



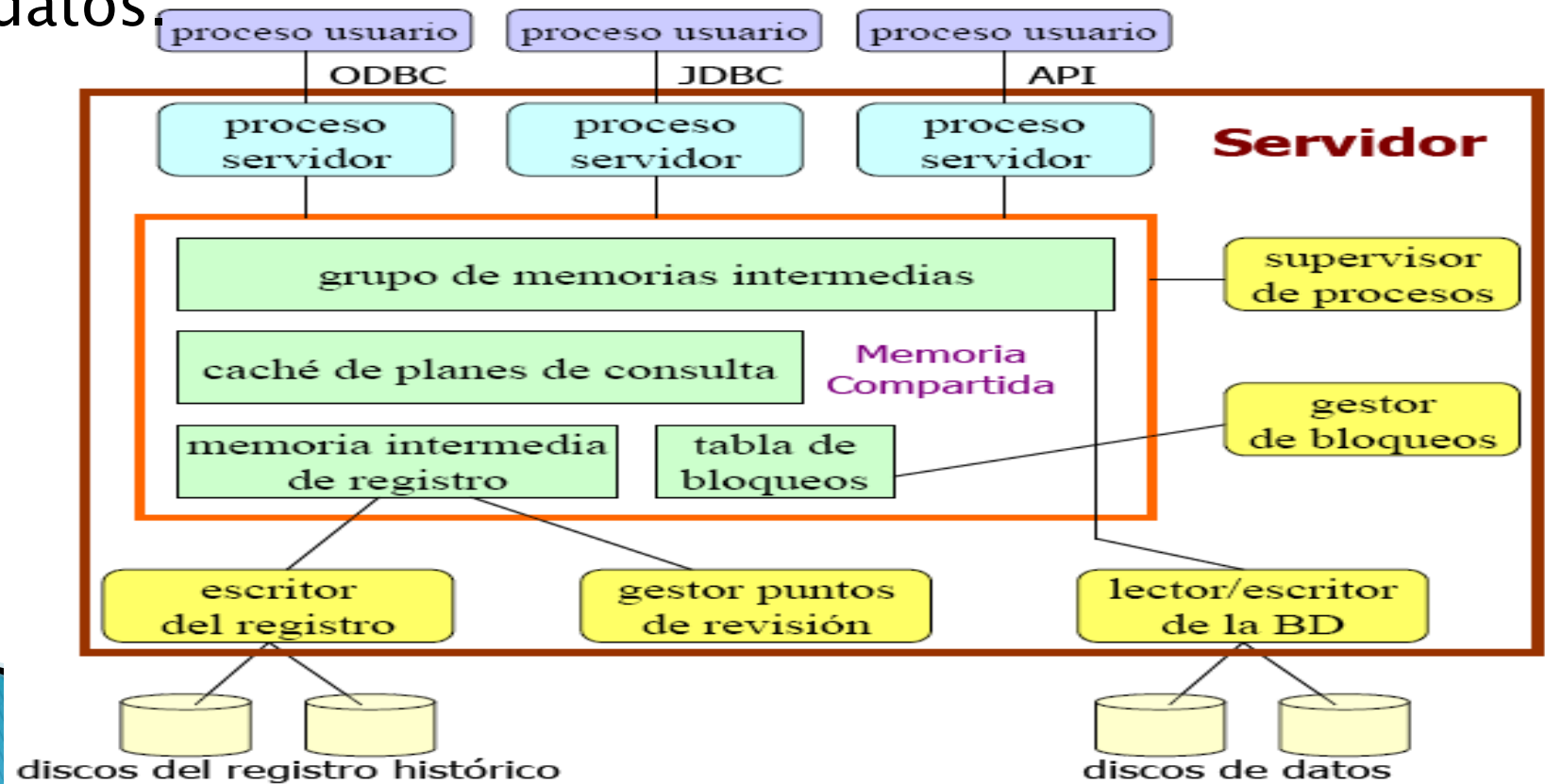
Tema 4

Conexiones dedicadas y compartidas: pool de conexiones.

