UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS

SECCION DE POSGRADO

APLICACIONES DE AFINAMIENTO DE RENDIMIENTO (PERFORMANCE TUNING) Y AUMENTO DE ESCALABILIDAD EN SISTEMAS DE INFORMACION EMPRESARIALES SOPORTADOS POR BASES DE DATOS RELACIONALES

TESIS

PARA OPTAR EL GRADO ACADEMICO DE:
MAESTRO EN CIENCIAS CON
MENCION EN
INGENIERIA DE SISTEMAS

ING. JOSÉ LUIS PONCE VERGARA

LIMA – PERU

2010

Dedico esta humilde obra a Dios y a mi familia.

AGRADECIMIENTO

Agradezco en primer lugar a mi asesor y jurados especialistas designados para esta tesis. Sus valiosos consejos y orientaciones se encuentran plasmados en este trabajo. Asimismo, a todos los profesores que tuve en la Maestría de Ingeniería de Sistemas por haber hecho de mí un mejor profesional y persona. Finalmente, al Jefe del Posgrado Dr. Peñafort Luis Huamán Ureta, que sin su apoyo no podría haber alcanzado este logro personal.

INDICE

CAPITULO IV

APLICACIÓN DE AFINAMIENTO DE RENDIMIENTO EN GRUPO ENDESA 120

DESCRIPTORES TEMÁTICOS

Database Servers / Servidores de Base de Datos

Performance Tuning / Afinamiento de Rendimiento

Response Time / Tiempo de Respuesta

Scalability / Escalabilidad

SQL Tuning / Afinamiento de sentencias SQL

Throughput / Transacciones por unidad de tiempo

RESUMEN

El rendimiento de un sistema información es un tópico que va cobrando importancia radical en la medida que los sistemas de computación se vuelven más complejos y extensos. Dentro de este gran ámbito, la tesis aborda el tema del rendimiento de servidores de Base de Datos, entendiéndose éste como eje central en la arquitectura de los sistemas de información empresariales.

La investigación expone la metodología con la que se planifica y maneja el rendimiento de los servidores de Base de Datos, y se revisarán casos de éxito en empresas de la escena local para validar las hipótesis formuladas.

La presente tesis tiene como objetivo mostrar los beneficios tangibles alcanzados al aplicar la metodología de afinamiento de rendimiento de sistemas (performance tuning) en las plataformas tecnológicas de empresas de primer nivel que operan en el mercado peruano.

El afinamiento de rendimiento de sistemas persigue como objetivo brindar rendimiento y escalabilidad a las soluciones informáticas aprovechando al máximo los recursos disponibles de la organización. El énfasis es brindar buen nivel de servicio al máximo número de usuarios minimizando consumos y costos.

Asimismo, tiene fuertes connotaciones económicas porque ahorra principalmente horas-hombre disminuyendo los tiempos de respuesta de los sistemas, ahorra dinero evitando la adquisición de nuevo hardware optimizando el uso de lo ya existente, y está en consonancia con las medidas de protección del medio ambiente, dado que un sistema optimizado consume menos recursos y energía.

Los negocios en los cuales aplicamos la metodología de afinamiento persiguen la excelencia operativa y la calidad total, y para ello los sistemas que soportan sus procesos deben estar acordes a estas premisas. El análisis de infraestructura y el afinamiento de la misma permiten alinear los objetivos de los negocios con los recursos de IT (Information Technology) actuales.

La metodología teórica tradicional enfoca al análisis de rendimiento bajo un enfoque holista que abarca todas fases del ciclo de vida de sistemas.

En la práctica, los esfuerzos de afinamiento se dan principalmente cuando los sistemas se encuentran en producción, con el objetivo de aliviar sobrecargas y mejorar el rendimiento de los sistemas en ejecución. [GA01]

La tesis intenta principalmente responder la pregunta si es posible optimizar los recursos empresariales, brindando a su vez una adecuada ejecución y escalabilidad utilizando técnicas de afinamiento de rendimiento de aplicaciones orientadas a Base de Datos.

El afinamiento de rendimiento y la consecuente mejora de la escalabilidad de las plataformas tecnológicas de información tienen una fuerte motivación económica debido a los ahorros y potenciales

ganancias que genera, y su adecuada aplicación redunda directamente en los resultados económico-financieros de un negocio.

La tesis empieza proponiendo una hipótesis general, y para determinar su validez utiliza la inducción, identificando casos empresariales donde metas de afinamiento deben ser cumplidas, y para cada una de ellas se aplica la Metodología de Mejora de Rendimiento de Base de Datos, que entre otras cosas observa el código que se ejecuta en el motor de Base de Datos y continúa con aspectos estructurales y organizativos de la Base de Datos y de la plataforma subyacente de los sistemas principales.

El análisis de rendimiento se centra en el servidor principal por ser éste el componente central de la arquitectura de sistemas. Asimismo, el componente central del servidor principal es el motor de Base de Datos Relacional Oracle, propiedad de Oracle Corporation (http://www.oracle.com). Nuestros esfuerzos de investigación del rendimiento revisan al detalle este componente de software altamente optimizable, que a su vez es el líder mundial en el segmento de Base de Datos Relacionales.

Para validar lo planteado en la tesis, presentamos casos de éxito obtenidos en empresas de primer nivel peruanas, que operan en los rubros de comercialización de alimentos, energía y salud pública.

INTRODUCCION

Todo producto intangible, y en esto el software se ajusta perfectamente a esta categoría, asegura un rendimiento óptimo cuando se incorpora técnicas de aseguramiento de rendimiento desde la fase de incepción del mismo. Conforme los tópicos de rendimiento se pospongan a las otras fases del ciclo de vida del software, los cambios para asegurar un rendimiento aceptable tendrán un costo ostensiblemente mayor.

El afinamiento del rendimiento típicamente se aplica a los sistemas de computación, pero los mismos métodos pueden ser aplicados a los mercados económicos, burocracias u otros sistemas complejos. La motivación del afinamiento es llamada un problema de rendimiento, y la cual puede ser real o anticipada. Muchos sistemas responderán a un incremento de la carga transaccional con algún grado de degradación del rendimiento. La habilidad de un sistema para aceptar mayores cargas es llamada escalabilidad, y modificar un sistema para manejar una carga mayor es sinónimo de afinamiento de rendimiento.

CAPITULO I

PLANTEAMIENTO DE LA INVESTIGACION

1.1. DIAGNOSTICO Y EVALUACION DEL PROBLEMA

Las empresas que apoyan sus procesos empresariales en sistemas de información computarizados, necesitan que tanto sus aplicaciones como la infraestructura que las soporta les brinden principalmente un rendimiento planificado, altos niveles de disponibilidad, y que la plataforma donde reside su solución informática sea escalable.

Como componente central de las arquitecturas de los sistemas de información se encuentra el Sistema Relacional de Gestión de Base de Datos (RDBMS por sus siglas en inglés). Un RDBMS es una aplicación que permite a quien la usa crear, acceder y mantener una colección de datos relacionados. Un RDBMS es un sistema complejo que está compuesto de una colección de subsistemas, cada uno de ellos con una tarea específica. Es el trabajo del RDBMS controlar cada uno de estos subsistemas durante su operación. Debido a la competencia inherente por los recursos de un sistema informático, es comprensible que lograr un alto nivel de rendimiento en un RDBMS es una tarea difícil. Los recursos del sistema informático, i.e. computador servidor, son asignados para su uso por el RDBMS a través de parámetros y configuraciones. La dificultad inicial que se confronta al

afinar un RDBMS es determinar cuáles de esos recursos necesitan ser ajustados a fin de resolver el problema de rendimiento (performance problem). [FU01]

¿Por qué es importante el afinamiento de rendimiento en sistemas de gestión de Base de Datos?

Porque ofrece:

Beneficios financieros

- Al aprovechar mejor el hardware se evita incurrir en nuevas compras de hardware.

- Fomenta el downsising, para generar ahorros al contar con una infraestructura más barata.

- El ahorro generado por el performance tuning puede ser redirigido en la adquisición de otros recursos o bajar los costos de producción; lo que a su vez otorga a la organización ventajas competitivas en el mercado.

- Al mejorar los tiempos de respuesta de las aplicaciones, los usuarios son más productivos. La misma carga transaccional puede ser absorbida por un menor número de operadores.

Beneficios humanos

- Fomenta la productividad al hacer la operación de los sistemas más ágil.

- Mejora la calidad de atención a los clientes, dado que los tiempos de respuesta por transacción son menores.

- Mejora la relación entre áreas de la organización. Si existen SLAs (Service Level Agreements por sus siglas en inglés) internos, estos pueden ser cubiertos, y se colabora a que los objetivos tanto a nivel de áreas como de la organización se cumplan. [SC01]

1.2. DEFINICION DEL PROBLEMA DE INVESTIGACION

¿Es posible brindar rendimiento y escalabilidad a las soluciones informáticas aprovechando al máximo los recursos disponibles de la organización?

El afinamiento del rendimiento (performance tuning) de los componentes de los sistemas de información responde a esta interrogante; siendo el componente central el motor de Base de Datos. El afinamiento de rendimiento es crucial en cualquier RDBMS. Muchas organizaciones responden a los problemas de rendimiento desperdiciando dinero, al comprar hardware más potente y caro, o contratando consultoría muchas veces onerosa. [GU01]

El afinamiento del rendimiento de Bases de Datos relacionales toma ventaja de las técnicas de optimización usando el hardware y software ya existente. Explota las características sobre disk stripping, mirroring, RAID, cliente/servidor, bases de datos distribuidas, MPPs, SMPs, etc. con las que pueda contar el hardware ya existente en una organización.

El ajuste del rendimiento de un sistema de gestión de Base de Datos relacional implica ajustar varios parámetros y opciones de diseño para mejorar su rendimiento para una aplicación concreta. Diferentes aspectos del diseño de los sistemas de bases de datos –que van desde aspectos de alto nivel, como el diseño de los esquemas y transacciones, a parámetros de la base de datos, como el tamaño de las memorias intermedias (buffers), hasta aspectos de hardware, como el número de discos- afectan al rendimiento de las aplicaciones. Puede ajustarse cada uno de estos aspectos de modo que mejore el rendimiento.

Muchas personas piensan del afinamiento de rendimiento como un tema complementario, o que se revisa cuando el rendimiento de los sistemas empieza a decaer y los usuarios empiezan a quejarse. Esta es una visión equivocada. El rendimiento es una meta de diseño, algo que se construye en el sistema desde el principio y no cuando las cosas empiezan a ponerse mal y los plazos son muy cortos. También el rendimiento es algo que nunca finaliza. Está presente en todo el ciclo de vida del software.

Existen directivas de rendimiento en cada fase del ciclo de vida del software que están relacionadas a los sistemas de gestión de bases de datos relacionales:

- Plan
- Análisis
- Diseño
- Desarrollo
- Aceptación

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

El objetivo general es brindar rendimiento y escalabilidad a las soluciones informáticas aprovechando al máximo los recursos disponibles de la organización.

1.3.2. OBJETIVO ESPECIFICO

Como objetivo específico es mostrar que sobre la base de analizar los principales cuellos de botella (eventos) que representan el mayor porcentaje de tiempo de Base de Datos (DB Time), es posible reducir el tiempo total de respuesta de una aplicación, permitiendo que la aplicación sirva en forma eficiente a los procesos empresariales que soporta; y a su vez con los mismos recursos incluya más usuarios dado el menor consumo de recursos por transacción individual.

1.4. HIPOTESIS DE LA INVESTIGACION

1.4.1. HIPOTESIS GENERAL

- El afinamiento de rendimiento de sistemas de gestión de bases de datos relacionales brindará rendimiento y escalabilidad a las soluciones informáticas aprovechando al máximo los recursos disponibles de la organización.

1.4.2. HIPOTESIS ESPECIFICAS

- El análisis y diseño de la aplicación es el punto de partida para asegurar un buen rendimiento.
- La empresa se beneficia más aplicando las directivas para un buen rendimiento en las fases iniciales o finales del ciclo de vida del software.
- El afinamiento SQL es la principal fuente de ganancia en rendimiento para aplicaciones que acceden a Base de Datos.

1.5. JUSTIFICACION Y DELIMITACION DE LA INVESTIGACION

1.5.1. JUSTIFICACION

Se justifica la tesis por lo siguiente:

- En el ámbito académico, porque permite comprobar en situaciones reales la validez de las hipótesis planteadas en la investigación.

- En el ámbito de negocios:

    - Mejorar los niveles de servicio a clientes.
    - Ahorro en costos, dado que la plataforma se hace más escalable y robusta, permitiendo procesos de negocio más ágiles sin invertir en nuevo hardware.
    - Conocer más a fondo la estructura de las aplicaciones y sus relaciones con otras, para responder de manera más rápida ante problemas futuros.

1.5.2. DELIMITACION

La investigación está enfocada en metodología y técnicas para la optimización de procesos en las Bases de Datos Relacionales Oracle.

CAPITULO II

MARCO TEÓRICO

2.1. DEFINICION DE AFINAMIENTO DE RENDIMIENTO

El afinamiento de rendimiento (Performance Tuning por sus siglas en inglés) es el arte de ofrecer la cantidad correcta de recursos a un sistema para que las aplicaciones ejecuten a la velocidad deseada al costo más bajo posible. Las soluciones que se generan van desde el aumento del número de recursos del sistema (adicionando más hardware), pasando por cambiar la lógica de las aplicaciones, y terminando en la alteración de la configuración de los componentes físicos y lógicos del sistema.

2.2. ANTECEDENTES

Un problema de rendimiento puede ser identificado por tiempos de respuesta excesivos. Esto ocurre usualmente a causa de un aumento de la carga de las aplicaciones, originando que parte del sistema alcance un límite en su capacidad para responder. Este límite dentro del sistema es referido como cuello de botella (bottleneck) [OR01].

El rendimiento de un sistema es diseñado y construido conforme este se va conceptualizando y construyendo, y no solo es un resultado natural del mismo que debemos aceptar y revisar una vez que el sistema se despliegue en producción. Los beneficios son mayores cuando los temas de rendimiento son considerados en las etapas de diseño y construcción, de acuerdo a la figura 02-01.

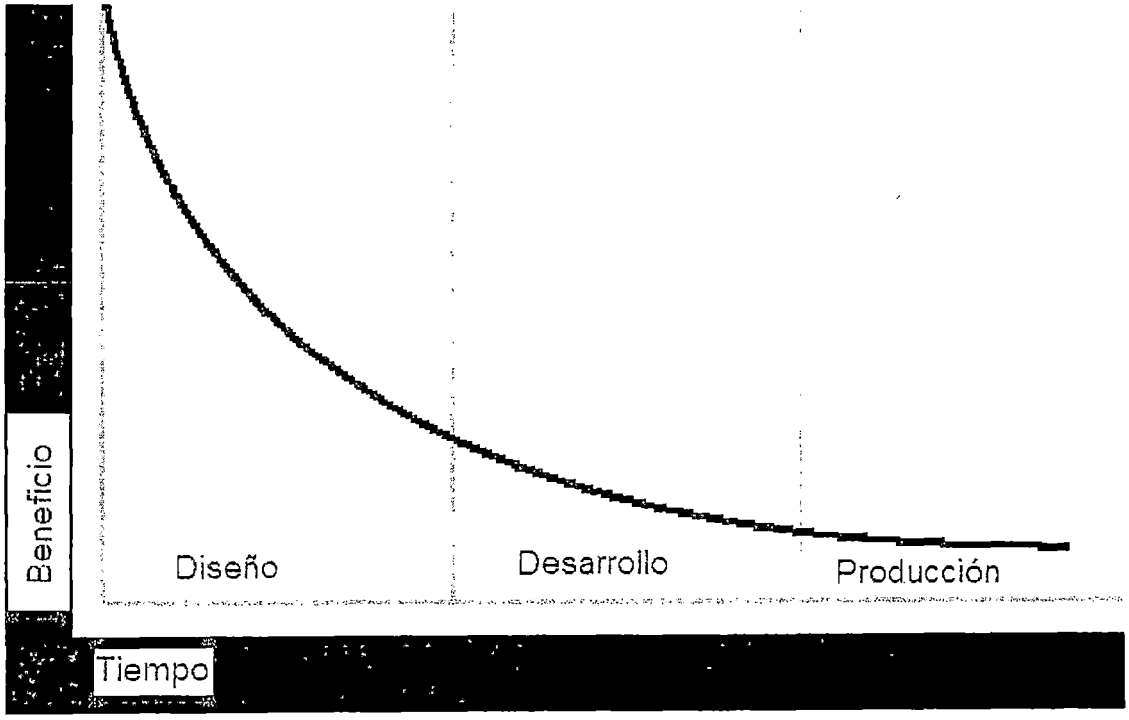


Figura 02-01. Beneficios de implantar el rendimiento en cada fase del ciclo de vida

La figura 02-01 muestra que los beneficios de aplicar estrategias para un óptimo rendimiento de un sistema son muy apreciables durante la fase de diseño del mismo. Estos beneficios bajan si estas estrategias se toman en cuenta recién en la fase de Desarrollo, y son de beneficio mínimo si las estrategias se toman en cuenta cuando el sistema ha sido desplegado en Producción.

Los problemas de rendimiento son a menudo el resultado de la contención en un recurso, o su agotamiento. Cuando un recurso del sistema está exhausto, el sistema es incapaz de escalar a mayores niveles de rendimiento. Si bien es cierto que aumentar hardware es una medida a tomar para mejorar el rendimiento, también es cierto que es una medida costosa, y en muchas situaciones no mejora los niveles de rendimiento a los que se quiere llegar.

Hay muchas razones para un pobre rendimiento en un sistema de gestión de Base de Datos relacional Oracle, que va desde una mala decisión de arquitectura, pasando por un diseño de Base de Datos inapropiado, problemas de programación y factores humanos.

Problemas con el Diseño y Desarrollo

Las causas de tiempos de respuesta excesivos se deben principalmente a los defectos encontrados en los programas de las aplicaciones y al diseño de la Base de Datos. Arreglar estos defectos son los más costosos en términos de esfuerzo y dinero cuando las aplicaciones se encuentran en la fase de producción.

De hecho, para evitar estos problemas se desarrollan nuevas metodologías complementarias a las tradicionales de Desarrollo de

Software que incluyen las actividades de ingeniería de rendimiento (performance engineering) en todas las fases del ciclo de vida del software (SDLC por sus siglas en inglés). Específicamente, podríamos mencionar a la metodología P* Software Development Life Cycle [AM01], la cual se describe en la figura 02-02.

Esta metodología, se inserta de manera activa en todas las fases del ciclo de vida del sistema. Detallando las interacciones por fase:

- En la fase de recolección de requerimientos, donde la metodología entrega la matriz de componentes principales del sistema, luego de evaluar los requerimientos de rendimiento.

- En la fase de Especificación Funcional y Arquitectura, se entrega a la metodología las especificaciones funcionales y los documentos de arquitectura. La metodología realiza un análisis del diseño del rendimiento, y entrega las Proyecciones de Diseño, las Proyecciones de Problemas y la Matriz de Rendimiento de la Infraestructura.

- En la fase de Diseño, se entrega a la metodología los Documentos de Diseño. La metodología establece la simulación de rendimiento y se simula el análisis estático de código. El análisis estático del código permite analizar los componentes y recursos de una aplicación sin tener que ejecutarla. La metodología entrega la Simulación de Rendimiento y las Pruebas de Concepto.

- En la fase de Codificación, se entrega a la metodología el código de primer nivel sin evaluación funcional. La metodología

realiza el análisis dinámico del código y caracteriza por perfiles el rendimiento. El análisis dinámico del código es la evaluación del mismo basado en su ejecución con datos de entrada seleccionados. La metodología entrega los perfiles de caracterización del código.

- En la fase de Evaluación, se entrega a la metodología la Evaluación Funcional completada. La metodología realiza la caracterización (Profiling), Simulación, Proyección de Capacidad, Evaluación del Rendimiento y Validación de la Infraestructura.

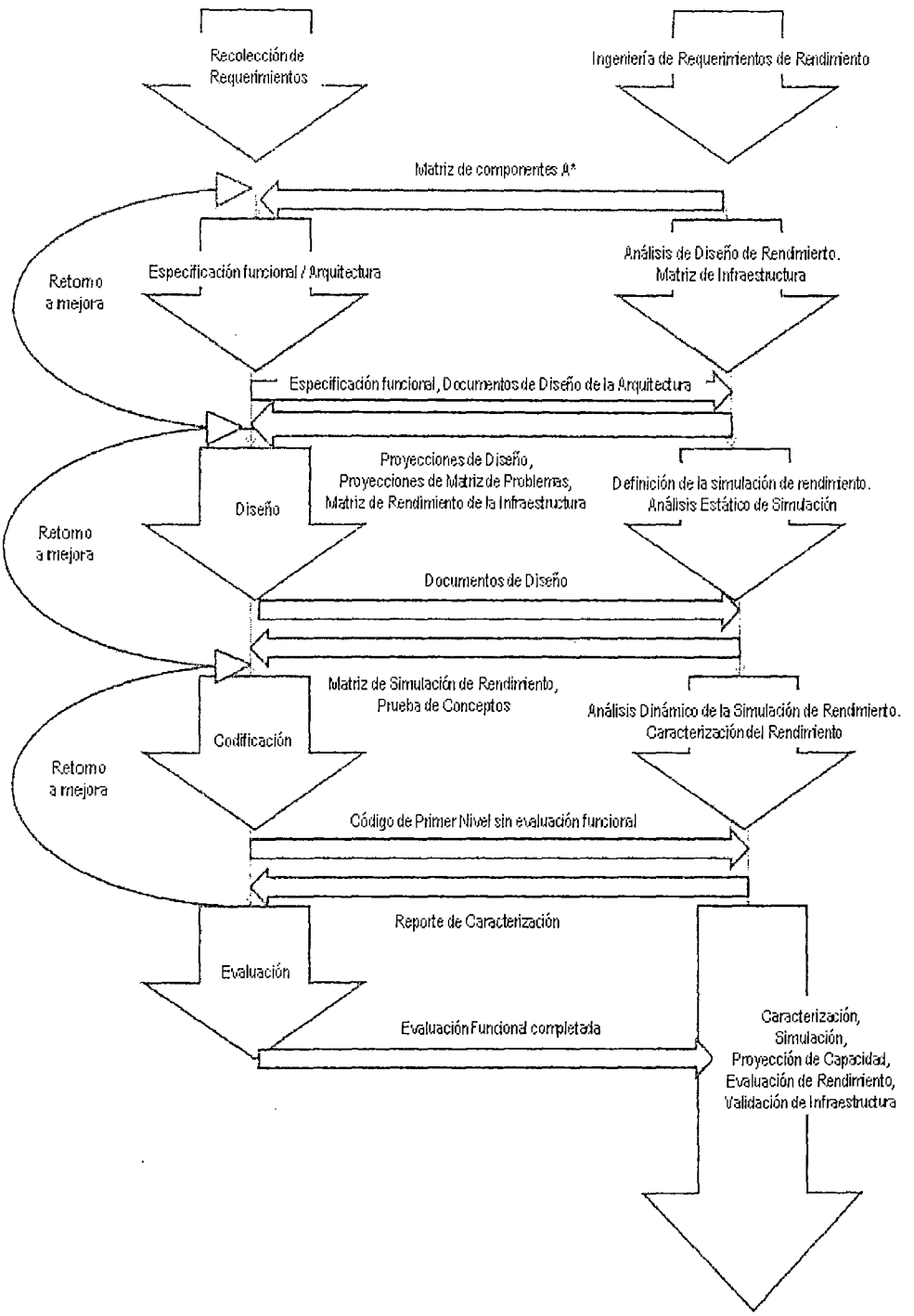


Figura 02-02. Metodología de "P* Ciclo de Vida de Desarrollo de Software"

Entiéndase al rendimiento como un Factor Crítico de Éxito en el negocio del software. Por ello, seguir una línea metodológica que incluya los asuntos de rendimiento desde la incepción de un sistema es un requisito obligatorio. En la figura 02-03 se puede apreciar una comparación entre cómo enfrentan los asuntos de rendimiento los enfoques tradicional y el orientado a Ingeniería de Rendimiento.

En el enfoque tradicional, la evaluación del rendimiento es pospuesta generalmente hasta poco antes de la fecha de liberación de un sistema, originando que no se cumpla con la fecha de liberación pactada y los ajustes de rendimiento se hagan cuando el sistema ya está en su fase de producción.

En el enfoque orientado a Ingeniería de Rendimiento, el rendimiento es analizado, construido y evaluado durante todas las fases del ciclo de vida, por ello, los plazos de entrega se cumplen con más exactitud y no se ejecutan ajustes de rendimiento sobre la marcha cuando el sistema está en su fase de producción.

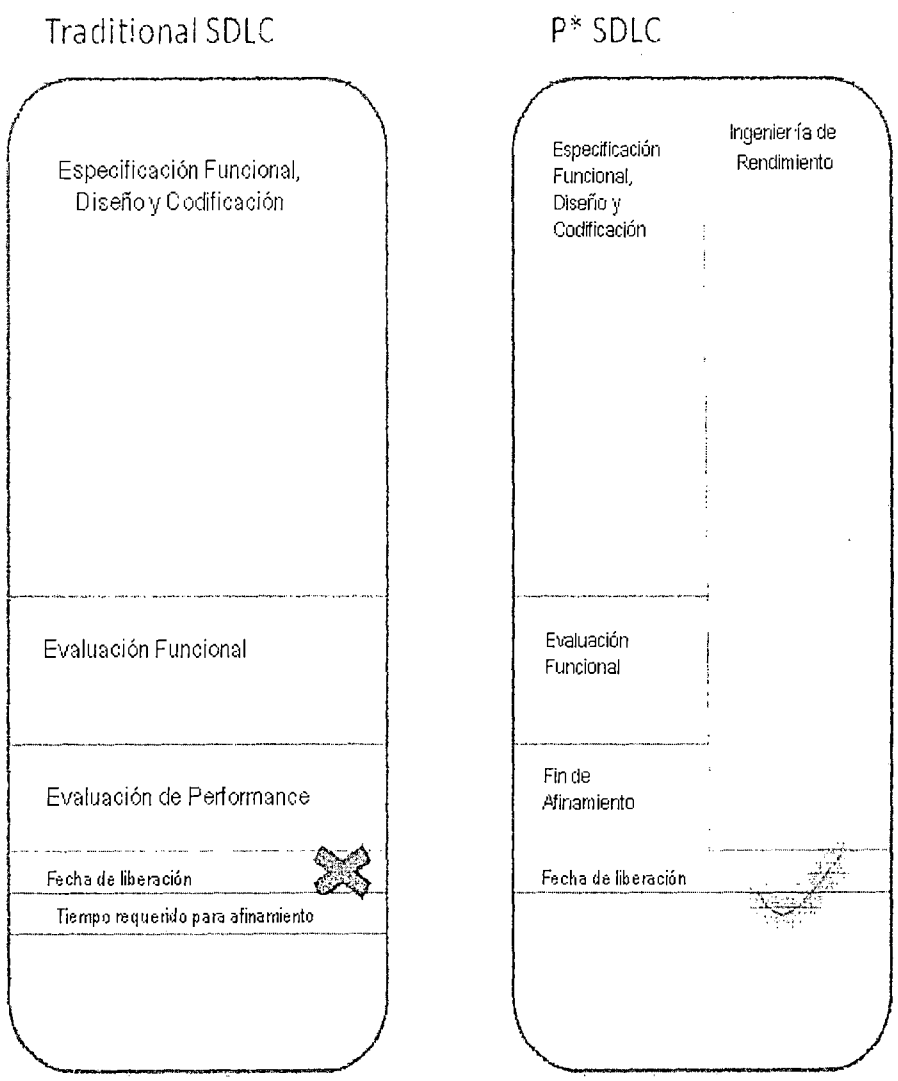


Figura 02-03. Alerta temprana de problemas de rendimiento bajo P* SDLC
(P* Ciclo de vida de desarrollo de software)

Diseño

La mejor manera de mejorar el rendimiento en el sistema es evitar que los problemas ocurran en primer lugar. Arreglar los problemas de un deficiente diseño del modelo de datos, o de una mala codificación no es tan simple como cambiar el valor de un parámetro o ejecutar un diagnóstico. La única forma de resolver este tipo de problemas es recodificar y remodelar.

Los problemas de diseño son típicamente causados por analistas y diseñadores que NO han:

- Considerado el rendimiento cuando se ha seleccionado una arquitectura. Por ejemplo, si se está considerando usar la arquitectura cliente-servidor, se debe reducir el número de paquetes que se transfiere a través de la red usando paquetes, procedimientos, restricciones (constraints), funciones y triggers.

- Considerado el rendimiento cuando se define el modelo de datos.

- Diseñado programas que son apropiados para una Base de Datos relacional.

- Diseñado programas que son apropiados para la configuración de hardware que se usa.

- Recibido el entrenamiento necesario de diseño para rendimiento.

- Separado los usuarios de procesos por lotes de los usuarios de procesos en línea.

- Permitido acceso de usuarios especializados en el diseño.

- Ejecutado un análisis adecuado sobre los volúmenes de datos, distribución de datos, uso, y picos y valles transaccionales.

- Puesto una estrategia de obsolescencia y eliminación de datos. Sin tal estrategia, cuando ocurra una conversión y los datos del sistema de producción sean reemplazados, se genere una sobrecarga adicional por una gran cantidad de datos no necesarios.

- Llegado a familiarizarse con las características de diseño que el motor de Base de Datos Oracle ofrece para apoyar en el rendimiento.

- Aprendido sobre los tipos de problemas de bloqueo que pueden potencialmente traer un sistema abajo.

Programación

Los problemas de programación son típicamente causados por los programadores cuyas sentencias SQL no ofrecen resultados a los usuarios en los tiempos planificados. Cabe destacar que en una aplicación empresarial, cada transacción de Base de Datos generalmente es caracterizada (profiled) y por ello se conoce a priori

su tiempo de respuesta más probable bajo circunstancias normales. Las principales causas de los problemas de rendimiento SQL son:

- Uso inapropiado de índices por incorrecta codificación.

