

Bases de Datos

Trabajo Práctico N°2

Fecha de Entrega: 19/06/2013

Fecha de Recuperatorio: 17/07/2013

Enunciado

El **Buffer Manager** es uno de los componentes más importantes dentro de un motor de BD. Su principal función es administrar un espacio de memoria de la BD, utilizado como una especie de memoria *caché*. El objetivo es que las diferentes aplicaciones que usan la BD y requieren páginas de disco, puedan recuperar la página de este espacio de memoria y accedan lo menos posible al disco.

El espacio de memoria administrado por el *Buffer Manager* puede ser organizado de diferentes formas y la estrategia para decidir cuál página reemplazar cuando ya no queda más espacio también puede variar.

1. Investigar el manejo de buffers de Oracle

Para esta parte, el objetivo será investigar acerca del manejo de memoria realizado por uno de los motores más usados en la actualidad, *Oracle*. La idea es que analicen su funcionamiento, considerando las posibles ventajas y desventajas de cada enfoque.

Se pretende que estudien las diferentes posibilidades que brinda el motor para distribuir los buffers de memoria y analicen el detalle de cómo funciona.

El estudio debe basarse en la última versión del motor: Oracle 11g. Es importante detallar las referencias de donde extraen la información.

2. Implementar el Catálogo

Aquí deberán incorporar al motor **UBADB** la posibilidad de levantar desde un archivo XML el catálogo de la BD. De esto deberá encargarse el Catalog Manager, el cual cargará este archivo y proveerá a los demás componentes información sobre las tablas.

El catálogo incluirá información de las tablas existentes en la BD, en particular, debe almacenar los siguientes de cada tabla:

- Id de tabla
- Nombre de tabla
- Archivo físico donde se almacena la tabla

Su implementación deberá contener **tests de unidad** (usando *JUnit*).

3. Implementar uso de múltiples Buffer Pools

Actualmente, en UBADB la única organización implementada es la de un único buffer compartido (todas las páginas de las tablas van al mismo Buffer Pool). Sin embargo, otra distribución posible es la que utiliza Oracle (buffers *Keep*, *Recycle* y *Default*).

Para este punto deberán incorporar el uso de múltiples Buffer Pools. La cantidad de pools existentes serán los mencionados por Oracle, pero el código debe contemplar la posibilidad de que más adelante se definan más pools con diferente nombre y tamaño.

Tener en cuenta que el *Buffer Manager* deberá interactuar con el *Catalog Manager*, ya que será este último quien indique a qué pool van las páginas de cada tabla. Además, el tamaño de cada pool se definirá al momento de creación.

Al igual que en el punto anterior, deberán realizar los respectivos *tests de unidad*.

4. Evaluación de múltiples Buffer Pools

Aquí deberán definir trazas que involucren a varias tablas y que muestren escenarios donde el uso múltiples Buffer Pools resulte beneficioso comparado contra un solo Buffer Pool.

La comparación de resultados debe considerar el *hit rate* (# páginas encontradas en memoria / # páginas solicitadas). Los resultados de esos experimentos deberán volcarse en el informe mediante gráficos, presentando además un análisis sobre los resultados y sus conclusiones.

Lo que se espera de los resultados es lo siguiente:

- Que se muestren en gráficos los resultados de la eficiencia (*hit rate*) para buffers pequeños y de mayor capacidad, con distintas trazas usando cada una de las estrategias.
- Una tabla comparativa de eficiencia de las distintas estrategias en los distintos contextos.

Para definir cada escenario pueden hacer uso de una herramienta para generar trazas dentro de *UBADB*.

Consideraciones

Para la corrección no sólo se evaluará que lo implementado funcione correctamente sino también la calidad del código generado. Con calidad nos referimos a usar comentarios, nombre declarativo para las variables o métodos, uso de métodos auxiliares, evitar duplicar código, etc. La idea es seguir utilizando este código en futuros cuatrimestres, por lo que es importante que traten de seguir las convenciones utilizadas a lo largo del código.

Entrega

La entrega deberá contar con el código que implementa lo pedido, un breve informe con:

- Detalles de implementación
- Decisiones tomadas
- Conclusiones

En los laboratorios, se llevará a cabo una demo con su tutor asignado.

Bibliografía

- Libro “Database Management Systems”, Raghu Ramakrishnan – Johannes Gehrke, 2º Edición, Capítulo 7.
- Paper “Principles of Database Buffer Management”, Effelsberg & Haerder, 1984.
- Paper “All About Oracle’s Touch Count Algorithm”, Shallahamer, 2004.