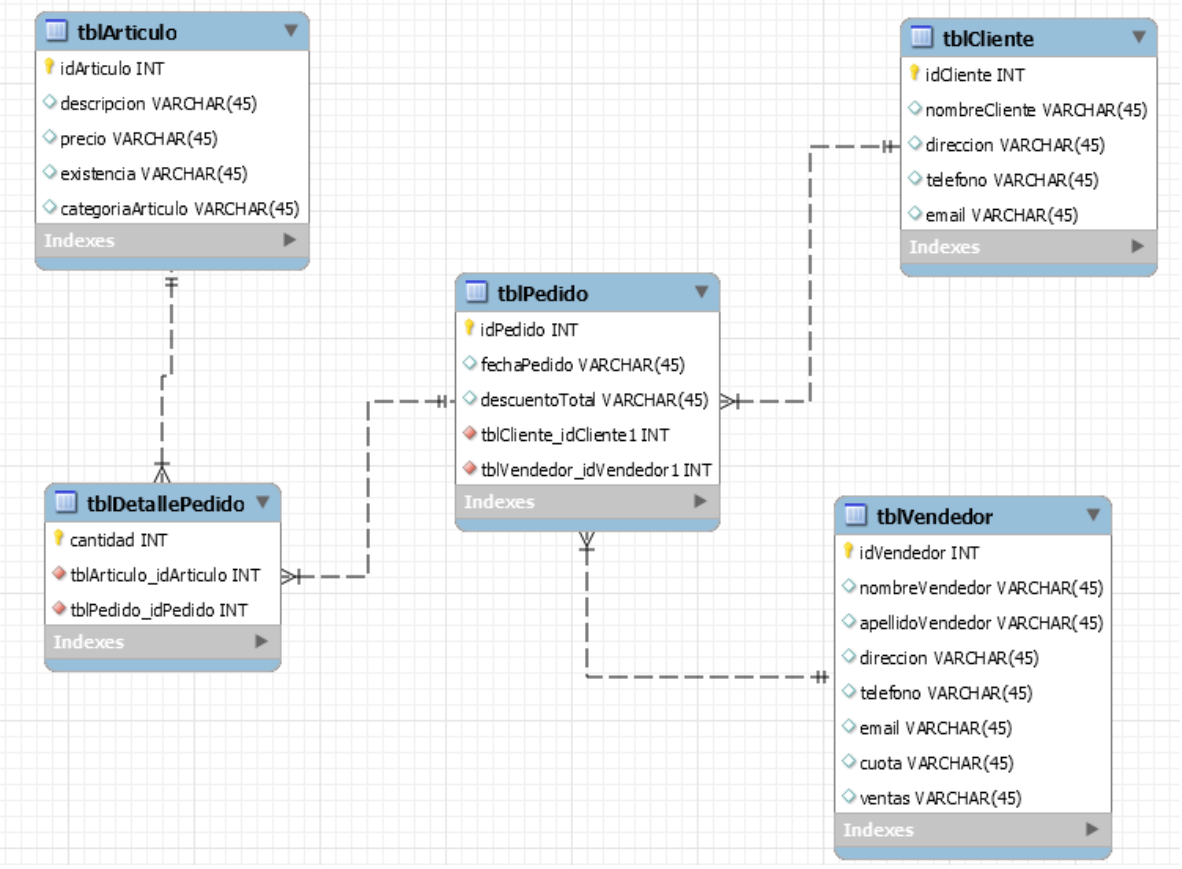


Fig.3-40[D] Diagrama entidad relación para pruebas

Lo siguiente paso será cargar información de una manera masiva e la base de datos para la generación de **dummy data** la realizaremos en Excel para posteriormente subirla al work bench de mysql y generar automáticamente un script que suba masivamente la información al gestor de dase de datos.

Finalmente ejecutamos el script para la creación de la base de datos con los 50000 registros.

