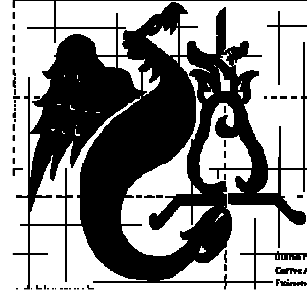




**Universidad Nacional
Autónoma de México**

**Facultad de
Estudios Profesionales
Acatlán**



**Tesis para obtener el título de Licenciado en Matemáticas
Aplicadas y Computación**

*Sistema de Monitoreo de Desempeño de Servidores en ambientes
SAP R3*

Presenta: Arturo de la Rosa Rosales

Número de Cuenta: 8935408-3



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicatoria:

*A la memoria de mi padre – quien con el ejemplo me enseñó como ser perseverante, el valor más importante que tengo
A mi madre – por todo el cariño, atención y motivación que me ha dado durante toda mi vida.*

Agradecimientos:

A la UNAM y a mi país, gracias por permitirme tener acceso al valor de la educación profesional.

Maria del Carmen Villar, mi asesora quien tuvo la paciencia de apoyarme por 7 años en mis bosquejos de tesis siempre con una sonrisa.

Marco Antonio Becerril, Rubén Rodea, Jaime Aguilar, Juan Cazares, Paola Lili, Isabel Ángeles y Jaime Castro, excelentes compañeros de trabajo quienes compartieron una gran cantidad conocimientos aplicados a este trabajo.

Ina Iliana, mi esposa por compartir los éxitos y fracasos de mi vida.

<i>Introducción</i>	<i>1</i>
<i>1 SAP R/3</i>	<i>3</i>
<i>1.1 SAP R/3, el sistema ERP de SAP</i>	<i>3</i>
<i>1.2 Historia del R/3</i>	<i>3</i>
<i>1.3 El módulo base</i>	<i>5</i>
<i>1.4 Esquema cliente / servidor de R/3</i>	<i>6</i>
<i>1.4.1 Servidor de presentación SAPGUI</i>	<i>6</i>
<i>1.4.2 Capa de presentación</i>	<i>7</i>
<i>1.5 Ámbito de los sistema SAP R3 (landscape)</i>	<i>7</i>
<i>1.6 Arquitectura interna de la instancia R/3</i>	<i>8</i>
<i>1.6.1 Despachador (Dispatcher)</i>	<i>8</i>
<i>1.6.2 Procesos de trabajo en R/3 (Work Processes)</i>	<i>9</i>
<i>1.6.3 Proceso de acceso al R/3 (Balanceo de Cargas)</i>	<i>10</i>
<i>1.6.4 Paso de diálogo (Dialog Step)</i>	<i>11</i>
<i>1.6.5 Memoria R/3</i>	<i>13</i>
<i>1.7 Importancia del desempeño de sistemas R3</i>	<i>14</i>
<i>1.8 Afinación del desempeño de sistemas R3</i>	<i>17</i>
<i>1.8.1 Afinación técnica</i>	<i>18</i>
<i>1.8.2 Afinación de la aplicación</i>	<i>18</i>
<i>2 Utilidad de un Sistema de Monitoreo de Desempeño de Servidores (SMDS)</i>	<i>19</i>
<i>2.1 Dimensionamiento de un servidor de aplicación</i>	<i>19</i>
<i>2.2 Capacidad de procesamiento de los servidores.</i>	<i>21</i>
<i>2.3 Factores que pueden saturar la utilización de un servidor.</i>	<i>22</i>
<i>2.4 La complejidad de analizar el desempeño de los servidores.</i>	<i>23</i>
<i>2.5 Herramientas de análisis de desempeño de servidores en Unix</i>	<i>25</i>
<i>2.5.1 sar</i>	<i>25</i>
<i>2.5.2 vmstat</i>	<i>25</i>
<i>2.5.3 iostat</i>	<i>26</i>
<i>2.5.4 Carga generada por comandos de monitoreo de sistema</i>	<i>27</i>
<i>3 Construcción del sistema de monitoreo de desempeño de servidores.</i>	<i>28</i>
<i>3.1 Herramientas de construcción</i>	<i>29</i>
<i>3.2 Recolección y almacenamiento de métricas de desempeño</i>	<i>30</i>

3.2.1	<i>Iniciador de colectores.</i>	31
3.2.2	<i>Ejecutor de comandos</i>	32
3.2.3	<i>Plantillas utilizadas por el ejecutor de comandos</i>	34
3.2.4	<i>Creación del usuario - esquema en Oracle para almacenar las métricas de desempeño.</i>	34
3.2.5	<i>Función carga_datos.</i>	35
3.2.6	<i>Creación de las tablas a almacenar las métricas.</i>	35
3.3	<i>Ejemplos de plantillas de programas shells para coleccionar las métricas de desempeño de servidores UNIX.</i>	36
3.3.1	<i>Colector de utilización de CPU (cpu.ksh)</i>	36
3.3.2	<i>Colector de estado de la memoria virtual (vmstat.ksh)</i>	38
3.3.3	<i>Colector del desempeño y utilización de discos en Solaris (disk.ksh)</i>	41
3.3.4	<i>Colector de rendimiento de discos HP-UX-Solaris (sard.ksh)</i>	42
3.4	<i>Ejemplos de plantillas de programas shells para coleccionar el desempeño de servidores Oracle.</i>	43
3.4.1	<i>Colector de eventos de espera de la base de datos (System_event.ksh)</i>	43
3.4.2	<i>Colector de estadísticas de uso y rendimiento de datafiles (datafiles.ksh)</i>	45
3.5	<i>Monitores de desempeño</i>	46
3.5.1	<i>Presentación de datos sobre el tiempo y de manera global (todos los servidores de aplicación)</i>	46
3.5.2	<i>Incorporación de alarmas y advertencias de rendimientos</i>	49
3.5.3	<i>Monitor rendimiento de cpu (vmstat)</i>	50
3.5.4	<i>Monitor rendimiento de cpu (sar -u y sar -q)</i>	53
3.5.5	<i>Monitor de desempeño de memoria</i>	54
3.5.6	<i>Construcción del mapa de discos Solaris-Volume Manager-EMC2</i>	57
3.5.7	<i>Mapa de discos HP-UX-Volume Manager - XP</i>	60
3.5.8	<i>Monitor de utilización de discos Top 15</i>	62
3.5.9	<i>Monitor de discos críticos (Hot Disks Top 15)</i>	64
3.5.10	<i>Monitor de desempeño del servidor de base de datos</i>	66
3.5.11	<i>Resumen de los 10 eventos Top de Oracle</i>	68
3.5.12	<i>Los datafiles más accesados (datafiles_oracle_top30.ksh)</i>	70
3.5.13	<i>Los sapdatas más accesados (sapdatas_oracle_top30.ksh)</i>	71
4	<i>Análisis de desempeño de SAP R3</i>	72
4.1	<i>Análisis de tiempo de respuesta y carga de trabajo (workload analysis).</i>	72

4.1.1 Tiempo de respuesta	72
4.1.2 ST03 transacción de análisis de tiempo de respuesta y carga de trabajo (Workload Analsys)	74
4.1.3 Tiempo de respuesta por transacciones	75
4.1.4 Tiempo de respuesta por usuario	77
4.1.5 Análisis de procesos en tiempo real	78
4.2 Desempeño de la memoria de R/3	80
4.2.1 Memoria compartida y local de R/3	80
4.2.2 Historia del comportamiento de la memoria y modificación de parámetros	82
4.2.3 Usuarios y procesos con mayor consumo de memoria	83
4.2.4 Transacciones con mayor consumo de memoria	84
4.2.5 Uso del programa memlimits	85
4.3 Desempeño de la base de datos (Oracle).	87
4.3.1 Optimizador de costos	87
4.3.2 Buffer de datos	88
4.3.3 Shared pool	90
4.3.4 Sentencias SQL costosas	92
4.4 Desempeño de hardware.	96
4.4.1 Procesos con mayor consumo de CPU	96
4.4.2 Interfaces de RED	97
4.4.3 Verificar la red vía ping	98
5 Diagrama de flujo con las acciones de afinación requeridas.	99
5.1 Diagrama análisis de CPU	100
5.1.1 Monitor de CPU por día	100
5.1.2 Monitor de CPU por hora	101
5.2 Diagrama de análisis de memoria.	102
5.2.1 Monitor global de memoria por día.	102
5.2.2 Monitor global de memoria por hora.	103
5.3 Diagrama de análisis de rendimiento de disco	104
5.4 Diagrama de análisis LAN	107
5.5 Diagrama de análisis WAN	108
6 Casos prácticos	109
6.1 Saturación de CPU por trabajos del sistema inadecuadamente configurados	109

<i>6.2 Problemas de contención en disco por procesos externos</i>	<i>113</i>
<i>6.3 Redistribución de carga de trabajo para evitar paginación.</i>	<i>118</i>
<i>6.4 Deficiencias en el enlace.</i>	<i>122</i>
<i>6.5 Problemas en la red de área local (LAN)</i>	<i>123</i>
<i>6.6 Problemas por inadecuada configuración en disco</i>	<i>125</i>
<i>Conclusiones</i>	<i>132</i>
<i>Aplicación de conocimientos adquiridos en la carrera en el presente trabajo</i>	<i>133</i>
<i>Derivación de trabajos sobre el tema para un alumno de MAC</i>	<i>134</i>
<i>Bibliografía</i>	<i>136</i>
<i>Documentos electrónicos</i>	<i>136</i>

Introducción

Cuando una empresa decide implantar SAP R3 el éxito de esta inversión de millones de dólares dependerá de varios factores:

- Compromiso directivo para impulsar el proyecto
- Administración del proyecto
- Planeación de recursos
- Adecuada infraestructura tecnológica.

Las expectativas de la empresa sobre la aplicación son muy altas, principalmente sobre el incremento de la productividad y la eficiencia dentro de la organización. La importancia de una adecuada infraestructura tecnológica toma relevancia en este sentido, los usuarios son motivados a trabajar y lo podrán realizar eficientemente si el sistema tiene un adecuado desempeño¹. Si este no cubre esta expectativa se cuestionará la viabilidad del proyecto poniéndolo en riesgo de fracasar en su implementación.

Cuando existen problemas de desempeño el primer cuestionamiento hacia los responsables de la infraestructura tecnológica es ***¿tenemos la suficiente capacidad de procesamiento en los servidores?***

Los servidores donde se planea ejecutar el sistema SAP R3 cuentan con una capacidad limitada de recursos, esta capacidad es dimensionada al principio de un proyecto de acuerdo a la carga esperada. Una vez que el sistema se encuentra productivo el administrador de SAP R3 es el responsable de analizar el desempeño del servidor y recomendar las tareas de afinación para mejorar su rendimiento. Por esta razón ***el administrador debe tener fundamentos adecuados que respalden su análisis sobre la capacidad de los servidores, estos fundamentos son datos estadísticos del desempeño del servidor.***

El presente trabajo de tesis se enfoca al desarrollo de un Sistema de Monitoreo de Desempeño de Servidores en Sistemas SAP R3 (SMDS) y de su utilización para determinar si se cuenta con suficiente capacidad de procesamiento para la ejecución de sistemas R3 sobre ambientes UNIX (Solaris y HP-UX) / Oracle.²

El trabajo se encuentra ordenado de la siguiente manera:

¹ En el ámbito profesional de México se utiliza el término inglés *performance* para expresar el desempeño o rendimiento de un sistema, sin embargo en este trabajo me voy a referir a esto con el término en español: ***desempeño.***

² Las razones de esto es que los sistemas que administro en la actualidad se encuentran en esta plataforma. Adicionalmente, en México las instalaciones UNIX-Oracle son las más frecuentes. Aunque las mayores instalaciones de SAP R3 en México (Telmex, PEMEX, CFE) se encuentran en UNIX-Informix, estas se encuentran en proceso de migración hacia UNIX-Oracle.

El primer capítulo presenta el contexto del trabajo: SAP R3, la historia y filosofía de SAP como empresa desarrolladora de software y las características técnicas de su sistema de planeación de recursos empresariales (ERP) el R3.

En el capítulo 2 se justifica la necesidad de monitorear el desempeño de los servidores donde se ejecutará SAP R3 para responder la pregunta sobre si existe suficiente capacidad de procesamiento. Basado en este análisis se define las características del Sistema de Monitoreo de Desempeño de Servidores (SMDS).

El capítulo 3 presenta el diseño de los componentes del SMDS así como la utilización de las comandos estándares de UNIX para medir el rendimiento de servidores.

El cuarto capítulo muestra las herramientas de SAP R3 para monitorear el desempeño de la aplicación, esto es requerido para determinar la afinación necesaria para corregir un problema de desempeño.

En el capítulo 5 se presentan las rutas de análisis para determinar la acción de afinación requerida de acuerdo al problema de desempeño encontrado CPU, memoria, disco o red.

El último capítulo se presentan casos prácticos en el que se ha utilizado el sistema para analizar la capacidad de los servidores así como el resultado de realizar afinaciones tanto en la aplicación como en la capacidad del equipo.

1 SAP R/3

1.1 SAP R/3, el sistema ERP de SAP

SAP son las siglas del término en inglés: Systems, Applications and Products in data processing. Fue fundada en 1972 por cuatro ex empleados de IBM en Walldorf, Alemania. Actualmente cuenta con un portafolio de soluciones enfocadas a diferentes áreas de aplicación empresarial: R3 suite de negocios integral (ERP), APO sistema de planeación y optimización, BW sistema datawarehouse y aplicaciones enfocadas a industrias específicas embotelladoras, gas, electricidad, petróleo, automotrices, etc.

El R3 es el sistema de planeación de recursos empresariales (ERP)¹ de SAP. Un ERP es un software diseñado para integrar a todos los departamentos y sus funciones dentro de una compañía en un solo sistema de cómputo con una base de datos única. Incluye funciones de planeación de producción, administración de inventarios, contabilidad, finanzas, interacción con proveedores y clientes, así como control y seguimiento de acciones.

El sistema R/3 está dirigido a la mayoría de las industrias: manufactura, distribución, petróleo, electricidad, salud, farmacéutica, banca, seguros, telecomunicaciones, transporte, química, etc.

1.2 Historia del R/3

En los años setenta, los sistemas de información eran generados de manera local y personalizada inicialmente sobre áreas críticas como la contabilidad y finanzas.

Posteriormente se introdujeron aplicaciones para áreas más complejas como la distribución y producción. Algunas compañías que realizaban este tipo de desarrollos personalizados identificaron la oportunidad de desarrollar software empresarial que pudiera ser utilizado por diferentes compañías. Sin embargo, estas compañías encontraron diferentes obstáculos para comercializar este tipo de software entre los que destacan:

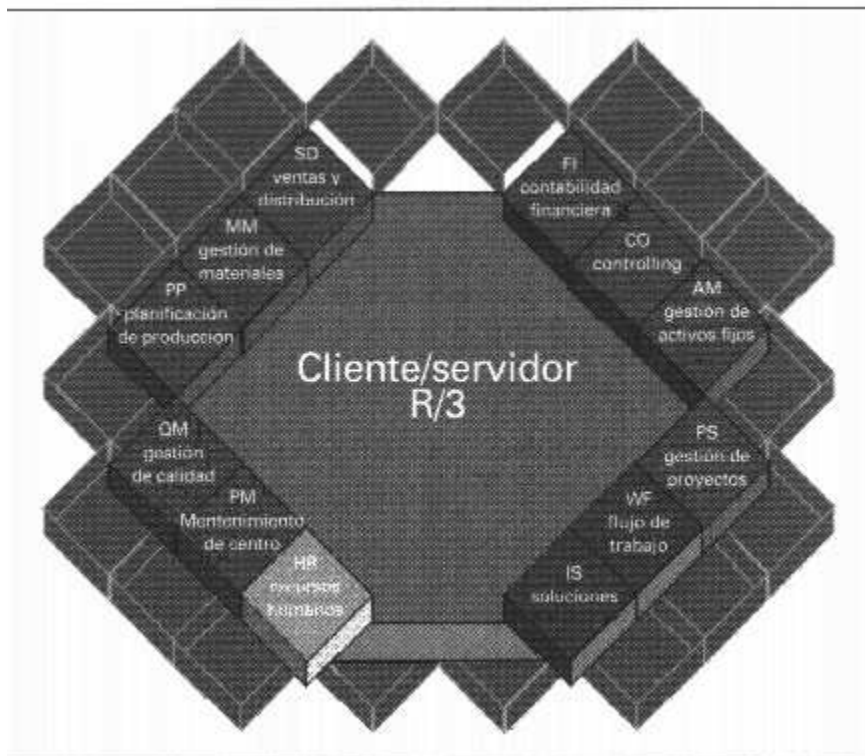
- La dependencia del hardware y software donde fueron desarrollados.
- Las compañías no se comportaban de forma estándar (el ejemplo más crítico es el cálculo de nómina).

¹ ERP por la siglas en inglés de Enterprise Resource Planning

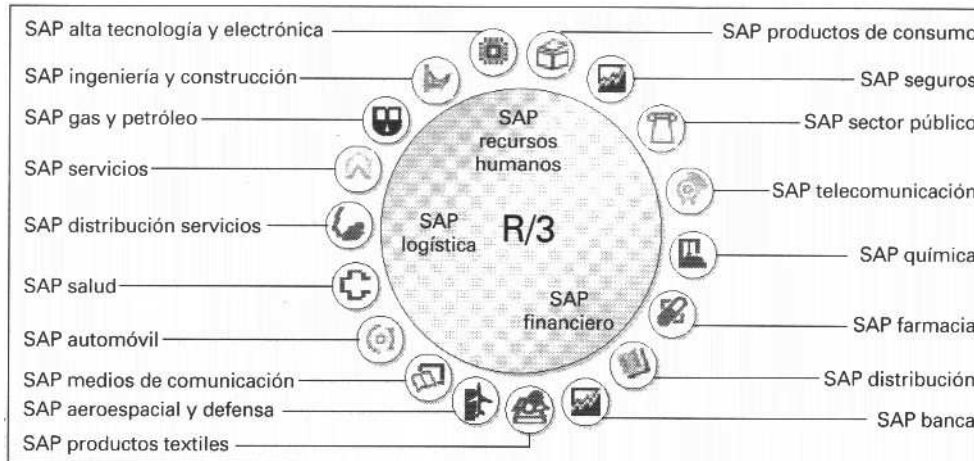
Durante los años 80's, con el surgimiento de sistemas abiertos, fue posible librar el obstáculo de la dependencia de la plataforma. La aparición de sistemas UNIX y Windows NT como sistemas operativos abiertos y base de datos como Oracle, Informix, Sybase y DB2, han permitido que las aplicaciones sean instaladas en diferentes plataformas de con procedimientos de operación estándares así como compartir información entre cada una de ellas.

Originalmente SAP tenía el producto R/2, el cuál cubría adecuadamente las necesidades de diferentes tipos de negocio en distintos países y áreas de aplicación, este producto estuvo disponible desde los años 70's. R/2 funcionaba sobre equipos mainframe con la suficiente flexibilidad para adaptarse a la forma de operar de diferentes compañías. Al contar con plataformas abiertas SAP evoluciono este producto denominándolo R/3. Esta versión tuvo la madurez necesaria para convencer a las compañías para implantarlo y abandonar el esquema de desarrollo a la medida que a largo plazo es más costoso.

La siguiente figura muestra los módulos funcionales que integran la solución de SAP R3.

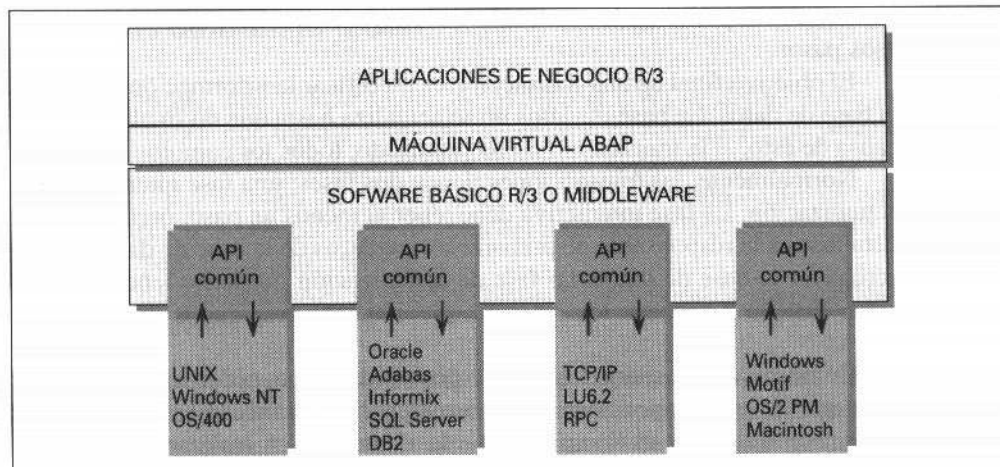


Con el éxito del R/3, SAP se enfocó a desarrollar aplicaciones de negocio específicas. Así SAP evolucionó de ser una solución global independiente del cliente a una solución de área de negocio específica.



1.3 El módulo base

Una clave del éxito del R/3 es la portabilidad y escalabilidad, esta flexibilidad la logra por medio de una capa intermedia (middleware) denominado módulo base (BASIS).



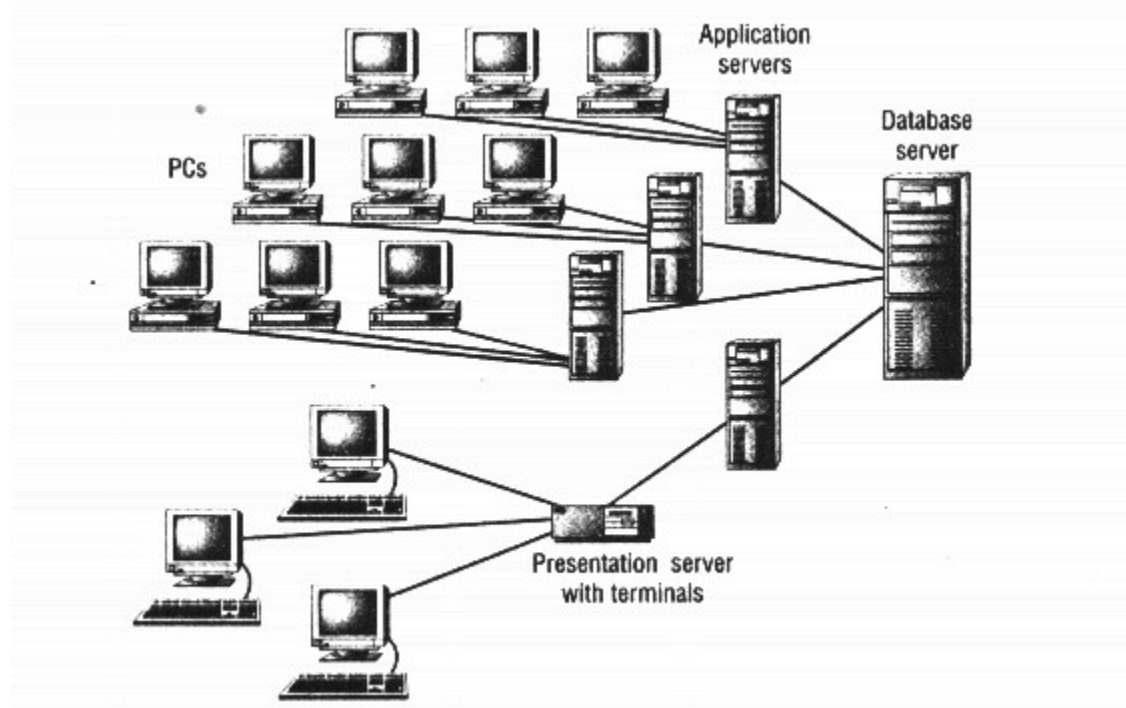
El módulo base es un conjunto de programas y herramientas que se relacionan con el sistema operativo, la base de datos y los protocolos e interfaces de comunicación.

Este software permite a las aplicaciones R/3 (FI Finanzas, CO Contabilidad, SD Ventas y Distribución, etc.) tener la misma funcionalidad y trabajar de la misma manera sin importar el sistema operativo ni la base de datos donde se instale.

El diseño del módulo está desarrollado sobre estándares de la industria: lenguajes ANSI-C y C++ para el entorno de ejecución (runtime), OPEN SQL para acceso a bases de datos, TCP/IP para las comunicaciones y las interfases gráficas como Windows, Motif o Macintosh.

1.4 Esquema cliente / servidor de R/3

El sistema R/3 tiene una arquitectura cliente - servidor de tres capas.



1. El servidor de base de datos es usado para administrar los datos de la aplicación (almacenamiento, lecturas, escrituras, manejo de consistencia, integridad, etc.).
2. El procesamiento de datos y los cálculos que requiere la aplicación ocurre en los servidores de aplicación.
3. La capa de presentación esta compuesta por los servidores de presentación (frontends SAP GUI), en estos se realiza la entrada y salida de datos para interactuar con el usuario final.

1.4.1 Servidor de presentación SAPGUI

Un servidor de presentación es normalmente configurado como una PC. En estos se instala el cliente SAP conocido como SAPGUI (Interfaz Gráfica de Usuario SAP) el cuál puede ejecutarse en diferentes plataformas: Windows, Macintosh, UNIX y JAVA. El cliente en todas las plataformas ofrece una presentación uniforme y operación consistente.

1.4.2 Capa de presentación

La capa de presentación en R/3 es escalable, permite tener varias computadoras trabajando en conjunto y balanceando las cargas de trabajo para optimizar el rendimiento del sistema. El administrador puede configurar tantos servidores de aplicación requiera para satisfacer las transacciones generadas en el sistema.

Para comprender esta escalabilidad es necesario definir los conceptos siguientes.

- **Instancia R/3.** Es un grupo de servicios los cuáles son iniciados y detenidos de manera conjunta. Usualmente, una instancia es un despachador con sus procesos de trabajo.
- **Instancia Central R/3.** Es instancia de R/3 que se caracteriza por contar con el proceso de encolamiento (enque).
- **Servidor de aplicación R/3.** Es una computadora donde una o más instancias R/3 se encuentran ejecutándose.
- **Sistema R/3.** Un sistema R/3 consiste en una o más instancia R/3 ejecutándose en una o más servidores de aplicación R/3. Toda la capa de presentación del R/3 es el conjunto de los 3 conceptos anteriores.
- **Servidor de base de datos.** Los RDBMS² son utilizados por R/3 como repositorio, existe solo un servidor de base de datos y sobre este reside la administración de los datos del R/3.

1.5 Ámbito de los sistema SAP R3 (landscape)

SAP recomienda instalar 3 sistemas R3 para asegurar la consistencia de un sistema productivo. Estos sistemas estarán orientados a tareas de desarrollo y aseguramiento de calidad de las configuraciones de los proyectos realizados:

Desarrollo (DEV): Sistema de trabajo donde se realizará la parametrización del sistema y desarrollo de programas.

Aseguramiento de Calidad (QAS): Sistema destinado a realizar pruebas de la parametrización del sistema y de los nuevos desarrollos con datos masivos y reales por parte de los usuarios clave funcionales.

Producción (PRD): Sistema utilizado para la producción diaria el cuál ha sido configurado y desarrollado con los estándares de SAP de aseguramiento de calidad.

² Sistema Relacional de Administración de Base de Datos, por sus siglas en inglés Relational Database Management System, los RDBMS compatibles con SAP son Oracle, Informix, DB2, SQLServer y SAPDB (Adabas)

Este estándar recomendado implica tener servidores de aplicación adicionales e independientes de los utilizados en producción. Uno de estos servidores será utilizado para realizar el SMDS propuesto en este trabajo evitando así generar carga adicional en un sistema productivo.

1.6 Arquitectura interna de la instancia R/3

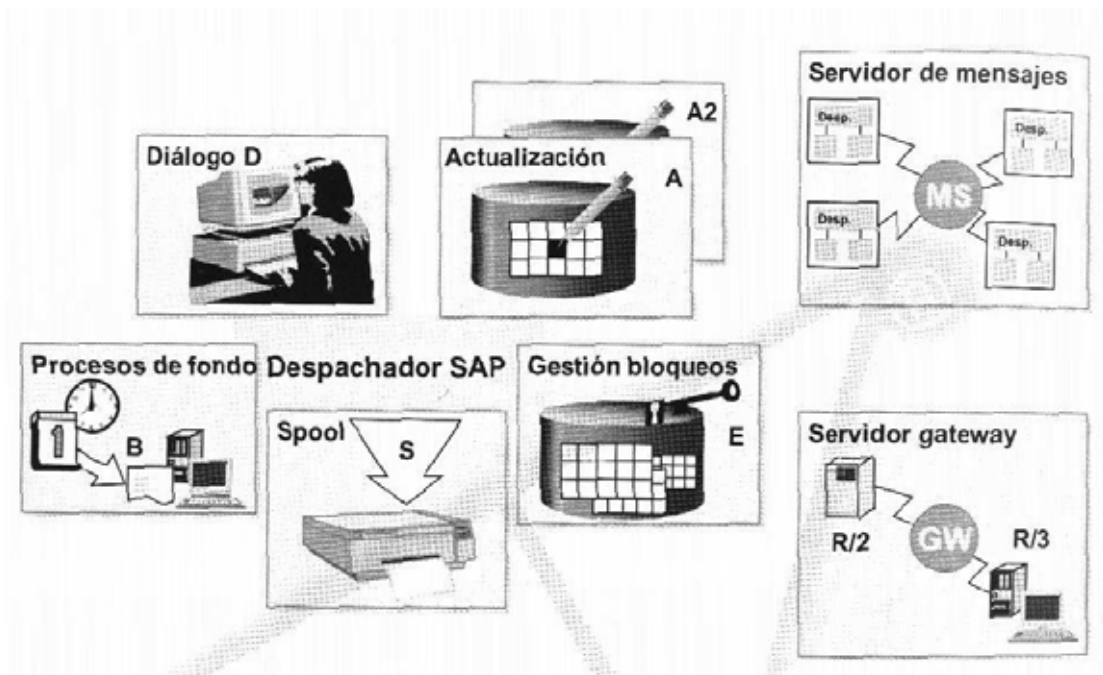
1.6.1 Despachador (Dispatcher)

El despachador es el proceso central de la capa de aplicación de R/3. El despachador administra los recursos de las aplicaciones de R/3 en coordinación con el sistema operativo. Entre las tareas principales del despachador se encuentran la distribución de la carga de las transacciones entre los procesos de trabajo, la conexión con la capa de presentación y la organización de las operaciones de comunicación.

1.6.2 Procesos de trabajo en R/3 (Work Processes)

Los procesos de trabajo cooperando en conjunto constituyen las instancias R/3. El número de procesos de trabajo depende de los recursos disponibles. Los diferentes procesos que pueden ser configurados son los siguientes:

- **Servidor de Mensajes:** se encarga de la comunicación entre los despachadores del sistema R/3.
- **Servidor de Puente (Gateway):** permite la comunicación entre un R/2 y un R/3 u otros sistemas de aplicación externos.
- **Procesos de Diálogo (Dialog):** atiende requerimientos de procesamiento de los usuarios en línea.
- **Procesos de impresión (Spool):** atiende peticiones de impresión.
- **Procesos de Actualización (Update):** coordina las tareas de actualización asíncrona a la base de datos.
- **Procesos de Fondo (Batch):** atiende requerimientos de procesamiento de los usuarios en modo invisible (batch).
- **Proceso de Bloqueos (Enque):** atiende los requerimientos de bloqueo de objetos a nivel R3.

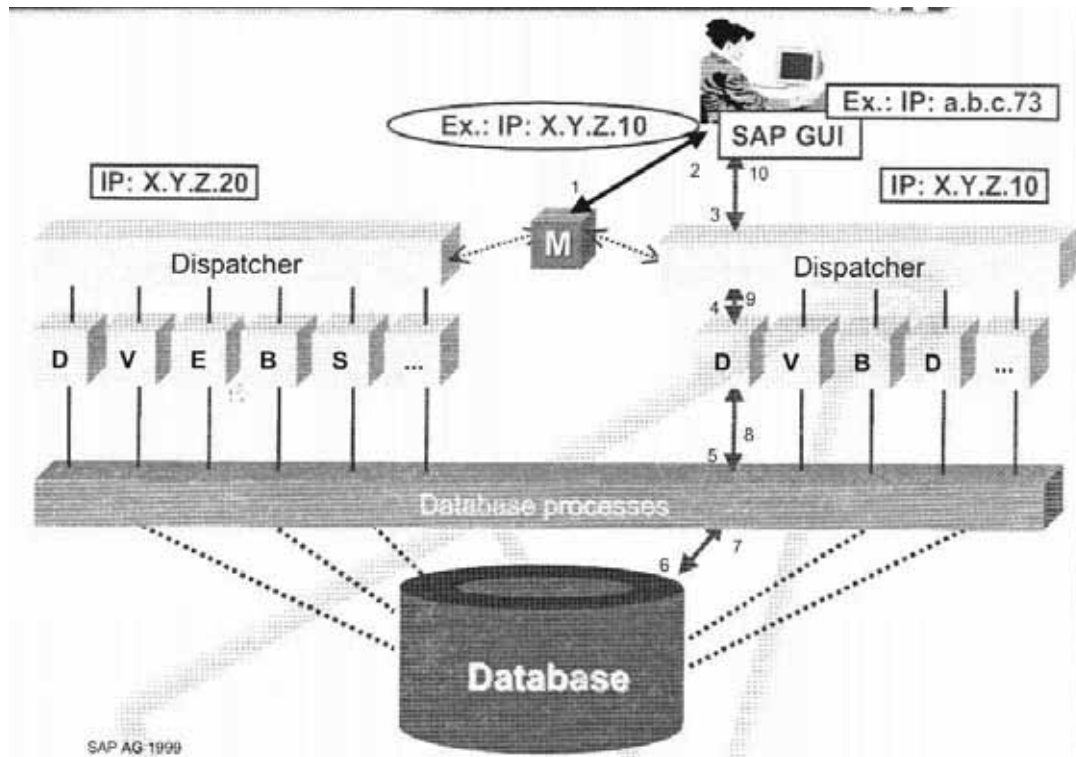


1.6.3 Proceso de acceso al R/3 (Balanceo de Cargas)

R3 puede utilizar de manera proporcional los servidores de aplicación conectando a cada usuario al servidor de aplicación que tenga el mejor tiempo de respuesta en los últimos 5 minutos.

La solicitud de acceso es enviado al servidor de mensajes y este se encarga de identificar aquel servidor con mejor tiempo de respuesta en los últimos 5 min.

Una vez identificado el servidor de mensajes regresa al cliente la dirección IP del despachador asignado. Este proceso es conocido como balanceo de cargas.

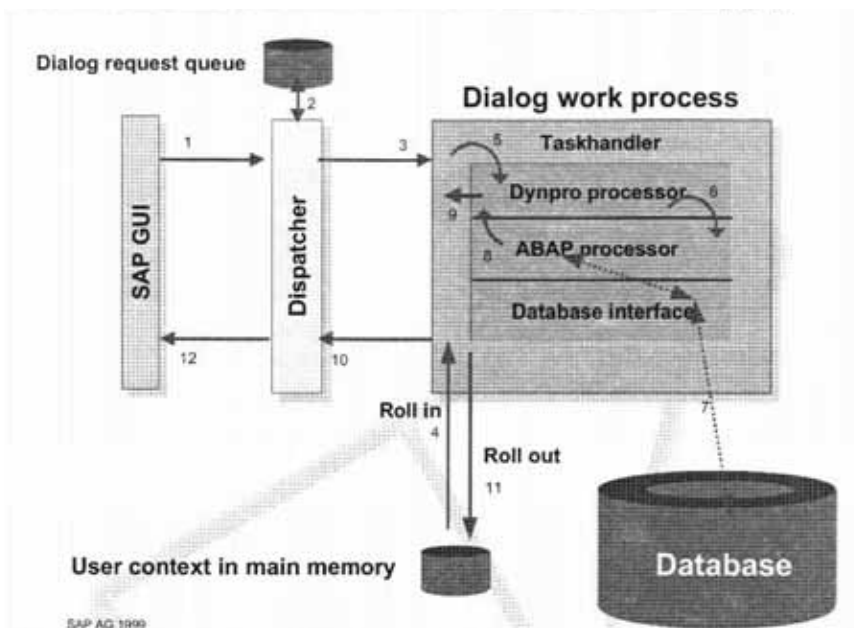


1.6.4 Paso de diálogo (Dialog Step)

Cuando un usuario está navegando en el sistema R3 de pantalla en pantalla esta realizando pasos de diálogo los cuáles en conjunto forman una transacción.

El número de pasos de diálogo permite medir la carga de trabajo en el sistema. El dimensionamiento de los servidores de aplicación es basado en pasos de diálogo.

En el momento que un usuario establece una conexión por medio del SAP GUI y una sesión es creada en el sistema se generan los siguientes pasos al ejecutarse una petición de trabajo:



4. Los datos son transmitidos del SAP GUI hacia el despachador.
5. El despachador clasifica la solicitud y la coloca en la cola de espera apropiada.
6. Un proceso de trabajo disponible es asignado a atender la solicitud.
7. El contexto de usuario (donde se almacena los permisos y datos propios de la sesión principalmente) son cargados por un componente llamado administrador de tareas (taskhandler).
8. El administrador de tareas llama al procesador de programas dinámico (dynpros) para convertir la pantalla correspondiente en variables ABAP.
9. El procesador ABAP ejecuta el código PAI (Process After Input) de la pantalla predecesora así como el PBO (Process Before Output) de la pantalla siguiente. Si es necesario también se comunica con la BD.

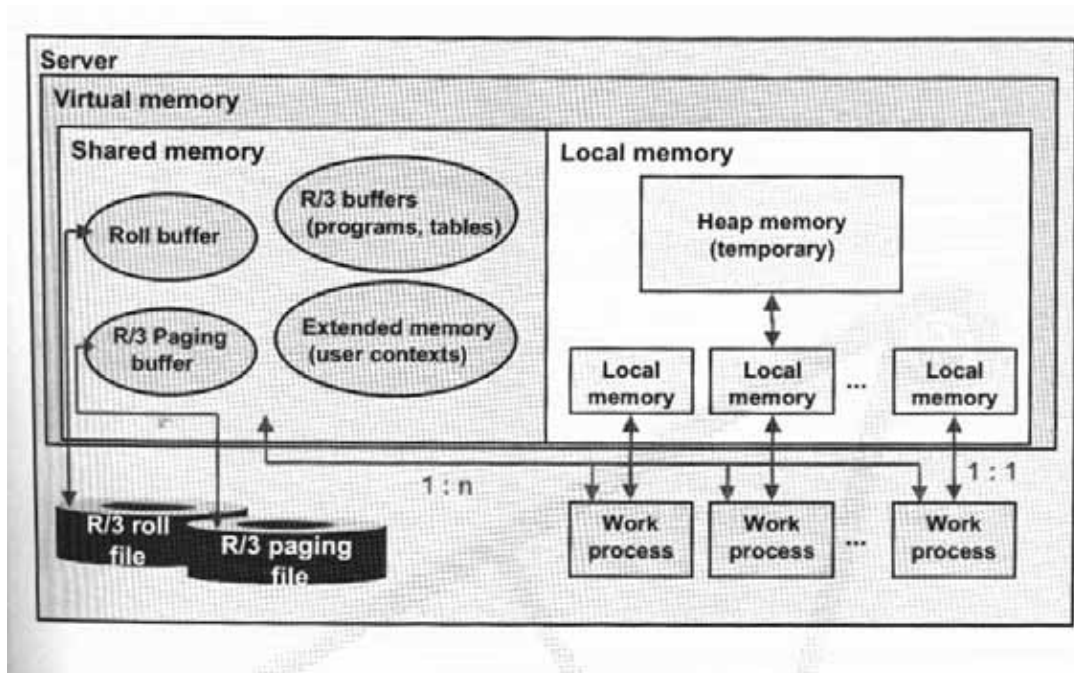
- 10.El procesador dinámico de programas convierte las variables ABAP de nuevo en campos de pantalla. Cuando el dynpro ha finalizado su tarea, el administrador de tareas se activa de nuevo.
- 11.El contexto de usuario actual es almacenado en la memoria compartida por el administrador de tareas, este proceso se conoce como roll-out.
- 12.Los datos resultados son enviados al SAP GUI por medio del despachador.

1.6.5 Memoria R/3

R/3 utiliza la memoria de manera eficiente para procesar la mayor cantidad de trabajo en ella. Para R/3 la memoria que puede utilizar es la memoria física del servidor más la memoria virtual definida (swap)³.

Capa proceso de R/3 utiliza un tipo de memoria que se especifica a continuación.

- **Memoria Local (Local Memory)** La memoria local esta asociada con procesos de trabajo de manera independiente. Se subdivide en memoria local de los procesos de trabajo (work process local area) y memoria heap
- **Memoria compartida (Shared Memory)**. La memoria compartida esta asociada a todos los procesos de la instancia. Se subdivide en Memoria Extendida (Extended Memory), R/3 Buffers, Memoria de Roll y Memoria de Paging



³ El swap requerido por R/3 es muy grande y varia de acuerdo a la plataforma, por ejemplo en Kernel Solaris de 64 bits requiere al menos 20 GB definidos y en kernel de 32 bits 3 veces la memoria física. OSS Note 146289

1.7 Importancia del desempeño de sistemas R3

El desempeño de un sistema de procesamientos de datos esta definido como la capacidad del sistema para atender las solicitudes de trabajo en proporción al tiempo de respuesta (***Response Time***) y el rendimiento del procesamiento de datos (***Throughput***)⁴. La definición anterior describe el desempeño de sistemas SAP en términos de dos unidades: el tiempo de respuesta y el rendimiento en dialog steps.

Adicionalmente al tiempo de respuesta y a la capacidad de procesamiento existen medidas adicionales que a continuación se describen:

Ancho de banda (Bandwith)

- Máxima capacidad que no puede excederse
- Una medida que ignora la sobre carga del sistema (overhead)
- Algo que nunca es llevado en la practica, siempre existe algo de ineficiencia y sobrecarga en el sistema

Rendimiento de procesamiento (Throughput)

- La cantidad de trabajo que puede resolverse en un intervalo de tiempo determinado.
- Lo que realmente se utiliza, esto es cuanto ancho de banda es actualmente utilizado.
- Una medida que es dependiente de varios factores: el hardware, software, humano y el azar.
- Un número que normalmente no es posible calcular pero puede ser aproximado.
- Máximo rendimiento de procesamiento si lo que se esta midiendo es cuanto trabajo puede realizarse al 100% de utilización.

Tiempo de respuesta (Response Time)

- El tiempo tomado en completar un petición de usuario.
- Tiempo transcurrido para que una operación se complete.
- Usualmente tienen el mismo significado que *estado latente* (latency).

Tiempo de servicio (Service Time)

- Tiempo de procesamiento actual de un requerimiento.
- Cuanto tiempo toma en realizarse un requerimiento, ignorando cualquier retraso por encolamiento.
- Es igual que el tiempo de respuesta cuando no existe encolamiento.

⁴ Thomas Shneider, SAP R/3 Performance Optimization, pág. XIX

Utilización

- La cantidad que un recurso es utilizado, ya sea en conjunto o individual, para realizar el trabajo medido.
- La cantidad del rendimiento de procesamiento realizado por un recurso.
- Un porcentaje que usualmente significa porcentaje de ocupación.

SAP ha definido pruebas estándares para determinar el desempeño de los diferentes componentes (módulos) en configuraciones de hardware ofrecidas por los proveedores. Los resultados de estas pruebas permiten conocer de antemano el desempeño de los sistemas medidos en SAPS (SAP Application Performance Standard). En términos prácticos 100 SAPS equivalen a procesar 2000 ordenes de ventas y distribución en una hora.⁵

A continuación se muestra una certificación de SAP del rendimiento de una configuración de hardware.



Este documento permite conocer el desempeño del servidor medido tanto en rendimiento (SAPS), tiempo de respuesta promedio obtenido y utilización del CPU. También se puede considerar el número de SAPS obtenido como el ancho del banda del sistema, es decir cuál sería su máximo rendimiento.

⁵ <http://www.sap.com/benchmark>, correspondiente a la prueba estándar de ventas y distribución.

SAP R/3 es un sistema diseñado para el procesamiento tipo OLTP (On Line Transaction Process), por esta razón el sistema se dimensiona, configura y administra buscando optimizar procesos transaccionales en paralelo y de tiempo de ejecución corto, como son la entrada de ordenes de compra, movimiento de materiales, entradas de ordenes de producción, contabilizaciones, facturaciones, etc. Todos los procesos de larga duración y de consulta sobre rangos extensos de información del tipo DSS (Decision Support Systems) deben de ser ejecutados en horarios donde no se afecte la operación transaccional en proceso de fondo (batch). En esta estricta teoría estos procesos no se dimensionan y se consideran excepcionales, sin embargo en la práctica son estos procesos los que generan una sobrecarga en el sistema provocando degradaciones significativas en el desempeño del sistema dando como resultado un redimensionamiento de la propuesta original.

Un estudio formal del desempeño del sistema debe contemplar al menos 3 aspectos:

- Rendimiento
- Tiempo de Respuesta
- Utilización

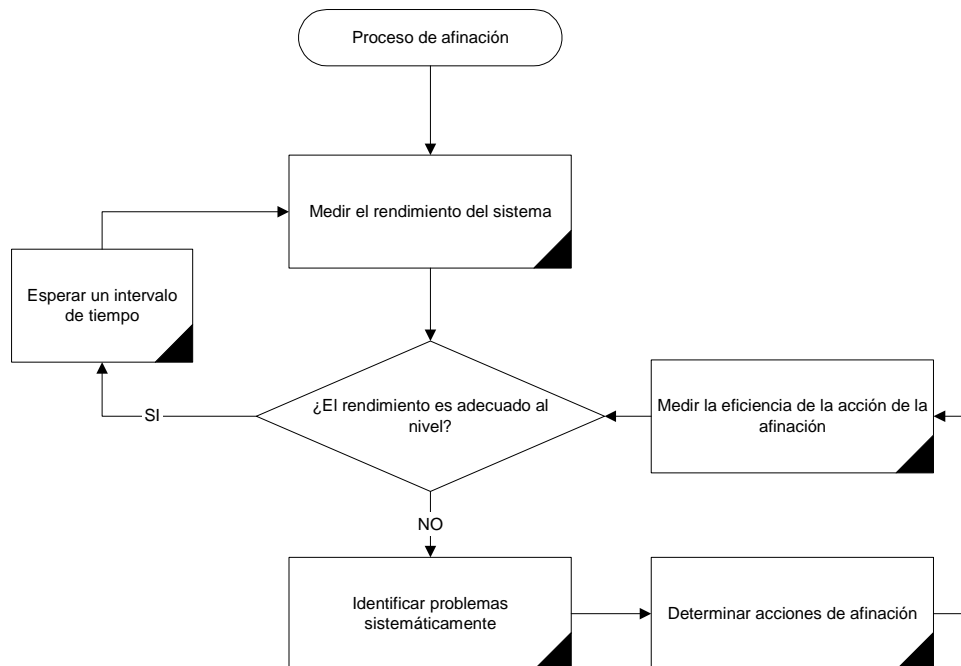
Si el sistema no alcanza el nivel de servicio acordado, puede comenzar a surgir la inconformidad de los usuarios y poner en riesgo el proyecto de implantación. En este punto el administrador debe contar con datos que le permita determinar las acciones a seguir para llegar al nivel de servicio, es decir que tipo de afinación requiere el sistema.

1.8 Afinación del desempeño de sistemas R3

Afinación significa hacer más eficiente el uso de recursos con los que un sistema cuenta para soportar la carga de trabajo que se espera atender. Cuando se realizan una afinación se esta buscando alcanzar como objetivo:

- Adaptar el sistema de acuerdo a la carga de trabajo
- Mejorar la percepción del usuario sobre el tiempo de respuesta del sistema.
- Optimizar el uso de recursos del sistema.
- Mejorar la utilización del sistema.
- Incrementar la capacidad del sistema.
- Reducir costos.

El proceso de afinación de manera general contempla medir el desempeño, analizarlo, realizar una acción correctiva y repetir el proceso de manera continúa hasta alcanzar el objetivo planteado.



Hay dos tipos de afinación que deben realizarse de manera conjunta pero que están delimitadas en cuanto a responsabilidades y alcances: la afinación técnica y la afinación de la aplicación.

1.8.1 Afinación técnica

Esta orientada a la correcta configuración del sistema, la distribución adecuada de los procesos R/3 sobre los recursos de hardware. El responsable directo es el administrador del módulo de base. Algunas tareas que contemplan esta afinación son:

- Configurar los parámetros de memoria de R/3, buffers de base de datos y parámetros de kernel de sistema operativo.
- Distribución de carga de trabajo en el sistema, configuración de procesos de trabajo, balanceo de cargas y modos de operación.
- Modificar o ampliar la capacidad de hardware: cpu, memoria, disco
- Modificar o ampliar la capacidad de los enlaces de comunicación locales (LAN) y remotos (WAN).
- Redistribución de espacios de la base de datos.
- Asegurar la calidad de los índices de la base de datos así como información precisa sobre la distribución de los datos para el funcionamiento correcto del optimizador de costos de la base de datos (actualización de estadísticas).

1.8.2 Afinación de la aplicación

La afinación de la aplicación esta destinada a eliminar carga innecesaria en el sistema provocada por programas ineficientes o con un uso inadecuado.

La aplicación de acciones correctivas adecuadas deben de contemplar ambos aspectos de la afinación e involucrar a todos los responsables del foro de desempeño para tomar decisiones en conjunto.

Entre las tareas realizadas se contempla:

- Revisar sentencias SQL costosas.
- Analizar los programas que más carga están generando en el sistema.
- Revisar parámetros de configuración de la aplicación.
- Capacitación de los usuarios.

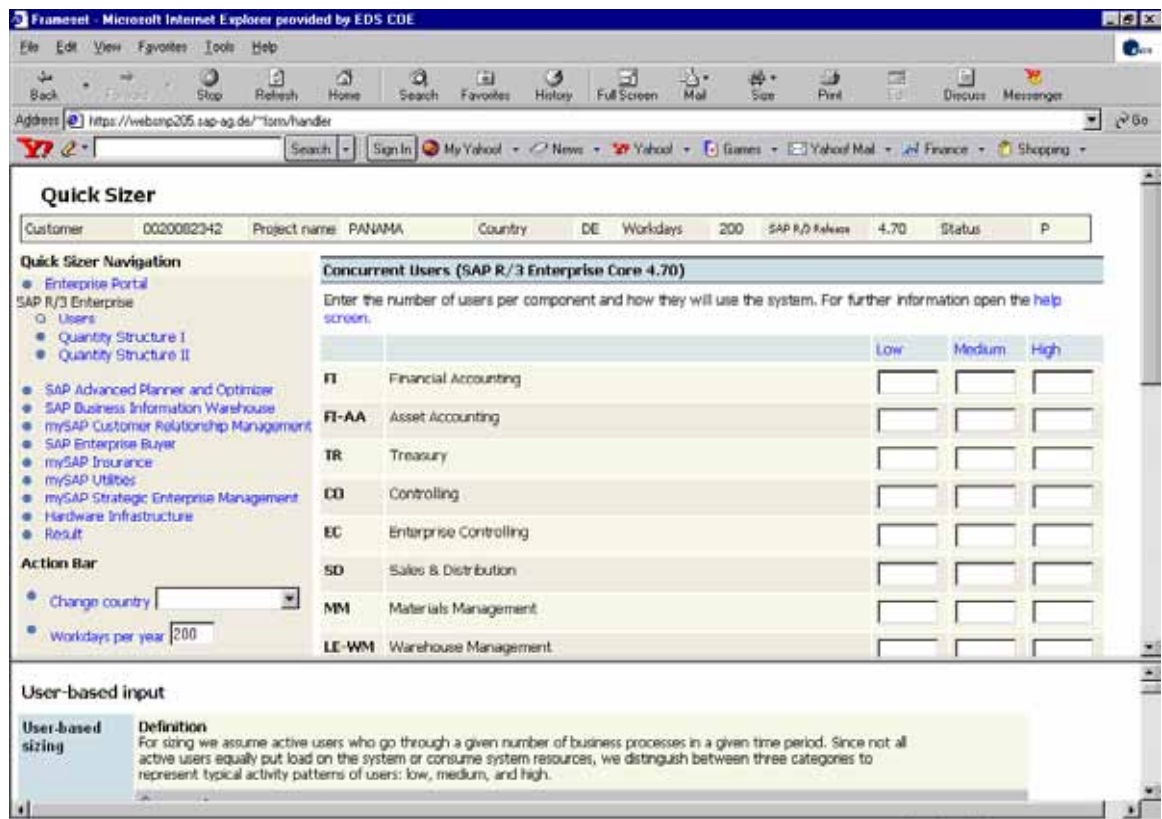
2 Utilidad de un Sistema de Monitoreo de Desempeño de Servidores (SMDS).

2.1 Dimensionamiento de un servidor de aplicación

Al iniciar un proyecto de implementación de SAP R/3 es necesario dimensionar la capacidad de los servidores de aplicación, estos servidores son:

- Servidor de base de datos
- Instancia central R/3
- Servidores de aplicación.

Este proceso de dimensionamiento esta basado en métodos matemáticos que proyecta la carga esperada en el servidor¹ medido en SAPS. La herramienta utilizada para esta proyección es el Quick Sizer de SAP que se encuentra disponible en la internet <http://sapnet.sap.com/quicksizing>.

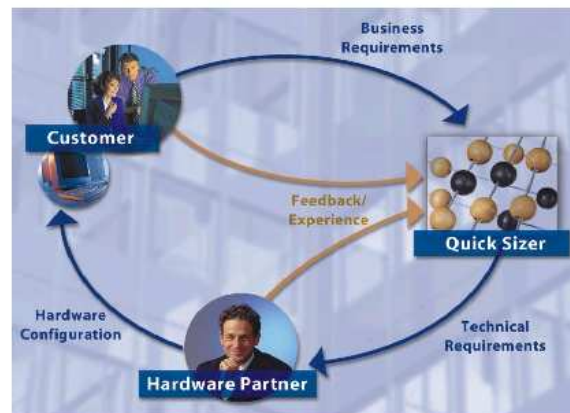


El dimensionamiento se basa en los siguientes datos:

¹ Sizing mySAP.com, SAP AG, metodología de dimensionamiento de hardware de SAP www.sap.com.