- Uso del optimizador incorrecto. El motor de Base de Datos Oracle, se configura para que trabaje que con un optimizador determinado. El optimizador es el que resuelve los planes de ejecución de las sentencias SQL. Si este optimizador cambia o no es el adecuado para las tareas que se realizan, el rendimiento de las sentencias SQL es seriamente afectado.

- No estar consciente del comportamiento del optimizador basado en costos en la Base de Datos. El optimizador puede tener diferentes tipos de comportamiento en ambientes de Desarrollo y en Producción, de acuerdo a los parámetros con los que se rige, volúmenes de datos, estadísticas, opciones activadas, etc.

Base de Datos

Los problemas de Base de Datos son típicamente causados por Administradores de Base de Datos que NO han:

- Usado los recursos de la máquina efectivamente.

- Trabajado cercanamente con el administrador de sistemas cuando los archivos de Base de Datos se manejan con

tecnologías de manejo de datos en disco específicas, tales como RAID.

- Estructurado la Base de Datos para mejorar la distribución de los datos en los volúmenes de disco asignados al servidor de Base de Datos, con el fin de evitar los cuellos de botella por I/O en disco.

- Establecido los adecuados valores para los parámetros de la Base de Datos, como por ejemplo, para evitar la contención en redo logs y otros objetos.

- Establecido una adecuada toma de estadísticas para los objetos de la Base de Datos.

- Hecho efectivo uso de la memoria del servidor de Base de Datos.

- Sido proactivos e informar al equipo de desarrollo de aplicaciones de las sentencias SQL que ejecutan pobremente y de los problemas de contención.

Como consecuencia, podría ocurrir:

- Excesivo I/O en disco, o I/O desbalanceado (discos que trabajan más que otros).

- Base de Datos fragmentada, con muchas filas migradas (migrated rows) y filas encadenadas (chained rows).

- Base de Datos no indexada efectivamente.

- Problemas de contención y bloqueo en la Base de Datos.

Sistemas

Los problemas de sistemas ocurren como resultado de:

- Otros sistemas, que no sean Base de Datos, que afecten al motor de Base de Datos.

- Un sistema operativo no afinado. El sistema operativo, o software base, es el que gestiona las demandas de las aplicaciones por recursos del computador: CPU, memoria, disco, interfaces, etc. Como todo software, es susceptible de mantenimiento y afinamiento. Sin un adecuado afinamiento, el sistema operativo no podrá responder eficientemente a los requerimientos de las aplicaciones.

- Usar tecnología inapropiada para ciertos tipos de archivos en la Base de Datos. Por ejemplo, RAID-5 para redo log files.

- Un tamaño o configuración de máquina inadecuada para soportar el motor de Base de Datos Oracle.

Problemas con los Recursos del Sistema

Para conseguir el mejor rendimiento del sistema, se debe comprender cómo los cuatro componentes básicos del servidor de Base de Datos interaccionan y afectan el rendimiento del sistema:

Memoria

Los cuellos de botella en memoria ocurren cuando no hay suficiente memoria para acomodar las necesidades del sitio. Cuando esto sucede, ocurre excesiva paginación (paging: mover porciones de procesos a disco) e intercambio (swapping: transferir procesos completos de memoria a disco) para liberar memoria.

Disco y Controladores de Disco

Los cuellos de botella en disco y los controladores de disco ocurren cuando uno o más discos, o un controlador de disco, exceden su tasa recomendada de I/O.

CPU

Los cuellos de botella en CPU ocurren cuando el sistema operativo o los programas del usuario están haciendo muchas demandas al CPU. Esto es a menudo causado por excesiva paginación o swapping. Podría ser consecuencia de un número inadecuado de CPUs.

Red

Los cuellos de botella de red ocurren cuando la cantidad de tráfico en la red es mucha o cuando ocurren colisiones. Los cuellos de botella de red son de interés particular cuando se está en un ambiente cliente-servidor o de n-capas.

Hay que recordar que los problemas en memoria, discos, CPUs y la red no existen en forma aislada. Cada uno de estos problemas afecta a los otros recursos también. Al mismo tiempo, la mejora en un recurso puede causar un mejor rendimiento en los otros componentes. Por ejemplo:

- Si su organización compra más memoria, la CPU puede necesitar gastar menos tiempo manejando las operaciones de paginación y swapping que ocurren cuando la memoria es escasa. Esto ayuda a evitar cuellos de botella en CPU también.

- Si se tiene un cuello de botella en disco, se podría ser capaz de usar la memoria para almacenar más datos, por lo tanto evitar de leer los datos desde disco una segunda vez.

- Si se tiene un problema en el tráfico de red en un ambiente de n-capas, se podría ser capaz de mejorar las cosas usando más memoria, más brazos en disco, y CPU tanto en los servidores como en los clientes para disminuir el tráfico a través de la red.

En el afinamiento de rendimiento, la meta es usar todos los recursos en forma eficiente, llegando a aprovechar su capacidad instalada sin llegar a situaciones de contención. Si el consumo de los recursos del

servidor es bajo de manera sostenida, indica que tenemos recursos de sobra y que se ha invertido en más de lo necesario en hardware. El sistema debe estar bien utilizado, sin llegar a situaciones de cuellos de botella. [FU01]

¿Qué constituye un sistema afinado idealmente? Es materia de debate. Se podría decir que debemos llegar a afinar un servidor de tal manera que:

- La memoria esté usada casi al 100%.
- La carga de disco esté repartida uniformemente entre todos los dispositivos, y que todos los discos estén operando marginalmente dentro de las máximos tasas recomendadas de I/O.
- La CPU esté utilizada casi al 100% durante períodos pico, sin que los programas de usuario estén en cola de espera.
- Si se tiene múltiples CPUs, cada CPU debe compartir la carga equitativamente.
- El tráfico de red esté marginalmente debajo del máximo recomendado, sin colisiones.
- Una cantidad insignificante de paging y swapping esté ocurriendo.
- La cantidad de transacciones por unidad de tiempo (throughput) y los tiempos de respuesta cumplan con los estándares establecidos en la organización.

La clave es que todos los recursos trabajen juntos de manera coordinada sin llegar a situaciones de contención. En la figura 02-04, se muestra un sistema no afinado. En tal sistema, un componente se degrada antes que los otros. En este caso particular, cuando se intenta lograr más throughput, esto es impedido por el cuello de

botella en memoria (esto es, toda la memoria ha sido usada y la paginación y el swapping ha sido excesiva). En este ejemplo, si se añade más memoria, la CPU será el próximo ítem que sea un cuello de botella en el rendimiento, seguidos luego por el disco y la red. [GU01]

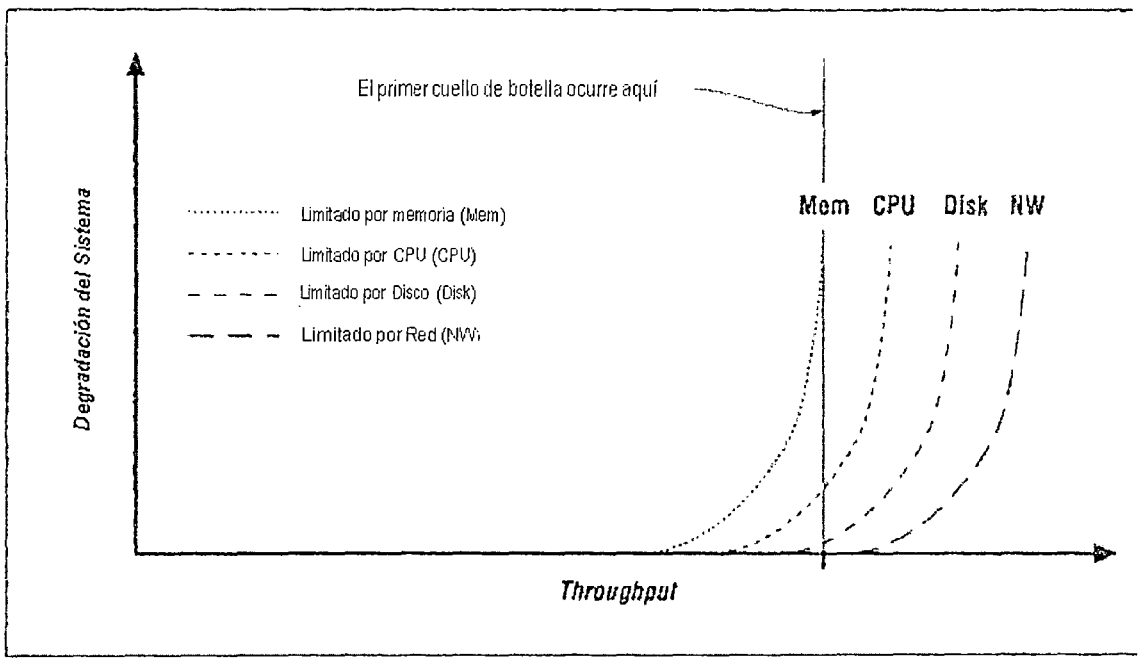


Figura 02-04. Un sistema no afinado

En un ambiente perfectamente afinado, por otro lado, todos los factores se degradan simultáneamente. Como se ilustra en la figura 02-05, hasta que se alcance el punto en el cual el rendimiento empiece a degradarse, todos los recursos del sistema están balanceados. Si el throughput requerido en el sistema es substancialmente menor al que empiece a degradar el rendimiento (aún luego de una expansión planeada a su aplicación), esto es indicativo que se ha comprado más hardware que el que realmente se requería. [GU01]

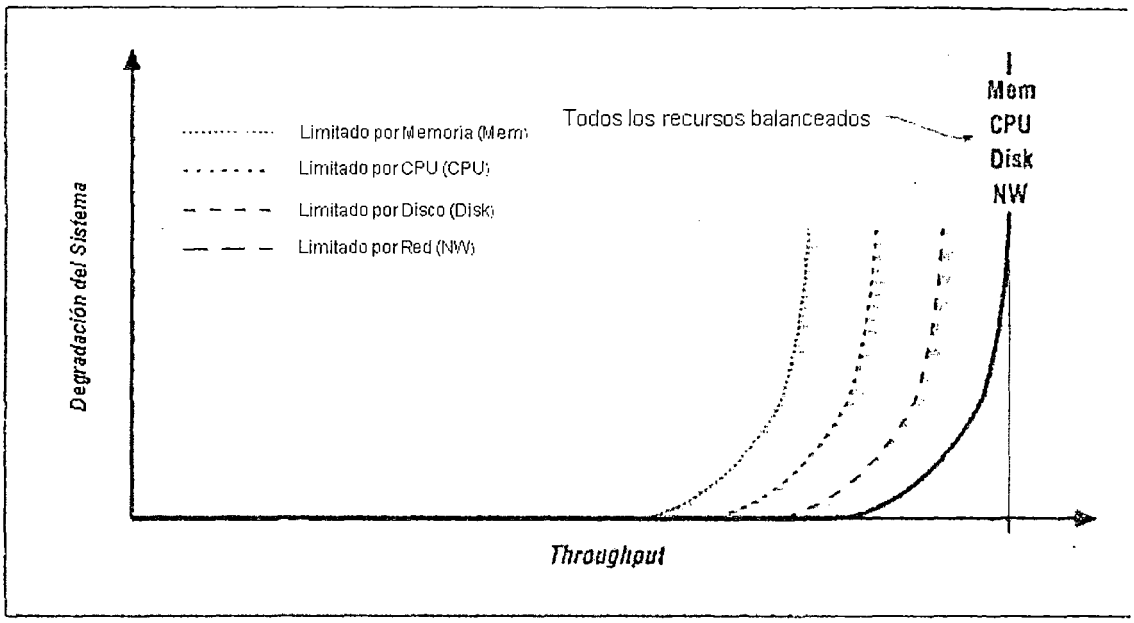


Figura 02-05. Un sistema afinado

2.3. MARCO TEÓRICO

El afinamiento del rendimiento tiene como objetivo aprovechar al máximo las capacidades instaladas de los recursos informáticos. Normalmente, los componentes se instalan con parámetros genéricos y no responden a necesidades particulares. Por ello, el afinamiento de rendimiento (performance tuning) busca la mejor configuración y distribución de los componentes de hardware, software y networking para soportar la demanda real de los mismos en una organización específica. [OR02]

La escalabilidad tiene como objetivo lograr que los componentes de la solución informática (hardware, software y networking) puedan soportar los aumentos de la carga transaccional aumentando éstos en forma proporcional. Por ejemplo, si se prevé una carga adicional de 50% de transacciones, la plataforma hardware es escalable si soporta esa carga aumentando el 50% de sus recursos principales, entiéndase CPU o memoria.

El afinamiento sistemático del rendimiento sigue los siguientes pasos: [WI02]

- Evaluar el problema y establecer valores que categoricen un comportamiento aceptable.

- Medir el rendimiento del sistema antes de la modificación.

- Identificar la parte del sistema que es crítica para mejorar el rendimiento. Esta parte es conocida como el cuello de botella.

- Modificar la parte del sistema para remover el cuello de botella.

- Medir el rendimiento del sistema después de la modificación.

Las técnicas más usadas para mejorar el rendimiento son optimización de código, balanceo de cargas, estrategias de caching (almacenamiento intermedio), computación distribuida y auto-afinamiento. El método de mejora de rendimiento incluye algunas de estas y otras técnicas para mejorar el rendimiento.

El fin último de un trabajo de afinamiento es disminuir el tiempo de respuesta de las transacciones de Base de Datos (User Response Time). Dentro de ese tiempo de respuesta, el tiempo de Base de Datos (DB Time) es una parte de él. [OR02]

En la figura 02-06 se muestra el tiempo de Base de Datos como parte integrante del tiempo total de respuesta que obtiene el usuario de una aplicación. El tiempo de Base de Datos representa el tiempo total consumido en llamadas a Base de Datos. [OR03]

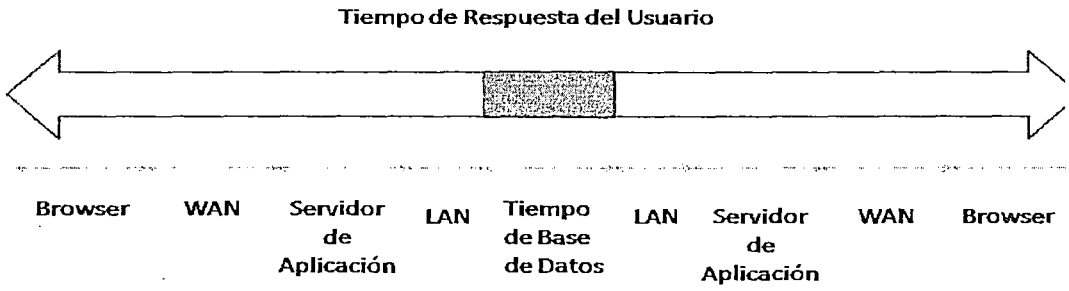


Figura 02-06. Componentes del Tiempo de Respuesta de una aplicación

En la realidad, el DB Time se reparte a lo largo de todo el tiempo que dura la transacción del usuario. En la figura 02-07 se detalla cómo se agrega la estadística DB Time a lo largo de la vida de una transacción en una aplicación.

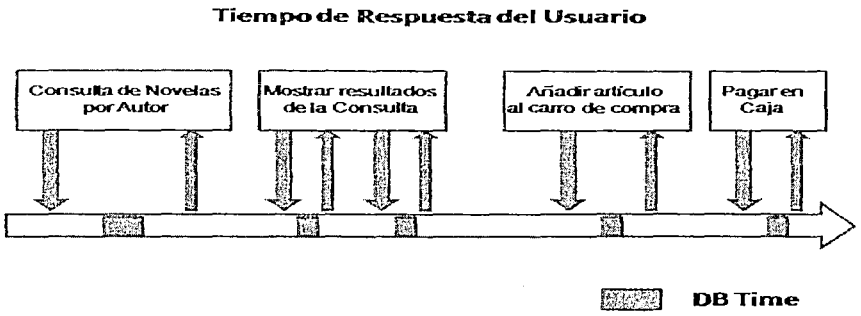


Figura 02-07. Repartición del tiempo de Base de Datos en el Tiempo de Respuesta

DB Time es calculado sumando el tiempo de CPU y el tiempo de espera de todas las sesiones que no están en espera por eventos sin actividad (idle events). [OR03]

Método de Mejora de Rendimiento de Oracle

1. Conseguir retroalimentación de los usuarios (feedback). Determinar el ámbito del proyecto de rendimiento y metas de rendimiento. El proceso es clave en futuros planeamientos de capacidades.

2. Conseguir un conjunto completo de estadísticas del Sistema Operativo, Base de Datos y Aplicaciones.

3. Revisión de sanidad (Sanity Check) de los sistemas operativos de todas las máquinas involucradas con el rendimiento del usuario. Por revisión de sanidad se entiende buscar recursos de hardware y sistema operativo que están totalmente utilizados. Cualquier recurso usado en exceso representa un síntoma a ser analizado posteriormente.

4. Revisar errores comunes cometidos con Oracle, y determinar si alguno de ellos representa un potencial problema.

5. Construir un modelo conceptual de lo que está sucediendo en el sistema usando los síntomas como pistas para comprender lo que causó los problemas de rendimiento.

6. Proponer una serie de acciones a tomar para llegar a un comportamiento deseado del sistema. Aplicarlas en un orden tal que los principales beneficios sean logrados primero. La premisa principal en el trabajo de Afinamiento de Rendimiento es que se cambia una cosa a la vez, y luego se mide las diferencias. Si tal enfoque no puede ser aplicado por restricciones de negocio, aplicar los cambios que no tengan interacción entre ellos.

7. Validar que los cambios hechos produzcan los resultados deseados.

Repetir los últimos tres (3) pasos hasta que las metas de rendimiento sean cumplidas o llegue a ser imposible debido a otras restricciones.

Impacto económico de la mejora del rendimiento y escalabilidad

El afinamiento de rendimiento y escalabilidad de las plataformas tecnológicas de información se sustentan económicamente y su adecuada aplicación redunda directamente en los resultados financieros de un negocio; dado que fomenta la productividad, el ahorro en nuevas inversiones, el downsizing para bajar los costos operativos, y permite obtener mejores resultados de un activo intangible de la empresa.

Escalabilidad (Scaleup)

Es la habilidad de una aplicación de mantener el tiempo de respuesta mientras el tamaño de un proceso o el volumen de transacciones se incrementa adicionando recursos físicos, tales como procesadores, memoria o disco. [TU01]

En las aplicaciones de Base de Datos, la escalabilidad puede ser batch o transaccional. Con la escalabilidad batch, procesos en lote más grandes pueden ser soportados sin pérdida de tiempo de respuesta. Con escalabilidad transaccional, un número más grande de transacciones puede ser soportado sin pérdida de tiempo de respuesta. En ambos casos, el tiempo de respuesta generalmente es mantenido por la adición de nuevos procesadores. Por ejemplo, un sistema de 4 procesadores puede ofrecer el mismo tiempo de respuesta con una carga de 400 transacciones por minuto que el tiempo de respuesta de un sistema con 1 procesador que soporta una carga de 100 transacciones por minuto.

La escalabilidad es medida en términos de cuánto volumen transaccional puede ser incrementado añadiendo más procesadores mientras se mantiene constante el tiempo de respuesta.

La Escalabilidad (Scaleup) es calculada usando la ecuación 4.1 (Ec. 4.1) presentada a continuación:

$$Scaleup = Volumen(m) / Volumen(1) \quad \ldots\ldots\ldots\ldots \text{(Ec. 4.1)}$$

Donde:

Volumen(m) es el volumen transaccional ejecutado en un intervalo de tiempo usando "m" procesadores.

Volumen(1) es el volumen transaccional ejecutado en el mismo intervalo de tiempo usando 1 procesador.

Para el ejemplo previo, reemplazando valores en (Ec. 4.1):

$$Scaleup = 400 / 100$$

$$Scaleup = 4$$

Este Scaleup de 4 es alcanzado con 4 procesadores. El ejemplo es ideal. La figura 02-08 muestra la tendencia de la escalabilidad ideal. Nótese que el tiempo de respuesta se mantiene constante incrementando el número de procesadores mientras el volumen transaccional se incrementa.

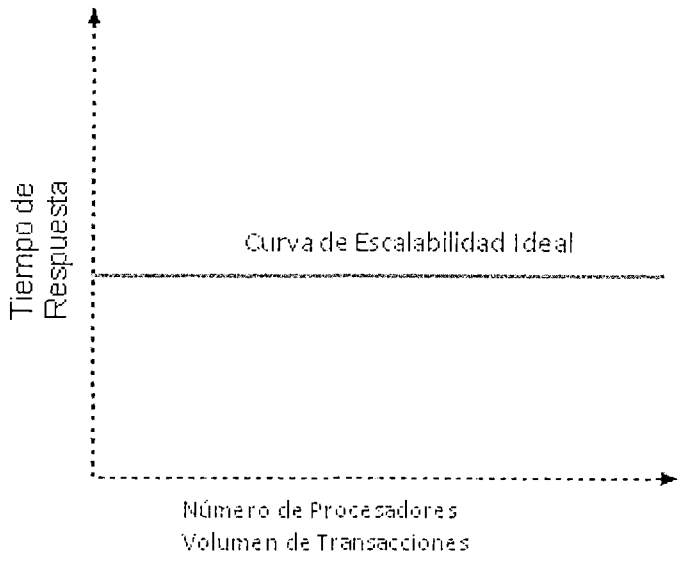


Figura 02-08. Curva ideal de escalabilidad

La curva, o línea recta, representa un ideal. En la realidad, después de un cierto punto, el tiempo de respuesta se incrementa para volúmenes transaccionales mayores aun si se añaden más procesadores.

Decisiones (Trade-offs) entre Tiempo de respuesta y Throughput

Las variables de tiempo de respuesta y throughput requieren hacer un balance de recursos para llegar a niveles deseados para ambos. Si afinamos teniendo en cuenta sólo una variable, sacrificamos los resultados de la otra.

Tiempo de respuesta

A consecuencia que el tiempo de respuesta es igual al tiempo de servicio más el tiempo de espera, se puede incrementar el rendimiento en dos formas: Reduciendo el tiempo de servicio (service time) o reduciendo el tiempo de espera (wait time).

La figura 02-09 muestra diez (10) tareas independientes compitiendo por un único recurso:

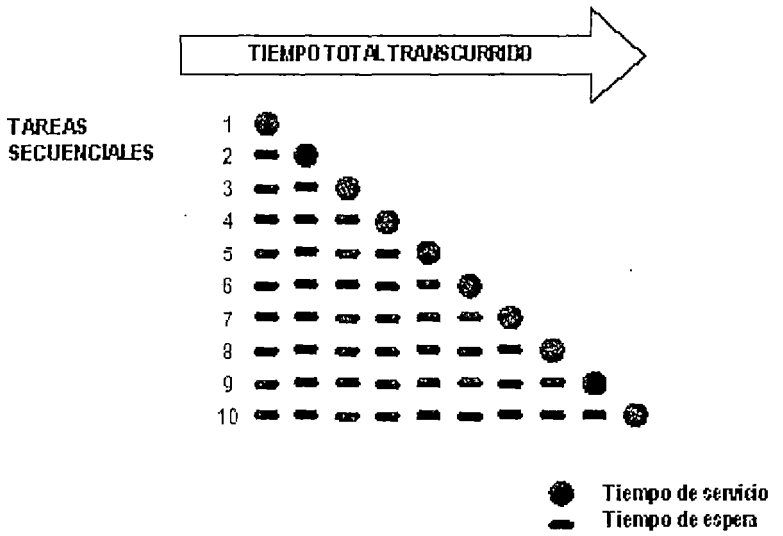


Figura 02-09. Procesamiento secuencial de múltiples tareas independientes

En este caso, sólo la tarea 1 corre sin tener que esperar. La tarea 2 debe esperar hasta que la tarea 1 haya completado; la tarea 3 hasta que las tareas 1 y 2 hayan culminado, y así sucesivamente. Aunque la

figura 02-09 muestra las tareas independientes del mismo tamaño, el tamaño de ellas puede variar.

En procesamiento paralelo, si se tienen múltiples recursos, entonces más recursos pueden ser asignados a las tareas. Cada tarea independiente ejecuta inmediatamente usando su propio recurso, no hay tiempo de espera.

Throughput (Transacciones por unidad de tiempo)

El throughput del sistema es igual a la cantidad de trabajo ejecutado en un intervalo de tiempo. Existen dos técnicas para aumentar el throughput:

- Hacer que más trabajo sea hecho con los mismos recursos, reduciendo el tiempo de servicio (service time).

- Hacer que el trabajo sea hecho más rápido reduciendo el tiempo total de respuesta, para ello reduciendo el tiempo de espera (wait time). Esto se logra duplicando los recursos que son motivo de espera para los usuarios. Por ejemplo, si hay contenciones en CPU, aumentar más CPUs.

Wait Time (Tiempo de Espera)

El tiempo de servicio para una tarea puede permanecer constante, pero el tiempo de espera se incrementa a medida que la contención aumenta. Si muchos usuarios están esperando por un servicio que demora un segundo, entonces el décimo usuario debe esperar 9 segundos para obtener un servicio que demora 1 segundo.

La figura 02-10 muestra que el tiempo de espera (wait time) aumenta a medida que se incrementa la contención en un recurso.

Figura 02-10. Aumento del Tiempo de Espera (Wait Time) ante el incremento de la contención en un recurso

Recursos críticos

Los recursos tales como la CPU, memoria, capacidad de I/O y el ancho de banda de la red son claves para reducir el tiempo de servicio. La adición de recursos posibilitan el aumento del throughput y la disminución de los tiempos de servicio. El rendimiento depende de lo siguiente:

- ¿Cuántos recursos están disponibles?
- ¿Cuántos clientes necesitan el recurso?

- ¿Cuánto tiempo deben esperar por el recurso?
- ¿Cuánto tiempo ellos deben tomar el recurso?

La figura 02-11 muestra que si el número de unidades requeridas aumenta, el tiempo para completar el servicio lo hace proporcionalmente.

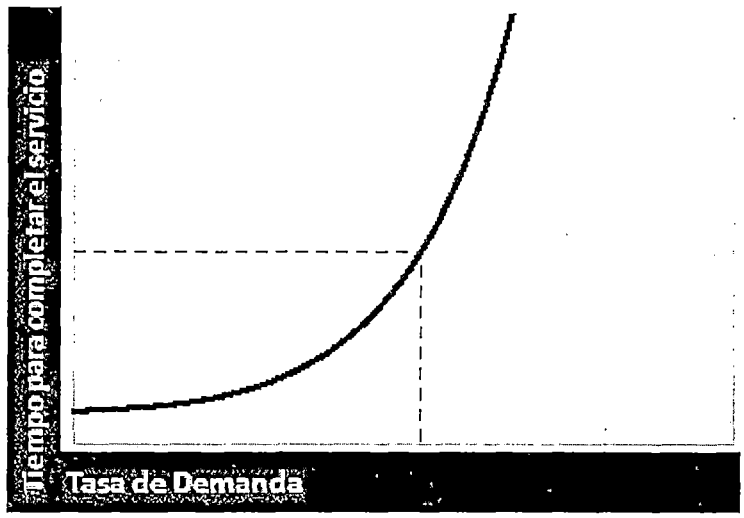


Figura 02-11. Tiempo para completar un servicio VS Tasa de Demanda

Para manejar esta situación se tienen dos opciones:

- Se puede limitar la tasa de demanda para mantener tiempos de respuesta aceptables.
- Alternativamente, se pueden añadir múltiples recursos: CPU o disco por ejemplo.

Efectos de la demanda excesiva

La demanda excesiva trae como consecuencia lo siguiente:

- Incrementa el tiempo de respuesta
- Reduce el throughput

Si hay alguna posibilidad que la tasa de demanda exceda el nivel de throughput aceptable, entonces de debe implantar un limitante a la demanda.

La figura 02-12 muestra la relación entre las transacciones por unidad de tiempo (throughput) y la tasa de demanda. La relación que se observa es que a mayores tasas de demanda el throughput decae, es decir, se disminuye la capacidad de procesar un cierto número de transacciones por unidad de tiempo. Si hay alguna posibilidad que la tasa de demanda exceda el throughput objetivo, entonces se debe imponer un límite a la demanda.

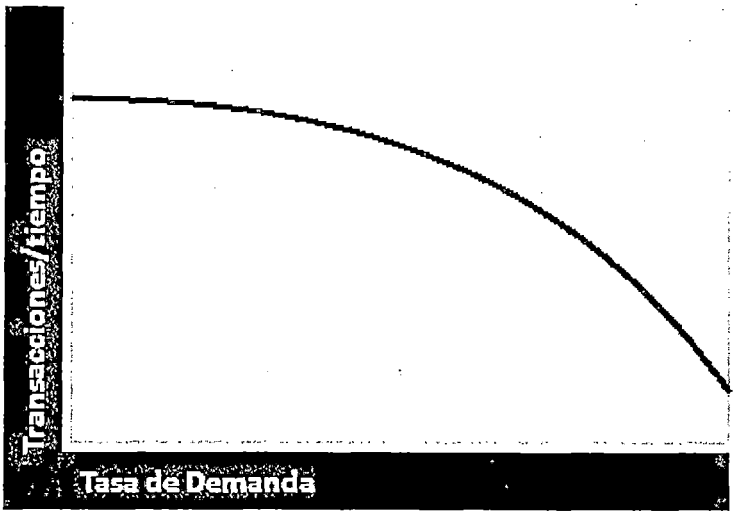


Figura 02-12. Transacciones/tiempo VS Tasa de Demanda

Ajustes para liberar problemas

Se pueden liberar problemas de rendimiento haciendo los siguientes ajustes:

Ajustar la unidad de consumo

Usando menos recursos por transacción o reduciendo el tiempo de servicio. O se pueden tomar enfoques tales como reducir el número de I/Os por transacción.

Ajustar la demanda funcional

Reprogramando o redistribuyendo la carga de trabajo.

Ajustar la capacidad

Incrementando o reasignando recursos físicos.

