



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

19 de Junio de 2013

Bases de Datos

Integrante	LU	Correo electrónico
Agustina Ciraco	630/06	agusciraco@gmail.com
Nadia Heredia	589/08	heredianadia@gmail.com
Pablo Antonio	290/08	pabloa@gmail.com
Vanesa Stricker	159/09	vanesastricker@gmail.com

Índice

1. Introducción	3
1.1. Formato	3
2. Múltiples Buffers	4
3. Detalles de Implementación	4
4. Resultados	5
4.1. Trazas utilizadas	5
5. Conclusiones	6
6. Bibliografía	7

1. Introducción

En este trabajo práctico, analizaremos el impacto de las diferentes estrategias para manejar los buffer pools. Nos concentraremos en el módulo Buffer Manager.

Las diferentes estrategias que consideraremos son

- un solo buffer
- múltiples buffers (distribución de Oracle)

Veremos las ventajas y desventajas de cada uno según la situación.

1.1. Formato

En primera instancia, presentaremos una breve explicación del uso de los múltiples buffer, luego se explicarán ciertos detalles de implementación, así como también las decisiones tomadas durante el desarrollo.

Por último, daremos nuestras conclusiones.

2. Múltiples Buffers

En esta sección se explica el funcionamiento de los múltiples buffer y su finalidad.

Oracle en su documentación no revela los detalles de su funcionamiento interno. Sin embargo existen ciertas publicaciones al respecto por parte de personas que trabajaron en esos proyectos, y también por parte de quienes analizaron el código fuente. Basamos nuestro análisis en la bibliografía mencionada en la sección correspondiente.

Puede pasar que bloques de distinta naturaleza interfieran entre sí, por lo cual en Oracle, se decidió utilizar *múltiples buffers*. Existen tres pools fijos

- **Keep pool** Utilizado para objetos pequeños que siempre deberían estar en memoria, ya que son muy frecuentemente usados.
- **Recycle pool** Utilizado para objetos poco usados que no se desea que interfieran con el resto. En general son objetos grandes, y no se quiere que estos ocupen lugar en el pool default.
- **Default pool** Utilizado para todo lo demás.

3. Detalles de Implementación

En esta sección se explican algunos detalles de las implementaciones de los algoritmos.

Se deseó implementar un Buffer Pool que delegara el almacenamiento de cada página en distintos pools. Esto se logró con `MultipleBufferPool`, que mantiene una lista de `BufferPoolAssignment`, cada uno de los cuales *acepta* algunas páginas y otras no. Para aquellas que acepta, les asigna un buffer pool.

En nuestro caso particular utilizamos un pool por tabla, por lo cual tenemos `TableToBufferPoolAssignment` que acepta todas las páginas de una tabla en particular. Tenemos también una subclase de `MultipleBufferPool` (`PerTableMultipleBufferPool`) que recibe una lista de definiciones de pools (`TablePool`); es decir, las tablas para las que creará un pool, el porcentaje del tamaño total que asignará a cada tabla y la estrategia de reemplazo de cada pool.

4. Resultados

En esta sección presentaremos los resultados obtenidos al utilizar diferentes trazas.

Se presenta para cada traza el hit rate de cada buffer.

4.1. Trazas utilizadas

5. Conclusiones

En esta sección presentaremos nuestras observaciones sobre las comparaciones explicadas en la sección anterior.

Al dividir el espacio de memoria en distintos pools, uno por tabla, el rendimiento disminuyó.

El rendimiento de cada estrategia depende fuertemente de la estructura de la traza, y probablemente estos resultados se deban a la naturaleza artificial de las utilizadas. Por ejemplo, en general se supone que una estrategia LRU es razonable porque luego de acceder un bloque es probable que esto vuelva a suceder.

Sin embargo, al armar una traza al azar o siguiendo algún criterio arbitrario, esto no se verifica.

Sería interesante, por lo tanto, realizar las pruebas con trazas más reales, y con mayor cantidad, que correspondan a contextos variados, analizando si por ejemplo se verifica la cercanía temporal de los pedidos.

En esos caso, sería de esperar que las estrategias utilizadas por Oracle fueran eficientes.

6. Bibliografía

- **Oracle Docs** Oracle Database Concepts
<http://docs.oracle.com/cd/E11882.01/server.112/e25789/toc.htm>
- **Oracle Docs** Oracle Database Performance Tuning Guide
<http://docs.oracle.com/cd/E11882.01/server.112/e16638/toc.htm>
- **Reference to Oracle Technologies** Keep Buffer Pool and Recycle Buffer Pool
<http://www.exploreoracle.com/2009/04/02/keep-buffer-pool-and-recycle-buffer-pool/>
- **Praetorate Oracle Support** Create Multiple Data Buffer Pools
http://www.praetorate.com/t_oracle_net_data_buffer_pools.htm