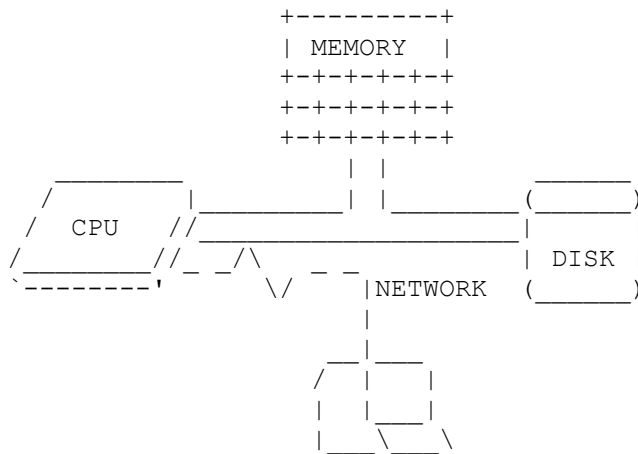


**REPASO DE ARQUITECTURA DE BASE DE DATOS, Y TUNING BASICO****Arquitectura básica de un RDBMS:**

Aquí nos referiremos a los componentes de una instancia de base de datos, cada una con sus propios componentes señalados a continuación independientemente que dos instancias corriendo en un mismo equipo deban compartir recursos físicos.

**Concepto de Performance.**

Entendemos por Performance de Manejo de bases de datos a la relación directa que existe entre los recursos que posee y el aprovechamiento de éstos. No existen unidades típicas de performance, pero generalmente están asociados bien a tiempos de respuesta, bien a porcentaje en el aprovechamiento de los recursos, o a una combinación de ambos.

Quizá el concepto en español más parecido a Performance sea rendimiento, sin embargo el término en inglés denota más que buena velocidad y aprovechamiento de los recursos, cuando usamos el verbo to perform, nos referimos a que el sistema, dispositivo o actividad realizándose está realizándose de forma adecuada.

Este rendimiento es producto del análisis a conciencia de las expectativas de nuestros usuarios respecto al equipo, al RDBMS y, por supuesto, a los sistemas que lo utilizan, sin embargo nos enfocaremos en este capítulo al performance del RDBMS.

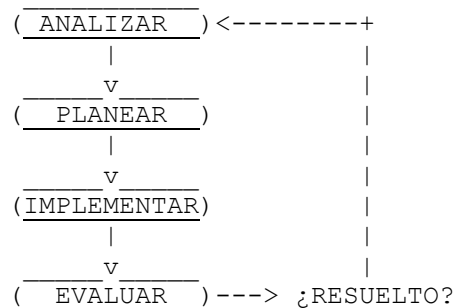
Una vez que nosotros hemos establecido las expectativas, y realizar el análisis de los diferentes puntos sujetos de mejora, se deben ajustar diferentes parámetros a nivel del RDBMS y del S.O., a esta labor de ajustar parámetros le denominamos afinación o tuning.

**Metodología.**

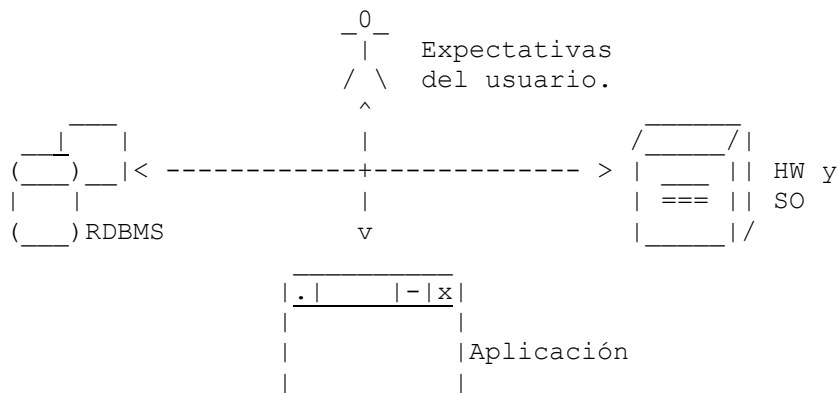
Los que conocimos los motores de automóvil con carburador, sabemos que no basta con saber cómo funciona el automóvil y como ajustar el tiempo del carburador y calibrar bujías para tenerlo afinado, se necesita de la experiencia y del continuo ejercicio para poder tener el resultado esperado.