La figura 02-13 ilustra un caso para ajustar la capacidad instalada y la demanda funcional. Si las horas pico de atención a clientes es de 9:00 horas a 10:30 horas y de 13:00 horas a 14:30 horas, entonces se pueden ejecutar los procesos por lote (batch) luego de las 14:30 horas, cuando hay más capacidad. De esta manera, se puede distribuir la demanda de recursos más uniformemente

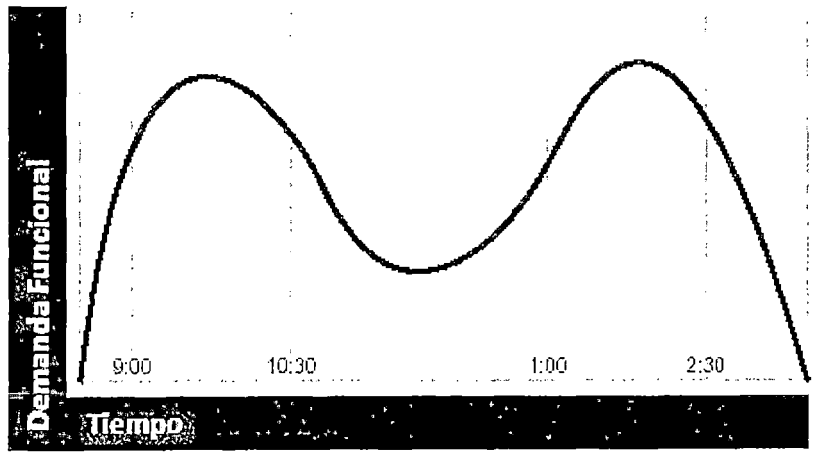


Figura 02-13. Ajustando Capacidad y Demanda funcional

Responsables del afinamiento

Todos los involucrados con el sistema tienen un rol en el afinamiento del rendimiento. Cuando ellos comunican y documentan las

características del sistema la tarea de afinamiento se hace más fácil y rápida.

Entonces, cada persona que tiene que ver con un aspecto de definición, diseño, construcción o administración del sistema es directa o indirectamente responsable de su rendimiento. La figura 02-14 muestra los actores principales por cada fase del ciclo de vida del sistema.

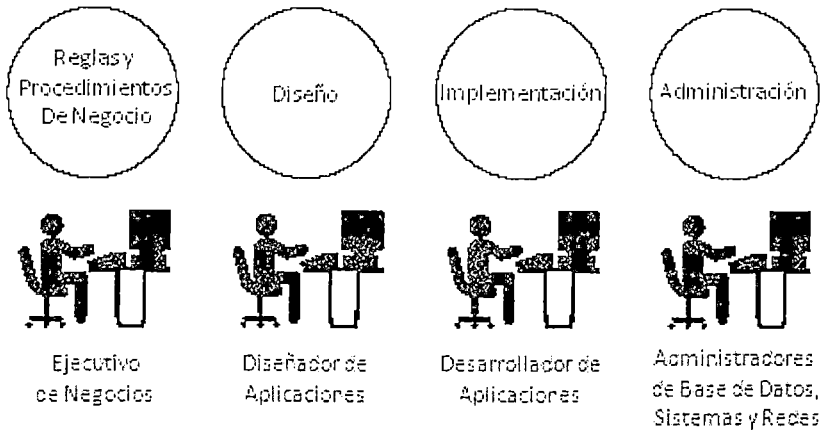


Figura 02-14. Afinadores del sistema

- Los ejecutivos del negocio deben definir y luego reexaminar las reglas y procedimientos del negocio para ofrecer un modelo claro y adecuado al diseñador de la aplicación. Ellos deben identificar los tipos específicos de reglas y procedimientos que influyen en el rendimiento del sistema entero.

- Los diseñadores de aplicaciones deben diseñar alrededor de los cuellos de botella potenciales. Ellos deben comunicar el diseño del sistema a todos los involucrados para que estos comprendan el flujo de datos de la aplicación.

- Los desarrolladores de la aplicación deben comunicar las estrategias de implementación seleccionadas para que los módulos y las sentencias SQL sean identificadas de manera rápida y fácil durante el afinamiento SQL.

- Los administradores de Base de Datos (DBAs) deben cuidadosamente monitorear y documentar la actividad del sistema para que ellos puedan identificar y corregir rendimientos inusuales en el sistema.

- Los administradores de hardware y software deben documentar y comunicar la configuración de los equipos a los involucrados para obtener un buen diseño y administración.

Las decisiones hechas en el desarrollo y diseño de las aplicaciones tienen el más grande efecto sobre el rendimiento. Una vez que la aplicación es desplegada, usualmente el Administrador de la Base de Datos tiene la responsabilidad principal en el afinamiento.

Estableciendo Objetivos de Rendimiento

Si se está diseñando o manteniendo un sistema, se deben establecer metas de rendimiento específicas para conocer cuándo afinar. Se consume demasiado tiempo afinando un sistema si se alteran parámetros de inicialización o sentencias SQL sin una meta específica.

Cuando se diseña el sistema, se debe establecer una meta tal como "lograr un tiempo de respuesta en el ingreso de órdenes de compra de

menos de tres segundos". Si la aplicación no cumple tal meta, identificar el cuello de botella que previene lograr la meta (por ejemplo, contención en I/O), determinar la causa raíz, y tomar la acción correctiva correspondiente. [KR01]

Durante el desarrollo, evaluar la aplicación para determinar si cumple con las metas de rendimiento establecidas en el diseño. Este es un requisito indispensable antes de desplegar la aplicación.

El afinamiento del rendimiento es usualmente una serie de decisiones (trade-offs). Una vez que se han identificado los cuellos de botella, se podría sacrificar otros recursos del sistema para lograr los resultados deseados. Por ejemplo, si el I/O es un problema, se puede necesitar comprar más memoria o más discos. Si la compra no es posible, se puede limitar la concurrencia al sistema para lograr el rendimiento deseado. Sin embargo, con las metas de rendimiento claramente definidas, la decisión sobre qué recurso potenciar o intercambiar a fin de mejorar el rendimiento es más fácil dado que se han identificado las áreas más importantes.

Siempre tener en cuenta que las metas de rendimiento en ningún momento deben prevalecer a las metas de recuperación de los datos. El rendimiento es importante, pero la habilidad de recuperar los datos es crítica. Estas metas no deben ser antagónicas. [HA01]

Estableciendo las expectativas de los Usuarios

Los desarrolladores de las aplicaciones y los administradores de Base de Datos deben cuidadosamente establecer expectativas de rendimiento apropiadas para los usuarios. Cuando el sistema ejecuta una operación particularmente complicada, el tiempo de respuesta

puede ser más lento que cuando está ejecutando una operación simple. En este caso, un tiempo de respuesta más lento es razonable.

Si un DBA promete un tiempo de respuesta de un (1) segundo, considere cómo esto puede ser interpretado. El DBA podría referirse a que la operación tomaría 1 segundo en la Base de Datos y podría ser capaz de lograr esta meta. Sin embargo, los usuarios que se encuentren en locaciones remotas podrían experimentar una latencia de un par de segundos por el tráfico de la red; pero ante un mensaje mal dado como meta de rendimiento, ellos sentirían que el tiempo de respuesta está degradado. [MI01]

Evaluando el Rendimiento

Con las metas del rendimiento claramente definidas, se puede determinar prontamente cuándo el afinamiento de rendimiento ha sido exitoso. El éxito depende de:

- Los objetivos funcionales que se ha establecido con la comunidad usuaria.
- La habilidad de medir objetivamente si la meta ha sido cumplida.
- La habilidad de tomar acciones correctivas para sobrellevar las excepciones.

CAPITULO III

APLICACIÓN DE AFINAMIENTO DE RENDIMIENTO EN GRUPO SAN FERNANDO S.A.

3.1. DESCRIPCION DEL GRUPO EMPRESARIAL SAN FERNANDO S.A.

Historia del Grupo San Fernando S.A.

San Fernando, fue fundado en 1948 por el señor Julio Soichi Ikeda Tanimoto y se inició como un negocio familiar con la crianza de patos.

Con el apoyo de sus hijos, en 1963 inicia la crianza de pollos parrilleros, y en 1971 la crianza y comercialización de pavos, después de eso en 1972 apertura la primera tienda San Fernando orientada al comercio detallista, convirtiéndose desde 1994 en los Multimarket San Fernando.

Con el objetivo de controlar el proceso productivo en su totalidad y garantizar la calidad de sus productos, San Fernando decide integrar verticalmente su negocio, iniciando la crianza de aves reproductoras que le permite autoabastecerse

de pollitos bebé y en 1977 pone en operación su primera planta de alimento balanceado.

Con la experiencia y los buenos resultados obtenidos, San Fernando decide incursionar en dos nuevos negocios: el de huevos comerciales en 1979 y la crianza de cerdos en 1986.

Durante más de medio siglo San Fernando ha demostrado invalorable constancia y creatividad para adaptarse a las nuevas tecnologías y requerimientos de las empresas modernas; para exceder las expectativas de un cliente cada día más exigente.

Hoy el Grupo San Fernando comercializa sus productos en todo el Perú, contando con: 7 plantas de incubación, 2 plantas de alimentos balanceados, 104 granjas de pollos, 8 granjas de pavos, 5 granjas de cerdos, 12 granjas de huevos, 2 plantas de beneficio de aves y 1 planta procesadora de productos carnicos. Para la distribución de sus productos cuenta con una extensa flota de camiones equipados con sistemas de refrigeración.

Gracias a sus estándares de calidad reconocidos internacionalmente, los productos San Fernando han logrado ingresar a mercados exigentes como Japón, México, Argentina, Venezuela, Colombia, Ecuador, Bolivia y El Salvador.

HITOS EN LA HISTORIA DE SAN FERNANDO

1948: Inicia sus actividades con 35 patas reproductoras
1963: Ingresa al negocio de POLLOS

1971: Ingresa al negocio de PAVOS
1976: Abre la primera tienda San Fernando
1977: Inaugura su propia planta de alimentos balanceados
1979: Ingresa al negocio de HUEVOS
1980: Inicia las ventas de genética
1986: Ingresa al negocio de CERDOS y engorde de ganado
1992: Implantación de la Calidad Total
1995: Asociación con empresas avícolas
1995: Inaugura planta moderna de Productos Procesados
1996: Reingeniería empresarial
1997: Proceso de consolidación operativa y financiera
1999: Inicia sus actividades en la Agricultura
2000: Estandarización (ISO 9000), Sistemas ERP, Gestión por Competencias
2002: Sistema de Gestión Ambiental (ISO 14000)

Empresas del Grupo

Distribuidora Wanka S.A.

Fue fundada en la provincia de Huancayo, departamento de Junín en 1988 y se dedica a la venta de carne y productos cárnicos; distribución y comercialización de productos avícolas y aves vivas.

Los Principales productos que distribuye son Pollo beneficiado, Pollo vivo, Embutidos, Pavo, Gallina, Cerdo y Huevos; ubicándolos en: Huancayo, Concepción, La Oroya, Jauja, Ayacucho y Cerro de Pasco.

Prodasa

Fue fundada en Bolivia el 19 de junio de 1995. Se dedica a la Producción y Comercialización de aves para consumo humano, desarrollando su cobertura en Santa Cruz, Tarija y La Paz.

Dentro de sus principales productos están: Huevo Incubable, Pollo BB, Pollo vivo, Pollo beneficiado y otros.

Agropecuaria CHIMU

Fue fundada el 10 de Junio de 1985 el departamento La Libertad, constituyéndose como una empresa de Producción y Comercialización de aves para consumo humano como Pollo vivo y beneficiado.

Tiene operaciones en Tumbes, Piura, Chiclayo, Jaén, Cajamarca, Chimbote, Huaraz, Lima, Huancayo, Huanuco, Tingo María.

Pesquera San Andrés del Sur S.A.

En 1983 se incorpora al mercado tomando como rubro principal la Producción y Comercialización de conservas de pescado y mariscos; obtiene presencia en todo el Perú, pues su principal cliente es DEMSA con la marca A1, quien tiene una importante participación en el mercado local.

También exporta sus productos a Chile, Panamá, República Dominicana, Estados Unidos, Uruguay y El Salvador. Dentro de los principales productos que distribuye se encuentran conservas de pescado en salsa de tomate, agua, sal y aceite vegetal. También mixto de mariscos, choros en agua y sal y, principalmente grated de pescado en agua y sal.

Misión Empresarial

"Contribuir al bienestar de la humanidad suministrando alimentos de consumo masivo en el mercado global."

El éxito de la misión se logra:

- Con personal que practique los valores de la empresa, competente, con espíritu de superación, comprometido con el cambio y promotor del trabajo en equipo.

- Con el desarrollo de una organización ágil, eficaz e innovadora que obtenga ventajas competitivas y sea rentable.

- Con el mejoramiento continuo de procesos, productos y servicios, en estrecha cooperación con nuestros proveedores, para satisfacer y exceder las expectativas del cliente.

- Con una cultura basada en los valores de honestidad, lealtad, laboriosidad, responsabilidad y respeto, la práctica de la filosofía de calidad total y una clara actitud de liderazgo.

- Con acciones orientadas a proteger y conservar el medio ambiente.

- La riqueza que generamos debe fortalecer nuestra empresa, contribuir a la realización personal y al bienestar de nuestros trabajadores, retribuir al capital invertido y permitirnos participar en el desarrollo de la comunidad.

Visión Empresarial

"Ser competitivos a nivel mundial suministrando productos de valor agregado para la alimentación humana"

Es tener la capacidad de conquistar y sostener la preferencia de los clientes y además ser rentables, operando con los más altos estándares en calidad, procesos y servicios relevantes; de esta manera, la empresa crece y aumenta su participación en el mercado.

La competitividad se crea y se desarrolla a través del aprendizaje, mejoramiento e innovación, y con la participación de los clientes, proveedores, personal de la empresa, accionistas y la comunidad.

Competimos con los suministradores de alimentos no sólo del país sino del resto del mundo, operando como una empresa de clase mundial. Nos regimos por los más altos estándares, por los procesos de planeamiento estratégico y gerencia de recursos totalmente integrados, con un enfoque intensivo al

cliente. Nos sustentamos en la innovación tecnológica, mejora continua, una optima gestión del personal y del conocimiento; y nos mantenemos vigentes con las tendencias mundiales. De esta manera, obtenemos una alta productividad.

La cadena de suministro abarca desde la recepción del pedido, pasando por los procesos de producción animal e industrial y comercialización, hasta el servicio post venta atendiendo tanto los mercados internos como los externos. Son productos con transformación creados a través de investigación y desarrollo para satisfacer y exceder las expectativas de los clientes, mejorando su calidad de vida.

Nos dedicamos a satisfacer las necesidades de alimentación de las personas, acorde a los nuevos hábitos alimentarios y en todas las oportunidades de consumo masivo, en los hogares o fuera de él.

Situación actual

El grupo Ikeda, a través de Avícola San Fernando S.A., es el mayor productor y abastecedor de carne de pollo, pavo, cerdo, y embutidos además de huevos comerciales del mercado local en todo el Perú. Atiende al consumidor de manera directa a través de sus cinco tiendas propias, y de los supermercados E.Wong, Metro y Santa Isabel. También abastece de manera exclusiva a la cadena KFC y a Burger King en sus requerimientos de pollo en el Perú, así como a importantes

cadenas de pollos a la brasa como Mediterráneo Chicken y Pardo's Chicken. La empresa también exporta a Japón pechuga de pavo horneado, pavos congelados a Ecuador, pollitas BB de postura a México y reproductoras BB de carne a Colombia, Venezuela, Argentina y Bolivia.

Hoy Grupo San Fernando es la única empresa del sector agropecuario en el Perú que cuenta con una integración vertical total y que comercializa sus productos con marca propia, habiendo logrado la certificación ISO 9000 en la planta procesadora de productos cárnicos, en la planta de alimentos balanceados de Lurín, en la granja de abuelos de Huarmey, en la planta de incubación de Chorrillos y en la planta de beneficio de aves de Chincha.

Las ventas del grupo San Fernando actualmente alcanzan los 300 millones de dólares anuales y lidera en todos los mercados que participa. Ha transcurrido más de medio siglo de mejoras permanentes, la empresa sigue manteniendo su misión: "Contribuir al bienestar de la humanidad suministrando alimentos de consumo masivo en el mercado global".

La cultura empresarial de San Fernando se basa en los principios de la Calidad Total y Excelencia Operativa.

Bajo estos principios San Fernando ha logrado estandarizar sus procesos, lo que le ha permitido obtener 12 certificaciones internacionales ISO 9000 y se encuentra actualmente en proceso de obtener la certificación ISO 14000.

San Fernando ha logrado eficiencia a través de la integración vertical de la cadena productiva, desde el procesamiento de

alimentos balanceados, reproducción e incubación, crianza, beneficio y finalmente la comercialización y distribución de los productos a nivel nacional e internacional.

El éxito de San Fernando se basa fundamentalmente en el trabajo en equipo de su personal altamente calificado y la adquisición de nuevas y modernas tecnologías, todo esto ha logrado posicionar a San Fernando como la empresa líder en el Perú en todas las líneas que produce y comercializa: Pollo, Pavo, Huevo, Cerdo y Productos Procesados. Y este liderazgo, que basándose en esfuerzo conjunto produce para San Fernando ventas anuales alrededor de los $300'000,000 de dólares americanos y una marca que es sinónima de calidad y garantía para todo el mundo.

Productos

Durante más de 50 años, San Fernando ha contribuido con la sociedad, brindando alimentos de excelente calidad. A pesar de haber transcurrido ya medio siglo de mejoras permanentes, la empresa sigue manteniendo el mismo objetivo de no sólo satisfacer sino exceder las expectativas de los clientes.

Para ello ha desarrollado una organización eficaz e innovadora que, poniendo en operación los últimos avances tecnológicos y científicos, ha logrado convertirse en una empresa que ofrece productos altamente competitivos a nivel internacional.

Los productos San Fernando en todas sus líneas: Pollo, Pavo, Cerdo, Embutidos, Procesados, Precocidos y Huevos se caracterizan por su buen sabor y frescura.

El estar a la vanguardia y adelantarse a los gustos y preferencias de nuestros consumidores es un sello que San Fernando posee y seguirá manteniendo.

La relación detallada de productos es la siguiente:

- Pollos
  - Pollo trozado
  - Pollo carne
  - Pollo brasa
  - Pollo congelado
  - Pollo roaster

- Pavos
  - Pavita trozada
  - Pavo entero

- Huevos (rosados)

- Gallina
  - Gallina criolla
  - Gallina doble pechuga
  - Pechuga con ala
  - Pierna con encuentro

- Cerdo
  - Cortes de Lomo
  - Cortes de Pierna
  - Cortes de Brazuelo
  - Cortes de Panceta
  - Cortes Especiales

- Cerdos Enteros

- Embutidos y procesados
  - Jamones
  - Salchichas y Hot Dog
  - Especialidades
  - Pate, Chorizo y Ahumados
  - Linea Súper
  - Fiambres
  - Semielaborados
  - Precocidos

Cobertura de servicios

La marca San Fernando es reconocida en todo el Perú, y sus productos se comercializan en Lima y el resto de las provincias del país.

En el mercado internacional, San Fernando exporta productos genéticos: huevos fértiles, pollitos BB, pavitos BB y reproductoras a países como México, Argentina, Venezuela, Colombia, Ecuador, El Salvador y Bolivia; y pavo y cerdo congelado a Japón, Ecuador y Bolivia.

3.2. DESCRIPCION DE LOS SISTEMAS DE INFORMACION DEL GRUPO SAN FERNANDO S.A.

Alcance funcional

Los sistemas de información del Grupo San Fernando S.A. cubren las siguientes áreas funcionales del negocio:

- Inventario
- Logística
- Importaciones
- Facturación
- Cuentas por pagar
- Cuentas por cobrar
- Costos
- Recursos humanos
- Contabilidad de costos
- Tesorería: Caja y cuentas corrientes
- Comercial: Marketing
- Sistema de Información Gerencial (SIG)

Cada una de estas funciones está soportada por una aplicación, que representa un módulo del sistema integrado administrativo y financiero de planeamiento y manejo de recursos empresariales del negocio. En la figura 03-01 se muestra el Mapa Funcional de Sistemas del Grupo San Fernando.

Los módulos acceden y registran sus operaciones en una Base de Datos relacional centralizada.

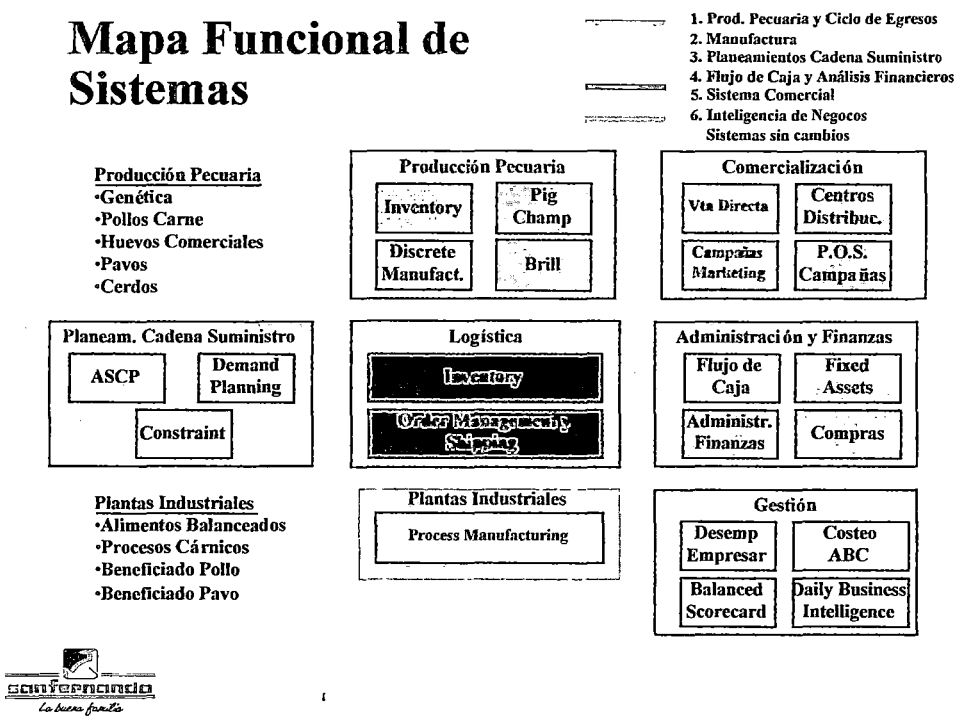


Figura 03-01. Mapa Funcional de Sistemas del Grupo San Fernando

Estas aplicaciones por la naturaleza de sus transacciones se categorizan en:

- Online Transaction Processing (OLTP), que tienen alto throughput (transacciones por unidad de tiempo) e intensivas en Insert/Update. El 90% de transacciones son de esta naturaleza. Se caracterizan por volúmenes crecientes de datos que entre 800 a 1000 usuarios acceden concurrentemente. Las metas a afinar en estas aplicaciones se relacionan con la disponibilidad (24x7), trhoughput, concurrencia y disponibilidad.

  La figura 03-02 muestra la interacción de Aplicaciones OLTP con la Base de Datos

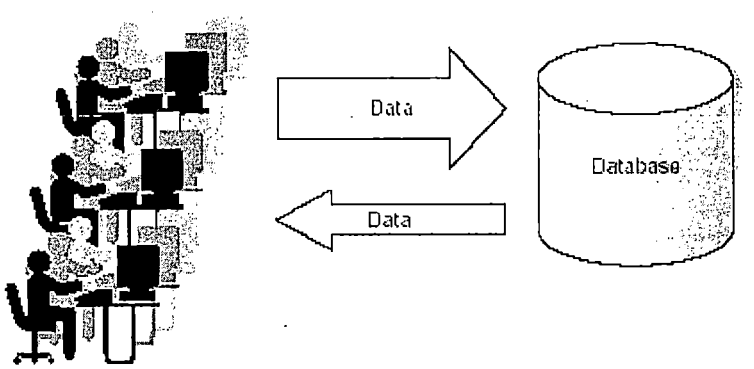


Figura 03-02. Interacción de aplicaciones OLTP con la Base de Datos

- Decision Support System (DSS), que típicamente convierten grandes cantidades de información en reportes definidos por el usuario. Este tipo de aplicaciones ejecutan consultas sobre una gran cantidad de información registrada por las aplicaciones OLTP. Los que toman decisiones en el negocio, usuarios del Sistema de Información Gerencial (SIG), usan estas aplicaciones para determinar qué estrategias la organización debe tomar.

  La figura 03-03 muestra la interacción de las aplicaciones DSS con la Base de Datos:

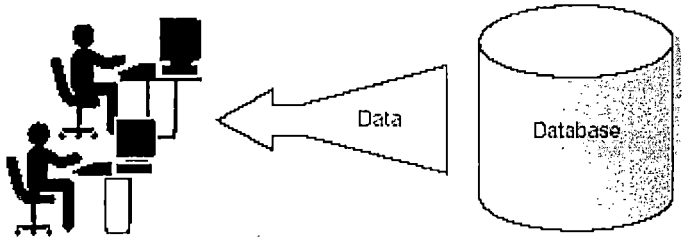


Figura 03-03. Interacción de aplicaciones DSS con la Base de Datos

- **Aplicaciones Multipropósito**

Algunas aplicaciones se basan sobre características OLTP y DSS. Una de ellas es la aplicación de marketing que determina patrones de consumo de clientes basada

en las compras de éstos en un horizonte de tiempo. La aplicación sumariza las ventas diarias registradas por Facturación y el staff de Marketing consulta estos datos para determinar los productos a vender por cliente y región. Este reporte es luego usado para determinar niveles óptimos de inventario para productos particulares en cada Centro de Distribución (CD).

La figura 03-04 ilustra múltiples configuraciones y aplicaciones accesando al RDBMS.

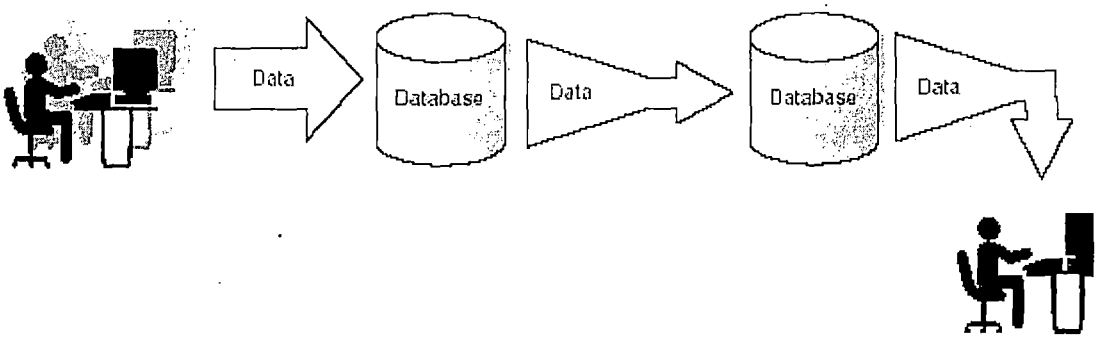


Figura 03-04. Múltiples configuraciones y aplicaciones accesando al RDBMS

Arquitectura de aplicaciones

Las aplicaciones en Grupo San Fernando S.A. se basan en la mezcla de dos arquitecturas:

- Sistemas distribuidos
- Configuraciones cliente/servidor de dos o más niveles

Sistemas distribuidos

Las aplicaciones distribuidas manejan la información sobre múltiples bases de datos sobre múltiples máquinas. El objetivo a cubrir es brindar rendimiento, site autonomy (independencia física) y reducción de costos. Varias máquinas pequeñas pueden ser menos caras y más flexibles que un gran servidor central. Las configuraciones distribuidas toman ventaja de máquinas pequeñas y poderosas con opciones de conectividad más baratas. Estos sistemas permiten también almacenar los datos en varios sitios, y cada sitio puede acceder transparentemente los datos.

La figura 03-05 muestra la configuración de un sistema distribuido:

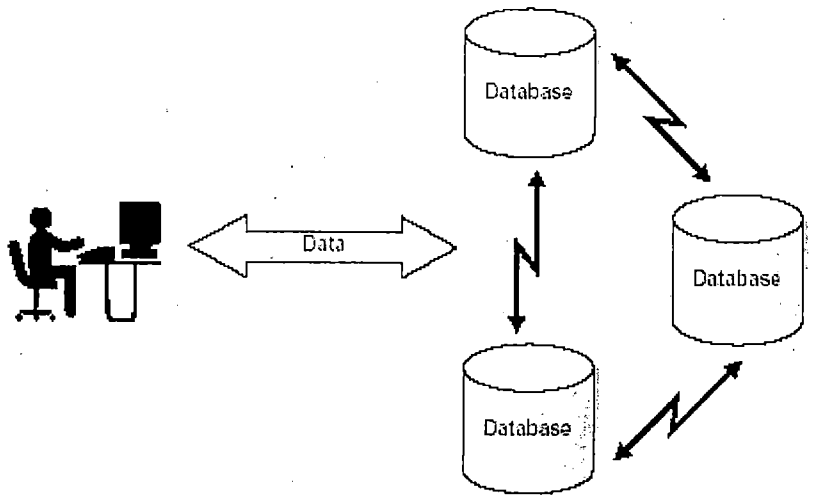


Figura 03-05. Sistema Distribuido en arquitectura cliente-servidor

Cliente/Servidor de dos niveles

La arquitectura cliente/servidor distribuye el trabajo del sistema entre la máquina cliente (aplicación) y el servidor (servidor de Base de Datos). Típicamente, las máquinas cliente son estaciones de trabajo que ejecutan una aplicación con interface GUI (Developer 6i) y están conectadas vía la red informática al servidor de Base de Datos (Oracle8i Release 3 Enterprise Edition).

3.3. INFRAESTRUCTURA IT

3.3.1. Cluster de Servidores de Base de Datos principal

La plataforma core es IBM RS/6000 HACMP cluster compuesta por dos nodos en configuración activo/pasivo. En la figura 03-06 se muestra el esquema topológico de esta arquitectura. El nodo principal activo es "Node 1" y el nodo secundario pasivo es "Node 2".