- **Módulos a instalar.** Cada módulo SAP generará una carga específica en el servidor, por ejemplo PP es el módulo que más carga genera al sistema, mientras que un HR requerirá menor cantidad de recursos.
- **Transacciones esperadas.** Número de las operaciones esperadas en el sistemas. Entre estas operaciones se contemplan contabilizaciones, creación de ordenes de producción, traslados, movimientos de materiales, generación de ordenes de venta, etc.
- **Usuarios.** Tipificación de usuarios concurrentes en bajos, medios y altos de acuerdo a la actividad esperada en el sistema.

Una vez determinado el número de SAPS requeridos el proveedor propondrá una infraestructura de acuerdo al nivel de servicio requeridos por el cliente.



Cuando el sistema es liberado a producción es necesario determinar si el dimensionamiento es adecuado a la carga de trabajo real, este es el primer requerimiento del sistema de monitoreo de desempeño del sistema planteado.

2.2 Capacidad de procesamiento de los servidores.

La propuesta del proveedor de hardware determinará la capacidad de los equipos para ejecutar el sistema R3 de manera óptima, esta capacidad está compuesta por:

- CPU
- Memoria
- Dispositivos de almacenamiento
- Interfaz de red (IR)².



Cuando algunos de los componentes llegar al límite de su utilización el desempeño del sistema comienza a degradarse. Cuando esto sucede se cuestiona la validez del dimensionamiento original y se comienza a evaluar la adquisición de mayor capacidad de cómputo, sin embargo esto no siempre es la mejor solución.

Para complementar el diagnóstico es necesario revisar los procesos de la aplicación para sustentar adecuadamente alguna recomendación. La razón principal de esto es que pueden existir factores en la aplicación que pueden saturar a un servidor de aplicación.

² La interfaz de red se refiere únicamente a la salida y entrada de datos del servidor. Si existen problemas con los enlaces entonces el usuario también percibirá lentitud. Sin embargo, esto puede recaer en cualquier eslabón de una red de área amplia (WAN) típica de las implementaciones de R3, las WAN requiere un equipo de administradores y a ellos se debe reportar problemáticas de comunicaciones fuera de la interfaz local de red.

2.3 Factores que pueden saturar la utilización de un servidor.

Si los servidores de aplicación se encuentran saturados se debe considerar los siguientes factores que pueden estar involucrados en esta problemática:



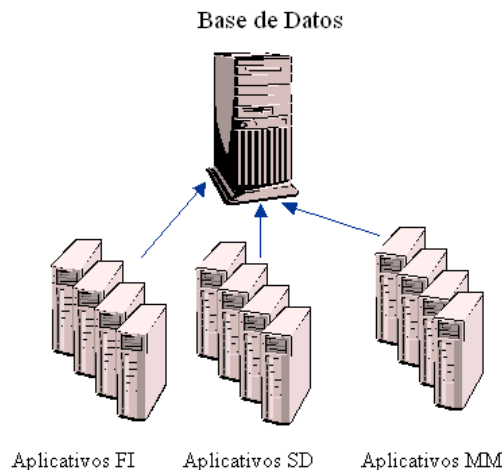
- **Inadecuado dimensionamiento de los servidores de aplicación.** No se considero los usuarios reales, se implementaron más módulos que los proyectados originalmente, falta de precisión por parte del proveedor de hardware.
- **Configuración inadecuada de la base de datos.** La memoria se encuentra definida mayor a la capacidad física, deficiente distribución de datos sobre los discos físicos, incorrecto mantenimiento de actualización de estadísticas, fragmentación de extents, falta de índices o el no uso de ellos.
- **Configuración inadecuada de R3.** Asignación incorrecta de memoria, inadecuada distribución de procesos de trabajo, no utilización de balanceo de carga.
- **Deficiente programación.** Liberación de programas sin las pruebas de calidad adecuadas (volumen y concurrencia).
- **Deficiente parametrización del sistema (customizing).** Los módulos de SAP requieren la adecuación a la empresa y en algunas ocasiones pueden generar detalle o validaciones que sobrecargan el sistema de manera innecesaria.
- **Uso inadecuado de programas.** Los usuarios carecen de entrenamiento adecuado, desconocimiento de las mejores prácticas de SAP para ejecutar ciertos procesos.

Una vez que se ha identificado un problema es necesario descartar cualquiera de estos factores. Esto se realizará con herramientas propias de SAP de análisis de desempeño que se presentan en el capítulo 4.

2.4 La complejidad de analizar el desempeño de los servidores.

Cuando el sistema entra en producción, se debe determinar si se cuenta con suficiente capacidad de cómputo, es decir conocer el rendimiento de cada uno de los servidores de aplicación y analizar si se encuentra en los umbrales deseados de rendimiento. Sin embargo, esta tarea en la práctica tiende a ser sumamente compleja.

La razón de lo anterior se debe a la escalabilidad de los sistemas SAP R3 que permite configurar tantos servidores de aplicación como la necesidad de procesamiento lo requiera. En la práctica un sistema productivo R/3 puede estar compuesto de 5, 10 o 20 servidores de aplicación. (Instalaciones como PEMEX y Telmex que son las más grandes de México con 5000 usuarios concurrentes tienen 20 servidores de aplicación cada uno). Cada servidor de aplicación deberá ser monitoreado en sus recursos de CPU, memoria y disco de manera individual. Este efecto es más drástico en los discos ya que un sistema de almacenamiento puede contener varios cientos lo que hace crecer la complejidad de manera exponencial.



Para ofrecer recomendaciones adecuadas sobre el desempeño de los servidores el administrador requiere una visión global del rendimiento de los servidores de aplicación que responda preguntas claves:

- ¿Existe saturación en la utilización de CPU en los servidores de aplicación? (o memoria, o disco, o IR)
- ¿Existe un balanceo en la utilización de los servidores de aplicación?
- ¿Cuál ha sido el uso de (CPU, memoria, disco, IR) de los servidores durante el día?

- ¿Cuál ha sido el uso de (CPU, memoria, disco, IR) de los servidores en días pasados?
- ¿Cuál es el desempeño de la distribución de los discos?

Las dos últimas preguntas permiten al administrador comparar el rendimiento de los servidores cuando el sistema tienen un desempeño adecuado contra los momentos en que el servidor no está cubriendo las expectativas.

Esta necesidad de monitoreo son los principios para diseñar y programar un sistema de recolección de métricas de desempeño y la facilidad de analizarlas posteriormente, al software que se ajuste a estas necesidades lo he denominado Sistema de Monitoreo de Desempeño de Servidores de Aplicación (SMDSA).

2.5 Herramientas de análisis de desempeño de servidores en Unix

El desempeño de los servidores de aplicación puede ser medido por comandos y herramientas nativas del sistema operativo, a continuación se presenta su utilización específicamente en ambiente Solaris. La variación con los restantes UNIX es poco significativa:

2.5.1 sar

Sar es acrónimo de "System activity reporter", ofrece un rango amplio de información estadística sobre la mayoría de las áreas del sistema:

- Llamadas a File System y sistema
- Uso de memoria física, paginación, swap y memoria de kernel.
- Actividad de dispositivos TTY y block device.
- Tabla de estado de procesos.
- Actividad SV IPC y uso de CPU.

Existen 16 tipos de reportes distintos los cuales pueden ser combinados para coleccionar grandes cantidades de datos.

2.5.2 vmstat

Vmstat es acrónimo de "Virtual memory statistics", estos reportes se enfocan sobre memoria y actividad del CPU. Existen 5 diferentes reportes.

La información ofrecida por vmstat es similar a la reportada por sar y en algunos casos incluye datos no contemplados en sar. Las áreas del sistema que contempla son:

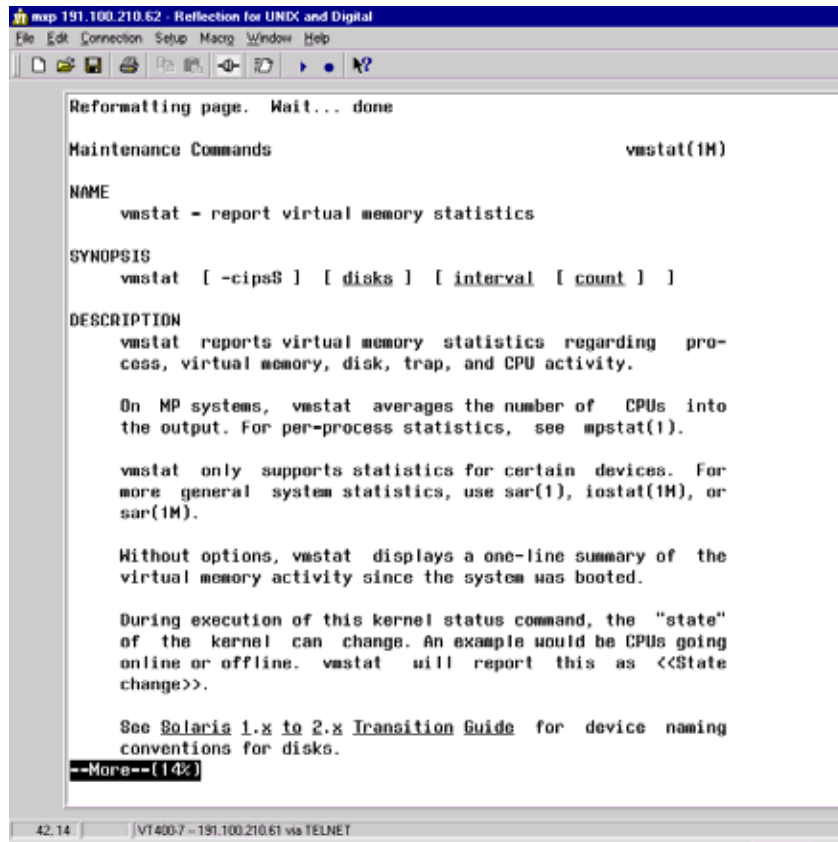
- Procesos, Interrupciones, cache de CPU y actividad de CPU.
- Memoria virtual y actividad de swapeo.
- Actividad en disco.

La salida del comando promedia 2 valores del kernel por esta razón la primera línea de la salida debe omitirse.

2.5.3 iostat

Iostat significa estadísticas de IO, este comando reporta datos del rendimiento del IO por disco y dispositivos TTY (terminales). Adicionalmente ofrece información de la utilización del CPU. La versión de Solaris 2.6 ofrece la posibilidad de rastrear puntos de montaje de NFS, particiones y el rango errores del dispositivo.

Los comandos anteriormente presentados se encuentran completamente documentadas en las páginas man de los sistemas UNIX.



```
map 191.100.210.62 - Reflection for UNIX and Digital
File Edit Connection Setup Macro Window Help
Reformatting page. Wait... done
Maintenance Commands                                vmstat(1M)
NAME
  vmstat - report virtual memory statistics
SYNOPSIS
  vmstat [ -cips$ ] [ disks ] [ interval [ count ] ]
DESCRIPTION
  vmstat reports virtual memory statistics regarding process,
  virtual memory, disk, trap, and CPU activity.
  On MP systems, vmstat averages the number of CPUs into the
  output. For per-process statistics, see mpstat(1).
  vmstat only supports statistics for certain devices. For
  more general system statistics, use sar(1), iostat(1M), or
  sar(1M).
  Without options, vmstat displays a one-line summary of the
  virtual memory activity since the system was booted.
  During execution of this kernel status command, the "state"
  of the kernel can change. An example would be CPUs going
  online or offline. vmstat will report this as <<State
  change>>.
  See Solaris 1.x to 2.x Transition Guide for device naming
  conventions for disks.
--More--(14%)
42.14 | |VT400-7 - 191.100.210.61 via TELNET
```

2.5.4 Carga generada por comandos de monitoreo de sistema

Un aspecto a cuidar en el SMDSA es no afectar negativamente el sistema a monitorear. Cuando se mide el desempeño de un sistema se debe tener precaución sobre el consumo de recursos que la herramienta utilizada puede generar dado que no es posible medir algo sin afectarlo de alguna manera.

Para el caso de UNIX la recolección de datos con los comandos descritos no generaran sobrecarga en el sistema. La razón de esto es realizan consultas a estructuras del kernel localizadas en la memoria del sistema por lo cuál son sumamente eficientes ejecutándose en microsegundos³.

A continuación se presenta el resultado de medir la utilización de los comandos iostat y vmstat. El resultado fue consistente en diversas horas del día y de manera constante **se observa el 0.1% de utilización de memoria y cpu.**

`/usr/ucb/ps -aux`

USER	PID	%CPU	%MEM	SZ	RSS	TT	S	START	TIME	COMMAND
mxpadm	5789	0.1	0.1	1048	848	?	S	16:51:00	0:00	vmstat 5 2
mxpadm	6190	0.1	0.1	1048	848	?	S	16:56:05	0:00	vmstat 5 2
mxpadm	8175	0.1	0.1	1048	848	?	S	17:26:37	0:00	vmstat 5 2
mxpadm	9586	0.1	0.1	1048	920	?	S	17:48:16	0:00	iostat -x 5 2
mxpadm	9581	0.0	0.1	1048	848	?	S	17:48:14	0:00	vmstat 5 2
mxpadm	9755	0.1	0.1	1048	920	?	S	17:53:17	0:00	iostat -x 5 2
mxpadm	10864	0.1	0.1	1048	848	?	S	18:12:28	0:00	vmstat 5 2
mxpadm	14485	0.1	0.1	1048	848	?	S	19:08:30	0:00	vmstat 5 2
mxpadm	15323	0.1	0.1	1048	848	?	S	19:24:01	0:00	vmstat 5 2
mxpadm	15940	0.1	0.1	1048	848	?	S	19:32:35	0:00	vmstat 5 2
mxpadm	16755	0.1	0.1	1048	848	?	S	19:49:12	0:00	vmstat 5 2
mxpadm	18252	0.1	0.1	1048	920	?	S	20:13:21	0:00	iostat -x 5 2
mxpadm	18792	0.1	0.1	1048	920	?	S	20:18:21	0:00	iostat -x 5 2
mxpadm	18924	0.1	0.1	1048	920	?	S	20:23:21	0:00	iostat -x 5 2
mxpadm	19130	0.1	0.1	1048	848	?	S	20:24:50	0:00	vmstat 5 2
mxpadm	19212	0.1	0.1	1048	920	?	S	20:28:21	0:00	iostat -x 5 2
mxpam	19666	0.1	0.1	1048	920	?	S	20:33:21	0:00	iostat -x 5 2
mxpadm	19803	0.1	0.1	1048	920	?	S	20:38:21	0:00	iostat -x 5 2
mxpadm	21827	0.1	0.1	1048	848	?	S	21:07:57	0:00	vmstat 5 2
mxpadm	22101	0.1	0.1	1048	848	?	S	21:16:28	0:00	vmstat 5 2
mxpadm	24784	0.1	0.1	1048	848	?	S	21:57:09	0:00	vmstat 5 2

Sin embargo, el SMDSA requerirá capacidad de procesamiento para almacenar y manipular los datos recolectados, por esta razón se recomienda utilizar los sistemas DEV o QAS del ámbito de SAP (sección 1.5).

³ Adrian Cockcroft – SUN Performance and Tunning Cap 15 Metric Collection Interfaces

3 Construcción del sistema de monitoreo de desempeño de servidores.

En el capítulo anterior se justificó la necesidad de desarrollar una herramienta de software que mida si el conjunto de servidores de aplicación de un sistema R3 tiene la suficiente capacidad de procesamiento para la ejecución del sistema óptimamente. Las 2 funciones del sistema requerido son:

- Recolección y almacenamiento de métricas de desempeño.
- Explotación de análisis de los datos colectados.

El presente capítulo se desarrollará el SMDSA con base al planteamiento del análisis de capítulo anterior.

3.1 Herramientas de construcción

Las versiones estándares de UNIX cuentan con el lenguaje de programación Korn Shell, todas las implementaciones de UNIX en el mercado donde se ejecuta R3 (HP-UX, AIX, Solaris, Linux, Sequent) cuentan con este lenguaje de programación. Las ventajas encontradas en este lenguaje son:

- **Portabilidad**, disponible en todos los UNIX donde se ejecuta SAP R3.
- **Costo Nulo**, no requiere costo de licenciamiento por desarrollo (incluida en la licencia de sistema operativo).
- **Cliente Ligero**, es posible ejecutarse en cualquier emulador terminal de carácter por lo que no requiere de costosos equipos para su consulta.

El repositorio de datos se utilizará la base de datos del sistema R3, en el caso desarrollado es Oracle 8i, pero en un servidor diferente a los productivos que puede ser calidad o a desarrollo. La explotación de los datos se realiza con SQL estándar.

Para presentar los datos se utiliza las funciones de graficación de Excel conectado vía ODBC al repositorio de datos.



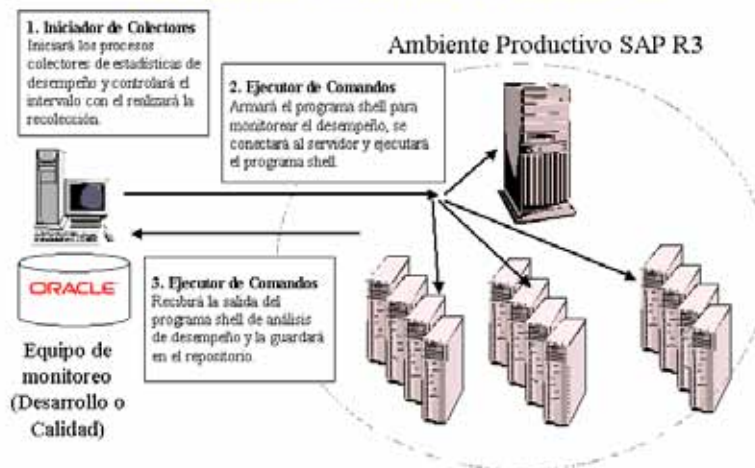
3.2 Recolección y almacenamiento de métricas de desempeño

Con los herramientas de monitoreo de UNIX presentadas en la sección 2.5 es posible medir el rendimiento de los componentes de sistema en tiempo real. Para un diagnóstico adecuado es necesario utilizar los comandos en intervalos constantes en tiempo de manera simultánea sobre todos los servidores de aplicación, de esta manera es posible tener puntos de comparación consistentes.

Este primer componente del SMDSA esta formado por 2 módulos:

- **Iniciador de colectores.** Inicia los colectores de estadísticas de desempeño para cada métrica definida controlando el intervalo de tiempo definido.
- **Ejecutor de Comandos.** Ejecutará los comandos de análisis de rendimiento sobre los servidores de aplicación y el resultado será almacenado en el repositorio.

Recolección y Métricas de Desempeño



3.2.1 Iniciador de colectores.

El colector de estadísticas será responsable de controlar los procesos de recolección de datos para cada métrica definida, esto lo realiza leyendo el archivo de configuración métricas.lst en el siguiente formato:

<métrica>|<intervalo>|<tipo>

Donde:

- **Métrica o Nombre de Estadística de Desempeño.** Es el nombre de la métrica a monitorear cpu, memoria, disco, etc.
- **Intervalo.** Define el tiempo en segundos de espera para ejecutar de nuevo el monitoreo.
- **Tipo.** Dependiendo de la métrica a monitorear base de datos (ora) o servidor de aplicación (so).

Ejemplo:

```
cpu|900|so
memory|900|so
disk|900|ora
vmstat|300|so
```

La primera fase de este proceso consiste en iniciar la ejecución los procesos de recolección de datos. El programa shell que realiza la primera parte de esta tarea es inicial.ksh, el pseudo código correspondiente será el siguiente:

```
Leer el archivo métricas.lst
Por cada métrica de desempeño:
  Iniciar el colector de datos
```

De este proceso se generarán n procesos de recolección de estadísticas de desempeño donde n es el número de métricas definidas en métricas.lst.

Una vez iniciado un colector este se encargara de invocar la Ejecución del Comando durante el intervalo definido específicamente para esa métrica.

El pseudo código de esta fase es:

```
En un ciclo infinito:
  Invocar el ejecutor de comandos
  Esperar durante el intervalo definido
```

El script colector.ksh del apéndice contiene el código del korn shell.

3.2.2 Ejecutor de comandos

Este componente es responsable de la ejecución de los comandos de monitoreo de desempeño hacia los servidores de aplicación.

La forma de realizar esta ejecución es utilizar la capacidad de los sistemas UNIX de ejecutar de manera remota comandos y programas. La sintaxis de la instrucción de ejecución remota (ssh¹) es la siguiente

```
ssh <-l username> ip <ruta el programa a ejecutar>
```

Para poder realizar esto es necesario definir al usuario con el cuál se esta ejecutando el script y la maquina origen como confiados (trusted) en el servidor a monitorear.

El archivo `$HOME/.ssh/known_hosts` localizado en el home del usuario destino permite que esto funcione correctamente así como las claves de cifrado.

```
<servidor origen> <usuario trusted>
```

Para el proyecto realizado utilice un equipo de desarrollo como repositorio y disparador de estadísticas denominado mxdev.

El siguiente ejemplo muestra la conexión remota a un servidor:

```
mxdev:oramxd 24% ssh -l mxpadm 191.100.210.62 cat .rhosts; uname -a
mxdev      oramxd
SunOS mxdev 5.7 Generic_106541-20 sun4u sparc SUNW,Ultra-60
```

Lo que realiza este comando es :

1. Conectarse al servidor con ip 191.100.210.62
2. Tomar el perfil de usuario mxpadm (en el servidor destino)
3. Lista el archivo `.rhosts` localizado por default en el home del usuario mxpadm.
4. Listar el nombre del servidor al cuál se está conectando (191.100.210.62 tiene el nombre de mxpro)

Por cada métrica el proceso colector estará ejecutando scripts para medir su utilización.

La configuración de este script requiere la lista de los servidores a monitorear con el siguiente formato `servers.lst`:

¹ El comando rsh estándar de los sistemas UNIX permite la comunicación entre servidores de manera confiada (trusted), sin embargo al activarlo se compromete la seguridad del sistema haciéndolo vulnerable a ataques documentados en diversas fuentes. Una de estas fuentes es openSSH quien a desarrollado la herramienta que remplaza a rsh permitiendo la misma funcionalidad de manera segura <http://www.openssh.org/>.

<servername>|<ip>|<usuario a conectarse>|<tipo de servidor c central a aplicativo>

Ejemplo:

```
mxpro|191.100.210.62|mxpadm|c
mxapp1|191.100.210.63|mxpadm|a
mxapp2|191.100.210.64|mxpadm|a
mxqa|191.100.210.55|mxpadm|a
mxapp4|191.100.210.69|mxpadm|a
mxapp5|191.100.210.81|mxpadm|a
crpro|191.100.211.62|crpadm|c
crapp1|191.100.211.63|crpadm|a
crapp2|191.100.211.64|crpadm|a
crqa|191.100.211.55|crpadm|a
```

El pseudo código de las tareas realizadas por el script es el siguiente:

Para cada servidor definido en [servers.lst](#)
Crear el encabezado de cada script a ejecutar
- definir el nombre de la métrica
- obtener el nombre del servidor
- definir la ip
- los intervalos para el comando SAR y VMSTAT
- fijar la fecha de ejecucion
Armar el nuevo script basado en la plantilla definida en
<métrica>.ksh y el encabezado
Copiarlo al sistema remoto y ejecutarlo
Cargarlo en la BD

El código correspondiente al programa shell que realiza esto es disparador.ksh.

Para reforzar la seguridad se debe utilizar un usuario sin privilegio alguno que comprometa la seguridad del sistema.

3.2.3 Plantillas utilizadas por el ejecutor de comandos

El ejecutor de comandos construye el programa shell a ejecutar en los servidores de aplicación utilizando plantillas de programas shells. Se debe crear una plantilla por cada métrica que se desea monitorear y coleccionar.

Para crear una plantilla para una métrica específica se debe considerar lo siguiente:

- Formatear la salida del script con pipes "|" para que la función carga_datos pueda almacenar el resultado.
- Toda salida incluye el nombre del servidor, la dirección IP y la fecha de recolección. Esto es incluido por el ejecutor de comandos cuando arma el programa shell por lo que únicamente se deben incluir estos campos en la salida final con pipes.

Tomando como ejemplo el muestreo del rendimiento de cpu utilizando la instrucción sar -u obtendremos la salida siguiente:

```
sar -u 20 1
SunOS mxdev 5.7 Generic_106541-20 sun4u 10/04/02
17:54:03 %usr %sys %wio %idle
17:54:24 2 3 3 93
```

Las métricas de interés son %usr, %sys, %wio y %idle, adicionalmente todas las tablas de métricas contienen campos comunes servidor y fecha que deben ser incluidos en la salida. Así la salida completa deberá ser:

```
servername|fecha|usr|sys|idle
```

Un ejemplo completo puede visualizarse en el script [cpu.ksh](#).

3.2.4 Creación del usuario - esquema en Oracle para almacenar las métricas de desempeño.

El resultado de los scripts será almacenado en una base de datos, en el caso práctico utilice un equipo de desarrollo para guardar cada una de ellas. El usuario sugerido es monitor, la siguiente instrucción creará el usuario a nivel Oracle:

```
create user monitor identified by <password>
default tablespace psapmond
temporary tablespace psaptemp
quota unlimited on psapmond;
grant connect, resource to monitor;
```

El nombre de este usuario será almacenado en el archivo [user.lst](#) con el formato siguiente:

```
<user>|<password>
```

3.2.5 Función carga_datos.

El disparador una vez que obtiene los datos de la muestra debe cargarlos en la base de datos. La utilidad sql loader de Oracle es ideal para cargar un archivo plano a la base de datos. La sintaxis de una carga con la herramienta es la siguiente:

```
sqlldr usuario/password control=controlfile log=logfile
```

El archivo de control contiene las instrucciones para cargar los datos con la siguiente sintaxis:

```
Load data
Infile <data_file>
Append into table <nombre_tabla>
Fields terminated '|' (
Nombre_columna [DATE 'YYYY/MM/DD HH24:mi'],
....
)
```

La función crea_sqlldr_ctl crea el archivo de control con la sintaxis requerida para cargar los datos a la tabla indicada.

Es requisito que exista una tabla con el mismo nombre de la métrica como se describe en la sección siguiente.

La función carga_datos realizará las siguientes tareas:

- Crear el archivo de control para SQL*Loader de Oracle*
- Cargar los datos*
- Revisar si existió algún error en la carga*

La función revisa_log valida si existió algún error en la carga de datos, en caso de ser así vaciará el control file, el datafile y el mensaje de error al archivo sqlldr.err.

Las funciones crea_sqlldr_ctl, revisa_log y carga_datos se encuentran en [carga_datos \(funcion de korn shell incluida en el archivo funciones.ksh\)](#).

3.2.6 Creación de las tablas a almacenar las métricas.

Una vez obtenida las métricas se almacenarán en la tabla correspondiente, la función carga_datos buscará una tabla con el mismo nombre que la métrica para guardar los datos obtenidos.

La tabla donde se almacenarán los datos tendrá la siguiente estructura:

```
create table <metrica>
(
server          char(20),
m_date          date,
metrica_1       number,
...
metrica_n       number
);
```

3.3 Ejemplos de plantillas de programas shells para coleccionar las métricas de desempeño de servidores UNIX.

Como se comentó en la sección 3.2.3 el Ejecutor de Comandos construye el programa shell para la ejecución y formateo de la salida del comando UNIX que mide el desempeño del servidor en un intervalo de tiempo. La forma de construir estas plantillas es utilizar los comandos UNIX para monitorear el desempeño de los servidores de aplicación como se describen a continuación.

3.3.1 Colector de utilización de CPU (cpu.ksh)

Este programa utiliza el comando sar para conocer la utilización del CPU con dos variantes:

`sar -u <intervalo> <repeticiones>` muestra el porcentaje de uso de CPU dividido en los siguientes componentes:

- User: porcentaje de uso ejecutando código de usuario
- Sys: porcentaje de uso ejecutando llamadas al sistema, es decir llamadas al kernel de UNIX. El porcentaje de sys no debe ser mayor que user, si esto llegara a suceder se debe analizar en detalle los procesos que se están ejecutando.
- Wio: porcentaje de espera por IO. Un alto valor indica problemas con el sistema de IO.

```
mxpro:mxpadm 13% sar -u 2 10
SunOS mxpro 5.7 Generic_106541-17 sun4u 10/18/02
10:31:54 %usr %sys %wio %idle
10:31:56 11 25 46 18
10:31:58 12 22 49 18
10:32:00 17 21 42 20
```

`sar -q <intervalo> <repeticiones>` muestra el porcentaje de uso de CPU dividido en los siguientes componentes:

- runq-sz: procesos en la cola de espera
- %runocc: porcentaje de procesos encolados

```
mxpro:mxpadm 1% sar -q 2 10
SunOS mxpro 5.7 Generic_106541-17 sun4u 10/28/02
16:38:56 runq-sz %runocc swpq-sz %swpocc
16:38:58 1.0 49
16:39:00 2.0 49
...
16:39:14 1.5 99
16:39:16 2.0 49
Average 1.2 64
```

Si los procesos en la cola de espera son más que cuatro veces el número de CPU existen problemas con la capacidad del CPU para ejecutar los procesos actuales(runq-sz)

El programa shell se incluye en el apéndice debido a su extensión:
[cpu.ksh](#)

3.3.2 Colector de estado de la memoria virtual (vmstat.ksh)

El comando vmstat es el más completo de las utilerías UNIX, de la salida se puede conocer la utilización de los siguientes componentes:

- CPU
- Memoria
- IO

Para medir el desempeño de CPU las columnas representativas son:

- r: procesos ejecutables esperando por un CPU
- us: porcentaje de uso por procesos usuario
- sy: porcentaje de uso por llamadas internas
- id: porcentaje de disponibilidad de CPU

Si la cola de procesos ejecutables (ready queue) es mayor que cuatro veces el número de CPU's entonces están esperando demasiado tiempo por una porción de CPU.

Si el porcentaje de disponibilidad del CPU (id idle) es menor que 35% también indica insuficiencia para ejecutar procesos actuales

```
mxpro:mxpadm 11% vmstat 5
procs  memory          page          disk          faults          cpu
r  b  w  swap  free  re  mf  pi  po  fr  de  sr  s0  s1  s1  s1  in  sy  cs  us  sy  id
0  2  0  11488  5568  136  246  1370  110  1300  0  1866  4  1  4  1  462  997  2277  12  14  74
1  6  0  19429752  246704  280  8  38062  38  19241  0  3267  0  1  0  1  4208  5263  8092  7  24  69
0  6  0  19429704  243288  247  403  38041  62  22928  0  4635  0  1  1  1  4135  8526  7924  13  23  63
```

En cuanto a la utilización de la memoria, SAP define que el servidor no debe paginar más del 20% de la memoria física del servidor en hora². El comando `vmstat` permite calcular esta proporción de la siguiente forma:

La columna `po` del comando `vmstat` indica la cantidad de Kb paginados por segundo en sistemas Solaris y lista de páginas libres en HPUX.

```
mxpro:mxpadm 11% vmstat 5
procs      memory      page      disk      faults      cpu
r  b  w  swap  free  re  mf  pi  po  fr  de  sr  s0  s1  s1  in  sy  cs  us  sy  id
0  2  0  11488  5568  136  246  1370  110  1300  0  1866  4  1  4  1  462  997  2277  12  14  74
1  6  0  19429752  246704  280  8  38062  38  19241  0  3267  0  1  0  1  4208  5263  8092  7  24  69
0  6  0  19429704  243288  247  403  38041  62  22928  0  4635  0  1  1  1  4135  8526  7924  13  23  63
```

El comando `prtconf` muestra la cantidad de memoria física en MB.

```
mxpro:mxpadm 18% prtconf | more
System Configuration: Sun Microsystems sun4u
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

SUNW,Ultra-Enterprise-10000
packages (driver not attached)
  terminal-emulator (driver not attached)
  deblocker (driver not attached)
  obp-tftp (driver not attached)
  disk-label (driver not attached)
  ufs-file-system (driver not attached)
```

Con estos dos datos se puede calcular la proporción de paginación en el sistema de la siguiente forma:

$$\text{Porcentaje de paginación} = \frac{(\text{vmstat.po})(60)(60)}{\text{prtconf.MemoriaFisica} / 1024}$$

² El diagnóstico del desempeño de la memoria no es claro en la documentación de Sun y en algunos casos son contradictorios. Por ejemplo, la interpretación típica que la columna `sr` (scan rate) del comando `vmstat` es indicador de problemas con la capacidad en memoria con la versión de Solaris 2.7 y algoritmo `priority paging` ya no aplica (*Nota OSS 200057 Enable Priority Paging in Solaris*). Otras métricas como la memoria libre, (columna `free` del comando `vmstat` o `freemem` del comando `sar -r`) podrían utilizar para determinar si un equipo cuenta con suficiente memoria, sin embargo esto no aplica debido que en las nuevas versiones Solaris toma memoria disponible como cache de para File Systems. Por esta razón se utiliza la recomendación de SAP documentada en *SAP R/3 Performance Optimization: The official SAP guide*, Thomas Schneider. Pág 33.

Por la parte de sistema de IO, el comando vmstat muestra cuantos procesos se encuentra bloqueados en espera por IO

- b: Procesos en espera de IO

Procesos bloqueados es signo que existen procesos en ejecución en espera por respuesta del sistema de IO, comúnmente de los discos.

Si el número de procesos bloqueados se aproxima o excede el número que existe en la cola de procesos ejecutables (ready queue) es un indicador de un problema en la configuración de los discos o que existen procesos mal afinados que están ejecutando una gran cantidad de operaciones IO.

```
mxpro:mxpadm 11% vmstat 5
procs      memory          page          disk          faults        cpu
r  b w  swap free re  mf pi po fr de sr s0 s1 s1 s1  in  sy  cs us sy id
0 2 0  11488 5568 136 246 1370 110 1300 0 1866 4 1 4 1 462 997 2277 12 14 74
1 6 0 19429752 246704 280 8 38062 38 19241 0 3267 0 1 0 1 4208 5263 8092 7 24 69
0 6 0 19429704 243288 247 403 38041 62 22928 0 4635 0 1 1 1 4135 8526 7924 13 23 63
```

El programa shell se incluye en el apéndice debido a su extensión:
[vmstat.ksh](#)

3.3.3 Colector del desempeño y utilización de discos en Solaris (disk.ksh)

El comando `iostat -x` permite conocer el rendimiento, utilización y tiempo de respuesta de los discos del sistema. Las columnas importantes en la salida de `iostat` son³:

- `Kr/s`: Kb leídos por segundo. Idealmente no debe ser mayor que 200Kb/s
- `Svc_t`: Tiempo de respuesta de los discos. Para discos con cache (EMC2 o XP) no debe ser mayor de 20 ms en arreglo de discos sin cache es el umbral es 40 ms.
- `%b`: Porcentaje de ocupación de los discos. En general se recomienda no sobrepasar el 40%

```
mxpro:mxpadm 10% iostat -x 5
extended device statistics
device  r/s  w/s  kr/s  kw/s  wait  actv  svc_t  %w  %b
sd0     0.7  3.8  19.3  32.1  0.1  0.1  37.4  0  2
sd1     0.2  0.7  12.0  8.9  0.0  0.0  12.5  0  0
sd15    0.5  3.8  18.1  32.1  0.1  0.1  36.8  0  2
sd16    0.2  0.7  12.0  9.0  0.0  0.0  12.3  0  1
sd30    0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
sd46    0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
sd80    0.5  0.4  17.3  4.9  0.0  0.0  23.8  0  1
sd81    11.2  0.2  297.3  2.8  0.0  0.2  18.0  0  13
sd82    11.2  0.1  297.7  2.4  0.0  0.2  18.1  0  13
sd83    11.2  0.1  297.4  2.5  0.0  0.2  17.8  0  13
sd84    11.2  0.1  297.1  2.4  0.0  0.2  17.8  0  13
sd85    11.2  0.1  297.3  2.4  0.0  0.2  17.6  0  13
sd86    11.2  0.1  297.2  2.3  0.0  0.2  17.6  0  13
sd87    11.2  0.1  297.5  2.3  0.0  0.2  17.6  0  12
sd88    11.2  0.1  297.4  2.3  0.0  0.2  17.7  0  13
...
sd264   0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
sd265   0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
sd266   0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
sd267   0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
sd268   0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
sd269   0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
nfs1    0.0  1.9  0.0  59.7  8.5  0.0  4517.2  1  1
nfs2    0.0  0.0  0.0   0.0  0.0  0.0  0.0  0  0
```

³ Un artículo sumamente útil para profundizar en el análisis de discos se encuentra en la siguiente dirección de internet <http://www.itworld.com/Comp/3743/UIR960601perf/> How disks really work? Adrian Cockcroft

El programa shell se incluye en el apéndice debido a su extensión:
[disk.ksh](#)

Objetivo de la afinación en discos se ejemplifica a continuación:

Svct <= 20 ms.

%b <= 20

krps <= 400

3.3.4 Colector de rendimiento de discos HPUX-Solaris (sard.ksh)

Como una extensión al análisis realizado con el comando iostat, el comando sar -d permite analizar el desempeño de los discos.

Incluir la salida del comando

Las columnas a analizar son:

avque – número de solicitudes en espera de ser atendidas por el dispositivo < 2

Avwait – tiempo promedio en ms que una petición de IO espera por el acceso a un dispositivo < 20

%Busy – utilización del dispositivo < 40

r+w/s – operaciones de IO realizadas por segundo < 70

3.4 Ejemplos de plantillas de programas shells para coleccionar el desempeño de servidores Oracle.

3.4.1 Colector de eventos de espera de la base de datos (System_event.ksh)

Dentro de los ambientes SAP lo más frecuente es encontrar una proporción muy alta del tiempo de base de datos (>40%). Una vez identificado esto el siguiente paso es analizar como esta compuesto el tiempo de respuesta por proceso en la base de datos. Los eventos de espera Oracle son una herramienta de análisis poderosa para determinar esta composición.

La vista del diccionario de datos v\$system_event⁴ almacena el total de la espera para cada uno de los eventos. La estructura de esta vista esta definida como:

```
SQL> desc v$system_event
Name                               Null?    Type
-----
EVENT                              VARCHA2(64)
TOTAL_WAITS                         NUMBER
TOTAL_TIMEOUTS                     NUMBER
TIME_WAITED                         NUMBER
AVERAGE_WAIT                       NUMBER
```

Los eventos más importantes a tomar atención⁵ son:

- **db file sequential read** – espera por lectura de un bloque de dato del disco. La expectativa de este valor debe ser menor a 15 ms para dispositivos con cache y 20 ms para dispositivos sin cache.
- **db file scattered read** – espera por lectura múltiples bloques de datos del disco. Este evento indica que se están realizando lecturas completas de tablas sin utilizar índices (full table scans).
- **log file sync** – espera por que el LGWR haya grabado todos los datos del redo log buffer a los online redo logs cuando se ejecuta un COMMIT o ROLLBACK. Este valor no debe exceder 15 ms para para dispositivos con cache y 40 ms para dispositivos sin cache.
- **buffer busy waits** – espera por un bloque debido que esta siendo importado o modificado por otra sesión. Este valor no debe exceder 15 ms para para dispositivos con cache y 40 ms para dispositivos sin cache.
- **undo block** – no existen suficientes segmentos de rollback

⁴ La vista v\$session_event permite analizar los eventos de espera por sesión, para un análisis detallado es una herramienta de ayuda adicional.

⁵ La nota OSS 619188 FAQ: Oracle Waits Events contiene una descripción detallada de estos eventos.

- **write completed waits** – espera por que el DBWR termine la escritura al disco. Este evento no debe aparecer en los 10 eventos más importantes.
- **free buffer waits** – espera mientras el DBWR ha terminado de escribir los bloques sucios al disco de tal forma que estos puedan ser reemplazados con nuevos bloques.
- **enque** – tiempo de espera por un lock de base de datos. Si este evento aparece frecuentemente es necesario revisar la lógica de programación,

Para los eventos db file sequential read, buffer busy waits y log file sync es necesario calcular el tiempo promedio de espera, esto se obtiene por la proporción del tiempo esperado entre el numero de esperas ($\text{time_waited} / \text{total_waits}$) y dado que este valor esta representado en centésimas de segundo (cs) se multiplica por 10 para representar milisegundos (ms). Es recomendable tomar dos muestras de los valores de la vista v\$system_event y realizar el calculo considerando las diferencias de estas 2 columnas.

El script system_event.ksh coleccionará los valores de la vista v\$system_event para su posterior análisis.

3.4.2 Colector de estadísticas de uso y rendimiento de datafiles (datafiles.ksh)

La vista de Oracle v\$file_stat almacena información sobre las operaciones de IO realizadas a nivel de base de datos. Por otra parte la vista dba_data_files contiene la relación de los datafiles con los tablespaces así como su tamaño.

```
SQL> desc v$filestat
```

Name	Null?	Type
FILE#		NUMBER
PHYRDS		NUMBER
PHYWRTS		NUMBER
PHYBLKRD		NUMBER
PHYBLKWRT		NUMBER
READTIM		NUMBER
WRITETIM		NUMBER
AVGIOTIM		NUMBER
LSTIOTIM		NUMBER
MINIOTIM		NUMBER
MAXIORTM		NUMBER
MAXIOWTM		NUMBER

```
SQL> desc dba_data_files
```

Name	Null?	Type
FILE_NAME		VARCHAR2(513)
FILE_ID		NUMBER
TABLESPACE_NAME		VARCHAR2(30)
BYTES		NUMBER
BLOCKS		NUMBER
STATUS		VARCHAR2(9)
RELATIVE_FNO		NUMBER
AUTOEXTENSIBLE		VARCHAR2(3)
MAXBYTES		NUMBER
MAXBLOCKS		NUMBER
INCREMENT_BY		NUMBER
USER_BYTES		NUMBER
USER_BLOCKS		NUMBER

De estas vistas la información relevante es la siguiente:

Promedio de lectura (ms) - La eficiencia de las lecturas realizadas sobre el datafile, vfile_stat.readtim/v$file_stat.phyrds$ la cuál no debe ser mayor a 15 ms para dispositivos con cache y 20 ms para dispositivos sin cache.⁶

Total de operaciones – La proporción de lecturas y escrituras de un datafile respecto a total. $(v$file_stat.phyrds + v$file_stat.phywrts * 2) / \text{sum}(v$file_stat.phyrds + v$file_stat.phywrts * 2)$, estas no deben ser mayor al 6%

Y la proporción del disco más utilizado respecto al menor no debe ser mayor al 30%

⁶ Esta métrica esta relacionado directamente con el evento db file sequential real y con el tiempo de respuesta del disco medido desde UNIX con el comando `sar -d`

3.5 Monitores de desempeño

La segunda función del SMDSA es la explotación de los datos colectados. En esta sección se presentan los programas que permiten la consulta de los datos de una manera útil para poder analizar y diagnosticar el desempeño de los servidores de aplicación.

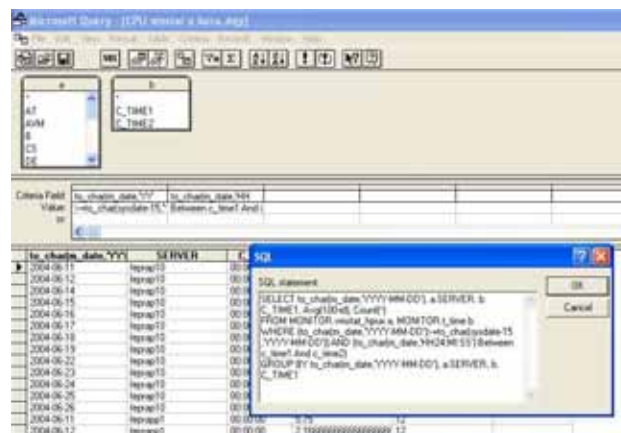
3.5.1 Presentación de datos sobre el tiempo y de manera global (todos los servidores de aplicación)

Las dos características identificadas como necesarias para la presentación de los datos son:

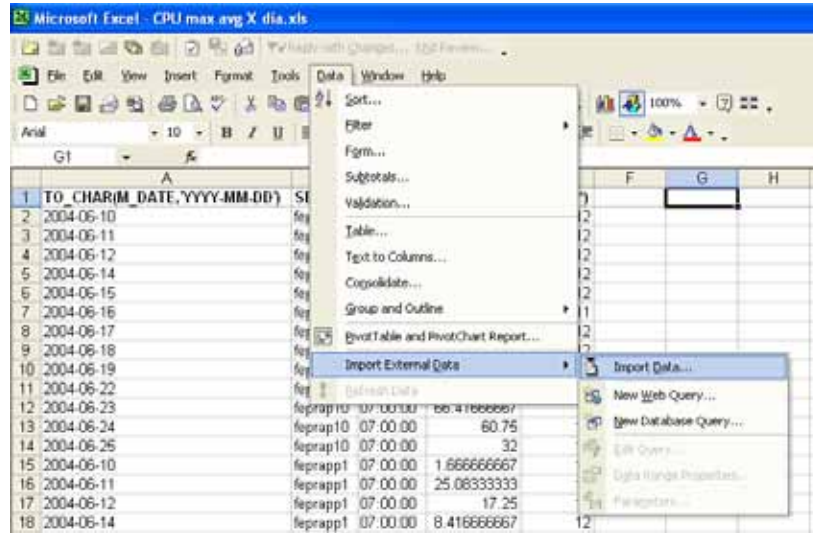
- **Vista detallada por hora** - Se visualizan los datos de un día detallando intervalos de tiempo por hora iguales. (15 minutos, 30 minutos o 1 hora) de todos los servidores de aplicación de una métrica en específico.
- **Vista detallada por día** - Se visualiza por día todos los servidores de aplicación con sus datos promedio de una métrica en específico.
- **Vista global de métricas** - Se visualiza los datos de todas las métricas de desempeño de un solo servidor de aplicación.
- **Cálculo automático de las estadísticas de Oracle** – Las estadísticas de rendimiento de Oracle requieren cálculos tomando dos muestras de datos, los programas de monitoreo automatizan estas cálculos facilitando el análisis.
- **Gráficas que representen los valores analizados** – Esenciales para mostrar visualmente el comportamiento del sistema.

Se desarrollaron los monitores de desempeño en modo carácter con korn shell.

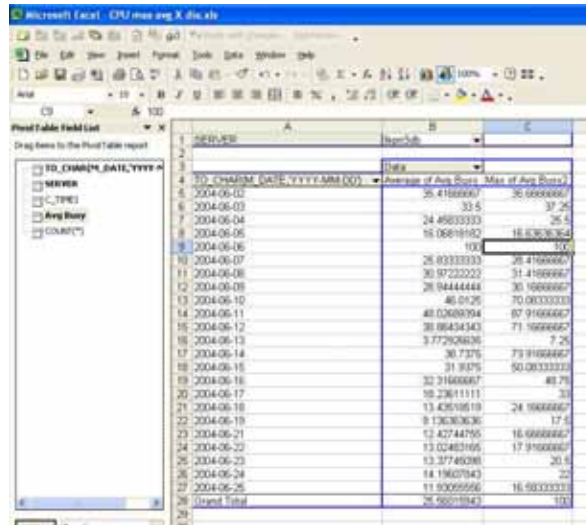
Para tener una presentación gráfica utilizo MsQuery para acceder los datos vía ODBC y Excel para graficar los datos.



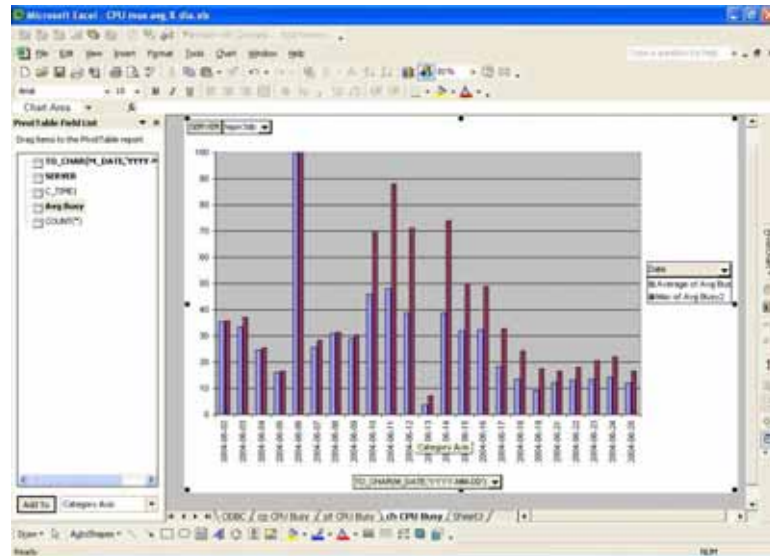
Se han creado formatos predefinidos en Excel que importar los datos para su posterior graficación. Para importar los datos de una sentencia SQL almacenada en MsQuery se debe acceder por Excel Data->Import Data ... -> seleccionar la sentencia SQL almacenada.



Posteriormente se utiliza una tabla pivote para presentar los datos de manera organizada.



La tabla pivote tiene asociada un gráfico para mostrar los valores y permite modificar los datos del gráficos de manera dinámica.



3.5.2 Incorporación de alarmas y advertencias de rendimientos

Se incorporan umbrales deseados de rendimiento para cada métrica analizada, cuando se visualizan los datos estos pueden caer en 3 categorías:

- **Normal:** el rendimiento esta dentro de los umbrales deseados.
- **Advertencia:** el rendimiento comienza a degradarse, requiere la atención del administrador.
- **Alarma:** el equipo se encuentra saturado y requiere la implementación inmediata de acciones de afinación.

Estos umbrales se encuentran configurados en el archivo alarmas_wa.lst en el formato:

```
Id_alarma|umbral_alarma|umbral_warning|descripción
ID|30|40|CPU_Idle
US|50|40|CPU_User
SY|50|40|CPU_Sys
IO|40|20|CPU_Wio
EQ|14|12|ENQUE_Process
REA|15|10|Ready_process
BLK|20|15|Block_process
MEM|60|20|%Paging_ram
SVCT|20|15|disk_service_time
PCTB|40|30|disk_busy%
RWS|70|55|disk_busy%
AVQUE|2|1|disk_busy%
DFSR|15|10|system_event
BBW|15|10|system_event
LFS|20|15|system_event
```

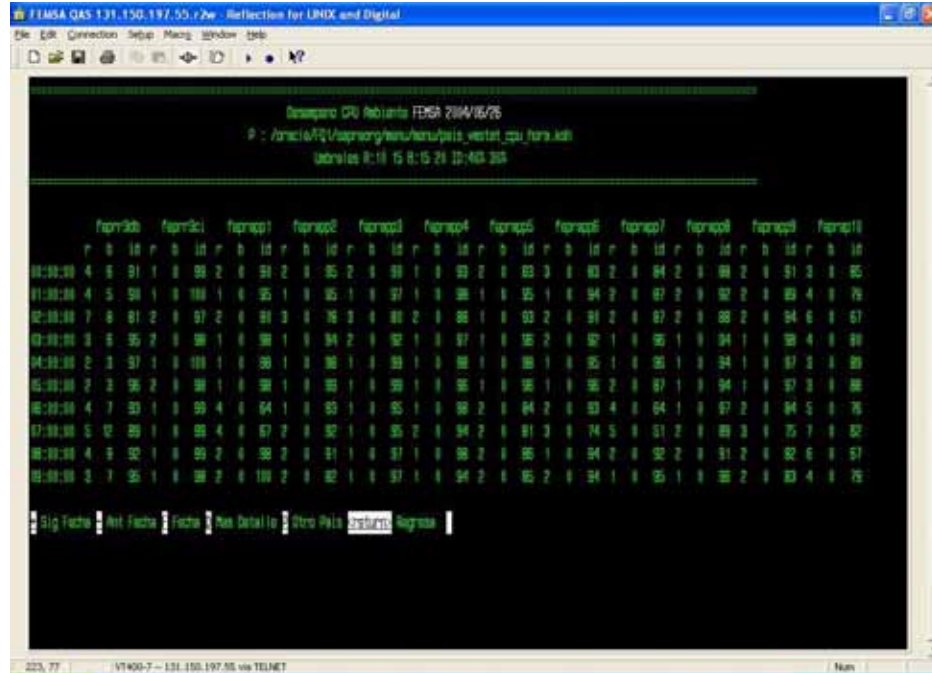
Estos umbrales son leídos al ejecutar un monitor de desempeño por la funcion de shell lee_alarmas_wa en el archivo funciones.ksh.

3.5.3 Monitor rendimiento de cpu (vmstat)

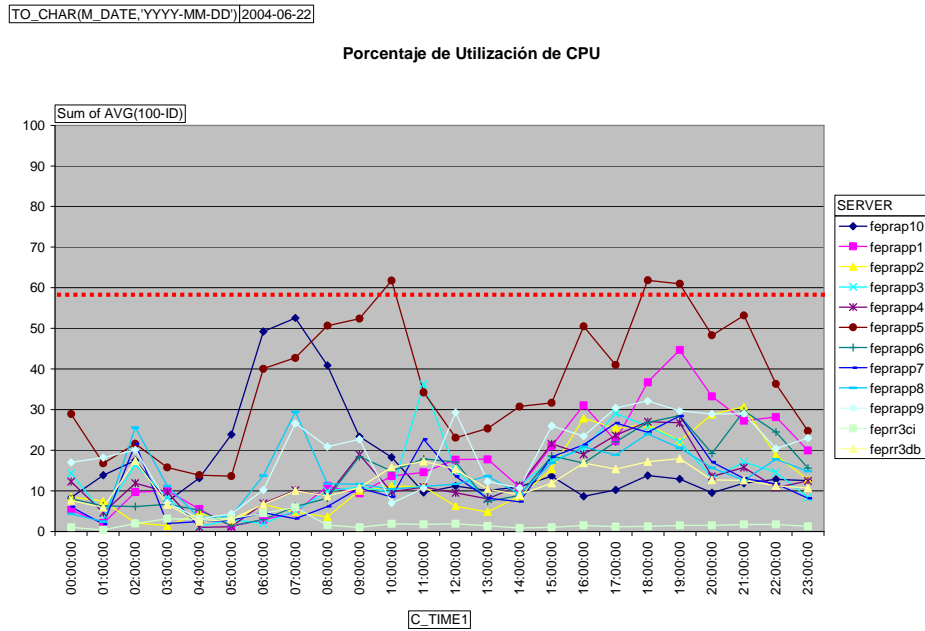
Se desarrollaron 2 monitores de desempeño de CPU de acuerdo al comando usado para su recolección de datos.