José Luis Pastrana Brincones (pastrana@lcc.uma.es)

Conexiones a Una Base de Datos

- Las conexiones de bases de datos son vínculos activos a una base de datos que permite leer y escribir datos, y crear objetos SQL en la base de datos.



Servidores Dedicados vs Servidores Compartidos

- ▶ En un servidor dedicado, se crea un proceso servidor por cada petición de conexión.
 - Según vayan aumentando el numero de conexiones el numero de procesos servidor ira aumentando y con ello la reserva de memoria RAM de nuestro sistema.
- ▶ En un servidor compartido, aparece el concepto dispatcher.
 - El dispatcher es el proceso que recibe y gestiona las conexiones.
 - Pasa las transacciones a una cola de procesado.
 - Una vez ejecutadas por el proceso servidor, éste depositara los resultados en una cola de respuesta, siendo el dispatcher el que responda al cliente con el resultado.

Servidores Dedicados vs Servidores Compartidos

- ▶ Usar un servidor dedicado es el método más común para conectar a una base de datos.
- ▶ Es más fácil de configurar y ofrece muchos beneficios cuando se hace debug sobre un proceso.
- ▶ Un servidor compartido posee una configuración más compleja pero puede ayudar a poder aceptar más conexiones usando el mismo hardware.

Servidores Dedicados vs Servidores Compartidos

- ▶ Las ventajas que ofrece un servidor compartido son:
 - Permite un mayor número de conexiones sin tener que aumentar los requerimientos de hardware.
 - Permite desconectar a clientes que llevan mucho tiempo inactivos.
 - Permite balanceo de conexiones entre los diferentes procesos servidor.

- ▶ Las desventajas respecto al servidor dedicado son:
 - Si la cantidad de información que hay que devolver al cliente aumenta, el dispatcher podría llegar a saturarse.
 - Al utilizar un servidor compartido, algunas funciones de administración no están permitidas: startup, shutdown, algunas funciones de recuperación, recreación de índices o análisis de tablas.

Conexiones a un servidor dedicado

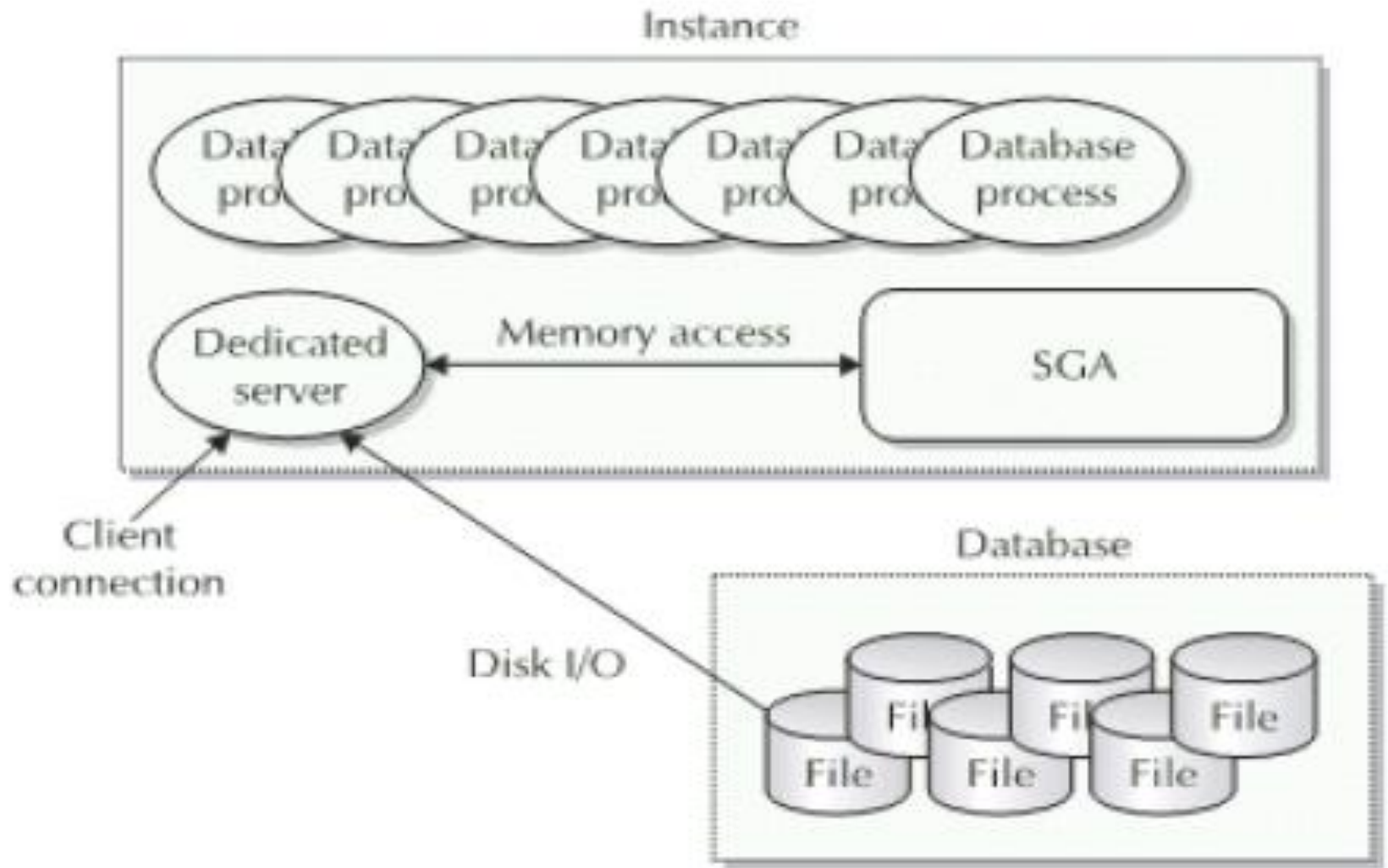
Pasos en una conexión a un servidor dedicado