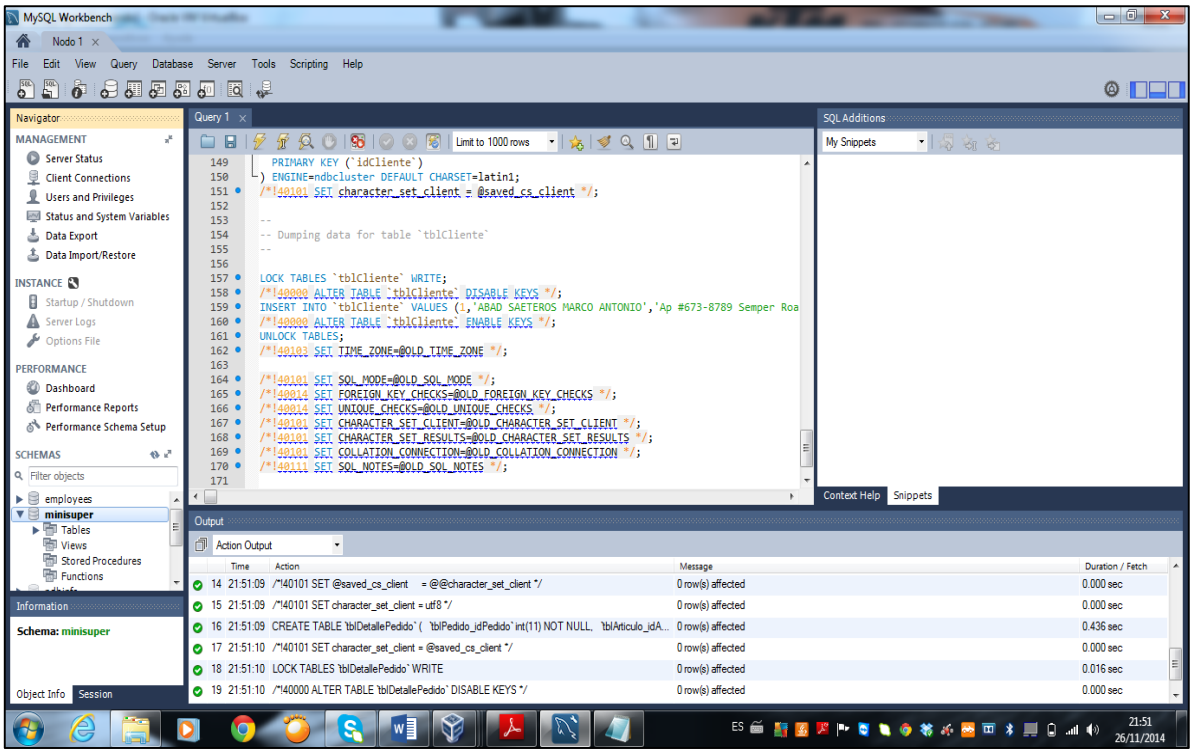


Fig.3-41[D] insercion de registros masivos

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES SOBRE EL ANÁLISIS DEL COSTO BENEFICIO

Dentro de este capitulo se analizara el costo de montar una base de datos en Cluster y de manera centralizada, en ambos análisis se tomara en cuenta los costos, el desempeño y la operatividad de ambas instancias donde se realizara las pruebas con la misma base de datos y los mismos scripts adicionalemente en las pruebas se realizara bajo las mismas condiciones con los servidores recién reiniciados, con el mismo sistema operativo y bajo un ambiente controlado.

4.1 Análisis costos de recurso humano

Dentro de este segmento consideraremos el precio de la implementación de la misma base de datos de dos maneras distintas, en un solo server y una sola instancia de MySql y en una base de datos en modo clúster, replicada en dos nodos.

Los costos serán definidos por horas hombre para implementar la misma base de datos, adicionalmente el tiempo que se debe dedicar en las tareas cotidianas de un DBA con ambos tipos de implementación.

Partiremos observando un promedio de salarios de 3 DBA junior en el mes de octubre del 2014 de 3 importantes empresas del país.

Sector Farmacéutico	Sector Alimenticio	Sector de Tecnología TI	Promedio
1150	1230	1100	1160

Tabla 4- 1 [D] Costo mensual de un experto

Partiendo de la tabla anterior procedemos a sacar el costo de una hora de trabajo de un DBA actualmente en Ecuador.

1160 dólares /160 horas mensuales = 7.25 dólares por hora.

Como referencia tomaré los tiempos que tomo instalar y configurar ambas instancias de las bases de datos

A) Para una sola instancia de MySQL

Instalación del gestor de base de datos	1 hora
Implementación del modelo ER en la instancia de MYSQL	4 horas
Migración de registros	5 horas
Pruebas	2 horas
Documentación	3 horas
Total	15 horas

Costo Total	$108.75

Tabla 4- 2 [D] Costo de la instalación de una sola instancia

B) Para modo Clúster con 2 nodos

Instalación del gestor de base de datos Managment	2 horas
Instalación Nodo 1 clúster	1 hora
Instalación Nodo 2 clúster	1 hora
Pruebas de conectividad entre nodos e instancia de managment	5 horas
Implementación del modelo ER en la instancia de MYSQL	5 horas
Migración de registros Modo replica	5 horas
Pruebas	3 horas
Documentación	5 horas
Total	27

Costo Total	$195.75

Tabla 4- 3[D] Costo de la instalación en un Cluster

´

4.2 Análisis de costo de infraestructura

La infraestructura es primordial para el desempeño de las consultas a la base de datos, Una base de datos centralizada necesitara un servidor con mejor desempeño para responder adecuadamente a las solicitudes ejecutadas desde cualquier tipo de aplicación y desde cualquier localidad donde se realizan las consultas.

Adicionalmente El Ancho de banda de datos entre localidades debe ser adecuado para que la solicitud de consultas se resuelva en tiempos prudentes de espera desde el **front end** que los usuarios finales ejecutan.

Debido a la topografía de red de la empresa es necesario hacer un análisis previo al momento de decidir en qué manera se va a implementar la gestión de la base de datos. Se debe tomar en cuenta factores clave como el presupuesto que cuenta la empresa, infraestructura existente, la calidad de conexión de datos entre los sitios geográficos de donde se accederá a la base de datos.