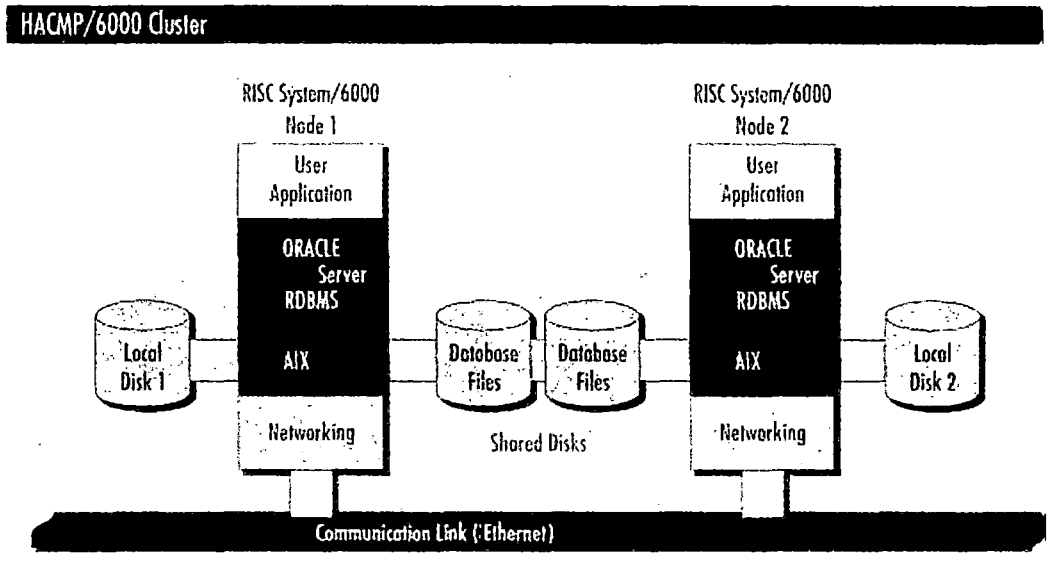
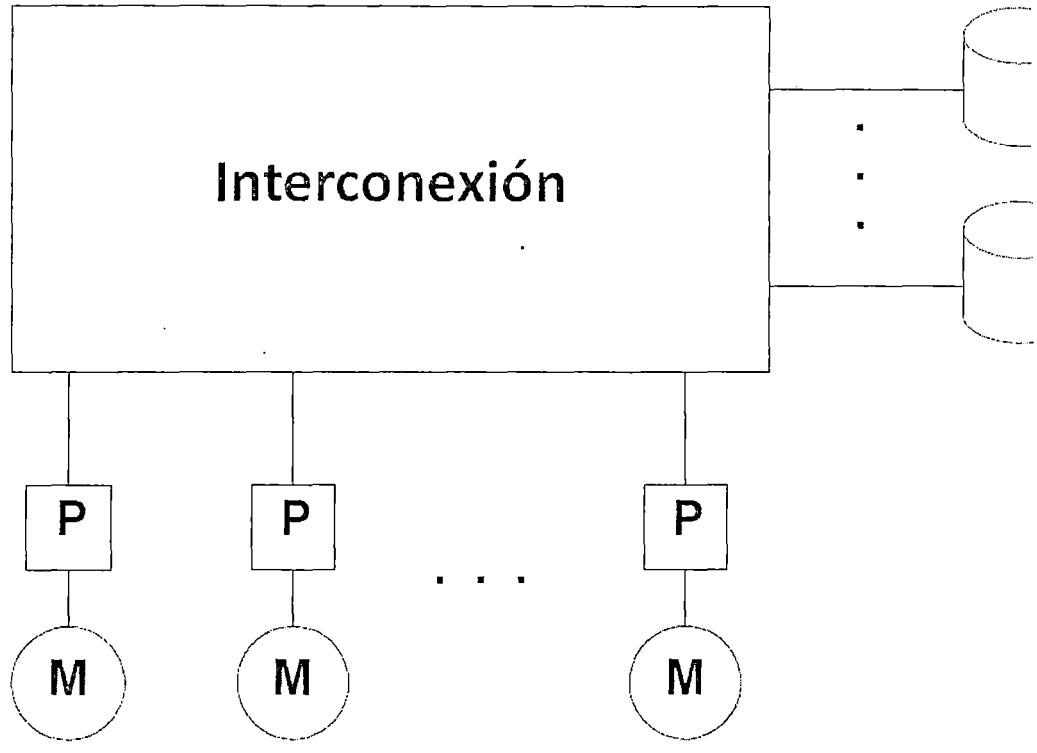


Figura 03-06. Cluster IBM RS/6000 de dos (2) nodos en Grupo San Fernando

Es una arquitectura de disco compartido, la cual esquemáticamente se simplifica en la figura 03-07.



P: **Procesador**

M: **Memoria**

Figura 03-07. Arquitectura de disco compartido

Las características de **hardware** en cada nodo se detallan en el Anexo 1, que se encuentra en la sección ANEXOS de la tesis.

El **software** instalado en cada nodo es el siguiente:

- IBM AIX 5L (Operating System)
- HACMP (High Availability Cluster Multi Processor)
- Oracle 8i Release 3 Enterprise Edition (RDBMS)
- Net8 (Middleware Oracle)

3.3.2. Estaciones cliente

Las estaciones cliente son IBM PC compatibles. La configuración promedio es:

- 1 Procesador Intel Pentium III o superior
- 256 MB de RAM
- 20 GB de disco duro
- 1 Tarjeta de red Ethernet 10/100 Mbps

Software

- Windows 95/NT/2000
- Oracle Forms/Reports runtime
- Net8 client

3.3.3. Networking

La empresa cuenta con una sede central en Lima-Surquillo y más de setecientas estaciones repartidas por todo el territorio nacional. La comunicación permanente entre sus oficinas es de vital importancia para la organización y se concreta en una red WAN.

Para la sede central, la empresa cuenta con una infraestructura de red LAN de alta velocidad con sistema de cableado estructurado, armario de comunicaciones y sala de sistemas.

Los puntos remotos son contactados vía la red WAN.

3.4. RETOS Y NECESIDADES DEL NEGOCIO

El Grupo San Fernando S.A. necesita que sus aplicaciones cuenten con una infraestructura de sistemas que les brinde buen rendimiento, altos niveles de disponibilidad, que la plataforma donde reside su solución informática sea escalable, y que existan adecuados niveles de seguridad.

3.4.1. Rendimiento

Se necesita mejorar el rendimiento de las aplicaciones OLTP y DSS. En épocas de campaña comercial (Fiestas Patrias y Navidad) se llega a procesar 18,000 transacciones por hora.

Para las aplicaciones OLTP esto se logra aumentando el throughput (número de transacciones por unidad tiempo). Estas aplicaciones procesan en horas pico 18,000 transacciones por hora.

Para las aplicaciones DSS esto se logra disminuyendo el tiempo de respuesta. En este tipo de aplicaciones no se tiene un patrón de uso definido. Por momentos una consulta puede recuperar sólo un puñado de registros, y en otros se puede procesar un query paralelo masivo que ordene miles de registros de diferentes tablas.

3.4.2. Disponibilidad

El negocio trabaja las 24 horas del día los 7 días de la semana. Necesita que la plataforma tecnológica: hardware y software ofrezca niveles de disponibilidad cercanos al 100%.

3.4.3. Escalabilidad

Para soportar una mayor carga de usuarios, así como permitir mayor capacidad de cómputo a los mismos es que se necesita que la plataforma tecnológica (hardware y software) sea escalable, esto es, permita mayores niveles de consumo aumentando las capacidades físicas de los recursos.

3.4.4. Seguridad

La plataforma debe ser segura, principalmente en el tema de integridad y respaldo de información.

3.5. ENFOQUE METODOLOGICO

Una metodología bien planeada es la clave del éxito en afinamiento de rendimiento (performance tuning) de sistemas. Las diferentes estrategias de afinamiento varían en su efectividad, y los sistemas con diferentes propósitos, tales como sistemas OLTP y DSS requieren diferentes métodos de afinamiento.

3.5.1. Oportunidad de afinamiento: Enfoque Costo/Beneficio

Para mejores resultados, es mejor realizar el afinamiento durante la fase de diseño en vez de esperar hasta la etapa de implementación. Para esta tesis, el afinamiento se ha realizado estando el sistema en producción (reactive tuning). Lamentablemente, la mayoría de esfuerzos de afinamiento en las empresas de toda envergadura son del tipo "mantenimiento correctivo", lo que es consecuencia de omisiones gruesas en las fases de planeamiento.

Como se establece en el marco teórico, el enfoque más efectivo para afinar es el proactivo (proactive tuning), previo al despliegue del sistema en producción y que se fundamenta en la metodología de la ingeniería de rendimiento. Los analistas del negocio y los diseñadores de las aplicaciones deben trabajar para establecer las metas de rendimiento y establecer expectativas reales. En esta fase se determina la combinación de recursos del sistema y características de software que mejor cumpla las necesidades.

Diseñar un sistema que ejecute bien minimiza su implementación y sus costos de mantenimiento administrativo. La figura 03-08 muestra el costo relativo de afinar durante cada fase del ciclo de vida de una aplicación. Los costos de afinamiento son menores durante la fase de

diseño de un sistema, pero van aumentando conforme se avanza a la fase de Desarrollo, y son mucho mayores si el sistema ya sirve a la organización en su fase de Producción.

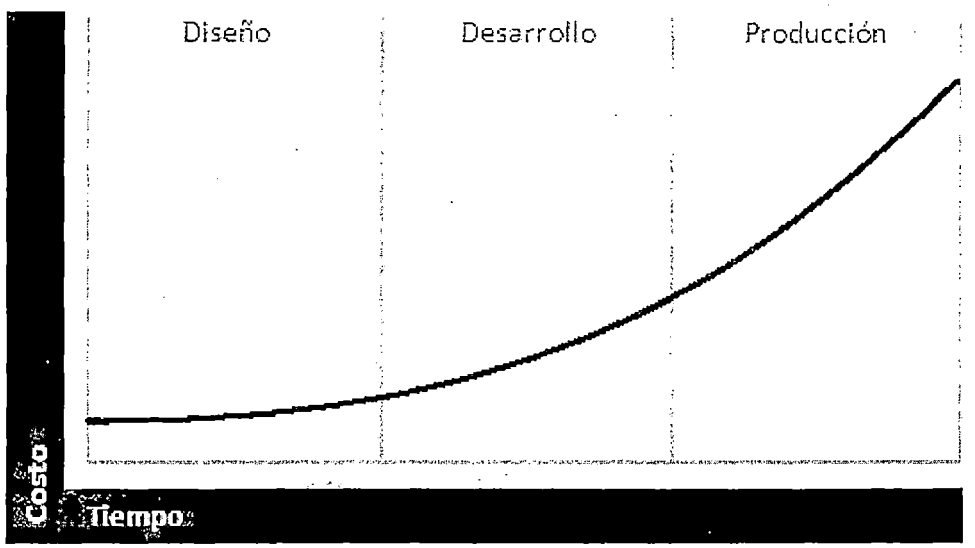


Figura 03-08. Costo relativo de afinamiento durante la vida de una aplicación

Para complementar esta vista, la figura 03-09 muestra que el beneficio relativo de afinar una aplicación sobre el curso de su vida es inversamente proporcional al costo incurrido en afinarlo.

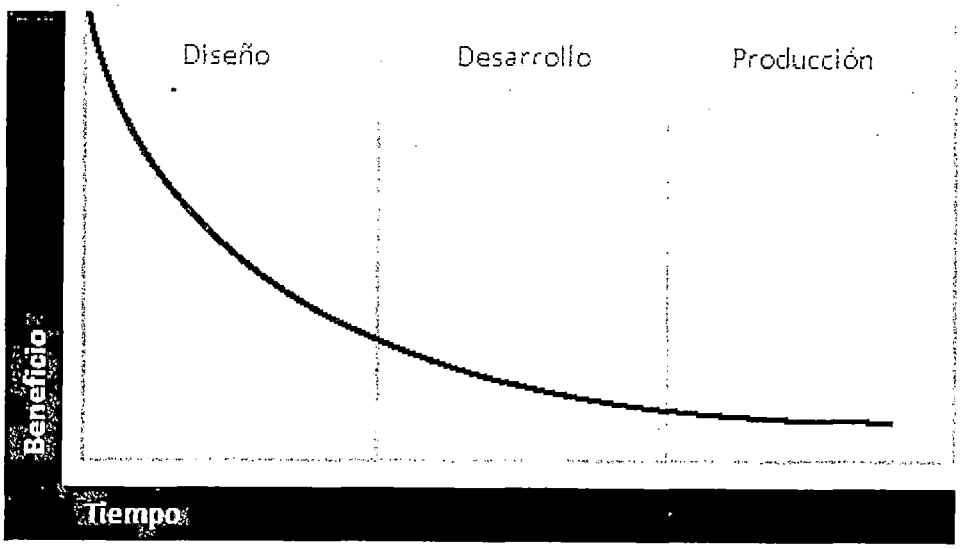


Figura 03-09. Beneficio relativo al afinar una aplicación durante la vida de una aplicación

El momento más efectivo para afinar el rendimiento de un sistema es durante las fases iniciales de su construcción, la que se simplifica en las figuras 03-08 y 03-09 como fase de Diseño. Es en las fases iniciales donde se consiguen los mejores beneficios al más bajo costo.

3.5.2. Metodología usada en Grupo San Fernando S.A.

El enfoque metodológico usado en el Grupo San Fernando S.A. es el afinamiento reactivo (reactive tuning). Para usar este enfoque, se empieza abajo hacia arriba (bottom-up), encontrando y arreglando cuellos de botella. La meta principal es que el motor de Base de Datos tuviera mejores niveles de rendimiento con los recursos actuales. En paralelo a este objetivo, para conseguir ganancias adicionales se puede necesitar afinar la aplicación o añadir recursos. El examen de

los cuellos de botella y la inclusión de características u opciones del motor de Base de Datos para eliminarlos es clave en la consecución de nuestro objetivo.

Entiéndase que el afinamiento del rendimiento (performance tuning) es una actividad contínua, y se convierte en una parte importante del mantenimiento de los sistemas.

3.5.3. Metodología de afinamiento en Base de Datos ORACLE

Los siguientes pasos muestran el método recomendado para afinar una aplicación cliente/servidor basada en una Base de Datos Oracle.

Estos pasos son priorizados en orden decreciente de beneficio: Los pasos con mayor impacto en el rendimiento aparecen primero. Para resultados óptimos, por lo tanto, se recomienda seguir los pasos en el orden mostrado, desde las fases de diseño y desarrollo hasta las fases de afinamiento de instancia.

Paso 1: Afine las reglas del negocio

Paso 2: Afine el diseño de datos

Paso 3: Afine el diseño de la aplicación

Paso 4: Afine la estructura lógica de la Base de Datos

Paso 5: Afine las operaciones de Base de Datos

Paso 6: Afine las rutas de acceso

Paso 7: Afine la alocación de memoria

Paso 8: Afine el I/O y la estructura física

Paso 9: Afine la contención de recursos

Paso 10: Afine la plataforma (sistema operativo-hardware)

Después de completar estos pasos, re-evalúe el rendimiento de la Base de Datos después de completar estos pasos y decida si un afinamiento adicional es necesario. El afinamiento es un proceso iterativo.

La figura 03-10 muestra en un diagrama de flujo los pasos de la metodología:

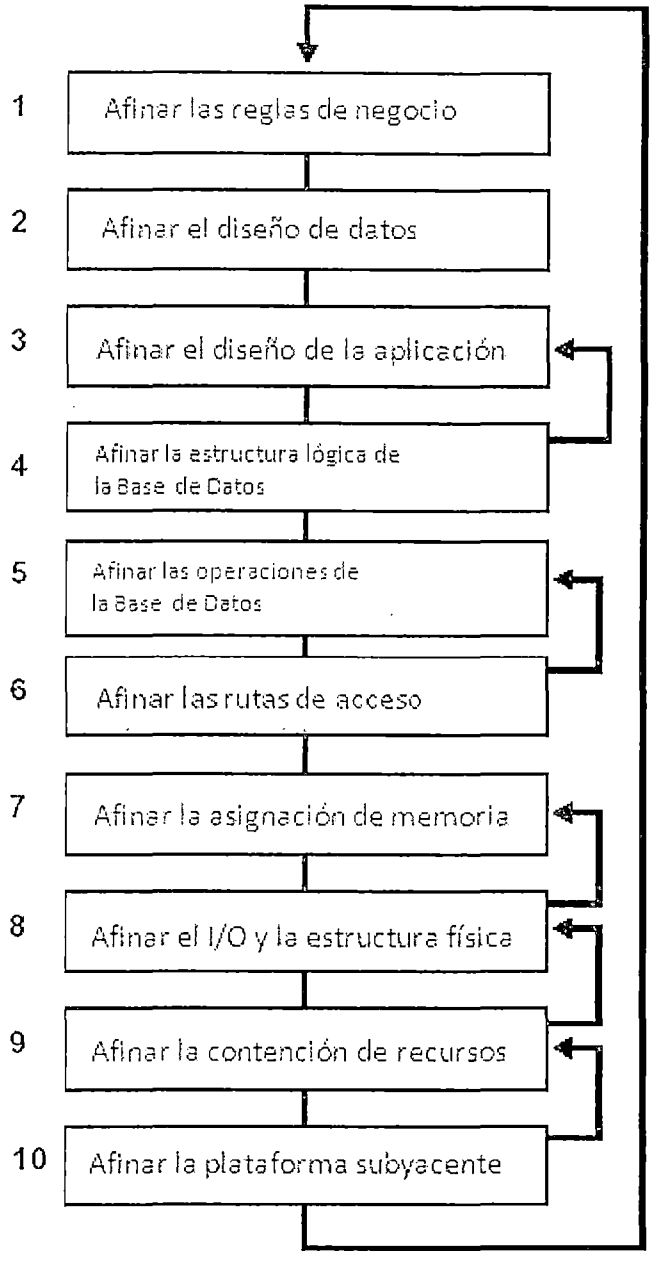


Figura 03-10. El Método de Afinamiento

3.5.4. Aplicación de la metodología en Grupo San Fernando S.A.

Debido a que nuestro enfoque es aplicar afinamiento reactivo (reactive tuning), se aplicará la siguiente metodología de afinamiento:

- Establecer las metas del afinamiento
- Analizar el cuello de botella
- Evaluar hipótesis
- Aplicar los cambios
- Evaluar las mejoras aplicadas
- Parar cuando el objetivo sea cumplido

3.6. Proceso de Afinamiento de rendimiento

3.6.1. Metas del afinamiento

Las metas del afinamiento serán:

Meta 1

"Reducir en 50% los tiempos de respuesta del sistema de costos."

Meta 2

"Eliminar por lo menos al 50% los principales cuellos de botella encontrados en la operación de los sistemas en horas pico durante la época de campaña comercial de 2003."

3.6.2. Afinamiento para Meta 1: Reducción de Tiempos de respuesta en sistema de costos industriales

3.6.2.1. Análisis de cuellos de botella

Los tiempos de respuesta obtenidos en los procesos de costeo mensual, de naturaleza principalmente batch, eran excesivos e impedían la fluidez de otras aplicaciones por estar la Base de Datos centralizada.

Se determina que hay accesos SQL con costo muy alto a tablas de volumen considerable.

3.6.2.2. Aplicación de Metodología

En afinamiento de aplicaciones lo principal es mejorar los accesos SQL que se realizan al motor de Base de Datos, y si se diera el caso, mejorar la organización de los objetos referenciados. Esto se sustenta en la metodología expuesta en la sección 3.5.3.

Aplicando los pasos 4 y 5 presentados en (3.5.3) que son:

Paso 4: Afine la estructura lógica de la Base de Datos

Paso 5: Afine las operaciones de Base de Datos

Nuestro enfoque será de dos pasadas, revisar los accesos SQL ya existentes, afinarlos y descubrir si se puede aprovechar características nuevas de Oracle8i para manejo de estructuras lógicas: Partitioning Option, Locally Managed tablespaces, compressed indexes.

Del paso 4, se verificó que los accesos SQL hacían principalmente full scan table e index full scan a las tablas ARCOHA y ARCOHA_OPL. Estas tablas conforman una relación padre-hijo, donde ARCOHA es padre y ARCOHA_OPL es hijo.

La descripción de estas tablas es la siguiente:

```
CREATE TABLE SF_COSTOS.ARCOHA
(
  NO_CIA          VARCHAR2(2)                    NOT NULL,
  ANO             NUMBER(4)                      NOT NULL,
  MES             NUMBER(2)                      NOT NULL,
  IND_ING_GAS     VARCHAR2(1)                    NOT NULL,
  N1              VARCHAR2(3)                    NOT NULL,
  N2              VARCHAR2(3)                    NOT NULL,
  N3              VARCHAR2(3)                    NOT NULL,
  N4              VARCHAR2(3)                    NOT NULL,
  N5              VARCHAR2(3)                    NOT NULL,
  ANALITICOS      VARCHAR2(400)                  NOT NULL,
  MOVIMIENTO_N    NUMBER(15,2)                       NULL,
  MOV_DB_N        NUMBER(15,2)                       NULL,
  MOV_CR_N        NUMBER(15,2)                       NULL,
  SALDO_N         NUMBER(15,2)                       NULL,
  MOVIMIENTO_D    NUMBER(15,2)                       NULL,
  MOV_DB_D        NUMBER(15,2)                       NULL,
  MOV_CR_D        NUMBER(15,2)                       NULL,
  SALDO_D         NUMBER(15,2)                       NULL,
  MOVIMIENTO_ND   NUMBER(15,2)                       NULL,
  MOV_DB_ND       NUMBER(15,2)                       NULL,
  MOV_CR_ND       NUMBER(15,2)                       NULL,
  SALDO_ND        NUMBER(15,2)                       NULL,
  MOVIMIENTO_DN   NUMBER(15,2)                       NULL,
  MOV_DB_DN       NUMBER(15,2)                       NULL,
  MOV_CR_DN       NUMBER(15,2)                       NULL,
  SALDO_DN        NUMBER(15,2)                       NULL,
  NO_ORDEN        VARCHAR2(11)                       NULL
)
TABLESPACE COSTOS_TAB_ARCOHA


    PRIMARY KEY
    NO_CIA, ANO, MES, IND_ING_GAS, N1, N2, N3, N4, N5,
ANALITICOS
```

INDICES

INDEX_NAME	COLUMN_NAME
IX_ARCOHA_01	NO_CIA
IX_ARCOHA_01	ANO
IX_ARCOHA_01	MES
IX_ARCOHA_01	N1
IX_ARCOHA_01	N2
IX_ARCOHA_01	N3
IX_ARCOHA_01	N4
IX_ARCOHA_01	N5
IX_ARCOHA_01	ANALITICOS
IX_ARCOHA_02	NO_CIA
IX_ARCOHA_02	ANO
IX_ARCOHA_02	MES
IX_ARCOHA_02	ANALITICOS
PK_ARCOHA_01	NO_CIA
PK_ARCOHA_01	ANO
PK_ARCOHA_01	MES
PK_ARCOHA_01	IND_ING_GAS
PK_ARCOHA_01	N1
PK_ARCOHA_01	N2
PK_ARCOHA_01	N3
PK_ARCOHA_01	N4
PK_ARCOHA_01	N5
PK_ARCOHA_01	ANALITICOS

```
CREATE TABLE SF_COSTOS.ARCOHA_OPL
(
  NO_CIA          VARCHAR2(2)                NOT NULL,
  ANO             NUMBER(4)                  NOT NULL,
  MES             NUMBER(2)                  NOT NULL,
  IND_ING_GAS     VARCHAR2(1)                    NULL,
  N1              VARCHAR2(3)                    NULL,
  N2              VARCHAR2(3)                    NULL,
  N3              VARCHAR2(3)                    NULL,
  N4              VARCHAR2(3)                    NULL,
  N5              VARCHAR2(3)                    NULL,
  ANALITICOS      VARCHAR2(400)                  NULL,
  NO_ORDEN        VARCHAR2(11)               NOT NULL,
  MOVIMIENTO_N    NUMBER(15,2)                   NULL,
  MOV_DB_N        NUMBER(15,2)                   NULL,
  MOV_CR_N        NUMBER(15,2)                   NULL,
  SALDO_N         NUMBER(15,2)                   NULL,
  MOVIMIENTO_D    NUMBER(15,2)                   NULL,
  MOV_DB_D        NUMBER(15,2)                   NULL,
  MOV_CR_D        NUMBER(15,2)                   NULL,
  SALDO_D         NUMBER(15,2)                   NULL,
  MOVIMIENTO_ND   NUMBER(15,2)                   NULL,
  MOV_DB_ND       NUMBER(15,2)                   NULL,
  MOV_CR_ND       NUMBER(15,2)                   NULL,
  SALDO_ND        NUMBER(15,2)                   NULL,
  MOVIMIENTO_DN   NUMBER(15,2)                   NULL,
  MOV_DB_DN       NUMBER(15,2)                   NULL,
  MOV_CR_DN       NUMBER(15,2)                   NULL,
```

```
  SALDO_DN          NUMBER(15,2)                    NULL,
  TIPO_ARTI         VARCHAR2(2)                     NULL,
  CLASE             VARCHAR2(3)                     NULL,
  CATEGORIA         VARCHAR2(3)                     NULL,
  FAMILIA           VARCHAR2(3)                     NULL,
  NO_ARTI           VARCHAR2(9)                     NULL,
  UNIDADES          NUMBER(12,3)                    NULL,
  CONTENIDO         NUMBER(12,3)                    NULL,
  SEC_INDIRECTOS    NUMBER(8)                       NULL,
  SEC_DIRECTOS      NUMBER(2)                       NULL,
  ANO_MES           NUMBER(6)                       NULL
)
TABLESPACE COSTOS_TAB_ARCOHA_OPL
```

FOREIGN KEY

ARCOHA
(NO_CIA, ANO, MES, IND_ING_GAS, N1, N2, N3, N4, N5, ANALÍTICOS)

INDICES

INDEX_NAME	COLUMN_NAME
IX_ARCOHA_OPL_01	NO_CIA
IX_ARCOHA_OPL_01	ANO
IX_ARCOHA_OPL_01	MES
IX_ARCOHA_OPL_01	IND_ING_GAS
IX_ARCOHA_OPL_01	N1
IX_ARCOHA_OPL_01	N2
IX_ARCOHA_OPL_01	N3
IX_ARCOHA_OPL_01	N4
IX_ARCOHA_OPL_01	N5
IX_ARCOHA_OPL_01	ANALITICOS
IX_ARCOHA_OPL_01	NO_ORDEN
IX_ARCOHA_OPL_02	NO_CIA
IX_ARCOHA_OPL_02	ANO
IX_ARCOHA_OPL_02	MES
IX_ARCOHA_OPL_02	NO_ORDEN
IX_ARCOHA_OPL_02	TIPO_ARTI
IX_ARCOHA_OPL_02	CLASE
IX_ARCOHA_OPL_02	CATEGORIA
IX_ARCOHA_OPL_02	FAMILIA
IX_ARCOHA_OPL_02	NO_ARTI
IX_ARCOHA_OPL_03	NO_CIA
IX_ARCOHA_OPL_03	ANO
IX_ARCOHA_OPL_03	MES
IX_ARCOHA_OPL_03	SEC_INDIRECTOS
IX_ARCOHA_OPL_04	NO_CIA
IX_ARCOHA_OPL_04	ANO
IX_ARCOHA_OPL_04	MES
IX_ARCOHA_OPL_04	SEC_DIRECTOS
IX_ARCOHA_OPL_05	NO_CIA

```
IX_ARCOHA_OPL_05                NO_ORDEN
IX_ARCOHA_OPL_05                ANO_MES
IX_ARCOHA_OPL_05                ANALITICOS
```

Un primer enfoque a aplicar es el particionamiento de la tabla por mes, algo factible de hacer usando Partitioning Option de Oracle8i. Asimismo, los índices de estas tablas debieran estar particionados localmente con la misma clave de particionamiento de la tabla para mejorar el rendimiento de los index full scans.

Asimismo, los índices que no participan en claves primarias no son selectivos. Por ello se recrearon comprimidos con la opción COMPRESS de la sentencia CREATE INDEX, disponible en Oracle8i lo que nos permitirá ahorrar espacio en Base de Datos y mejorar el rendimiento de los index full scans.

Del análisis de sentencias SQL, la siguiente sentencia demoraba 1 hora y 2 minutos y era la más utilizada por los procesos de costeo:

```
SELECT DISTINCT
TRUNC(mn.fecha) fecha,
      mn.no_arti
  FROM arintd td,
      arinmn1 mn,
      (SELECT a.no_cia COL1,
            a.no_arti COL2
        FROM arinmn1 a,
            arprop o,
            arintd d
       WHERE a.no_cia = o.no_cia
         AND a.no_orden = o.no_orden
         AND 'GP' = o.tipo
         AND a.no_cia = d.no_cia
         AND a.tipo_doc = d.tipo_m
         AND 'P' = d.clase_transc
         AND 'E' = d.movimi
         AND d.trans_ref IS NULL
         AND o.no_cia = d.no_cia
       GROUP BY a.no_cia,
             a.no_arti) TEMP0
 WHERE '01' = td.no_cia
   AND 'S' = NVL(td.afect_cost, 'N')
   AND td.no_cia = mn.no_cia
   AND 2003 = mn.ano
   AND 8 = mn.mes
   AND td.tipo_m = mn.tipo_doc
   AND mn.no_cia = '01'
   AND td.no_cia = TEMP0.COL1
   AND mn.no_arti = TEMP0.COL2
;
```

Se hace necesario optimizarla buscando una sentencia SQL alternativa que ofrezca los mismos resultados.

3.6.2.3. Aplicación de cambios

i) Se particionaron horizontalmente por mes las tablas ARCOHA y ARCOHA_OPL. El particionamiento es una técnica que permite dividir una tabla o un índice en segmentos más pequeños y manejables de acuerdo a una clave de particionamiento. La técnica más común de particionamiento es por rango, e indica que una partición contendrá los valores definidos dentro de un rango específico.

Los principales motivantes para el particionamiento de tablas son:

- Acelerar el rendimiento.
- Incrementar la disponibilidad.
- Mejorar la administración.
- Habilita la Administración del Ciclo de Vida de la Información.