- ▶ Cuando nos conectamos a un servidor dedicado a través de la red seguimos estos pasos:
 1. Nuestro proceso de cliente se conecta por la red al listener que normalmente está en el servidor de base de datos. Hay situaciones (como en el Oracle Cluster o RAC) en donde el listener no estará en la misma máquina que la instancia de base de datos.
 2. El listener crea un nuevo proceso dedicado (Unix) o solicita a la base de datos la creación de un nuevo thread dedicado (Windows) para nuestra conexión.
 3. El listener traspasa la conexión al nuevo proceso o nuevo thread.
 4. Nuestro cliente envía las peticiones al servidor dedicado el cual las procesa y nos retorna los resultados.

Conexiones a un servidor dedicado

- ▶ La figura de la siguiente transparencia muestra cómo funciona la configuración de servidor dedicado.
- ▶ La instancia de la Base de Datos será un único proceso con threads (Windows) o cada burbuja será un proceso físico separado (Unix).
- ▶ En una configuración de servidor dedicado cada cliente tiene su propio proceso asociado a él.
- ▶ El servidor dedicado recibirá nuestra sentencia SQL y la ejecutará. Leerá los archivos de datos y pondrá datos en cache o mirará en la cache por si ya están los datos, ejecutará sentencias UPDATE y ejecutará nuestro código SQL.
- ▶ Su único objetivo es responder a las llamadas SQL que le enviemos.

Conexiones a un servidor dedicado



Conexiones a un servidor dedicado

- ▶ En el modo servidor dedicado hay un mapeo uno-a-uno entre procesos de la base de datos y procesos de cliente.
- ▶ Cada conexión tiene un servidor dedicado designado exclusivamente para ella.
- ▶ Esto no significa que cada sesión tenga un servidor dedicado (es posible para una aplicación tenga muchas conexiones activas a la base de datos). Sin embargo por normal general cuando se trabaja en modo servidor dedicado hay una relación uno-a-uno entre una sesión y un servidor dedicado.

Conexiones a un servidor dedicado

Ventajas de las conexiones dedicadas

- ▶ Es fácil configurarlo. De hecho no requiere ninguna configuración.
- ▶ Es el modo más rápido de operar en la base de datos al usar el camino más corto.
- ▶ Debido a que los archivos de traza están ligados a un proceso las utilidades como SQL_TRACE se benefician de este modo.
- ▶ Todas las tareas administrativas están disponibles.
- ▶ La memoria para el Área Global de Usuario (UGA), que es una zona de memoria específica por sesión no necesita ser configurada.