Haremos un análisis sobre el costo en ambos casos de instancias, en modo clúster y en una base de datos centralizada para que el desempeño de las consultas a la base no se vean afectadas por limitaciones de recursos de hardware o software.

A) Para una sola instancia de MySQL

Rubro	Precio USD
Servidor Xeon 3 procesadores 32 GB de ram	5490
Licencia Windows Server 2008 R2	750
Conexión de datos (contrato 4 años) 2MBPS Clear Channel	9600
Soporte técnico para la base de datos depreciación a 4 años	14400
Total para el funcionamiento de la BDD en un periodo de 4 años	30240

Tabla 4- 4[D] Costo Infraestructura Modo Censtralizado

Cabe recalcar que estos costos pueden variar a través del tiempo por la inflación anual a la que está sometida el país.

B) Para My SQL en modo Cluster

Rubro	Precio USD
Dos Servers Xeon un solo procesador 16 gigas de RAM.	6300
2 Licencia Windows Server 2008 R2	1500
Conexión de datos (contrato 4 años) 2MBPS Clear Channel	9600
Soporte técnico para la base de datos depreciación a 4 años	23040
Total para el funcionamiento de la BDD en un periodo de 4 años	40440

Tabla 4- 5[D] Costo infraestructura modo Cluster

4.3Análisis de los Rubros previamente mencionados

Dentro del hardware en el caso de proceder mediante un server con una instancia de mysql es necesario tener en cuenta que el servidor será sometido a mas stress porque ejecutara todas las peticiones de ambas zonas geográficas por lo cual se debe tener muy en cuenta la capacidad que este debe tener, en este caso en particular hacemos uso del caso hipotético de que esta va a ser la base de datos central de una empresa con 20 Usuarios finales ejecutando consultas simultáneamente en ambas locaciones.

En el caso de una base de datos distribuida no es necesario tener un hardware con un desempeño de alta gama dado que la carga de trasacciones se balancean en dos nodos y mejora el desempeño.

En el tema de licencias simplemente el servidor siempre va a requerir una versión de Windows server para su mejor administración adicionalmente para este caso hipotético se ha optado por el software con costo de licencia, dado que es más fácil poder encontrar gente capacitada para el soporte posterior.

La base de datos en modo Clúster se ha optado por usar igualmente wondows 2008 Server, por temas de seguridad y administración.

El ancho de bada es del mismo ya que es el que posee la empresa actualmente contaratado.

Dentro del rubro de soporte se calculó de la siguiente manera

Para la base de datos centralizada

300 dólares mensuales por consultor externo*48 meses= 14400

Para la base de datos en Clúster dada la complejidad el valor aumenta significativamente

480 dólares mensuales por consultor externo * 48 meses =23040.

En este caso vemos que el soporte al clúster demanda de más tiempo por lo mismo su valor es mayor.

El costo de la base de datos en Clúster es mayor debido a los siguientes problemas que pueden suscitarse.

Problemas fundamentales a resolver en las bases de datos distribuidas:

- Diseño de bases de datos distribuidas
- Procesamiento y optimización de consultas
- Manejo de transacciones y control de concurrencia

4.4 Pruebas de rendimiento

Se puede indicar que un Sistema Administrador de Base de Datos (SGBD), se maneja como un servicio que permite proporcionar datos a la mayoría de aplicaciones.

Cuando se habla de ajustes en el rendimiento estos tienen como objetivo optimizar el mismo para que sea utilizado para la administración.

Se determina también que la optimización deberá reducir al mínimo en lo referente al tiempo que toma la respuesta por las consultas realizadas así como también aumentar al máximo el rendimiento del servidor de la base de datos lo que se logra con la reducción del tráfico en la red tanto en la entrada como en la salida así como en el disco y tomando en cuenta el tiempo de CPU.

Al tratar de conseguir el objetivo planteado se deberá comprender primero la estructura lógica y física de los datos siendo requisitos primordiales además del equilibrio que se deben dar entre los usos conflictivos de la base de datos, tomándose como ejemplo el proceso de transacciones en línea con la toma de las decisiones.

Cuando se implementa un sistema se debe tomar muy en cuenta los problemas de rendimiento en su fase inicial o ciclo de desarrollo además de su fase final, es así que la mayoría de las soluciones que permiten optimizar el rendimiento del sistema gracias al diseño pormenorizado en la fase inicial del proyecto.

Si tomamos en cuenta tanto el software y el hardware y la repercusión que tienen al afectar el rendimiento del sistema por lo que un SABD administra en automático los recursos del hardware del que se dispone.

Al realizar el diseño de una base de datos hay que observar que todas las operaciones se realicen de forma rápida y eficaz, se determina que existen problemas de rendimiento los mismos que se solucionan cuando la base de datos esta en producción, no obstante existen otros que constituyen el resultado de haber realizado un diseño no bien estructurado lo cual se mejora cambiando la estructura y el diseño de la base de datos.