El primero explota los datos colectados por [vmstat.ksh](#) y con el criterio de [Vmstat \(columnas r, id \)](#), se presenta todos los servidores de aplicación de un ambiente R/3. A continuación se muestra las salidas por hora y por día.

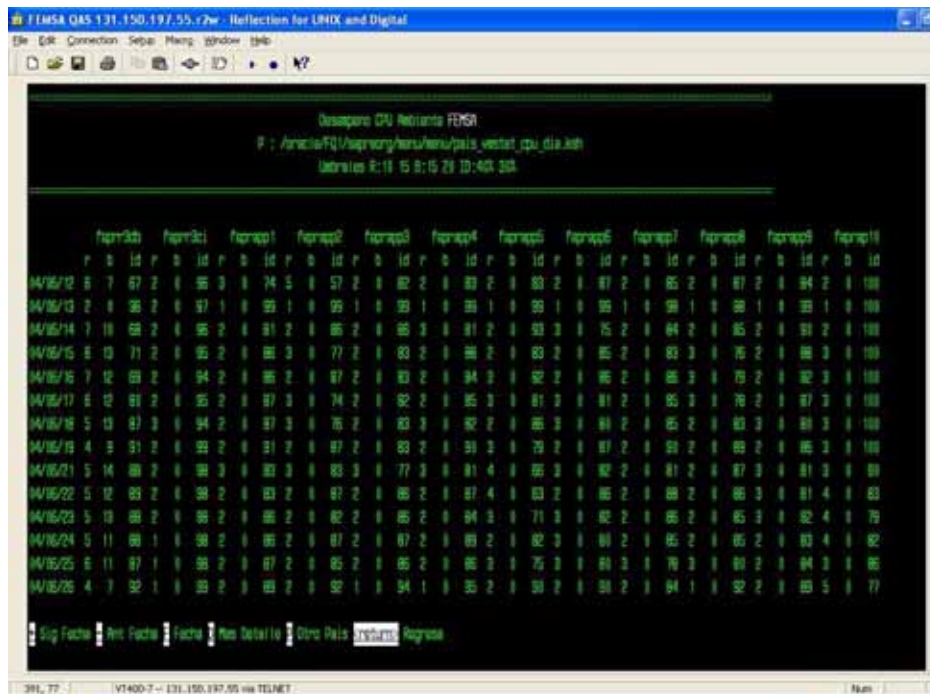
Monitor por Hora: [pais_vmstat_memoria_hora.ksh](#)



La presentación gráfica de estos datos se genera con el formato de Excel CPU avg X hora.xls. Se obtiene el siguiente gráfico:



El monitor en modo carácter se presenta los valores máximos y mínimos de utilización por hora es el siguiente:

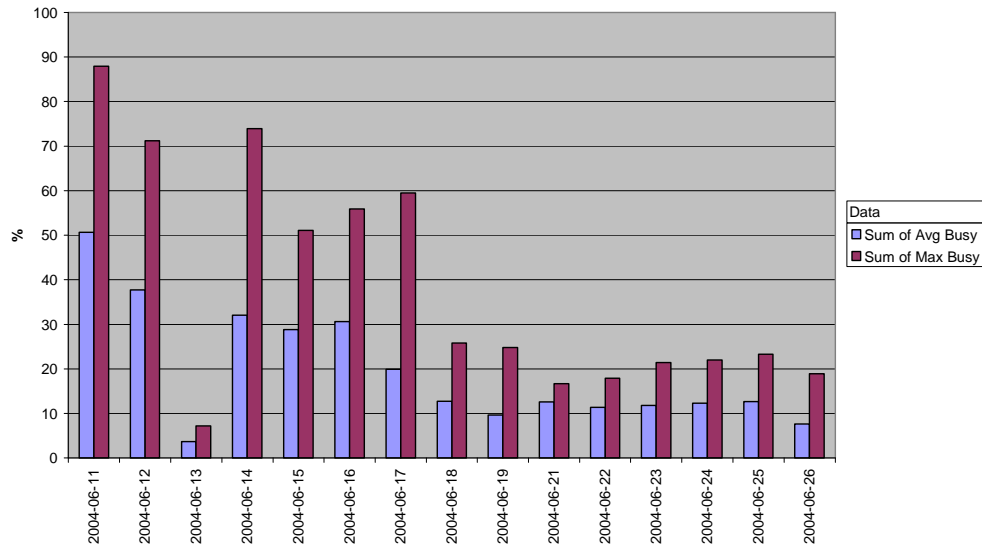


La presentación gráfica de estos datos se genera con el formato de Excel CPU max avg X dia.xls. Se obtienen las siguientes representaciones gráficas:

Por un solo servidor de aplicación:

Server|fepr3db

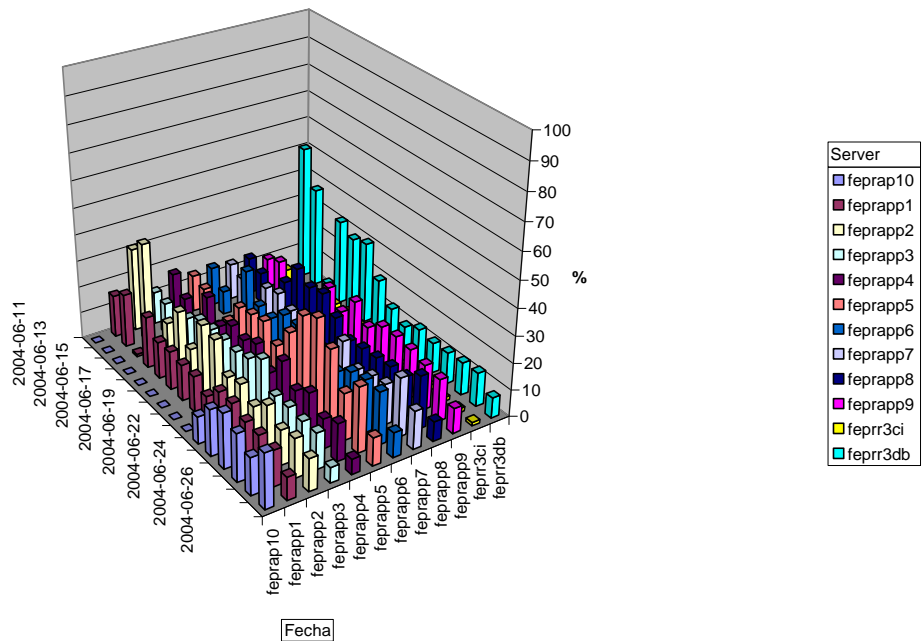
Uso de CPU x Hora (Avg y Max)



Promedio de utilización de todos los servidores de aplicación:

Promedio de Utilización de CPU

Average of Avg Busy2



3.5.4 Monitor rendimiento de cpu (sar -u y sar -q)

El segundo monitor explota los datos colectados por [cpu.ksh](#) y con los criterios de [Sar -u \(user, sys, wio, idle \)](#) y [Sar -q \(runq-sz, %runocc \)](#), se presenta cada uno de los servidores de aplicación de un ambiente R/3. Las reglas de parametrización de alarmas se definen en [alarmas_wa.lst](#).

A continuación se presentan las salidas por hora y por día.

Monitor por Hora: [pais_sar_cpu_hora.ksh](#)

```

=====
CPU por Pais mexico 2003/02/21
P: /oracle/MXD/sapreorg/tesis/menu/pais_sar_cpu_hora.ksh
Umbral US:20 40 SY:20 40 IO:20 40 ID:40 20 EQ:2 4
=====
      mxpro      mxapp1      mxapp2      mxqa      mxapp4      mxapp5
  us sy io  id eq us sy io  id eq us sy io  id eq us sy io  id eq us sy io  id eq us sy io
00:00 28 28 30 14 1 0 0 0 99 0 1 1 2 96 0 18 4 3 77 0 2 1 0 97 0 11 1 1 87 0 18 4
01:00 47 40 7 6 3 7 1 0 0 92 0 0 1 2 97 0 14 2 2 82 0 2 1 0 98 0 25 0 1 73 0 14 2
02:00 23 14 2 61 0 0 0 0 99 0 1 1 1 98 0 14 1 3 82 1 4 0 0 95 0 9 0 1 89 0 14 1
03:00 5 2 1 92 1 0 0 0 99 0 2 1 1 96 0 19 1 3 76 0 2 1 0 97 0 0 0 2 90 0 19 1
04:00 3 8 16 72 0 0 1 0 99 0 0 1 0 99 0 13 1 2 84 0 0 0 0 99 0 3 1 1 86 0 13 1
06:00 5 6 14 76 0 0 0 0 99 0 2 1 2 96 0 19 2 3 77 0 1 0 0 98 0 2 1 2 96 0 19 2
07:00 11 17 27 44 1 6 1 1 92 0 6 1 1 92 0 25 3 7 65 0 2 1 0 96 0 5 1 1 94 0 25 3
08:00 20 30 42 8 1 5 1 0 94 0 49 4 2 46 1 15 2 10 72 0 17 2 0 81 0 14 1 1 84 0 15 2 1
08:00 21 39 36 4 3 8 1 1 89 0 33 2 5 59 1 21 3 4 72 0 12 2 0 86 0 9 2 1 88 0 21 3
10:00 26 42 30 2 2 18 2 0 90 0 23 4 2 72 0 24 3 1 72 0 10 2 0 88 0 16 1 2 82 0 24 3
11:00 17 34 43 6 1 7 2 1 90 0 14 3 4 79 1 21 3 2 74 1 13 2 1 84 0 6 1 1 93 0 21 3
12:00 20 34 40 6 2 7 1 0 92 0 19 4 32 45 0 29 3 2 66 1 10 2 1 87 0 22 1 1 75 0 29 3
=====
Sig Fecha  Ant Fecha  Fecha  Mas Detalle  Otro Pais  <return> Regresa
    
```

La misma información se presenta agrupada por día: [pais_sar_cpu_dia.ksh](#)

```

=====
CPU por Pais mexico 2003/02/21
P: /oracle/MXD/sapreorg/tesis/menu/pais_sar_cpu_dia.ksh
Umbral US:20 40 SY:20 40 IO:20 40 ID:40 20 EQ:2 4
=====
      mxpro      mxapp1      mxapp2      mxqa      mxapp4      mxapp5
  us sy io  id eq us sy io  id eq us sy io  id eq us sy io  id eq us sy io  id eq us sy io
03/01/21 13 18 16 53 1 7 1 0 92 0 9 1 1 88 0 16 3 4 78 1 9 1 1 90 0 7 1 1 91
03/01/22 13 19 13 55 1 8 1 1 92 0 11 1 1 87 0 9 2 1 87 0 9 1 0 90 0 6 1 1 92
03/01/23 19 26 15 40 2 9 1 0 89 0 7 2 1 90 0 6 2 1 91 0 9 2 1 88 0 12 1 1 86
03/01/24 24 26 14 35 2 11 1 1 87 0 10 1 1 88 0 23 2 1 73 1 13 1 0 86 0 7 1 1 91
03/01/25 11 13 10 65 1 6 1 0 93 0 8 1 1 89 0 8 2 1 89 0 3 1 0 96 0 2 1 1 96
03/01/26 7 7 2 85 1 2 1 0 97 0 1 1 1 97 0 6 2 1 91 0 0 0 0 99 0 0 1 1 98
03/01/27 15 17 15 53 1 7 1 0 92 0 12 2 1 85 0 14 3 2 80 0 8 1 0 91 0 10 1 1 88
03/01/28 19 25 21 35 2 7 1 1 91 0 10 2 1 87 0 18 2 1 79 0 10 1 1 88 0 13 1 1 85
03/01/29 21 31 24 25 2 8 1 1 91 0 9 1 1 88 0 13 2 1 84 0 12 1 1 86 0 19 2 2 78
03/01/30 20 27 22 30 2 9 1 1 89 0 11 1 1 86 0 33 2 2 63 0 12 1 0 86 0 12 1 2 85
03/01/31 22 34 23 21 2 12 2 0 86 0 9 1 1 88 0 9 2 1 87 0 13 1 0 85 0 9 1 1 88
03/02/01 14 22 22 42 1 7 1 2 90 0 14 2 1 82 0 9 2 1 88 0 6 1 0 92 0 4 1 1 93
03/02/02 10 11 3 78 1 1 0 1 90 0 1 1 1 97 0 3 2 1 94 0 3 1 0 96 0 1 1 1 97
03/02/03 14 25 30 31 2 11 1 1 87 0 10 2 2 86 0 16 3 1 80 0 14 1 1 84 0 10 1 1 87
03/02/04 20 30 34 15 2 13 1 1 85 0 9 1 2 88 0 10 2 1 87 0 12 1 0 86 0 9 1 1 89
03/02/05 22 33 19 26 2 6 1 0 93 0 5 1 1 93 0 10 1 1 88 0 2 1 0 97 0 3 1 1 95
    
```

3.5.5 Monitor de desempeño de memoria

El monitor global de memoria presenta la paginación de los servidores con respecto a la memoria física.

El monitor explota los datos colectados por [vmstat.ksh](#) de acuerdo a los criterios de [Análisis de rendimiento de memoria de acuerdo con SAP con vmstat](#).

Monitor por Hora: [pais_vmstat_memoria_hora.ksh](#)

```

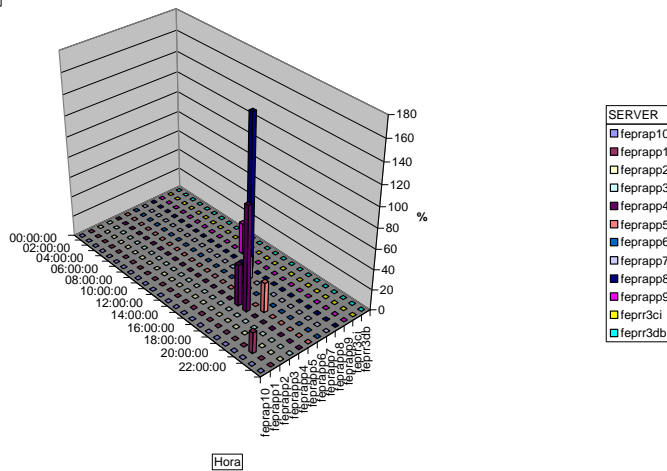
Paginacion de Memoria Ambiente FMSA 2004/06/24
P : /oracle/TQ1/saprcng/mnu/mnu/pais_vmstat_memoria_hora.ksh
Umbral MEM : 20% 60%
-----
fepr3db fepr3ci feprapp1 feprapp2 feprapp3 feprapp4 feprapp5 feprapp6 feprapp7 feprapp8 feprapp9 feprapp10
66560 66560 8192 8192 8192 8192 8192 8192 8192 8192 8192 24576
Pout Pagn Pout Pagn Pout Pagn Pout Pagn Pout Pagn Pout Pagn Pout Pagn Pout Pagn Pout Pagn Pout Pagn
/skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb /skb
14:00:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 132 23 0 0 0 0 0 0
15:00:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16:00:00 0 0 0 0 0 0 0 0 0 237 41 0 0 0 0 0 0 0 0 0 0 0 0
17:00:00 0 0 0 0 0 0 0 0 0 589 103 0 0 0 0 0 0 0 0 0 0 0 0
18:00:00 0 0 0 0 0 0 0 0 0 0 172 30 0 0 0 0 0 0 0 0 0 0 0
19:00:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
20:00:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
21:00:00 0 0 0 0 120 21 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22:00:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
23:00:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-----
| Sig Fecha | Ant Fecha | Fecha | Mas Detalle | Otro Pais | return | Regresa |
    
```

La representación gráfica de estos datos se genera con la plantilla paging x hora hpux.xls

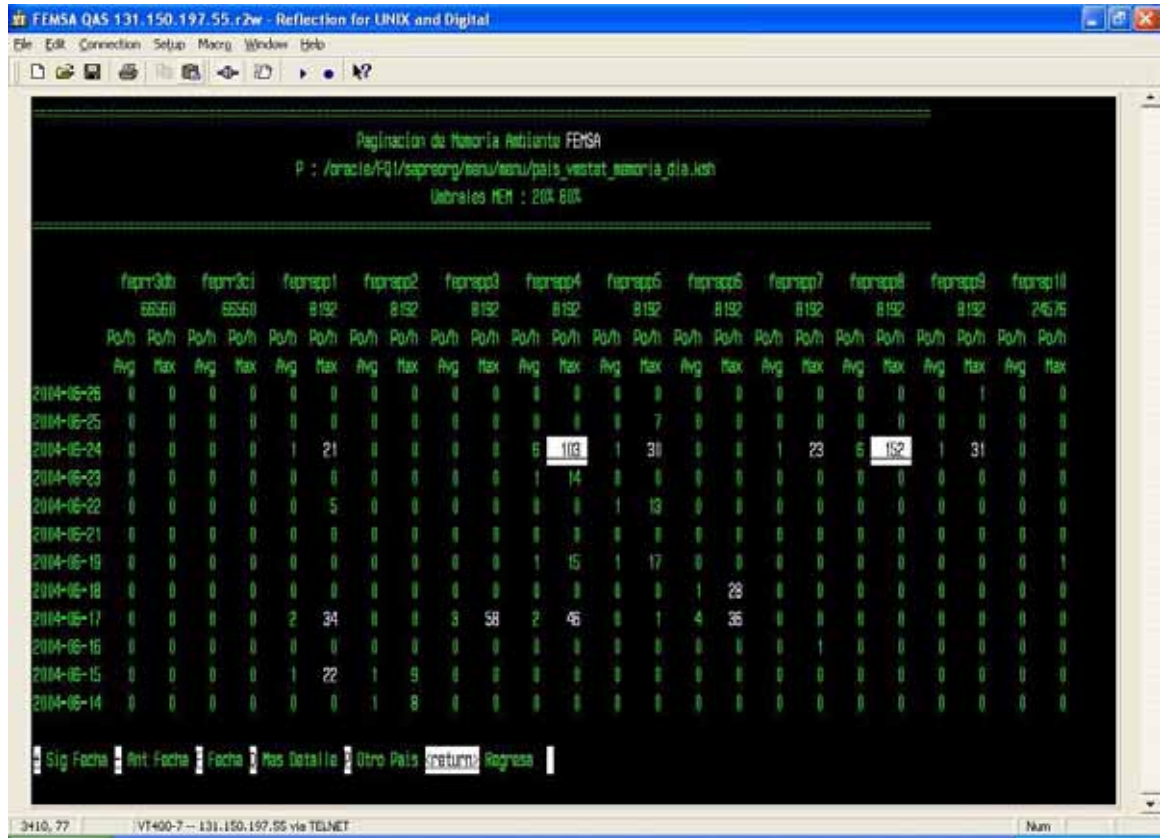
Date|2004-06-24

Págination hr / Memoria Fisica

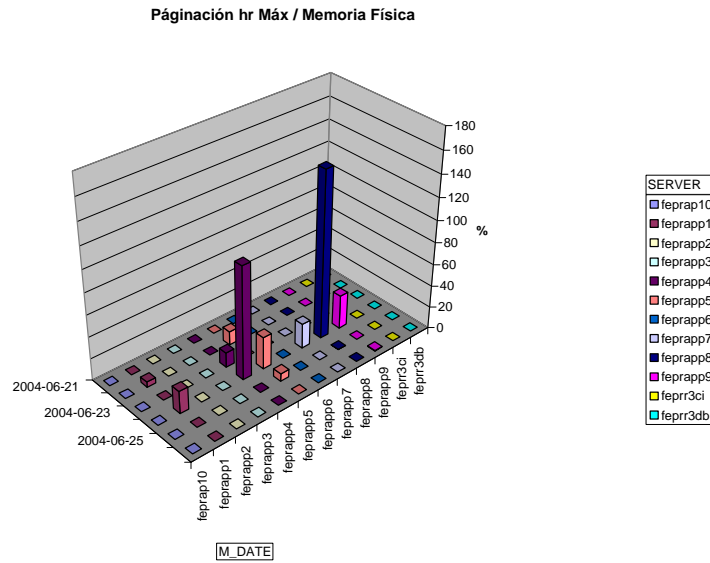
Average of Pg / Mem Phy



Monitor por Día: [pais_vmstat_memoria_dia.ksh](#)



La representación gráfica de estos datos se genera con la plantilla paging x día hpux.xls



Para el funcionamiento correcto de los monitores de memoria es necesario parametrizar los siguientes archivos:

- [países_detalle.lst](#) contiene la lista de servidores que componen todo el sistema R/3
- Tabla Oracle: servers_memoria con la siguiente estructura

```
SQL> desc servers_memoria
Name                               Null?    Type
-----
SERVER                              CHAR(20)
MEMORIA_MB                          NUMBER
```

Donde server es el nombre del servidor correspondiente al archivo [memoria.lst](#) y memoria_mb la cantidad de memoria física. [memoria.lst](#): contiene también la memoria física de cada servidor, únicamente para efectos de imprimir en pantalla no para cálculo.

3.5.6 Construcción del mapa de discos Solaris-Volume Manager-EMC2

El monitoreo de desempeño de discos requiere conocer la relación que existe entre dispositivos físicos y dispositivos lógicos, es decir que datos (File Systems, datafiles de la base de datos, swap, etc.) residen en ellos.

Hay diferentes opciones de manejo de discos en UNIX: format, disk suit y volume manager. Dependiendo de la configuración instalada será la forma de construir el mapa de discos.

De acuerdo al ambiente del caso práctico se muestra el procedimiento de mapeo de discos con Veritas Volume Manager – Solaris - Oracle⁷.

El comando vxprint muestra la construcción de volúmenes de un sistema:

```
root-mxpro- / >vxprint
Disk group: mxdb
```

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
dg	mxdb	mxdb	-	-	-	-	-	-
dm	mxdb01	c2t0d105s2	-	17671680	-	-	-	-
dm	mxdb02	c2t0d106s2	-	17671680	-	-	-	-
dm	mxdb03	c2t0d107s2	-	17671680	-	-	-	-
dm	mxdb04	c2t0d108s2	-	17671680	-	-	-	-
dm	mxdb05	c2t0d27s2	-	17671680	-	-	-	-
dm	mxdb06	c2t0d28s2	-	17671680	-	-	-	-
dm	mxdb07	c2t0d29s2	-	17671680	-	-	-	-
dm	mxdb08	c2t0d30s2	-	17671680	-	-	-	-
dm	mxdb09	c2t0d31s2	-	17671680	-	-	-	-
dm	mxdb10	c2t0d32s2	-	17671680	-	-	-	-
dm	mxdb11	c2t0d33s2	-	17671680	-	-	-	-
dm	mxdb12	c2t0d34s2	-	17671680	-	-	-	-
dm	mxdb13	c2t0d35s2	-	17671680	-	-	-	-
dm	mxdb14	c2t0d36s2	-	17671680	-	-	-	-
dm	mxdb15	c2t0d37s2	-	17671680	-	-	-	-
dm	mxdb16	c2t0d38s2	-	17671680	-	-	-	-
dm	mxdb17	c2t0d39s2	-	17671680	-	-	-	-
dm	mxdb18	c2t0d40s2	-	17671680	-	-	-	-
dm	mxdb19	c2t0d41s2	-	17671680	-	-	-	-
dm	mxdb20	c2t0d42s2	-	17671680	-	-	-	-
dm	mxdb21	c2t0d43s2	-	17671680	-	-	-	-
dm	mxdb22	c2t0d44s2	-	17671680	-	-	-	-
v	ora_MXP_sapdata1	fsgen	ENABLED	235520000	-	ACTIVE	-	-
p1	ora_MXP_sapdata1-01	ora_MXP_sapdata1	ENABLED	235530560	-	ACTIVE	-	-
sd	mxdb06-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb07-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb08-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb09-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb10-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb11-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb12-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb13-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb14-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb15-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb16-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb17-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-
sd	mxdb18-01	ora_MXP_sapdata1-01	ENABLED	11215680	0	-	-	-

⁷ En un ambiente SAP-UNIX Sun-Oracle si no existiera un manejador de volúmenes debe recomendarse inmediatamente su adquisición e instalación.


```
sd mxdb19-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
sd mxdb20-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
sd mxdb21-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
sd mxdb22-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
sd mxdb01-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
sd mxdb02-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
sd mxdb03-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
sd mxdb04-01 ora_MXP_sapdata1-01 ENABLED 11215680 0 - - -
```

El extracto de la salida de vxprint muestra que el volume group mxdb esta construido por 22 discos c2t0d27s2, c2t0d28s2, .., c2t0d44s2.

Dentro de este volume group se encuentra definido un volumen lógico (logical volume) donde se estarán montando los file systems del sistema operativo. La salida muestra el volumen lógico ora_MXP_sapdata1, el cuál estará montado el file system /oracle/MXP/sapdata1, esto se puede validar con el comando UNIX df -k.

```
root-mxpro- / >df -k
Filesystem          kbytes  used  avail capacity  Mounted on
/dev/vx/dsk/mxdb/ora_MXP_sapdata1
                    117760000 116604128 1146912   100%    /oracle/MXP/sapdata1
```

Veritas Volume Manager utiliza nomenclatura descriptiva cXdXtX de los dispositivos de discos del sistema, esta nomenclatura debe ser traducida al formato sdXXX. Esto puede ser realizado por el siguiente script ksh:

```
iostat -ex | awk '{print $1 }' > sdxxx.out
iostat -nx | awk '{print $11"S2"}' > cxtxdx.out
paste sdxxx.out cxtxdx.out > relacion.out
```

La salida relacion.out contiene 2 columnas con el formato sdXXX y su correspondiente cXtXdXsX.

Toda esta lógica esta programado en el script [Mapa discos.sh](#).

Este script debe ejecutarse como root en el equipo donde se desea obtener el mapa de discos.

La salida del comando se almacenará en la tabla mapa_discos, antes de cargar la información se elimina todos los si es que existiera algún dato previo con la siguiente instrucción SQL:

```
delete from mapa_discos
where server = 'SERVER';
commit;
```

La salida se almacena el archivo mapa_discos.out con el formato siguiente:

```
mxpro|export|export01|c0t2d0s2|sd2
mxpro|export|export02|c0t3d0s2|sd3
mxpro|export|export03|c0t4d0s2|sd4
```

Posteriormente debe cargarse el archivo a la base de datos, el archivo de control mapa_discos.ctl contine la instrucciones para esto y la instrucción sqlldr invoca a la utilería sql loader de Oracle:

```
sqlldr monitor/monitor control=mapa_discos.ctl log=mapa_discos.log
```

Por otra parte los dispositivos de Oracle pueden ser relacionados con el siguiente script sql.

```
col name for a45
col tbsname for a12
col phyblkrd for 999,999,999
col mb for 99,999
```

```
set heading on;
set pagesize 1000;
set linesize 150
```

```
select tbs.name tbsname, df.name, phyrds rds, phywrts*2 wrs, phyrds+phywrts*2 tot,
bytes/1024/1024 mb, phyrds/readtim rdms, maxiortm
from v$filestat fs, v$datafile df, v$tablespace tbs
where fs.file# = df.file#
and tbs.ts# = df.ts#
order by tbsname, df.name;
```

La salida del script mostrara los datafiles configurados en el sistema:

TBSNAME	NAME	RDS	WRS
TOT	MB	RDMS	MAXIORTM

PSAPBTAB1D	/oracle/FP1/sapdata60/btab1d_4/btab1d.data4	1900722	626
1901348	2,048	2.53192612	140
PSAPBTAB1D	/oracle/FP1/sapdata62/btab1d_3/btab1d.data3	2103415	29214
2132629	2,048	2.39310789	219
...			

3.5.7 Mapa de discos HPUX-Volume Manager - XP

En el caso de HPUX y arreglos XP el método de obtención del mapa de discos es diferente. La construcción del mapa de discos con HPUX requiere el acceso al comando `vgdisplay -all` :

```
root > vgdisplay -all
VG Name                /dev/vg02
VG Write Access        read/write
VG Status              available, exclusive
Max LV                 255
Cur LV                1
Open LV                1
Max PV                 16
Cur PV                1
Act PV                1
Max PE per PV         8803
VGDA                   2
PE Size (Mbytes)      4
Total PE               8801
Alloc PE              8750
Free PE                51
Total PVG              0
Total Spare PVs       0
Total Spare PVs in use 0

--- Logical volumes ---
LV Name                /dev/vg02/lvol1
LV Status              available/syncd
LV Size (Mbytes)      35000
Current LE             8750
Allocated PE          8750
Used PV                1

--- Physical volumes ---
PV Name                /dev/dsk/c14t0d0
PV Name                /dev/dsk/c27t0d0 Alternate Link
PV Status              available
Total PE               8801
Free PE                51
Autoswitch             On
```

De la copia de esta salida se identifica que el volume group VG02 esta constituido de 1 dispositivo lógico identificado con 2 paths `/dev/dsk/c14t0d0` y `/dev/dsk/c27t0d0`. En este caso es un esquema de alta disponibilidad y el dispositivo es alcanzado por 2 canales de IO.

Sobre este dispositivo esta montado el logical volume `/dev/vg02/lvol1`

Para identificar el que File System esta montado en este logical volumen el comando bdf genera la siguiente salida:

```
$ bdf | grep vg02  
/dev/vg02/lvol1 35840000 35044752 789112 98% /oracle/FP1/sapdata2
```

Esta información es almacenada en las siguientes tablas:

```
SQL> desc device_vg
```

Name	Null?	Type
-----	-----	-----
DEVICE		CHAR(16)
VG		CHAR(8)

```
SQL> desc fs_vg
```

Name	Null?	Type
-----	-----	-----
FS		CHAR(45)
VG		CHAR(8)

3.5.8 Monitor de utilización de discos Top 15

Como se describió en la sección 2.4 el monitoreo de discos es sumamente compleja. Esta complejidad se elimina por completo para el administrador con el monitor de desempeño de disco.

La siguiente pantalla de monitoreo muestra el mapa de discos y las métricas de desempeño resumidas por día y mostrando volúmenes (volume group), dispositivo de acuerdo al procedimiento de [Mapa de discos](#).

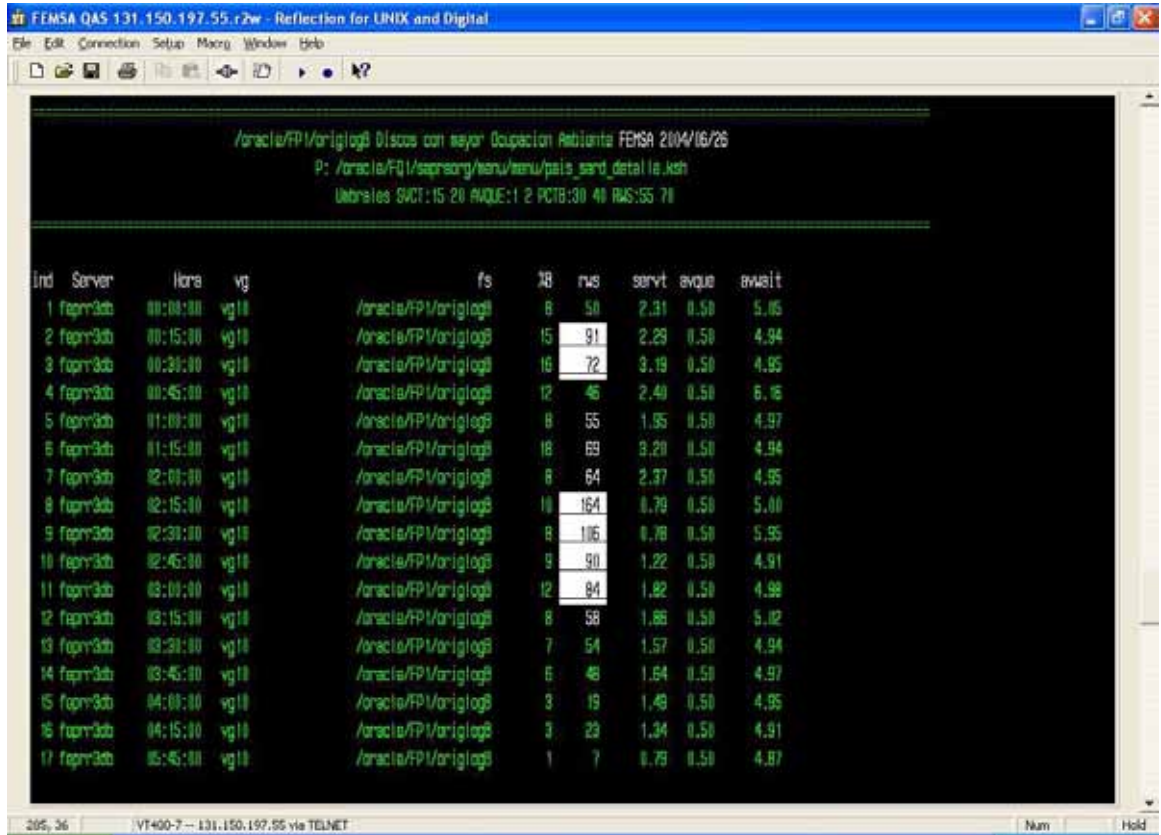
La salida de este monitor muestra los dispositivos más utilizados en promedio por hora de acuerdo al criterio de sar -d.

Disks con mayor Ocupación Ambiente FEMSA 2014/06/26
P: /oracle/FP1/saparch/manu/manu/pais_sar_d_hora.ksh
Umbrales SVCT: 15 20 AVQUE: 1 2 PCIB: 30 40 IAS: 55 70

Ind	Server	Hora	vg	fs	KB	rus	servt	avque	avwait
1	feprn3db	19:00:00	vg10	/oracle/FP1/origlog0	30	1681	2.27	0.50	5.02
2	feprn3db	16:00:00	vg10	/oracle/FP1/origlog0	27	2011	1.97	0.50	4.99
3	feprn3db	16:00:00	vg10	/oracle/FP1/origlog0	27	182	2.10	0.50	5.00
4	feprn3db	16:00:00	vg72	/oracle/FP1/sapdata32	26	791	4.36	0.50	5.00
5	feprn3db	17:00:00	vg10	/oracle/FP1/origlog0	25	141	2.58	0.50	5.03
6	feprn3db	17:00:00	vg23	/oracle/FP1/archtap	24	186	3.77	0.50	5.12
7	feprn3db	15:00:00	vg23	/oracle/FP1/archtap	24	139	2.51	0.50	4.99
8	feprn3db	20:00:00	vg10	/oracle/FP1/origlog0	22	140	2.01	0.50	5.01
9	feprn3db	02:00:00	vg58	/oracle/FP1/sapdata28	21	136	2.53	0.50	5.01
10	feprn3db	17:00:00	vg07	/oracle/FP1/saparch	21	374	0.86	0.50	4.76
11	feprn3db	22:00:00	vg07	/oracle/FP1/saparch	21	287	1.17	0.50	3.86
12	feprn3db	19:00:00	vg07	/oracle/FP1/saparch	21	277	0.96	0.50	4.66
13	feprn3db	19:00:00	vg09	/oracle/FP1/almlog0	21	152	1.62	0.50	5.00
14	feprn3db	16:00:00	vg07	/oracle/FP1/saparch	21	410	0.64	0.50	4.51
15	feprn3db	16:00:00	vg60	/oracle/FP1/sapdata20	21	66	5.59	0.50	4.98

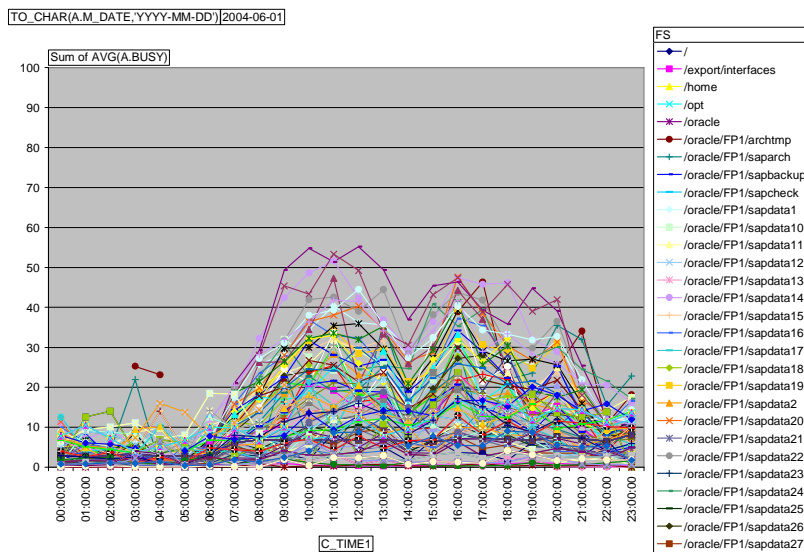
Sig Facha | Ant Facha | Detalle x Disco-Hora | Horas | Diaz | Otro Pais | [return] Regresa

Se puede analizar el comportamiento en detalle por un cierto dispositivo al presionar D y el índice del volumen a revisar.



Una vez que se ha identificado un volumen con un %utilización mayor al 40% de manera continua debe reorganizarse la información que reside en este.

La versión gráfica de se genera con la plantilla de Excel sard busy x hora.xls



3.5.9 Monitor de discos críticos (Hot Disks Top 15)

La práctica recomendada para analizar los dispositivos es analizar la salida del comando `sar -d` e identificar los dispositivos que registren más del 10% de las muestras los siguientes criterios:

- %Utilización > 40 y rws > 70.
- Avg Srvt > 20 y avque > 2

Cualquier dispositivo que se encuentre en este rango debe ser reorganizado para distribuir las lecturas - escrituras de los discos.

Discos con mayor Ocupacion Ambiente FEHSA 2014/06/25
P: /oracle/FQ1/sapreorg/menu/menu país_sar_d_hotdisk.ksh
Umbral:es SVCT:15 20 AVQUE:1 2 PCTB:30 40 RWS:55 70

Ind	Server	fecha	vg	fs	r0	rws	servt	avque	await	veces
1	feprn3db	2014-06-25	vg102	/oracle/FP1/sapdata62	54	268	3.26	0.50	5.01	21
2	feprn3db	2014-06-25	vg102	/oracle/FP1/sapdata84	54	268	3.26	0.50	5.01	21
3	feprn3db	2014-06-25	vg23	/oracle/FP1/archtmp	53	221	3.98	0.50	5.00	19
4	feprn3db	2014-06-25	vg80	/oracle/FP1/sapdata20	45	194	3.01	0.50	5.01	19
5	feprn3db	2014-06-25	vg07	/oracle/FP1/saparch	90	999	3.99	0.51	5.05	18
6	feprn3db	2014-06-25	vg70	/oracle/FP1/sapdata30	45	294	2.39	0.50	5.01	17
7	feprn3db	2014-06-25	vg72	/oracle/FP1/sapdata32	45	257	3.04	0.50	5.01	17
8	feprn3db	2014-06-25	vg84	/oracle/FP1/sapdata24	51	329	2.63	0.50	5.01	13
9	feprn3db	2014-06-25	vg86	/oracle/FP1/sapdata16	43	252	1.96	0.50	5.01	12
10	feprn3db	2014-06-25	vg80	/oracle/FP1/sapdata40	49	287	3.17	0.50	5.00	11

Sig Fecha | Ant Fecha | Detalle x Disco-Hora | Horas | Dias | Otro País | | Regresa

Al igual que el resumen de utilización de disco se puede analizar el detalle de la actividad en los dispositivos al presionar D y el índice del volumen a revisar.

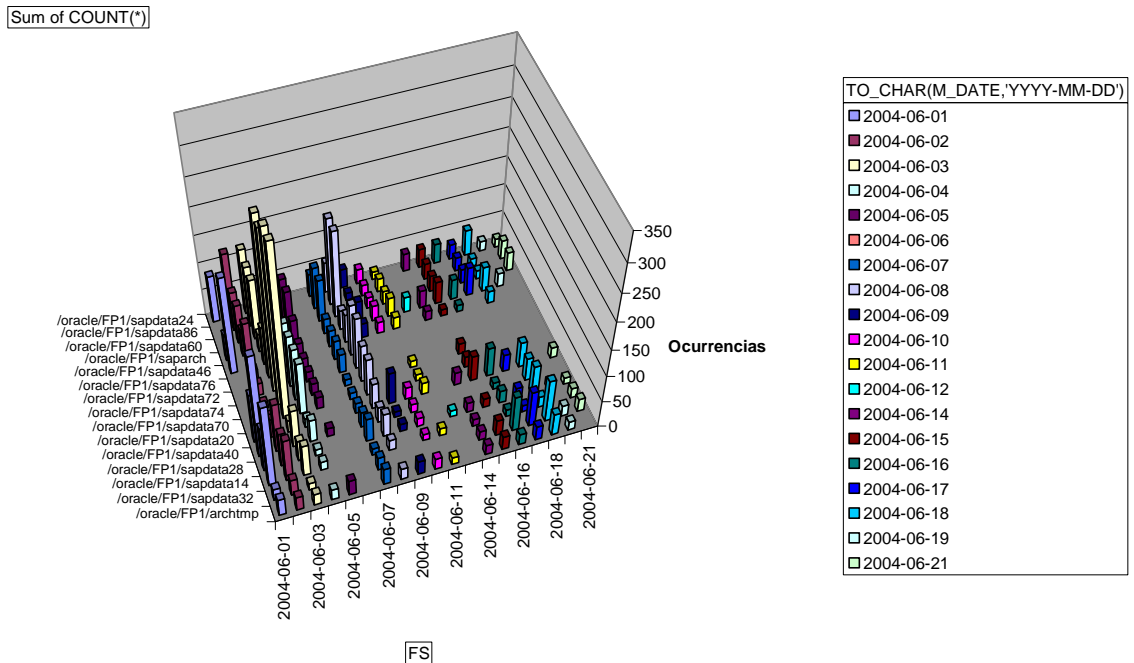
```

/oracle/FP1/sapdata62 Discos con mayor Ocupacion Ambiente FEHSA 2014/06/25
P: /oracle/FP1/sapprog/menu/menu/pais_sand_detalle.ksh
Umbral: SVCT: 15 20 AVQUE: 1 2 PCTB: 30 40 PAS: 55 70
    
```

Ind	Server	Hora	vg	fs	%	r/s	servt	avque	await
1	Feprn3db	00:10:00	vg102	/oracle/FP1/sapdata62	5	23	4.99	0.50	5.17
2	Feprn3db	00:15:00	vg102	/oracle/FP1/sapdata62	7	50	7.61	0.50	5.73
3	Feprn3db	00:30:00	vg102	/oracle/FP1/sapdata62	5	19	6.56	0.50	5.20
4	Feprn3db	00:45:00	vg102	/oracle/FP1/sapdata62	3	14	4.38	0.50	5.13
5	Feprn3db	01:00:00	vg102	/oracle/FP1/sapdata62	16	62	3.68	0.50	5.28
6	Feprn3db	01:15:00	vg102	/oracle/FP1/sapdata62	5	15	3.85	0.50	5.02
7	Feprn3db	01:30:00	vg102	/oracle/FP1/sapdata62	3	11	3.46	0.50	5.12
8	Feprn3db	01:45:00	vg102	/oracle/FP1/sapdata62	3	10	8.31	0.50	5.11
9	Feprn3db	02:00:00	vg102	/oracle/FP1/sapdata62	21	114	2.05	0.50	5.01
10	Feprn3db	02:15:00	vg102	/oracle/FP1/sapdata62	9	50	2.62	0.50	5.02
11	Feprn3db	02:30:00	vg102	/oracle/FP1/sapdata62	13	57	2.38	0.50	5.01
12	Feprn3db	02:45:00	vg102	/oracle/FP1/sapdata62	12	70	2.31	0.50	5.00
13	Feprn3db	03:00:00	vg102	/oracle/FP1/sapdata62	8	34	3.15	0.50	5.00
14	Feprn3db	03:15:00	vg102	/oracle/FP1/sapdata62	5	10	4.34	0.50	5.03

La representación gráfica de estos datos se obtiene con la plantilla hot disks.xls.

Los discos críticos (Top Hot Disk 15)



3.5.10 Monitor de desempeño del servidor de base de datos

El servidor más crítico es donde reside la base de datos, debido que lectura y escritura de datos se realiza de manera centralizada en este servidor. Si existe alguna falla en este equipo esto generará un problema generalizado en el sistema. Un monitoreo específico sobre este servidor permitirá tener una visión completa de su desempeño general.

El monitor de la instancia central muestra en una sola pantalla el consumo de cpu, memoria y disco. Las métricas mostradas son:

- R ready queue
- B procesos bloqueados en espera de recursos de IO
- ID porcentaje de disponibilidad del CPU
- Po/s Paginación por segundo
- Avgpo proporción de la paginación por hora respecto a la memoria física del servidor central.
- Svc_t promedio del tiempo de servicio de todos los discos del sistema.
- Pct_b porcentaje de uso de todos los discos de la instancia central.
- Krps promedio de Kbytes leídos por segundo de todos los discos del sistema
- rps promedio de lecturas por segundo de todos los discos del sistema

Se implemento 2 variantes del mismo programa:

Un ciclo infinito de cada 80 segundos antes de refrescar la columna.

Otro que permite la consulta inmediata de manera iterativa.

```
Todas las metricas mexico 2002/10/08
P: /oracle/MXD/sapreorg/tesis/menu/central_all_refresh.ksh
Umbralas R:2 4 B:10 15 ID:40 20 MEM:50 90 SVCT:20 35 PCTB:20 30
```

hora	r	b	id	po	avgpo	svc_t	pctb	krps	rps
00:00:00	0	2	46	1	0	33	30	544	11
00:30:00	0	0	68	10	0	19	11	233	5
01:00:00	0	0	82	1	0	10	1	31	1
01:30:00	0	0	98	0	0	10	1	10	0
02:00:00	0	0	81	3	0	6	4	306	9
02:30:00	0	0	90	0	0	8	0	3	0
03:00:00	0	0	98	1	0	5	0	4	0
03:30:00	0	0	97	2	0	3	0	0	0
05:30:00	0	0	98	0	0	7	0	2	0
06:00:00	0	0	92	1	0	10	4	146	6
06:30:00	0	1	91	14	1	10	0	3	0
07:00:00	0	0	83	14	1	10	6	186	7
07:30:00	0	1	78	2207	95	11	9	215	9
08:00:00	1	0	71	35	1	10	4	93	4
08:30:00	0	6	64	31	1	11	12	254	11
09:00:00	1	11	51	8	0	19	28	410	20
09:30:00	0	20	51	16	1	27	35	380	20

Nótese que el intervalo con el cuál se muestra la información de 30 minutos.

El único archivo de parametrización es [alarmas wa.lst](#). Los scripts correspondientes son [central_all_refresh.ksh](#) y [central_all.ksh](#) se encuentran en el anexo correspondiente.

Este monitor es especialmente para monitorear el desempeño del servidor de base de datos donde los problemas de desempeño son críticos.

3.5.11 Resumen de los 10 eventos Top de Oracle

Este monitor analiza la distribución de los eventos de espera de Oracle. El monitor esta configurado para alarmar y advertir de cualquier anomalía dentro de estos eventos de acuerdo a los explicado en la sección 3.4.1 System Event.

Con el monitor es posible detectar el origen de una desviación en los tiempos de respuesta de la base de datos.

```
Los 10 eventos de espera mas significativos de Oracle FEWSA 2004/06/26
P: /oracle/FQ1/sapreorg/manu/manu/pais_eventos_oracle_top10.ksh
Umbral de db file sequential read :10 15 buffer busy waits :10 15 log file sync :15 20
Especial atencion en los eventos db file scattered read, undo block, write completed waits, free buffer waits
```

Ind	Event	Date	Start	End	Time (cs)	Wait (ms)	%
1	db file sequential read	2004-06-26	00:29:00	23:29:00	48354177	2.99	64.04
2	log file sync	2004-06-26	00:29:00	23:29:00	10745617	35.71	14.23
3	buffer busy waits	2004-06-26	00:29:00	23:29:00	61023457	4.58	7.99
4	latch free	2004-06-26	00:29:00	23:29:00	4628254	12.15	6.13
5	log file parallel write	2004-06-26	00:29:00	23:29:00	2272253	9.63	3.01
6	enqueue	2004-06-26	00:29:00	23:29:00	1294576	109.01	1.64
7	db file scattered read	2004-06-26	00:29:00	23:29:00	1139798	5.58	1.51
8	log buffer space	2004-06-26	00:29:00	23:29:00	871438	45.53	1.15
9	SQL*Net more data to client	2004-06-26	00:29:00	23:29:00	167100	0.41	0.22
10	log file sequential read	2004-06-26	00:29:00	23:29:00	55357	0.42	0.08

Sig Fecha | Ant Fecha | Detalle x Disco-Hora | Horas | Dias | Otro Pais | Regresa

Continuando con la misma filosofía de los monitores se puede analizar el detalle del evento durante un día al presionar D y el índice del evento.

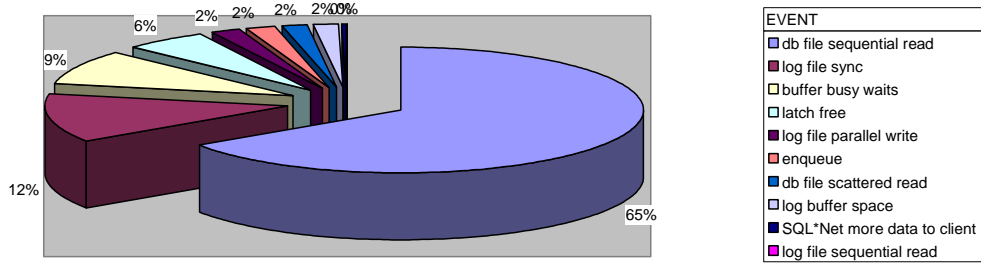
```
Detalle del Evento Oracle log file sync FEWSA 2004/06/26
P: /oracle/FQ1/sapreorg/manu/manu/pais_eventos_oracle_detalle.ksh
Umbral de <N>
```

Event	Date	Start	End	Time (cs)	Wait (ms)
log file sync	26-JUN-04	01:29:00	01:29:00	187925	29.42
log file sync	26-JUN-04	01:29:00	02:29:00	81883	7.77
log file sync	26-JUN-04	02:29:00	03:29:00	147983	9.97
log file sync	26-JUN-04	03:29:00	04:29:00	52180	9.14
log file sync	26-JUN-04	04:29:00	05:29:00	9024	3.51
log file sync	26-JUN-04	05:29:00	06:29:00	71974	14.15
log file sync	26-JUN-04	06:29:00	07:29:00	178191	17.72
log file sync	26-JUN-04	07:29:00	08:29:00	165979	16.29
log file sync	26-JUN-04	08:29:00	09:29:00	25352	7.97
log file sync	26-JUN-04	09:29:00	10:29:00	35566	6.92
log file sync	26-JUN-04	10:29:00	11:29:00	35487	6.42
log file sync	26-JUN-04	11:29:00	12:29:00	42712	7.20

La representación gráfica de la proporción de los eventos de estos valores se genera con la plantilla system events x dia.xls.

Date|2004-06-24

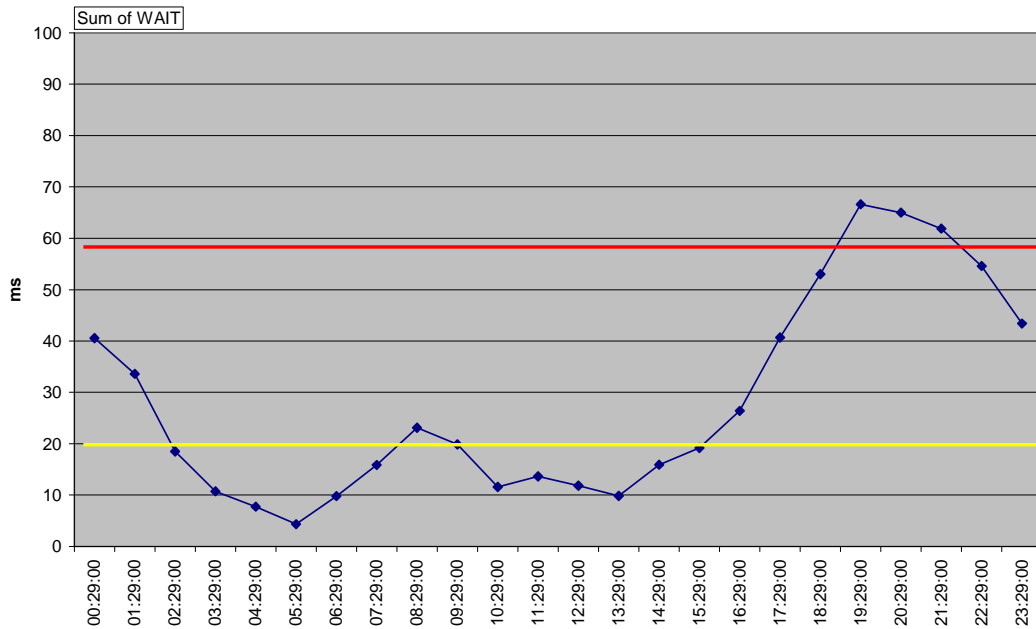
Top 10 System Events



El detalle del tiempo promedio por evento se genera con la plantilla system events x dia resumen.xls.

DATE|2004-06-23 Wednesday|EVENT|log file sync

Log File Sync Avg Wait



3.5.12 Los datafiles más accesados (datafiles_oracle_top30.ksh)

Este monitor muestra el uso de los datafiles más accesados calculando el % del total de operaciones realizadas (physical reads + 2 * physical writes) y el tiempo promedio de espera por lectura (read time / physical reads * 10).