Conexiones a un servidor dedicado

Desventajas de las conexiones dedicadas

- ▶ El cliente tiene un proceso/thread consumiendo recursos (memoria, CPU, etc.) del servidor desde que la sesión empieza hasta que termina;
- ▶ Al aumentar el número de usuarios el sistema operativo puede sobrecargarse debido al manejo de sus procesos/threads.
- ▶ Se tiene un control limitado sobre la cantidad de sesiones que pueden estar activas simultáneamente.
- ▶ Si se tiene un ratio muy alto de conexiones/desconexiones a los datos, los procesos de creación y destrucción de procesos/threads pueden ser molestos.

Para la mayoría de sistemas los beneficios sobrepasan a las desventajas. En los servidores convencionales (máquinas con 2 CPU) se necesitarían de 200 a 400 conexiones concurrentes antes de considerar un tipo diferente de conexión. En máquinas con más potencia el límite es mucho más alto.

Conexiones a un servidor Compartido

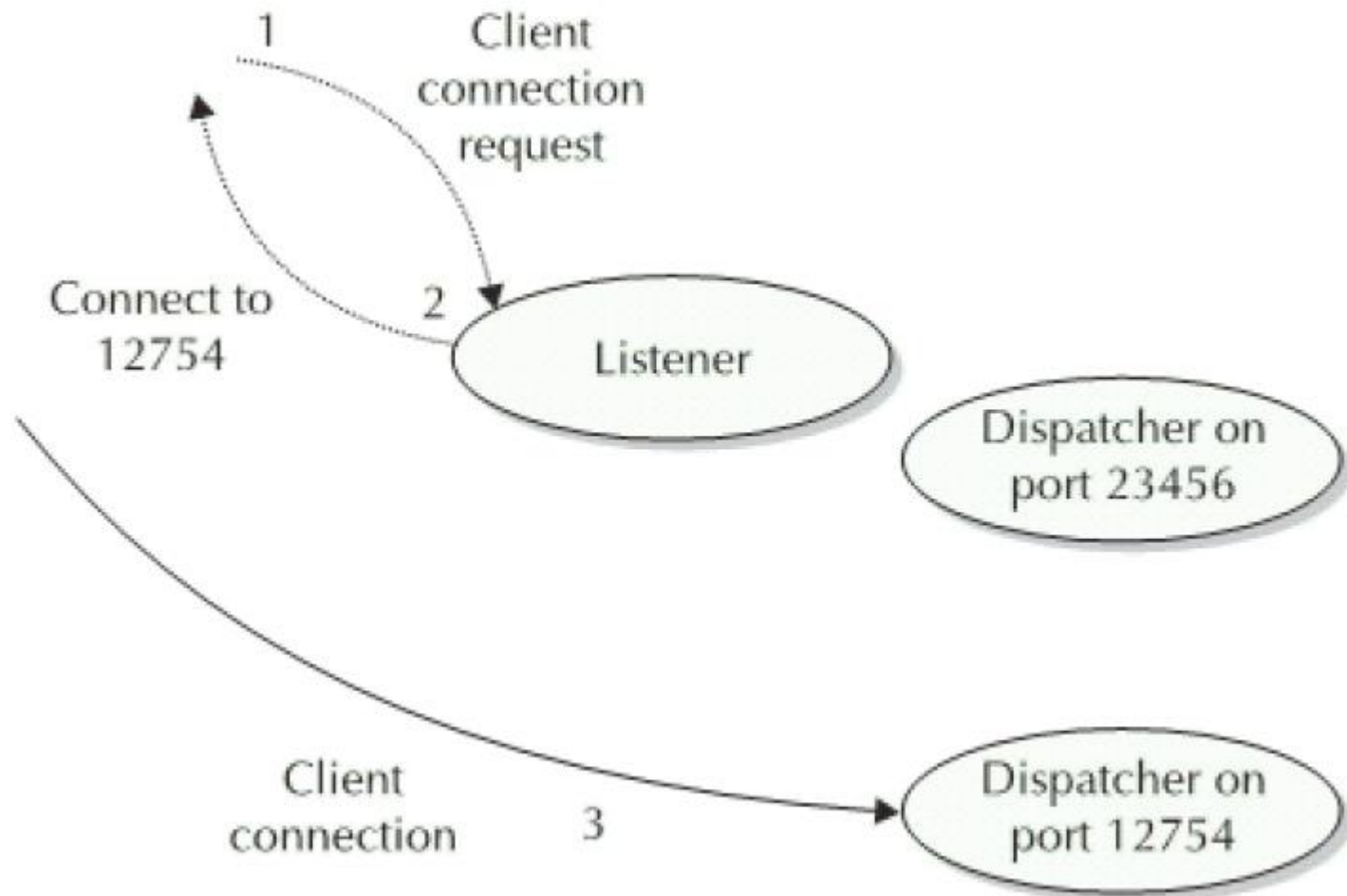
- ▶ El protocolo de conexión para una conexión a un servidor compartido es muy diferente del usado para una configuración de servidor dedicado.
- ▶ En el modo "shared server" no hay un mapeo uno-a-uno entre clientes (sesiones) y procesos/ threads del servidor.
- ▶ Hay un "repositorio" (= pool) llamado "shared servers" que realiza las mismas operaciones que un servidor dedicado, pero las realizan para múltiples sesiones en vez de para una sesión sólo.
- ▶ Los clientes estarán conectados a un "despachador" (dispatchers) durante toda la vida de su sesión.
- ▶ El dispatcher facilitará el manejo de sus peticiones hacia un "shared server" y les retornará la respuesta.
- ▶ Cuando nos conectamos el "listener" nos redirigirá hacia el "dispatcher" disponible.