Al implementar la base de datos se deberá identificar las tablas que presentan un mayor tamaño y el rendimiento al aumento del número de usuario que tienen acceso a la base de datos así como también se dará importancia a los procesos que presentan alta complejidad.

Se estima que los cambios en el diseño mejoran el rendimiento entre los más importantes tenemos los siguientes:

- Cuando se realiza un in forme diario quiere decir que es el resumen de una tabla la cual contiene miles de filas, además en este informe se puede añadir una o varias columnas las cuales están con datos que serán utilizados luego en el informe.
- Se determina la normalización en exceso de la mayoría de bases de datos,
  Al tratar de combinar los datos relacionados la base de datos debe procesar todas las tablas lo que permite realizar más operaciones.
- Se estima que para mejorar el rendimiento de una base de datos se deberá realizar una normalización y así simplificar los procesos.

Para usaremos las realizaremos de la siguiente manera:

Modo Centralizado

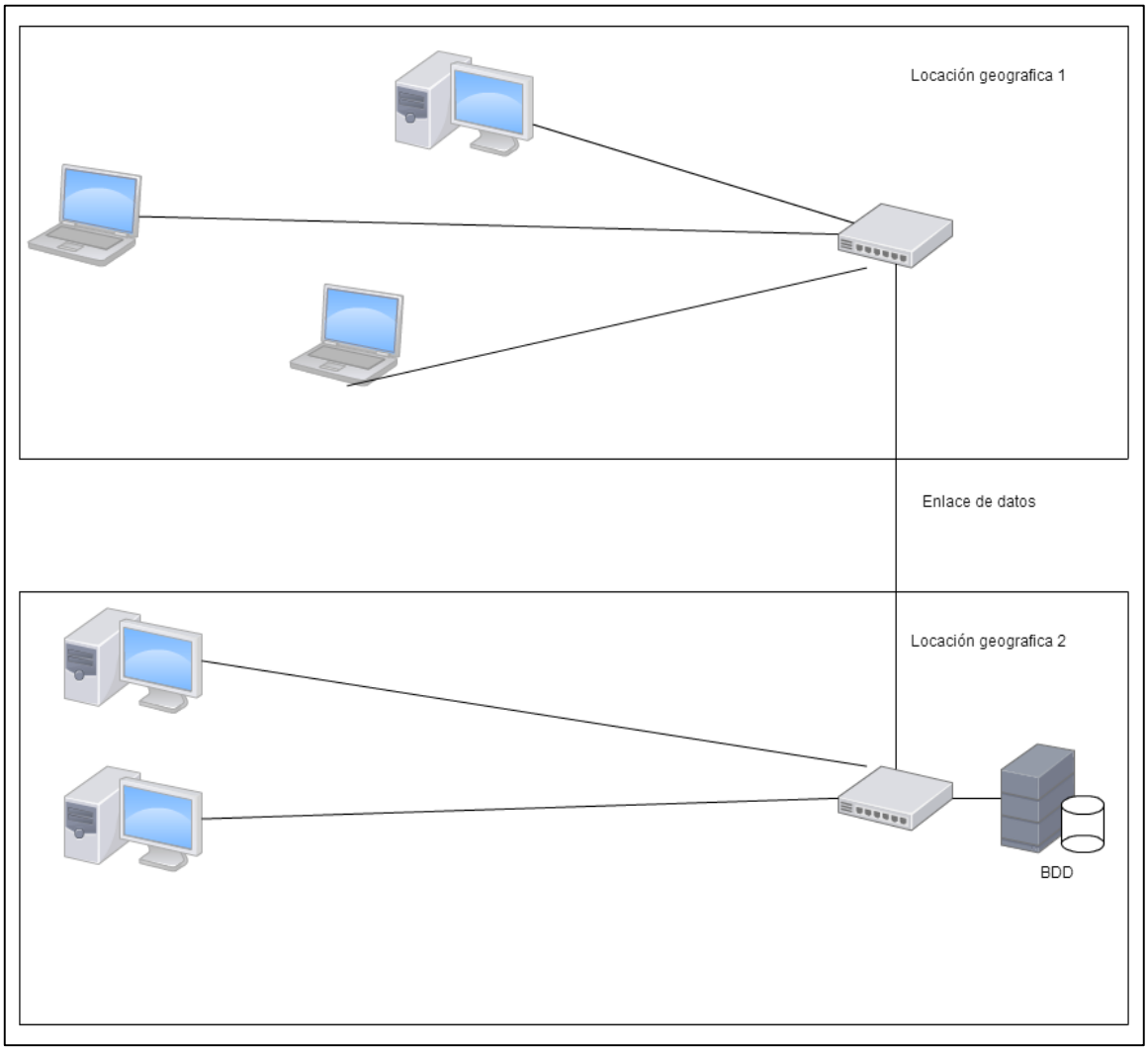


Fig.4-01[D] Diagrama de red Modo Centralizado

En esta etapa realizaremos consultas y actualizaciones dentro de la base de datos simultáneamente en varias tablas a la vez para esta prueba utilizaremos 5 máquinas dos en Ponceano Alto al norte de la ciudad de quito y mediante un enlace de datos de dos megabits en clear chanel montaremos 3 máquinas adicionales que se encontraran en la localidad de cumbaya, todas las solicitudes serán atendidas en Ponceano Alto.