Las recomendaciones sobre el desempeño explicadas en la sección 2.3.6 son aplicadas y este monitor y se muestran como una alarma.

Si en un datafile se concentra demasiado IO (%tot > 5%) se marca como un hot datafile y es necesario identificar los objetos están contenidos en este para reubicarlos en otros datafiles. Este proceso es conocido como reorganización.

En caso que el tiempo de espera sea > 15 ms se debe analizar los discos donde están ubicados usando el monitor de rendimiento de discos.

```

Datafiles con mayor operacion de IO FEMSA
P: /oracle/FQ1/saporg/manu/manu/pais/datafiles_oracle_top30.ksh
Librales ReadTime(s) : 18.15 Porc Tot: 5.10

```

Ind	Tabloespaco	DataFile	ReadTime	PhysRds	RdTim(ms)	PhysWrts	Tot	XTot
1	PSAPSTRBO	/oracle/FQ1/sapdata24/otabd_25/otabd_data25	713117	3169192	2.30	1194	3161400	2.10
2	PSAPSTRBI	/oracle/FQ1/sapdata11/otabi_11/otabi_data11	602977	2612773	3.19	39250	2612779	1.86
3	PSAPSTRBO	/oracle/FQ1/sapdata60/otabd_4/otabd_data4	553279	2126618	2.53	40579	2216976	1.59
4	PSAPSTRBO	/oracle/FQ1/sapdata14/otabd_14/otabd_data14	791374	2116560	3.74	10450	2137460	1.53
5	PSAPSTRBO	/oracle/FQ1/sapdata61/otabd_1/otabd_data1	640844	2020150	3.19	44169	2108256	1.51
6	PSAPSTRBO	/oracle/FQ1/sapdata40/otabd_41/otabd_data41	536660	2064077	2.58	5392	2065461	1.50
7	PSAPSTRBO	/oracle/FQ1/sapdata38/otabd_38/otabd_data38	543516	2039180	2.57	4266	2047652	1.47
8	PSAPSTRBO	/oracle/FQ1/sapdata8/otabd_9/otabd_data8	754682	1767178	4.16	124429	2015836	1.46
9	PSAPSTRBO	/oracle/FQ1/sapdata36/otabd_37/otabd_data37	433686	1887896	2.30	34109	1955914	1.40
10	PSAPSTRBO	/oracle/FQ1/sapdata38/otabd_39/otabd_data39	607116	1919147	3.16	4943	1928933	1.38
11	PSAPSTRBO	/oracle/FQ1/sapdata4/otabd_90/otabd_data90	672373	1918127	3.52	988	1919113	1.37
12	PSAPSTRBI	/oracle/FQ1/sapdata19/otabi_19/otabi_data19	510375	1779715	2.87	48174	1875853	1.34
13	PSAPSTRBO	/oracle/FQ1/sapdata26/otabd_25/otabd_data26	401836	1642761	2.18	9773	1652317	1.34
14	PSAPSTRBO	/oracle/FQ1/sapdata30/otabd_30/otabd_data30	430189	1786335	2.44	3917	1793945	1.29
15	PSAPSTRBO	/oracle/FQ1/sapdata64/otabd_2/otabd_data2	397045	1719182	2.32	11582	1739213	1.24

3.5.13 Los sapdatas más accesados (sapdatas_oracle_top30.ksh)

Este monitor muestra la misma información que el monitor anterior pero agrupándolos por VG-Sapdata.

En especial esto es útil para planear reorganizaciones

```

=====
Discos con mayor ocupacion recibiendo FEMSA 2004/06/28
P: /oracle/APP1/saptranz/menu/menu/pais_sapdatas_oracle_top30.ksh
librerias SWC1: 15 20 ANQUE: 1 ? PC18: 30 40 RAG: 15 70
=====

```

Ind	Time1	Time2	vg	Sapdata	ReadTime	PhyRds	RdSvt [ms]	PhyWrites	tot	pct
1	09:45:00	15:45:00	vg51	/oracle/APP1/sapdata24	119213	4655166	2.51	5318	4655682	3.34
2	09:45:00	15:45:00	vg112	/oracle/APP1/sapdata62	135298	4281573	3.15	146835	4572243	3.28
3	09:45:00	15:45:00	vg76	/oracle/APP1/sapdata38	1173478	3683517	2.95	10123	4013753	2.87
4	09:45:00	15:45:00	vg811	/oracle/APP1/sapdata41	1116336	3653955	2.86	36312	3342578	2.83
5	09:45:00	15:45:00	vg54	/oracle/APP1/sapdata14	1371571	3581455	3.83	71643	3724741	2.57
6	09:45:00	15:45:00	vg114	/oracle/APP1/sapdata54	1116179	3631291	3.15	12295	3654881	2.52
7	09:45:00	15:45:00	vg83	/oracle/APP1/sapdata43	142562	239571	5.95	1573751	3399172	2.44
8	09:45:00	15:45:00	vg71	/oracle/APP1/sapdata31	881612	3331119	2.55	19618	3371235	2.42
9	09:45:00	15:45:00	vg59	/oracle/APP1/sapdata19	977315	3119179	3.13	118612	3366283	2.41
10	09:45:00	15:45:00	vg82	/oracle/APP1/sapdata42	731156	3212313	2.28	71661	3346155	2.40
11	09:45:00	15:45:00	vg1	/oracle/APP1/sapdata11	1123911	3115312	3.61	61211	3235772	2.32
12	09:45:00	15:45:00	vg85	/oracle/APP1/sapdata45	216811	295184	7.11	146167	3217318	2.31
13	09:45:00	15:45:00	vg74	/oracle/APP1/sapdata34	654214	3132711	2.76	21435	3175371	2.28
14	09:45:00	15:45:00	vg58	/oracle/APP1/sapdata28	716349	3141763	2.33	19117	3161197	2.21
15	09:45:00	15:45:00	vg56	/oracle/APP1/sapdata26	87741	2981798	2.54	25341	3132481	2.17
16	09:45:00	15:45:00	vg72	/oracle/APP1/sapdata32	1111481	2971951	3.71	16464	3111488	2.15
17	09:45:00	15:45:00	vg111	/oracle/APP1/sapdata61	963513	2891251	3.33	46542	2981234	2.14

4 Análisis de desempeño de SAP R3

El SMDSA permite conocer si el desempeño de los servidores de aplicación es adecuado o no medido en la utilización y rendimiento de los dispositivos. Una vez identificado un problema con el desempeño de los servidores de aplicación es necesario analizar en detalle las posibles causas de esta problemática, con esto se estará sustentando una recomendación para mejorar el desempeño del sistema, es decir que afinación requiere.

A continuación se presentan los conceptos de desempeño de SAP R3 y las transacciones estándares de análisis, con ellas de podrá determinar el desempeño del sistema medido en tiempo de respuesta y rendimiento.

4.1 Análisis de tiempo de respuesta y carga de trabajo (workload analysis).

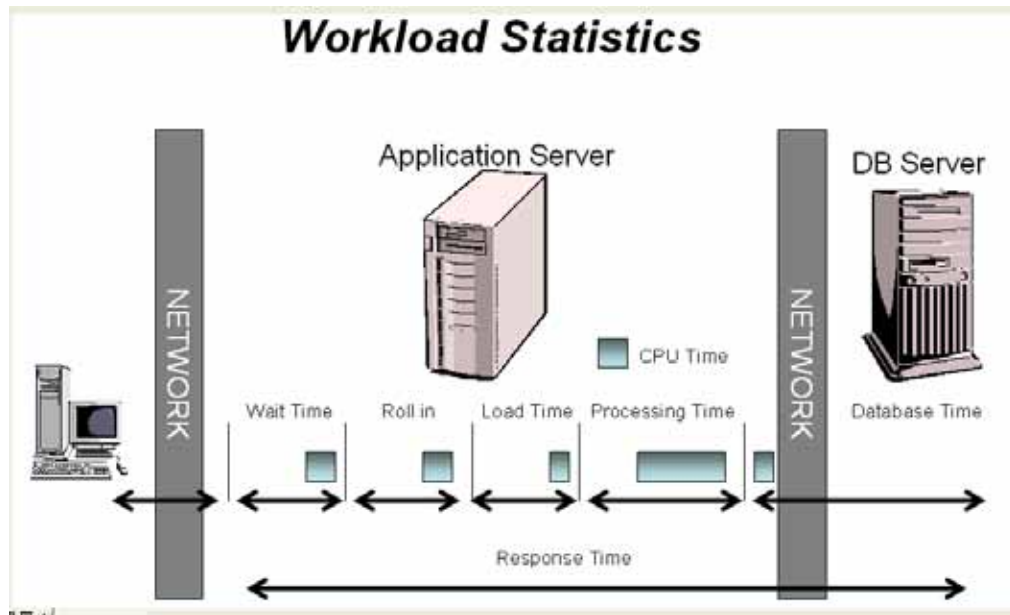
El tiempo de respuesta y la distribución de la carga de trabajo son los conceptos esenciales del análisis de rendimiento de SAP R3, a continuación se presentan los conceptos básicos.

4.1.1 Tiempo de respuesta

El tiempo de respuesta comprende las siguientes etapas.

- **Tiempo de Respuesta (Response Time).** Inicia cuando una solicitud de usuario ingresa a la cola de espera del despachador y termina cuando la siguiente pantalla es regresada al usuario. Este tiempo de respuesta no incluye el tiempo requerido en transferir la pantalla al servidor de presentación.
- **Tiempo de Espera (Wait Time).** Tiempo que la petición se encuentra en la cola del despachador en espera de ser asignado a un proceso de trabajo.
- **Tiempo de Roll-In (Roll-In Time).** Tiempo requerido para copiar o mapear el contexto del usuario en el proceso de trabajo.

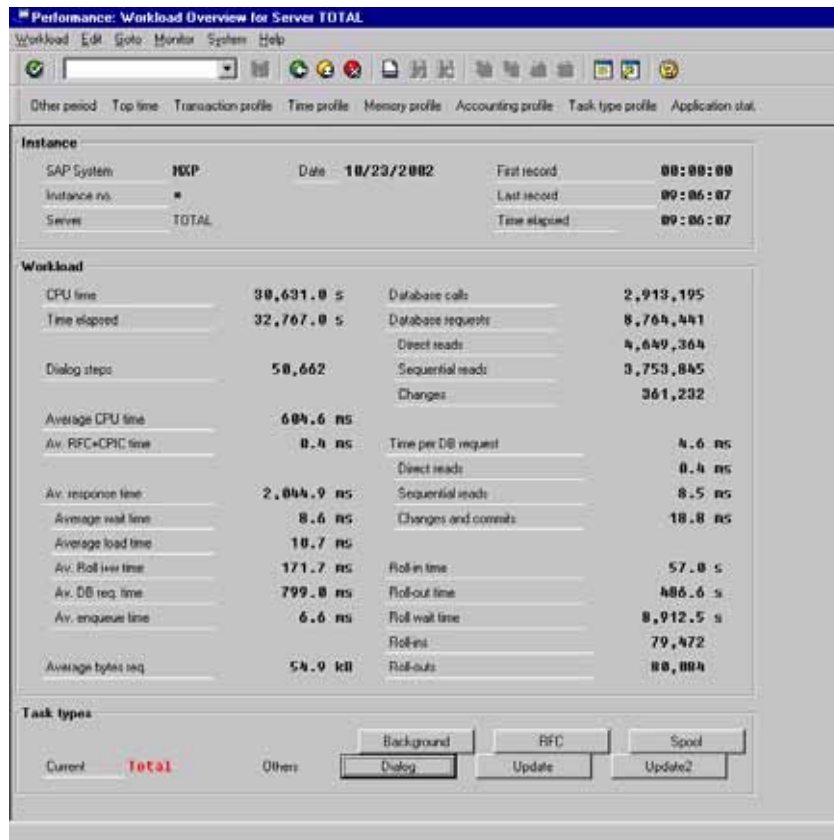
- **Tiempo de Carga (Load Time).** Tiempo requerido en cargar de la base de datos y generar objetos tales como código ABAP, CUA e información de las pantallas.
- **Tiempo de base de datos (Database Request Time).** Inicia cuando la solicitud es enviada al RDBMS vía la interfaz de la base de datos y termina cuando esta entrega el resultado.
- **Tiempo de CPU.** Tiempo en que el proceso de trabajo utiliza el CPU.
- **Tiempo de Encolamiento.** Tiempo en el cuál es ubicado una petición de trabajo en la cola del despachador.
- **Tiempo de Procesamiento.** Tiempo total de Respuesta menos la suma de: Tiempo de Espera + Tiempo de Roll + Tiempo de Carga + Tiempo de Base de Datos



4.1.2 ST03 transacción de análisis de tiempo de respuesta y carga de trabajo (Workload Analysis)

El tiempo de respuesta promedio por paso de diálogo permite conocer el tiempo promedio en realizar un cambio de pantalla. Es decir, mide el tiempo en que cada petición que llega al servidor comienza a atenderse hasta que se regresan los datos al servidor de aplicación. (Como de describió en [Paso de diálogo \(Dialog Step\)](#))

La transacción ST03 de SAP permite visualizar los componentes del tiempo de respuesta. La forma básica de consulta es ST03->Choose for Análisis->TOTAL->Today.



La Transacción ST03 ofrece estadísticas históricas y vistas sobre transacciones, usuarios y horarios de carga de trabajo. Estas vistas serán analizadas en el capítulo de la metodología de desempeño.

4.1.3 Tiempo de respuesta por transacciones

La vista de carga de trabajo tiempo de respuesta por transacción permite identificar los programas donde se centra el consumo de recursos del sistema.

Para consultar esta vista ST03 -> Choose for Analysis -> This month -> Transaction Profile, una vez generado el resultado posicionarse en un valor de la columna Response Time Total (s) y ordenarlo descendientemente (Shift+F4).

Instance: SAP System HXP, Server TOTAL, Instance no. *
 First record 10.01.2002 00:00:00, Last record 10.23.2002 03:05:27, Elapsed time 22 days, 03:05:26
 Period Month Start 10 / 2002, Task type Total

Syst.no. 01/02/03/04/05 Entries: 2926 Sorted by Response time

Program or Tcode	Background job	Dialog steps	Response time		CPU time		Wait total (s)
			total (s)	avg (ms)	total (s)	avg (ms)	
=TOTAL=		5,892,800	10,570,920	1,794	2,540,290	431	11,499
MBST		74,946	1,559,883	20,803	6,391	85	13
Z10017		45,517	1,037,902	22,803	229,762	5,048	61
SESSION_MANN		252,269	656,400	2,602	71,050	282	1,553
NI00		552,258	590,185	1,069	240,974	442	1,063
RFC		426,802	338,725	794	175,159	410	1,444
FS10H		96,708	254,899	2,636	141,092	1,459	57
IM01		142,037	233,297	1,643	53,534	377	299
MDSB		62,440	229,459	3,675	7,704	123	4
NE51H		88,692	226,621	2,555	55,303	624	254
SAPMSYST		169,902	225,341	1,326	92,712	546	124
ZDIS_02_LADY	ZDIS_02_LADY_HU0	12	207,266	17272129	1,613	134,384	0
ZH051		8,240	191,200	23,205	3,414	414	3
NE21H		203,423	180,892	889	66,353	326	233
RSDRAJOB	DBA:ALLONLINE_0230	18	156,489	8693,830	9,315	517,524	0
F0L0N		28,604	153,802	5,377	46,900	1,640	15
MDS1		11,810	151,529	12,831	4,841	410	5
NI00		70,573	151,429	2,146	33,107	469	113
NE2V		81,075	144,820	1,786	62,640	773	6
NE53H		52,900	117,425	2,214	30,243	572	82
NE5R		200,582	110,920	553	19,146	95	18
IM32		128,296	103,821	809	39,262	306	261
AutoABAP		37,563	103,489	2,755	91,244	2,420	59
NE23H		70,622	100,656	1,425	29,255	414	174

De estas transacciones se debe investigar:

- Revisar notas OSS con el código de la transacción y la palabra performance si es una transacción estándar de SAP.
- Solicitar al desarrollador ABAP revisar el código en caso de ser un desarrollo propio.
- Si están dentro de las instrucciones SQL más costosas. Revisar es susceptible de generar índice o generar estadísticas para el optimizador de costos.
- Determinar si las tablas accesadas están fragmentadas, si cuentan con estadísticas adecuadas o si los índices cuentan con una calidad suficiente.
- Asegurar que los usuarios cuenten con una adecuada capacitación sobre los criterios para ejecutar un reporte, la hora para ejecución del proceso y si es posible programarlo como un proceso de fondo (batch).

Esta lista es especialmente útil para generar análisis de mayor profundidad sobre la afinación de arquitectura o de la aplicación.

Nótese que se analiza tomando en consideración el tiempo de respuesta total y no los pasos de diálogo.

4.1.4 Tiempo de respuesta por usuario

La vista de consumo de recursos por usuario permite identificar cuáles usuarios generan la mayor cantidad de carga en el sistema.

ST03 -> Choose for análisis -> TOTAL -> This month -> Goto -> Profiles -> User Profile. Una vez generado el resultado posicionarse en un valor de la columna Response Time Total (s) y ordenarlo descendientemente (Shift+F4).

User	Dialog steps	Response time		CPU time		Wait time		DB time total (s)
		total (s)	avg (ms)	total (s)	avg (ms)	total (s)	avg (ms)	
TOTAL	5,892,800	10,573,920	1,794	2,540,290	431	11,499	2	5241,491
ADELAROSA	30,248	559,814	18,507	286,029	9,456	126	4	50,201
JPALDIAR	9,750	221,836	22,752	4,747	487	21	2	104,925
SAPSYS	640,296	126,877	198	96,822	151	1,728	3	17,647
HURBINAR	6,356	115,944	18,242	4,700	739	27	4	71,669
YHEMSEH	3,920	109,697	27,984	1,272	324	2	0	58,476
NIARTINEZ	56,381	101,795	1,805	42,384	752	98	2	47,793
BCARDILA	5,022	100,744	20,460	1,274	254	6	1	52,341
FUMCBATC	84	98,434	1171,831	55,718	663,304	0	0	43,349
MHERNAN	12,000	89,791	7,403	26,633	2,205	6	0	55,050
MCARDENC	3,796	87,937	23,166	3,049	803	2	1	50,112
JAAGUILAH	8,957	81,639	9,115	15,602	1,742	7	1	65,175
OFARIASD	1,569	81,398	51,879	1,230	704	2	1	47,597
LTOARESC	17,410	76,237	4,379	19,264	1,100	304	17	47,187
CREYNOSU	1,410	66,987	47,500	916	650	1	1	31,796
UFERNAND	14,381	66,839	4,648	15,699	1,092	7	1	46,349
GCUERRAL	3,654	64,938	17,772	4,353	1,191	2	1	24,615
DLARRU	3,622	63,505	17,533	10,526	2,906	4	1	44,077
SNAVARRO	1,620	62,941	30,852	28,448	17,561	18	11	27,790
BNBRAHJC	8,489	60,594	7,138	6,343	747	9	1	43,626
NSANCHE	6,143	59,619	9,705	5,778	941	3	1	39,437
MBENAVIJ	235	54,282	230,986	7,547	32,115	0	1	41,048
NPORTILH	4,143	53,563	12,928	21,129	5,100	3	1	25,779
ACUATEPC	12,502	52,500	4,206	6,028	482	47	4	36,423

La lista de usuarios debe analizarse sobre el tipo de trabajo que realiza los usuarios:

- El perfil del usuario: su rol dentro de la compañía, necesidades de información.
- Capacitación del usuario: uso adecuado de transacciones de R/3, capacitación en procesos de fondo.

4.1.5 Análisis de procesos en tiempo real

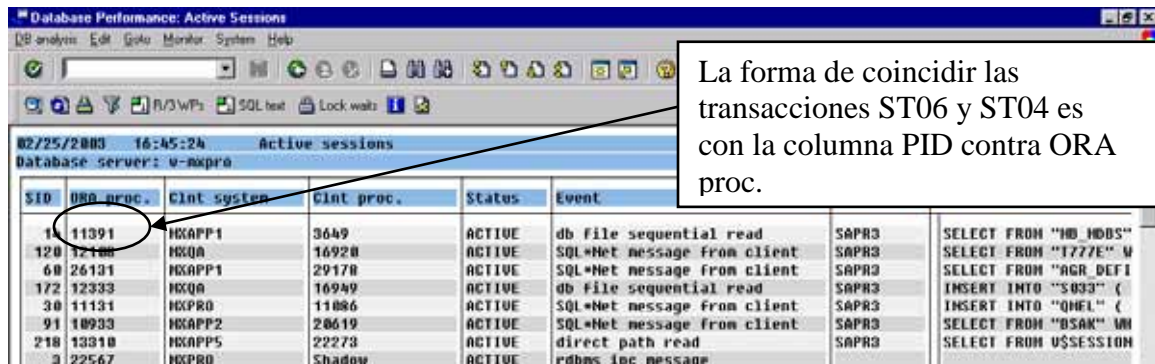
Cuando un monitor de CPU detecta un problema en algún servidor de aplicación es necesario revisar los procesos que se encuentran ejecutándose en ese momento en el servidor. Los pasos a seguir para diagnosticar en tiempo real son:

1. Identificar los procesos de CPU de mayor consumo con la transacción ST06->Detail Analisis Menu->Top Process. Esta pantalla puede mostrar procesos externos, procesos de base de datos y procesos de trabajo SAP.

Pid	Username	Command	CPU Util [%]	CPU Time [s]	Resident size [KB]	PID
11,395	oransp	oracleHWP (DESCRIPT	43.67	125:31	2549,200	59
11,385	oransp	oracleHWP (DESCRIPT	39.52	103:18	2546,456	59
11,622	oransp	oracleHWP (DESCRIPT	38.43	102:55	2550,592	59
26,171	oransp	oracleHWP (DESCRIPT	34.38	065:06	2547,688	60
12,092	oransp	oracleHWP (DESCRIPT	31.42	070:06	2552,696	0
11,391	oransp	oracleHWP (DESCRIPT	29.64	117:28	2545,976	60
11,381	oransp	oracleHWP (DESCRIPT	19.66	109:26	2545,528	59
11,656	oransp	oracleHWP (DESCRIPT	19.16	092:58	2550,472	59
7,922	oransp	oracleHWP (DESCRIPT	10.37	016:27	2551,680	59

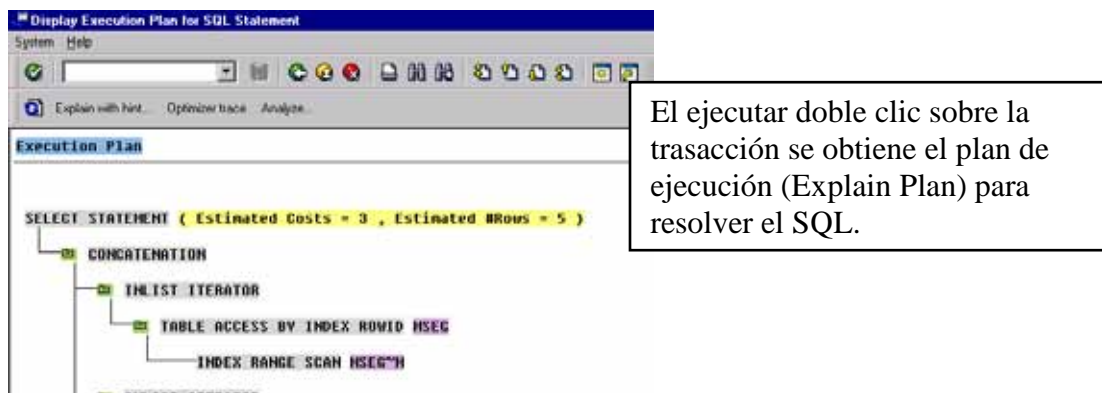
Se presentan los procesos con mayor consumo y su ID de Proceso.

- En caso de ser procesos de base de datos es necesario analizar el costo de la sentencia SQL. ST04->Detail Analisis Menu-> Oracle Process.



La forma de coincidir las transacciones ST06 y ST04 es con la columna PID contra ORA proc.

SID	ORA proc.	Client system	Client proc.	Status	Event	ORA proc.	SQL text
11	11391	MSAPP1	3649	ACTIVE	db file sequential read	SAPR3	SELECT FROM "MS_HDBS"
120	12100	MSQA	16920	ACTIVE	SQL*Net message from client	SAPR3	SELECT FROM "T77ZE" W
60	26131	MSAPP1	29170	ACTIVE	SQL*Net message from client	SAPR3	SELECT FROM "AGR_DEFI
172	12333	MSQA	16949	ACTIVE	db file sequential read	SAPR3	INSERT INTO "S033" (
30	11131	MSPRO	11086	ACTIVE	SQL*Net message from client	SAPR3	INSERT INTO "QHFL" (
91	10933	MSAPP2	20619	ACTIVE	SQL*Net message from client	SAPR3	SELECT FROM "OSAK" WH
218	13310	MSAPP5	22273	ACTIVE	direct path read	SAPR3	SELECT FROM U\$SESSION
3	22567	MSPRO	Shadow	ACTIVE	rdbms ipc message		

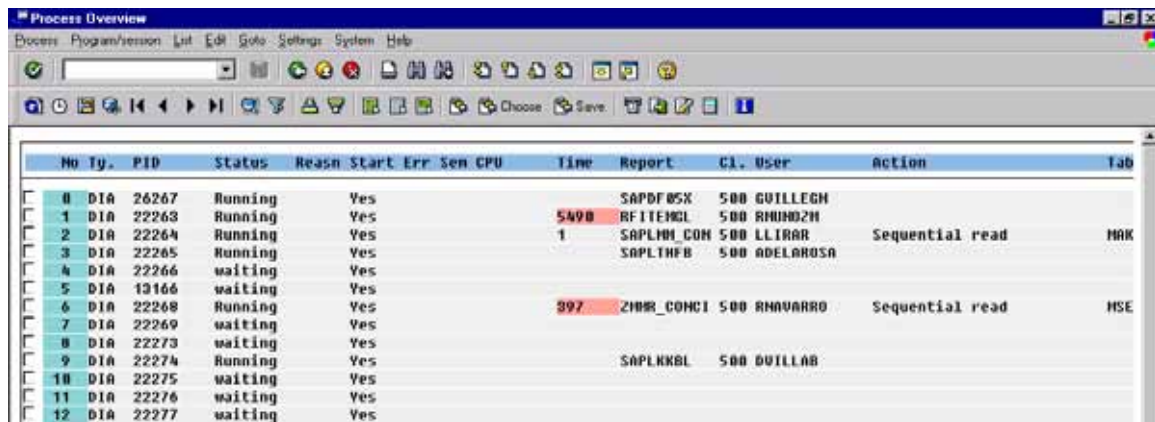


El ejecutar doble clic sobre la trasacción se obtiene el plan de ejecución (Explain Plan) para resolver el SQL.

```

SELECT STATEMENT ( Estimated Costs = 3 , Estimated #Rows = 5 )
├── CONCATENATION
│   └── INLIST ITERATOR
│       └── TABLE ACCESS BY INDEX ROWID MSEG
│           └── INDEX RANGE SCAN MSEG"1"
    
```

- Identificar los procesos de trabajo de SAP que están causando problemas (SM50)



No	Ty.	PID	Status	Reason	Start	Err	Sen	CPU	Time	Report	CL	User	Action	Tab
0	DIA	26267	Running	Yes						SAPDF05X	500	GUILLEGH		
1	DIA	22263	Running	Yes					5490	RFITEMCL	500	AMUNDZM		
2	DIA	22264	Running	Yes					1	SAPLHM_CON	500	LLIRAR	Sequential read	MAK
3	DIA	22265	Running	Yes						SAPLTHFB	500	ADELAROSA		
4	DIA	22266	waiting	Yes										
5	DIA	13166	waiting	Yes										
6	DIA	22268	Running	Yes					397	ZMRM_CONCI	500	AMUARRD	Sequential read	MSE
7	DIA	22269	waiting	Yes										
8	DIA	22273	waiting	Yes										
9	DIA	22274	Running	Yes						SAPLKNBL	500	DUILLAB		
10	DIA	22275	waiting	Yes										
11	DIA	22276	waiting	Yes										
12	DIA	22277	waiting	Yes										

4.2 Desempeño de la memoria de R/3

4.2.1 Memoria compartida y local de R/3

El desempeño de la memoria compartida de R/3 es analizada con la transacción ST02, con esta transacción es posible identificar todos los componentes de la memoria de R/3 registros históricos y parámetros a configurar. Esta consulta debe realizarse por cada instancia de R/3 configurada.

Buffer	Hitratio [%]	Allocated [kB]	Free space [kB]	Free space [%]	Dir. size Entries	Free directory Entries	Free directory [%]	Suaps	Da ac
Nanetab (NTAB)									
Table definition	99.00	5,046	3,098	75.52	30,000	22,658	75.53	0	
Field description	99.51	41,348	13,622	34.93	60,001	52,912	88.19	0	
Short NTAB	99.91	5,348	2,567	85.57	60,001	57,563	95.94	0	
Initial records	99.90	8,348	4,728	78.80	60,001	55,839	93.06	0	
Program									
COB	99.79	411,271	12,617	3.15	100,000	92,425	92.43	10,169	
Screen	99.72	8,000	1,713	24.59	4,000	2,785	69.63	0	
Calendar	99.83	25,391	4,217	17.00	10,000	8,602	86.02	0	
Tables									
Generic key	99.80	82,520	5,421	6.73	13,000	960	7.38	68	
Single record	98.76	30,000	24,968	83.39	500	331	66.20	0	
Export/Import									
	94.97	30,000	16,560	66.66	20,000	15,561	77.81	0	
SAP memory									
		Current use [%]	Current use [kB]	Max. use [kB]	In memory [kB]	On disk [kB]	SAP cursor cache	Hitratio [%]	
Roll area	0.79	23,040	26,120	131,072	131,072		IDs	98.58	
Paging area	7.66	19,600	25,872	65,536	190,464		Statements	99.00	
Extended Memory	33.53	1,404,928	1,683,456	4,190,208					
Heap Memory		0	64,680						
Call statistics									
	Hitratio	ABAP/A	Processor	Database					

La métricas deseables son:

- La memoria definida en la instancias R/3 no deben sobrepasar la capacidad de la memoria física y deben ajustarse a los límites del sistema operativo.¹
- Buffers de R/3: no tengan registros de SWAP < 1000 por día, en caso contrario requieren ampliar la capacidad de los buffers. (exceptuando program buffer). El hit ratio debe ser > 98%

¹ Las restricciones de memoria para ambientes de 32 bits se encuentran documentados en la nota OSS 97497 Memory Management Parameter (3.0/3.1). Aplica aun en versiones 4.X de 32 bits.

- Extended Memory: El Max Use [kb] nunca debe ser el 80% del definido en In Memory. En caso contrario debe considerarse aumentar el valor de la memoria extendida (em/initial_size_MB).² El valor máximo que puede mapear el kernel de R/3 puede ser consultado en sección siguiente.
- Heap Memory: Si existen un valor > 0 en Max. Use. y el Max. Use. de Extended Memory es igual a In Memory los procesos de trabajo están en modo privado debido que no hay suficiente Memoria Extendida para alojar el contexto de usuario.

Los Buffers de R/3 no deben mostrar más de 1000 swaps por día, en caso de que esto suceda es necesario aumentar los parámetros referentes al buffer específico.

Entre las posibles causas por la cuál los buffers swapeen es debido a

- Tablas que se están activando en modo buffer y son demasiado grandes o están siendo modificadas constantemente.
- Pasó de transportes generan recopilación del código fuente ABAP.

² OSS Note: 146289 TCC Recommendations for 64 bit R/3 Kernel y 103747 Performance in 4.0/4.5/4.6: Parameter Recommendation

4.2.2 Historia del comportamiento de la memoria y modificación de parámetros

Para determinar históricamente cuántos swaps han ocurrido la transacción ST02 permite consultar el estado de la memoria en días pasados. ST02->History y posteriormente revisar el las diferentes áreas de memoria de R/3. Con esto se puede identificar si el problema es un pico o continuamente se presenta el problema.

System: vzapp1_UZP_01										
Program buffer (PKA)										
Date	Hitratio [%]	Quality [%]	Buffer size [kB]			Database accesses	Directory entries		Swaps	#
			allocated	available	free		maximum	Free		
Mon 09/23/2002	99,54	99,41	405,354	400,000	1,276	8,077	50,000	47,177	8	1
Sun 09/22/2002	99,82	99,73	405,354	400,000	15,660	39,073	50,000	47,039	9,505	2
Sat 09/21/2002	99,82	99,72	405,354	400,000	15,900	37,959	50,000	47,042	8,470	2
Fri 09/20/2002	99,80	99,70	405,354	400,000	16,520	36,981	50,000	47,018	8,161	2
Thu 09/19/2002	99,79	99,69	405,354	400,000	15,124	30,918	50,000	46,979	6,263	2
Wed 09/18/2002	99,76	99,65	405,354	400,000	13,740	25,605	50,000	46,982	4,714	2
Tue 09/17/2002	99,73	99,60	405,354	400,000	12,688	18,882	50,000	46,868	2,723	2
Average	99,75	99,65	405,354	400,000	12,987	28,185	50,000	47,015	5,549	
Maximum usage since 08/30/2002										
Mon 09/23/2002	Buffer		400,000	1,276	2,823 directory entries used					
Tue 09/17/2002	Directory		390,172 kB buffer usage			50,000	46,864			

4.2.3 Usuarios y procesos con mayor consumo de memoria

Los usuarios que consumieron mayor cantidad de memoria en este servidor ST02 -> Detail Analysis Menu -> Others Buffers SAP Memory -> Mode List.

The screenshot shows the SAP TUNE: Detail Analysis window for system mxapp5_MXP_05. The window title is 'TUNE: Detail Analysis (mxapp5_MXP_05)'. The menu bar includes 'Tune', 'Edit', 'Goto', 'Environment', 'Monitor', 'System', and 'Help'. The toolbar contains various icons for navigation and analysis. Below the toolbar, there are tabs for 'Current parameters', 'Overview', 'Quotas', and 'Ext. memory blocks'. The main content area displays the following information:

System: mxapp5_MXP_05 SAP memory: Mode list
 Date & time of snapshot: 10/23/2002 17:45:40 Startup: 10/21/2002 00:49:52

No.	Name	Attchd	Ext Mem [kB]	Heap [kB]	I node Clob1/kB	E node Clob1/kB	I node 0 [kB]	I node 1 [kB]	I node 2 [kB]	I node 3 [kB]	I node 4 [kB]	I node 5 [kB]	I no [k
1	ANNCUIAR		12408	0	1	0	1,890	3,987	6,530	0	0	0	
2	AARIASC		6,374	0	1	0	1,881	2,021	2,471	0	0	0	
3	ACASTROL		126852	0	1	0	1,888	4,094	128870	0	0	0	
4	ACASTROL		109374	0	1	0	1,888	3,342	104144	0	0	0	
5	ACORDANA		9,065	0	1	0	1,761	7,302	0	0	0	0	
6	ACUATEPC		1,742	0	1	0	1,742	0	0	0	0	0	
7	ADELAROSA		3,843	0	1	0	1,615	2,227	0	0	0	0	
8	ADELAROSA	X	3,216	0	1	0	758	2,457	0	0	0	0	
9	ADELAROSA		1,141	0	1	0	758	383	0	0	0	0	
10	AGARCIAO		3,459	0	1	0	1,881	1,577	0	0	0	0	
11	ALARAR		4,749	0	1	0	1,881	2,867	0	0	0	0	
12	ALARAR		11089	0	1	0	1,881	4,534	4,673	0	0	0	
13	ASOLISS		8,994	0	1	0	1,634	5,568	1,799	0	0	0	
14	CBUMSQH		2,515	0	1	0	1,753	761	0	0	0	0	
15	CHEMDO2P		2,902	0	1	0	2,142	759	0	0	0	0	
16	DLARAU		1,742	0	1	0	1,741	0	0	0	0	0	
17	EGASCAT		9,673	0	1	0	1,379	8,293	0	0	0	0	
18	EM-CONTEXT		448	0	1	0	447	0	0	0	0	0	
19	FHEMDEZT		7,760	0	2	0	1,877	5,881	0	0	0	0	
20	FHEMDEZT		1,879	0	1	0	1,877	0	0	0	0	0	
21	HSOSAC		1,758	0	1	0	1,758	0	0	0	0	0	
22	HZARCOE		4,627	0	1	0	1,374	1,629	1,623	0	0	0	
23	IESCOBAC		1,881	0	1	0	1,881	0	0	0	0	0	
24	IMORALEO		367	0	1	0	367	0	0	0	0	0	
25	JOUILAH		10396	0	1	0	1,750	8,646	0	0	0	0	

The status bar at the bottom right shows '0.05 | mxapp5 | INS | 17:57'.

4.2.4 Transacciones con mayor consumo de memoria

Una vez identificada la transacción por usuario se puede consultar las tareas realizadas en el día utilizando la transacción stad

The screenshot shows the 'Workload: Statistical Records' window in SAP. The window title is 'Workload: Statistical Records' and it includes a menu bar with 'Workload', 'Edit', 'Tools', 'Goto', 'Monitor', 'System', and 'Help'. Below the menu bar is a toolbar with various icons. The main content area displays the following information:

System: HXP Number of RFCs which responded (without errors): 6 (6)
 Analysed time: 10/23/2002 / 00:19:00 - 10/24/2002 / 02:29:00 Time Frame: +/- 04:02:00
 Display mode: All statistic records, sorted by time

Started	Server	Transaction	Program	T Scr. Up	User	Memory used (kB)
		*	*	*	ACASTROL	
16:09:13	mcapp5_HXP_05	FS10H	RFGLBALANCE	D 1000 0	ACASTROL	3,803
16:09:16	mcapp5_HXP_05	FS10H	RFGLBALANCE	D 1000 0	ACASTROL	3,803
16:09:17	mcapp5_HXP_05	FS10H	RFGLBALANCE	D 0030 0	ACASTROL	4,391
16:09:46	mcapp5_HXP_05	FS10H	RFITEHGL	D 0120 0	ACASTROL	91,924
16:09:49	mcapp5_HXP_05	FS10H	RFITEHGL	D 0120 1	ACASTROL	79,655
16:15:52	mcapp5_HXP_05	FS10H	RFITEHGL	D 0101 1	ACASTROL	79,771
16:16:00	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:16:06	mcapp5_HXP_05	FS10H	RFITEHGL	D 0101 0	ACASTROL	92,040
16:16:09	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 0	ACASTROL	186,238
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:04	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:05	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:05	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:05	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:05	mcapp5_HXP_05	FS10H	RFITEHGL	D 0200 1	ACASTROL	161,701
16:19:05	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 1	ACASTROL	161,699
16:19:41	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 0	ACASTROL	186,238
16:19:41	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 0	ACASTROL	186,238
16:19:41	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 0	ACASTROL	186,238
16:19:41	mcapp5_HXP_05	FS10H	RFITEHGL	D 0100 0	ACASTROL	186,238

The status bar at the bottom right shows '0.27' and '(mcapp5) INS 17:56'.

De esta manera se puede identificar plenamente que usuario / transacción cantidad de memoria.

Idealmente estos trabajos deben ser programados como proceso de fondo durante la noche y con esto se evitaría problemas de consumo excesivo de memoria.

4.2.5 Uso del programa memlimits

Memlimits es un programa diseñado para determinar el límite de memoria que puede direccionarse desde el sistema operativo. Al ejecutar el programa desde la línea de comando se obtendrá lo siguiente:

```

mxpro:mxpadm 1% memlimits
SAP R/3 address space configuration test tool V3.2 (98/01/14)
=====

Check the maximum data size per process (malloc)
Check the available swap space (malloc in several processes)
Process 26770 allocating ... Size = 4096MB Total: 4096MB
Upper size limit 4096 reached. Finish checking

Total available swap space = 4096MB

Check the maximum size of mapped file (mmap anonymous,dev/zero)
Check protection operations on this area (protect)
Trying to mmap      512MB ... successfull
Trying to mmap      768MB ... successfull
Trying to mmap     1024MB ... successfull
Trying to mmap     1280MB ... successfull
Trying to mmap     1536MB ... successfull
Trying to mmap     1792MB ... successfull
Trying to mmap     2048MB ... successfull
Trying to mmap     2304MB ... successfull
Trying to mmap     2560MB ... successfull
Trying to mmap     2816MB ... successfull
Trying to mmap     3072MB ... successfull
Trying to mmap     3328MB ... successfull
Trying to mmap     3584MB ... successfull
Trying to mmap     3840MB ... successfull
Trying to mmap     4096MB ... successfull

Upper size limit 4096 reached. Finish checking
Trying to mprotect 4096MB ... successfull

Maximum mapped file size: 4096MB
Maximum mprotect size:   4096MB

R/3 parameter em/initial_size_MB up to 4096 permitted

Check the maximum address space per process usable
both by process local memory and mapped file

Maximum address space ( mmap(2044 MB)+ malloc(4096MB) ): 6140MB

+-----+
|                               |
|                               |
+-----+
Maximum heap size per process.....: 4096 MB
  measurement limited to 4096 MB
  this value is probably limited by swap space
Maximum mapped file size (mmap).....: 4096 MB
  measurement limited to 4096 MB
Maximum protectable size (mprotect)..: 4096 MB
  measurement limited to 4096 MB
  em/initial_size_MB > 4096 MB will not work
Maximum address space per process....: 6140 MB
Total available swap space.....: 4096 MB
  measurement limited to 4096 MB
  main memory size x 3 recommended , minimum 1 GB

```

Para sistemas UNIX de 64 bits se debe ejecutar una variante de memlimits para determinar el máximo rango de memoria a direccionar.

```
memlimits -l 20000
```

Los puntos importantes de esta salida son:

- **Maximum heap size per process.** Cantidad de memoria local puede ser direccionada por un proceso de trabajo R/3.
- **Maximum protectable size (mprotected).** Este valor limita la cantidad de memoria extendida que R/3 puede direccionar.
- **Maximum address space per process.** Indica el máximo espacio direccionado por proceso de trabajo de R/3. Limita la suma de todas las áreas de memoria que un proceso de trabajo puede direccionar.

4.3 Desempeño de la base de datos (Oracle).

La base de datos es un componente crucial en un sistema R/3, debido que en ella se concentra la actividad de administración de datos un cuello de botella genera una baja en el desempeño general del sistema independientemente de la capacidad de los servidores de aplicación.

4.3.1 Optimizador de costos

Oracle, así como todo RDBMS, ofrece la capacidad de seleccionar la mejor manera de resolver alguna petición de acceso a datos de la base de datos sin necesidad de definirlo explícitamente. El optimizador de costos determina esta estrategia de acuerdo a los siguientes parámetros:

- Tablas consultadas
- Campos especificados en la cláusula WHERE de la sentencia SQL
- Índices creados para las tablas

El optimizador calcula el costo de diferentes estrategias con base a información estadística de las tablas e índices de la base de datos, por ejemplo:

- Número de registros en la tabla e índices y el número de bloques alojados para cada objeto.
- Número de valores distintos en cada columna de la tabla.

Esta información estadística es colectada por la instrucción SQL de Oracle `analyze table`. Si estos datos estadísticos no son actuales entonces pueden ocurrir potencialmente problemas de desempeño.

SAP ofrece mecanismos para la actualización de estadísticas de Oracle por medio del programa SAPDBA el cuál es descrito a continuación.

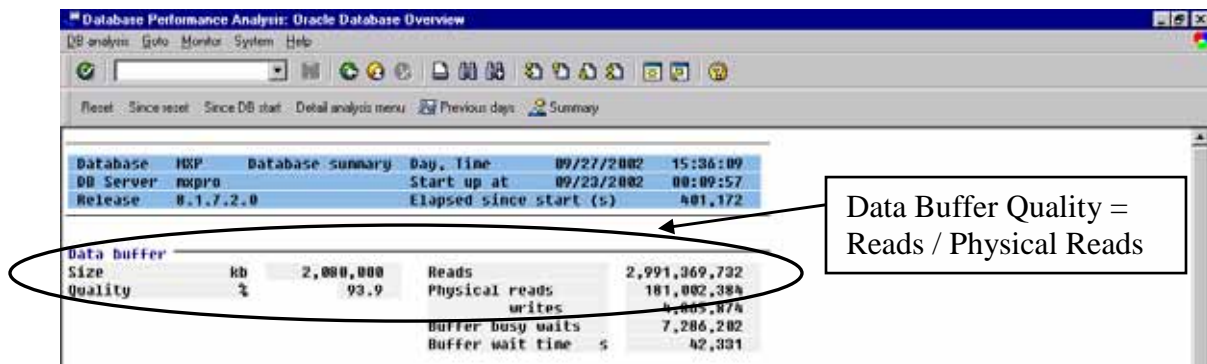
4.3.2 Buffer de datos

Una lectura física debe acceder al disco para obtener datos de la base de datos. Cuando una lectura física ocurre, una copia del bloque de dato es escrito al buffer de datos y después leído y analizado por los procesos shadow de Oracle.

Una lectura lógica solo lee el bloque de datos del área del buffer de datos. Este acceso es 1000 veces más rápido que un acceso a disco.

Por otra parte, cuando se realiza una actualización el bloque de dato es actualizado en el buffer de datos inmediatamente y después escrito al disco de manera asíncrona.

El hit ratio o calidad del buffer de la base de datos esta definido como el porcentaje de bloque de datos accedados (Physical Reads + Logical Reads) comparado contra el número total de lecturas a disco (Physical Reads). Este hit ratio es desplegado por SAP con la transacción ST04 y debe ser mayor a 94%.



El hit ratio es muy bajo después de las primeras horas que se inicio el servidor de base de datos, se debe evaluar el hit ratio después que el sistema ha estado en operación por un tiempo. Un parámetro general puede ser después de 20,000,000 de lecturas (Reads).

Si el hit ratio es inadecuado es necesario considerar los siguiente:

- Verificar que el procedimiento para generar estadísticas sea adecuado ([Cambio del procedimiento estándar de SAP de actualización de estadísticas](#)), para las tablas implicadas en la sentencia SQL.
- Crear un mensaje en OSS si la sentencia corresponde a programa estándar de SAP.
- Rediseñar la aplicación si es un desarrollo propio.
- Verificar si existen [sentencias SQL costosas](#).
- Verificar la paginación del sistema operativo. Si el equipo página quiere decir que falta memoria física.

- La limitación física del sistema operativo por direccionar segmentos de memoria. Las áreas de data buffer, shared pool y log buffer no debe rebasar este limite.

En caso de no tener problema con los puntos anteriores y se tiene un bajo hit ratio entonces es necesario ampliar el área del buffer de datos. Este se realiza modificando el archivo `$ORACLE_HOME/dbs/init<SID>.ora` a nivel UNIX incrementado el parámetro `db_block_buffers`.

La ventaja de mantener afinado el buffer de datos afinado es reducción del acceso a disco.

4.3.3 Shared pool

El shared pool consiste en dos áreas:

- **Shared SQL area.** area de memoria donde las sentencias SQL ya parseadas son almacenadas para compartirse con todos los procesos shadow.
- **Row Cache.** área de memoria donde se mantiene la información del diccionario de datos de Oracle, incluso las estadísticas para el optimizador de costos.

Una llamada de usuario (User call) se refiere a un proceso shadow accediendo al Shared SQL Area para cada sentencia parsea SQL.

Una llamada recursiva (Recursive call) se refiere al Row Cache realizando una lectura física para cargar objetos del diccionario de datos de Oracle del tablespace system.

La eficiencia del shared pool tiene varios indicadores que pueden ser visualizados con la transacción ST04.

The screenshot shows the Oracle Database Performance Analyzer (ST04) interface. The 'Shared Pool' section displays the following metrics:

Size	kb	441,256	DB-Cache quality %	92.3	Entries	1,120
Quality	%	96.8	SQL Area getratio %	98	Allocation retries	3,654,714
			pinratio %	99	Alloc fault rate %	0.0
			reloads/pins %	0.011	Redo log wait s	13

The 'Calls' section displays the following metrics:

User calls	12,512,647	Recursive calls	8,177,986
commits	70,745	Parses	505,818
rollbacks	2,385	User/Recursive calls	1.5
		Reads / User calls	45.1

Hand-drawn annotations on the screenshot include:

- Annotation 1: "1. >= 2" pointing to the "Reads / User calls" value of 45.1.
- Annotation 2: "2. > 80" pointing to the "DB-Cache quality %" value of 92.3.
- Annotation 3: "3. >=95%" pointing to the "SQL Area getratio %" value of 98.
- Annotation 4: "4. <= 0.04" pointing to the "reloads/pins %" value of 0.011.

1. La proporción de User Calls entre Recursive Calls debe ser al menos 2.
2. La calidad de la calidad el Data Dictionary Cache debe ser mayor a 80%

3. El pinratio debe ser mayor o igual a 95%
4. La proporción de reloads entre pins debe ser máximo de .04

La eficiencia del Shared Pool debe verificarse después que el sistema ha estado en operación por un tiempo. Un parámetro general puede ser después de 20,000,000 de lecturas (Reads).

En caso que el tamaño del shared pool no sea adecuado es posible que sea consecuencia de la creación de estadísticas. Para descartar esta situación hay que revisar la bitácora de días anteriores (DB24->Performance).

Si el tamaño de shared pool es inadecuado es necesario considerar lo siguiente:

- Verificar la paginación del sistema operativo ([Uso de Memoria](#)). Si el equipo página quiere decir que falta memoria física.
- La limitación física del sistema operativo por direccionar segmentos de memoria. Las áreas de data buffer, shared pool y log buffer no debe rebasar este limite.

En caso de no tener problema con los puntos anteriores es necesario ampliar el área del Shared Pool. Este se realiza modificando el archivo \$ORACLE_HOME/dbs/init<SID>.ora a nivel UNIX incrementado el parámetro shared_pool_size.

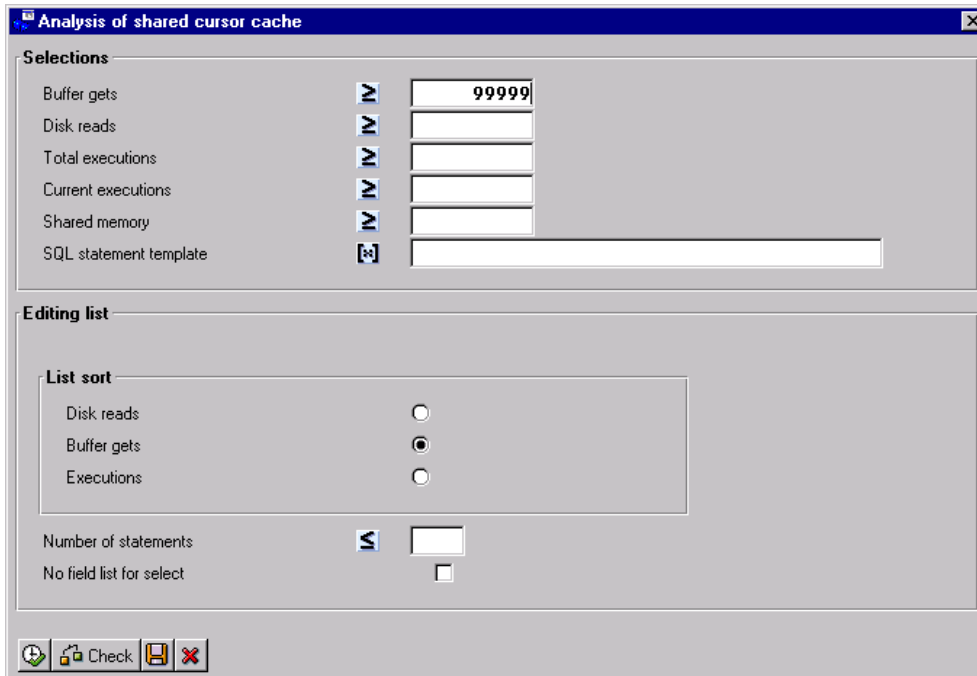
4.3.4 Sentencias SQL costosas

Una sentencia SQL mal diseñada puede generar cuellos de botella en todo el sistema. Si el acceso a los datos se realiza no utilizando el mejor índice o listando todos los campos de una tabla cuando solo requiere algunos de ellos provocará que la base de datos tenga que mover bloques de memoria del shared pool para poder alojar los nuevos solicitados.

Un indicador de sentencias SQL costosas es la proporción de Reads entre User Calls en la transacción ST04 este valor debe ser menor que 20.

Database Performance Analysis: Oracle Database Overview			
DB analysis Go to Monitor System Help			
Reset Since reset Since DB start Detail analysis menu Previous days Summary			
Database	HXP	Database summary	Day, time 09/30/2002 15:33:36
DB Server	mxpro	Start up at	09/30/2002 00:32:07
Release	8.1.7.2.0	Elapsed since start (s)	54,009
Data buffer			
Size	kb	2,080,000	Reads 731,221,194
Quality	%	96.4	Physical reads 25,962,153
			writes 575,982
			Buffer busy waits 1,360,810
			Buffer wait time s 23,782
Shared Pool		Log buffer	
Size	kb	441,256	Size kb 1,120
DB-Cache quality	%	92.6	Entries 5,100,398
SQL Area getratio	%	91	Allocation retries 764
pinratio	%	100	Alloc fault rate % 0.0
reloads/pins	%	0.010	Redo log wait s 20
			Log files (in use) 10(1)
Calls			
User calls		19,007,556	Recursive calls 8,511,524
commits		105,665	Parses 553,829
rollbacks		2,426	User/Recursive calls 2.2
			Reads / User calls 38.5
Time statistics			
Busy wait time	s	566,161	Sessions busy % 9.25
CPU time	s	80,017	CPU usage % 20.34
Time/User call	ms	34	CPU count 8

SAP permite acceder la información del shared pool sobre las estadísticas sentencias SQL ejecutadas con la siguiente ruta ST04->Detail Analisis Menu->SQL Request y posteriormente llenar el siguiente cuadro de diálogo con los datos siguientes:



Al final se obtendrá la lista de sentencias SQL del área de memoria del shared pool.