Se muestra el detalle de particionamiento por acceso a la información registrada en el diccionario de datos:

```
SQL> SELECT TABLE_NAME, PARTITION_NAME, HIGH_VALUE
     FROM DBA_TAB_PARTITIONS WHERE TABLE_NAME = 'ARCOHA';

TABLE_NAME           PARTITION_NAME               HIGH_VALUE
-------------------- ---------------------------- --------------------
ARCOHA               ARCOHA_P200209               '01', 2002, 10
ARCOHA               ARCOHA_P200210               '01', 2002, 11
ARCOHA               ARCOHA_P200211               '01', 2002, 12
ARCOHA               ARCOHA_P200212               '01', 2003, 01
ARCOHA               ARCOHA_P200301               '01', 2003, 02
ARCOHA               ARCOHA_P200302               '01', 2003, 03
ARCOHA               ARCOHA_P200303               '01', 2003, 04
ARCOHA               ARCOHA_P200304               '01', 2003, 05
ARCOHA               ARCOHA_P200305               '01', 2003, 06
ARCOHA               ARCOHA_P200306               '01', 2003, 07
ARCOHA               ARCOHA_P200307               '01', 2003, 08
ARCOHA               ARCOHA_P200308               '01', 2003, 09
ARCOHA               ARCOHA_P200309               '01', 2003, 10
ARCOHA               ARCOHA_P200310               '01', 2003, 11
ARCOHA               ARCOHA_P200311               '01', 2003, 12
```

```
SQL> SELECT TABLE_NAME, PARTITION_NAME, HIGH_VALUE
       FROM DBA_TAB_PARTITIONS
      WHERE TABLE_NAME = 'ARCOHA_OPL';

TABLE_NAME           PARTITION_NAME               HIGH_VALUE
-------------------- ---------------------------- --------------------
ARCOHA_OPL           ARCOHA_OPL_P200209           '01', 2002, 10
ARCOHA_OPL           ARCOHA_OPL_P200210           '01', 2002, 11
ARCOHA_OPL           ARCOHA_OPL_P200211           '01', 2002, 12
ARCOHA_OPL           ARCOHA_OPL_P200212           '01', 2003, 01
ARCOHA_OPL           ARCOHA_OPL_P200301           '01', 2003, 02
ARCOHA_OPL           ARCOHA_OPL_P200302           '01', 2003, 03
ARCOHA_OPL           ARCOHA_OPL_P200303           '01', 2003, 04
ARCOHA_OPL           ARCOHA_OPL_P200304           '01', 2003, 05
ARCOHA_OPL           ARCOHA_OPL_P200305           '01', 2003, 06
ARCOHA_OPL           ARCOHA_OPL_P200306           '01', 2003, 07
ARCOHA_OPL           ARCOHA_OPL_P200307           '01', 2003, 08
ARCOHA_OPL           ARCOHA_OPL_P200308           '01', 2003, 09
ARCOHA_OPL           ARCOHA_OPL_P200309           '01', 2003, 10
ARCOHA_OPL           ARCOHA_OPL_P200310           '01', 2003, 11
ARCOHA_OPL           ARCOHA_OPL_P200311           '01', 2003, 12
```

La tabla 03-01 muestra la nomenclatura usada para el nombramiento de las particiones.

Tabla	Comando
ARCOHA	alter table sf_costos.ARCOHA add PARTITION ARCOHA_P200311 VALUES LESS THAN (01, 2003, 12);
ARCOHA_OPL	alter table SF_COSTOS.ARCOHA_OPL add PARTITION ARCOHA_OPL_P2003[illegible] VALUES LESS THAN (01, 2003, 12);

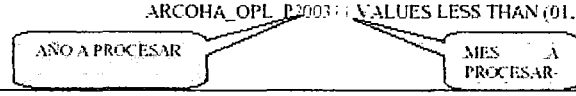


Tabla 03-01: Nomenclatura usada en las particiones de las tablas ARCOHA y ARCOHA_OPL

ii) Se recrearon los índices no participantes en primary keys como comprimidos. Se muestran en las siguientes consultas con el campo COMPRESSION con el valor de "ENABLED":

```
select index_name, uniqueness, compression
  from dba_indexes
 where table_name = 'ARCOHA'

INDEX_NAME                     UNIQUENES COMPRESSION
------------------------------ --------- -----------
PK_ARCOHA_01                   UNIQUE    DISABLED
IX_ARCOHA_01                   NONUNIQUE ENABLED
IX_ARCOHA_02                   NONUNIQUE ENABLED


select index_name, uniqueness, compression
  from dba_indexes
 where table_name = 'ARCOHA_OPL'

INDEX_NAME                     UNIQUENES COMPRESSION
------------------------------ --------- -----------
IX_ARCOHA_OPL_01               NONUNIQUE ENABLED
IX_ARCOHA_OPL_02               NONUNIQUE ENABLED
IX_ARCOHA_OPL_03               NONUNIQUE ENABLED
IX_ARCOHA_OPL_04               NONUNIQUE ENABLED
IX_ARCOHA_OPL_05               NONUNIQUE ENABLED
```

iii) Se particionaron todos los índices de las tablas ARCOHA y ARCOHA_OPL por mes (clave NO_CIA, ANO, MES). El detalle de los índices particionados se puede apreciar en el Anexo 2, que se encuentra en la sección ANEXOS de la tesis.

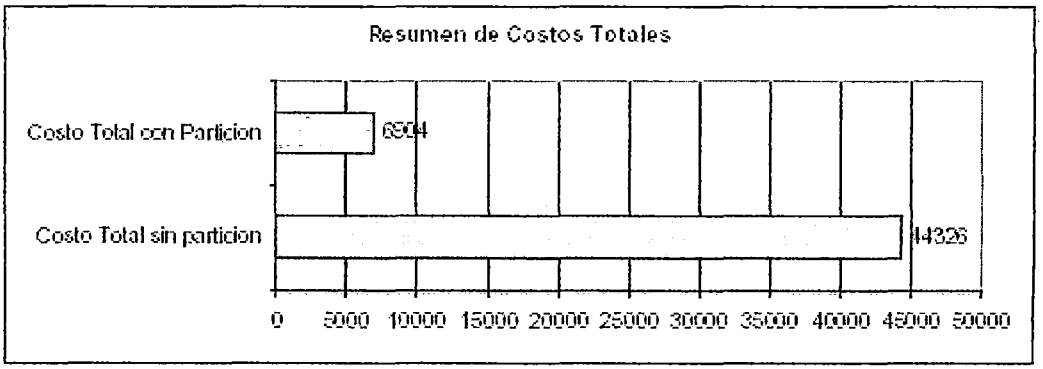
iv) Para la sentencia SQL ineficiente observada, se evaluaron diversas sintaxis alternativas, quedando finalmente el siguiente reemplazo:

```
SELECT        /*+ index (mn x5_arinmn1)  */
     DISTINCT TRUNC (mn.fecha) fecha,
mn.no_arti
         FROM arintd td, arinmn1 mn
        WHERE td.no_cia = :b1
          AND NVL (td.afect_cost, 'N') = 'S'
          AND mn.no_cia = td.no_cia
          AND mn.ano = :b2
          AND mn.mes = :b3
          AND mn.tipo_doc = td.tipo_m
          AND EXISTS ( SELECT 1
                         FROM arinmn1 a, arprop
o, arintd d
                        WHERE a.no_cia =
mn.no_cia
                          AND a.no_arti =
mn.no_arti
                          AND o.no_cia = a.no_cia
                          AND o.no_orden =
a.no_orden
                          AND o.tipo = :b4
                          AND d.no_cia = a.no_cia
                          AND d.tipo_m =
a.tipo_doc
                          AND d.clase_transc = 'P'
                          AND d.movimi = 'E'
                          AND d.trans_ref IS NULL)
```

3.6.2.4. Evaluación de mejoras aplicadas

i) Luego de la realización del particionamiento horizontal de las tablas más grandes del sistema de costeo, que son ARCOHA y ARCOHA_OPL, por los campos NO_CIA, ANO y MES se realizó un estudio de los costos de todas las sentencias SQL que accesaban a estas tablas con y sin particionamiento aplicado.

Los resultados son se muestran en la tabla 03-02:



Costo Total sin particion	44326
Costo Total con Particion	6904

Tabla 03-02. Comparación de Costos totales de sentencias SQL a tablas ARCOHA y ARCOHA_OPL antes y después del particionamiento.

Esto demuestra que aplicar particionamiento a estas tablas vía Partitioning Option resultó beneficioso en un 85%, ya

que en ese porcentaje se bajaron los costos de las sentencias SQL del proceso de costeo.

ii) La optimización de las sentencias SQL individuales también mejoraron el rendimiento. La más saltante es la que se presentó como muestra en el punto anterior, y que era cuello de botella porque la ejecutaban la mayoría de los procesos de costeo. De 3,720 segundos se bajó a 12 segundos. Una ganancia porcentual de 99.7%.

iii) El cuadro resumen del proceso de costeo mostrando los tiempos anteriores y posteriores a los cambios efectuados se muestra a continuación. En todas las actividades referidas a interacción directa con la aplicación informática, la ganancia promedio mínima es mayor a 50%.

Actividades	t de ahora	t antes
Proceso de distribución de gastos indirectos a actividades primarias - Segunda secuencia		
Copia de plantillas.	0h:1m	0h:1m
Generación de plantillas.	0h:20m	0h:20m
Ejecución del proceso de distribución de costos.	1h:35m	2h:25m
Revisión y validación de la información generada.		
Proceso de distribución de gastos de actividades a... órdenes / lotes de producción - Segunda secuencia		
Generación y Llenado de plantillas.	2h:14m	23h
Ejecución del proceso de distribución de costos indirectos a OP / Lotes.	2h:8m	6h:40m
Revisión y validación de la información generada.	4h	4h
Pruebas técnicas costos industriales (pre-producción)	25min	1h:15m
Ejecución del proceso de distribución de costos directos a OP / Lotes.	0h:49 m	3h:09m
Elaboración de costos industriales	0h:17m	1h:12m

Tabla 03-03. Cuadro Comparativo Resumen de los Tiempos de Ejecución del Proceso de Costeo Industrial antes y después del afinamiento de rendimiento

3.6.2.5. Verificación de meta cumplida

Como los tiempos de respuesta mejoraron en más del 50% respecto la situación inicial, se concluye que la actividad de performance tuning del proceso de costeo concluyó satisfactoriamente.

3.6.2.6. Validación de hipótesis

Se verifica la validez de la hipótesis planteada en la tesis.

El afinamiento del rendimiento mejoró los tiempos de respuestas de los procesos de costeo industrial. Esto permitió liberar recursos del computador central y permitir que más transacciones a nivel global se concreten, mejorando a su vez la escalabilidad de las soluciones en conjunto.

El análisis y diseño del modelo lógico de datos es parte crucial del rendimiento. La redefinición de las tablas permitió bajar en más del 50% el consumo de recursos inicial.

3.6.3. Afinamiento para Meta 2: Eliminar al 50% los cuellos de botella en operación normal

3.6.3.1. Análisis de cuellos de botella

Para eliminar los cuellos de botella de la operación normal y teniendo en cuenta la arquitectura centralizada en el servidor de Base de Datos, se deben tomar muestras de actividad de este componente central. El servidor es un computador IBM RS/6000 pSeries modelo 7038-6M2. Está ubicado en el nivel alto de servidores midrange por la capacidad que ostenta (índice rperf 12.5 con 6 procesadores y 8.93 con 4 procesadores). Este servidor integra un cluster activo/pasivo de dos servidores HACMP que brinda a la organización alta disponibilidad. El otro servidor es un computador IBM RS/6000 modelo 7025-F80.

Se tomaron muestras de actividad a nivel de Base de Datos y de plataforma (sistema operativo).

Para las muestras de sistema operativo AIX (versión UNIX de IBM) nos valimos de los comandos **sar**, **vmstat**, **iostat** y **netstat**.

Para las muestras de Base de Datos nos valimos de los utilitarios UTLBSTAT/UTLESTAT.

i) Muestras de sistema operativo

Vía netstat se mostró que había un congestionamiento en la tarjeta de red principal "en0" (tarjeta Fast Ethernet) que trabaja a 100 Mbps. Todos los usuarios (en promedio generando 1500 conexiones) accesaban al servidor por esta tarjeta. Se adjunta una muestra del comando:

```
$ netstat -I | grep en0
Name  Mtu   Network     Address            Ipkts Ierrs
Opkts Oerrs  Coll
en0   1500  link#2      0.0.25.55.a3.6e   566634788
0 304172943     0     0
en0   1500  172.16      SFPRODUCCION      566634788
0 304172943     0     0
```

Vía vmstat y sar, los consumos de CPU eran en promedio 70% y había altos niveles de paginación. La configuración inicial era 4 procesadores de 1.4 GHz y 8 GB de memoria RAM.

Vía iostat se verificó que el I/O en el servidor se distribuía de la siguiente manera: 70 % read y 30% write. De la parte correspondiente a WRITE, el 76% del mismo estaba cargado a un solo disco (hdisk0): respaldos lógicos DMPs, AIX sw, RDBMS sw, 50% de paging space.

ii) Muestras de Base de Datos

Por UTLBSTAT/UTLESTAT, ejecutados en intervalos de tiempo que consideraban las horas pico de atención se

encontró que los eventos de espera de Base de Datos más representativos eran:

- "db file sequential read"
- "enqueue"
- "db file scattered read"
- "log file sync"
- "library cache load lock"
- "inactive session"

De las muestras de sistema operativo y Base de Datos se evidenció cuatro grandes cuellos de botella. Ellos representan los siguientes rubros a optimizar:

- Desbalance de tráfico de networking
- Saturación de recursos computacionales: CPU y memoria
- Desbalance de I/O
- Contención de recursos: locks de aplicación.

3.6.3.2. Aplicación de Metodología

Se realizó una pasada completa al método de afinamiento expuesto en la sección 3.5.3, que abarcó desde el afinamiento las operaciones de Base de Datos hasta el afinamiento de la plataforma subyacente.

i) Para mejorar el desbalance de tráfico de networking se plantearon las siguientes opciones:

- Adición de tarjeta de red Gigabit-Ethernet (1000 Mbps) al servidor 6M2.

- Cambio de roles de tarjetas en servidor principal: La principal sería la tarjeta GB Ethernet y la secundaria la tarjeta Fast Ethernet.

- Adición de segundo listener a la tarjeta GB Ethernet para balanceo de carga (load balance).

- Redistribución de carga usuaria: 70% a nueva tarjeta y 30% a tarjeta anterior.

- Aplicación de Transparent Failover a los clientes. Si no ingresan al servidor por la tarjeta por defecto establecida para ellos, cambian a un segundo listener escuchando en la otra tarjeta.

- Desactivación del algoritmo de Nagle para que el tráfico TCP/IP fluya más rápidamente (protocol.ora: TCP.NODELAY=YES).

- Declaración de servidor 6M2 como nodo principal para transacciones distribuidas estableciéndolo como Commit Point Site (parámetro de instancia COMMIT_POINT_STRENGTH = 200).

- Cambio de modo síncrono a asíncrono en las replicaciones de datos implementadas in-house.

ii) Para mejorar el desbalance de I/O se planteó lo siguiente:

- Adición de nuevo disco independiente para la paginación.

- Adición de nuevo disco independiente para albergar los respaldos.

- Mover los tablespaces más utilizados a unidades separadas.

- Separar los tablespaces de datos e índices en unidades separadas.

ii) Para mejorar la contención de recursos por eventos se planteó lo siguiente:

db file sequential read

Impacto en el rendimiento:

Esperas para completar un I/O read.

Estrategias para reducirlo:

- Revisar accesos SQL que usan Index Scans no selectivos.

- Incrementar el cache de datos: DB_BLOCK_BUFFERS (requirió upgrade a Oracle9i de 64 bits)

- Particionamiento de tablas más grandes.

enqueue

Impacto en el rendimiento:

Esperas por "locks" de aplicaciones y RDBMS.

Estrategias para reducirlo:

- Eliminación de espera por "local lock". Hay tres tipos de locks: TX, TM y ST.

    - TX: Referido al manejo de las aplicaciones.
    - TM: Si foreign key constraints no han sido indexados.
        - Añadir en lo posible índices a FKs
    - ST: Por sorts y coalesce:
        - Mejorar sorts (tratando de disminuir sorts en disco)
        - Aumentar SORT_AREA_SIZE, SORT_MULTIBLOCK_READ_COUNT.
        - Tablespaces temporales de Dictionary Managed a Locally Managed
    - Reducir fragmentación de tablespaces
        - Tablespaces de Dictionary Managed a Locally Managed

db_file_scattered_read

Impacto en el rendimiento:

Sesiones esperando multiblock I/O para completar (Full table scans e Index Fast Full Scans).

Estrategias para reducirlo:

- Disminuir Full Table Scans e Index Fast Full Scans: Mejorar accessos SQL.
- Aumentar DB_FILE_MULTIBLOCK_READ_COUNT.
- CACHE option para tablas frecuentemente consultadas.

log file sync

Impacto en el rendimiento:

Tiempo esperado por los usuarios en la confirmación de commits o rollbacks.

Estrategias para reducirlo:

- Reducir la frecuencia de commits en las aplicaciones.
- Acelerar la escritura a redo logs.
- Considerar la opción de NOLOGGING al crear objetos (tablas e índices), principalmente temporales.
- FAST_START_PARALLEL_ROLLBACK a HIGH

library cache load lock

Impacto en el rendimiento:

Invalidaciones de objetos degradan el rendimiento de la BD.

Estrategias para reducirlo:

- Evitar compilaciones o recompilaciones de vistas u objetos PL/SQL durante horas pico.
- Administrar privilegios principalmente por roles de BD en vez de asignarlos explícitamente a cada usuario.
- Evitar dependencias cruzadas en PL/SQL packages (jerarquizar las estructuras).

inactive session

Impacto en el rendimiento:

Los procesos inactivos consumen recursos.

Estrategias para reducirlo:

- Activar "Dead Connection Detection" (Parámetro de Net8: SQLNET.EXPIRE_TIME). En horas pico de campaña comercial se ha llegado a registrar 1500 conexiones.

3.6.3.3. Aplicación de cambios

i) Desbalance de tráfico de networking

Para mejorar este rubro se aplicaron los siguientes cambios.

- Adición de tarjeta de red Gigabit-Ethernet (1000 Mbps) al servidor 6M2.

- Cambio de roles de tarjetas en servidor principal: La principal es GB Ethernet y la secundaria Fast Ethernet.

- Adición de segundo listener a la tarjeta GB Ethernet. El listener es un proceso background en espera de nuevas conexiones de los clientes al servidor de Base de Datos.

- Redistribución de carga usuaria por la red: 70% a la nueva tarjeta y 30% a la tarjeta anterior.

- Aplicación de Transparent Failover para las conexiones de los clientes al servidor de Base de Datos. Si no ingresan al servidor por la tarjeta por defecto establecida para ellos, cambian a un segundo listener escuchando en la otra tarjeta.

- Desactivación del algoritmo de Nagle para que el tráfico TCP/IP fluya más rápidamente (registro en el archivo protocol.ora: TCP.NODELAY=YES).

- Declaración de servidor 6M2 como nodo principal para transacciones distribuidas estableciéndolo como Commit Point Site (parámetro de instancia de Base de Datos: COMMIT_POINT_STRENGTH = 200).

- Cambio de modo síncrono a asíncrono en las replicaciones de datos implementadas in-house para asegurar la fluidez de las aplicaciones.

ii) Saturación de recursos computacionales: CPU y memoria

Para mejorar este rubro se aplicaron los siguientes cambios.

- Se aumentó en 50% la capacidad de cómputo del servidor de Base de Datos central, aumentando en esa proporción CPUs y memoria RAM. De 4 procesadores se pasó a 6 procesadores, y de 8 GB de RAM se pasó a 12 GB de RAM.

- Se activó el feature de pinned shared memory a nivel de sistema operativo AIX 5L (/usr/samples/kernel/vmtune -S 1) y de Base de Datos (parámetro LOCK_SGA=TRUE). Esto mejora el rendimiento de Base de Datos al impedir que el algoritmo de paginación normalmente considere a las páginas del SGA de la instancia de Oracle (caché de datos y SQL) para intercambios entre memoria y disco.

iii) Desbalance de I/O

Para mejorar este rubro se aplicaron los siguientes cambios.

- El tablespace más leído fue FACTURACIÓN, y se independizó en un volumen RAID-1 independiente.

- Los tablespaces más escritos fueron los temporales, por lo que se movieron a otro volumen RAID-1 independiente.

iv) Contención de recursos

Para mejorar este rubro se aplicaron los siguientes cambios.

- Cambio de sintaxis de las sentencias SQL que inducían FULL SCAN TABLE a las tablas más grandes. Esto se hace modificando las sentencias SQL para los accesos a las tablas resueltos por el motor tomen los índices disponibles.

- Compresión de índices no selectivos. Se llegó a ganar el 80% de espacio consumido por los índices no comprimidos. Asimismo, esto mejora el rendimiento de los Index Fast Full Scans.

- Adición de índices a foreign keys en tablas que participaban regularmente en deadlocks.

- Aumento del parámetro SORT_AREA_SIZE de 130 KB a 1 MB para trasladar los sorts de disco a memoria.

- Aumento del parámetro SORT_MULTIBLOCK_READ_COUNT de 2 a 4 para agilizar los sorts a disco (grandes ordenaciones).

- Migración de los tablespaces temporales de DICTIONARY MANAGED a LOCALLY MANAGED. Esto disminuye la contención en el diccionario de datos ocasionado por los procedimientos de administración automática de espacios.

- Adición del parámetro NOLOGGING a las sentencias de creación de tablas temporales para evitar su registro en el transaction log (redo log) de la Base de Datos.

- Aumento del parámetro FAST_START_PARALLEL_ROLLBACK de LOW (2*CPU_COUNT) a HIGH (4*CPU_COUNT). Esto agiliza el rendimiento de los rollback transactions ya que aumenta la paralelización de estas transacciones.

- Activación de Dead Connection Detection (SQLNET.EXPIRE_TIME=10) para evitar sesiones inactivas consumiendo y bloqueando recursos en la Base de Datos. Esto permite que

Net8 envíe tramas cada 10 minutos sensando la actividad de los usuarios. Si un usuario no responde, es desconectado y los recursos que tomó (en modo shared lock o exclusive lock) son liberados. El valor de 10 minutos es el recomendado por Oracle Corporation.

3.6.3.4. Evaluación de mejoras aplicadas

A nivel de plataforma-sistema operativo, las tasas de consumo de CPU bajaron en promedio en 20%. Esto es consecuencia del aumento de capacidad disponible del equipo principal. Se muestra en la tabla 03-04 un comparativo resumen:

Configuración Servidor	Rperf	Uso Inf	Uso Sup	Max uso recomendado de CPU de 70%
6M2 6 way 1450Mhz	12.55	5.30	6.02	Consumo de CPU entre 40% 50%
6M2 4 way 1450Mhz	8.93	5.30	6.02	Consumo de CPU entre 60-70%

Tabla 03-04. Cuadro Comparativo Resumen del uso de los procesadores antes y después de la tarea de afinamiento

En realidad, el consumo de CPU es el mismo (45% de 6 es igual a 65% de 4) pero el número de usuarios conectados ha aumentado en 50%.

A nivel de se balanceó la carga de red. La tarjeta en0 pasó a ser la GB Ethernet y la en3 es ahora la inicial Fast Ethernet. El

70% del tráfico se ha llevado a una tarjeta 10 veces más rápida que la original.

```
$ netstat -I | grep -e en0 -e en3
Name  Mtu   Network     Address            Ipkts Ierrs
Opkts Oerrs  Coll
en0   1500  link#2      0.0.25.55.a3.6e   485624783
0 294134945     0     0
en0   1500  172.16      SFPRODUCCION      485624783
0 294134945     0     0
en3   1500  link#5      0.2.55.33.75.64   259633824
0 118381486     0     0
en3   1500  172.16      SFPRODUCCION      259633824
0 118381486     0     0
```

A nivel de Base de Datos, podemos mencionar los siguientes hitos:

- La carga usuaria ha aumentado en 50% y los niveles de uso de la máquina son los mismos. Eso indica que mejoramos el funcionamiento del componente software para aprovechar mejor los recursos del computador. Escalabilidad y rendimiento.

- El 50% de ordenaciones que se ha transferido de disco a memoria. Un acceso a memoria es casi un millón de veces más rápido que a disco.

- Se han reducido en 50% los FULL SCAN tables a las tablas más grandes. Esto mejora los tiempos de respuesta y los niveles de servicio.

- Se procesan fluidamente hasta 16 transacciones por segundo. Antes de los cambios implantados, se procesaban 3 transacciones por segundo, y ante cargas mayores las latencias eran notorias. En campaña comercial se aumentó en 50% el número de usuarios concurrentes y los tiempos de respuesta de las transacciones OLTP mejoraron en 20%.

3.6.3.5. Verificación de meta cumplida

Se mejoró en 50% los principales indicadores de actuación de los sistemas y del equipo central de la arquitectura IT del negocio.

3.6.3.6. Validación de hipótesis

Se verifica la validez de la hipótesis planteada en la tesis.

El afinamiento del rendimiento mejoró la escalabilidad de la solución sin hacer ampliaciones al hardware.

El análisis y diseño del modelo lógico de datos es parte crucial del rendimiento. Se optimizaron muchas sentencias SQL para llegar a los niveles de rendimiento y escalabilidad obtenidos.

CAPITULO IV

APLICACIÓN DE AFINAMIENTO DE RENDIMIENTO EN GRUPO ENDESA

4.1. DESCRIPCION DEL GRUPO ENDESA

ENDESA es la mayor empresa eléctrica de España y la primera compañía eléctrica privada de Iberoamérica. Es un operador eléctrico relevante en el arco europeo mediterráneo. Además, tiene una presencia creciente en el mercado español de gas natural.

MISIÓN

- Un operador del negocio energético y de servicios conexos, centrado en la electricidad.
- Una compañía multinacional, responsable, eficiente y competitiva, comprometida con la seguridad, la salud y el medio ambiente.
- Una empresa preparada para competir globalmente.

VISIÓN

- Maximizar el valor de la inversión de sus accionistas.
- Servir a sus mercados superando las expectativas de sus clientes.
- Contribuir al desarrollo de sus empleados.

VALORES

- Personas

  Asegurarnos las oportunidades de desarrollo basadas en el mérito y en la aportación profesional.

- Seguridad y salud

  Nos comprometemos decididamente con la seguridad y salud laboral promoviendo una cultura preventiva.

- Trabajo en equipo

  Fomentamos la participación de todos para lograr un objetivo común, compartiendo la información y los conocimientos.

- Conducta Etica

  Actuamos con profesionalidad, integridad moral, lealtad y respeto a las personas.

- Orientación al cliente

  Centramos nuestro esfuerzo en la satisfacción del cliente, aportando soluciones competitivas y de calidad.

- Innovación

  Promovemos la mejora continua y la innovación para alcanzar la máxima calidad desde criterios de rentabilidad.

- Orientación a resultados

  Dirigimos nuestras actuaciones hacia la consecución de los objetivos del proyecto empresarial y de la rentabilidad de nuestros accionistas, tratando de superar sus expectativas.

- Comunidad y Medio Ambiente

  Nos comprometemos social y culturalmente con la Comunidad y adaptamos nuestras estrategias empresariales a la preservación del medio ambiente.

ENDESA es la primera compañía eléctrica de España y la principal empresa eléctrica en Chile, Argentina, Colombia y Perú y está presente en Brasil. Es un operador eléctrico relevante en el arco europeo mediterráneo.

A través de sus compañías participadas, controla 39.642 MW de potencia instalada, con una generación en 2009 de 137.054 GWh. ENDESA produce fuera de España el 40% del total de la electricidad que genera. Las ventas de electricidad en los mercados en los que opera ascendieron a 169.966 GWh.

ENDESA cuenta con 24,6 millones de clientes.

En Perú se halla el 10,2% de los activos de ENDESA en Iberoamérica.

ENDESA gestiona 1.598 MW de potencia. Tiene una participación de control del 83,60% en la generadora Edegel, con 1.467 MW, y del 60% de la Empresa Eléctrica de Piura (Eepsa), con 131 MW de potencia instalada.

Endesa posee además, junto con Enersis, una participación de control en Edelnor, empresa que distribuye energía en la zona norte de Lima a 1 millón clientes.

4.2. DESCRIPCION DE LOS SISTEMAS DE INFORMACION

Las aplicaciones de ENDESA trabajan en modelo cliente-servidor de n-capas, y han sido desarrolladas por terceros. Es decir, muchas de las aplicaciones trabajan con interfaces, dado que cada aplicación cubre una cierta área funcional del grupo. Para distribuir esta disímil mezcla, las aplicaciones han sido segregadas físicamente por área funcional a la que sirven. Las áreas son:

- Comercial
- Técnica
- Intranet/Internet: Web de Reclamos, Web de Lecturas, Web de Morosidad.
- Bases de Datos Especializadas, en la que se incluye Gestión Documentaria.

4.3. INFRAESTRUCTURA IT

Los servidores de Base de Datos de la infraestructura IT de ENDESA se resumen en la figura 04-01.