Conexiones a un servidor Compartido

Pasos en una conexión a un servidor compartido

- ▶ Cuando nos conectamos vía conexiones de servidor compartido se suceden estos pasos:
 1. El proceso cliente se conecta a través de la red al "listener" funcionando en la base de datos. Este "listener" elegirá un "dispatcher" desde el "pool" de "dispatchers" disponibles para que se conecte.
 2. El "listener" retornará la dirección del "dispatcher" elegido para que el cliente se conecte a él, o también podrá ser redirigido directamente al "dispatcher". Esto sucede a nivel TCP/ IP y depende del sistema operativo. Si no hemos sido conectados directamente al "dispatcher" nuestro cliente se desconectará del "listener" y se conectará al "dispatcher".
 3. El proceso de cliente envía su petición al "dispatcher".

Conexiones a un servidor Compartido



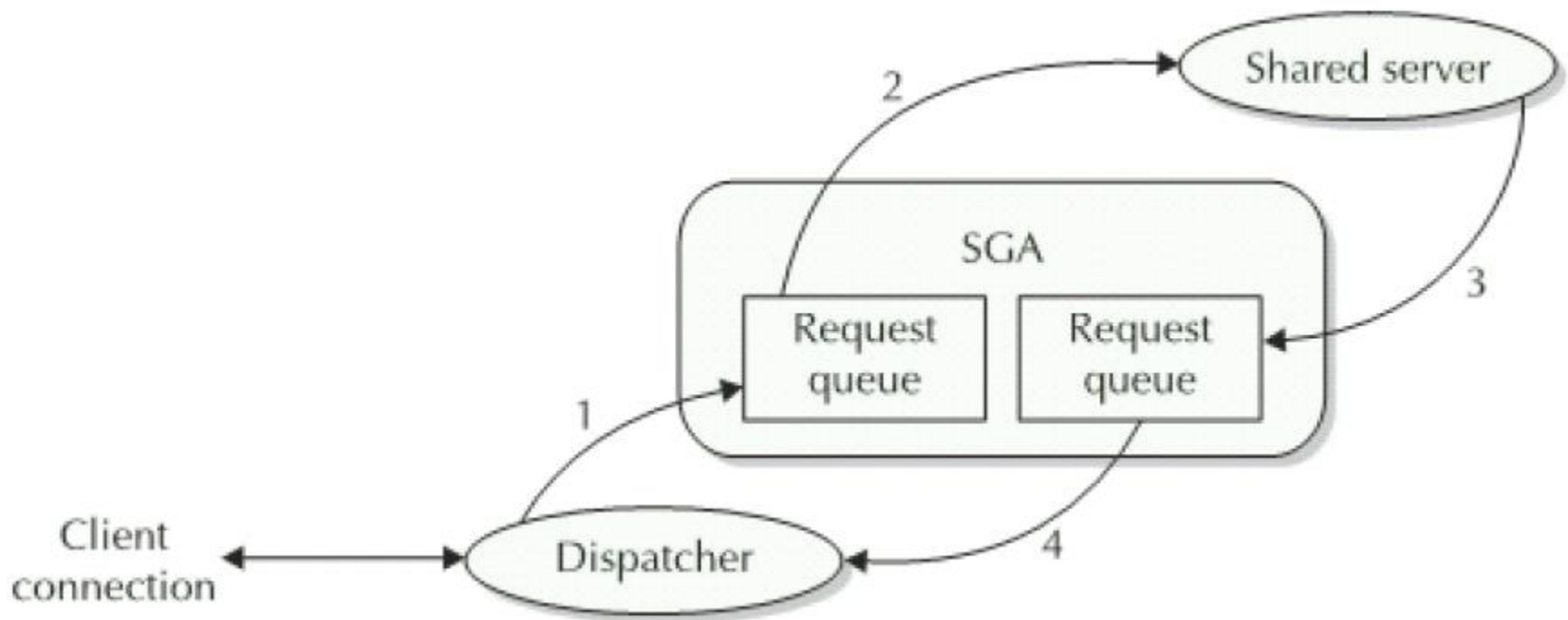
Conexiones a un servidor Compartido

Procesamiento de comandos en un servidor compartido

1. El "dispatcher" coloca la petición en una cola común (compartida por todos los "dispatchers") de la SGA (System Global Area).
2. El primer "shared server" disponible coge y procesa esa petición.
3. El "shared server" coloca la respuesta a esa petición en su cola privada de respuestas de la SGA.
4. El "dispatcher" al que estamos conectados cogerá esa respuesta y nos la retornará.

Conexiones a un servidor Compartido

Procesamiento de comandos en un servidor compartido



Conexiones a un servidor dedicado

Ventajas de las conexiones compartidas