Para la prueba ejecutare los scripts ubicados en Anexo 2 Anexo 3 y Anexo 4 en las 5 máquinas simultáneamente mediante teamviewer para poder realizarlas a la par en todas las instancias y tomaré los tiempos de ejecución en cada máquina.

Modo Clúster

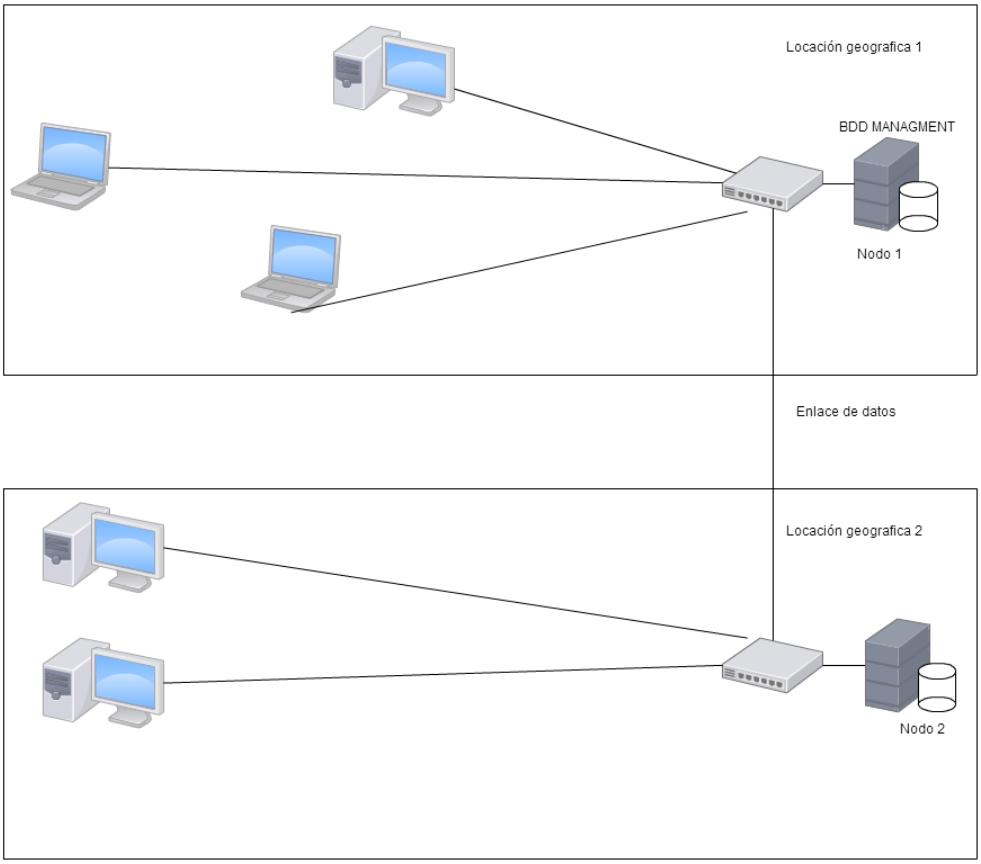


Fig.4-02[D] Diagrama de red modo Cluster

En esta etapa realizaremos consultas y actualizaciones dentro del nodo más cercano simultáneamente en varias tablas a la vez para esta prueba utilizaremos 5 máquinas dos en Ponceano Alto al norte de la ciudad de quito y mediante un enlace de datos de dos megabits en clear chanel montaremos 3 máquinas adicionales que se encontraran en la localidad de cumbaya adicionalmente una máquina que poseerá el nodo 2.

Para la prueba ejecutare los scripts ubicados en Anexo 2 Anexo 3 y Anexo 4 en las 5 máquinas simultáneamente mediante teamviewer para poder realizarlas a la par en todas las instancias y tomaré los tiempos de ejecución en cada máquina.

4.2.3 Resultados de pruebas

Después de realizar 3 corridas en ambos tipos de instancias, hemos obtenido los siguientes resultados.