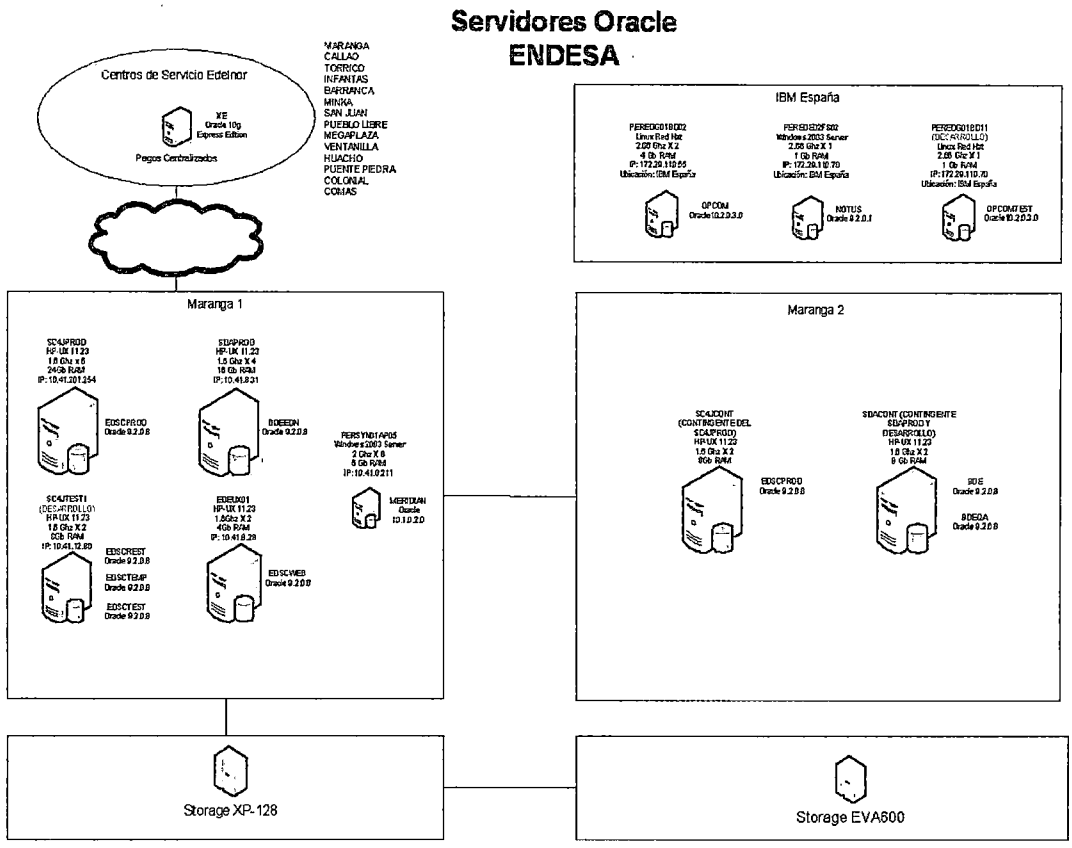


Figura 04-01. Servidores de Base de Datos Oracle de la operación de ENDESA en Perú

En la figura 04-02 se detalla el recuadro "Maranga 1" de la figura 04-01, que es donde se concentran los servidores de Base de Datos críticos de la operación.

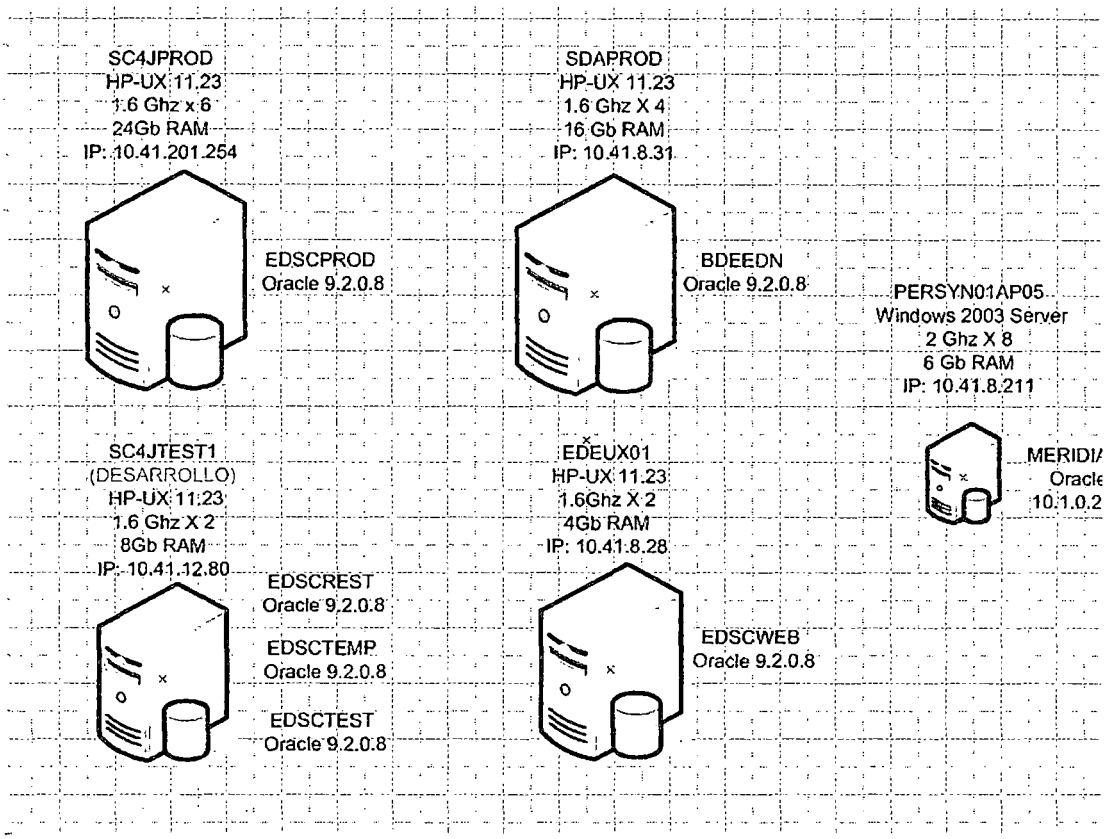


Figura 04-02. Servidores críticos de Base de Datos Oracle de la operación de ENDESA en Perú

Inventario de Servidores:

Bases de Datos de Producción:

- SC4JPROD / EDSCPROD: Sistema Comercial
- SDAPROD / BDEEDN: Sistemas Técnicos
- PERSYN01AP05 / MERIDIAN: Gestión Documentaria
- EDEUX01 / EDSCWEB: Portales Web de EDELNOR

Base de Datos QA:

- SC4JTEST1 / EDSCREST: Sistema Comercial
- SDACONT / BDEQA: Sistemas Técnicos

Bases de Datos de Desarrollo:

- SC4JTEST1 / EDSCTEMP: Sistema Comercial
- SC4JTEST1 / EDSCTEST: Portales Web EDELNOR
- SDACONT / BDE: Sistemas Técnicos

4.4. RETOS Y NECESIDADES DEL NEGOCIO

ENDESA requería que se eliminasen los problemas de rendimiento en los siguientes servidores de Base de Datos:

- SDAPROD: Servidor donde reside la Base de Datos BDEEDN que soporta a los Sistemas Técnicos.

- EDEUX01: Servidor donde reside la Base de Datos EDSCWEB que soporta a los Sistemas Web desplegados en la Intranet e Internet.

4.5. ENFOQUE METODOLOGICO

Se aplicará el enfoque metodológico expuesto en las secciones 2.2 y 3.5.3.

4.6. PROCESO DE AFINAMIENTO

4.6.1. Metas de Afinamiento

Meta 1: Bajar los tiempos de transacción de Base de Datos promedio en 50% en Sistemas Técnicos

Meta 2: Estabilizar en el servidor de Base de Datos Web los niveles iniciales de consumo de CPU por introducción de nueva aplicación y bajar el consumo de memoria a menos de 80%

4.6.2. Afinamiento de Meta 1: Bajar los tiempos de transacción de Base de Datos promedio en 50% en Sistemas Técnicos

4.6.2.1. Análisis de cuellos de botella

El equipo cuenta con 4 procesadores PA-RISC y 16 GB de RAM con sistema operativo HP-UX 11i. En dicho equipo se encuentra operando una sola instancia de Base de Datos (BDEEDN), la cual ocupa 6.5 GB en memoria. La Base de Datos mide 285 GB.

La CPU trabaja al 50% ó más en forma sostenida. Hay contención de recursos tanto en CPU como en I/O. En I/O, se llegan a llenar archive logs de 100 MB a una tasa acelerada.

Hay 498 conexiones de Base de Datos, y se observa contención por bloqueos (lock waits).

DATOS RESALTANTES

- Soporta los sistemas técnicos y GIS.
- Oracle9i Enterprise Edition Release 9.2.0.8.0, que reside en un servidor con 4 procesadores, 16 GB de RAM bajo HP-UX 11i.
- La Base de Datos está configurada en 6.5 GB de memoria.
- La Base de Datos mide 286 GB, y soporta 800 usuarios (soft limit).
- DB Links a las Base de Datos Comercial, Web y Meridian.
- Soporta carga batch a demanda.
- Problemas observados:
    - Considerable uso de CPU: 50% en promedio
    - Espera por recurso I/O: 30% en promedio del tiempo de CPU
    - Alto uso de I/O (cache buffer LRU chains): Buffer cache size: Buffer Cache Hit Ratio entre 70% a 80%
    - 1 Full Scan Table por segundo: db_file_mulitblock_read_count = 64

El cuello de botella es el evento de espera "db file sequential read", lo cual se manifiesta externamente por los altos niveles de uso de CPU y memoria. Asimismo, el buffer cache hit ratio es relativamente bajo, considerando que la carga mayoritariamente es transaccional.

4.6.2.2. Aplicación de Metodología

La forma de atenuar los efectos del evento de espera "db file sequential read" y mejorar el "buffer cache hit ratio" es aumentar el área de caché de datos, denominada DB_CACHE_SIZE.

La estrategia de CACHING es un método fundamental de remover cuellos de botella de rendimiento que son resultados de accesos lentos a los datos. La buferización (caching) mejora el rendimiento reteniendo información utilizada frecuentemente en memoria de alta velocidad, léase RAM, lo cual reduce el tiempo de acceso y por lo tanto mejora el rendimiento.

La buferización es una manera efectiva de mejorar el rendimiento en situaciones donde se manifiesta el PRINCIPIO DE LOCALIDAD DE REFERENCIA (locality of reference principle) [WI01]. Este principio es el fenómeno cuando un mismo valor o locaciones de memoria son frecuentemente consultadas.

4.6.2.3. Aplicación de cambios

De acuerdo al cuello de botella identificado, se procedió a aumentar el valor del parámetro DB_CACHE_SIZE de 3 GB a 4.5 GB

La vista inicial (con DB_CACHE_SIZE de 3 GB) de operación de la Base de Datos de Sistemas Técnicos (BDEEDN) se muestra en la figura 04-03.

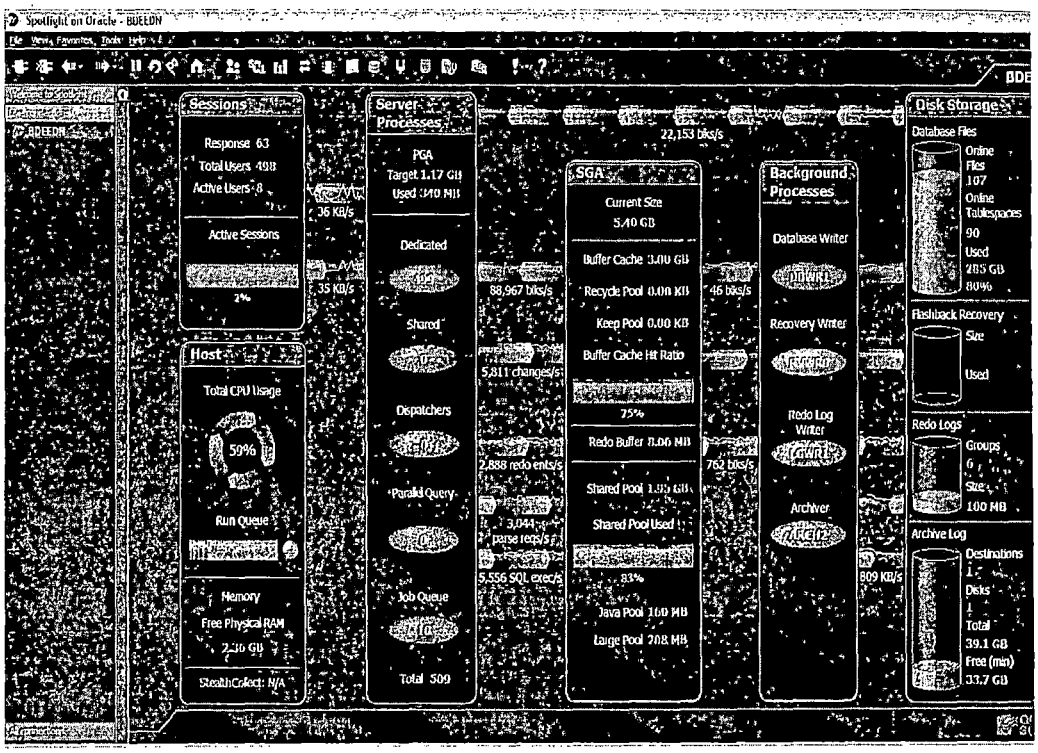


Figura 04-03. Vista de operación de servicio de Base de Datos de Sistemas Técnicos (BDEEDN)

4.6.2.4. Evaluación de mejoras aplicadas

Se evaluaron los indicadores de escalabilidad para la Base de Datos en ventanas de pico de uso:

- 09:00 a 12:00 horas: Perfil Mañana
- 15:00 a 18:00 horas: Perfil Tarde
- 23:00 a 06:00 horas: Perfil Noche

Se obtuvieron los resultados mostrados en la tabla 04-01.

	PERFIL MAÑANA (09:00 - 12:00)		
	Lecturas Físicas/segundo	Transacciones/segundo	Buffer Hit %
Antes del Cambio (01/02/2010 al 02/02/2010)	22,570.97	6.70	67.80
Después del Cambio (08/02/2010 al 09/02/2010)	19,248.79	7.97	79.98

	PERFIL TARDE (15:00 - 18:00)		
	Lecturas Físicas/segundo	Transacciones/segundo	Buffer Hit %
Antes del Cambio (01/02/2010 al 02/02/2010)	19,306.28	3.67	70.21
Después del cambio (08/02/2010 al 09/02/2010)	18,575.90	8.62	77.36
	PERFIL NOCHE (23:00 - 06:00)		
	Lecturas Físicas/segundo	Transacciones/segundo	Buffer Hit %
Antes del Cambio (01/02/2010 al 02/02/2010)	17,786.27	29.77	55.26
Después del Cambio (08/02/2010 al 09/02/2010)	16,385.95	63.87	81.88

Tabla 04-01. Comparativo de resultados de Escalabilidad antes y después el cambio

En todas las ventanas de uso, se aumentó el throughput (transacciones por segundo), se disminuyeron las lecturas a disco y se mejoraron los "buffer cache hit ratio".

La tabla 04-02 muestra los tiempos de respuesta en el perfil MAÑANA antes y después del afinamiento de rendimiento.

TIEMPO DE RESPUESTA – ANTES

	Tiempo	Porcentaje	Por Ejecución	Por Llamada de Usuario	Por Transacción
Tiempo de Respuesta	8,651,878.00	100.00 %	0.69	1.39	11,474.64
Tiempo de CPU	1,863,878.00	21.54 %	0.15	0.3	2,471.99
Tiempo de Espera	6,788,000.00	78.46 %	0.54	1.09	9,002.65

TIEMPO DE RESPUESTA – DESPUES

	Tiempo	Porcentaje	Por Ejecución	Por Llamada de Usuario	Por Transacción
Response Time	4,816,856.00	100.00 %	0.62	0.37	3,688.25
Tiempo de CPU	2,144,356.00	44.52 %	0.28	0.17	1,641.93
Tiempo de Espera	2,672,500.00	55.48 %	0.35	0.21	2,046.32

Tabla 04-02. Comparativo de resultados en Tiempos de Respuesta del Perfil Mañana antes y después el cambio

La tabla 04-03 muestra los tiempos de respuesta en el perfil TARDE antes y después del afinamiento de rendimiento.

TIEMPO DE RESPUESTA – ANTES

	Tiempo	Porcentaje	Por Ejecución	Por Llamada de Usuario	Por Transacción
Tiempo de Respuesta	13,745,867.00	100.00 %	1.4	1.86	15,799.85
Tiempo de CPU	1,708,467.00	12.43%	0.17	0.23	1,963.76
Tiempo de Espera	12,037,400.00	87.57%	1.23	1.63	13,836.09

TIEMPO DE RESPUESTA – DESPUES

	Tiempo	Porcentaje	Por Ejecución	Por Llamada de Usuario	Por Transacción
Response Time	8,225,903.00	100.00 %	0.67	0.78	8,393.78
Tiempo de CPU	1;981,103.00	24.08%	0.16	0.19	2,021.53
Tiempo de Espera	6,244,800.00	75.92%	0.51	0.59	6,372.24

Tabla 04-03. Comparativo de resultados en Tiempos de Respuesta del Perfil Tarde antes y después el cambio

La tabla 04-04 muestra los tiempos de respuesta en el perfil NOCHE antes y después del afinamiento de rendimiento.

TIEMPO DE RESPUESTA – ANTES

	Tiempo	Porcentaje	Por Ejecución	Por Llamada de Usuario	Por Transacción
Response Time	19,154,082.00	100.00%	1.07	1.62	354,705.22
Tiempo de CPU	2,823,182.00	14.74%	0.16	0.24	52,281.15
Tiempo de Espera	16,330,900.00	85.26%	0.91	1.38	302,424.07

TIEMPO DE RESPUESTA – DESPUES

	Tiempo	Porcentaje	Por Ejecución	Por Llamada de Usuario	Por Transacción
Response Time	29,713,455.00	100.00%	1.39	1.9	48,710.58
Tiempo de CPU	4,095,355.00	13.78%	0.19	0.26	6,713.70
Tiempo de Espera	25,618,100.00	86.22%	1.2	1.64	41,996.89

Tabla 04-04. Comparativo de resultados en Tiempos de Respuesta del Perfil Noche antes y después el cambio

El tiempo de respuesta promedio por transacción ha bajado. El resumen es el siguiente:

- Tiempo de respuesta promedio por transacción (Perfil Mañana) baja de 114 seg a 36 seg

- Tiempo de respuesta promedio por transacción (Perfil Tarde) baja de 157 seg a 83 seg

- Tiempo de respuesta promedio por transacción (Perfil Noche) baja de 3,547 seg a 487 seg

4.6.2.5. Verificación de meta cumplida

Los tiempos de respuesta bajaron más del 50% de lo establecido en la meta. Asimismo, la escalabilidad de la solución ha mejorado porque se utilizan menos recursos para manejar la misma carga, lo que permite que más transacciones se procesen por unidad de tiempo.

4.6.2.6. Validación de hipótesis

Se verifica la validez de la hipótesis planteada en la tesis.

El afinamiento de rendimiento brinda mejoras sustanciales a los tiempos de respuesta, mejorando asimismo la escalabilidad de la solución al aprovechar mejor los recursos computacionales.

4.6.3. Afinamiento de Meta 2: Estabilizar en el servidor de Base de Datos Web los niveles iniciales de consumo de CPU por introducción de nueva aplicación y bajar el consumo de memoria de 99% a 80%

4.6.3.1. Análisis de cuellos de botella

DATOS RESALTANTES

- Soporta los portales web de EDELNOR: Web de Reclamos, Web de Lecturas y Web de Morosidad.
- Oracle9i Enterprise Edition Release 9.2.0.8.0 a 64 bits, que reside en un servidor con 2 procesadores, 4 GB de RAM bajo HP-UX 11i.
- La Base de Datos está configurada con 1 GB de memoria.
- La Base de Datos mide 47.6 GB, y soporta 129 usuarios.
- Indicadores de rendimiento:
    - Uso de CPU moderado: 24% en promedio
    - Alto consumo de memoria: Encima del 90% en promedio.
    - Buffer Cache Hit Ratio es 20%
    - Contención en I/O: 50% de espera por I/O

El cuello de botella de la operación es la memoria. Debido a su saturación, nuevos procesos no pueden conectarse a la Base de Datos. Por ejemplo, se manifiesta este error:

El proceso que sube los libros a la base de dato de la web de lectura está fallando por conexión a la base de

datos, el mensaje que sale es el siguiente para que le envíes a los administradores de BD

Error al Leer Libro: I9126503.071 Io exception: Connection refused(DESCRIPTION=(TMP=)(VSNNUM=153094144)(ERR=12500)(ERROR_STACK=(ERROR=(CODE=12500)(EMFI=4))(ERROR=(CODE=12540)(EMFI=4))(ERROR=(CODE=12560)(EMFI=4))(ERROR=(CODE=510)(EMFI=4))(ERROR=(BUF='HPUX Error: 12: Not enough space'))))

Asimismo, una nueva aplicación con accesos SQL no eficientes está entrando. Hay que mejorar el manejo de la memoria RAM del servidor a nivel del sistema operativo (fuera del ámbito de Base de Datos) y afinar las sentencias SQL ineficientes.

En lo concerniente a los nuevos accesos SQL no optimizados, momentos luego del despliegue el consumo de CPU y disco se elevó a niveles del 100% y 40% respectivamente. En la figura 04-04 se muestra la vista previa del rendimiento a nivel de sistema operativo obtenido con el utilitario "glance".

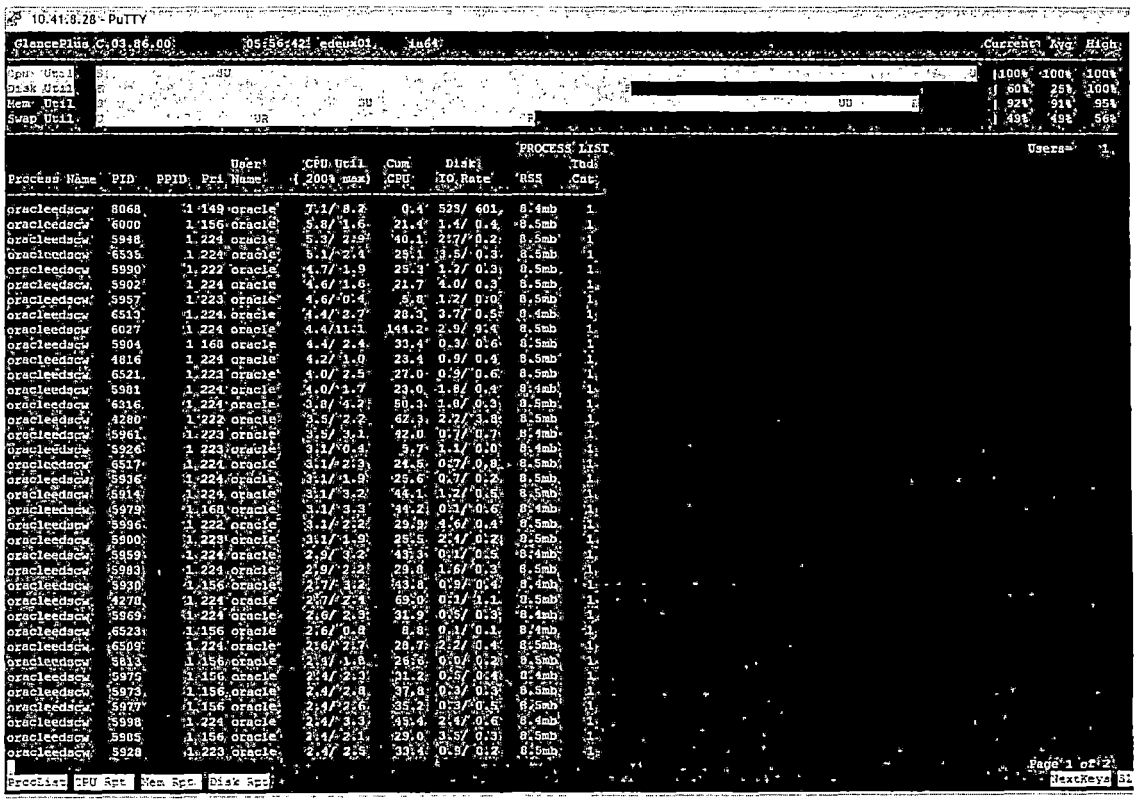


Figura 04-04. Vista de indicadores de sistema operativo antes de afinamiento de rendimiento

4.6.3.2. Aplicación de Metodología

De acuerdo a la metodología ya fundamentada:

En lo concerniente a la optimización del recurso memoria, había que revisar las mejores prácticas referentes al manejo de memoria en servidores de Base de Datos:

- El área de paginación debe ser por lo menos el doble de la memoria RAM, de acuerdo al proveedor del software de Base de Datos.

- El proveedor del sistema operativo HP-UX recomienda que la paginación sea el doble de la memoria RAM.

- Debido a la escasez de memoria, el proceso de creación de sesiones de Base de Datos se ve entorpecido dado que el listener al crear un proceso servidor dedicado no cierra su proceso copia (spawn) reservando memoria virtual que no se liberaba. Esto sobrecargaba aún más el consumo de memoria.

  Esto se observaba en la lista de procesos de la siguiente manera:

```
$ ps -ef | grep tnslsnr

oracle9i 8909 1 0 Sep 15 ? 902:44 /u05/9iHOME/DBHOME/bin/tnslsn
r sales -inherit
oracle9i 22685 8909 0 14:19:23 ? 0:00 /u05/9iHOME/DBHOME/bin/tn
slsnr sales -inherit
```

En lo concerniente a los nuevos accesos SQL no optimizados, era evidente la optimización de código. Para nuestros propósitos, los servidores de aplicación en gran número ejecutaban sentencias que hacía FULL SCAN TABLE a la tabla VISION.MEDICAO, la cual tenía 1'710,724 registros. La sentencia SQL más relevante era:

```
edscweb> SELECT medicao0_.medicao_id medicao1_10_, medicao0_.idv2s
idv2_10_,
  2         medicao0_.eventos eventos10_, medicao0_.datahora
datahora10_,
  3         medicao0_.zona zona10_, medicao0_.canal1 canal6_10_,
  4         medicao0_.periodo periodo10_, medicao0_.canal2 canal8_10_,
```

```
  5         medicao0_.canal3 canal9_10_, medicao0_.canal4 canal10_10_,
  6         medicao0_.canal5 canal11_10_, medicao0_.canal6
canal12_10_,
  7         medicao0_.canal7 canal13_10_, medicao0_.canal8 canal14_10_
  8      FROM medicao medicao0_
  9      WHERE medicao0_.idv2s = 1
 10        AND medicao0_.datahora = systimestamp

Execution Plan
----------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE (Cost=1169 Card=117 Bytes=
5733)
   1    0   TABLE ACCESS (FULL) OF 'MEDICAO' (Cost=1169 Card=117
Bytes=5733)
```

No existía índice por la clave compuesta (IDV2S, DATAHORA).

4.6.3.3. Aplicación de cambios

Para mejorar el consumo de memoria RAM se aplicaron los siguientes cambios:

- Aumento de la paginación de 4 GB a 8 GB (el doble de la memoria RAM).

- Inclusión del parámetro SUBSCRIBE_FOR_NODE_DOWN_EVENT_LISTENER con valor en OFF en el archivo listener.ora. Esto se menciona en la nota 340091.1 de Metalink (portal de resolución de problemas de Oracle Corporation).

Para optimizar la sentencia SQL se creó el índice por la clave compuesta (IDVS, DATAHORA). El costo de la sentencia bajó

de 1169 a 130, tal como se muestra a continuación en el plan de ejecución optimizado:

```
        Execution Plan
----------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE (Cost=130 Card=117
Bytes=5733)
   1    0   TABLE ACCESS (BY INDEX ROWID) OF 'MEDICAO' (Cost=130
Card=117 Bytes=5733)
   2    1     INDEX (RANGE SCAN) OF 'IDX01_MEDICAO' (NON-UNIQUE)
(Cost=42 Card=117)
```

4.6.3.4. Evaluación de mejoras aplicadas

A nivel de consumo de memoria RAM:

- El consumo de memoria RAM bajó de 99% a 74%
- El consumo de SWAP (paginación) bajó de 80% a 40%

El aumento de la paginación estaba destinado a eliminar los errores de "startup" (fork) de procesos UNIX. En HP-UX, cuando un proceso se levanta, separa espacio en memoria virtual, Si no la encuentra, no se levanta. Y la paginación antes del cambio se usaba al 80%.

La escalabilidad (throughput) mejoró porque se procesó más transacciones por unidad de tiempo:

Lunes 01/03/2010:	0.64 transacciones/segundo
Martes 02/03/2010:	3.84 transacciones/segundo
Semana anterior 23/02/2010:	2.86 transacciones/segundo

En la figura 04-05 se muestran los resultados descritos a nivel de sistema operativo.

Figura 04-05. Vista de indicadores de sistema operativo después de afinamiento de rendimiento en Memoria

A nivel de accesos SQL, luego de la adición del índice, el consumo de recursos se estabilizó. Normalmente, el servidor EDEUX01 no es un gran consumidor de CPU. De memoria consume entre 75% a 85% (esto se logró con el cambio al

parámetro de LISTENER.ORA). Y de disco, de 20% a 30% en promedio.

La figura 04-06 muestra la vista de los indicadores de rendimiento de sistema operativo luego de la adición del índice.

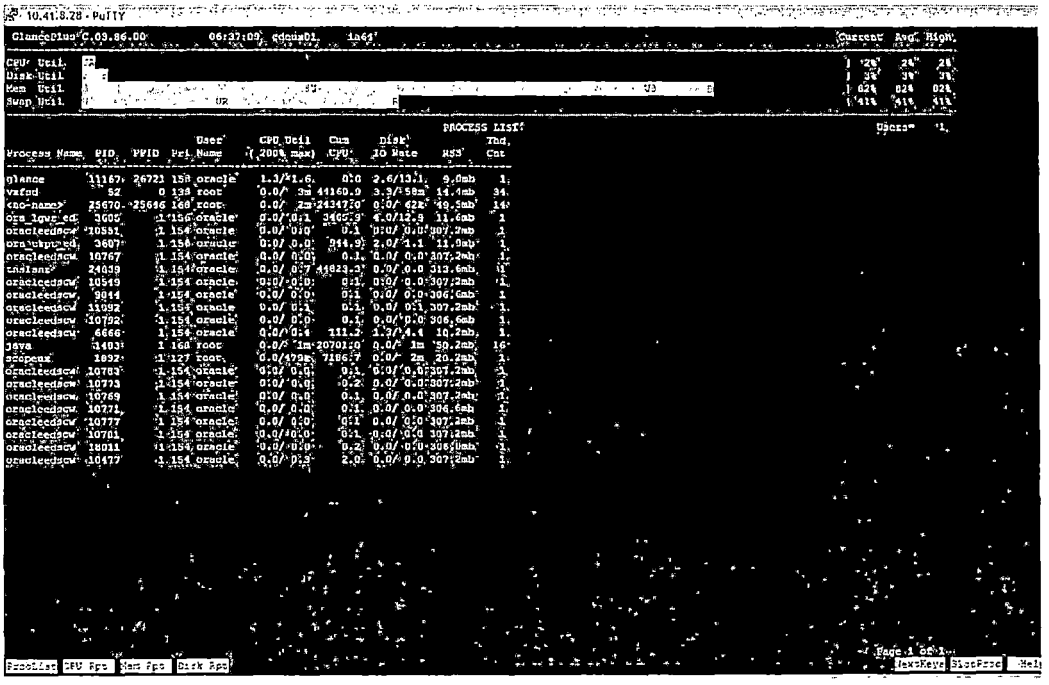


Figura 04-06. Vista de indicadores de sistema operativo después de afinamiento de rendimiento a accesos SQL

4.6.3.5. Verificación de meta cumplida

El rendimiento del servidor de Base de Datos se ve mejorado a nivel global porque la memoria disponible aumentó. Esto redundó en el aumento de la escalabilidad.