- ▶ Es tan escalable así como necesitemos aumentarlo. Una máquina que pueda físicamente manejar 100 conexiones de servidor dedicado será capaz de manejar 1.000 ó más conexiones a servidor compartido, dependiendo de la aplicación.
- ▶ Notad que este escalamiento se refiere sólo a la CPU (todavía necesitamos memoria suficiente para esas 1.000 conexiones).
- ▶ Permite un fino control sobre cuantas sesiones pueden estar activas de forma concurrente;.
- ▶ Impone un límite "duro" al número total de sesiones activas.

Conexiones a un servidor dedicado

Ventajas de las conexiones compartidas

- ▶ Usa marginalmente menos memoria que la configuración de servidor dedicado.
- ▶ Permite que haya $N+1$ usuarios conectados cuando en un servidor dedicado sería imposible.
- ▶ Si tenemos un alto ratio de conexiones/desconexiones el uso de servidores compartidos puede ser más rápido que servidores dedicados.
- ▶ La creación y destrucción de procesos y threads es muy cara y puede sobrecargar al sistema.

Conexiones a un servidor dedicado

Desventajas de las conexiones compartidas

- ▶ Es más compleja de preparar que la configuración de servidor dedicado, aunque las herramientas GUI proporcionadas con la base de datos la ocultan generalmente.
- ▶ En un sistema donde las conexiones de servidor dedicado trabajan bien, las conexiones de servidor compartido pueden ser más lentas.
- ▶ La ruta hacia un "shared server" es por definición más larga que la ruta hacia un servidor dedicado.
- ▶ Algunas de las características de la base de datos no están disponibles cuando nos conectamos vía "shared server".