Corrida 1

Resultados para el modo centralizado

Script	Localización en la ejecución	Tiempo de ejecución Segundos
Script 1	Locación 1	4.3
Script 2	Locación 1	5.2
Script 3	Locación 1	3.8
Script 1	Locación 2	17.4
Script 3	Locación 2	15.66

Resultados para modo Clúster

Script	Localización en la ejecución	Tiempo de ejecución Segundos
Script 1	Locación 1	4.3
Script 2	Locación 1	5.2
Script 3	Locación 1	3.8
Script 1	Locación 2	3.5
Script 3	Locación 2	4.6

Tabla 4- 6[D] Resultados test 1

Corrida 2

Resultados para el modo centralizado

Script	Localización en la ejecución	Tiempo de ejecución Segundos
Script 1	Locación 1	4.2
Script 2	Locación 1	3.7
Script 3	Locación 1	4.5
Script 1	Locación 2	15.3
Script 3	Locación 2	18.2

Resultados para modo Clúster

Script	Localización en la ejecución	Tiempo de ejecución Segundos
Script 1	Locación 1	4.2
Script 2	Locación 1	4.8
Script 3	Locación 1	3.2
Script 1	Locación 2	2.1
Script 3	Locación 2	5.2

Tabla 4- 7[D] Resultados Test 2

Corrida 3

Resultados para el modo centralizado

Script	Localización en la ejecución	Tiempo de ejecución Segundos
Script 1	Locación 1	3.4
Script 2	Locación 1	5.6
Script 3	Locación 1	4.5
Script 1	Locación 2	18.2
Script 3	Locación 2	20.5

Resultados para modo Clúster

Script	Localización en la ejecución	Tiempo de ejecución Segundos
Script 1	Locación 1	4.5
Script 2	Locación 1	4.4
Script 3	Locación 1	4.5
Script 1	Locación 2	4.8
Script 3	Locación 2	3.8

Tabla 4- 8[D] Resultados Test 3

Como podemos apreciar en las 3 corridas el patrón es muy parecido en todos los casos cuando no existe un nodo local los tiempos de ejecución son más largos y es menos confiable la disponibilidad de los datos, mientas que con una instancia en clúster es posible tener una mayor disponibilidad de los datos y es más fácil y rápido realizar consultas o transacciones en el nodo más cercano.

4.3 Conclusiones

Después de realizar las pruebas correspondientes en ambos ambientes lo que se supuso desde un inicio en la práctica fue demostrado, pudimos observar que los beneficios de montar una base de datos en Clúster son tangibles, si bien los costos en el recurso humano es decir el especialista que se encargara del mantenimiento de la base de datos, la fiabilidad, la alta disponibilidad y sobre todo el desempeño en cualquier locación geográfica de donde se realicen las consultas y transacciones.

La disponibilidad es un factor clave, las bases de datos distribuidas en dos o más nodos poseen altas ventajas en cuanto a la disponibilidad se refiere la probabilidad que dos lugares geográficamente apartados sufran problemas técnicos es baja por lo cual para el usuario es transparente el momento que un servidor no es accesible.

Más transacciones pueden ser ejecutadas en menor tiempo, las bases de datos distribuidas poseen un desempeño superior en comparación de una base de datos centralizada, pudiendo ejecutar más trasacciones y reducir considerablemente los tiempos de espera por parte de los usuarios.

El poseer una base de datos distribuida es mas costoso desde el punto de vista económico pero es importante recalcar que al poseer un mejor performace los usuarios pueden ejecutar mas transacciones en menor tiempo lo cual es altamente medible en el desempeño de de el numero de trasacciones ejecutados en un dia lo cual significa un ahorro en horas de recurso humano.

Montar una Base de datos en modo cluster no siempre es la mejor opción, La necesidad del negocio y los recursos del mismo influye en su totalidad en la decisión de la implementación o no de un cluster, aparte del presupuesto es importante también en tomar en cuenta la seguridad y disponibilidad de los datos.

Un Cluster es altamente Aplicable a nuevas maneras de programación como Arquitectura orientada a servicios donde en el momento que exista una caída de un servidor solo se puede reconfigurar el servicio web y el **front end** del usuario seguirá funcionando, mateniendo asi la continuidad del negocio.

4.4 Recomendaciones

Para el montaje de una base de datos en Cluster existen algunas recomendaciones para tomar en cuenta:

- Se debe analizar detenidamente el presupuesto y comparar con los beneficios tangibles para determinar si es necesario utilizar un Cluster o un modelo centralizado.
- Es importante considerar el tema de la disponibilidad, si bien es cierto que el gasto será mucho mayor en un cluster los clientes deben estar concientes de que un cluster va a tener una alta disponibilidad y se puede asegurar una mayor continuidad del negocio.
- Se recomienda siempre trabajar de la mano con las personas que estén encargadas del desarrollo de la aplicación, y es muy importante que se desarrolle en arquitectura SOA ya que en el momento de un incidente el servicio es modificado y los usuarios no tendrán un alto tiempo de caída.

Como estudiante de la Facultad de ingeniería poseo algunas recomendaciones adicionales

- Recomiendo que la carrera sea mas orientada a situaciones del mundo laboral real, si bien es cierto existen materias que deben ser atravesadas como estudiante, considero que seria mucho mejor si materias practicas como evaluación de sistemas, base de datos entre otras sean dadas a partir de tercer nivel y asi formar profesionales con un enfoque diferente al momento de salir al mundo laboral.
- Es altamente recomendable desde mi punto de vista realizar prácticas profesionales dentro de la universidad ya que mucho de lo que es el ámbito laboral lamentablemente no se lo puede ver en horas de clase.
- A la Pontificia Universidad Catolica del Ecuador recomiendo una mayor motivación a los estudiantes de Ingenieria para la investigación, actualmente el país requiere agregar valor a cada producto y la investigación es el camino para hacerlo.