La optimización de accesos SQL hizo que de niveles de uso de 100% en CPU se baje a 20% en promedio.

4.6.3.6. Validación de hipótesis

Se comprueba la validez de las hipótesis de la tesis. La aplicación sistemática de las metodologías y técnicas de afinamiento de rendimiento permiten obtener mejores tiempos de respuesta y aumentar la escalabilidad de las soluciones sin incurrir en aumento drástico de hardware.

CAPITULO V

MUESTREO UTILIZADO

La investigación para generalizar los resultados y comprobar la validez de la hipótesis, presenta en la tabla 05-01 una muestra de 32 casos de éxito en situaciones de afinamiento de rendimiento en empresas del mercado peruano. Es una cantidad de muestras sugerida para este tipo de investigación sobre teorías fundamentadas [HE01].

El campo FRECUENCIA RELATIVA indica con qué frecuencia relativa se presentan situaciones de afinamiento de rendimiento en el área de optimización correspondiente.

No	EMPRESA / PROBLEMA	AREA DE OPTIMIZACION / DESCRIPCION DE LA SOLUCION	METRICAS INICIALES	METRICAS FINALES	FRECUENCIA RELATIVA
1	ALERTA MEDICA Tiempos de respuesta degradados en aplicaciones OLTP	Afinamiento de sentencias SQL El cuello de botella era que ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 1-3 minutos	Tiempo promedio de respuesta = 1-5 segundos	Alta
2	ALERTA MEDICA Indisponibilidad periódica del servicio de Base de Datos por caída intempestiva del mismo.	Afinamiento de la plataforma subyacente El cuello de botella era que el software de Base de Datos instalado no era el adecuado (era una versión de prueba) y que no ofrecía los niveles de servicio requeridos por el negocio (Workgroup Server). Se oficializó la licencia e instaló la versión Enterprise del software.	Caídas del servicio por día = 2-5 caídas	Caídas del servicio por día = 0 caídas	Baja
3	ALICORP Baja transaccionalidad en sistema SAP	Afinamiento de asignación de memoria El cuello de botella era el evento "db file sequential read". Se aumentó el área de caché de datos de 24 GB a 30 GB	Transacciones por segundo = 30	Transacciones por segundo = 40	Media

4	ALICORP Baja transaccionalidad en sistema SAP	Afinamiento de I/O y estructura física El cuello de botella era el evento "log file sync". Se aumentaron los tamaños de los archivos "redo log" de 50 MB a 1 GB.	Transacciones por segundo = 40	Transacciones por segundo = 45	Media
5	ALICORP Baja transaccionalidad en sistema SAP	Afinamiento de I/O y estructura física El cuello de botella era el evento "db file sequential read". Se crearon veinte (20) índices adicionales por evaluación de los accesos SQL más pesados.	Transacciones por segundo = 45	Transacciones por segundo = 60	Alta
6	BANCO WIESE Tiempos de respuesta degradados en aplicaciones OLTP	Afinamiento de sentencias SQL El cuello de botella era que ciertas sentencias SQL consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 1 minuto	Tiempo promedio de respuesta = 1-5 segundos	Alta
7	BANCO WIESE Tiempos de respuesta excesivos en procesos por lotes de la rutina nocturna	Afinamiento de operaciones de Base de Datos El cuello de botella era los cambios de contexto. Se convirtieron un cierto número de programas de lenguaje COBOL a lenguaje PL/SQL. Se bajaron considerablemente los cambios de contexto.	Tiempo promedio de ejecución = 2 horas	Tiempo promedio de ejecución = 15-20 minutos	Media

8	BANCO WIESE Tiempos de respuesta excesivos en transacciones en línea por efectos de demanda excesiva	Afinamiento de la plataforma subyacente El cuello de botella era la saturación de CPU y memoria del servidor central. Se trasladó el servidor central a un equipo con el doble de procesadores y el doble de memoria.	Tiempo promedio de respuesta = 5-10 segundos	Tiempo promedio de respuesta = 1-5 segundos	Baja
9	BANCO WIESE Tiempos de respuesta excesivos en transacciones en línea de sistema SAP	Afinamiento de rutas de acceso El cuello de botella era que los planes de acceso no eran los adecuados para los volúmenes de datos contenidos en las tablas. Una investigación resaltó que las tablas más grandes no tenían las estadísticas actualizadas.	Tiempo promedio de respuesta = 10-120 segundos	Tiempo promedio de respuesta = 1-5 segundos	Media
10	COPEINCA Tiempos de respuesta degradados en aplicaciones OLTP	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 1-3 minutos	Tiempo promedio de respuesta = 1-5 segundos	Alta

11	ENDESA Tiempos de respuesta degradados en aplicaciones OLTP	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 1 minuto	Tiempo promedio de respuesta = 1-5 segundos	Alta
12	ENDESA Baja transaccionalidad en Sistemas Técnicos	Afinamiento de asignación de memoria El cuello de botella era el evento de espera "db file sequential read", lo cual se manifiesta externamente por los altos niveles de uso de CPU y memoria. Asimismo, el buffer cache hit ratio es relativamente bajo, considerando que la carga mayoritariamente es transaccional. Se aumentó el área de caché de la Base de Datos.	Transacciones por segundo (Mañana/Tarde/Noche) = (6.70,3.67, 29.77	Transacciones por segundo (Mañana/Tarde/Noche) = (7.97,8.62, 63.87)	Media
13	ENDESA Alto consumo de memoria en servidor de Base de Datos de aplicaciones Web	Afinamiento de la plataforma subyacente El cuello de botella de la operación es la memoria. Debido a su saturación, nuevos procesos no pueden conectarse a la Base de Datos. Se aumenta el área de paginación y se incluye el parámetro SUBSCRIBE_FOR_NODE_DOWN_EVENT_LISTENER con valor en OFF en el archivo listener.ora.	Consumo de memoria RAM = 99% Consumo de paginación = 44%	Consumo de memoria RAM = 74% Consumo de paginación = 40%	Baja

14	ESSALUD Tiempos de respuesta de procesos en general degradados	Afinamiento de sentencias SQL El cuello de botella era que a nivel de sistema operativo se llegaba a un límite de procesos concurrentes. La causa raíz era que no se siguió el checklist de instalación.	Transacciones por minuto = 60	Transacciones por minuto = 400	Baja
15	ESSALUD Tiempos de respuesta degradados en aplicaciones batch	Afinamiento de sentencias SQL El cuello de botella era que ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 8 horas	Tiempo promedio de respuesta = 10 minutos	Alta
16	GRUPO SAN FERNANDO Imposibilidad de contar en línea con la información de ventas en campaña navideña. Los tiempos de los procesos batch de conciliación de datos demoraban más de 48 horas.	Afinamiento del diseño de datos Implantación de replicación avanzada (n-vías asíncrono). Se interconectaron 10 Bases de Datos que permitieron a los procesos contar con información actualizada de un grupo seleccionado de 40 tablas en los 10 sitios, eliminándose las copias de datos desde sitios remotos.	Tiempo de respuesta = 48 horas	Tiempo de respuesta = 20 minutos	Baja

17	GRUPO SAN FERNANDO Inclusión de nueva aplicación logística a la plataforma de producción, hace que los tiempos de respuesta en producción sean inaceptables, y elevando el consumo de recursos general en el servidor de Base de Datos.	Afinamiento de sentencias SQL Se identificaron las sentencias SQL que más consumían recursos en el servidor, y se procedió a reescribirlas y crear nuevos índices para que los costos de sus planes de ejecución bajen.	Tiempo de respuesta promedio ONLINE = 3 minutos	Tiempo de respuesta promedio ONLINE = 2 segundos	Media
18	GRUPO SAN FERNANDO Proceso batch de cierre contable y apertura de cuentas de inventario demoraba un tiempo excesivo (2 días). Se hizo un	Afinamiento de la contención de recursos El cuello de botella identificado fue el evento "SQL Net message from client". El proceso se ejecutaba en el servidor de Base de Datos, pero usaba como medio de transporte de datos el adaptador del protocolo TCP/IP. Cambiando al adaptador BEQUEATH se bajó la latencia del evento.	Transacciones por minuto = 27 cuentas por minuto	Transacciones por minuto = 60 cuentas por minuto	Baja

	upgrade de hardware (aumento de CPUs) y los tiempos de ejecución no variaron.				
19	GRUPO SAN FERNANDO Base de Datos parecía inhibirse de seguir operando (hanging) por bloqueo. Muchas tablas con claves foráneas sin índices de soporte.	Afinamiento del diseño de datos Se identificó primariamente que el cuello de botella era "enqueue". Muchas tablas tenían claves foráneas (foreign keys) pero no tenían índices de soporte. Si bien es cierto, el motor de Base de Datos permite esta situación, es una deficiencia de diseño no tener índices de soporte a los FKs. Sin FKs, las actualizaciones en las tablas padre bloquean completamente las actualizaciones en las tablas hijo. Se procedieron a crear todos los índices de soporte faltantes.	Número promedio de inhibición de Base de Datos por bloqueo al día = 3	Número promedio de inhibición de Base de Datos por bloqueo al día = 0	Media

20	GRUPO SAN FERNANDO Reducción de tiempos de ejecución de proceso de costeo industriales (Fase 1)	Afinamiento del diseño de datos El evento cuello de botella era "db file scattered read". Se identificaron accesos masivos a dos (2) tablas principales del proceso, que no tenían política de depuración. Por ello, se procedió a particionarlas sobre una base mensual.	Tiempo promedio de ejecución = 45 horas	Tiempo promedio de ejecución = 24 horas	Baja
21	GRUPO SAN FERNANDO Tiempos de ejecución considerables en proceso de costeo industriales (Fase 2)	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de ejecución = 24 horas	Tiempo promedio de ejecución = 12 horas	Alta
22	GRUPO SAN FERNANDO Tiempos de ejecución considerables en proceso de compras	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que ejecutaban accesos FULL SCAN TABLE. Se añadieron índices y se bajaron los costos de las sentencias de 5-6 dígitos a 1-2 dígitos.	Tiempo promedio de ejecución = 12 horas	Tiempo promedio de ejecución = 3 horas	Alta
23	GRUPO SAN FERNANDO Tiempos de respuesta degradados en aplicaciones OLTP	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando	Tiempo promedio de respuesta = 1-3 minutos	Tiempo promedio de respuesta = 1-5 segundos	Alta

		nuevos índices.			
24	HYUNDAI PERU AUTOMOTORES Tiempos de ejecución considerables en procesos contables	Afinamiento de la asignación de memoria El cuello de botella eran procesos batch que demoraban en conjunto varios días. Se aumentó la memoria del servidor y se aumentaron las áreas de caché SQL y de datos en la Base de Datos.	Tiempo total del proceso de cierre contable = 36 horas	Tiempo total del proceso de cierre contable = 8 horas	Media
25	HYUNDAI PERU AUTOMOTORES Tiempos de respuesta degradados en aplicaciones OLTP	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 1-3 minutos	Tiempo promedio de respuesta = 1-5 segundos	Alta
26	MINCETUR Tiempos de respuesta degradados en los portales Web de PROMPEX	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 1-3 minutos	Tiempo promedio de respuesta = 1-5 segundos	Alta

27	MINCETUR Tiempos de respuesta degradados en los portales Web de PROMPEX por consolidación de servidores	Afinamiento de la plataforma subyacente Al consolidar muchos servicios de Base de Datos, el cuello de botella era la escasez de recursos del servidor donde se consolidaron los servicios. Se aumentaron los procesadores y la memoria del servidor principal, y se reasignaron las áreas de caché en todas los servicios consolidados.	Tiempo promedio de respuesta = 5-10 segundos	Tiempo promedio de respuesta = 1-5 segundos	Baja
28	OFICINA DE NORMALIZACION PREVISIONAL Tiempos de cálculo inaceptables para bonos de reconocimiento tipo I	Afinamiento de operaciones de la Base de Datos El cuello de botella eran los cambios de contexto entre los programas cliente y el motor procedural de la Base de Datos, y los procedimientos almacenados y la Base de Datos. Se procedió a eliminar los cambios de contexto en los clientes moviendo el código SQL de la aplicación cliente a los procedimientos almacenados, y eliminando las llamadas a funciones privadas desde las sentencias SQL	Tiempo promedio de ejecución = 48 horas	Tiempo promedio de ejecución = 5 horas	Baja

29	OFICINA DE NORMALIZACION PREVISIONAL Tiempos de cálculo inaceptables para bonos de reconocimiento tipo II	Afinamiento del diseño de la aplicación El cuello de botella eran los accesos por un proceso a tablas muy grandes (200'000,000 de registros o más en promedio), lo que generaba inestabilidad en el proceso, y muchas de las veces se colgase y no concluyese. Se paralelizó el proceso, haciendo que varios procesos trabajen concurrentemente sobre una sector específico de la tabla.	Tiempo promedio de ejecución = 72 horas	Tiempo promedio de ejecución = 6 horas	Baja
30	OFICINA DE NORMALIZACION PREVISIONAL Tiempos de respuesta degradados en aplicaciones OLTP a nivel general luego de migración a nueva plataforma.	Afinamiento de la plataforma subyacente El cuello de botella era la prioridad de procesos de sistema operativo era asignado por éste. Uniformizando la prioridad de los procesos de sistema operativo correspondientes a la Base de Datos, a través del parámetro HPUX_SCHED_NOAGE permitió eliminar el problema.	Tiempo promedio de respuesta = 1 minuto	Tiempo promedio de respuesta = 2 segundos	Baja
31	OFICINA DE NORMALIZACION PREVISIONAL Tiempos de respuesta degradados en	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando	Tiempo promedio de respuesta = 1-3 minutos	Tiempo promedio de respuesta = 1-5 segundos	Alta

	aplicaciones OLTP	nuevos índices.			
32	TOYOTA Tiempos de respuesta degradados en aplicaciones OLTP	Afinamiento de sentencias SQL El cuello de botella eran ciertas sentencias que consumían gran parte del DB Time. Se optimizaron estos accesos puntuales reescribiendo las sentencias para aprovechar los índices disponibles o agregando nuevos índices.	Tiempo promedio de respuesta = 1-3 minutos	Tiempo promedio de respuesta = 1-5 segundos	Alta

Tabla 05-01. Muestra de Casos de Afinamiento de Rendimiento

5.1. CUADROS RESUMEN

De la muestra presentada, se han tipificado 9 tipos de afinamiento, los cuales se han clasificado en 3 categorías. Estos tipos y categorías se muestran en el cuadro 05-01.

CASOS DE AFINAMIENTO POR TIPO

Tipo de Afinamiento	Número	%	Categoría
Afinamiento de sentencias SQL	14	43.75%	SQL
Afinamiento de la plataforma subyacente	5	15.63%	Plataforma
Afinamiento del diseño de datos	3	9.38%	Diseño y Programación
Afinamiento de asignación de memoria	3	9.38%	Plataforma
Afinamiento de I/O y estructura física	2	6.25%	Plataforma
Afinamiento de operaciones de Base de Datos	2	6.25%	Diseño y Programación
Afinamiento de la contención de recursos	1	3.13%	Plataforma
Afinamiento de rutas de acceso	1	3.13%	Diseño y Programación
Afinamiento del diseño de la aplicación	1	3.13%	Diseño y Programación
Total	32	100.00%	

Cuadro 05-01. Casos de Afinamiento de Rendimiento por Tipo

La figura 05-01 muestra la frecuencia de cada tipo de afinamiento en la muestra.

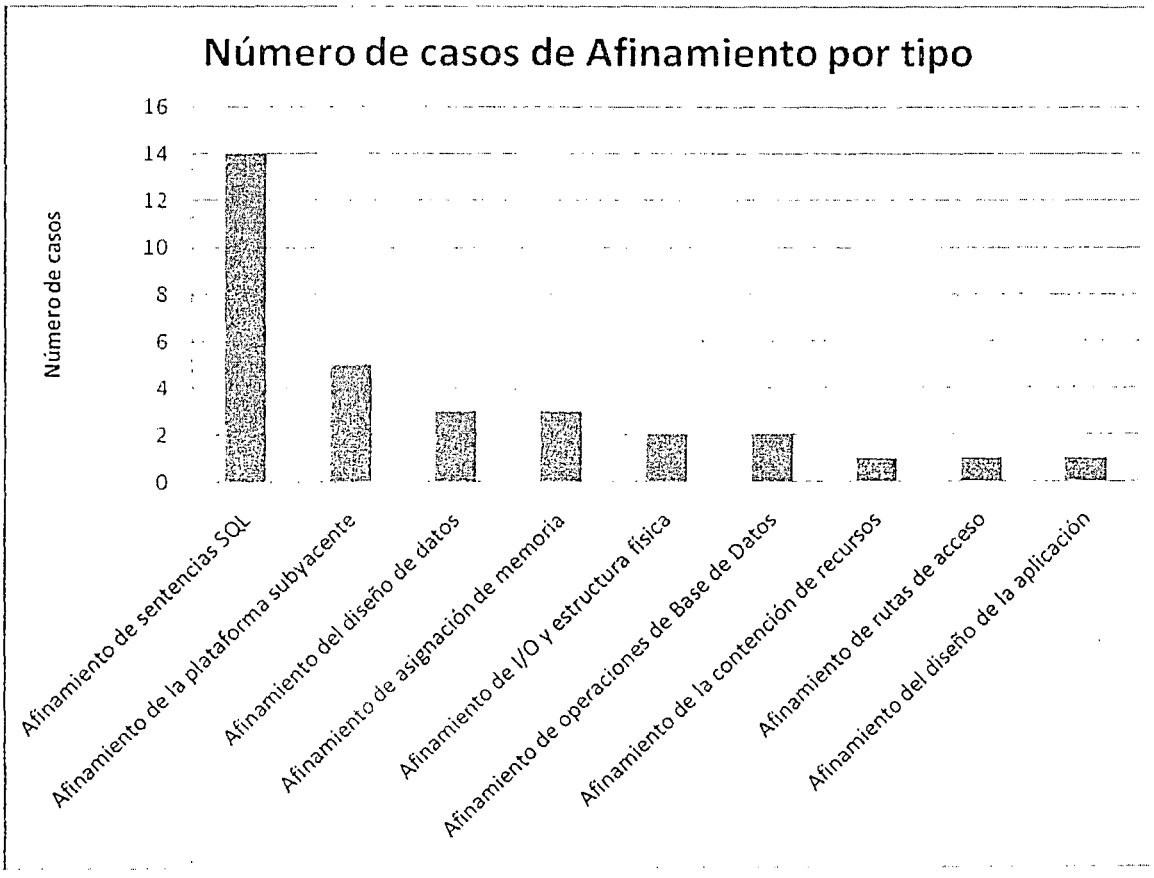


Figura 05-01. Número de Casos de Afinamiento de Rendimiento por Tipo

El cuadro 05-02 muestra un resumen de la muestra clasificada por categoría.

CASOS DE AFINAMIENTO POR CATEGORIA

Categoría	Número	%
SQL	14	43.75%
Plataforma	11	34.38%
Diseño y Programación	7	21.88%
Total	32	100.00%

Cuadro 05-02. Casos de Afinamiento de Rendimiento por Categoría

La figura 05-02 muestra la frecuencia de cada categoría en la muestra.

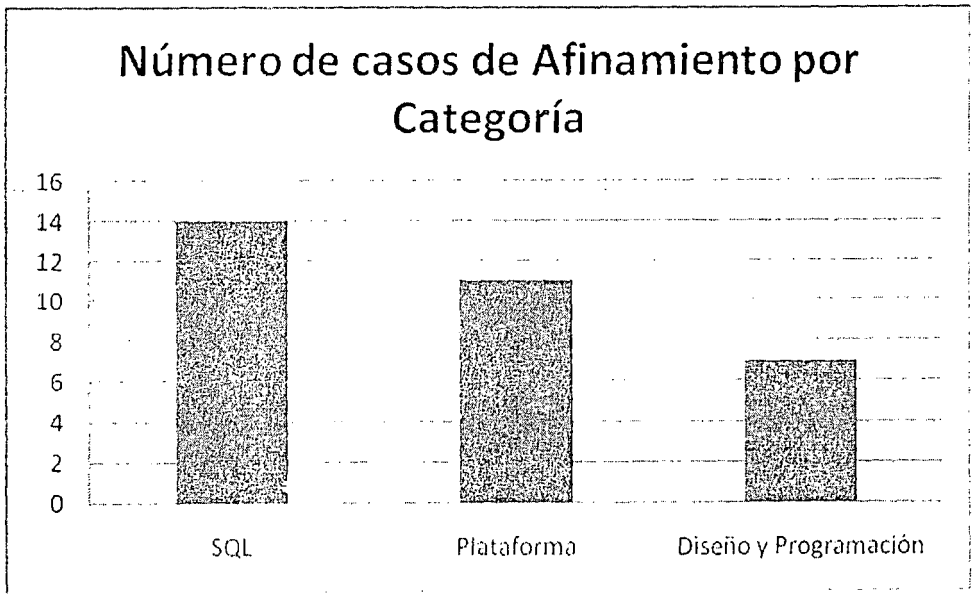


Figura 05-02. Número de Casos de Afinamiento de Rendimiento por Categoría

5.2. ANALISIS DE RESULTADOS

Del cuadro 05-02 se deduce lo siguiente:

- Los casos de afinamiento por cambios en sentencias SQL representan el 43% de la casuística. Permite ofrecer buenos resultados a un bajo costo. La dificultad radica en saber cómo reescribir las sentencias o qué nuevos índices añadir para aprovechar la oportunidad que ofrece este tipo de afinamiento.

- Los casos de afinamiento por cambios en el Diseño y la Programación representan el 22% de la casuística (Diseño de datos, Operaciones de Base de Datos, Rutas de Acceso y Diseño de la Aplicación). Permite ofrecer buenos resultados a un moderado costo, dado que la introducción de cambios al modelo de datos o en la lógica de los programas involucra un impacto mayor y un compromiso de más actores en la implementación de los cambios.

- Los casos de afinamiento por cambios en la Plataforma representan el 34% de la casuística. Permite ofrecer buenos resultados a un moderado o alto costo, dado que la introducción de este tipo de mejoras normalmente involucra la adición de nuevo hardware, o el sacrificio de ciertos componentes a favor del servidor de Base de Datos.

5.3. VALIDACION DE HIPOTESIS

Los resultados de esta muestra confirman las hipótesis formuladas en la tesis.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- La ingeniería del rendimiento debe ser considerada como una actividad omnipresente en todas las fases del ciclo de vida de un sistema de información. Con una buena base de planeamiento, las tareas de afinamiento reactivo se reducirían al mínimo, lo que influiría en menores costos de operación del sistema una vez puesto en producción.

- La correcta aplicación de metodologías y técnicas de afinamiento de rendimiento (performance tuning) impactan positivamente en los resultados económicos de un negocio, ya que permite brindar escalabilidad a una solución informática aprovechando las capacidades no utilizadas, eliminando cuellos de botella y

redistribuyendo recursos disponibles. De otra manera, para escalar una plataforma ante aumentos de demanda de carga usuaria es indispensable invertir en hardware adicional.

- El afinamiento empieza en las fases de planeamiento y diseño del sistema y continúa a través de todo el ciclo de vida del sistema. Las mayores ganancias se obtienen en la aplicación de afinamiento en las fases de planeamiento y análisis. Postergarlo hace que la obtención de mejoras y niveles deseados de rendimiento sea más costoso.

- La labor de afinamiento de sistemas es "tarea de todos" en mayor o menor grado: Desde el ejecutivo de negocio, pasando por el diseñador y el desarrollador de la aplicación, hasta los administradores de Base de Datos, de red y de sistemas.

- El establecimiento de "targets" de afinamiento es importante, porque si bien es cierto el afinamiento es una actividad continua, ellos definen resultados tangibles a ser mostrados luego de una labor de afinamiento.

- Sólo el afinamiento continuo permite mantener un sistema operando eficientemente. Los datos obtenidos en el afinamiento son claves para hacer labores de capacity planning, uso de recursos actual y registro histórico de actuación.

- Mostrar los resultados del afinamiento a la organización es un factor crítico de éxito. Sirve como herramienta

administrativa para futuros afinamientos, y como herramienta de marketing para "vender" los resultados positivos del trabajo.

RECOMENDACIONES

- La documentación de los logros del rendimiento es crucial para reusar las soluciones ya aplicadas en otras problemáticas iguales o similares que se estén presentando en otras áreas, o de cara al futuro. Es deseable asimismo contar con una base de conocimientos especializada.

- No es recomendable afinar sin haber identificado plenamente al cuello de botella. Si por ejemplo, a un servidor con problemas de rendimiento (no siendo la CPU el cuello de botella) se resuelve a priori aumentarle la velocidad a los procesadores, podríamos degradar aún más el rendimiento. Si por ejemplo, el cuello de botella real fuera el disco, aumentar la velocidad de las CPUs aumentaría aún más la contención en disco. Las CPUs procesarán instrucciones más rápido, exacerbando los ya existentes cuellos de botella en disco debido a la mayor demanda de I/O.

- Para un alto número de usuarios se recomienda migrar de arquitectura de dos (2) niveles a arquitectura multi-tier (3 o más niveles). Esto se logra inicialmente poniendo una capa intermedia de lógica de negocio entre el cliente

y el servidor de Base de Datos que tiene como objetivos distribuir el procesamiento y multiplexar las conexiones al servidor de Base de Datos. Las arquitecturas Web-enabled son las que siguen esta tendencia.

- Para aprovechar al máximo las capacidades del hardware y brindar un nivel de escalabilidad superior a la solución se debe migrar a Oracle Real Application Clusters (inicialmente conocida como Oracle Parallel Server), lo cual haría que la configuración de cluster activo/pasivo en Grupo San Fernando pase a ser cluster activo/activo, ya que existiría una instancia de Base de Datos en cada nodo accesando a la Base de Datos. Oracle RAC es la solución ad hoc para escalabilidad de aplicaciones soportadas por Bases de Datos.

- La plataforma de hardware servidora es 64-bit y el software está operando en modo 32-bit. Es necesario hacer el upgrade a 64-bit tanto en el kernel como en el software RDBMS para brindar escalibilidad adicional a la solución. Esto pasa por migrar de Oracle8i a Oracle9i por restricciones de Oracle8i en AIX 5L.

- El particionamiento de tablas e índices bien estructurado permite que la escalabilidad de la solución se mantenga constante o mejore cuando no se tiene una política de depuración de datos definida. En el Grupo San Fernando, se recomendaría aplicar particionamiento horizontal a las tablas más grandes (Partitioning Option de Oracle) de los sistemas de Inventarios y Facturación tal como se realizó en Logística para las tablas ARCOHA y ARCOHA_OPL. Los sistemas de Inventarios y

Facturación albergan las tablas más grandes de la Base de Datos.

- Para mejorar la seguridad de la información, específicamente la integridad de los datos, se debe activar en la Base de Datos las opciones DB_BLOCK_CHESKSUM y DB_BLOCK_CHECKING, que activan la seguridad contra corrupciones lógicas y físicas respectivamente. Ya se han reportado problemas de este tipo en algunos índices. Si bien es cierto que la activación de estos parámetros añaden un overhead adicional (de 1% a 10%), proporcionan un nivel de seguridad mucho mayor al comportamiento default de la Base de Datos.

- Para mejorar los accesos de I/O, se recomienda migrar de RAID-1 a RAID-0+1. RAID-1 implementa mirroring, mientras RAID-0+1 implementa stripping y mirroring. Con RAID-0+1, adicionamos stripping, que mejora los accesos a disco ya que distribuye los datos a múltiples "spindles" (ejes de disco).

- Para evitar contenciones migrar los tablespaces de DMT (dictionary managed tablespace) a LMT (locally managed tablespace). DMT implementa el mapeo de extensiones en el diccionario de datos, mientras LMT implementa las extensiones en un bitmap (mapa de bits) inserto en el mismo tablespace, liberando al diccionario de las actividades de administración de espacio y disminuyendo por ende la contención de recursos.

- Crear instancias adicionales para los esquemas (agrupaciones de tablas) independientes, lo que mejoraría la escalabilidad de la solución a través de cachés de memoria independiente para cada instancia.

- Es primordial el manejo administrativo con la parte usuaria y dentro del staff de sistemas. Con la parte usuaria el establecimiento de horarios para ciertas actividades discriminando lo que son tareas OLTP y batch. Se sugiere independizar la actividad OLTP y DSS en Bases de Datos independientes, toda vez que las actividades DSS no requieren una visión de la información up-to-date (al momento). Con el staff de sistemas eliminar los accesos a los ambientes productivos, por motivos de rendimiento y seguridad.