Conexiones a un servidor dedicado

Desventajas de las conexiones compartidas

- ▶ Por ejemplo usando conexiones "shared server" no podemos detener o iniciar la instancia,
- ▶ realizar recuperaciones o usar "Log Miner".
- ▶ La memoria total de todas las sesiones concurrentes debe ser computada y configurada como parte del parámetro LARGE_POOL.
- ▶ El traceado (usando SQL_TRACE) no es algo aplicable de forma cómoda a las conexiones "shared server".
- ▶ Los archivos de traza son específicos de un proceso, y en un servidor compartido un proceso no es específico de una sesión.

Conexiones a un servidor dedicado

Desventajas de las conexiones compartidas

- ▶ Existe el peligro de "deadlocks" artificiales.
- ▶ Ocurren porque una vez que un "shared server" inicia el procesamiento de una petición no retornará nada hasta que haya acabado con ella.
- ▶ Por lo tanto si el "shared server" está enviando una petición UPDATE y la fila ya está bloqueada, se bloqueará.
- ▶ Supongamos que la sesión bloqueante estuvo sin hacer nada durante un período de tiempo y el resto de los otros "shared servers" están bloqueados por esta sesión.

Conexiones a un servidor dedicado

Desventajas de las conexiones compartidas

- ▶ Cuando la sesión que mantiene el bloqueo intenta realizar un COMMIT o ROLLBACK (lo cual liberaría los bloqueos), se colgará porque no existen más "shared server" libres.
- ▶ Esta es una situación de "deadlock" artificial donde el bloqueante no puede liberar a los bloqueados porque éstos tienen copados al resto de servidores compartidos.

Conexiones a un servidor dedicado

Desventajas de las conexiones compartidas

- ▶ Hay un riesgo potencial de monopolización del "shared server".
- ▶ Esto ocurre por la misma razón que los "deadlocks" artificiales.
- ▶ Si usamos un "shared server" con transacciones muy largas nuestras sesiones monopolizarán un recurso compartido, previniendo a los otros de usarlo.
- ▶ Si tenemos muchas de estas transacciones largas el rendimiento del sistema caerá de forma abismal porque las transacciones pequeñas deberán esperar a que las largas se completen.

Falsos mitos sobre los servidores dedicados

Los servidores compartidos usan muchísima menos memoria.

- ▶ La memoria necesitada dependerá del número de procesos utilizados.
- ▶ En el caso de un servidor dedicado estará determinada por el número de procesos dedicados y en el caso de un servidor compartido por el número de dispatchers.
- ▶ La configuración "shared server" puede ahorrar memoria pero no tanta como mucha gente cree.

Falsos mitos sobre los servidores dedicados

Se debe usar siempre el servidor compartido con servidores de aplicaciones.

- ▶ Si tenemos una granja inmensa de servidores de aplicación, donde cada uno tiene un gran "pool" de conexiones conectando a una única instancia de base de datos, debemos considerar usar "shared server".
- ▶ Si tenemos un gran número de conexiones y estamos sobrecargando el servidor de base de datos, es mejor usar conexiones "shared server" para reducir el número de procesos físicos.
- ▶ En el caso de que no tenemos una gran cantidad de sesiones y no necesitamos rápidos ratios de conexión/desconexión, lo mejor es usar servidores dedicados.

Falsos mitos sobre los servidores dedicados

Los servidores compartidos sólo son para aplicaciones cliente/ servidor.

- ▶ No se debería usar "shared server" para un servidor de aplicaciones dado que él ya hace un "connection pooling".
- ▶ El servidor de aplicación es sólo un cliente. Tenemos clientes (conexiones del servidor de aplicaciones) y un servidor (la base de datos).
- ▶ Desde la perspectiva de la base de datos todo son conexiones.

Falsos mitos sobre los servidores dedicados

Debemos mirar el número de conexiones concurrentes a la base de datos. Si se generan más procesos de los que el sistema puede manejar las conexiones "shared server" serán la solución.

En caso contrario, lo mejor será usar un servidor dedicado.