GLOSARIO

ANSI	:	El Instituto Nacional Estadounidense de Estándares (ANSI, por sus siglas en inglés:AmericanNationalStandardsInstitute)
BDDes	:	una colección de múltiples BD, lógicamente interrelacionadas, diseminadas sobre una red de computadoras
CPUs	:	Unidades de Procesos Computarizados
DB2	:	es una marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base de datos
DBMS	:	Capa de software entre la base de datos física y el usuario. El sistema DBMS administra todo el acceso a la base de datos.
DBMS.	:	(Data Base Management System). Son las siglas en inglés para los Sistemas de Gestión de Bases de Datos (SGBD)
DDBMS	:	El sistema de administración de base de datos distribuida (DDBMS), está formado por las transacciones y los administradores de base de datos distribuidos
DML	:	Subconjunto de instrucciones SQL usado para recuperar y manipular datos. Las instrucciones DML empiezan normalmente con SELECT, INSERT, UPDATE o DELETE.
DMV	:	Conjunto de vistas integradas que devuelven información de estado de servidor sobre valores, objetos y configuraciones de SQL Server.
DNS	:	Domain Name Server
DPF	:	Función de partición de Base de datos
DSN	:	Recopilación de información usada para conectar una aplicación con una base de datos ODBC determinada.
EC2	:	Elastic Compute Cloud (EC2) proporciona capacidad informática escalable en la nube de AWS con tarifas de pago por uso.
ESQL	:	es una base de datos SQL relacional integrada
Foreing Key	:	Clave en una tabla de base de datos que procede de otra tabla (también conocida como "tabla de referencia") y cuyos valores coinciden con la clave principal (PK) o clave única en la tabla a que se hace referencia.
IBM	:	International Business Machines Empresa Multinacional Estadounidense de tecnología y consultoría
MySql	:	Gestor de Base de datos actualmente propiedade de Oracle Corp.

nodo de clúster	:	Equipo individual en un clúster de servidores.
ODBS	:	Es una interfaz de nivel de llamado estándar, desarrollado por Microsoft.
SGBD	:	Un sistema de gestión de bases de datos constituye un conjunto de programas que establecen el almacenamiento, modificación datos
SOA	:	Del Ingles (Service-oriented architecture) Arquitectura orientada a Servicios
SQL	:	Lenguaje de consulta estructurado
SQL.	:	(por sus siglas en inglés QueryStructuredLanguage) es un lenguaje declarativo de acceso a bases de datos
SQL/DS	:	(Structured Query Language/Data System), released in 1981, was IBM's first commercial relational database management system.
SYS.SYSFILE	:	Tabla del sistema que permite guardar un registro de los archivos dbspace
XML	:	siglas en inglés de extensibles MarkupLanguage ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web

Tablas e Ilustraciones.

Ilustraciones

Fig. 1-01[A] 2 Evolución de los Sistemas de gestión de bases de datos 7
Fig. 1-02[8] Representacion de un servidor stand alone 19
Fig. 1-03[8] Conexión con el servidor 20
Fig. 1-04[8] Representacion de replicación remota 22
Fig. 1-05[8] Jerarquía de la Replicación. 23
Fig. 1-06[8] Ejemplo en Sybase Central 24
Fig. 1-07[8]Ejemplo de un servidor Adaptive Anywhere en Sybase Central 24
Fig. 1-08[8] Servidor stand alone en red 27
Fig. 1-09[8] Sistema de base de datos en una red 28
Fig. 1- 10[8] Ejemplo de Replicacion en Red 29
Fig. 1- 11[8] Archivo central de base de datos 31
Fig. 1-12[8] Diagrama de archivos adicionales en DBspaces 31
Fig. 1-13[B] ejecución en red de la replicacion 32
Fig. 1-14[B] Ejemplo de nodos en red 33

Fig.2-01[6] Infraestructura detallada de la base de datos 36
Fig.2-02[5] SGBD qye soportan distribución 44
Fig.2-03[5] Comparacion de fortalezas y debilidades de los SGBD 44

Fig.3-01[D] Representacion de Un Cluster 45
Fig.3-02[D] Representacion de nodos 46
Fig.3-03[D] Instalacion de la Maquina Virtual 47
Fig.3-04 [D] Elijiendo el disco duro virtual 48
Fig.3-05[D] Tipo de disco duro virtual 48
Fig.3-06[D] Eleccion del formato de disco duro virtual 49
Fig.3-07[D] Eleccion dinámica de disco duro 49
Fig.3-08[D] Selección del tamaño del disco virtual 50
Fig.3-09[D] Clonacion de Maquina Virtual 51
Fig.3-10[D] Clonacion independiente de maquina virtual 52
Fig.3-11[D] Pagina Principal de MySql 53
Fig.3-12[D] Descarga de Mysql Cluster 53
Fig.3-13[D] Selección de versión de MySql 54
Fig.3-14[D] Descarga del Cluster 54
Fig.3-15[D] Instalacion del .Ini 55
Fig.3-16[D] Creacion de parametros 56
Fig.3-17[D] Configuracion de IPS de los Nodos 56
Fig.3-18[D] Ingreso de la Ip del Managment Server 57