GLOSARIO DE TERMINOS

TÉRMINO	SIGNIFICADO
BATCH	Proceso por lote.
BOTTLENECK	Cuello de botella.
BROWSER	Programa que permite localizar y mostrar páginas Web.
CACHING	Almacenamiento en memoria intermedia de datos y código utilizados frecuentemente.
COALESCE	Desfragmentación de un segmento de Base de Datos a través de la fusión de sus extensiones libres.
CPU	Unidad Central de Procesamiento. Componente principal de un computador.
DATA	Datos, que son entrada o salida de un proceso. Plural del singular en inglés datum (dato).
DATABASE	Base de Datos.
DBA	Administrador de Base de Datos.
DSS	Sistema de Soporte a Decisiones.
DISK MIRRORING	Mantenimiento de copias idénticas de discos para mejorar la tolerancia ante fallos.
DISK STRIPPING	Distribuir la información a través de varios bloques sobre múltiples discos para mejorar el rendimiento.
DOWNSIZING	Trasladar a una plataforma de hardware de

	menor cuantía una solución informática.
GUI	Interfaz gráfica de usuario.
I/O	Entrada/Salida a disco.
IT	Tecnología de Información.
LAN	Red de área local.
MPP	Procesadores masivamente paralelos. Arquitectura donde un gran número de procesadores coordinados funcionan en conjunto.
OLTP	Proceso en transacciones en línea.
PAGING	Intercambio de páginas entre memoria y disco.
PERFORMANCE TUNING	Afinamiento de rendimiento.
RAID	Arreglo redundante de discos independientes.
RDBMS	Sistema de Administración de Base de Datos Relacional.
SCALEUP	Escalabilidad.
SDLC	Ciclo de vida del desarrollo de software.
SLA	Acuerdo de Nivel de Servicio.
SMP	Multiprocesamiento simétrico. Arquitectura donde múltiples procesadores que comparten un sistema operativo común y la memoria.
SORT	Ordenación.
SQL	Lenguaje estructurado de consultas.
SWAPPING	Intercambio de páginas entre memoria y disco, aplicado generalmente al intercambio de secciones completas de programas.
TABLESPACE	Espacio de tablas. Repositorio lógico donde residen tablas de una Base de Datos.
THROUGHPUT	Transacciones por unidad de tiempo.
TRADE-OFF	Intercambio, balance de alternativas, referido

	normalmente a decisión por comparación.
TRIGGER	Disparador. En Base de Datos, código que se ejecuta al producirse una determinada acción, por ejemplo, la ejecución de una sentencia INSERT.
WAN	Red de área ancha.

REFERENCIAS BIBLIOGRAFICAS

[AM01] White paper: An introduction to P*Software Development Life Cycle.
http://www.amendtechnologies.com/Whitepaperpdf.pdf

[FU01] Fundamentos de Base de Datos, 3era. edición
Abraham Silberschatz, Henry F. Korth, S. Sudarshan
Editado por: McGraw-Hill
ISBN: 84-481-2021-3

[GA01] Oracle Performance Tuning 101
Por Gaja Krishna Vaidyanatha, Kirtikumar Deshpande, John A. Kostelac
Editado por: McGraw-Hill Osborne Media; (May 29, 2001)
ISBN: 0072131454

[GU01] Oracle Performance Tuning, 2nd Edition
Por Mark Gurry, Peter Corrigan
2nd Edition, November 1996
Editado por O'Reilly & Associates, Inc.
ISBN: 1-56592-237-9

[HA01] Oracle SQL High-Performance Tuning (2nd Edition)
Por Guy Harrison
Editado por Prentice Hall PTR (29 de Diciembre de 2000)
ISBN: 0130123811

[HE01] Metodología de la Investigación
Por Roberto Hernández Samplieri, Carlos Fernández Collado et al
Editado por Mc Graw Hill
ISBN: 978-605-15-0291-9

[KR01] Oracle Database Administration: The Essential Reference
Por David C. Kreines, Brian Laskey
Editado por O'Reilly & Associates, Inc.
ISBN: 1-56592-516-5

[MI01] Optimizing Oracle Performance
Por Cary Millsap, With Jeff Holt
Editado por O'Reilly & Associates, Inc.
ISBN: 0-596-00527-X

[OR01] Oracle8i
Tuning
Part No. A67775-01

[OR02] Oracle Database
Performance Tuning Guide
Part No. B10752-01

[OR03] Oracle Database
2 Day + Performance Tuning Guide
Part No. B28051-03

[SC01] Oracle Performance Troubleshooting: With Dictionary Internals, SQL & Tuning Scripts (Oracle In-Focus series)
Por Robin Schumacher
Editado por Rampant Techpress (1ro. de Agosto de 2003)
ISBN: 0972751343

[TU01] Oracle Parallel Processing
Por Tushar Mahapatra, Sanjay Mishra
Editado por O'Reilly & Associates, Inc.
ISBN: 1-56592-701-X

[WI01] Definition of Locality of Reference
http://en.wikipedia.org/wiki/Locality_of_reference

[WI02] Performance Tuning
http://en.wikipedia.org/wiki/Performance_tuning

ANEXOS

ANEXO 1: DESCRIPCION FISICA DE LOS NODOS DEL CLUSTER EN GRUPO SAN FERNANDO

NODO 1

Características generales del Equipo

System Model: IBM,7038-6M2
Machine Serial Number: 10027DA
Processor Type: PowerPC_POWER4
Number Of Processors: 6
Processor Clock Speed: 1452 MHz
CPU Type: 64-bit
Kernel Type: 32-bit
LPAR Info: 1 NULL
Memory Size: 12288 MB
Good Memory Size: 12288 MB
Firmware Version: IBM,RG030909_d65e03_s
Console Login: enable
Auto Restart: false
Full Core: false

Network Information
Host Name: SFPRODUCCION

```
IP Address: 172.16.2.20
Sub Netmask: 255.255.224.0
Gateway: 172.16.1.1
Name Server:
Domain Name:

Paging Space Information
Total Paging Space: 18432MB
Percent Used: 3%
```

Volúmenes definidos sobre Discos

```
Volume Groups Information
==========================================================
===
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk0            active            546         316
98..00..00..109..109
hdisk1            active            546         473
109..105..41..109..109
==========================================================
===

vgsf1:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk6            active            271         8
00..00..00..00..08
hdisk7            active            271         9
00..00..00..00..09
hdisk8            active            271         9
00..00..00..00..09
hdisk3            active            271         9
00..00..00..00..09
```

```
hdisk4            active            271         9
00..00..00..00..09
================================================================
===

vgsf2:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk2            active            271         0
00..00..00..00..00
hdisk9            active            543         1
00..00..00..00..01
hdisk11           active            543         2
00..00..00..00..02
================================================================
===

vgsf3:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk5            active            271         0
00..00..00..00..00
================================================================
===

vgarch:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk10           active            543         0
00..00..00..00..00
hdisk13           active            543         0
00..00..00..00..00
hdisk14           active            543         13
00..00..00..00..13
================================================================
===

prueba:
```

```
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk12           active            542         542
109..108..108..108..109
=============================================================
===
```

Lista de componentes instalados:

Procesadores, Memoria, Discos, Tarjetas de Red

```
INSTALLED RESOURCE LIST

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
*   = Diagnostic support not available.

  Model Architecture: chrp
  Model Implementation: Multiple Processor, PCI bus

+ sys0              00-00             System Object
+ sysplanar0        00-00             System Planar
+ mem0              00-00             Memory
+ proc0             00-00             Processor
+ L2cache0          00-00             L2 Cache
+ proc1             00-01             Processor
+ proc2             00-02             Processor
+ proc3             00-03             Processor
+ proc6             00-06             Processor
+ proc7             00-07             Processor
* pci0              00-400000000110   PCI Bus
* isa0              02-18             ISA Bus
+ fda0              01-D1             Standard I/O Diskette Adapter
+ fd0               01-D1-00-00       Diskette Drive
* siokma0           01-K1             Keyboard/Mouse Adapte:
+ sioka0            01-K1-00          Keyboard Adapter
+ kbd0              01-K1-00-00       PS/2 keyboard
+ sioma0            01-K1-01          Mouse Adapter
+ mouse0            01-K1-01-00       3 button mouse
+ sa0               01-S1             Standard I/O Serial Port
+ tty0              01-S1-00-00       Asynchronous Terminal
+ sa1               01-S2             Standard I/O Serial Port
+ sa2               01-S3             Standard I/O Serial Port
+ sa3               01-S4             Standard I/O Serial Port
```

```
* pci1            00-400000000111   PCI Bus
* pci3            02-10             PCI Bus
+ scsi0           02-08             Wide/Ultra-3 SCSI I/O Controller
+ rmt0            02-08-00-0,0      SCSI 4mm Tape Drive (20480 MB)
+ cd0             02-08-00-1,0      16 Bit SCSI Multimedia CD-ROM
Drive
                                    (650 MB)
+ hdisk0          02-08-00-8,0      16 Bit LVD SCSI Disk Drive
(146800
                                    MB)
+ hdisk1          02-08-00-9,0      16 Bit LVD SCSI Disk Drive
(146800
                                    MB)
+ hdisk12         02-08-00-10,0     16 Bit LVD SCSI Disk Drive
(36400
                                    MB)
+ ses0            02-08-00-14,0     SCSI Enclosure Services Device
+ ses1            02-08-00-15,0     SCSI Enclosure Services Device
+ scsi1           02-09             Wide/Ultra-3 SCSI I/O Controller
* pci4            02-12             PCI Bus
+ ent0            02-08             10/100 Mbps Ethernet PCI Adapter
II
                                    (1410ff01)
* pci5            02-13             PCI Bus
* pci13           02-08             PCI Bus
+ lai1            02-00             GXT135P Graphics Adapter
* pci6            02-14             PCI Bus
+ ent1            02-08             10/100 Mbps Ethernet PCI Adapter
II
                                    (1410ff01)
* pci7            02-16             PCI Bus
+ ssa0            02-08             IBM SSA 160 SerialRAID Adapter
                                    (14109100)
* pci2            00-400000000112   PCI Bus
* pci8            02-10             PCI Bus
+ ent3            02-08             Gigabit Ethernet-SX PCI-X
Adapter
                                    (14106802)
* pci9            02-12             PCI Bus
+ ent4            02-08             Gigabit Ethernet-SX PCI-X
Adapter
                                    (14106802)
* pci11           02-14             PCI Bus
+ ent2            02-08             10/100 Mbps Ethernet PCI Adapter
II
                                    (1410ff01)
* pci12           02-16             PCI Bus
+ ssa2            02-08             IBM SSA 160 SerialRAID Adapter
```

```
                                    (14109100)
+ enclosure0       00-00-C04E       SSA Enclosure
* hdisk14          02-08-L          SSA Logical Disk Drive
+ pdisk21          02-08-C04E-13-P  SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk20          02-08-C04E-15-P  SSA160 Physical Disk Drive
(18200
                                    MB)
+ enclosure1       00-00-4506       SSA Enclosure
* hdisk2           02-08-L          SSA Logical Disk Drive
* hdisk3           02-08-L          SSA Logical Disk Drive
* hdisk4           02-08-L          SSA Logical Disk Drive
* hdisk5           02-08-L          SSA Logical Disk Drive
* hdisk6           02-08-L          SSA Logical Disk Drive
* hdisk7           02-08-L          SSA Logical Disk Drive
* hdisk8           02-08-L          SSA Logical Disk Drive
* hdisk9           02-08-L          SSA Logical Disk Drive
* hdisk10          02-08-L          SSA Logical Disk Drive
* hdisk11          02-08-L          SSA Logical Disk Drive
* hdisk13          02-08-L          SSA Logical Disk Drive
+ pdisk10          02-08-C04E-01-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk12          02-08-C04E-02-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk7           02-08-C04E-03-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk9           02-08-C04E-04-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk1           02-08-C04E-05-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk11          02-08-C04E-06-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk8           02-08-C04E-07-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk0           02-08-C04E-08-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk6           02-08-C04E-09-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk13          02-08-C04E-10-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk5           02-08-C04E-11-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk4           02-08-C04E-12-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk2           02-08-C04E-14-P  SSA160 Physical Disk Drive (9100
MB)
```

```
+ pdisk3          02-08-C04E-16-P   SSA160 Physical Disk Drive (9100
MB)
+ pdisk17         02-08-4504-01-P   SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk22         02-08-4504-02-P   SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk16         02-08-4504-04-P   SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk14         02-08-4504-05-P   SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk23         02-08-4504-06-P   SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk15         02-08-4504-08-P   SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk19         02-08-4504-09-P   SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk18         02-08-4504-12-P   SSA160 Physical Disk Drive
(18200
                                    MB)
```

NODO 2

Características generales del Equipo

```
System Model: IBM,7025-F80
Machine Serial Number: 2608149
Processor Type: PowerPC_RS64-IV
Number Of Processors: 4
Processor Clock Speed: 602 MHz
CPU Type: 64-bit
Kernel Type: 32-bit
LPAR Info: -1 NULL
Memory Size: 5120 MB
Good Memory Size: 5120 MB
```

```
Firmware Version: IBM,M2P030828_condor_
Console Login: enable
Auto Restart: false
Full Core: false

Network Information
Host Name: XPROD
IP Address: 172.16.2.21
Sub Netmask: 255.255.224.0
Gateway: 172.16.1.1
Name Server:
Domain Name:

Paging Space Information
Total Paging Space: 7040MB
Percent Used: 6%
```

Volúmenes definidos sobre Discos

```
Volume Groups Information
==============================================================
===
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk1            active            542         304
108..00..00..87..109
hdisk0            active            542         303
99..00..00..95..109
==============================================================
===

oraclevg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
```

```
hdisk4            active            542         1
00..00..00..00..01
=====================================================
===

vghist1:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk5            active            542         0
00..00..00..00..00
hdisk6            active            542         0
00..00..00..00..00
hdisk7            active            542         0
00..00..00..00..00
hdisk8            active            542         0
00..00..00..00..00
hdisk9            active            542         9
00..00..00..00..09
=====================================================
===

vghist2:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk2            active            543         2
00..00..00..00..02
=====================================================
===

vgexporta:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk10           active            543         210
76..00..00..25..109
=====================================================
===

vgindges:
```

```
PV_NAME           PV STATE          TOTAL PPs   FREE PPs
FREE DISTRIBUTION
hdisk3            active            543         223
06..00..00..108..109
hdisk11           active            543         191
00..00..00..82..109
=============================================================
===
```

Lista de componentes instalados:

Procesadores, Memoria, Discos, Tarjetas de Red

```
INSTALLED RESOURCE LIST

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
*   = Diagnostic support not available.

  Model Architecture: chrp
  Model Implementation: Multiple Processor, PCI bus

+ sys0              00-00            System Object
+ sysplanar0        00-00            System Planar
+ mem0              00-00            Memory
+ proc0             00-00            Processor
+ L2cache0          00-00            L2 Cache
* pmc0              00-00            Power Management Controller
+ proc2             00-02            Processor
+ proc4             00-04            Processor
+ proc6             00-06            Processor
* pci0              00-fff7f09000    PCI Bus
* isa0              10-80            ISA Bus
+ fda0              01-D1            Standard I/O Diskette Adapter
+ fd0               01-D1-00-00      Diskette Drive
* siokma0           01-K1            Keyboard/Mouse Adapter
+ sioka0           01-K1-00         Keyboard Adapter
+ kbd0             01-K1-00-00      PS/2 keyboard
+ sioma0           01-K1-01         Mouse Adapter
+ mouse0           01-K1-01-00      3 button mouse
+ ppa0             01-R1            CHRP IEEE1284 (ECP) Parallel
Port
                                    Adapter
```

```
+ sa0              01-S1           Standard I/O Serial Port
+ tty0             01-S1-00-00     Asynchronous Terminal
+ sa1              01-S2           Standard I/O Serial Port
+ sa2              01-S3           Standard I/O Serial Port
+ sa3              01-S4           Standard I/O Serial Port
* pci2             10-58           PCI Bus
+ scsi0            11-08           Wide/Ultra-2 SCSI I/O Controller
+ rmt0             11-08-00-0,0    4.0 GB 4mm Tape Drive
+ cd0              11-08-00-1,0    SCSI Multimedia CD-ROM Drive
(650
                                   MB)
+ hdisk0           11-08-00-4,0    16 Bit LVD SCSI Disk Drive
(18200
                                   MB)
+ scsi1            11-09           Wide/Ultra-2 SCSI I/O Controller
* pci3             10-5a           PCI Bus
* pci4             10-5c           PCI Bus
* pci5             10-5e           PCI Bus
+ ent0             1A-08           10/100 Mbps Ethernet PCI Adapter
II
                                   (1410ff01)
* pci1             00-fff7f0a000   PCI Bus
* pci6             20-58           PCI Bus
+ ent1             21-08           IBM 10/100 Mbps Ethernet PCI
Adapter
                                   (23100020)
* pci7             20-5a           PCI Bus
+ ssa0             27-08           IBM SSA 160 SerialRAID Adapter
                                   (14109100)
* pci8             20-5c           PCI Bus
* pci9             20-5e           PCI Bus
+ mg20             2D-08           GXT130P Graphics Adapter
* pci10            20-60           PCI Bus
* pci11            20-62           PCI Bus
+ scsi2            34-08           Wide/Fast-20 SCSI I/O Controller
+ hdisk1           34-08-00-2,0    16 Bit LVD SCSI Disk Drive
(18200
                                   MB)
* pci12            20-64           PCI Bus
+ ent2             37-08           IBM 10/100 Mbps Ethernet PCI
Adapter
                                   (23100020)
* pci13            20-66           PCI Bus
+ ssa1             3A-08           IBM SSA 160 SerialRAID Adapter
                                   (14109100)
+ enclosure0       00-00-C04E      SSA Enclosure
* hdisk15          3A-08-L         SSA Logical Disk Drive
* hdisk16          3A-08-L         SSA Logical Disk Drive
```

```
* hdisk17          3A-08-L          SSA Logical Disk Drive
* hdisk18          3A-08-L          SSA Logical Disk Drive
* hdisk19          3A-08-L          SSA Logical Disk Drive
* hdisk20          3A-08-L          SSA Logical Disk Drive
* hdisk21          3A-08-L          SSA Logical Disk Drive
* hdisk24          3A-08-L          SSA Logical Disk Drive
* hdisk25          27-08-L          SSA Logical Disk Drive
+ pdisk33          3A-08-C04E-13-P  SSA160 Physical Disk Drive
(18200
                                    MB)
+ pdisk32          3A-08-C04E-15-P  SSA160 Physical Disk Drive
(18200
                                    MB)
+ enclosure1       00-00-4506       SSA Enclosure
* hdisk9           27-08-L          SSA Logical Disk Drive
* hdisk10          27-08-L          SSA Logical Disk Drive
* hdisk11          27-08-L          SSA Logical Disk Drive
* hdisk22          27-08-L          SSA Logical Disk Drive
* hdisk23          27-08-L          SSA Logical Disk Drive
+ pdisk26          3A-08-C04E-01-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk28          3A-08-C04E-02-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk23          3A-08-C04E-03-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk25          3A-08-C04E-04-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk17          3A-08-C04E-05-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk27          3A-08-C04E-06-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk24          3A-08-C04E-07-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk16          3A-08-C04E-08-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk22          3A-08-C04E-09-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk29          3A-08-C04E-10-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk21          3A-08-C04E-11-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk20          3A-08-C04E-12-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk18          3A-08-C04E-14-P  SSA160 Physical Disk Drive (9100
MB)
+ pdisk19          3A-08-C04E-16-P  SSA160 Physical Disk Drive (9100
MB)
* hdisk2           27-08-L          SSA Logical Disk Drive
```

```
* hdisk3          27-08-L          SSA Logical Disk Drive
* hdisk4          27-08-L          SSA Logical Disk Drive
* hdisk5          27-08-L          SSA Logical Disk Drive
* hdisk6          27-08-L          SSA Logical Disk Drive
* hdisk7          27-08-L          SSA Logical Disk Drive
* hdisk8          27-08-L          SSA Logical Disk Drive
* hdisk12         27-08-L          SSA Logical Disk Drive
* hdisk13         27-08-L          SSA Logical Disk Drive
* hdisk14         27-08-L          SSA Logical Disk Drive
+ pdisk0          27-08-P          SSA160 Physical Disk Drive
(36400
                                   MB)
+ pdisk1          27-08-P          SSA160 Physical Disk Drive
(36400
                                   MB)
+ pdisk2          27-08-P          SSA160 Physical Disk Drive (9100
MB)
+ pdisk3          27-08-P          SSA160 Physical Disk Drive (9100
MB)
+ pdisk4          27-08-P          SSA160 Physical Disk Drive (9100
MB)
+ pdisk5          27-08-P          SSA160 Physical Disk Drive (9100
MB)
+ pdisk6          27-08-P          SSA160 Physical Disk Drive (9100
MB)
+ pdisk7          27-08-P          SSA160 Physical Disk Drive (9100
MB)
+ pdisk8          27-08-P          SSA160 Physical Disk Drive
(36400
                                   MB)
+ pdisk9          27-08-P          SSA160 Physical Disk Drive
(36400
                                   MB)
+ pdisk30         27-08-P          SSA160 Physical Disk Drive
(36400
                                   MB)
+ pdisk31         27-08-P          SSA160 Physical Disk Drive
(36400
                                   MB)
+ pdisk13         27-08-4504-01-P  SSA160 Physical Disk Drive
(18200
                                   MB)
+ pdisk34         27-08-4504-02-P  SSA160 Physical Disk Drive
(18200
                                   MB)
+ pdisk12         27-08-4504-04-P  SSA160 Physical Disk Drive
(18200
                                   MB)
```

```
+ pdisk10          27-08-4504-05-P   SSA160 Physical Disk Drive
(18200
                                     MB)
+ pdisk35          27-08-4504-06-P   SSA160 Physical Disk Drive
(18200
                                     MB)
+ pdisk11          27-08-4504-08-P   SSA160 Physical Disk Drive
(18200
                                     MB)
+ pdisk15          27-08-4504-09-P   SSA160 Physical Disk Drive
(18200
                                     MB)
+ pdisk14          27-08-4504-12-P   SSA160 Physical Disk Drive
(18200
                                     MB)
```

ANEXO 2: PARTICIONAMIENTO DE INDICES DE LAS TABLAS ARCOHA Y ARCOHA_OPL

- Para los índices de la tabla ARCOHA:

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'IX_ARCOHA_01'
 order by 1
```

PARTITION_NAME	HIGH_VALUE
ARCOHA_P200209	'01', 2002, 10
ARCOHA_P200210	'01', 2002, 11
ARCOHA_P200211	'01', 2002, 12
ARCOHA_P200212	'01', 2003, 01
ARCOHA_P200301	'01', 2003, 02
ARCOHA_P200302	'01', 2003, 03
ARCOHA_P200303	'01', 2003, 04
ARCOHA_P200304	'01', 2003, 05
ARCOHA_P200305	'01', 2003, 06
ARCOHA_P200306	'01', 2003, 07
ARCOHA_P200307	'01', 2003, 08
ARCOHA_P200308	'01', 2003, 09
ARCOHA_P200309	'01', 2003, 10
ARCOHA_P200310	'01', 2003, 11
ARCOHA_P200311	'01', 2003, 12
ARCOHA_P200312	'01', 2004, 01

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'IX_ARCOHA_02'
 order by 1
```

PARTITION_NAME	HIGH_VALUE
ARCOHA_P200209	'01', 2002, 10
ARCOHA_P200210	'01', 2002, 11
ARCOHA_P200211	'01', 2002, 12
ARCOHA_P200212	'01', 2003, 01
ARCOHA_P200301	'01', 2003, 02
ARCOHA_P200302	'01', 2003, 03
ARCOHA_P200303	'01', 2003, 04
ARCOHA_P200304	'01', 2003, 05
ARCOHA_P200305	'01', 2003, 06
ARCOHA_P200306	'01', 2003, 07
ARCOHA_P200307	'01', 2003, 08
ARCOHA_P200308	'01', 2003, 09
ARCOHA_P200309	'01', 2003, 10
ARCOHA_P200310	'01', 2003, 11
ARCOHA_P200311	'01', 2003, 12
ARCOHA_P200312	'01', 2004, 01

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'PK_ARCOHA_01'
 order by 1
```

PARTITION_NAME	HIGH_VALUE
ARCOHA_P200209	'01', 2002, 10
ARCOHA_P200210	'01', 2002, 11
ARCOHA_P200211	'01', 2002, 12
ARCOHA_P200212	'01', 2003, 01
ARCOHA_P200301	'01', 2003, 02
ARCOHA_P200302	'01', 2003, 03
ARCOHA_P200303	'01', 2003, 04
ARCOHA_P200304	'01', 2003, 05
ARCOHA_P200305	'01', 2003, 06
ARCOHA_P200306	'01', 2003, 07
ARCOHA_P200307	'01', 2003, 08
ARCOHA_P200308	'01', 2003, 09
ARCOHA_P200309	'01', 2003, 10
ARCOHA_P200310	'01', 2003, 11
ARCOHA_P200311	'01', 2003, 12
ARCOHA_P200312	'01', 2004, 01

- Para los índices de la tabla ARCOHA_OPL:

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'IX_ARCOHA_OPL_01'
 order by 1
```

PARTITION_NAME	HIGH_VALUE
ARCOHA_OPL_P200209	'01', 2002, 10
ARCOHA_OPL_P200210	'01', 2002, 11
ARCOHA_OPL_P200211	'01', 2002, 12
ARCOHA_OPL_P200212	'01', 2003, 01
ARCOHA_OPL_P200301	'01', 2003, 02
ARCOHA_OPL_P200302	'01', 2003, 03
ARCOHA_OPL_P200303	'01', 2003, 04
ARCOHA_OPL_P200304	'01', 2003, 05
ARCOHA_OPL_P200305	'01', 2003, 06
ARCOHA_OPL_P200306	'01', 2003, 07
ARCOHA_OPL_P200307	'01', 2003, 08
ARCOHA_OPL_P200308	'01', 2003, 09
ARCOHA_OPL_P200309	'01', 2003, 10
ARCOHA_OPL_P200310	'01', 2003, 11
ARCOHA_OPL_P200311	'01', 2003, 12
ARCOHA_OPL_P200312	'01', 2004, 01

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'IX_ARCOHA_OPL_02'
 order by 1
```

PARTITION_NAME	HIGH_VALUE
ARCOHA_OPL_P200209	'01', 2002, 10
ARCOHA_OPL_P200210	'01', 2002, 11
ARCOHA_OPL_P200211	'01', 2002, 12
ARCOHA_OPL_P200212	'01', 2003, 01
ARCOHA_OPL_P200301	'01', 2003, 02
ARCOHA_OPL_P200302	'01', 2003, 03
ARCOHA_OPL_P200303	'01', 2003, 04
ARCOHA_OPL_P200304	'01', 2003, 05
ARCOHA_OPL_P200305	'01', 2003, 06
ARCOHA_OPL_P200306	'01', 2003, 07
ARCOHA_OPL_P200307	'01', 2003, 08
ARCOHA_OPL_P200308	'01', 2003, 09
ARCOHA_OPL_P200309	'01', 2003, 10

```
ARCOHA_OPL_P200310              '01', 2003, 11
ARCOHA_OPL_P200311              '01', 2003, 12
ARCOHA_OPL_P200312              '01', 2004, 01
```

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'IX_ARCOHA_OPL_03'
 order by 1
```

```
PARTITION_NAME                  HIGH_VALUE
------------------------------- --------------------
ARCOHA_OPL_P200209              '01', 2002, 10
ARCOHA_OPL_P200210              '01', 2002, 11
ARCOHA_OPL_P200211              '01', 2002, 12
ARCOHA_OPL_P200212              '01', 2003, 01
ARCOHA_OPL_P200301              '01', 2003, 02
ARCOHA_OPL_P200302              '01', 2003, 03
ARCOHA_OPL_P200303              '01', 2003, 04
ARCOHA_OPL_P200304              '01', 2003, 05
ARCOHA_OPL_P200305              '01', 2003, 06
ARCOHA_OPL_P200306              '01', 2003, 07
ARCOHA_OPL_P200307              '01', 2003, 08
ARCOHA_OPL_P200308              '01', 2003, 09
ARCOHA_OPL_P200309              '01', 2003, 10
ARCOHA_OPL_P200310              '01', 2003, 11
ARCOHA_OPL_P200311              '01', 2003, 12
ARCOHA_OPL_P200312              '01', 2004, 01
```

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'IX_ARCOHA_OPL_04'
 order by 1
```

```
PARTITION_NAME                  HIGH_VALUE
------------------------------- --------------------
ARCOHA_OPL_P200209              '01', 2002, 10
ARCOHA_OPL_P200210              '01', 2002, 11
ARCOHA_OPL_P200211              '01', 2002, 12
ARCOHA_OPL_P200212              '01', 2003, 01
ARCOHA_OPL_P200301              '01', 2003, 02
ARCOHA_OPL_P200302              '01', 2003, 03
ARCOHA_OPL_P200303              '01', 2003, 04
ARCOHA_OPL_P200304              '01', 2003, 05
ARCOHA_OPL_P200305              '01', 2003, 06
ARCOHA_OPL_P200306              '01', 2003, 07
ARCOHA_OPL_P200307              '01', 2003, 08
ARCOHA_OPL_P200308              '01', 2003, 09
```

```
ARCOHA_OPL_P200309              '01', 2003, 10
ARCOHA_OPL_P200310              '01', 2003, 11
ARCOHA_OPL_P200311              '01', 2003, 12
ARCOHA_OPL_P200312              '01', 2004, 01
```

```
select partition_name, high_value
  from dba_ind_partitions
 where index_name = 'IX_ARCOHA_OPL_05'
 order by 1
```

```
PARTITION_NAME                  HIGH_VALUE
------------------------------- ----------------------
ARCOHA_OPL_P200209              '01', 2002, 10
ARCOHA_OPL_P200210              '01', 2002, 11
ARCOHA_OPL_P200211              '01', 2002, 12
ARCOHA_OPL_P200212              '01', 2003, 01
ARCOHA_OPL_P200301              '01', 2003, 02
ARCOHA_OPL_P200302              '01', 2003, 03
ARCOHA_OPL_P200303              '01', 2003, 04
ARCOHA_OPL_P200304              '01', 2003, 05
ARCOHA_OPL_P200305              '01', 2003, 06
ARCOHA_OPL_P200306              '01', 2003, 07
ARCOHA_OPL_P200307              '01', 2003, 08
ARCOHA_OPL_P200308              '01', 2003, 09
ARCOHA_OPL_P200309              '01', 2003, 10
ARCOHA_OPL_P200310              '01', 2003, 11
ARCOHA_OPL_P200311              '01', 2003, 12
ARCOHA_OPL_P200312              '01', 2004, 01
```