Executions	Curs.Ex	Disk reads	Reads/Exec	Buffer gets	Bgets/Exec	Proc. rows	Rproc/Exec	Bgets/row	SQL sort	SQL statement
189,988	0	455	0.0	207,278,683	1,091.0	119	0.0	1,741,837.7	0	SELECT * FR
121,213	0	406	0.0	123,289,010	939.0	94	0.0	1,310,724.1	0	SELECT * FR
2,676,700	1	1,132,569	0.4	12,321,726	4.6	12,849	0.0	959.0	0	SELECT * FR
11	0	200,676	18,243.3	11,837,874	1,076,170.4	465,304	42,300.4	25.4	0	SELECT "HRLNR
9,595	0	211,071	22.0	11,487,062	1,197.2	1,511,482	157.5	7.0	0	SELECT * FR
1,828,418	0	699,782	0.4	8,437,700	4.6	5,026	0.0	1,678.8	0	SELECT * FR
4	0	597,707	149,426.0	8,173,699	2,043,424.0	1,108	284.5	7,182.5	0	SELECT T_00 .
29	0	0	0.0	6,985,180	240,868.3	29	1.0	240,868.3	29	select 'DATAP
991,818	0	1,287,092	1.3	6,633,877	6.7	74,085	0.1	89.5	0	SELECT /*+ FI
2	0	641,477	320,738.5	6,464,497	3,232,248.5	887	443.5	7,288.0	0	SELECT T_00 .
3	0	441,473	147,157.7	6,009,062	2,003,020.7	2,251	750.3	2,669.5	0	SELECT T_00 .

Las Sentencias SQL costosas tienen número alto de *Buffer Gets per Row*

El detalle de las instrucciones SQL el Shared SQL Area muestra el nombre del programa que los esta ejecutando:

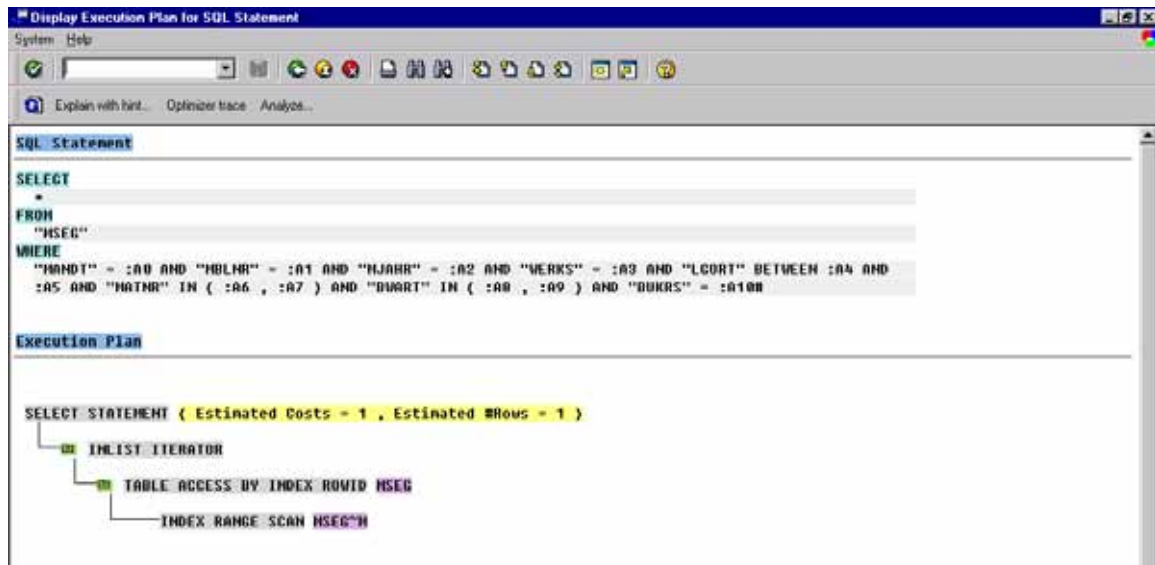
Invalid	Loads	Parse call	Program name	First load time
0	1	2	ZMM_REPORTE_CAJAS_C%	2002-09-30/08:52:55
0	1	7	ZMM_REPORTE_CAJAS_C%	2002-09-30/12:11:51
0	1	30	ZMM_REPORTE_CAJAS_C%	2002-09-30/06:24:04
0	1	4	ZMMR_CONCILIACION_1%	2002-09-30/12:35:02
0	1	1	SAPLABRA	2002-09-30/08:28:19
0	1	25	ZMM_REPORTE_CAJAS_C%	2002-09-30/07:05:00
0	1	4	SAPINBBS	2002-09-30/09:18:05
0	1	29	SAPLPIUS	2002-09-30/01:15:25
0	1	10	ZMM_BUEGAS_COMPLET%	2002-09-30/00:34:43
0	1	1	SAPINBBS	2002-09-30/08:31:53

Y adicionalmente la sentencia SQL directamente:

```

SELECT
*
FROM
"MSEG"
WHERE
"MANDT" = :A0 AND "MBLNR" = :A1 AND "MJAHR" = :A2 AND "WERKS" = :A3 AND
"LGORT" BETWEEN :A4 AND :A5 AND "MATNR" IN ( :A6 , :A7 ) AND "BWART" IN ( :A8
, :A9 ) AND "BUKRS" = :A10#
    
```

Una vez que se ha identificado la sentencia SQL con problemas se debe revisar el plan de ejecución de Oracle:



Después de ejecutar el explain plan sobre las la instrucción SQL se debe revisar:

- Verificar las opciones del optimizador de costos para las tablas implicadas.
- Crear un mensaje en OSS si la sentencia corresponde a programa estándar de SAP.
- Rediseñar la aplicación si es un desarrollo propio.
- Verificar si la sentencia SQL fue calificada pobremente.

4.4 Desempeño de hardware.

4.4.1 Procesos con mayor consumo de CPU

Cuando exista un problema de consumo excesivo de CPU la transacción ST06 (monitor de sistema operativo) permite identificar procesos críticos en su ejecución.

La ruta de este acceso es ST06->Detail Analysis Menu->TOP CPU.

Pid	Username	Command	CPU Util [%]	CPU Time [s]	Resident size [kB]	Prior.
5,041	oransp	oracleRSP (DESCRIPT	99.50	188:50	2553,136	59
6,279	ncpadn	du.sapRSP_DUEBMS00	29.27	063:19	179,312	0
8,465	ncpadn	/usr/local/bin/top	19.09	000:23	2,240	0
13,129	oransp	oracleRSP (DESCRIPT	16.12	109:36	2549,320	59
20,063	oransp	oracleRSP (DESCRIPT	11.96	070:30	2549,856	59
3	root	fsflush	4.94	1782:29	0	60
6,230	ncpadn	du.sapRSP_DUEBMS00	3.46	060:06	372,184	12
14,656	root	FatrolAgent	2.86	1786:39	17,640	0
6,485	oransp	oracleRSP (DESCRIPT	1.78	026:54	2539,776	59
7,225	ncpadn	du.sapRSP_DUEBMS00	1.48	004:25	67,464	59
27,772	oransp	oracleRSP (DESCRIPT	1.28	093:16	2546,984	59
8,833	ncpadn	/usr/sap/RSP/SYS/ex	1.18	320:54	2,312	59
20,037	oransp	oracleRSP (DESCRIPT	1.18	005:13	2548,360	59
20,049	oransp	oracleRSP (DESCRIPT	1.18	001:40	2548,776	59
12,036	oransp	oracleRSP (DESCRIPT	0.98	062:54	2549,264	59
5,111	oransp	oracleRSP (DESCRIPT	0.89	149:52	2549,752	59
29,442	oransp	oracleRSP (DESCRIPT	0.89	026:43	2541,856	59
4,752	ncpadn	opa_lque_RSP	0.69	023:47	2529,424	59
1,101	root	/opt/VRTSsona/jre/b	0.59	055:47	11,368	59
20,116	oransp	oracleRSP (DESCRIPT	0.59	019:52	2548,040	59
7,408	oransp	oracleRSP (DESCRIPT	0.49	001:59	2535,696	59

Para una instancia central solo procesos Oracle oracle<SID> y SAP dw.sap<SID> deberán estar consumiendo recursos considerables de cpu, en caso que procesos deferentes sean responsables de la degradación entonces se deben eliminar o ejecutar en otro servidor.

Para una servidores de aplicación solo se deben esperar procesos SAP dw.sap<SID>.

Cuando los procesos corresponden a SAP u Oracle entonces se debe considerar lo siguiente:

- Analizar a detalle el proceso que se encuentra consumiendo los recursos y tratar de afinarlo relacionando id de proceso oracle (ST04->Oracle Session) y el id del proceso de trabajo (SM50).
- El administrador puede distribuir la carga de los work process hacia otros servidores de aplicación, utilizando los modos de operación RZ04, RZ03 y SM63.
- Aumentar la capacidad del equipo

4.4.2 Interfaces de RED

La transacción ST06->Detail Analsis Menu->Compare Recent Days muestra si existen retransmisiones o perdidas de paquetes directamente registradas en las interfaces de red del servidor de aplicación.

The screenshot shows the SAP ST06 interface for 'Compare Recent Days' for server '69.908'. The table displays various system metrics. A red circle highlights the network-related columns: 'Lan packets/h', 'Lan errors/h', and 'Lan collisions/h', all of which show zero values across all time intervals.

Date/time	Pages out/h		Swap space			Lan packets/h		Lan errors/h		Lan collisions/h	
	avg	max	avg	max size	min free	max	avg	max	avg	max	avg
09/25/2002 09:30:48	0	0	0	0	0	0	0	0	0	0	0
d 1h-15	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0
69,908	154,139	97,73h	8,19h,14h	4,265,285	0	0	0	0	0	0	0

Normalmente no deben existir errores ni colisiones, en caso contrario es necesario revisar la interfaz de red del servidor y en general la LAN

4.4.3 Verificar la red vía ping

La conexión hacia los clientes es verificada enviando un ping, SAP incorpora esta funcionalidad dentro de la transacción ST06->Detail Analisis Menu -> LAN check by ping.

Para el envío de un paquete de 4Kb se espera:

- No tener pérdidas de paquetes
- Tiempos de respuestas:
 - < 50 ms en LAN
 - < 150 ms en WAN
 - < 500 ms en Modems de 56K

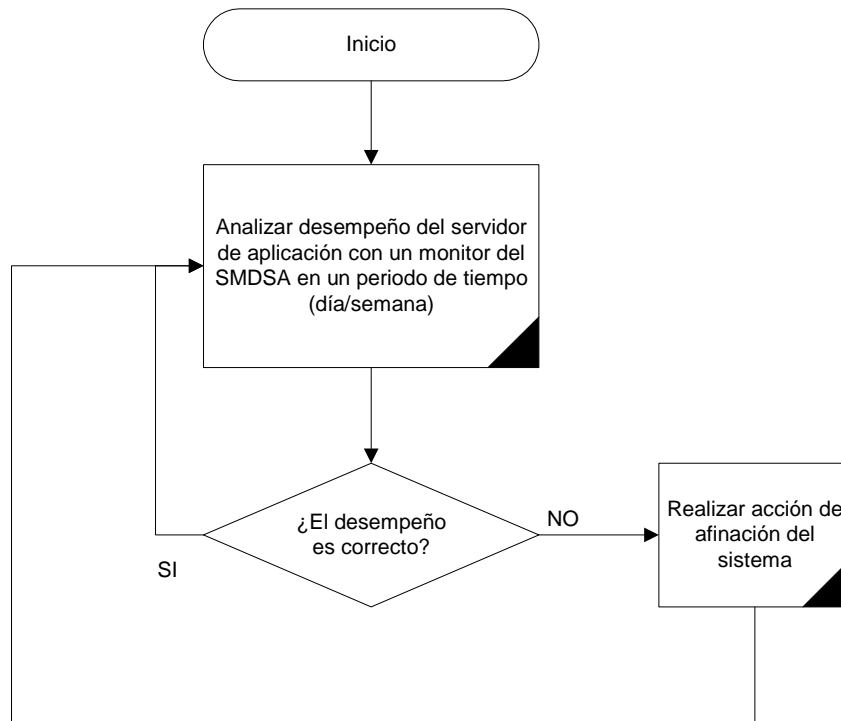
Si estas premisas no son satisfechas se debe revisar el enlace hasta el punto donde se encuentra el cliente SAPGUI.

Servername	Server-IP	Min ms	Avg ms	Max ms	Loss %
bodega	183.165.100.148	574	676	914	
g18521	181.249.100.122	438	473	525	
guclb29	188.87.100.26	620	649	717	
gurhb41	188.87.100.255	733	803	869	
n31694	181.249.100.20	406	455	471	
n1jui01	184.130.100.120	4742	9387	14587	
		436	2,074	14587	

5 Diagrama de flujo con las acciones de afinación requeridas.

Los siguientes mapas tienen como objeto orientar las posibles acciones de afinación del sistema basado en los documentos existentes de SAP y en mi experiencia propia.

El proceso de decisión supone que se realizan mediciones del mismo intervalo y duración que el SMDSA genera. Este proceso quedará representado de la siguiente manera:

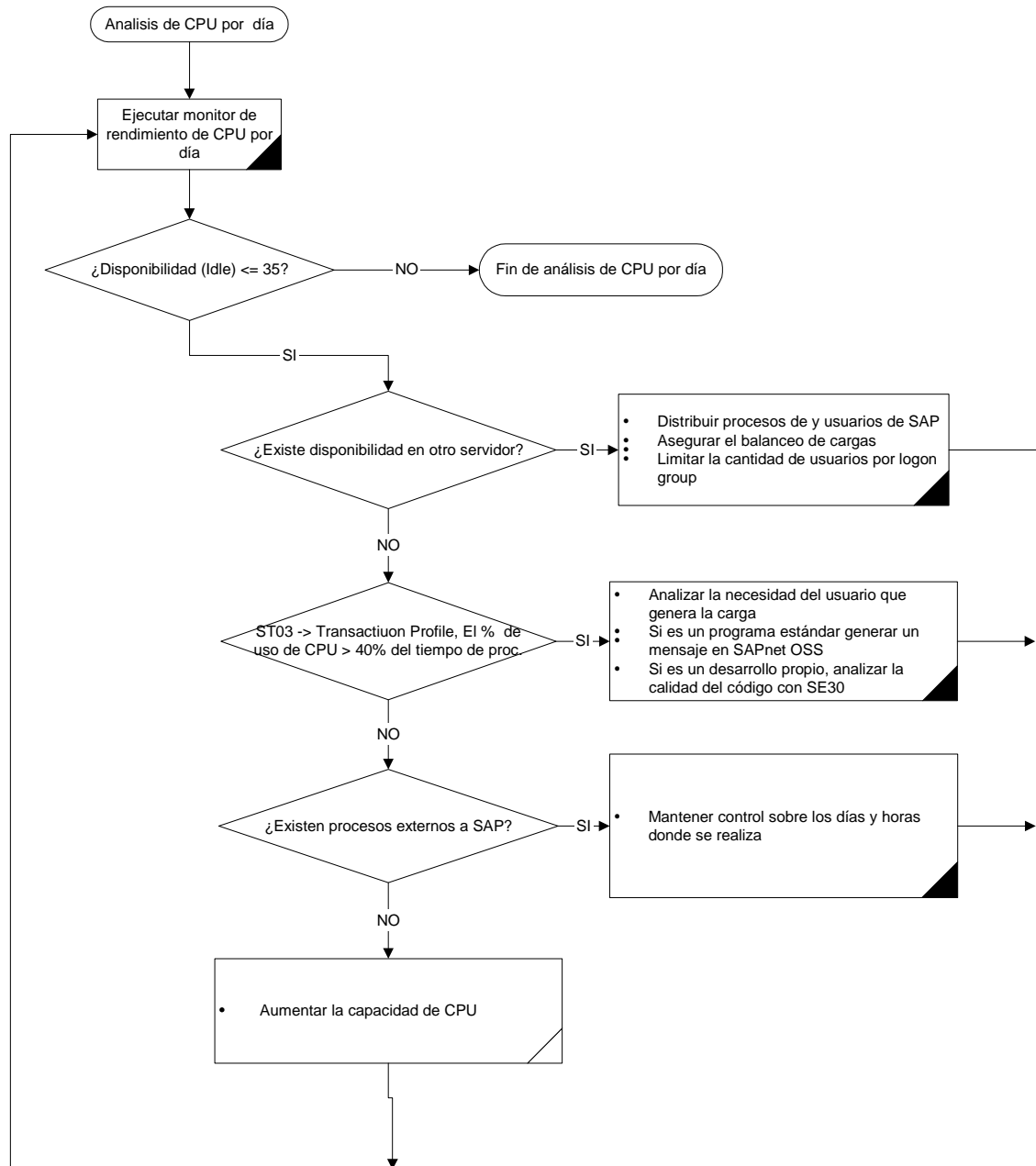


Como se puede observar este es un ciclo infinito el cuál representa el proceso de afinación como un proceso continuo.

5.1 Diagrama análisis de CPU

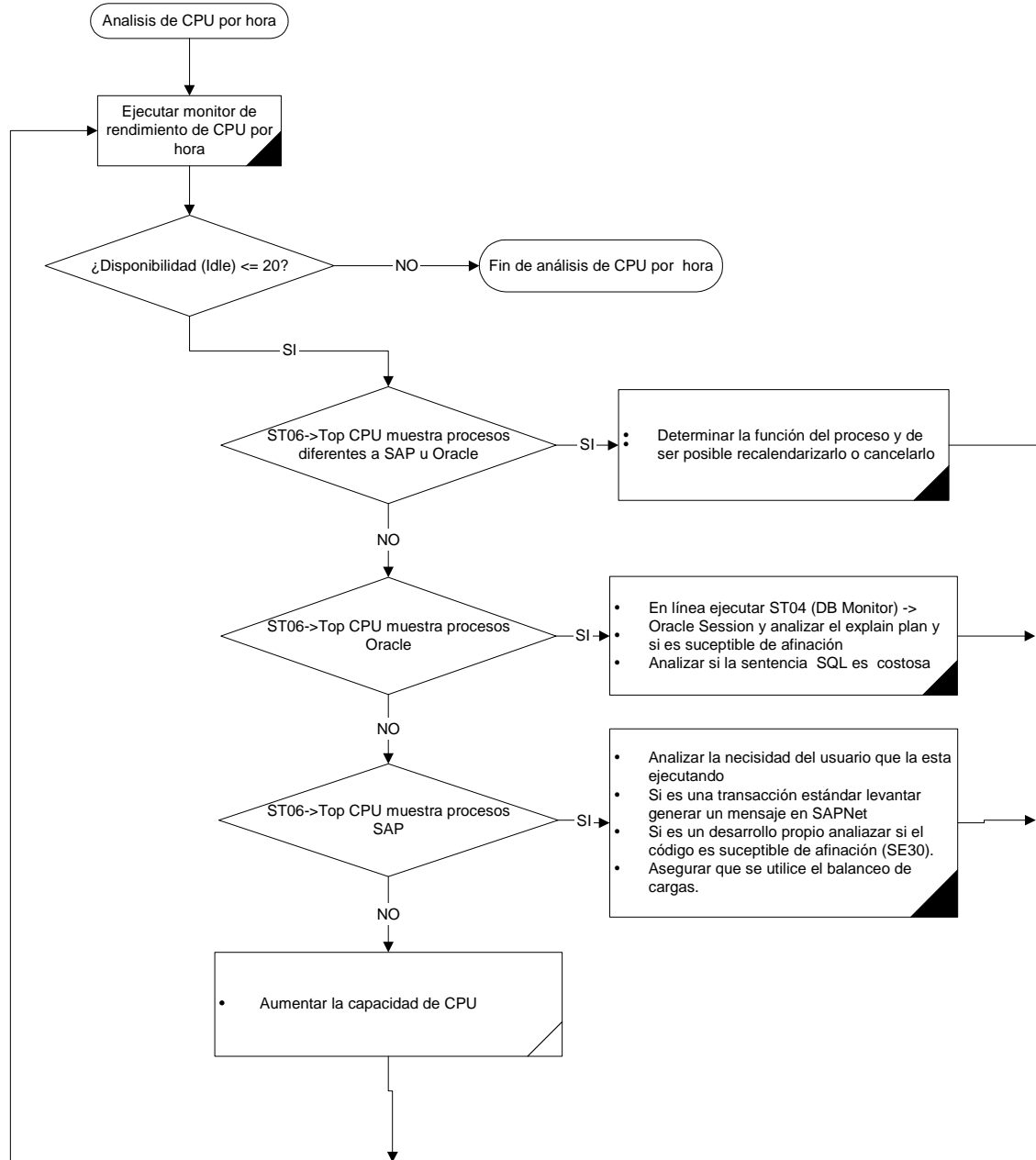
5.1.1 Monitor de CPU por día

El siguiente mapa describe el análisis de CPU sobre el tiempo. Este monitor permite analizar de manera global el desempeño del sistema en un periodo de tiempo extenso (días, semanas o meses).



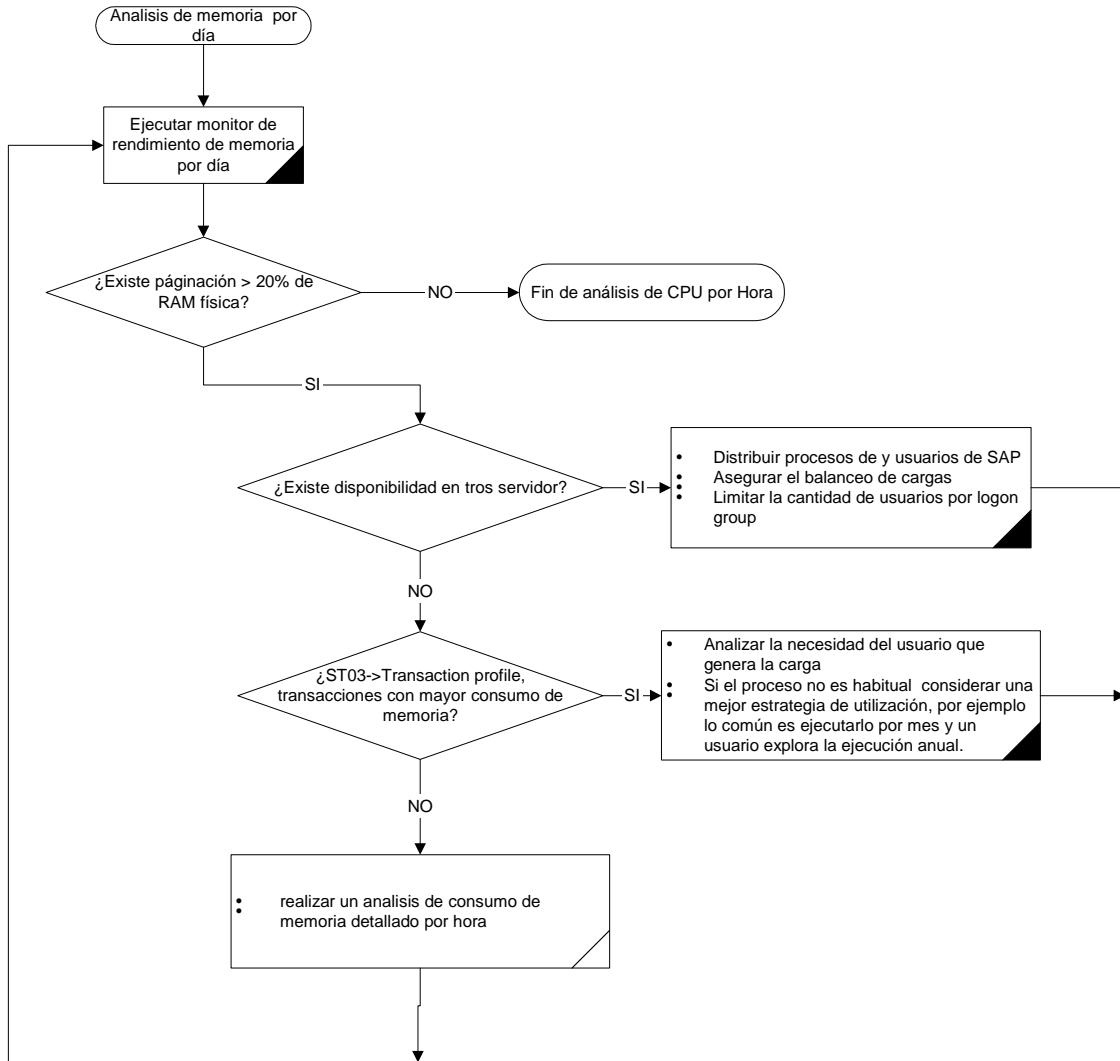
5.1.2 Monitor de CPU por hora

El siguiente mapa describe el análisis de CPU en tiempo real orientándose a los proceso que se están ejecutando. Debe utilizarse como complemento del monitor por día, el detalle presentado permitirá profundizar en la causa raíz por la que el uso CPU pueda estar saturado.

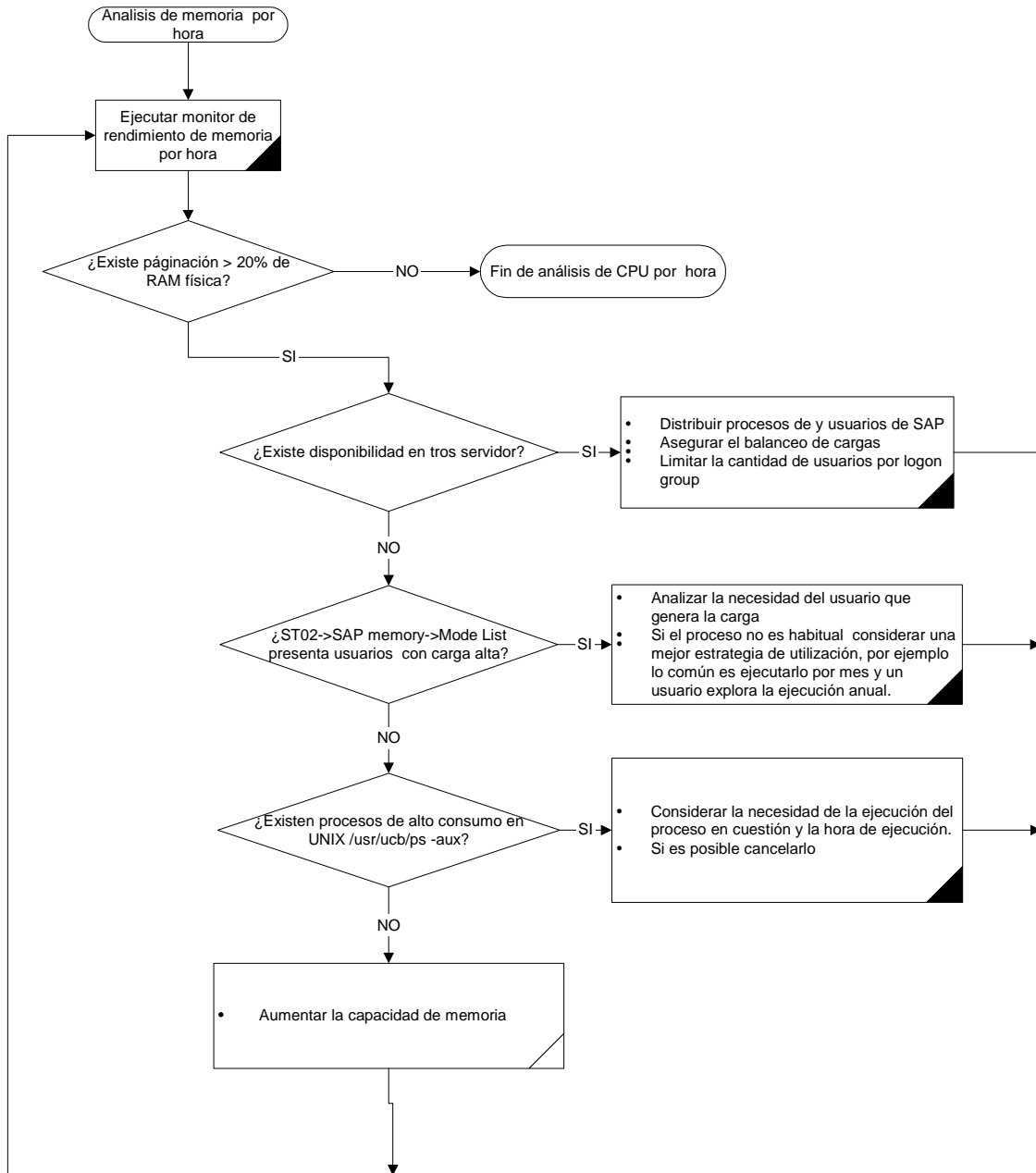


5.2 Diagrama de análisis de memoria.

5.2.1 Monitor global de memoria por día.



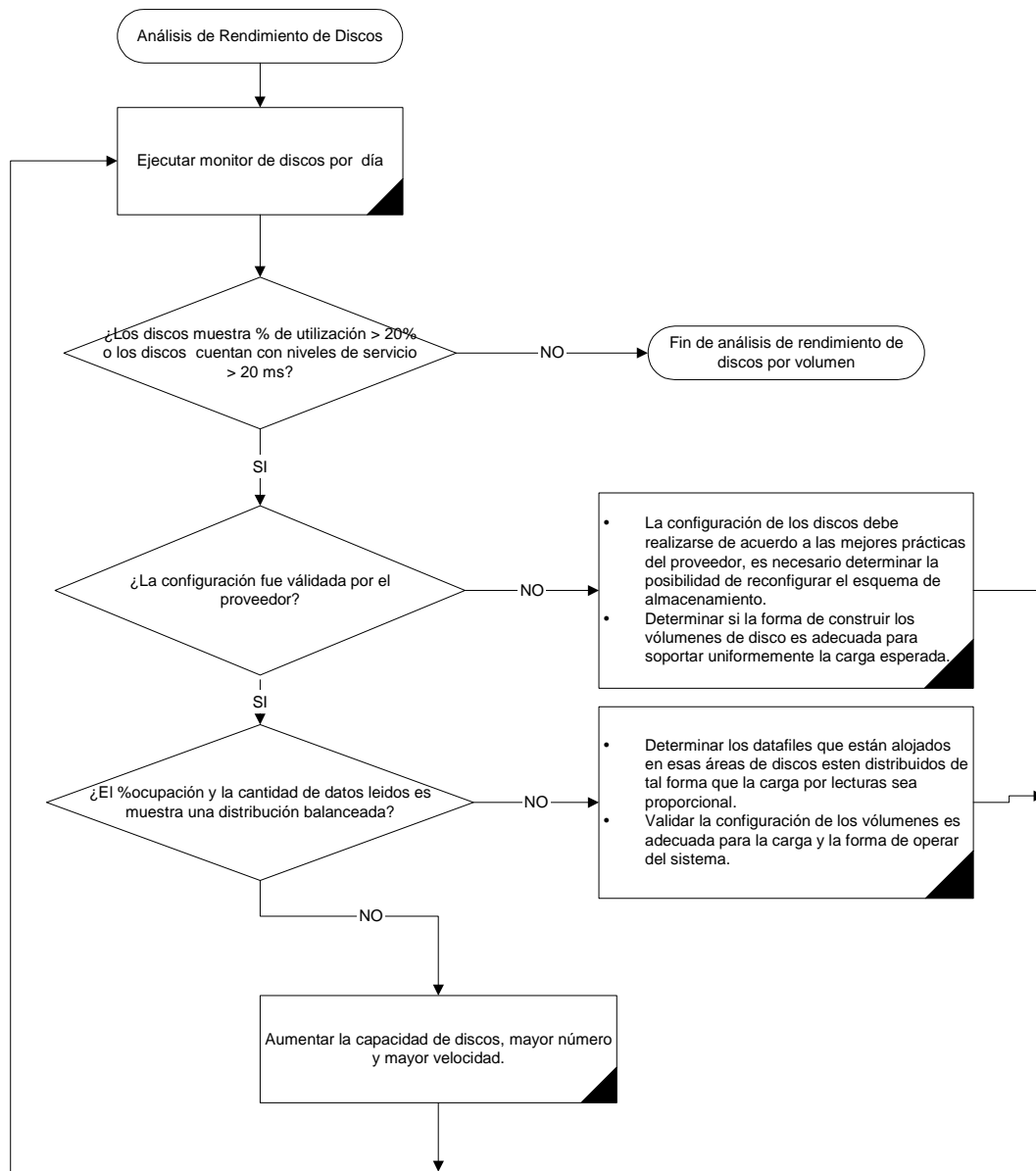
5.2.2 Monitor global de memoria por hora.



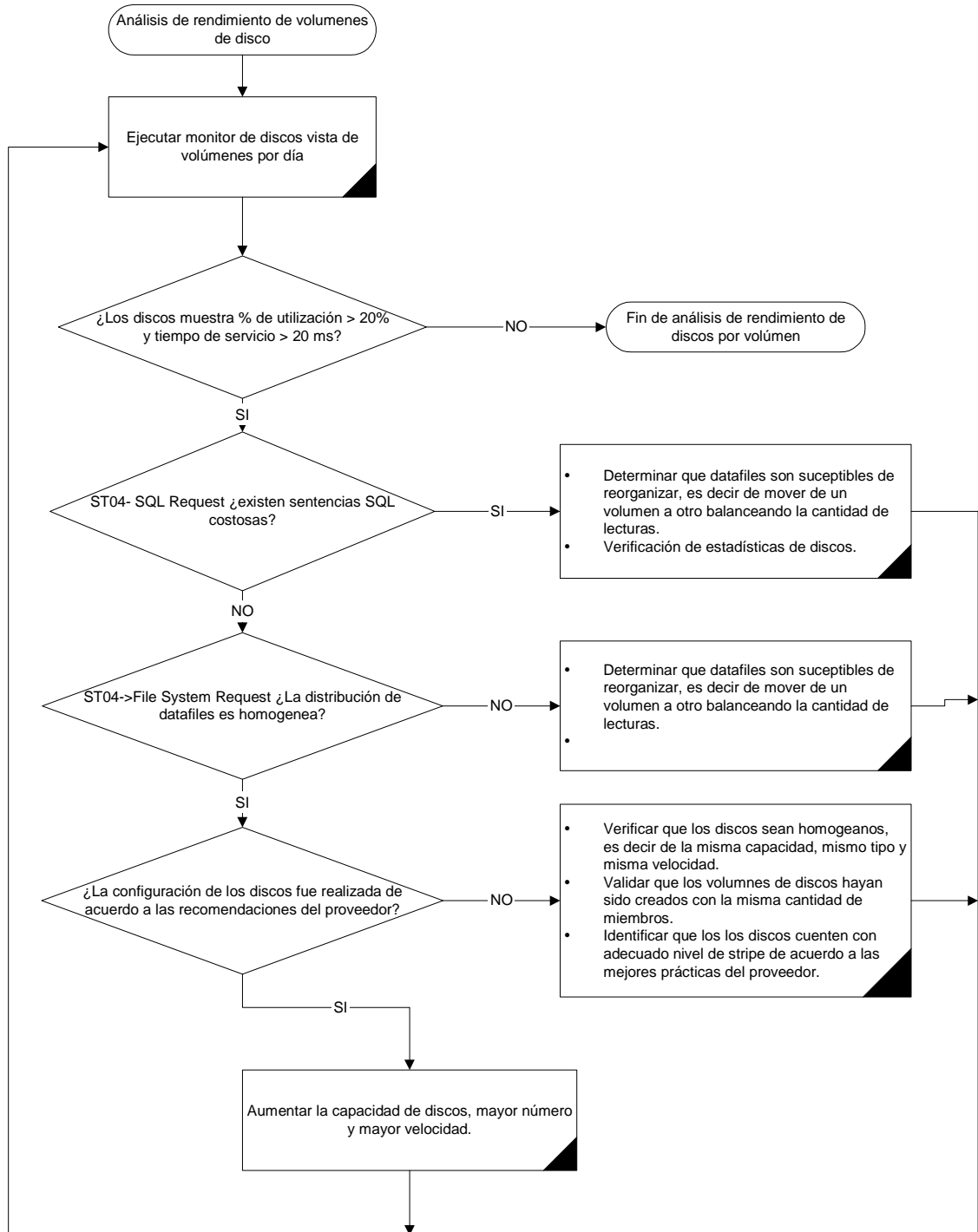
5.3 Diagrama de análisis de rendimiento de disco

Un rendimiento óptimo del disco depende en gran medida de la estrategia de configuración del equipo de almacenamiento. Esta actividad es de suma importancia, la experiencia en varias implantaciones me ha permitido comprobar que un error en la configuración impactará negativamente el desempeño general del sistema y tendrá un costo posterior muy alto en tiempo de paro del sistema para corregirlo.

El primer mapa de desempeño esta orientado a determinar el rendimiento general de los discos y si la configuración de los discos es eficiente.



El siguiente mapa de desempeño considera que existe una distribución de cargas por volúmenes.



En un ambiente típico SAP R3 el almacenamiento requerido puede alcanzar cantidades de datos de Terabytes. Esta cantidad de almacenamiento normalmente implica el uso de varias decenas de discos por lo que el monitorear cada uno de ellos puede perder objetividad.

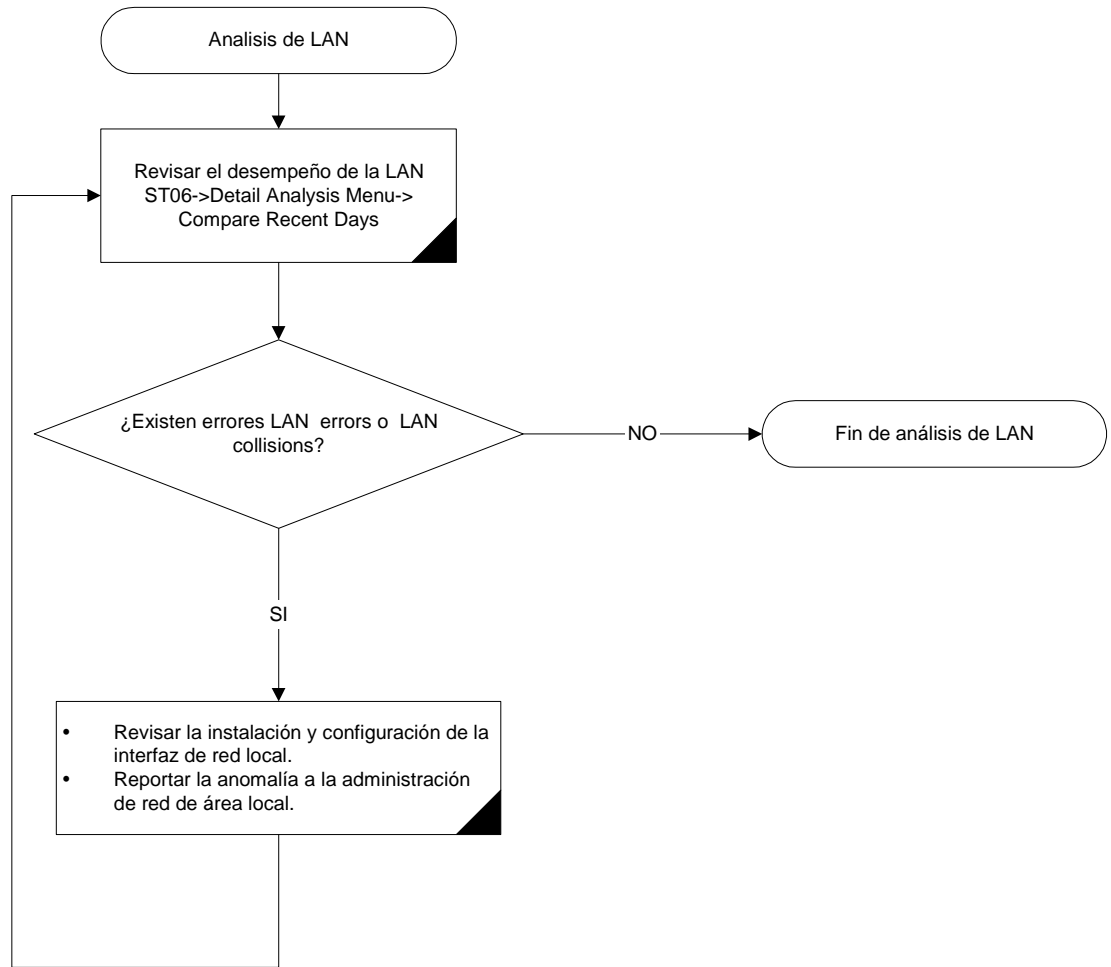
Los discos en UNIX normalmente son administrados con software de administración de volúmenes (el más común es Veritas Volume Manager tanto en SUN como HP). Con estas herramientas los discos son configurados en conjunto para almacenar áreas de datos conocidas como sapdatas o en terminología de Oracle datafiles.

Lo primero que debe asegurar el administrador es un nivel de stripe adecuado en los discos, es decir que se distribuya la carga en todos los discos del grupo. Esto puede realizarse por hardware o por software, en caso que sea por software es imprescindible la utilización de una herramienta especializada de volúmenes.

Bajo el supuesto que los discos se encuentran configurados y agrupados de manera adecuada el primer mapa es validar que las cargas en ambos se encuentre distribuido lo más uniforme posible.

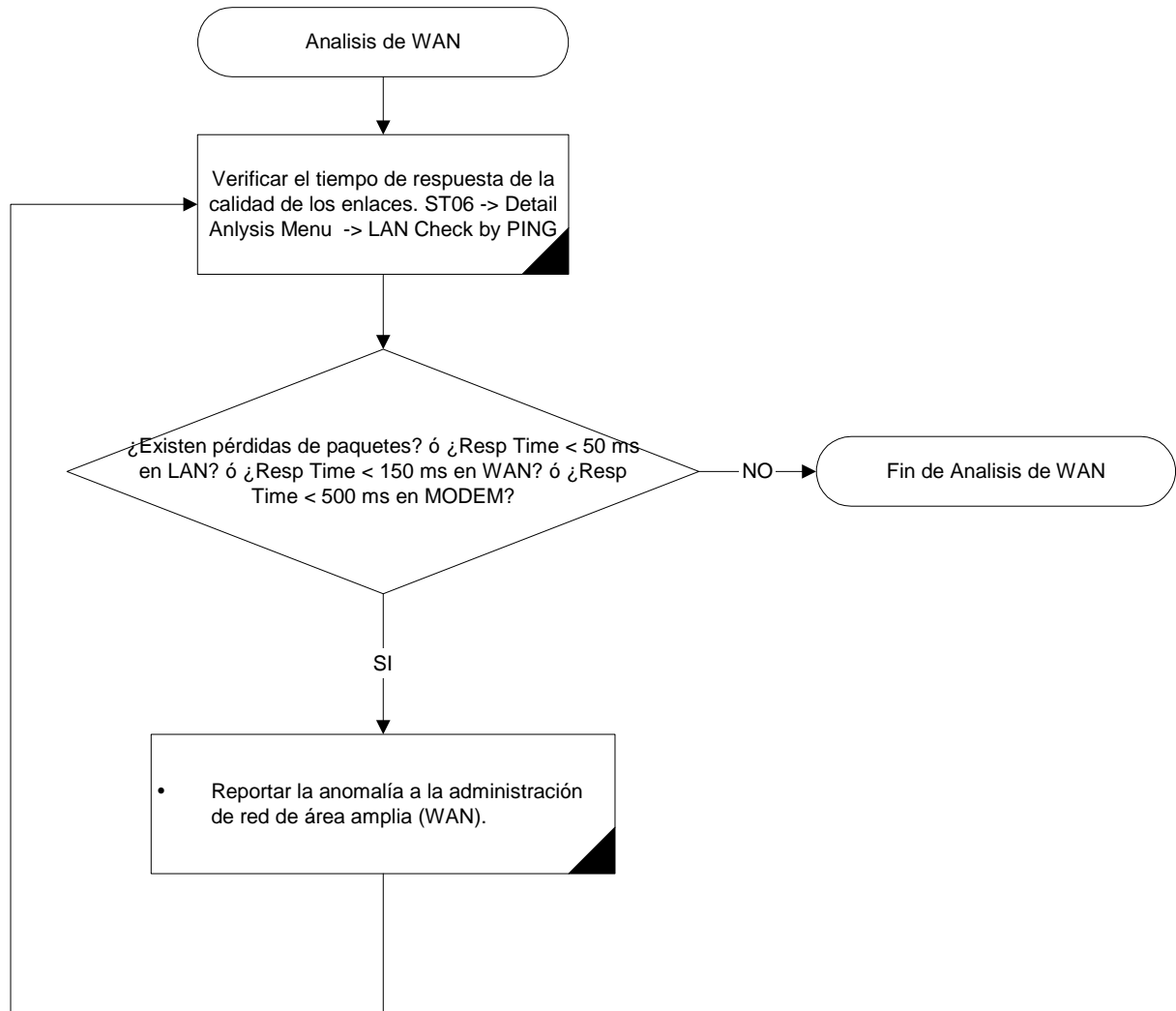
5.4 Diagrama de análisis LAN

El mapa de desempeño se enfoca a la no existencia de colisiones o errores en la transmisión de paquetes. Las tareas que el administrador del sistema debe realizar se limita a asegurar que la interfaz de red se encuentre correctamente instalada y configurada, después de esto si el problema continúa debe reportarse a la administración de red local.



5.5 Diagrama de análisis WAN

Desde el punto de vista de un administrador de SAP el comando de sistema operativo PING es la herramienta de análisis para determinar la calidad del el enlace. Un problema de enlace normalmente tiene la característica de afectar a solo un región de usuarios mientras que el resto trabaja adecuadamente.



6 Casos prácticos

6.1 Saturación de CPU por trabajos del sistema inadecuadamente configurados

La inadecuada configuración de los trabajos del sistema puede saturar por completo con recursos de un equipo, el primer caso presentado corresponde a sistema donde dos procesos consumen una proporción muy grande de CPU.

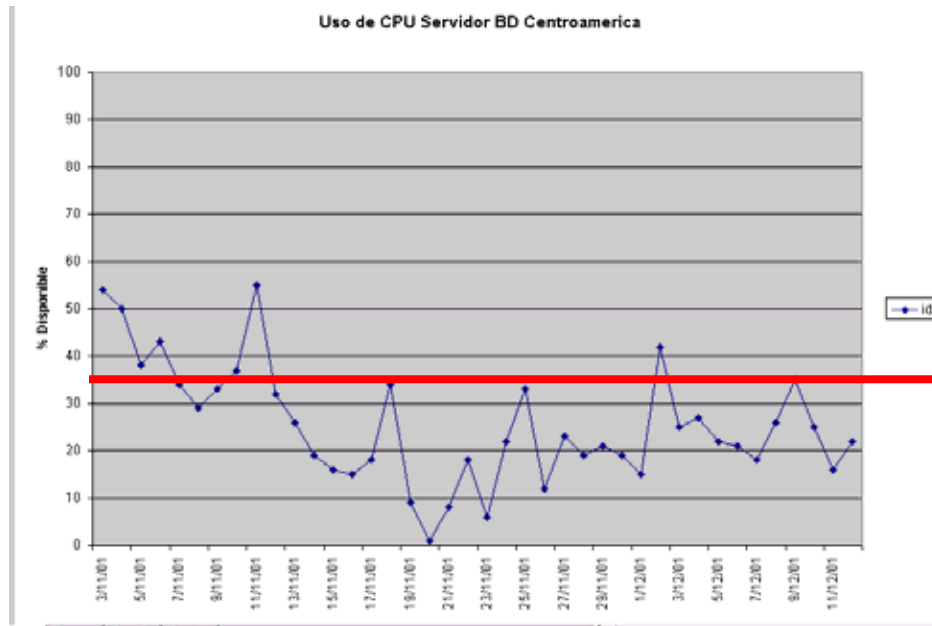
La consola del SMDS de monitor de CPU por día indica que los recursos de CPU de la equipo en revisión se encuentra saturado:

```

=====
CPU por Pais centro_america 2003/03/04
P: /oracle/MXD/sapreorg/tesis/menu/pais_sar_cp
Unbrates US:20 40 SY:20 40 IO:20 40 ID:40 20
=====
          crpro          crapp1          crapp2
  us sy io id eq us sy io id eq us sy io id eq
01/11/03 36 7 2 54 0 2 2 7 09 0 1 1 4 93 0
01/11/04 37 10 4 50 0 2 2 2 93 0 4 2 3 91 0
01/11/05 45 14 4 38 0 11 4 5 81 0 14 4 2 80 0
01/11/06 42 11 4 43 0 5 2 4 09 0 10 2 1 86 0
01/11/07 50 11 5 34 0 14 2 3 81 0 9 2 1 88 0
01/11/08 50 16 5 29 0 10 6 4 80 0 19 6 3 72 0
01/11/09 50 12 5 33 0 8 3 2 87 0 8 3 1 88 0
01/11/10 49 10 4 37 0 3 2
01/11/11 35 7 3 25 0 2 2
01/11/12 51 13 3 32 0 10 3
01/11/13 61 11 3 26 9 4
01/11/14 70 9 2 19 2 10 1
01/11/15 72 10 2 16 3 8 1
01/11/16 74 9 2 15 2 9 1
01/11/17 72 8 2 18 2 6 1
01/11/18 57 7 2 34 1 2 1
01/11/19 70 20 1 9 4 15 1
01/11/20 75 23 0 1 6 7 1
01/11/21 74 17 1 8 5 9 1
01/11/22 69 11 2 19 2 7 1 3 88 0 9 1 2 88 17
01/11/23 81 11 1 6 4 7 1 2 90 0 9 1 2 88 24
01/11/24 86 9 2 22 2 20 1 2 76 0 6 1 2 92 32
01/11/25 50 9 7 33 2 30 2 1 67 1 4 1 1 94 19
01/11/26 73 13 3 12 3 36 2 2 60 6 13 1 2 84 16
01/11/27 62 10 4 23 2 20 2 3 67 13 14 2 2 83 19
01/11/28 66 11 4 19 2 17 2 4 77 13 20 2 1 77 13
  
```

La disponibilidad de CPU se encuentra por debajo del 35% en el monitor de desempeño de CPU y los procesos de usuario US están excesivamente altos.

La gráfica de este comportamiento permite observar visualmente esta saturación.



De acuerdo al mapa de decisión para CPU como primer paso se recomienda analizar las transacciones con mayor consumo. Esto se realiza ejecutando la transacción ST03, el resultado obtenido fue el siguiente:

Program or Tcode	Background job	Dialog steps	Response time total (s)	Response time avg (ms)
RFC		950,537	1,193,505	1,256
RFDRRGEN	COMPAUTDEUDTICA	362	922,995	2,549,709
RFDRRGEN	EVALDEUDNICA	359	911,171	2,538,081
RFDRRGEN	EVALDEUDTICA	360	899,359	2,498,220
RFKRRGEN	EVALACRE	8,630	544,645	63,111
YTRSCOLL	OPTISYSTEMS_DATA_COLLECT	20	247,265	12363228
SESSION_MANA		74,050	234,508	3,167
MB51		55,771	226,000	4,052
RSDBAJOB	DBA:ALLONLINE_____@180	27	217,189	8,044,053
ME51N		44,166	181,909	4,119
AutoABAP		25,615	179,758	7,018
RSAL_BATCH_T	SAP_CCMS_MONI_BATCH_DP	687	121,391	176,697
ME53N		37,883	109,502	2,891
ZMMB		1,794	108,160	60,290
YWAC0000	YWA_COLLECTOR	361	98,178	271,961
ME21N		50,617	94,373	1,864

El primer programa RFC, se identifico como interfaces de movimientos de inventarios y se catalogaron como normales de operación.

Los segundo programas corresponden a una actualización de saldos deudores de cuentas por pagar.

Sobre estos programas se analiza los usuarios quienes lo ejecutan y la frecuencia con la que lo realizan:

Program or Tcode	Background job	Usuario	Frecuencia	Clase Job	Server
RFDRRGEN	COMPAUTDEUDTICA	TICCALVO	Cada 2 hrs	C	Balanceo
RFDRRGEN	EVALDEUDNICA	NIYMOLINA	Cada 2 hrs	C	Balanceo
RFDRRGEN	EVALDEUDTICA	TICCALVO	Cada 2 hrs	C	Balanceo
RFKRRGEN	EVALACRE	TIMCHACON	Cada 1 hr	A	crpro

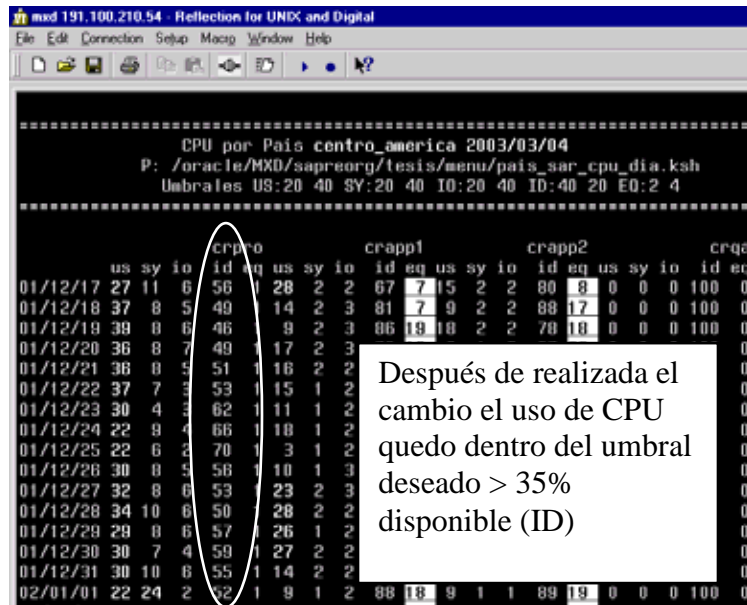
La forma en que se estaban ejecutando era diariamente cada 2 horas, sin embargo los trabajos tenían duración de 2.5 horas o 3 horas lo que indicaba que uno de ellos no había terminado cuando los otros ya estaban procesándose, algo totalmente incongruente.