En el caso de los RDBMS sucede algo parecido, se requiere experiencia y práctica para poder afinar adecuadamente un RDBMS, pero es indispensable seguir un método a fin de realizar las actividades adecuadamente y saber realmente si las acciones tomadas fueron adecuadas o no.

La metodología más sencilla se representa en el siguiente diagrama de flujo:



Aspectos a tomar en cuenta en el establecimiento de los objetivos durante el proceso de Performance and Tunning:



Expectativas de usuario: Si las expectativas de usuario no son susceptibles de ser alcanzadas bajo la situación óptima del equipo, no hay nada que el DBA pueda hacer. Es importante concientizar al usuario a fin de que sus expectativas sean lo más cercanas a la realidad posible.

Hardware y Sistema Operativo: Esta categoría abarca disco, procesador, redes y kernel del sistema operativo. Es importantísimo el examinar el performance del equipo físico en el que corre nuestro sistema.





- CPU utilization: segundo de proceso efectivo de CPU sobre segundos reales por cien.
- I/O Rate: Número de Accesos a I/O por segundo.
- I/O access time: Tiempo de uso al I/O en segundos.
- Paging rate: Número de páginas subidas a memoria por segundo.

Componente de Disco:

Access Time = Seek Time + Latency + Transfer Time

(--)--| Seek Time: Tiempo que le toma a la cabeza trasladarse
a un cilindro específico.
(--)--|
(--)--| Latency: Tiempo que le toma al disco el rotar sobre un
cilindro específico.

Transfer Time: Que tan rápido puede transferir la data desde y hacia el disco.

El disco es el factor más crítico en cuanto a performance se refiere, al ser el único dispositivo mecánico del RDBMS uno se puede imaginar cómo puede afectarse el performance en este componente.

Algunas recomendaciones al respecto son:

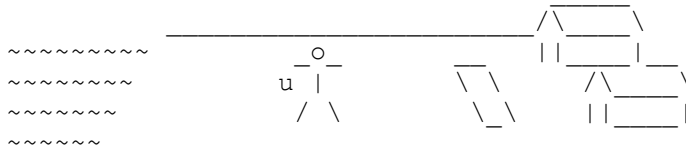
- Procure que las tablas más accedadas queden hacia los cilindros centrales del disco.
- Procure que el tamaño de los extents sea lo suficientemente grande para evitar que la cabeza tenga que trasladarse a otro sector y cilindro a menos de que sea necesario.
- Balancee las cargas en los discos disponibles fragmentando cuando esta sea posible.
- Si la fragmentación no es factible, use un esquema de RAID 0 para balancear las cargas.
- Evite en medida de los posibles el uso de RAID 5 (Performance Killer).
- Dimensione adecuadamente sus buffers.

Los buffers y el disco:

Imagine que tiene usted un lago repleto con agua y un poblado que requiere agua, usted tiene que trasladar el agua desde el lago hasta el pueblo con una cubeta :-) depositándola en una pileta común. En este ejemplo suponemos que en el traslado lo que realmente nos pesa es la distancia, no la cantidad, y dado que una vez que la ciudad requiera más agua tendremos que hacer más viajes pensemos:



¿Es mejor hacer más viajes con una cubeta chica?
¿Es mejor hacer menos viajes con una cubeta grande?



Dado que el peso del agua no es tanto problema, preferiremos hacer menos viajes con una cubeta más grande, en este ejemplo el lago son los datos en disco, el consumo de agua del pueblo es la demanda de información por parte de los usuarios y nuestra cubeta son los buffers, dado que en un RDBMS el acceso a disco es el más penalizado, es por eso que cada viaje cuesta más que el agua que se pueda cargar.