Fig.3-19[D] Configuracion final del config.ini 58
Fig.3-20[D] Arranque inicial del Mysql MGMT 58
Fig.3-21[D] Ejecucion del NDBD en los nodos 58
Fig.3-22[D] Ejecucion del Cliente en el cluster 59
Fig.3-23[D] Ejecucion de comando show en MGT 59
Fig.3-24[D] Ejecucion de Consola en los nodos 60
Fig.3-25[D] Carga de Consola en cada nodo 60
Fig.3-26[D] Ejecucion de comando Show en mgmt 61
Fig.3-27[D] Ingreso a una instancia como root 61
Fig.3-28[D] Test de acceso a la base de datos 62
Fig.3-29[D] Creacion de base de datos en un nodo 62
Fig.3-30[D] Test de Existencia de base de datos creada 63
Fig.3-31[D] Creacion de una tabla en un nodo 63
Fig.3-32[D] Test de replicacion 64
Fig.3-33[D] configuración por default en modo Cluster 64
Fig.3- 34[D] Selección de la fuente 65
Fig.3-35[D] Test de Insercion en modo Cluster 65
Fig.3-36[D] Transformacion a modo Servicio de windows 65
Fig.3-37[D] Inicio y detención del servicio 66
Fig.3-38[D] inicio de servicio en el nodo 66
Fig.3-39[D] instalación del servicio en los nodos 66
Fig.3-40[D] Diagrama entidad relación para pruebas 70
Fig.3-41[D] insercion de registros masivos 71

Fig.4-01[D] Diagrama de red Modo Centralizado 78
Fig.4-02[D] Diagrama de red modo Cluster 79

Tablas

Tabla 4- 2 [D] Costo mensual de un experto 72
Tabla 4- 3 [D] Costo de la instalación de una sola instancia 73
Tabla 4- 4[D] Costo de la instalación en un Cluster 73
Tabla 4- 5[D] Costo Infraestructura Modo Censtralizado 74
Tabla 4- 6[D] Costo infraestructura modo Cluster 75
Tabla 4- 7[D] Resultados test 1 80
Tabla 4- 8[D] Resultados Test 2 81
Tabla 4- 9[D] Resultados Test 3 82

Bibliografía

1. Bachman, C. (1973). *The programmer as navigator*. Recuperado el 10 de 04 de 2014, de http://www.cs.uiuc.edu/class/fa05/cs511/Spring05/other_papers/p653-bachman.pdf
2. Monografias.com. (2013). *Bases de Datos DBMS.* Recuperado el 22 de 05 de 2014, de http://www.monografias.com/trabajos82/fundamento-y-diseno-base-datos/fundamento-y-diseno-base-datos3.shtml
3. MONOGRAFIAS.COM. (10 de 12 de 2014). *www.monografias.com*. Recuperado el 10 de 12 de 2014, de http://www.monografias.com/trabajos82/base-datos-distribuidas/base-datos-distribuidas2.shtml
4. Paulsen, K. (2013). *Sybase SQL Server Desempeño and Tuning Guide*. Recuperado el 10 de 04 de 2014, de http://enos.itcollege.ee/~priit/(I216)%20Ab-de%20administreerimine/J02.%20SyBase%20admin%20juhend/perf.pdf
5. Wiesse, F. (13 de 02 de 2012). *La evolución de sistemas de administración de base de datos.* Recuperado el 24 de 05 de 2014, de http://beta.udep.edu.pe/hoy/2012/la-evolucion-de-sistemas-de-administracion-de-base-de-datos/
6. Wikipedia. (2013). *Base de Datos.* Recuperado el 22 de 05 de 2014, de http://en.wikipedia.org/wiki/Database
7. Wikipedia. (2013). *Sistema de Gestión de Bases de Datos*. Recuperado el 28 de 03 de 2014, de http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos
8. William, K. (1983). *A Simple Guide to five Normal in Relational Database Theory*. Recuperado el 10 de 04 de 2014, de http://www.iai.uni-bonn.de/III//lehre/vorlesungen/TDWA/WS07/1.pdf

Bibliografía Libros

A. [A] [La evolución de sistemas de administración de base de datos Wiesse López Federico, el 13 de febrero 2012]

B. [B][Sistemascomputacionales.wordpress.com.bases-de-datos distribuidas] Wiesse, Federico (2012)

C. [Estructura de un DBMS] Sanchez Jorge (2004)

D. [ANALISIS COSTO BENEFICIO DE UNA BASE DE DATOS DISTRIBUIDA] Benavides Luis, Enero 2015.