Al cuestionar a los responsables de los trabajos sobre su necesidad operativa respondieron que estos eran necesarios como parte de la parametrización del modulo de Cuentas por Pagar. Se solicito soporte a SAP e indico que únicamente era necesario ejecutar 2 de ellos una sola vez al día, se aplico esta recomendación programándose durante la madrugada.

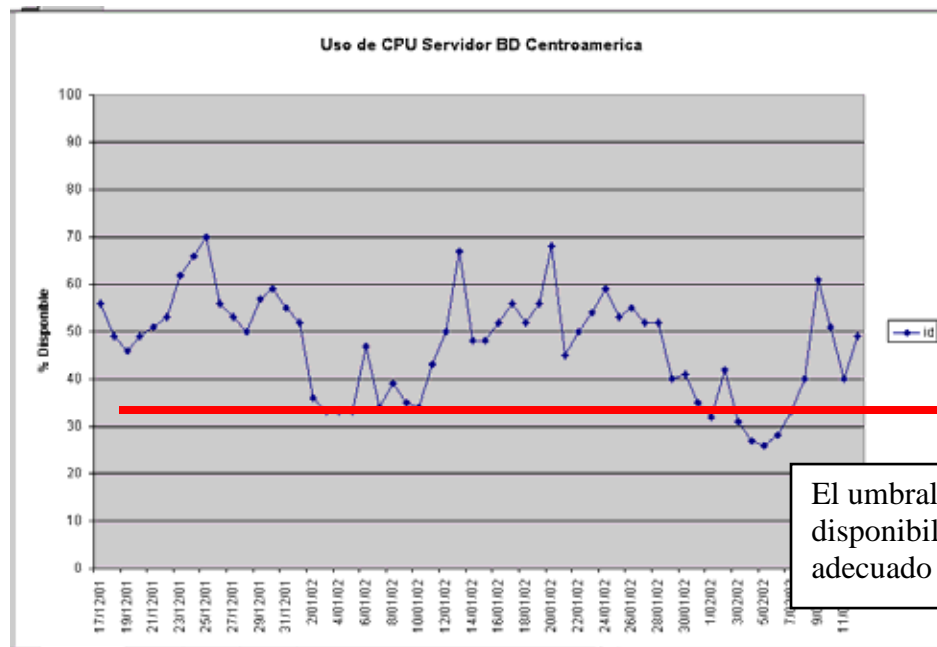
De tal forma que solo quedaron programados 2 tareas:

Program or Tcode	Background job	Usuario	Frecuencia	Clase Job	Server
RFDRRGEN	EVALDEUDNICA	NIYMOLINA	Diario	C	Balanceo
RFDRRGEN	EVALDEUDTICA	TICCALVO	Diario	C	Balanceo

El resultado en el monitor de desempeño de CPU después de esta afinación fue el siguiente:



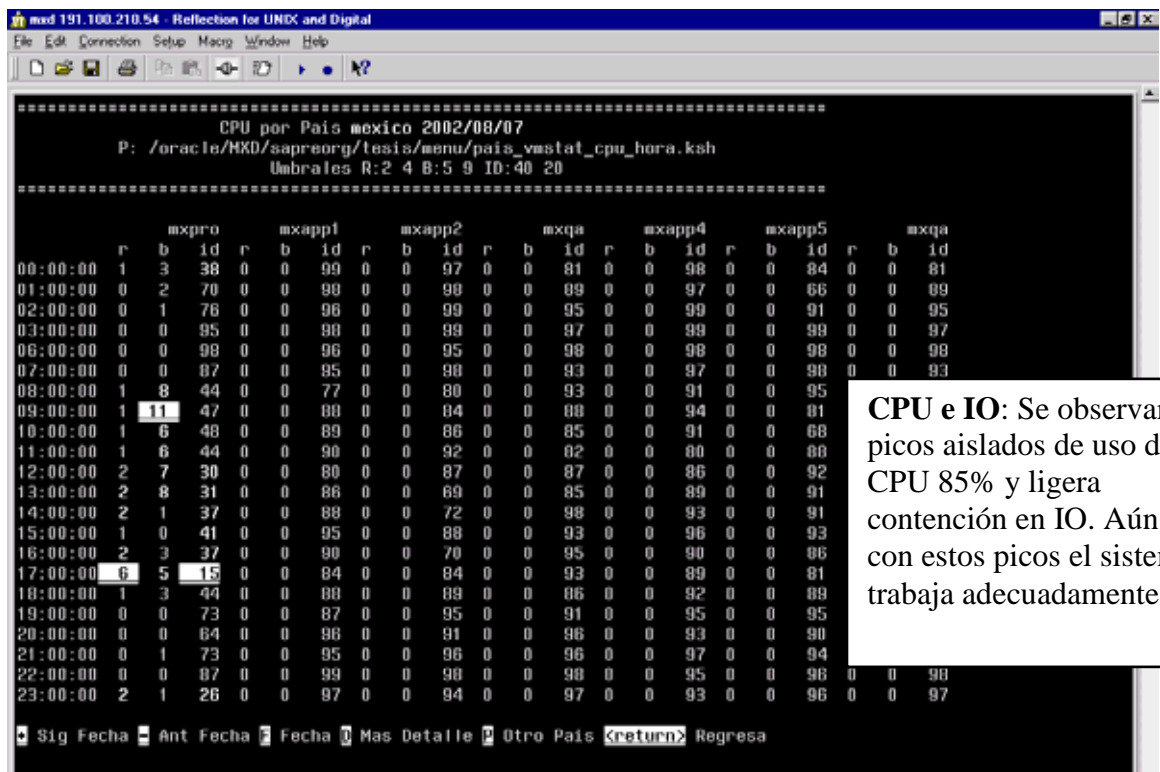
Gráficamente el resultado es más evidente:



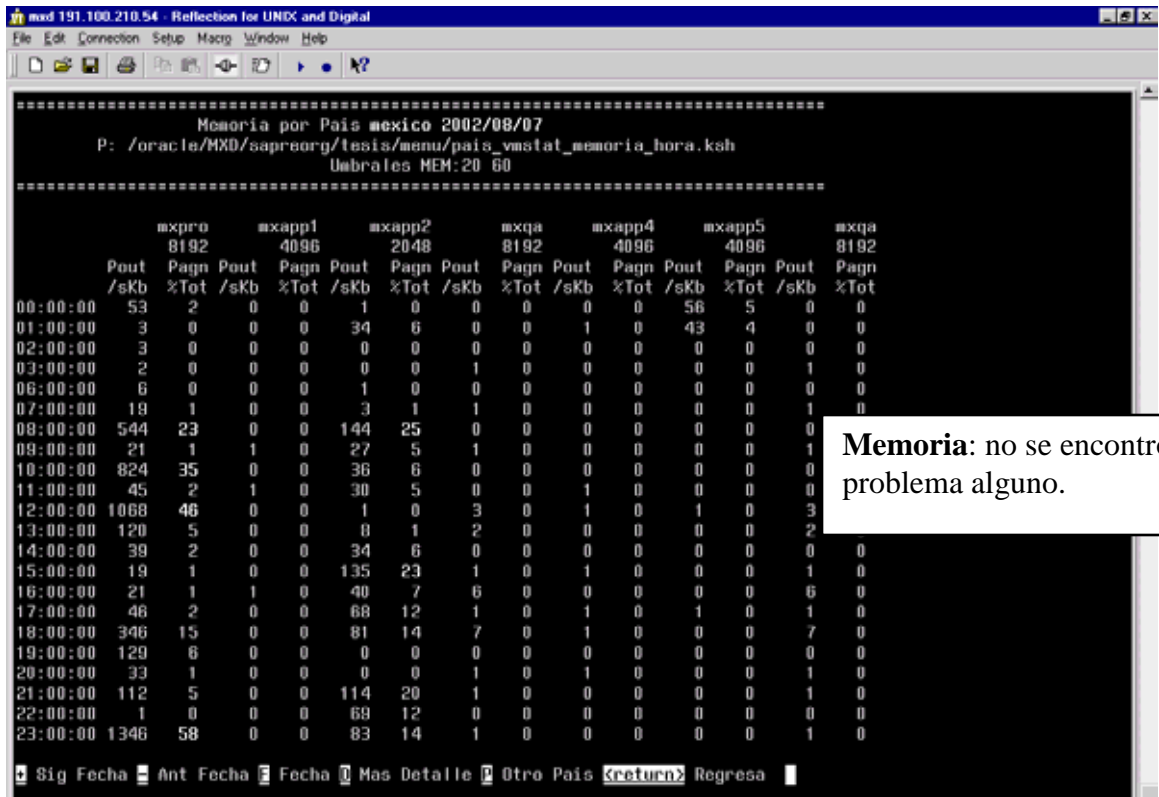
6.2 Problemas de contención en disco por procesos externos

El siguiente caso corresponde un problema reportado como lentitud extrema presentada en algunos días sin justificación inmediata a tal grado que ningún usuario podía trabajar sin ninguna justificación inmediata.

Se instaló el SMDS y se colectaron datos del rendimiento general en un día laborable con carga habitual que no mostraba la problemática reportada (Miércoles 7 de Agosto 2002).

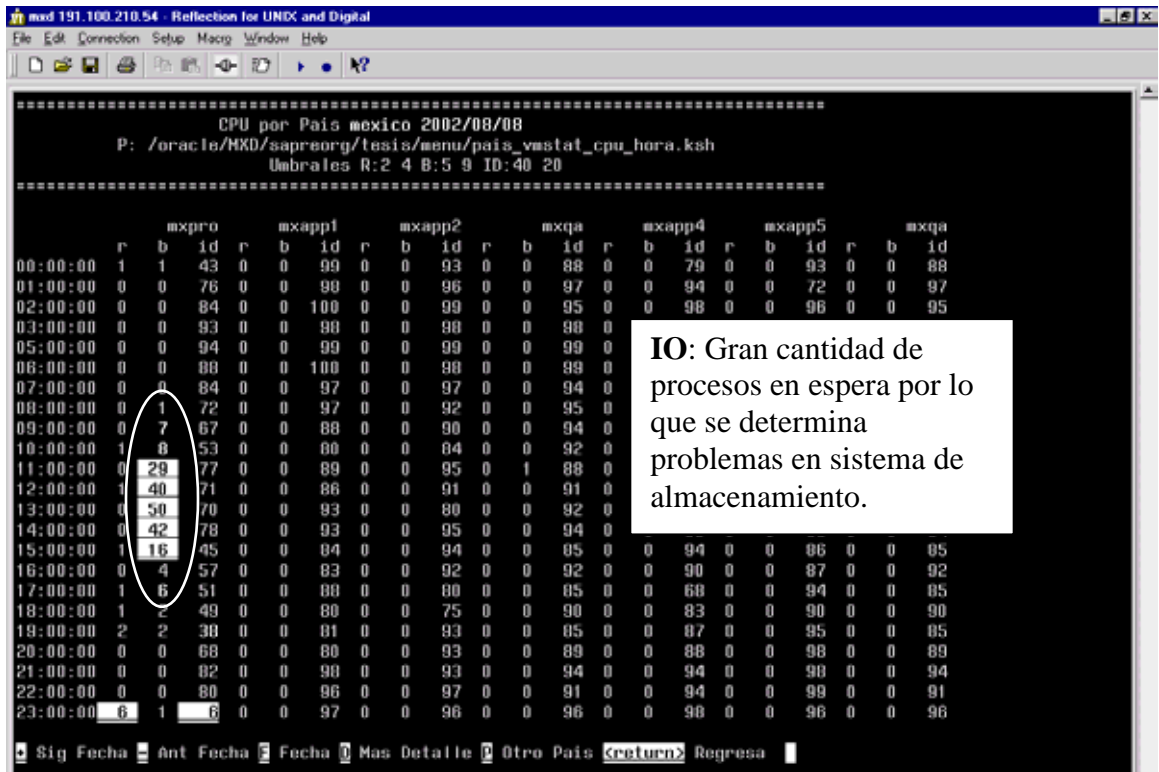


CPU e IO: Se observaron picos aislados de uso de CPU 85% y ligera contención en IO. Aún con estos picos el sistema trabaja adecuadamente.



Memoria: no se encontró problema alguno.

Un día posterior se presentó la problemática y se utilizaron los mismos monitores de rendimiento observándose lo siguiente:



IO: Gran cantidad de procesos en espera por lo que se determina problemas en sistema de almacenamiento.

Memoria por Pais mexico 2002/08/08
P: /oracle/MXD/sapreorg/tesis/menu/pais_vmstat_memoria_hora.ksh
Umbrales MEM:20 60

	mxpro	mxapp1	mxapp2	mxqa	mxapp4	mxapp5	mxqa
	0192	4096	2048	0192	4096	4096	0192
	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout
	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb
00:00:00	30	1	0	76	13	0	0
01:00:00	11	0	1	68	11	1	0
02:00:00	1	0	1	1	0	1	0
03:00:00	1	0	0	12	2	1	0
05:00:00	3	0	1	1	0	0	0
06:00:00	1	0	0	1	0	1	0
07:00:00	9	0	1	1	0	56	2
08:00:00	7	0	0	121	21	1	0
09:00:00	15	1	0	122	21	1	0
10:00:00	36	2	0	54	9	0	0
11:00:00	30	1	0	0	0	4	0
12:00:00	157	7	0	117	20	1	0
13:00:00	115	5	1	72	12	4	0
14:00:00	5	0	0	191	33	2	0
15:00:00	147	6	0	95	16	1	0
16:00:00	92	4	0	93	16	1	0
17:00:00	50	2	1	141	24	2	0
18:00:00	140	6	5	43	7	0	0
19:00:00	92	4	0	14	2	48	2
20:00:00	23	1	0	52	9	64	3
21:00:00	33	1	0	51	9	1	0
22:00:00	21	1	1	3	1	0	0
23:00:00	21	1	0	2	0	6	0

Memoria: no se encontró problema alguno.

Los procesos por espera de IO son un indicador de que existen problemas en los discos. Para verificar esto se analizo el desempeño de estos en ambos días:

DISCO de la Instancia Central Pais mexico 2002/08/07
P: /oracle/MXD/sapreorg/tesis/menu/pais_iostat_hist.ksh
Umbrales SVCT:15 20 PCTB:14 18

ind	Server	cxtXdXsX	vg	vxdisk	sdXXX	scvt	%B	krps	rps	kups	ups
1	mxpro	c0t0d0s2	swap1	rootdisk	sd0	56	2	1	0	31	4
2	mxpro	c1t0d0s2	swap1	disk02	sd15	50	2	3	0	31	4
3	mxpro	c2t0d27s2	md	mxdb05	sd00	18	5	23	1	7	1
4	mxpro	c1t1d0s2	swap2	disk03	sd16	15	0	0	0	4	1
5	mxpro	c0t1d0s2	swap2	disk01	sd1	15	0	0	0	4	1
6	mxpro	c2t0d43s2	datos	mxdb21	sd96	14	16	403	16	2	0
7	mxpro	c2t0d30s2	datos	mxdb00	sd03	13	16	403	16	2	0
8	mxpro	c2t0d29s2	datos	mxdb07	sd82	13	15	401	16	2	0
9	mxpro	c2t0d44s2	datos	mxdb22	sd97	13	15	401	16	2	0
10	mxpro	c2t0d105s2	datos	mxdb01	sd161	13	15	402	16	2	0
11	mxpro	c2t0d31s2	datos	mxdb09	sd84	13	15	400	16	2	0
12	mxpro	c2t0d40s2	datos	mxdb10	sd93	12	15	401	16	2	0
13	mxpro	c2t0d39s2	datos	mxdb17	sd92	12	15	401	16	2	0
14	mxpro	c2t0d20s2	datos	mxdb06	sd01	12	15	403	16	2	0
15	mxpro	c2t0d36s2	datos	mxdb14	sd89	12	15	405	16	2	0
16	mxpro	c2t0d34s2	datos	mxdb12	sd07	12	15	408	16	2	0
17	mxpro	c2t0d38s2	datos	mxdb16	sd91	12	15	407	16	2	0
18	mxpro	c2t0d106s2	datos	mxdb02	sd162	12	15	401	16	2	0
19	mxpro	c2t0d99s2	indices	mxidx07	sd155	12	15	361	16	2	0
20	mxpro	c2t0d37s2	datos	mxdb15	sd90	12	15	397	16	2	0
21	mxpro	c2t0d33s2	datos	mxdb11	sd96	12	15	401	16	2	0
22	mxpro	c2t0d97s2	indices	mxidx05	sd153	12	15	354	16	2	0
23	mxpro	c2t0d41s2	datos	mxdb19	sd94	12	15	407	16	2	0
24	mxpro	c2t0d98s2	indices	mxidx06	sd154	12	15	366	16	2	0
25	mxpro	c2t0d35s2	datos	mxdb13	sd88	12	15	402	16	2	0
26	mxpro	c2t0d96s2	indices	mxidx04	sd152	12	15	362	16	2	0
27	mxpro	c2t0d100s2	indices	mxidx00	sd156	12	15	361	16	2	0

Discos: el monitor de discos muestra el origen de la problemática: los discos en un día normal presentan tiempos de respuesta de promedios máx 16 ms y en el día con problema se observan promedio máx de 124 ms.

```

=====
DISCO de la Instancia Central Pais mexico 2002/08/08
P: /oracle/MXD/sapreorg/tesis/menu/pais_iostat_hist.ksh
Umbrales SVCT:15 20 PCTB:14 18
=====

```

ind	Server	cXtXdXsX	vg	vxdiak	sdXXX	svct	%B	krps	rps	kups	ups
1	mxpro	c2t0d96s2	indices	mxidx04	sd152	124	22	253	12	16	1
2	mxpro	c2t0d97s2	indices	mxidx05	sd153	122	21	250	12	17	1
3	mxpro	c2t0d27s2	md	mxdb05	sd80	120	11	29	1	7	0
4	mxpro	c2t0d95s2	indices	mxidx03	sd15	70	23	250	12	17	1
5	mxpro	c2t0d101s2	indices	mxidx09	sd157	65	23	251	12	16	1
6	mxpro	c0t0d0s2	swapl	rootdisk	sd0	63	2	1	0	32	4
7	mxpro	c2t0d102s2	indices	mxidx10	sd158	57	21	253	12	18	1
8	mxpro	c1t0d0s2	swapl	disk02	sd15	56	2	7	1	32	4
9	mxpro	c2t0d100s2	indices	mxidx00	sd156	53	21	252	12	16	1
10	mxpro	c2t0d31s2	datos	mxdb09	sd84	52	22	312	12	2	0
11	mxpro	c2t0d107s2	datos	mxdb03	sd163	50	24	318	12	2	0
12	mxpro	c2t0d28s2	datos	mxdb06	sd81	49	21	312	12	3	0
13	mxpro	c2t0d93s2	indices	mxidx01	sd113	47	20	250	12	17	1
14	mxpro	c2t0d33s2	datos	mxdb11	sd86	45	23	314	13	2	0
15	mxpro	c2t0d99s2	indices	mxidx07	sd155	43	19	249	12	16	1
16	mxpro	c2t0d34s2	datos	mxdb12	sd87	41	22	310	12	2	0
17	mxpro	c2t0d98s2	indices	mxidx06	sd154	40	20	255	12	16	1
18	mxpro	c2t0d94s2	indices	mxidx02	sd114	38	18	251	12	16	1
19	mxpro	c2t0d32s2	datos	mxdb10	sd85	36	21	310	12	3	0
20	mxpro	c2t0d40s2	datos	mxdb18	sd93	32	19	317	12	2	0
21	mxpro	c2t0d29s2	datos	mxdb07	sd82	32	19	312	12	2	0
22	mxpro	c2t0d41s2	datos	mxdb19	sd94	32	21	314	12	2	0
23	mxpro	c2t0d106s2	datos	mxdb02	sd162	30	18	317	12	2	0
24	mxpro	c2t0d38s2	datos	mxdb17	sd92	30	18	318	13	3	0
25	mxpro	c2t0d42s2	datos	mxdb20	sd95	30	18	314	12	2	0
26	mxpro	c2t0d35s2	datos	mxdb13	sd88	29	20	313	12	2	0
27	mxpro	c2t0d38s2	datos	mxdb16	sd91	29	19	318	13	2	0

Se realizó un análisis a detalle por hora sobre el comportamiento del disco más cargado (sd152) y se encontró lo siguiente:

```

=====
USO DEL DISCO sd152 c2t0d96s2 mxidx04 indices
EN EL SERVIDOR mxpro 2002/08/08
P: /oracle/MXD/sapreorg/tesis/menu/iostat_disco_hora.ksh
Umbrales SVCT:15 20 PCTB:14 18
=====

```

hora	svct	pctb	krps	rps	kups	ups
00:00:00	2739	100	30	1	0	0
01:00:00	12	2	54	2	0	0
02:00:00	6	2	122	4	1	0
03:00:00	12	1	26	1	0	0
05:00:00	1	0	0	0	0	0
06:00:00	5	6	205	15	0	0
07:00:00	8	2	89	3	0	0
08:00:00	15	10	106	8	2	0
09:00:00	27	27	333	15	2	0
10:00:00	24	31	501	26	7	1
11:00:00	76	39	148	8	13	1
12:00:00	295	66	147	10	4	0
13:00:00	243	59	130	9	1	0
14:00:00	174	42	130	9	3	0
15:00:00	26	38	525	29	106	4
16:00:00	15	17	431	17	5	0
17:00:00	16	34	677	41	6	0
18:00:00	10	13	302	19	2	0
19:00:00	8	13	397	18	130	5
20:00:00	10	6	234	8	4	0
21:00:00	7	5	222	8	2	0
22:00:00	11	2	57	2	3	0
23:00:00	8	11	614	16	9	1

Sig Fecha Ant Fecha Cambia vista dia/hora Restar días **return** Regresa

Con esta evidencia se diagnosticó el origen del problema en el sistema de discos. Siguiendo el mapa de análisis se realizaron las siguientes tareas:

- Revisión de los procesos de SAP R3 y se encontró que se habían ejecutado los mismos programas con los mismos usuarios en general.
- Se cuestiono la liberación de algún módulo nuevo o programa resultando negativo.
- Revisión de procesos externos como actualización de estadísticas, respaldos en línea, ejecución de utilerías del sistema operativo, programas adicionales a SAP R3 no encontrándose anomalía alguna.

Con esto se descarto que la sobrecarga de discos proviniera de la aplicación, posteriormente se solicito al administrador del sistema de almacenamiento revisar su calendario de actividades y este reporte que no existía actividad alguna durante esa hora.

Descartando ambos puntos se recomendó solicitar al proveedor de los discos revisar su estado y su configuración.

En este caso el proveedor explico el área de almacenamiento era compartido por varios sistemas y que existía la posibilidad de que otro sistema estuviera provocando la saturación.

A revisar a detalle la administración del dispositivo de almacenamiento de los otros sistemas se encontró una actividad crítica realizada de manera incorrecta denominada sincronización de discos.

Este proceso facilita la realización de respaldos vía una tercera copia de discos. Antes de separarse esta copia debe sincronizarse con las otras dos copias generando una sobrecarga en sistema. Para esta actividad existen dos tipos de sincronización completa o incremental. El proveedor recomienda realizar una sincronización completa solo al iniciar un ciclo de respaldos y durante horarios de baja concurrencia (por ejemplo el fin de semana) y realizar incrementales dentro del ciclo de respaldos definidos.

El problema reportado en SAP R3 era provocado debido que el administrador de otro sistema enviaba la ejecución de una sincronización completa cada vez que realizaba un respaldo. Al dejar definido el procedimiento de sincronización de discos el sistema trabajo adecuadamente.

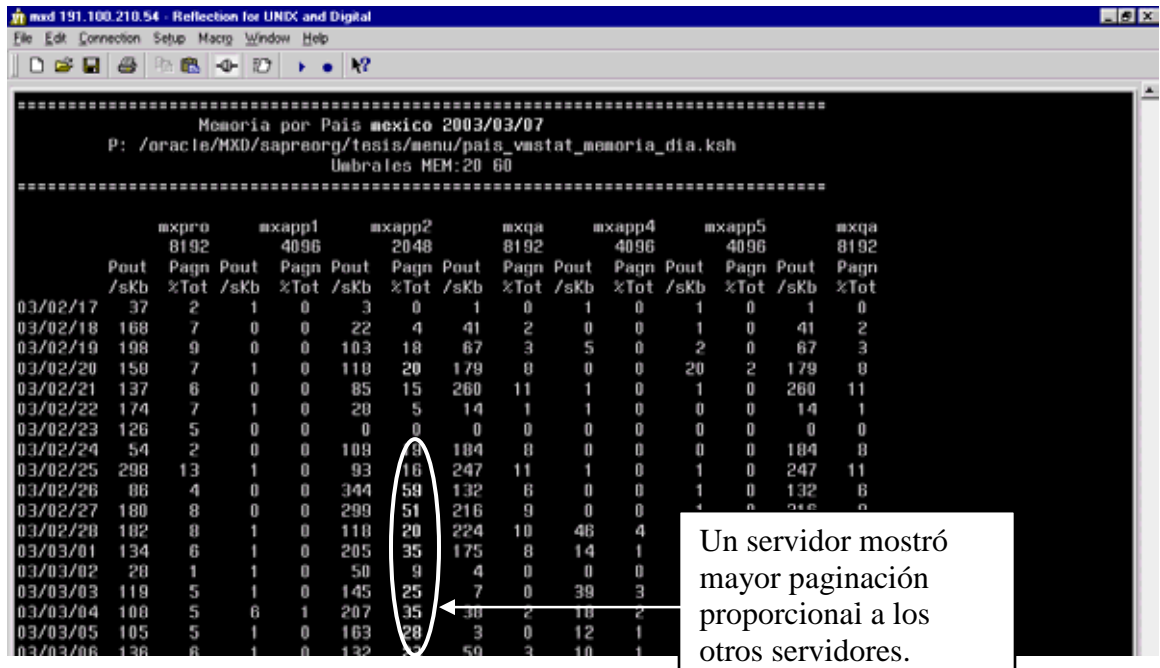
6.3 Redistribución de carga de trabajo para evitar paginación.

El siguiente problema se presentó en un ambiente productivo con 5 servidores de aplicación, se instaló el SMDS con objeto de validar la capacidad de los servidores.

The screenshot shows a terminal window titled "mxid 191.100.210.54 - Reflection for UNIX and Digital". The terminal output displays CPU usage statistics for various applications (mxpro, mxapp1, mxapp2, mxqa, mxapp4, mxapp5) over a period from 03/01/26 to 03/02/26. The statistics are presented in a table format with columns for application name and sub-columns for 'r' and 'b' values. The data shows varying levels of activity across the different applications over time.

Date	mxpro		mxapp1		mxapp2		mxqa		mxapp4		mxapp5		mxqa					
	r	b	r	b	r	b	r	b	r	b	r	b	r	b				
03/01/26	0	0	95	0	0	97	0	0	99	0	0	99	0	0	99	0	0	92
03/01/27	0	0	71	0	0	92	0	0	89	0	0	92	0	0	93	0	0	89
03/01/28	0	0	58	0	0	91	0	0	90	0	0	88	0	0	88	0	0	88
03/01/29	0	0	49	0	0	90	0	0	90	0	0	85	0	0	87	0	0	70
03/01/30	0	0	53	0	0	91	0	0	89	0	0	86	0	0	87	0	0	86
03/01/31	0	0	45	0	0	86	0	0	89	0	0	89	0	0	88	0	0	87
03/02/01	0	0	66	0	0	92	0	0	85	0	0	90	0	0	95	0	0	95
03/02/02	0	0	81	0	0	99	0	0	98	0	0	95	0	0	97	0	0	99
03/02/03	0	0	60	0	0	88	0	0	87	0	0	93	0	0	84	0	0	88
03/02/04	0	0	51	0	0	85	0	0	91	0	0	89	0	0	89	0	0	89
03/02/05	0	0	46	0	0	92	0	0	95	0	0	89	0	0	87	0	0	97
03/02/06	0	0	45	0	0	90	0	0	90	0	0	88	0	0	91	0	0	90
03/02/07	0	0	37	0	0	98	0	0	96	0	0	96	0	0	98	0	0	98
03/02/14	0	0	77	0	0	96	0	0	91	0	0	86	0	0	95	0	0	96
03/02/15	0	0	77	0	0	96	0	0	93	0	0	94	0	0	94	0	0	97
03/02/16	0	0	87	0	0	99	0	0	97	0	0	94	0	0	98	0	0	99
03/02/17	0	0	71	0	0	88	0	0	82	0	0	88	0	0	92	0	0	93
03/02/18	0	0	63	0	0	94	0	0	95	0	0	89	0	0	82	0	0	90
03/02/19	0	0	60	0	0	89	0	0	76	0	0	85	0	0	85	0	0	92
03/02/20	0	0	57	0	0	91	0	0	81	0	0	85	0	0	92	0	0	89
03/02/21	0	0	60	0	0	92	0	0	87	0	0	77	0	0	93	0	0	91
03/02/22	0	0	73	0	0	96	0	0	94	0	0	93	0	0	96	0	0	93
03/02/23	0	0	85	0	0	99	0	0	99	0	0	67	0	0	98	0	0	100
03/02/24	0	0	60	0	0	91	0	0	83	0	0	73	0	0	82	0	0	91
03/02/25	0	0	51	0	0	90	0	0	87	0	0	73	0	0	83	0	0	91
03/02/26	0	0	58	0	0	93	0	0	80	0	0	86	0	0	93	0	0	91

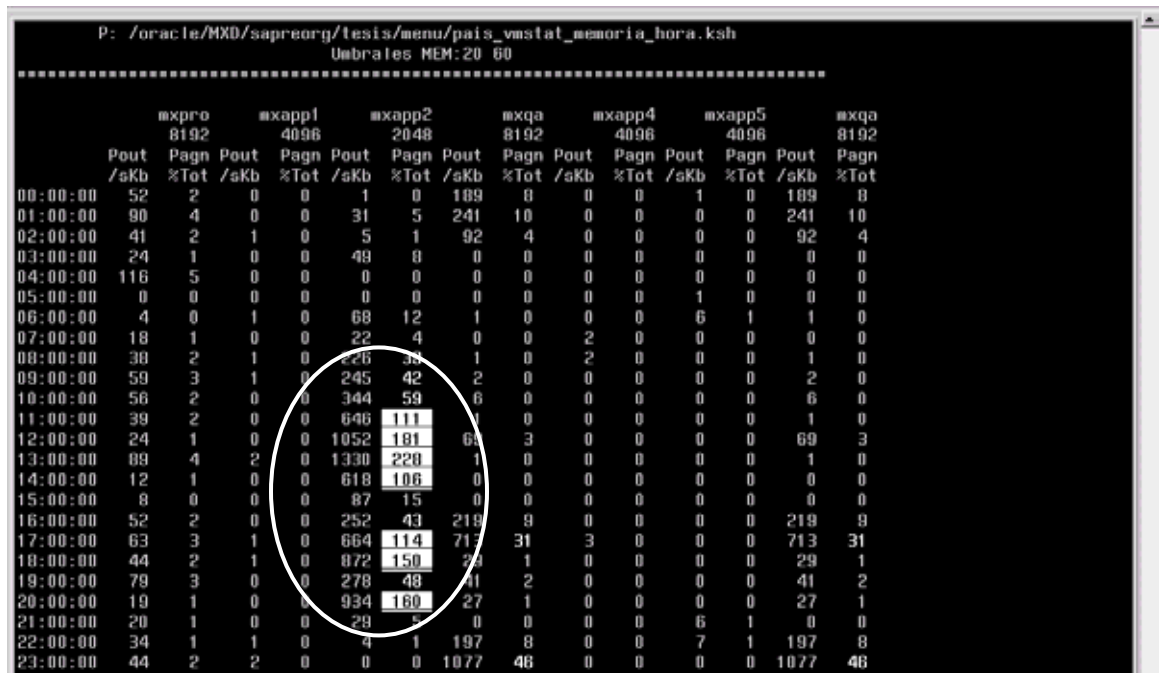
El monitor de CPU e IO mostró que el rendimiento en general fue adecuado en los sistemas.



Memoria por País mexico 2003/03/07
P: /oracle/MXD/sapreorg/tesis/menu/pais_vmstat_memoria_dia.ksh
Umbrates MEM:20 60

	mxpro 8192		mxapp1 4096		mxapp2 2048		mxqa 8192		mxapp4 4096		mxapp5 4096		mxqa 8192	
	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn
	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot
03/02/17	37	2	1	0	3	0	1	0	1	0	1	0	1	0
03/02/18	168	7	0	0	22	4	41	2	0	0	1	0	41	2
03/02/19	198	9	0	0	103	18	67	3	5	0	2	0	67	3
03/02/20	158	7	1	0	118	20	179	8	0	0	20	2	179	8
03/02/21	137	6	0	0	85	15	260	11	1	0	1	0	260	11
03/02/22	174	7	1	0	20	5	14	1	1	0	0	0	14	1
03/02/23	126	5	0	0	0	0	0	0	0	0	0	0	0	0
03/02/24	54	2	0	0	109	9	184	8	0	0	0	0	184	8
03/02/25	298	13	1	0	93	16	247	11	1	0	1	0	247	11
03/02/26	86	4	0	0	344	59	132	6	0	0	1	0	132	6
03/02/27	180	8	0	0	299	51	216	9	0	0	1	0	216	9
03/02/28	182	8	1	0	118	20	224	10	46	4	1	0	224	10
03/03/01	134	6	1	0	205	35	175	8	14	1	1	0	175	8
03/03/02	28	1	1	0	50	9	4	0	0	0	0	0	4	0
03/03/03	119	5	1	0	145	25	7	0	39	3	1	0	7	0
03/03/04	108	5	6	1	207	35	39	2	18	2	1	0	39	2
03/03/05	105	5	1	0	163	28	3	0	12	1	1	0	3	0
03/03/06	136	6	1	0	132	23	50	3	10	1	1	0	50	3

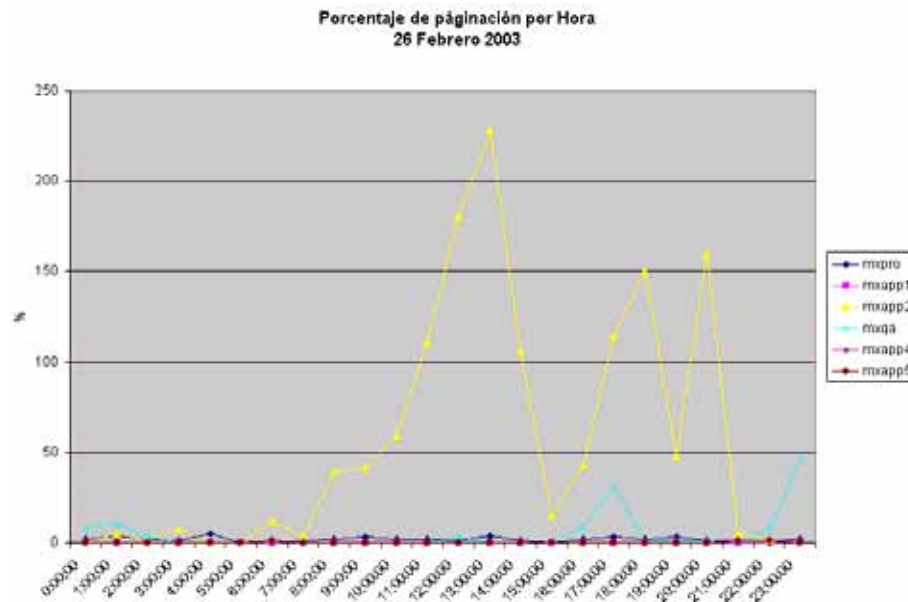
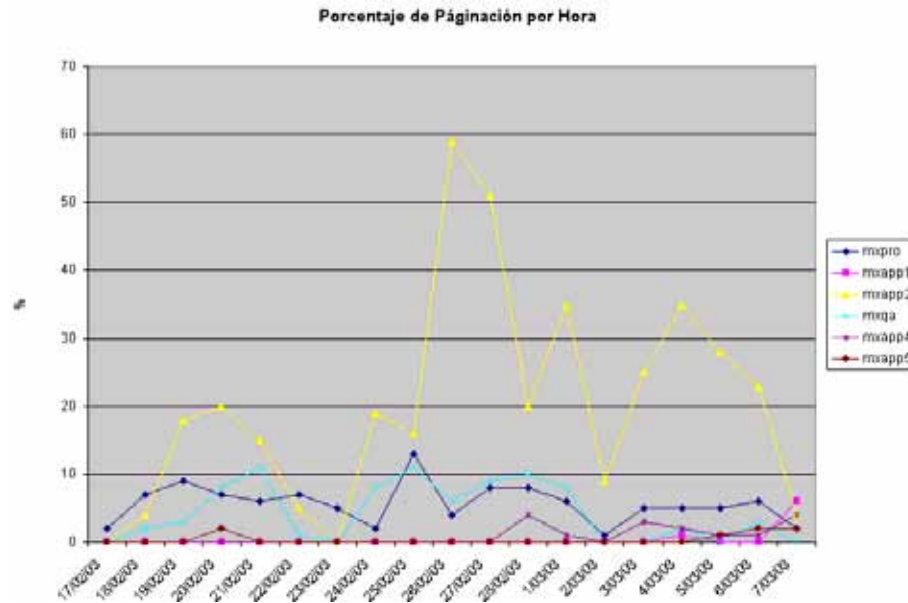
El monitor de Memoria mostró paginación en uno de los servidores. El análisis a detalle de este día muestra problemas de paginación en uno de los servidores y momentos en el día en donde la memoria se encuentra completamente utilizada.



P: /oracle/MXD/sapreorg/tesis/menu/pais_vmstat_memoria_hora.ksh
Umbrates MEM:20 60

	mxpro 8192		mxapp1 4096		mxapp2 2048		mxqa 8192		mxapp4 4096		mxapp5 4096		mxqa 8192	
	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn	Pout	Pagn
	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot	/sKb	%Tot
00:00:00	52	2	0	0	1	0	189	8	0	0	1	0	189	8
01:00:00	90	4	0	0	31	5	241	10	0	0	0	0	241	10
02:00:00	41	2	1	0	5	1	92	4	0	0	0	0	92	4
03:00:00	24	1	0	0	49	8	0	0	0	0	0	0	0	0
04:00:00	116	5	0	0	0	0	0	0	0	0	0	0	0	0
05:00:00	0	0	0	0	0	0	0	0	0	0	1	0	0	0
06:00:00	4	0	1	0	68	12	1	0	0	0	6	1	1	0
07:00:00	18	1	0	0	22	4	0	0	2	0	0	0	0	0
08:00:00	38	2	1	0	226	38	1	0	2	0	0	0	1	0
09:00:00	59	3	1	0	245	42	2	0	0	0	0	0	2	0
10:00:00	56	2	0	0	344	59	6	0	0	0	0	0	6	0
11:00:00	39	2	0	0	646	111	1	0	0	0	0	0	1	0
12:00:00	24	1	0	0	1052	191	6	0	0	0	0	0	6	0
13:00:00	88	4	2	0	1330	228	1	0	0	0	0	0	1	0
14:00:00	12	1	0	0	618	106	0	0	0	0	0	0	0	0
15:00:00	8	0	0	0	87	15	0	0	0	0	0	0	0	0
16:00:00	52	2	0	0	252	43	219	9	0	0	0	0	219	9
17:00:00	63	3	1	0	664	114	71	31	3	0	0	0	713	31
18:00:00	44	2	1	0	872	150	29	1	0	0	0	0	29	1
19:00:00	79	3	0	0	278	48	41	2	0	0	0	0	41	2
20:00:00	19	1	0	0	934	160	27	1	0	0	0	0	27	1
21:00:00	20	1	0	0	29	5	0	0	0	0	6	1	0	0
22:00:00	34	1	1	0	4	1	197	8	0	0	7	1	197	8
23:00:00	44	2	2	0	0	0	1077	48	0	0	0	0	1077	48

Esta desproporción es visualizada plenamente en la siguiente gráfica.



En este caso se encontró que el servidor el servidor con el problema cuenta con solo 2 GB de memoria física cuando los restantes cuentan con 4 y 8 GB. Con esta limitante es necesario considerar de acuerdo al mapa de decisión de afinación en memoria.

Si existe capacidad en otros servidores mover procesos hacia otros servidores.

Lo que se procedió a realizar es limitar el número de usuarios que trabajen a este servidor limitando el balanceo de cargas de SAP (smlg).

Después de aplicar esta modificación el servidor dejó de paginar memoria solucionándose el problema.

6.4 Deficiencias en el enlace.

Los problemas de enlace son en la práctica muy fácil de identificar dado que normalmente afectan de manera asilada a un grupo de usuarios.

En estos caso las herramientas de SAP son muy eficientes para diagnosticarlos ejecutando una prueba de PING hacia los clientes del sistema.

El siguiente caso fue ejecutado para un grupo de usuarios que reportaban lentitud extrema cuando en otras localidades se encontraba trabajando adecuadamente.

Servername	Server-IP	Min ms	Avg ms	Max ms	Loss %
Omar-Mendezsh	148.249.103.145	563	565	573	10
adelarosa	172.26.26.90	10	10	12	0
alejandro-aguilar	148.249.103.174	563	565	570	0
cpgarcia64	148.249.103.127	563	567	571	0
		10	427	573	

De acuerdo al mapa de solución al ejecutar un PING de 4Kb en 10 ciclos se debe observar tiempos de respuesta menores a 150 ms. y sin perdida de paquetes.

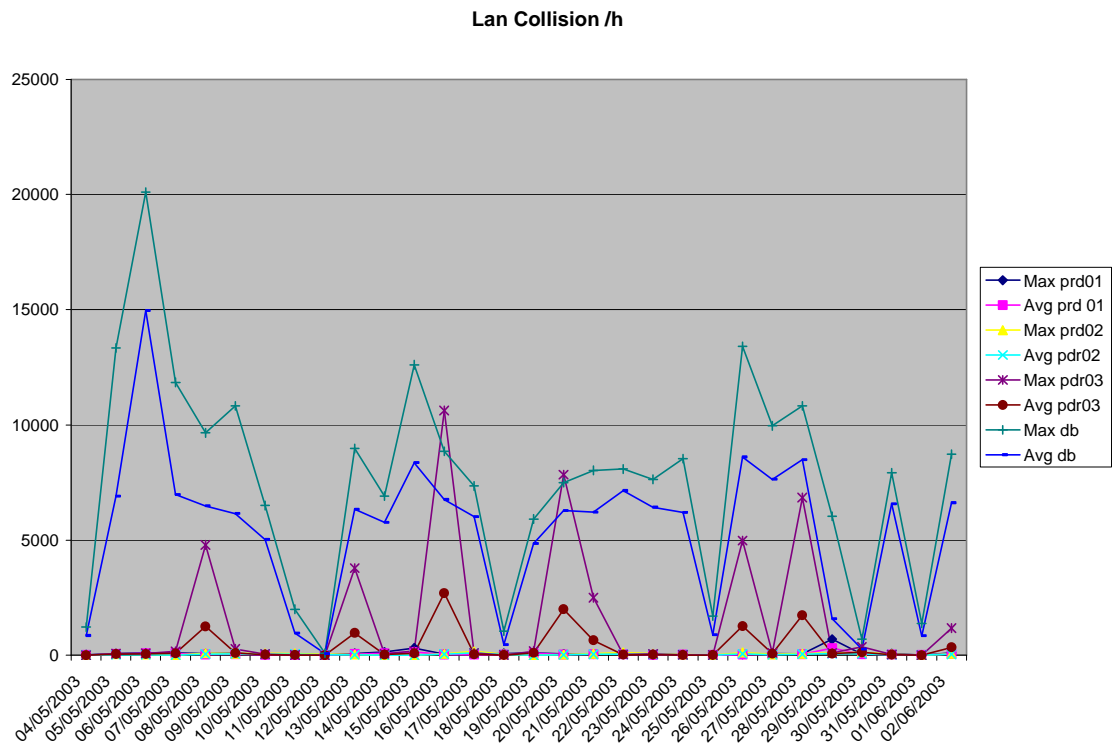
El resultado del ping sobre los usuarios problema mostró tiempos superiores a los 500 ms. muy por encima del umbral deseado cuándo en otros localidades mostró 10 ms.

En este caso se requirió ampliar el ancho de banda del enlace para solucionar el problema debido que se encontraba saturado.

6.5 Problemas en la red de área local (LAN)

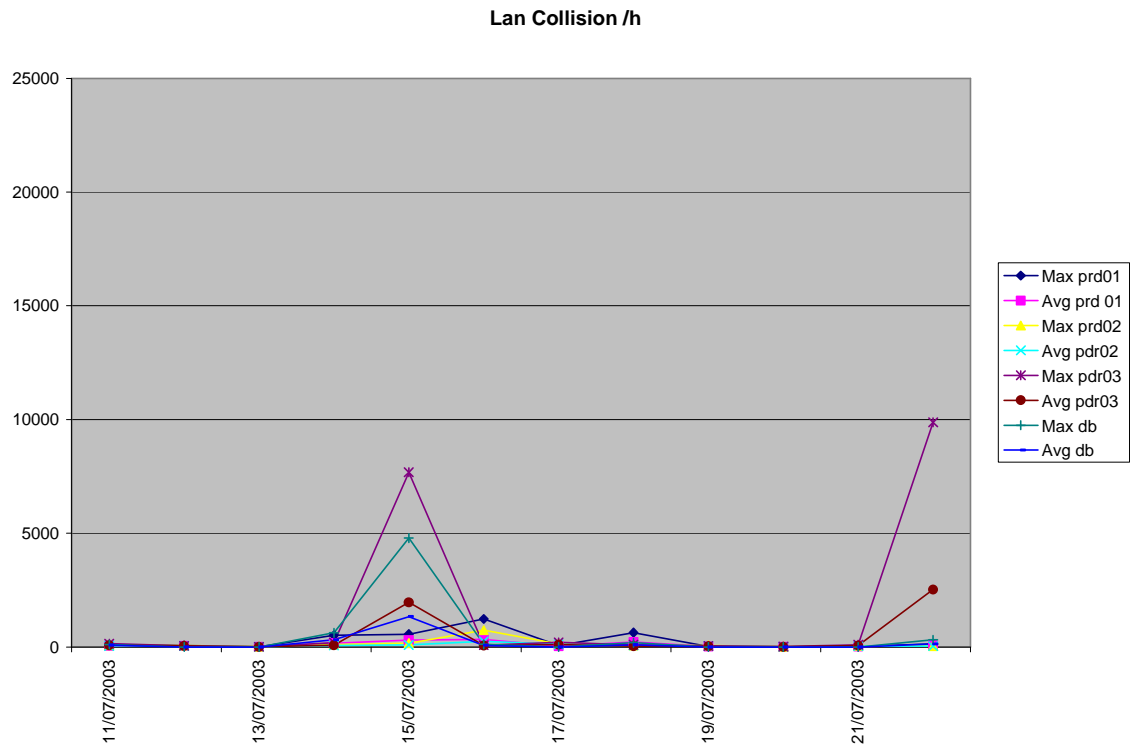
La infraestructura de comunicaciones tiene un papel primordial en el sistema, de acuerdo al mapa de decisión una LAN eficiente debe tener un mínimo de colisiones y no existir errores en la transmisión de paquetes.

El siguiente caso fue identificado en un cliente con problemas en su nivel de servicio de la aplicación. Entre los problemas identificados se encontró un nivel altísimo de colisiones (ST06->Detail Analisis Menu->Performance Database). La gráfica siguiente muestra los datos colectados.



Lo primero que se revisó de acuerdo al mapa de decisión fue la configuración de la interfaz de red y del esquema de conectividad del sistema. Todo se encontró correctamente configurado e instalado.

Una vez verificado la infraestructura del servidor se solicitó el apoyo del área de administración de red, después del análisis de ellos se identificó que existía tráfico excesivo en el sistema por un nuevo esquema de telefonía IP sobre voz. Esta problemática reforzó la solicitud de regresar el esquema anterior de telefonía digital controlándose parte del problema.



6.6 Problemas por inadecuada configuración en disco

El monitor de disco permite tener una visión global del sistema y poder determinar la eficiencia del rendimiento obtenido con una cierta configuración.

El siguiente caso corresponde a un análisis de configuración de un sistema de almacenamiento. El sistema analizado comenzó a presentar una baja general del rendimiento después de realizar una expansión de capacidad del sistema.

Los sistema SMDSA fue instalado antes y después de esta expansión generándose los siguientes datos del análisis de de rendimiento por volúmenes de discos, a continuación se muestra el rendimiento de los discos en un día operacional.

```

DISCO POR GRUPO DE IC mexico 2003/02/01
P: /oracle/PROD/sapreorg/tesis/menu/pais_iostat_grupo_hist.ksh
Umbralas SVCT:15 20 PCTB:14 18

=====
ind Server          vg      scvt  %B  lrps  rps  kups  wps
-----
1  mxpro             7,11   33  38   912  33   2    0
2  mxpro             ora,9, reorg, mnt  22  28   607  20   1    0
3  mxpro             sa, sys  15   0    39   0   16   1
4  mxpro             sa, root 15  12    16   2  170  20
5  mxpro             1,3,5,10 15  14   337  11   8    0
6  mxpro             2,4,6,8, arch 15  11   229   9   10   1
7  mxpro             mirr, sapfs 8   3    23   1   55   3
=====
Sig Fecha  Ant Fecha Detalle x Disco-Hora  Horas  Dias  Otro Pais  <return> Regresa

```

Se aprecia la carga excesiva en ciertos discos tanto en porcentaje de utilización como en lecturas realizadas. El día mostrado corresponde a un día de cierre contable.

Para tener una visión completa se genero un listado del rendimiento del sistema en por día:

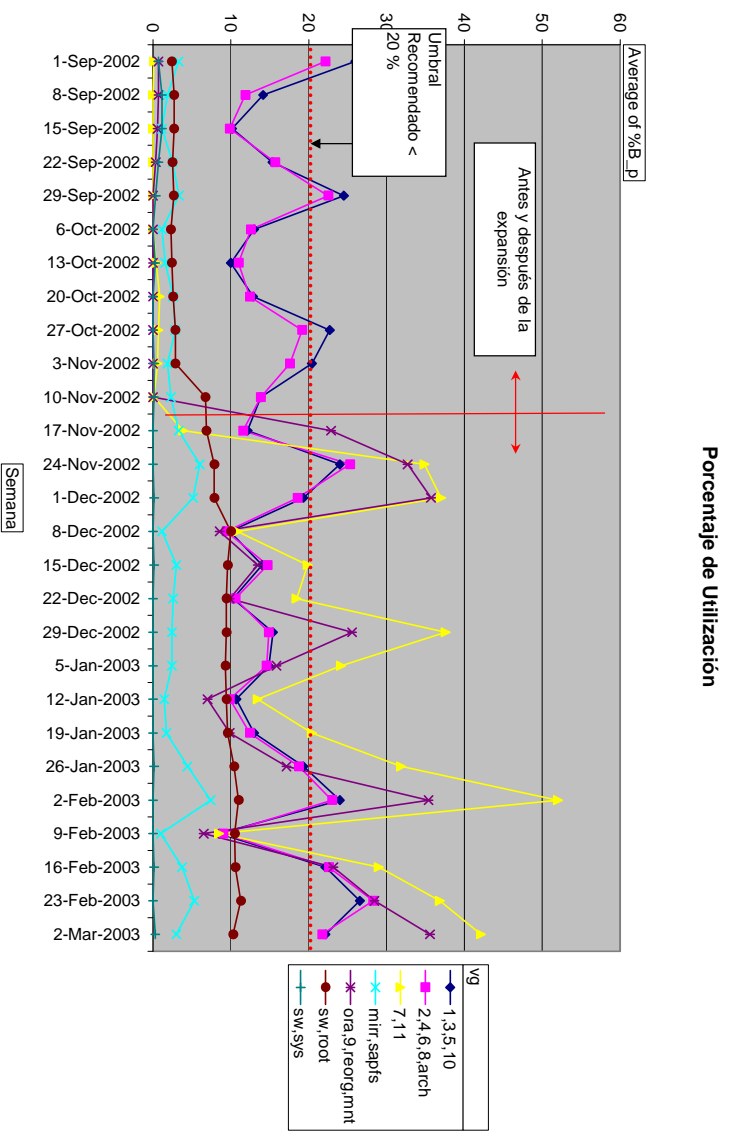
```

=====
DISCO historico de las 06:00 a las 20:00 Max y Min mexico 2003/00/10
P: /oracle/MXD/sapreorg/tesis/menu/pais_lostat_historia_old.ksh
Uabrales SVCT:15 20 PCTB:14 18
=====

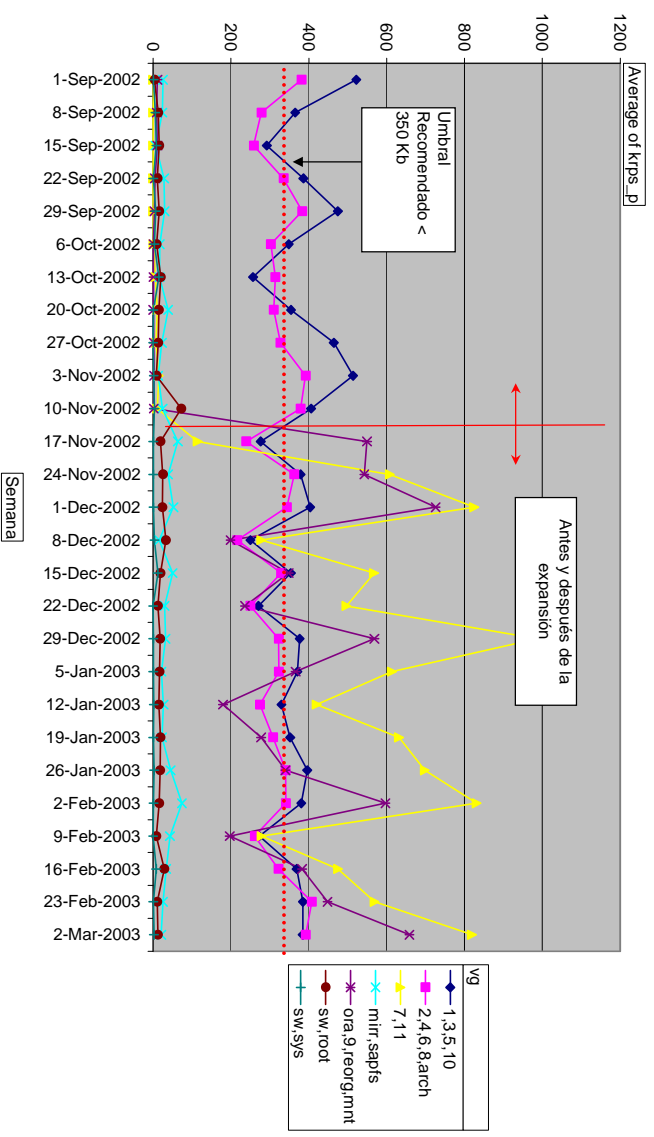
```

ddmmyy	Server	vg	scot_p	scot_M	%B_p	%B_M	krps_p	krps_M	rps_p	rps_M
20030308	mxpro	1,3,5,10	9	30	2	14	81	558	2	12
20030309	mxpro	2,4,6,8,arch	8	15	1	8	73	535	1	8
20030309	mxpro	7,11	5	14	1	5	63	602	1	8
20030309	mxpro	airr,sapfs	3	7	0	0	1	2	0	0
20030309	mxpro	ora,9,reorg,ant	4	13	1	10	73	1108	1	17
20030309	mxpro	su,root	8	12	8	11	1	3	0	0
20030309	mxpro	su,sys	12	17	0	1	0	0	0	0
20030310	mxpro	1,3,5,10	20	48	19	46	334	555	13	23
20030310	mxpro	2,4,6,8,arch	19	38	21	42	375	680	16	30
20030310	mxpro	7,11	25	46	42	70	845	1602	30	58
20030310	mxpro	airr,sapfs	8	29	2	7	14	1	1	2
20030310	mxpro	ora,9,reorg,ant	23	47	33	56	661	1073	23	38
20030310	mxpro	su,root	18	22	18	14	0	18	1	2
20030310	mxpro	su,sys	14	18	0	1	0	1	0	0
20030311	mxpro	1,3,5,10	17	27	20	40	427	661	17	27
20030311	mxpro	2,4,6,8,arch	15	28	23	44	520	766	23	35
20030311	mxpro	7,11	22	39	43	78	999	1572	37	66
20030311	mxpro	airr,sapfs	8	14	3	15	34	379	2	8
20030311	mxpro	ora,9,reorg,ant	20	38	32	59	718	1112	25	43
20030311	mxpro	su,root	18	27	10	16	52	631	2	15
20030311	mxpro	su,sys	14	18	0	1	0	5	0	0
20030312	mxpro	1,3,5,10	15	28	17	34	376	818	14	24
20030312	mxpro	2,4,6,8,arch	16	89	17	35	379	706	18	30
20030312	mxpro	7,11	18	27	18	42	382	921	14	34
20030312	mxpro	airr,sapfs	6	15	2	13	26	310	1	7
20030312	mxpro	ora,9,reorg,ant	13	24	12	34	297	825	10	28
20030312	mxpro	su,root	18	54	10	14	9	20	1	5
20030312	mxpro	su,sys	13	18	0	1	0	1	0	0

La pantalla anterior solo es un extracto de los datos colectados del 1 de septiembre del 2002 al 15 de Marzo del 2003. La graficación de estos datos permite apreciar el comportamiento de estos discos antes y después de la ampliación.