Es decir, de un buffer adecuadamente grande, dependerá en gran medida el correcto acceso a disco. Sin embargo la analogía no queda ahí, no necesariamente más es mejor, es decir, llega un punto en el que el tamaño de la cubeta satisface el abastecimiento y no por más grande que sea la cubeta vamos a tener menos viajes, por lo que un buffer exageradamente grande puede ser un desperdicio de recursos.

Componente de Memoria.

No hay gran cosa que se pueda afinar al respecto salvo el tomar las siguientes consideraciones:

- Asigne suficientes buffers, como se mencionó en el apartado anterior a fin de que el grado de paginación (número de lecturas de disco a memoria) sea el menor posible.
- Algunos sistemas operativos permiten forzar la permanencia de los segmentos en memoria física, evitando el tener que sea paginada y bajada a espacios temporales del S.O., cuando sea posible force la residencia en memoria física.
- Si el número de segmentos crece a más de cuatro (particularmente por el segmento virtual), se corre mayor riesgo de que sea generado en memoria virtual, es preferible tener un segmento de componente virtual grande que varios pequeños.

Componente de CPU.

Dado que la arquitectura de un RDBMS se basa en el procesos que simulan ser CPU's, llamados Procesadores Virtuales, la medición y optimización del performance en esta componente debe estar orientada al monitoreo de uso de los procesadores a nivel de S.O. (con el comando sar). De los VP's los más críticos y los que más consumen recursos son los CPU VP's por lo que recomendamos su monitoreo siguiendo las siguientes recomendaciones:



- Habilitar el No Aging para los CPU VP's cuando el S.O. lo permita.
- Habilitar y forzar el processor affinity (asociar un proceso a un procesador físico) para los CPU VP's cuando el S.O. lo permita.
- No habilitar más de $n-1$ CPU VP's donde n es el número de CPU's físicos.
- Si se tienen 2 CPU's físicos, configurar el RDBMS como si estuviera corriendo en máquinas con un solo procesador.

Componente de RED.

En realidad esta es la capa donde menos control se tiene, se pueden detectar cuellos de botella usando el comando de S.O. netstat (recordar que la primera línea siempre se ignora), si observamos que las colisiones son de dos dígitos (>9) se puede incorporar una tarjeta nueva, y un alias nuevo para esta tarjeta a fin de balancear la carga de red.

Por otra parte el uso de Stored Procedures en un esquema Cliente-Servidor, puede ayudar a desahogar la cantidad de datos que se transmiten por la red.

Estadísticas.

La mayoría de los RDBMS actuales tienen varias formas de resolver un mismo query, para identificar cual es la mejor manera de resolver dichos queries, los RDBMS poseen un componente llamado optimizador.

Dicho optimizador es el que toma la decisión de cómo resolver un query específico, obviamente, como en el mundo real, las estadísticas sobre los datos, índices y esquemas de la base de datos ayudan a nuestro optimizador a conseguir mejores tiempos de respuestas.

Para ello existen instrucciones para actualizar dichas estadísticas como por ejemplo UPDATE STATISTICS.

Se recomienda actualizar las estadísticas en las siguientes condiciones:

- Después de cargas masivas de información.
- Después de actualizaciones masivas de información.
- Periódicamente (por ejemplo una vez por semana).

Las estadísticas son la principal herramienta para optimizar procesos y queries problemáticos y deberán ser corridos ANTES de iniciar cualquier análisis de performance.



Conceptos vistos:

Performance
Tiempo de Respuesta
Tiempo de Comunicación
Tiempo de Procesamiento
Tiempo de Disco
Tiempo de Paginación
Tiempo de Respuesta de Usuario
Tiempo de Respuesta Interno
Tiempo de Respuesta de B.D.
Tiempo de Respuesta de I/O.
Uso CPU
Paging rate
Transaction Rate
I/O Rate
Access Time
Seek Time
Latency
Transfer time
Buffers de Memory
Estadísticas
Benchmark