Kb leídos por Segundo



Las gráficas permiten observar claramente que el inicio del problema coincide con la ampliación de los discos, al revisar la configuración de los discos definidos se determinó que estos habían sido configurados de manera incorrecta ya que los volúmenes creados no eran equitativos:

Volumen	Discos
Datos1	21
Indices1	10
Indices2	4
Redologs	2
Datos2	5
Swap1	2
Swap2	2

Esta configuración tuvo que aplicarse debido a la ampliación consistió en 9 discos los cuales no podían agregarse al esquema original.

Otro problema encontrado fue que los datafiles de la base de datos eran de 15 a 20 GB por lo que requerían volúmenes muy grandes para alojarlos. Esto no se considera adecuado ya que uno de ellos puede contener hasta el 40% de operaciones de lectura del sistema imposibilitando utilizar todo el potencial de discos disponibles.

Por último, aún con la ampliación generada no era posible soportar el crecimiento del sistema por más de 8 meses.

La recomendación generada al respecto fue:

Reconstruir los volúmenes del sistema.

Reconstruir la BD en datafiles de entre 2 GB y 4GB.

Esta solución requería parar por completo la operación del sistema por 2 o 3 días y la adquisición de un nuevo sistema de almacenamiento por lo que se tuvo que planear la actividad por 6 meses.

Una vez autorizado esta actividad de afinación se obtuvo un nuevo sistema de almacenamiento con mayor cantidad de discos y se sugirió el siguiente esquema de distribución de volúmenes:

Volumen	Discos
lv1_datos	7
lv2_datos	7
lv3_indices	7
lv4_indices	7
lv5_datos	7
lv6_datos	7
lv7_indices	7
lv8_datos	7
rootdg	4

El resultado de ejecutar el cambio fue totalmente exitoso como se puede apreciar en los siguientes monitores:

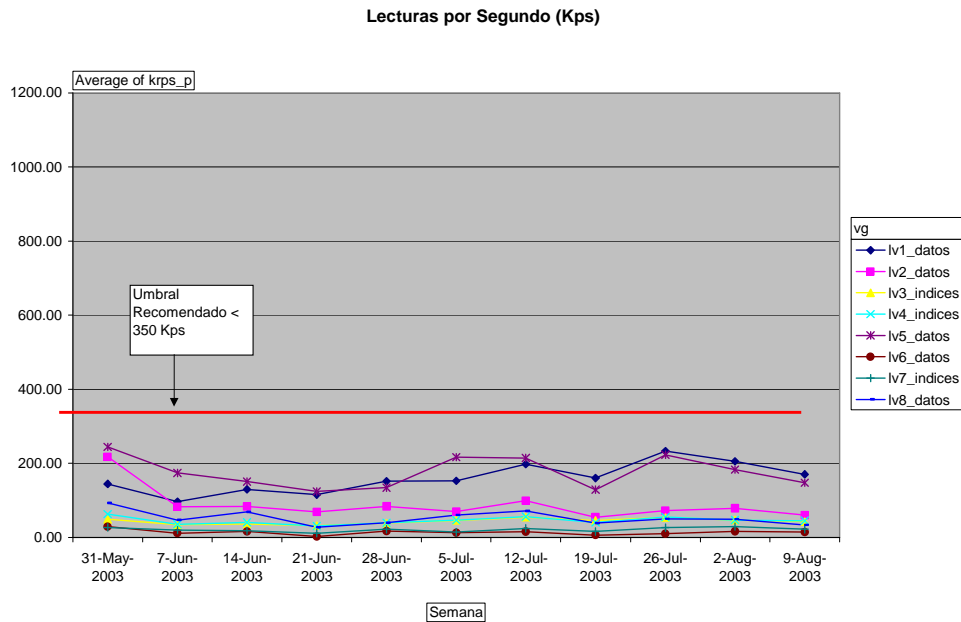
```

=====
DISCO historico de las 06:00 a las 20:00 Max y Min mexico 2003/08/18
P: /oracle/MXD/sapreorg/tesis/menu/pais_iostat_historia_old.ksh
Umbral: SVCI:15 20 PCTB:14 18
=====

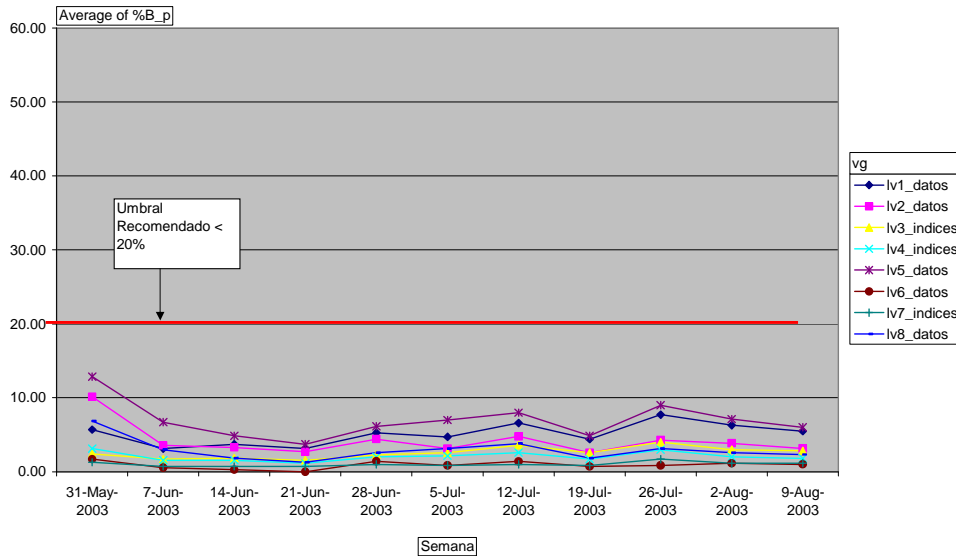
```

ddmmyy	Server	vg	scvt_p	scvt_M	%B_p	%B_M	krps_p	krps_M	rps_p	rps_M
20030704	mxpro	export	9	13	0	0	0	0	0	0
20030704	mxpro	lv1_datos	12	101	5	16	107	405	5	19
20030704	mxpro	lv2_datos	11	42	4	13	76	351	4	15
20030704	mxpro	lv3_indices	10	77	3	9	38	131	2	8
20030704	mxpro	lv4_indices	10	70	2	9	36	98	2	5
20030704	mxpro	lv5_datos	12	30	8	25	155	519	7	22
20030704	mxpro	lv6_datos	11	50	2	13	50	454	2	9
20030704	mxpro	lv7_indices	12	231	1	4	20	115	1	6
20030704	mxpro	lv8_datos	13	92	3	7	49	237	3	10
20030704	mxpro	rootdg	12	21	5	11	1	9	0	1
20030705	mxpro	export	9	13	0	0	0	0	0	0
20030705	mxpro	lv1_datos	8	15	2	7	58	250	2	11
20030705	mxpro	lv2_datos	7	16	1	4	20	97	1	4
20030705	mxpro	lv3_indices	8	32	1	7	22	76	1	3
20030705	mxpro	lv4_indices	6	18	1	2	21	77	1	2
20030705	mxpro	lv5_datos	8	18	3	13	141	765	5	26
20030705	mxpro	lv6_datos	7	27	0	1	4	18	0	2
20030705	mxpro	lv7_indices	7	16	0	1	4	13	0	1
20030705	mxpro	lv8_datos	8	17	1	4	17	57	1	4
20030705	mxpro	rootdg	12	21	4	10	0	5	0	1
20030706	mxpro	export	9	13	0	0	0	0	0	0
20030706	mxpro	lv1_datos	11	723	0	6	20	166	1	6
20030706	mxpro	lv2_datos	3	12	0	1	1	9	0	1

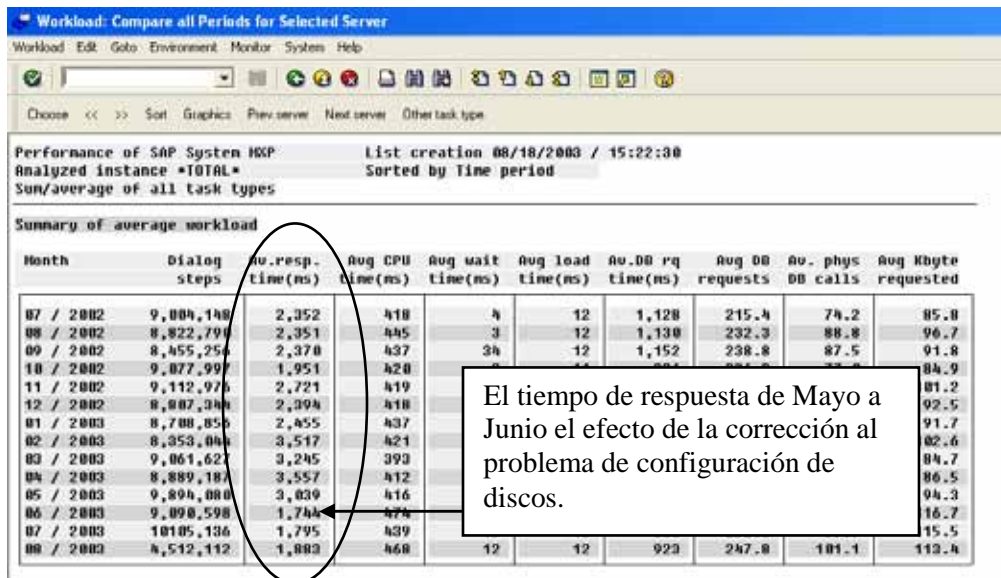
De manera gráfica el resultado se muestra evidente:



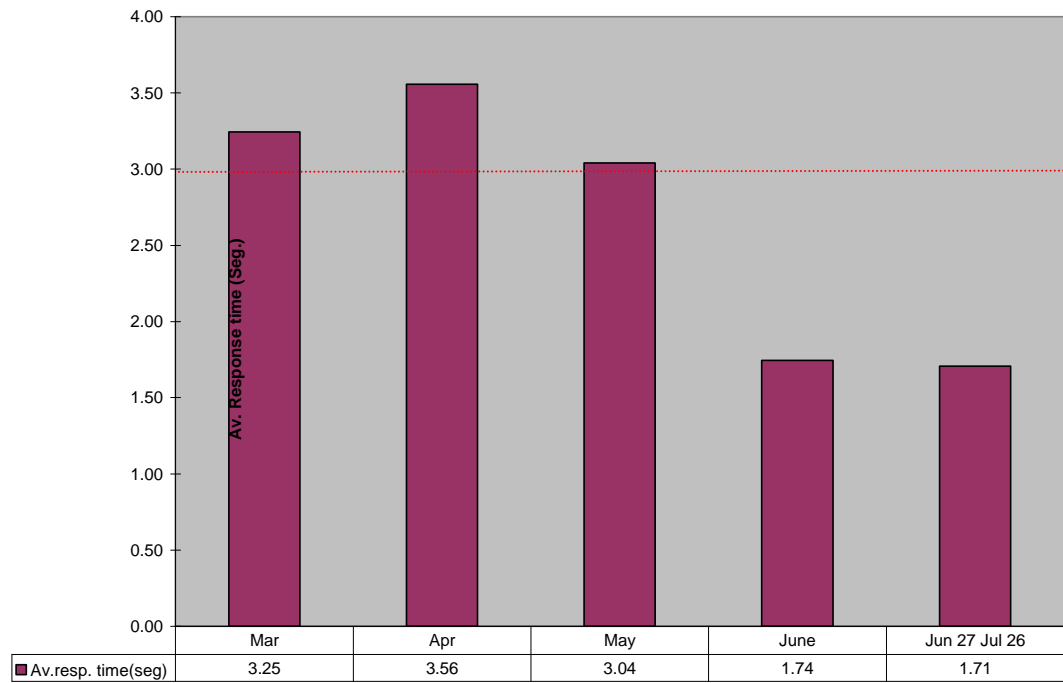
Procentaje de Utilización



Esta mejora en el desempeño se reflejo en el tiempo total de respuesta del sistema como se muestra a continuación.



Average Response Time



Conclusiones

El rendimiento adecuado de los servidores de aplicación en un ambiente SAP R3 es una pieza fundamental para el éxito de un proyecto de implantación y garantizar la operabilidad del sistema una vez que este se encuentra productivo. Cuando el rendimiento no cumple los niveles de servicio pactados el administrador debe tener elementos suficientes para dictaminar las acciones necesarias para alcanzar el objetivo de rendimiento. La herramienta propuesta y los procedimientos documentados permitirá tener estos elementos con las siguientes ventajas:

- **Facilidad para obtener muestras representativas de desempeño:** el sistema tiene la flexibilidad de configurarse para automatizar la realización del muestreo en intervalos constantes sobre una gran cantidad de servidores simultáneamente. Adicionalmente, debido a su configuración centralizada, permite que el mantenimiento sea muy fácil de realizar.
- **Objetividad en el análisis realizado:** se elimina la necesidad de utilizar un conjunto de comandos amplios y el análisis de varias columnas para determinar el rendimiento de un sistema que pueden perder al administrador en el enfoque del análisis, el SMDSA muestra de manera puntual las métricas necesarias para determinar si el CPU, la memoria o los discos se encuentran saturados con el mínimo de pantallas y datos.
- **Visibilidad completa del rendimiento del sistema:** la observación global del desempeño de todos los servidores permite analizar y detectar cualquier anomalía de manera inmediata.
- **Agilidad en el diagnóstico:** el tiempo requerido para realizar un dictamen del rendimiento del sistema es mínimo una vez que se han colectado una muestra amplia de datos.
- **Acciones de afinación puntuales:** cuando se ha detectado alguna falla en un componente la posibilidad de acciones de corrección son limitadas y orientadas de manera específica. Con esto el administrador de evitar divagar en suposiciones erróneas.
- **Entender el comportamiento del desempeño del sistema:** el sistema puede ser utilizado como una herramienta didáctica para que los administradores puedan ser inducidos en las prácticas de medición del rendimiento del sistema.

Aplicación de conocimientos adquiridos en la carrera en el presente trabajo

Los conocimientos adquiridos durante la carrera han sido fundamentales en mi trabajo profesional y la base para mi especialización en el mundo de consultoría SAP R3/Oracle/UNIX.

Esto es muy importante de subrayar, la preparación es la base y esta debe complementarse con estudios posteriores de acuerdo a la tecnología utilizada en el ámbito profesional.

De manera específica el presente trabajo utilice conocimiento de las siguientes materias:

- **Computación Básica, Datos y Estructuras de Almacenamiento y Programación Avanzada** – la habilidad de programar utilizando estructuras de datos, estructuras de control y basado en una metodología estructural fueron resultado de estas materias.
- **Diseño e implementación de sistemas y Análisis y Diseño de Sistemas** – el planear, diseñar, implementar y documentar el SMDSA esta basado en las metodologías estudiadas en los cursos correspondientes a estas materias. (Identificación de necesidades del usuario final, Diagramas de flujo, documentación de procesos, pruebas de validación, etc.).
- **Arquitectura de computadoras** – esta materia me ha permitido tener las bases para comprender las diferentes tecnologías existentes comercialmente así como identificar las oportunidades de mejora al implementar nuevas tecnologías en el ámbito de los sistemas SAP R3. (bus de comunicación entre dispositivos, arquitectura de sistemas, diferenciación de procesadores RISC y CISC, etc.)
- **Análisis de Algoritmos** – esta materia fue fundamental para justificar la necesidad de mejorar bloques de código y algoritmos utilizados de manera ineficiente en la extracción de datos.
- **Base de datos** – de manera indiscutible esta materia ha sido un pilar esencial en mi vida profesional, el tener conocimientos de diseño de base de datos y SQL me permitieron iniciar mi carrera en el mundo de las bases de datos y la posterior incursión al mundo SAP. El presente trabajo contiene nociones esenciales de base de datos aprendidas en el curso: diferenciación entre un diseño entidad relación en sistemas OLTP y diagrama estrella en Datawarehouse, creación de índices, sentencias SQL y DML, etc. Actualmente sigue siendo la materia que más he explotado profesionalmente.

- **Sistemas Operativos** – el conocimiento de los sistemas operativos desde su diseño hasta sus herramientas utilitarias me ha permitido manejar conceptos como memoria virtual, direccionamiento físico, arquitectura del sistema operativo, manejo de discos, programación en shell, etc.
- **Teleproceso** – el mundo de IT requiere conocimientos básico de telecomunicaciones: protocolos de comunicación, seguridad y encriptación, diferenciación entre WAN y LAN. Los temas abarcados me permitieron tener los elementos necesarios para trabajar con Ingenieros en Comunicaciones.
- **Estadística** – métodos simples de muestreo para la recolección de datos de desempeño.

Muchos profesores trataron de dar un valor agregado a sus clases y mantener actualizado sus cátedras con el mundo actual, de los cuáles debo resaltar:

- Durante el curso de métodos numéricos y estructuras de datos nos sugirieron utilizar C en lugar de FORTRAN, el manejar adecuadamente C me ha permitido aprender otros lenguajes de programación como Korn Shell y AWK, (los cuáles son dialectos de C).
- En 1993 el DSC de la escuela impartió un curso obligatorio a todos los alumnos de MAC sobre UNIX para utilizar la VAX para proyectos de programación. Hasta la fecha sigo utilizando estos conocimientos del entorno UNIX día a día.
- En el curso de base de datos el profesor nos permitió trabajar con software comercial como Informix, esto me permitió entrar al mundo de los administradores de base de datos profesionalmente.

Derivación de trabajos sobre el tema para un alumno de MAC

El tema del rendimiento de servidores es un muy extenso y puede derivar en trabajos de investigación relacionados al presente para un estudiante de MAC.

La metodología de dimensionamiento (sizing) de SAP abarca temas matemáticos de teoría de colas y cadenas de Markov para la evaluación del desempeño de aplicaciones de cómputo¹, una excelente referencia sobre las técnicas matemáticas y su aplicación en el análisis de desempeño en servidores de cómputo es el libro The Art of Computer Systems Performance Analysis de Raj Jain o el libro The Practical Performance Analyst, Neil Gunter.

¹ Sizing mySAP.com, SAP AG May 2001

Una mejora muy importante al presente trabajo es traducir el código de korn shell a ABAP (el lenguaje de las aplicaciones de SAP) y adaptar el sistema a explotar el colector de estadísticas de SAP saposcol. Esto permitirá una integración completa con el sistema y no solo se ejecutará en UNIX sino en cualquier sistema soportado por SAP.

Bibliografía

Thomas Scheneider, SAP R/3 Performance Optimization Editorial SYBEX ISBN 0-7821-2563-8

Juan Antonio Hernández Muñoz, SAP R/3 Handbook, 2nd Edition Editorial Osborne-McGraw Hill ISBN 0-07-135413-1

Adrian Cockcroft, Sun Performance and Tuning Editorial Prentice Hall/Sun Microsystems Press ISBN 0-13-095249-4

Jim Dinali, Bruce Ernst, Oracle 8i Performance Tuning Workshop Editorial Oracle Corp. (Training Course Material)

Solaris System Performance Management SA-400 Student Guide, Sun Microsystems, October 1999

Neil Gunther, The Practical Performance Analysis McGraw Hill 1997

Documentos electrónicos

SAP AG. Performance in 4.0/4.5/4.6: Parameter Recommendation. SAP Support Portal. Notes Search. (Restricted Support for Customers and Partners) [En línea] Walldrof, Alemania SAP AG 07.11.2002 07.07.2004 OSS Note 146289 Disponible <<https://websmp202.sap-ag.de/notes>> (requiere usuario y contraseña)

SAP AG. FAQ: Oracle performance. SAP Support Portal. Notes Search. (Restricted Support for Customers and Partners) [En línea] Walldrof, Alemania SAP AG 07.11.2002 07.07.2004 OSS Note 618868 Disponible <<https://websmp202.sap-ag.de/notes>> (requiere usuario y contraseña)

SAP AG. Oracle DB parameters for R/3 Release >= 4.x. SAP Support Portal. Notes Search. (Restricted Support for Customers and Partners) [En línea] Walldrof, Alemania SAP AG 07.11.2002 07.07.2004 OSS Note 124361 Disponible <<https://websmp202.sap-ag.de/notes>> (requiere usuario y contraseña)

SAP AG. FAQ: Oracle Wait Events. SAP Support Portal. Notes Search. (Restricted Support for Customers and Partners) [En línea] Walldrof, Alemania SAP AG 07.11.2002 07.07.2004 OSS Note 619188 Disponible <<https://websmp202.sap-ag.de/notes>> (requiere usuario y contraseña)

Oracle Corp. Oracle Server - Rdbms Database Performance Frequently Asked Questions. Metalink. Technical Note (Required user id) [En línea] Estados Unidos. Oracle Corp. Jul-6-2001 . 07.07.2004 . Document ID 146560.1 . Disponible <http://metalink.oracle.com/metalink/plsql/ml2_gui.startup> (requiere usuario y contraseña)

Oracle Corp. Oracle Server - Performance Tuning Approach for Oracle(8.1.6 - 9.2.0.5) on UNIX. Metalink. Technical Note (Required user id) [En línea] Estados Unidos. Oracle Corp. Jul-6-2001 . 07.07.2004 . Document ID 282806.1. Disponible <http://metalink.oracle.com/metalink/plsql/ml2_gui.startup> (requiere usuario y contraseña)

Oracle Corp. Oracle Server - Systemwide Tuning using STATSPACK Reports. Metalink. Technical Note (Required user id) [En línea] Estados Unidos. Oracle Corp. Jul-6-2001 . 07.07.2004 . Document ID 228913.1. Disponible <http://metalink.oracle.com/metalink/plsql/ml2_gui.startup> (requiere usuario y contraseña)

Oracle Corp. Oracle Server - Tuning I/O-related waits. Metalink. Technical Note (Required user id) [En línea] Estados Unidos. Oracle Corp. Jul-6-2001 . 07.07.2004 . Document ID 223117.1. Disponible <http://metalink.oracle.com/metalink/plsql/ml2_gui.startup> (requiere usuario y contraseña)

Sun Microsystems. Performance and Tuning on Solaris 2.6, 7 and 8. SunSolve. Support and Training [En línea] Estados Unidos Sun Microsystems 06.16.2004 07.07.2004 ID21622 Disponible < http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=finfodoc%2F21622&zone_32=Performance%20Unix >

Richard Mc Dougall, Triet Vo. Priority Paging - Frequently Asked Questions. SunSolve. Support and Training [En línea] Estados Unidos Sun Microsystems 01.08.1998 07.07.2004 Sun Performance Disponible < http://www.sun.com/sun-on-net/performance/priority_paging.html>

Adrian Cockcroft. How do disks really work?. UNIX Insider. [En línea] Estados Unidos. IT World.com 01.06.1996 07.07.2004 Storage Disponible <http://storage.itworld.com/4650/UIR960601perf/page_1.html>

Stephan Kuhlmann Oracle Performance Tuning Workshop [en línea]. SAP AG, 2002 [citado 7 Julio 2004]. Disponible a través de: SAP Support Academy, Walldrof, Alemania

SAP AG, Oracle SQL Cache Analysis [en línea]. SAP AG, 2001 [citado 7 Julio 2004]. Disponible a través de: SAP Support Academy, Walldrof, Alemania

Kevin Hartley. Database I/O Performance Analysis and Optimization . Walldrof, Alemania SAP AG [citado 7 Julio 2004]. Disponible a través de: SAP Tech Ed 2000

Apéndice 1 Archivos fuente del colector de estadísticas

1.A Programas de recolección de estadísticas

1.A.1 inicial.ksh

```
#!/usr/bin/ksh

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: inicial.ksh

# Objetivo:
# Iniciar el colector de estadísticas

# Archivo de parametrizacion:
# metricas.lst
#=====

HOME_PATH=/oracle/MXD/sapreorg/tesis

PROCESS_LOG=$HOME_PATH/process_list.lst

print "FECHA $(date '+%Y%m%d %H%M')" > $PROCESS_LOG
for i in $( cat $HOME_PATH/metricas.lst )
do
    metrica=$(echo $i | awk -F"|" '{print $1}')
    frecuencia=$(echo $i | awk -F"|" '{print $2}')
    tipo=$(echo $i | awk -F"|" '{print $3}')
    nohup $HOME_PATH/colector.ksh $metrica $frecuencia $tipo
$HOME_PATH>$HOME_PATH/$metrica.log 2>$HOME_PATH/$metrica.err &
    print "$!|$metrica|$frecuencia|$tipo" >> $PROCESS_LOG
    print "Proceso de coleccion $! $metrica a ejecutar cada
$frecuencia segs tipo $tipo"
done
```

1.A.2 colector.ksh

```
#!/usr/bin/ksh

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: colector.ksh

# Objetivo:
# Colector estadísticas en cada uno de los servidores definidos

# Archivo de parametrizacion:
#
#=====
```

```

# Funcion Utilitaria para detectar si la BD se encuentra abajo
function valida_bd
{
LOGBD=/tmp/$_valida_bd.log

sqlplus -S internal<<EOF>$_LOGBD
select * from v$instance;
EOF

grep "ORACLE not available" $_LOGBD>/dev/null 2>/dev/null
if [ $? -eq 0 ]
then
    let BD_STATUS=0
else
    let BD_STATUS=1
fi
export BD_STATUS

rm $_LOGBD
}
typeset -fx valida_bd

metrica=$1
frecuencia=$2
tipo=$3
HOME_PATH=$4
export HOME_PATH

while [ true ]
do
    fecha=$(date '+%Y%m%d %H%M%S')
    print "$HOME_PATH $fecha $metrica $tipo $frecuencia"

    valida_bd
    if [ $_BD_STATUS -eq 1 ]
    then
        $HOME_PATH/disparador.ksh $metrica $tipo
    else
        print "$fecha BD no disponible"
    fi

    sleep $frecuencia
done

```

1.A.3 disparador.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: disparador.ksh

# Objetivo:
# Para caada servidor definido en servers.lst
#     Crear el encabezado de cada script a ejecutar

```

```

#           - metrica
#           - servidor
#           - ip
#           - los intervalos para el comando SAR y VMSTAT
#           - fecha de ejecucion
#   Armar el nuevo script en base a la plantilla definida en
<metrica>.ksh y el encabezado
#   Copiarlo al sistema remoto y ejecutarlo
#   Cargarlo en la BD

# Archivo de parametrizacion:
# servers.lst
#=====

METRICA=$1
TIPO=$2

# Esta variable es definida durante la instalacion
#HOME_PATH=/oracle/MXD/sapreorg/tesis
#export HOME_PATH
print "$0 $HOME_PATH"

SAR_LENGTH=$( grep SAR_LENGTH $HOME_PATH/monitor.lst | awk '{print
$2}')
SAR_INTERVAL=$( grep SAR_INTERVAL $HOME_PATH/monitor.lst | awk '{print
$2}')
WTMP=$( grep WTMP $HOME_PATH/monitor.lst | awk '{print $2}'))

USUARIO=$(cat $HOME_PATH/user.lst | awk -F"|" '{print $1}')
CLAVE=$(cat $HOME_PATH/user.lst | awk -F"|" '{print $2}')
export USUARIO
export CLAVE

# Activa funciones de soporte
. $HOME_PATH/funciones.ksh

#valida_remote_shell

# metrica es el nombre del shell de monitoreo y de la tabla
# donde almacenar recursos

for i in $(cat $HOME_PATH/servers.lst)
do
    SERVERNAME=$(echo $i | awk -F"|" '{print $1}')
    SERVERIP=$(echo $i | awk -F"|" '{print $2}')
    USERNAME=$(echo $i | awk -F"|" '{print $3}')
    INSTANCE_TYPE=$(echo $i | awk -F"|" '{print $4}')

    if [ "$TIPO" = "so" ]
    then
        # Prepara las variables iniciales de todo script
        fecha_sistema
        VARFILE=/tmp/$$_${SERVERNAME}_${fecha}_.varfile
        print "#!/usr/bin/ksh" > $VARFILE
        print "metrica=$METRICA" >> $VARFILE
    fi
done

```

```

print "servername=$SERVERNAME" >> $VARFILE
print "serverip=$SERVERIP" >> $VARFILE
print "username=$USERNAME" >> $VARFILE
print "SAR_LENGTH=$SAR_LENGTH" >> $VARFILE
print "SAR_INTERVAL=$SAR_INTERVAL" >> $VARFILE
print "fecha=\$(date '+%Y/%m/%d %H:%M')" >> $VARFILE

# arma el script, lero var ambiente 2do el script
FILEEXE=$WTMP/$$_${SERVERNAME}_${METRICA}_${fecha}.exe
REMOTEEXE=$WTMP/$$_${SERVERNAME}_${METRICA}_${fecha}.exe
cat $VARFILE $HOME_PATH/$METRICA.ksh >> $FILEEXE

# lo copia y cambia permisos en el server remoto
rcp $FILEEXE $USERNAME@$SERVERIP:$REMOTEEXE
remsh -l $USERNAME $SERVERIP chmod 755 $REMOTEEXE

# ejecutarlo y guardarlo en un datfile de oracle
DATFILE=$WTMP/$$_${SERVERNAME}_${fecha}.dat
remsh -l $USERNAME $SERVERIP $REMOTEEXE > $DATFILE

# cargarlos a la tabla correspondiente
carga_datos $DATFILE $METRICA $SERVERNAME

remsh -l $USERNAME $SERVERIP rm $REMOTEEXE
rm $VARFILE
rm $FILEEXE
rm $DATFILE
else
    nada="nada"
fi

if [ "$TIPO" = "ora" -a "$INSTANCE_TYPE" = "c" ]
then
# Prepara las variables iniciales de todo script
fecha_sistema
VARFILE=/tmp/$$_${SERVERNAME}_${fecha}_.varfile
print "#!/usr/bin/ksh" > $VARFILE
print "metrica=$METRICA" >> $VARFILE
print "servername=$SERVERNAME" >> $VARFILE
print "serverip=$SERVERIP" >> $VARFILE
print "username=$USERNAME" >> $VARFILE
print "SAR_LENGTH=$SAR_LENGTH" >> $VARFILE
print "SAR_INTERVAL=$SAR_INTERVAL" >> $VARFILE
print "fecha=\$(date '+%Y/%m/%d %H:%M')" >> $VARFILE

# arma el script, lero var ambiente 2do el script
FILEEXE=$WTMP/$$_${SERVERNAME}_${METRICA}_${fecha}.exe
REMOTEEXE=$WTMP/$$_${SERVERNAME}_${METRICA}_${fecha}.exe
cat $VARFILE $HOME_PATH/$METRICA.ksh >> $FILEEXE

# lo copia y cambia permisos en el server remoto
rcp $FILEEXE $USERNAME@$SERVERIP:$REMOTEEXE
remsh -l $USERNAME $SERVERIP chmod 755 $REMOTEEXE

# ejecutarlo y guardarlo en un datfile de oracle
DATFILE=$WTMP/$$_${SERVERNAME}_${fecha}.dat

```

```

remsh -l $USERNAME $SERVERIP $REMOTEEXE > $DATFILE

# cargarlos a la tabla correspondiente
carga_datos $DATFILE $METRICA $SERVERNAME

remsh -l $USERNAME $SERVERIP rm $REMOTEEXE
rm $VARFILE
rm $FILEEXE
rm $DATFILE
else
    nada="nada"
fi
done

```

1.A.4 carga_datos (funcion korn shell incluida en el archivo funciones.ksh)

```

#
*****
*****
# funtion: crea_sqlldr_ctl
# crea el archivo de control para cargar los datos obtenidos del script
de monitoreo
# para su carga via SQL*Loader de Oracle
#
# tipo: secundario
# funcion_primaria: carga_datos
#
*****
*****
function crea_sqlldr_ctl
{

metrica_tabla=$1
control_file=$2
data_file=$3

tempo=/tmp/${$_sqlldr}.tmp
aux=/tmp/${$_sqlldr}.aux
sqlplus -S $USUARIO/$CLAVE<<EOF>>$tempo
desc $metrica_tabla;
exit;
EOF
exit;
EOF

# Quita los encabezados
awk '{ if ( i>=2 && $1 != "" ) print $0; i++; }' $tempo > $aux

let n_columnas=0
while read x
do
    let n_columnas=$n_columnas+1
done < $aux

print "load data" >> $control_file
print "infile '$data_file'" >> $control_file

```

```

print "append into table $metrica_tabla" >> $control_file
print "fields terminated '|'" >> $control_file
print "(" >> $control_file

let k=0
while read columna tipo
do
    let k=$k+1
    if [ $tipo = 'DATE' ]
    let k=$k+1
    if [ $tipo = 'DATE' ]
    then
        printf "$columna $tipo 'YYYY/MM/DD HH24:mi'" >>
$control_file
    else
        printf "$columna " >> $control_file
    fi

    if [ $k -eq $n_columnas ]
    then
        print " " >> $control_file
    else
        print "," >> $control_file
    fi
done < $aux

print ")" >> $control_file

rm $tempo
rm $aux
}

typeset -fx crea_sqlldr_ctl

#
*****
*****
# funcion: revisa_log
# Revisa si existio algun error en la carga de datos en caso de ser
afirmativo
# lo sumarizara en el archivo sqlldr.err
#
# tipo: secundario
# funcion_primaria: carga_datos
#
*****
*****

function revisa_log
{
let sucess=$( grep "successfully loaded" $1 | awk '{print $1}' )
let err=$( grep "not loaded due" $1 | awk '{print $1}' )

#depuracion en prueba.ksh
export sucess err

```

```

if [ $sucess -eq 0 -o $err -gt 0 ]
then
    print "FECHA $(date '+%Y%m%d')" >> $HOME_PATH/sqlldr.err
    print "FECHA $fecha" >> $HOME_PATH/sqlldr.err
    print '### LOG '>> $HOME_PATH/sqlldr.err
    cat $1 >> $HOME_PATH/sqlldr.err
    print '### CONTROLFILE '>> $HOME_PATH/sqlldr.err
    cat $2 >> $HOME_PATH/sqlldr.err
    print '### DATAFILE '>> $HOME_PATH/sqlldr.err
    cat $3 >> $HOME_PATH/sqlldr.err
    print "**** xxx *****" >>
$HOME_PATH/sqlldr.err
fi
}
typeset -fx revisa_log

#
*****
*****
# funcion: carga_datos
# carga el resultado del script de monitoreo a la BD Oracle via
SQL*Loader
#
# tipo: primario
#
*****
*****
function carga_datos
{
FILEDAT=$1
METRICA=$2
SERVERNAME=$3
CTL=$WTMP/$$_$_{SERVERNAME}_$_{fecha}.ctl
LOG=$WTMP/$$_$_{SERVERNAME}_$_{fecha}.log

#crea_sqlldr_ctl " " $CTL $FILEDAT
crea_sqlldr_ctl $METRICA $CTL $FILEDAT
sqlldr $USUARIO/$CLAVE control=$CTL log=$LOG >/dev/null 2>/dev/null
revisa_log $LOG $CTL $FILEDAT
revisa_log $LOG $CTL $FILEDAT

rm $LOG
rm $CTL
}
typeset -fx carga_datos



## 1.A.5 cpu.ksh


#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: cpu.ksh

# Objetivo:
# Monitorear el uso y estado del cpu en base al comando sar -u y sar -q
#

```

```

# Archivo de parametrizacion:
# monitor.lst -- contiene los rangos de ejecución del sar <opcion> <num
intervalos> <segs>
#
# Salida:
#   usr: % uso de cpu proceso de usuarios
#   sys: % " " sistema
#   wio: % de cpu en espera por IO
#   encolado: procesos en la cola de espera
#   porc_procenc: porcentaje de procesos encolados
#=====

tmp=/tmp/$_$_${metrica}_${server}.tmp
tmp1=/tmp/$_$_${metrica}_${server}_1.tmp

sar -u $SAR_LENGTH $SAR_INTERVAL | tail -1 > $tmp &
p1=$!
sar -q $SAR_LENGTH $SAR_INTERVAL | tail -1 > $tmp1 &
p2=$!

wait $p1
wait $p2

usr=$(cat $tmp | awk '{print $2}')
sys=$(cat $tmp | awk '{print $3}')
wio=$(cat $tmp | awk '{print $4}')
idle=$(cat $tmp | awk '{print $5}')
t_enc=$(cat $tmp1 | awk '{print substr($0,10,7)}' )
if [ -z "$t_enc" ]
then
    let encolado=0
else
    let encolado=$(cat $tmp1 | awk '{print $2}' )
fi
t_por=$(cat $tmp1 | awk '{print substr($0,18,7)}' )
if [ -z "$t_enc" ]
then
    let percent=0
else
    let percent=$(cat $tmp1 | awk '{print $3}' )
fi

print "$servername|$fecha|$usr|$wio|$sys|$idle|$encolado|$percent"

rm $tmp
rm $tmp1

```

1.A.6 vmstat.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: vmstat.ksh

# Objetivo:
# Obtener metricas de desempeño de:

```



```

# I/O:      b procesos bloqueado
# Memoria:  po kb paginacion por segundo
# CPU:      r ready queue
#           id porcentaje de disponibilidad de CPU
#           us porcentaje de uso de CPU por proceso de usuario
#           sy porcentaje de uso de CPU por proceso de sistema
(kernel UNIX)
#
# Archivo de parametrizacion:
#
# Salida:
# Todas las columnas de vmstat, las usuales hasta este momento se
documentaron
# en la sección anterior:
# vmstat
# procs      memory          page          disk          faults
cpu
# r b w  swap free re mf pi po fr de sr s0 s1 s6 s3  in  sy  cs
us sy id
#
#=====

```

```

vmstat 5 2 | tail -1 | /usr/xpg4/bin/awk -v server="$servername" -v
fec="$fecha" '{ print
server|"fec|" $1|" $2|" $3|" $4|" $5|" $6|" $7|" $8|" $9|" $10|" $11|"
$12|" $13|" $14|" $15|" $16|" $17|" $18|" $19|" $20|" $21|" $22|" }'

```

1.A.7 disk.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: disk.ksh

# Objetivo:
# Obtener metricas de desempeño de:
# Kr/s: Kb leídos por segundo. Idealmente no debe rebasar los
200Kb/s
# Svc_t: Tiempo de respuesta de los discos. Para discos EMC2 no
debe ser mayor de 20 ms.
# %b: Porcentaje de ocupación de los discos. Para discos EMC2 no
debe rebasar el 15%
#
#
# Archivo de parametrizacion:
#
# Salida:
# Todas las columnas de iostat -x, las usuales hasta este momento
se documentaron
# en la sección anterior:
# device  r/s w/s kr/s kw/s wait actv svc_t %w %b
#
#=====

```

```

tmp1=/tmp/$$tmp1
tmp2=/tmp/$$tmp2
tmp3=/tmp/$$tmp3

# discos
iostat -exp 60 2 | awk '{ if ( iteracion == 2 ) print $0; if ( $1 ==
"device" ) iteracion=iteracion+1; }' | grep -v "," | grep -v nfs >
$tmp1

# disk          char(20),
# rps           number,
# wps           number,
# krps          number,
# kwps          number,
# wait          number,
# actv          number,
# svc_t         number,
# pctw          number,
# pctb          number,
# errsw         number,
# errhw         number,
# errtrn        number,
# errtot        number

while read disco rps wps krps kwps wait actv svc_t pctw pctb errsw
errhw errtrn errtot
do
    print
"$servername|$fecha|$disco|$rps|$wps|$krps|$kwps|$wait|$actv|$svc_t|$pc
tw|$pctb|$errsw|$errhw|$errtrn|$errtot"
done < $tmp1

rm $tmp1

```

1.A.8 network.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: network.ksh

# Objetivo:
# Obtener metricas de desempeño de red
#
#
# Archivo de parametrizacion:
#
# Salida:
#     Todas las columnas de netstat -i
#
#=====

tmp1=/tmp/$$tmp1

```

```

tmp2=/tmp/$$tmp2
tmp3=/tmp/$$tmp3

netstat -i | grep "$servername ">$tmp1
sleep $SAR_LENGTH
netstat -i | grep "$servername ">$tmp2

awk '{print $4|" "$5|" "$6|" "$7|" "$8|" "$9|" "$10}' $tmp2 > $tmp3

for i in $(awk '{print $4|" "$5|" "$6|" "$7|" "$8|" "$9|" "$10}' $tmp1 )
do
#Address|Ipkts|Ierrs|Opkts|Oerrs|Collis|Queue
  address=$( echo $i | awk -F"|" '{print $1}')
  j=$(grep $address $tmp3)

  let pktsin1=$( echo $i | awk -F"|" '{print $2}')
  let errorsin1=$( echo $i | awk -F"|" '{print $3}')
  let pktsout1=$( echo $i | awk -F"|" '{print $4}')
  let errorsout1=$( echo $i | awk -F"|" '{print $5}')
  let colision1=$( echo $i | awk -F"|" '{print $6}')
  let queue1=$( echo $i | awk -F"|" '{print $7}')

  let pktsin2=$( echo $j | awk -F"|" '{print $2}')
  let errorsin2=$( echo $j | awk -F"|" '{print $3}')
  let pktsout2=$( echo $j | awk -F"|" '{print $4}')
  let errorsout2=$( echo $j | awk -F"|" '{print $5}')
  let colision2=$( echo $j | awk -F"|" '{print $6}')
  let queue2=$( echo $j | awk -F"|" '{print $7}')

  let pktsin=$pktsin2-$pktsin1
  let errorsin=$errorsin2-$errorsin1
  let pktsout=$pktsout2-$pktsout1
  let errorsout=$errorsout2-$errorsout1
  let colision=$colision2-$colision1
  let queue=$queue2-$queue1

  print
"$servername|$fecha|$pktsin|$errorsin|$pktsout|$errorsout|$colision|$queue"
done

rm $tmp1
rm $tmp2
rm $tmp3

```

1.B Programas de explotación de estadísticas

1.B.1 pais_vmstat_cpu_hora.ksh

```

=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: pais_vmstat_cpu_hora.ksh

# Objetivo:
#   Mostrar la disponibilidad del cpu resumida por día

```

```

#           - Cantidad de procesos en Ready Queue
#           - Procesos en espera por recursos de I/O
#           - Disponibilidad del servidor

# Archivo de parametrizacion:
#           - paises_detalle.lst      relación de servidores con una
instancia R/3
#=====

tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
tmp2=/tmp/$$.$tmp2
tmp3=/tmp/$$.$tmp3
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=pais_vmstat_memoria_$$_$f_

let F_LOOP=1

lee_alarmas

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

rotulos "CPU por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales R:$REA
B:$BLK ID:$ID $TIME"
#" $b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"

let col=0
let tot_ren=0
let tot_col=0
let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    str=${str}$SERVER" "
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str})"
#print $str

#presione_enter

#           mem=$( grep $SERVER $MEMORIA_LIST | awk -F"|" '{print $2}' )

```

```

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

create table $TMPTABLE as
select to_char(m_date,'HH24:MI:SS') Hora,
server, r, b, id
from vmstat
where to_char(m_date,'YMMDD') = to_char(sysdate+($DIAS),'YMMDD')
and server IN $str
order by m_date;

select 'xxx', h.c_time1, server,
avg(r),
avg(b),
avg(id)
from $TMPTABLE, $TIME h
where hora between c_time1
and c_time2
group by 1, h.c_time1, server;

select unique 'yyy' || ' ' || h.c_time1
from $TMPTABLE, $TIME h
where hora between c_time1
and c_time2;

drop table $TMPTABLE;

exit;
EOF

#cat $tmp
#presione_enter
grep xxx $tmp > $tmp1

grep yyy $tmp > $tmp3

let ren=0
while read nada fecha
do
    let ren=$ren+1
    a_hora[$ren]=$fecha
done < $tmp3
let tot_ren=$ren

let ren=1

while [ $ren -le $tot_ren ]
do

let col=0
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # ler for
do

```

```

SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
# grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2
let indice=$ren+$col*$tot_ren
# print $indice $ren $col $tot_ren
grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2 2>/dev/null
if [ $? -eq 0 ]
then
#       cat $tmp2
       while read nada fecha server rea blk id
       do
               a_r[$indice]=$rea
               a_b[$indice]=$blk
               a_id[$indice]=$id
               #print $b ${a_r[$indice]} ${a_b[$indice]}
${a_id[$indice]} $n
               done < $tmp2
       else
               a_r[$indice]=0
               a_b[$indice]=0
               a_id[$indice]=100
               #print ${a_r[$indice]} ${a_b[$indice]} ${a_id[$indice]}
       fi
       let col=$col+1
done

let ren=$ren+1
done

#while read nada hora rea blk id
#do
#   let ren=$ren+1
#   let indice=$ren+$col*$tot_ren
#   a_hora[$ren]=$hora
#   a_r[$indice]=$rea
#   a_b[$indice]=$blk
#   a_id[$indice]=$id
#   print $indice $ren $col $po $paging
#done < $tmp1

if [ $col -eq 0 ]
then
       let tot_ren=$ren
fi

#let col=$col+1
#a_server[$col]=$SERVER
#a_memoria[$col]=$mem

rm $tmp
rm $tmp1

#done # 1er for

let tot_col=$col

printf "%8s" " "

```

```

let c=1
while [ $c -le $tot_col ]
do
    server=${a_server[$c]}
    printf "%12s" $server
    let c=$c+1
done
printf " \n"

#printf "%8s" " "
#let c=1
#while [ $c -le $tot_col ]
#do
#    mem=${a_memoria[$c]}
#    printf "%11s" $mem
#    let c=$c+1
#done
#printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%3s %3s %4s" r b id
    let c=$c+1
done
printf " \n"

#printf "%8s" " "
#let c=1
#while [ $c -le $tot_col ]
#do
#    printf "%5s %5s" "/sKb" "%Tot"
#    let c=$c+1
#done
#printf " \n"

let r=1
while [ $r -le $tot_ren ]
do
    printf "%8s" ${a_hora[$r]}

    let c=1
    while [ $c -le $tot_col ]
    do
        let indice=$r+($tot_ren*($c-1))
#        a_po[$indice]=$po
#        a_paging[$indice]=$paging

        if [ -z "${a_r[$indice]}" ]
        then
            a_r[$indice]=0
        fi

        if [ -z "${a_b[$indice]}" ]
        then

```

```

        a_b[$indice]=0
    fi

    if [ -z "${a_id[$indice]}" ]
    then
        a_id[$indice]=0
    fi

    if [ ${a_r[$indice]} -ge $REA -o ${a_b[$indice]} -ge $BLK -
o ${a_id[$indice]} -le $ID ]
    then
        printf "${b}%3.0f %3.0f %3.0f%%${n}" ${a_r[$indice]}
${a_b[$indice]} ${a_id[$indice]}
#         printf "%5.0f %4.0f%%" ${a_po[$indice]}
${a_paging[$indice]}
    else
        printf "%3.0f %3.0f %3.0f%%" ${a_r[$indice]}
${a_b[$indice]} ${a_id[$indice]}
    fi

    let c=$c+1
done

    printf "\n"
    let r=$r+1
done

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

modifica_parametros_query_paises

done # -- 1er while

rm $tmp
rm $tmp1
rm $tmp2
rm $tmp3

```

1.B.2 pais_vmstat_cpu_dia.ksh

```

=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: pais_vmstat_cpu_dia.ksh

# Objetivo:
#     Mostrar la disponibilidad del cpu resumida por día
#         - Cantidad de procesos en Ready Queue
#         - Procesos en espera por recursos de I/O
#         - Disponibilidad del servidor

# Archivo de parametrizacion:
#         - paises_detalle.lst      relación de servidores con una
instancia R/3
=====

```



```

tmp=/tmp/$$tmp
tmp1=/tmp/$$tmp1
tmp2=/tmp/$$tmp2
tmp3=/tmp/$$tmp3
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=pais_vmstat_memoria_$$_f_

let F_LOOP=1

#lee_alarmas
lee_alarmas_wa

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

#rotulos "Memoria por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
R:$REA B:$BLK ID:$ID"
rotulos "CPU por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
R:$REA_W $REA_A B:$BLK_W $BLK_A ID:$ID_W $ID_A "
#" $b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"

let col=0
let tot_ren=0
let tot_col=0
let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    str=${str}$SERVER"|"
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str})"
#print $str

#presione_enter

# mem=$( grep $SERVER $MEMORIA_LIST | awk -F"|" '{print $2}' )

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

```

```

create table $TMPTABLE as
select to_char(m_date,'YY/MM/DD') fecha,
server, r, b, id
from vmstat
where m_date between sysdate-31 and sysdate
and server IN $str
order by m_date;

select 'xxx', fecha, server,
avg(r),
avg(b),
avg(id)
from $TMPTABLE
group by 1, fecha, server;

select unique 'yyy' || ' ' || fecha
from $TMPTABLE;

drop table $TMPTABLE;

exit;
EOF

#cat $tmp
grep xxx $tmp > $tmp1

grep yyy $tmp > $tmp3

let ren=0
while read nada fecha
do
    let ren=$ren+1
    a_hora[$ren]=$fecha
done < $tmp3
let tot_ren=$ren

let ren=1

while [ $ren -le $tot_ren ]
do

let col=0
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
# grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2
let indice=$ren+$col*$tot_ren
# print $indice $ren $col $tot_ren
grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2 2>/dev/null
if [ $? -eq 0 ]
then
# cat $tmp2
while read nada fecha server rea blk id
do

```

```

        a_r[$indice]=$rea
        a_b[$indice]=$blk
        a_id[$indice]=$id
        #print $b ${a_r[$indice]} ${a_b[$indice]}
${a_id[$indice]} $n
        done < $tmp2
    else
        a_r[$indice]=0
        a_b[$indice]=0
        a_id[$indice]=100
        #print ${a_r[$indice]} ${a_b[$indice]} ${a_id[$indice]}
    fi
    let col=$col+1
done

let ren=$ren+1
done

#while read nada hora rea blk id
#do
#    let ren=$ren+1
#    let indice=$ren+$col*$tot_ren
#    a_hora[$ren]=$hora
#    a_r[$indice]=$rea
#    a_b[$indice]=$blk
#    a_id[$indice]=$id
#    print $indice $ren $col $po $paging
#done < $tmp1

if [ $col -eq 0 ]
then
    let tot_ren=$ren
fi

#let col=$col+1
#a_server[$col]=$SERVER
#a_memoria[$col]=$mem

rm $tmp
rm $tmp1

#done # 1er for

let tot_col=$col

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    server=${a_server[$c]}
    printf "%12s" $server
    let c=$c+1
done
printf " \n"

#printf "%8s" " "

```

```

#let c=1
#while [ $c -le $tot_col ]
#do
#   mem=${a_memoria[$c]}
#   printf "%11s" $mem
#   let c=$c+1
#done
#printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%3s %3s %4s" r b id
    let c=$c+1
done
printf " \n"

#printf "%8s" " "
#let c=1
#while [ $c -le $tot_col ]
#do
#   printf "%5s %5s" "/sKb" "%Tot"
#   let c=$c+1
#done
#printf " \n"

let r=1
while [ $r -le $tot_ren ]
do
    printf "%8s" ${a_hora[$r]}

    let c=1
    while [ $c -le $tot_col ]
    do
        let indice=$r+($tot_ren*($c-1))
#       a_po[$indice]=$po
#       a_paging[$indice]=$paging

        if [ -z "${a_r[$indice]}" ]
        then
            a_r[$indice]=0
        fi

        if [ -z "${a_b[$indice]}" ]
        then
            a_b[$indice]=0
        fi

        if [ -z "${a_id[$indice]}" ]
        then
            a_id[$indice]=0
        fi

#       if [ ${a_r[$indice]} -ge $REA -o ${a_b[$indice]} -ge $BLK -
o ${a_id[$indice]} -le $ID ]

```

```

#           then
#           printf "${b}%3.0f %3.0f %3.0f%%${n}" ${a_r[$indice]}
${a_b[$indice]} ${a_id[$indice]}
#           else
#           printf "%3.0f %3.0f %3.0f%%" ${a_r[$indice]}
${a_b[$indice]} ${a_id[$indice]}
#           fi

if [ ${a_r[$indice]} -lt $REA_W ]
then
    printf "%3.0f " ${a_r[$indice]}
elif [ ${a_r[$indice]} -ge $REA_W -a ${a_r[$indice]} -lt
$REA_A ]
then
    printf "${b}%3.0f ${n}" ${a_r[$indice]}
elif [ ${a_r[$indice]} -ge $REA_A ]
then
    printf "${sit}%3.0f ${rit}" ${a_r[$indice]}
fi

if [ ${a_b[$indice]} -lt $BLK_W ]
then
    printf "%3.0f " ${a_b[$indice]}
elif [ ${a_b[$indice]} -ge $BLK_W -a ${a_b[$indice]} -lt
$BLK_A ]
then
    printf "${b}%3.0f ${n}" ${a_b[$indice]}
elif [ ${a_b[$indice]} -ge $BLK_A ]
then
    printf "${sit}%3.0f ${rit}" ${a_b[$indice]}
fi

if [ ${a_id[$indice]} -gt $ID_W ]
then
    printf "%4.0f" ${a_id[$indice]}
elif [ ${a_id[$indice]} -le $ID_W -a ${a_id[$indice]} -gt
$ID_A ]
then
    printf "${b}%4.0f${n}" ${a_id[$indice]}
elif [ ${a_id[$indice]} -le $ID_A ]
then
    printf "${sit}%4.0f${rit}" ${a_id[$indice]}
fi

    let c=$c+1
done

    printf "\n"
    let r=$r+1
done

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

modifica_parametros_query_paises

```

```
done # -- 1er while
```

```
rm $tmp
rm $tmp1
rm $tmp2
rm $tmp3
```

1.B.3 pais_sar_cpu_dia.ksh

```
=====
```

```
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales
```

```
# Programa: pais_sar_cpu_dia.ksh
```

```
# Objetivo:
```

```
#   Mostrar la disponibilidad del cpu resumida por dia
#   Porcentaje de utilización por:
#       - Usr procesos de usuario
#       - Sys procesos de kernel (sistema, llamadas a UNIX)
#       - Wio procesos en espera por IO
#       - enq procesos en la cola de espera
```

```
# Archivo de parametrizacion:
```

```
#   - paises_detalle.lst      relación de servidores con una
instancia R/3
```

```
=====
```

```
tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
tmp2=/tmp/$$.$tmp2
tmp3=/tmp/$$.$tmp3
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=pais_sar_cpu_dia_$$_$f_
```

```
let F_LOOP=1
```

```
#lee_alarmas
lee_alarmas_wa
```

```
while [ $F_LOOP -eq 1 ] # 1er while
do
```

```
clear
```

```
#rotulos "CPU por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales R:$REA
B:$BLK ID:$ID $TIME"
#" $b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"
rotulos "CPU por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
US:$US_W $US_A SY:$SY_W $SY_A IO:$IO_W $IO_A ID:$ID_W $ID_A EQ:$EQ_W
$EQ_A "
```

```
let col=0
let tot_ren=0
let tot_col=0
```

```

let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # ler for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    str=${str}$SERVER"|"
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str}"),"

#print $str
#presione_enter

# mem=$( grep $SERVER $MEMORIA_LIST | awk -F"|" '{print $2}' )

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

create table $TMPTABLE as
select to_char(m_date,'YY/MM/DD') fecha,
server, p_user, p_sys, p_wio, p_idle, enqueue
from cpu
where m_date between sysdate-31 and sysdate
and server IN $str
order by m_date;

select 'xxx', fecha, server,
avg(p_user),
avg(p_sys),
avg(p_wio),
avg(p_idle),
avg(enqueue)
from $TMPTABLE
group by 1, fecha, server;

select unique 'yyy' || ' ' || fecha
from $TMPTABLE;

drop table $TMPTABLE;

exit;
EOF

#cat $tmp
#presione_enter

```

```

grep xxx $tmp > $tmp1

grep yyy $tmp > $tmp3

let ren=0
while read nada fecha
do
    let ren=$ren+1
    a_hora[$ren]=$fecha
done < $tmp3
let tot_ren=$ren

let ren=1

while [ $ren -le $tot_ren ]
do

let col=0
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
#    grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2
let indice=$ren+$col*$tot_ren
#    print $indice $ren $col $tot_ren
grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2 2>/dev/null
if [ $? -eq 0 ]
then
#        cat $tmp2

#select 'xxx', h.c_time1, server,
#avg(p_user),
#avg(p_sys),
#avg(p_wio),
#avg(p_idle),
#avg(enque)
while read nada fecha server us sy io id eq
do
    a_us[$indice]=$us
    a_sy[$indice]=$sy
    a_io[$indice]=$io
    a_id[$indice]=$id
    a_eq[$indice]=$eq
done < $tmp2
else
    a_us[$indice]=0
    a_sy[$indice]=0
    a_io[$indice]=0
    a_id[$indice]=100
    a_eq[$indice]=0
fi
let col=$col+1
done

let ren=$ren+1

```



```

done

if [ $col -eq 0 ]
then
    let tot_ren=$ren
fi

rm $tmp
rm $tmp1

#done # 1er for

printf "%8s " " "
let c=1
while [ $c -le $tot_col ]
do
    server=${a_server[$c]}
    printf "%15s " $server
    let c=$c+1
done
printf " \n"

printf "%8s " " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%2s %2s %2s %3s %2s " us sy io id eq
    let c=$c+1
done
printf " \n"

let r=1
while [ $r -le $tot_ren ]
do
    printf "%8s " ${a_hora[$r]}

    let c=1
    while [ $c -le $tot_col ]
    do
        let indice=$r+($tot_ren*($c-1))
#         a_us[$indice]=0
#         a_sy[$indice]=0
#         a_io[$indice]=0
#         a_id[$indice]=100
#         a_eq[$indice]=0

        if [ -z "${a_us[$indice]}" ]
        then
            a_us[$indice]=0
        fi

        if [ -z "${a_sy[$indice]}" ]
        then
            a_sy[$indice]=0

```

```

fi

if [ -z "${a_io[$indice]}" ]
then
    a_io[$indice]=0
fi

if [ -z "${a_id[$indice]}" ]
then
    a_id[$indice]=0
fi

if [ -z "${a_eq[$indice]}" ]
then
    a_eq[$indice]=0
fi

if [ ${a_us[$indice]} -lt $US_W ]
then
    printf "%2.0f " ${a_us[$indice]}
elif [ ${a_us[$indice]} -ge $US_W -a ${a_us[$indice]} -lt
$US_A ]
then
    printf "${b}%2.0f ${n}" ${a_us[$indice]}
elif [ ${a_us[$indice]} -ge $US_A ]
then
    printf "${sit}%2.0f ${rit}" ${a_us[$indice]}
fi

if [ ${a_sy[$indice]} -lt $SY_W ]
then
    printf "%2.0f " ${a_sy[$indice]}
elif [ ${a_sy[$indice]} -ge $SY_W -a ${a_sy[$indice]} -lt
$SY_A ]
then
    printf "${b}%2.0f ${n}" ${a_sy[$indice]}
elif [ ${a_sy[$indice]} -ge $SY_A ]
then
    printf "${sit}%2.0f ${rit}" ${a_sy[$indice]}
fi

if [ ${a_io[$indice]} -lt $IO_W ]
then
    printf "%2.0f " ${a_io[$indice]}
elif [ ${a_io[$indice]} -ge $IO_W -a ${a_io[$indice]} -lt
$IO_A ]
then
    printf "${b}%2.0f ${n}" ${a_io[$indice]}
elif [ ${a_io[$indice]} -ge $IO_A ]
then
    printf "${sit}%2.0f ${rit}" ${a_io[$indice]}
fi

if [ ${a_id[$indice]} -gt $ID_W ]
then
    printf "%3.0f " ${a_id[$indice]}

```

```

        elif [ ${a_id[$indice]} -le $ID_W -a ${a_id[$indice]} -gt
$ID_A ]
        then
            printf "${b}%3.0f ${n}" ${a_id[$indice]}
        elif [ ${a_id[$indice]} -le $ID_A ]
        then
            printf "${sit}%3.0f ${rit}" ${a_id[$indice]}
        fi

        if [ ${a_eq[$indice]} -lt $EQ_W ]
        then
            printf "%2.0f " ${a_eq[$indice]}
        elif [ ${a_eq[$indice]} -ge $EQ_W -a ${a_eq[$indice]} -lt
$EQ_A ]
        then
            printf "${b}%2.0f ${n}" ${a_eq[$indice]}
        elif [ ${a_eq[$indice]} -ge $EQ_A ]
        then
            printf "${sit}%2.0f ${rit}" ${a_eq[$indice]}
        fi

        let c=$c+1
    done

    printf "\n"
    let r=$r+1
done

```

```

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

```

```

modifica_parametros_query_paises

```

```

done # -- 1er while

```

```

rm $tmp
rm $tmp1
rm $tmp2
rm $tmp3

```

1.B.4 pais_sar_cpu_hora.ksh

```

=====

```

```

# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

```

```

# Programa: pais_sar_cpu_hora.ksh

```

```

# Objetivo:

```

```

#   Mostrar la disponibilidad del cpu resumida por hora
#   Porcentaje de utilización por:
#       - Usr procesos de usuario
#       - Sys procesos de kernel (sistema, llamadas a UNIX)
#       - Wio procesos en espera por IO
#       - enq procesos en la cola de espera

```

```

# Archivo de parametrizacion:
#      - paises_detalle.lst      relación de servidores con una
instancia R/3
#=====

tmp=/tmp/$$tmp
tmp1=/tmp/$$tmp1
tmp2=/tmp/$$tmp2
tmp3=/tmp/$$tmp3
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=pais_sar_memoria_$$_f_

let F_LOOP=1

#lee_alarmas
lee_alarmas_wa

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

#rotulos "CPU por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales R:$REA
B:$BLK ID:$ID $TIME"
#" $b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"
rotulos "CPU por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
US:$US_W $US_A SY:$SY_W $SY_A IO:$IO_W $IO_A ID:$ID_W $ID_A EQ:$EQ_W
$EQ_A "

let col=0
let tot_ren=0
let tot_col=0
let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    str=${str}$SERVER""
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str})"
#print $str

#presione_enter

#      mem=$( grep $SERVER $MEMORIA_LIST | awk -F"|" '{print $2}' )

```

```

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

create table $TMPTABLE as
select to_char(m_date,'HH24:MI:SS') Hora,
server, p_user, p_sys, p_wio, p_idle, enqueue
from cpu
where to_char(m_date,'YYMMDD') = to_char(sysdate+($DIAS),'YYMMDD')
and server IN $str
order by m_date;

select 'xxx', substr(h.c_time1,1,5), server,
avg(p_user),
avg(p_sys),
avg(p_wio),
avg(p_idle),
avg(enqueue)
from $TMPTABLE, $TIME h
where hora between c_time1
and c_time2
group by 1, h.c_time1, server;

select unique 'yyy' || ' ' || substr(h.c_time1,1,5)
from $TMPTABLE, $TIME h
where hora between c_time1
and c_time2;

drop table $TMPTABLE;

exit;
EOF

#cat $tmp
#presione_enter

grep xxx $tmp > $tmp1

grep yyy $tmp > $tmp3

let ren=0
while read nada fecha
do
    let ren=$ren+1
    a_hora[$ren]=$fecha
done < $tmp3
let tot_ren=$ren

let ren=1

while [ $ren -le $tot_ren ]
do

```

```

let col=0
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
#    grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2
    let indice=$ren+$col*$tot_ren
#    print $indice $ren $col $tot_ren
    grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2 2>/dev/null
    if [ $? -eq 0 ]
    then
#        cat $tmp2

#select 'xxx', h.c_time1, server,
#avg(p_user),
#avg(p_sys),
#avg(p_wio),
#avg(p_idle),
#avg(enque)
        while read nada fecha server us sy io id eq
        do
            a_us[$indice]=$us
            a_sy[$indice]=$sy
            a_io[$indice]=$io
            a_id[$indice]=$id
            a_eq[$indice]=$eq
        done < $tmp2
    else
            a_us[$indice]=0
            a_sy[$indice]=0
            a_io[$indice]=0
            a_id[$indice]=100
            a_eq[$indice]=0
        fi
        let col=$col+1
done

let ren=$ren+1
done

if [ $col -eq 0 ]
then
    let tot_ren=$ren
fi

rm $tmp
rm $tmp1

#done # 1er for

printf "%5s " " "
let c=1
while [ $c -le $tot_col ]
do
    server=${a_server[$c]}

```

```

        printf "%15s " $server
        let c=$c+1
done
printf " \n"

printf "%5s " " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%2s %2s %2s %3s %2s " us sy io id eq
    let c=$c+1
done
printf " \n"

let r=1
while [ $r -le $tot_ren ]
do
    printf "%5s " ${a_hora[$r]}

    let c=1
    while [ $c -le $tot_col ]
    do
        let indice=$r+($tot_ren*($c-1))
#         a_us[$indice]=0
#         a_sy[$indice]=0
#         a_io[$indice]=0
#         a_id[$indice]=100
#         a_eq[$indice]=0

        if [ -z "${a_us[$indice]}" ]
        then
            a_us[$indice]=0
        fi

        if [ -z "${a_sy[$indice]}" ]
        then
            a_sy[$indice]=0
        fi

        if [ -z "${a_io[$indice]}" ]
        then
            a_io[$indice]=0
        fi

        if [ -z "${a_id[$indice]}" ]
        then
            a_id[$indice]=0
        fi

        if [ -z "${a_eq[$indice]}" ]
        then
            a_eq[$indice]=0
        fi

        if [ ${a_us[$indice]} -lt $US_W ]
        then

```

```

        printf "%2.0f " ${a_us[$indice]}
elif [ ${a_us[$indice]} -ge $US_W -a ${a_us[$indice]} -lt
$US_A ]
then
    printf "${b}%2.0f ${n}" ${a_us[$indice]}
elif [ ${a_us[$indice]} -ge $US_A ]
then
    printf "${sit}%2.0f ${rit}" ${a_us[$indice]}
fi

if [ ${a_sy[$indice]} -lt $SY_W ]
then
    printf "%2.0f " ${a_sy[$indice]}
elif [ ${a_sy[$indice]} -ge $SY_W -a ${a_sy[$indice]} -lt
$SY_A ]
then
    printf "${b}%2.0f ${n}" ${a_sy[$indice]}
elif [ ${a_sy[$indice]} -ge $SY_A ]
then
    printf "${sit}%2.0f ${rit}" ${a_sy[$indice]}
fi

if [ ${a_io[$indice]} -lt $IO_W ]
then
    printf "%2.0f " ${a_io[$indice]}
elif [ ${a_io[$indice]} -ge $IO_W -a ${a_io[$indice]} -lt
$IO_A ]
then
    printf "${b}%2.0f ${n}" ${a_io[$indice]}
elif [ ${a_io[$indice]} -ge $IO_A ]
then
    printf "${sit}%2.0f ${rit}" ${a_io[$indice]}
fi

if [ ${a_id[$indice]} -gt $ID_W ]
then
    printf "%3.0f " ${a_id[$indice]}
elif [ ${a_id[$indice]} -le $ID_W -a ${a_id[$indice]} -gt
$ID_A ]
then
    printf "${b}%3.0f ${n}" ${a_id[$indice]}
elif [ ${a_id[$indice]} -le $ID_A ]
then
    printf "${sit}%3.0f ${rit}" ${a_id[$indice]}
fi

if [ ${a_eq[$indice]} -lt $EQ_W ]
then
    printf "%2.0f " ${a_eq[$indice]}
elif [ ${a_eq[$indice]} -ge $EQ_W -a ${a_eq[$indice]} -lt
$EQ_A ]
then
    printf "${b}%2.0f ${n}" ${a_eq[$indice]}
elif [ ${a_eq[$indice]} -ge $EQ_A ]
then
    printf "${sit}%2.0f ${rit}" ${a_eq[$indice]}

```



```

        fi

        let c=$c+1
    done

    printf "\n"
    let r=$r+1
done

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

```

```

modifica_parametros_query_paises

```

```

done # -- 1er while

```

```

rm $tmp
rm $tmp1
rm $tmp2
rm $tmp3

```

1.B.5 pais_vmstat_memoria_dia.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: pais_vmstat_memoria_dia.ksh

# Objetivo:
#   Mostrar la cantidad de paginación del servidor de un día
#   sumariado por día
#   - Despliega la po/s y la proporción de la po/h referente al
#   total de memoria
#   - La cantidad de memoria del servidor

# Archivo de parametrizacion:
#   - paises_detalle.lst      relación de servidores con una
#   instancia R/3
#   - tabla_servers_memoria  memoria fisica de los servidores
#   (para calculo)
#   - memoria_list.lst      memoria fisica de los servidores
#   (para impresion)
#=====

tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
tmp2=/tmp/$$.$tmp2
tmp3=/tmp/$$.$tmp3
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=pais_vmstat_memoria_$$_$f_

let F_LOOP=1

#lee_alarmas
lee_alarmas_wa

```

```

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

#rotulos "Memoria por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
PAGING:${MEM}% $TIME"
rotulos "Memoria por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
MEM:$MEM_W $MEM_A "
#" $b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"

let col=0
let tot_ren=0
let tot_col=0
let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    if [ $mm -eq 1 ]
    then
        str="(' "
    else
        str=${str}","' "
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    a_mem[$indice]=$( grep $SERVER $MEMORIA_LIST | awk -F"|" '{print
$2}' )
    str=${str}$SERVER"' "
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str}*)" "
#print $str

#presione_enter

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

create table $TMPTABLE as
select to_char(m_date,'YY/MM/DD') fecha,
server, po
from vmstat
where m_date between sysdate-31 and sysdate
and server IN $str
order by m_date;

select 'xxx', fecha, s.server,
avg(po), avg ( 100 * (po*60*60) / (memoria_mb*1024) )
from $TMPTABLE t, servers_memoria s

```

```

where s.server = t.server
group by l, fecha, s.server;

select unique 'yyy' || ' ' || fecha
from $TMPTABLE;

drop table $TMPTABLE;

exit;
EOF

#cat $tmp
#presione_enter

grep xxx $tmp > $tmp1
grep yyy $tmp > $tmp3

let ren=0
while read nada fecha
do
    let ren=$ren+1
    a_hora[$ren]=$fecha
done < $tmp3
let tot_ren=$ren

let ren=1

while [ $ren -le $tot_ren ]
do

let col=0
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
#    grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2
    let indice=$ren+$col*$tot_ren
#    print $indice $ren $col $tot_ren
    grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2 2>/dev/null
    if [ $? -eq 0 ]
    then
#        cat $tmp2
        while read nada fecha server po pag
        do
            a_po[$indice]=$po
            a_paging[$indice]=$pag
            #print $b ${a_r[$indice]} ${a_b[$indice]}
            ${a_id[$indice]} $n
            done < $tmp2
        else
            a_po[$indice]=0
            a_paging[$indice]=0
            #print ${a_r[$indice]} ${a_b[$indice]} ${a_id[$indice]}
        fi
        let col=$col+1
    fi
done

```

```

done

let ren=$ren+1
done

#while read nada hora rea blk id
#do
#   let ren=$ren+1
#   let indice=$ren+$col*$tot_ren
#   a_hora[$ren]=$hora
#   a_r[$indice]=$rea
#   a_b[$indice]=$blk
#   a_id[$indice]=$id
#   print $indice $ren $col $po $paging
#done < $tmp1

if [ $col -eq 0 ]
then
    let tot_ren=$ren
fi

#let col=$col+1
#a_server[$col]=$SERVER
#a_memoria[$col]=$mem

rm $tmp
rm $tmp1

#done # 1er for

let tot_col=$col

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    server=${a_server[$c]}
    printf "%11s" $server
    let c=$c+1
done
printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%11s" ${a_mem[$c]}
    let c=$c+1
done
printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%5s %5s" "Pout" "Pagn"

```

```

        let c=$c+1
done
printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%5s %5s" "/sKb" "%Tot"
    let c=$c+1
done
printf " \n"

let r=1
while [ $r -le $tot_ren ]
do
    printf "%8s" "${a_hora[$r]}

    let c=1
    while [ $c -le $tot_col ]
    do
        let indice=$r+($tot_ren*( $c-1))

        printf "%5.0f " ${a_po[$indice]}

        if [ ${a_paging[$indice]} -lt $MEM_W ]
        then
            printf "%4.0f " ${a_paging[$indice]}
        elif [ ${a_paging[$indice]} -ge $MEM_W -a
${a_paging[$indice]} -lt $MEM_A ]
        then
            printf "${b}%4.0f ${n}" ${a_paging[$indice]}
        elif [ ${a_paging[$indice]} -ge $MEM_A ]
        then
            printf "${sit}%4.0f ${rit}" ${a_paging[$indice]}
        fi

        let c=$c+1
    done

    printf "\n"
    let r=$r+1
done

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

modifica_parametros_query_paises

done # -- 1er while

rm $tmp
rm $tmp1
rm $tmp2
rm $tmp3

```

1.B.6 pais_vmstat_memoria_hora.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: pais_vmstat_memoria_hora.ksh

# Objetivo:
#   Mostrar la cantidad de paginación del servidor de un día
#   desglosado por hora
#       - Despliega la po/s y la proporción de la po/h referente al
#   total de memoria
#       - La cantidad de memoria del servidor

# Archivo de parametrizacion:
#   - paises_detalle.lst      relación de servidores con una
#   instancia R/3
#   - tabla_servers_memoria  memoria fisica de los servidores
#   (para calculo)
#   - memoria_list.lst      memoria fisica de los servidores
#   (para impresion)
#=====

tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
tmp2=/tmp/$$.$tmp2
tmp3=/tmp/$$.$tmp3
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=pais_vmstat_memoria_$$_f_

let F_LOOP=1

#lee_alarmas
lee_alarmas_wa

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

rotulos "Memoria por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
MEM:$MEM_W $MEM_A "
#"$b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"

let col=0
let tot_ren=0
let tot_col=0
let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    fi

```

```

        SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
        let indice=$tot_col+1
        a_server[$indice]=$SERVER
        a_mem[$indice]=$( grep $SERVER $MEMORIA_LIST | awk -F"|" '{print
$2}' )
        str=${str}$SERVER""
        let mm=$mm+1
        let tot_col=$tot_col+1
done
str=${str}""
#print $str

#presione_enter

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

create table $TMPTABLE as
select to_char(m_date,'HH24:MI:SS') Hora,
server, po
from vmstat
where to_char(m_date,'YMMDD') = to_char(sysdate+($DIAS),'YMMDD')
and server IN $str
order by m_date;

select 'xxx', h.c_time1, s.server,
avg(po), avg ( 100 * (po*60*60) / (memoria_mb*1024) )
from $TMPTABLE t, $TIME h, servers_memoria s
where hora between c_time1
and c_time2 and t.server = s.server
group by 1, h.c_time1, s.server;

select unique 'yyy' || ' ' || h.c_time1
from $TMPTABLE, $TIME h
where hora between c_time1
and c_time2;

drop table $TMPTABLE;

exit;
EOF

#cat $tmp
#presione_enter
grep xxx $tmp > $tmp1

grep yyy $tmp > $tmp3

let ren=0
while read nada fecha
do
    let ren=$ren+1

```

```

        a_hora[$ren]=$fecha
done < $tmp3
let tot_ren=$ren

let ren=1

while [ $ren -le $tot_ren ]
do

let col=0
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
do
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
#    grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2
    let indice=$ren+$col*$tot_ren
#    print $indice $ren $col $tot_ren
    grep $SERVER $tmp1 | grep "${a_hora[$ren]}" > $tmp2 2>/dev/null
    if [ $? -eq 0 ]
    then
#        cat $tmp2
        while read nada fecha server po pag
        do
            a_po[$indice]=$po
            a_paging[$indice]=$pag
            #print $b ${a_r[$indice]} ${a_b[$indice]}
${a_id[$indice]} $n
            done < $tmp2
        else
            a_po[$indice]=0
            a_paging[$indice]=0
            #print ${a_r[$indice]} ${a_b[$indice]} ${a_id[$indice]}
        fi
        let col=$col+1
    done

let ren=$ren+1
done

#while read nada hora rea blk id
#do
#    let ren=$ren+1
#    let indice=$ren+$col*$tot_ren
#    a_hora[$ren]=$hora
#    a_r[$indice]=$rea
#    a_b[$indice]=$blk
#    a_id[$indice]=$id
#    print $indice $ren $col $po $paging
#done < $tmp1

if [ $col -eq 0 ]
then
    let tot_ren=$ren
fi

#let col=$col+1

```



```

#a_server[$col]=$SERVER
#a_memoria[$col]=$mem

rm $tmp
rm $tmp1

#done # 1er for

let tot_col=$col

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    server=${a_server[$c]}
    printf "%11s" $server
    let c=$c+1
done
printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%11s" ${a_mem[$c]}
    let c=$c+1
done
printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%5s %5s" "Pout" "Pagn"
    let c=$c+1
done
printf " \n"

printf "%8s" " "
let c=1
while [ $c -le $tot_col ]
do
    printf "%5s %5s" "/sKb" "%Tot"
    let c=$c+1
done
printf " \n"

let r=1
while [ $r -le $tot_ren ]
do
    printf "%8s" ${a_hora[$r]}

    let c=1
    while [ $c -le $tot_col ]
    do
        let indice=$r+($tot_ren*($c-1))

```

```

#         if [ ${a_paging[$indice]} -ge $MEM ]
#         then
#             printf "${b}%5.0f %4.0f%%${n}" ${a_po[$indice]}
${a_paging[$indice]}
#         else
#             printf "%5.0f %4.0f%%" ${a_po[$indice]}
${a_paging[$indice]}
#         fi

        printf "%5.0f " ${a_po[$indice]}

        if [ ${a_paging[$indice]} -lt $MEM_W ]
        then
            printf "%4.0f " ${a_paging[$indice]}
        elif [ ${a_paging[$indice]} -ge $MEM_W -a
${a_paging[$indice]} -lt $MEM_A ]
        then
            printf "${b}%4.0f ${n}" ${a_paging[$indice]}
        elif [ ${a_paging[$indice]} -ge $MEM_A ]
        then
            printf "${sit}%4.0f ${rit}" ${a_paging[$indice]}
        fi

        let c=$c+1
    done

    printf "\n"
    let r=$r+1
done

```

```

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

```

```

modifica_parametros_query_paises

```

```

done # -- ler while

```

```

rm $tmp
rm $tmp1
rm $tmp2
rm $tmp3

```

1.B.7 pais_iostat.ksh

```

#=====

```

```

# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

```

```

# Programa: pais_iostat.ksh

```

```

# Objetivo:

```

```

#     Mostrar el mapa de relación sdXXX cXtXdXsX discoVX
#     Mostrar el comportamiento de los discos resumido por día con las
siguientes metricas.

```

```

#         - svc_t tiempo de respuesta (ms)
#         - pctb porcentaje de uso

```

```

#         - krps  Kbytes leídos por segundo
#         - rps   lecturas por segundo
#         - kwps  Kbytes escritos por segundo
#         - wps   escrituras por segundo

# Archivo de parametrización:
#         - alarma.lst      relación de servidores con una instancia R/3

# Tabla de parametrización
#         - mapa_discos     server, sdxxx, cxtxdxsx, vxdisk
#=====

tmp=/tmp/$$tmp
tmp1=/tmp/$$tmp1
tmp2=/tmp/$$tmp2
tmp3=/tmp/$$tmp3
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=pais_iostat_$$_f_

let F_LOOP=1

lee_alarmas

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

rotulos "DISCO de la Instancia Central Pais $b$PAIS $FECHA_QUERY$n" "P:
$0" "Umbrales SCVT:${SVCT} BUSY:${PCTB}%"

let col=0
let tot_ren=0
let tot_col=0
let mm=1
#for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST ) # 1er for
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST | head -1 ) #
ler for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    str=${str}$SERVER"|"
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str} ")"

#print $str

```

```

#presione_enter

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

select 'xxx', disk.server, cctxdxsx, vg, vxdisk, sdxxx, avg(svc_t),
avg(pctb),
      avg(krps), avg(rps), avg(kwps), avg(wps)
from disk, mapa_discos
where to_char(m_date,'YMMDD') = to_char(sysdate+($DIAS),'YMMDD')
and disk.server IN $str
and trim(disk) = trim(sdxxx)
and disk.server = mapa_discos.server
group by disk.server, cctxdxsx, vg, vxdisk, sdxxx
order by avg(svc_t) desc;

exit;
EOF

#more $tmp

#rotulos "DISCO por Pais $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
SCVT:${SVCT}% %BUSY:${PCTB}"

#cat $tmp
#presione_enter

grep 'xxx' $tmp > $tmp1

#select 'xxx', disk.server, cctxdxsx, vg, vxdisk, sdxxx, avg(svc_t),
avg(pctb)
printf "${b}%3s %6s %12s %6s %10s %7s %7s %4s %6s %4s %4s %4s${n}\n"
"ind" "Server" cXtXdXsX vg vxdisk sdXXX "scvt" "%B" krps rps kwps wps

let r=1

while read nada server cctxdxsx vg vxdisk sdxxx svc_t pctb krps
rps kwps wps
do
    a_server[$r]=$server
    a_sdxxx[$r]=$sdxxx
    a_cctxdxsx[$r]=$cctxdxsx
    a_vxdisk[$r]=$vxdisk
    a_vg[$r]=$vg
    a_svc_t[$r]=$svc_t
    a_pctb[$r]=$pctb
    a_krps[$r]=$krps
    a_rps[$r]=$rps
    a_kwps[$r]=$kwps
    a_wps[$r]=$wps
    printf "%3d %6s %12s %6s %10s %7s %7.0f %4.0f %6.0f %4.0f %4.0f
%4.0f\n" $r ${a_server[$r]} $cctxdxsx $vg $vxdisk ${a_sdxxx[$r]}

```

```

${a_svc_t[$r]} ${a_pctb[$r]} ${a_krps[$r]} ${a_rps[$r]} ${a_kwps[$r]}
${a_wps[$r]}
    let r=$r+1
done < $tmp1

let NUM OPCIONES=$r-1

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

modifica_parametros_query_iostat

if [ $OPC_SELECC -ne 0 ]
then
    let i=$OPC_SELECC
    let CAMBIO_VISTA_DETALLE=1
    while [ $CAMBIO_VISTA_DETALLE -eq 1 ]
    do
        let CAMBIO_VISTA_DETALLE=0
        if [ $TIPO_DETALLE = 'dia' ]
        then
            a_server[$r]=$server
            a_sdxxx[$r]=$sdxxx
            a_cxtxdxsx[$r]=$cxtxdxsx
            a_vxdisk[$r]=$vxdisk
            $MENU_HOME/iostat_disco_dia.ksh ${a_server[$i]}
            ${a_sdxxx[$i]} ${a_cxtxdxsx[$i]} ${a_vxdisk[$i]} ${a_vg[$i]}
        else
            $MENU_HOME/iostat_disco_hora.ksh ${a_server[$i]}
            ${a_sdxxx[$i]} ${a_cxtxdxsx[$i]} ${a_vxdisk[$i]} ${a_vg[$i]}
        fi
        cual_vista
        printf "$0\n $CAMBIO_VISTA_DETALLE $TIPO_DETALLE\n"
    done
    OPC_SELECC=0
fi

done # -- ler while

rm $tmp
rm $tmp1

```

1.B.8 iostat_disco_hora.ksh

```

=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: iostat_pais_hora.ksh

# Objetivo:
#   Mostrar el comportamiento de los discos de un día desglosado por
hora
#   - svc_t tiempo de respuesta (ms)
#   - pctb porcentaje de uso
#   - krps Kbytes leidos por segundo
#   - rps lecturas por segundo

```

```

#           - kwps  Kbytes escritos por segundo
#           - wps   escrituras por segundo

# Archivo de parametrizacion:
#           - alarma.lst      relación de servidores con una instancia R/3

#=====

tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=iostat_disco_$$_$$f_

let F_LOOP=1

while [ $F_LOOP -eq 1 ]
do

clear

#$MENU_HOME/iostat_disco_hora.ksh ${a_server[$i]} ${a_sdxxx[$i]}
${a_cxtxdxsx[$i]} ${a_vxdisk[$i]} ${a_vg[$i]}

SERVER=$1
DISCO=$2
cxtxdxsx=$3
vxdisk=$4
vg=$5

rotulos "USO DEL DISCO $b$DISCO $cxtxdxsx $vxdisk $vg $n" "EN EL
SERVIDOR $b$SERVER$n Ultimos n dias" "P: $0" "Umbrales SCVT:${SVCT}
BUSY:${PCTB}%"
#####
#rotulos "EN EL SERVIDOR $b$SERVER $FECHA_QUERY$n" "P: $0"
#" $b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"

sqlplus -S monitor/monitor<<EOF>>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

create table $TMPTABLE as
select to_char(m_date,'HH24:MI:SS') Hora,
disk, svc_t, pctb, krps, rps, kwps, wps
from disk
where to_char(m_date,'YYMMDD') = to_char(sysdate+($DIAS),'YYMMDD')
and server = '$SERVER'
and disk = '$DISCO';

col krps for 99990
col rps for 9990
col kwps for 99990
col wps for 9990
select 'xxx', t_time.c_time1,
avg(svc_t),
avg(pctb ),

```

```

avg(krps),
avg(rps),
avg(kwps) kwps,
avg(wps) wps
from $TMPTABLE, t_time
where hora between c_time1
and c_time2
group by t_time.c_time1;

drop table $TMPTABLE;

exit;
EOF

#more $tmp

grep xxx $tmp > $tmp1

printf "${b}%12s %8s %8s %6s %4s %6s %4s${n}\n" hora  svc_t pctb krps
rps kwps wps
while read nada hora  svc_t pctb krps rps kwps wps
do
    if [ $svc_t -ge $SVCT -o $pctb -ge $PCTB ]
    then
        printf "${b}%12s %8.0f %8.0f %6.0f %4.0f %6.0f %4.0f ${n}
\n" $hora $svc_t $pctb $krps $rps $kwps $wps
    else
        printf "%12s %8.0f %8.0f %6.0f %4.0f %6.0f %4.0f \n" $hora
$svc_t $pctb $krps $rps $kwps $wps
#         printf "%12s %8.0f %8.0f %6.0f %4.0f %6.0f %4.0f \n" $hora
$svc_t $pctb $krps $rps $kwps $wps
    fi
done < $tmp1

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0
cambia_vista 0 $TIPO_DETALLE
modifica_parametros_query_detalle

done

```

```

rm $tmp
rm $tmp1

```

1.B.9 iostat_disco_dia.ksh

```

=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: iostat_pais_hora.ksh

# Objetivo:
#   Mostrar el comportamiento de los discos en los últimos 30 días
# resumido por día
#       - svc_t tiempo de respuesta (ms)
#       - pctb porcentaje de uso

```

```

#         - krps  Kbytes leidos por segundo
#         - rps   lecturas por segundo
#         - kwps  Kbytes escritos por segundo
#         - wps   escrituras por segundo

# Archivo de parametrizacion:
#         - alarma.lst      relación de servidores con una instancia R/3

#=====

tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=cpu_$$_f_

let F_LOOP=1

while [ $F_LOOP -eq 1 ]
do

clear

#$MENU_HOME/iostat_disco_hora.ksh ${a_server[$i]} ${a_sdxxx[$i]}
${a_cxtxdxsx[$i]} ${a_vxdisk[$i]} ${a_vg[$i]}

SERVER=$1
DISCO=$2
cxtxdxsx=$3
vxdisk=$4
vg=$5

rotulos "USO DEL DISCO $b$DISCO $cxtxdxsx $vxdisk $vg $n" "EN EL
SERVIDOR $b$SERVER$n Ultimos n dias" "P: $0" "Umbrales SCVT:${SVCT}
BUSY:${PCTB}%"

sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

select 'xxx', to_char(m_date,'YY/MM/DD') fecha,
avg(svc_t), avg(pctb), avg(krps), avg(rps), avg(kwps), avg(wps)
from disk
where m_date between sysdate-360 and sysdate
and server = '$SERVER' and disk = '$DISCO'
group by to_char(m_date,'YY/MM/DD');

exit;
EOF

#more $tmp

grep xxx $tmp > $tmp1

```



```

printf "%8s %8s %8s %8s %7s %8s %7s\n" fecha svc_t B% krps rps kwps wps
while read nada fecha svc_t pctb krps rps kwps wps
do
    if [ $svc_t -ge $SVCT -o $pctb -ge $PCTB ]
    then
        printf "${b}%8s %8.2f %8.2f %8.2f %7.2f %8.2f %7.2f ${n}\n"
$fecha $svc_t $pctb $krps $rps $kwps $wps
    else
        printf "%8s %8.2f %8.2f %8.2f %7.2f %8.2f %7.2f \n" $fecha
$svc_t $pctb $krps $rps $kwps $wps
    fi
done < $tmp1

```

```

cambia_vista 0 $TIPO_DETALLE
modifica_parametros_query_detalle

```

```
done
```

```
rm $tmp
rm $tmp1
```

1.B.10 central_all.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: central_all.ksh

# Objetivo:
#   Mostrar el comportamiento de cpu, memoria y disco desglosado por
hora
#       CPU
#       - r      Cantidad de procesos en Ready Queue
#       - b      Procesos en espera por recursos de I/O
#       - id     Disponibilidad del servidor
#       MEMORIA
#       - po     Kbytes paginados/s , columna po de vmstat
#       - avgpo  Proporción de po/h respecto al total de memoria del
servidor
#       DISCO
#       - svc_t  Tiempo de respuesta (ms)
#       - pctb   Porcentaje de uso
#       - krps   Kbytes leidos por segundo
#       - rps    Lecturas por segundo

# Archivo de parametrizacion:
#       - alarma_wa.lst   relación de servidores con una instancia
R/3

#=====

tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
tmp2=/tmp/$$.$tmp2
tmp3=/tmp/$$.$tmp3

```

```

f=$( date '+%Y%m%d%H%M%S' )
TMPTABLE=central_all_$$$_f_

let F_LOOP=1

lee_alarmas_wa

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

rotulos "Todas las metricas $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
R:$REA_W $REA_A B:$BLK_W $BLK_A ID:$ID_W $ID_A MEM:$MEM_W $MEM_A
SVCT:$SVCT_W $SVCT_A PCTB:$PCTB_W $PCTB_A "
#"$b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"

let col=0
let tot_ren=0
let tot_col=0
let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST | head -1 ) #
ler for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    str=${str}$SERVER""
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str} ")"

cat /dev/null > $tmp
sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

select 'xxx', t.c_time1,
avg(r), avg(b), avg(id),
avg(po), avg ( 100 * (po*60*60) / (memoria_mb*1024) )
from vmstat v, t_time2 t, servers_memoria s
where to_char(m_date,'YYMMDD') = to_char(sysdate+($DIAS),'YYMMDD')
and v.server = '$SERVER'
and to_char(m_date,'HH24:MI:SS') between c_time1 and c_time2
and v.server = s.server
group by c_time1
order by c_time1;

```

```

select 'yyy', c_time1, avg(svc_t), avg(pctb), avg(krps), avg(rps)
from disk, t_time2
where to_char(m_date,'YMMDD') = to_char(sysdate+($DIAS),'YMMDD')
and to_char(m_date,'HH24:MI:SS') between c_time1 and c_time2
and server = '$SERVER'
group by c_time1
order by c_time1;

exit;

EOF

#cat $tmp
#presione_enter

grep xxx $tmp > $tmp1
grep yyy $tmp > $tmp3

#printf "%12s %3.0f %3.0f %4.0f %7.0f %5.0f %6.0f %6.0f %6.0f %4.0f \n"
$hora $rea $blk $id $po $avgpo $svc_t $pctb $krps $rps
printf "${b}%12s %3s %3s %4s %7s %5s %6s %6s %6s %4s ${n}\n" hora r b
id po avgpo svc_t pctb krps rps

while read nada hora rea blk id po avgpo
do
    svc_t=$(grep $hora $tmp3 | awk '{print $3}')
    pctb=$(grep $hora $tmp3 | awk '{print $4}')
    krps=$(grep $hora $tmp3 | awk '{print $5}')
    rps=$(grep $hora $tmp3 | awk '{print $6}')
    if [ -z "$svc_t" ]
    then
        let svc_t=-1
        let pctb=-1
        let krps=-1
        let rps=-1
    fi
    printf "%12s " $hora

    if [ $rea -lt $REA_W ]
    then
        printf "%3.0f " $rea
    elif [ $rea -ge $REA_W -a $rea -lt $REA_A ]
    then
        printf "${b}%3.0f ${n}" $rea
    elif [ $rea -ge $REA_A ]
    then
        printf "${sit}%3.0f ${rit}" $rea
    fi

    if [ $blk -lt $BLK_W ]
    then
        printf "%3.0f " $blk
    elif [ $blk -ge $BLK_W -a $blk -lt $BLK_A ]
    then
        printf "${b}%3.0f ${n}" $blk
    elif [ $blk -ge $BLK_A ]

```

```

then
    printf "${sit}%3.0f ${rit}" $blk
fi

if [ $id -gt $ID_W ]
then
    printf "%4.0f " $id
elif [ $id -le $ID_W -a $id -gt $ID_A ]
then
    printf "${b}%4.0f ${n}" $id
elif [ $id -le $ID_A ]
then
    printf "${sit}%4.0f ${rit}" $id
fi

printf "%7.0f " $po

if [ $avgpo -lt $MEM_W ]
then
    printf "%5.0f " $avgpo
elif [ $avgpo -ge $MEM_W -a $avgpo -lt $MEM_A ]
then
    printf "${b}%5.0f ${n}" $avgpo
elif [ $avgpo -ge $MEM_A ]
then
    printf "${sit}%5.0f ${rit}" $avgpo
fi

#rotulos "Todas las metricas $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
R:$REA_W $REA_A B:$BLK_W $BLK_A ID:$ID_W $ID_A MEM:$MEM_W $MEM_A
SVCT:$SVCT_W $SVCT_A PCTB:$PCTB_W $PCTB_A "
if [ $svc_t -lt $SVCT_W ]
then
    printf "%6.0f " $svc_t
elif [ $svc_t -ge $SVCT_W -a $svc_t -lt $SVCT_A ]
then
    printf "${b}%6.0f ${n}" $svc_t
elif [ $svc_t -ge $SVCT_A ]
then
    printf "${sit}%6.0f ${rit}" $svc_t
fi

if [ $pctb -lt $PCTB_W ]
then
    printf "%6.0f " $pctb
elif [ $pctb -ge $PCTB_W -a $pctb -lt $PCTB_A ]
then
    printf "${b}%6.0f ${n}" $pctb
elif [ $pctb -ge $PCTB_A ]
then
    printf "${sit}%6.0f ${rit}" $pctb
fi

printf "%6.0f %4.0f \n" $krps $rps

# printf "%6.0f %3.0f %6.0f %4.0f \n" $svc_t $pctb $krps $rps

```

```

#      printf "%12s %3.0f %3.0f %4.0f %7.0f %5.0f %6.0f %6.0f %6.0f
%4.0f \n" $hora $rea $blk $id $po $avgpo $svc_t $pctb $krps $rps
done < $tmp1

#if [ ${a_r[$indice]} -ge $REA -o ${a_b[$indice]} -ge $BLK -o
${a_id[$indice]} -le $ID ]
#printf "${b}%3.0f %3.0f %3.0f%%${n}" ${a_r[$indice]} ${a_b[$indice]}
${a_id[$indice]}

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

modifica_parametros_query_paises

done # -- 1er while

rm $tmp
rm $tmp1
rm $tmp3

```

1.B.11 central_all_refresh.ksh

```

#=====
# UNAM ENEP ACATLAN
# Matematicas Aplicadas y Computacion
# Arturo de la Rosa Rosales

# Programa: central_all.ksh

# Objetivo:
#   Mostrar el comportamiento de cpu, memoria y disco desglosado por
hora actualizandose
#   automaticamente cada 80 segundos:
#       CPU
#       - r      Cantidad de procesos en Ready Queue
#       - b      Procesos en espera por recursos de I/O
#       - id     Disponibilidad del servidor
#       MEMORIA
#       - po     Kbytes paginados/s , columna po de vmstat
#       - avgpo  Proporción de po/h respecto al total de memoria del
servidor
#       DISCO
#       - svc_t  Tiempo de respuesta (ms)
#       - pctb   Porcentaje de uso
#       - krps   Kbytes leidos por segundo
#       - rps    Lecturas por segundo

# Archivo de parametrizacion:
#       - alarma_wa.lst   relación de servidores con una instancia
R/3

#=====

tmp=/tmp/$$.$tmp
tmp1=/tmp/$$.$tmp1
tmp2=/tmp/$$.$tmp2
tmp3=/tmp/$$.$tmp3
f=$( date '+%Y%m%d%H%M%S' )

```

```

TMPTABLE=central_all_$$_f_

let F_LOOP=1

lee_alarmas_wa

while [ $F_LOOP -eq 1 ] # 1er while
do

clear

rotulos "Todas las metricas $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
R:$REA_W $REA_A B:$BLK_W $BLK_A ID:$ID_W $ID_A MEM:$MEM_W $MEM_A
SVCT:$SVCT_W $SVCT_A PCTB:$PCTB_W $PCTB_A "
#"$b$SERVER $FECHA_QUERY $DIAS $CADENA_DIAS$n"

let col=0
let tot_ren=0
let tot_col=0
let mm=1
for server_aux in $( grep $PAIS $PAISES_DETALLE_LIST | head -1 ) #
ler for
do
    if [ $mm -eq 1 ]
    then
        str="("
    else
        str=${str}","
    fi
    SERVER=$( echo $server_aux | awk -F"|" '{print $2}' )
    let indice=$tot_col+1
    a_server[$indice]=$SERVER
    str=${str}$SERVER" "
    let mm=$mm+1
    let tot_col=$tot_col+1
done
str=${str} ")"

cat /dev/null > $tmp
sqlplus -S monitor/monitor<<EOF>$tmp
set heading off;
set pagesize 50000;
set linesize 200;

select 'xxx', t.c_time1,
avg(r), avg(b), avg(id),
avg(po), avg ( 100 * (po*60*60) / (memoria_mb*1024) )
from vmstat v, t_time2 t, servers_memoria s
where to_char(m_date,'YMMDD') = to_char(sysdate+($DIAS),'YMMDD')
and v.server = '$SERVER'
and to_char(m_date,'HH24:MI:SS') between c_time1 and c_time2
and v.server = s.server
group by c_time1
order by c_time1;

select 'yyy', c_time1, avg(svc_t), avg(pctb), avg(krps), avg(rps)

```

```

from disk, t_time2
where to_char(m_date,'YMMDD') = to_char(sysdate+($DIAS),'YMMDD')
and to_char(m_date,'HH24:MI:SS') between c_time1 and c_time2
and server = '$SERVER'
group by c_time1
order by c_time1;

exit;

EOF

#cat $tmp
#presione_enter

grep xxx $tmp > $tmp1
grep yyy $tmp > $tmp3

#printf "%12s %3.0f %3.0f %4.0f %7.0f %5.0f %6.0f %6.0f %6.0f %4.0f \n"
$hora $rea $blk $id $po $avgpo $svc_t $pctb $krps $rps
printf "${b}%12s %3s %3s %4s %7s %5s %6s %6s %6s %4s ${n}\n" hora r b
id po avgpo svc_t pctb krps rps

while read nada hora rea blk id po avgpo
do
    svc_t=$(grep $hora $tmp3 | awk '{print $3}')
    pctb=$(grep $hora $tmp3 | awk '{print $4}')
    krps=$(grep $hora $tmp3 | awk '{print $5}')
    rps=$(grep $hora $tmp3 | awk '{print $6}')
    if [ -z "$svc_t" ]
    then
        let svc_t=-1
        let pctb=-1
        let krps=-1
        let rps=-1
    fi
    printf "%12s " $hora

    if [ $rea -lt $REA_W ]
    then
        printf "%3.0f " $rea
    elif [ $rea -ge $REA_W -a $rea -lt $REA_A ]
    then
        printf "${b}%3.0f ${n}" $rea
    elif [ $rea -ge $REA_A ]
    then
        printf "${sit}%3.0f ${rit}" $rea
    fi

    if [ $blk -lt $BLK_W ]
    then
        printf "%3.0f " $blk
    elif [ $blk -ge $BLK_W -a $blk -lt $BLK_A ]
    then
        printf "${b}%3.0f ${n}" $blk
    elif [ $blk -ge $BLK_A ]
    then

```

```

        printf "${sit}%3.0f ${rit}" $blk
    fi

    if [ $id -gt $ID_W ]
    then
        printf "%4.0f " $id
    elif [ $id -le $ID_W -a $id -gt $ID_A ]
    then
        printf "${b}%4.0f ${n}" $id
    elif [ $id -le $ID_A ]
    then
        printf "${sit}%4.0f ${rit}" $id
    fi

    printf "%7.0f " $po

    if [ $avgpo -lt $MEM_W ]
    then
        printf "%5.0f " $avgpo
    elif [ $avgpo -ge $MEM_W -a $avgpo -lt $MEM_A ]
    then
        printf "${b}%5.0f ${n}" $avgpo
    elif [ $avgpo -ge $MEM_A ]
    then
        printf "${sit}%5.0f ${rit}" $avgpo
    fi

#rotulos "Todas las metricas $b$PAIS $FECHA_QUERY$n" "P: $0" "Umbrales
R:$REA_W $REA_A B:$BLK_W $BLK_A ID:$ID_W $ID_A MEM:$MEM_W $MEM_A
SVCT:$SVCT_W $SVCT_A PCTB:$PCTB_W $PCTB_A "
    if [ $svc_t -lt $SVCT_W ]
    then
        printf "%6.0f " $svc_t
    elif [ $svc_t -ge $SVCT_W -a $svc_t -lt $SVCT_A ]
    then
        printf "${b}%6.0f ${n}" $svc_t
    elif [ $svc_t -ge $SVCT_A ]
    then
        printf "${sit}%6.0f ${rit}" $svc_t
    fi

    if [ $pctb -lt $PCTB_W ]
    then
        printf "%6.0f " $pctb
    elif [ $pctb -ge $PCTB_W -a $pctb -lt $PCTB_A ]
    then
        printf "${b}%6.0f ${n}" $pctb
    elif [ $pctb -ge $PCTB_A ]
    then
        printf "${sit}%6.0f ${rit}" $pctb
    fi

    printf "%6.0f %4.0f \n" $krps $rps

#    printf "%6.0f %3.0f %6.0f %4.0f \n" $svc_t $pctb $krps $rps

```



```

#      printf "%12s %3.0f %3.0f %4.0f %7.0f %5.0f %6.0f %6.0f %6.0f
%4.0f \n" $hora $rea $blk $id $po $avgpo $svc_t $pctb $krps $rps
done < $tmp1

#if [ ${a_r[$indice]} -ge $REA -o ${a_b[$indice]} -ge $BLK -o
${a_id[$indice]} -le $ID ]
#printf "${b}%3.0f %3.0f %3.0f%%${n}" ${a_r[$indice]} ${a_b[$indice]}
${a_id[$indice]}

# en caso de realizar un modificacion define la variable F_LOOP en 1
# caso contrario 0

#modifica_parametros_query_paises
segs=1
while [ $segs -le 80 ]
do
    sleep 1
    printf "."
    let segs=$segs+1
done

done # -- 1er while

rm $tmp
rm $tmp1
rm $tmp3

```

1.B.12 lee_alarma

1.B.13 lee_alarma.ksh

1.C Programas de utilitarios

1.C.1 Mapa_discos.sh

```
DIR=/home/respaldo/fs2
```

```
FECHA=$(date +%Y/%m/%d')
```

```
iostat -ex | awk '{print $1|" }' > sdxxx.out
iostat -nx | awk '{print $11"s2"}' > cxtxdx.out
paste sdxxx.out cxtxdx.out > relacion.out
```

```
# Obtiene la salida del comando vxprint
printf "Server : "
read server
/usr/bin/sudo su - root -c "vxprint" > $DIR/salida.out
/usr/xpg4/bin/awk -v server=$server -f $DIR/volumenes2.awk
$DIR/salida.out > vx.out
```

```
for i in $( awk -F"|" '{print $4}' vx.out)
do
    printf "$(grep $i vx.out)"
    sdxxx=$(grep $i relacion.out|awk -F"|" '{print $1}')
```

```
print "|$sdxxx"
# grep $i relacion.out
done
```

1.C.2 volumenes2.awk

```
{
```

```
    if ( $1 == "Disk" && $2 == "group:" )
        group= $3

    if ( $1 == "dm" )
    {
        print server|"group"|"$2"|"$3
    }
}
```

1.C.3 mapa_discos.ctf

```
load data
infile mapa_discos.out
append into table mapa_discos
fields terminated '|' (
    SERVER,
    VG,
    VXDISK,
    CXTXDXSX,
    SDXXX
)
```