An Overview of Buffer Pool Management in DB2

James Teng, Ph.D.
Distinguished Engineer
IBM Silicon Valley Laboratory
Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The opinions expressed are those of the presenter at the time, not necessarily the current opinion and certainly not that of the company.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries: AIX, AS/400, DATABASE 2, DB2*, Enterprise Storage Server, ESCON*, IBM, iSeries, Lotus, NOTES, OS/400, pSeries, RISC, WebSphere, xSeries, z/Architecture, z/OS, zSeries, System p, System i, System z

The following terms are trademarks or registered trademarks of the Microsoft Corporation in the United States and/or other countries: MICROSOFT, WINDOWS, ODBC

For more copyright & trademark information see ibm.com/legal/copytrade.phtml
Agenda

- The Role of Buffer Manager
- Multiple Buffer Pools
- Buffer Search Hash Table
- Buffer Replacement Methods (LRU/MRU/FIFO)
- Sequential/List/Dynamic Prefetch
- Deferred Writes
- Page Latch
- ALTER/DISPLAY Buffer Pool commands
- Goal Oriented Buffer Pool Management
Key DB2 Components

- Utility
- Data/Index Manager
- Buffer Manager
- RDS
- DB2 Data
- Catalog Directory
Buffer Pools

- Multiple Buffer Pools
  - 4K Page Size: BP0, BP1, ..., BP49
  - 8K Page Size: BP8K0, BP8K1, ..., BP8K9
  - 16K Page Size: BP16K0, BP16K1, ..., BP16K9
  - 32K Page Size: BP32K, BP32K1, ..., BP32K9

- Data buffering is used to minimize disk I/Os

- BPs are created in DB2's DBM1
  - Created at first Data Set Open
  - Deleted when all referenced data sets are closed

- Database and BP association is done via
  - CREATE/ALTER TABLESPACE
  - CREATE/ALTER INDEX
Large Buffer Pools

- **Objectives:**
  - Exploit large main memories
  - Cache DB data in memory for high performance
  - Support multiple BPs
  - 4K, 8K, 16K, 32K page size

- **64-bit BPs up to 1TB**
  - Exploits 64-bit virtual
Simple Update Tx Flow

Lock Page P1 X
GetPage P1
  BP Miss - Read P1 from disk
  BP Hit - Use Page
Latch P1 X
Set-Write P1
Write update log record
Unlatch P1
RelPage P1
Commit
  Release locks
Buffer Search Hashing Table

- Use hashing to locate a requested page
  - Hash Table Anchor Points
  - Search time independent of buffer pool size
- Use multiple latches to serialize buffer search
LRU

- Use the least recently used (LRU) algorithm to reuse buffers

Least Recently Used

D

A

E

Y

Reference Page A (RelPage)

D

E

Y

A

Reference Page B (RelPage)

E

Y

A

B

© 2010 IBM Corporation
IBM Software Group  |  DB2 Data Management Software
MRU

- Apply most recently used (MRU) logic to data that are unlikely to be re-referenced
  - Data from LOAD/REORG/RECOVER
  - Workfile data referenced during MERGE phase of SORT

Least Recently Used

Reference Page E
(RefPage)

Reference Page Y
(RefPage)
FIFO - First In First Out

- PGSTEAL(FIFO) option on the ALTER BPOOL command
- Steal the buffer containing the oldest page read from DASD
- Don't need to move buffers on each reference
- Good for in-memory tablespaces/indexes
  - Avoid CPU overhead and Latch contention to manage the LRU chain
**Buffer Pool Caching Priority**

- **Problem:** Prevent sequentially accessed data from monopolizing buffer pool

- **Solution:**
  - Limit the number of buffers for SEQUENTIAL data
    - VPSEQT - Sequential Steal Threshold (0 to 100%)
    - Use Sequential-LRU chain to enforce the limit

---

**Diagram:**

- BPn
  - VPSEQT
  - Random Data
  - Sequential and Random Data
Multiple LRU Chains

- Multiple subpools are created for
  - A Large Bufferpool
  - Threshold is controlled by DB2
- LRU is managed within each subpools
  - Reduce BP latch contention when
    - Degree of Parallelism is high
- Round-robin subpools when stealing buffers

BPn

![Diagram of Multiple LRU Chains]

- Green
- Purple
- Red
Sequential Prefetch

- Benefit: Improve I/O efficiency when accessing data or index in sequential pattern
- Used for table and clustered index scans, Sort, and Utilities
- Prefetch occurs asynchronously
- Greatly speeds up queries and batch jobs
  - Increase I/O efficiency by batching I/Os

```
Get Page
32 33 34 35 36 37 38 39 . . .
Prefetch
63-95
```
List Prefetch

▪ A data page access method usable by index scans
  f Single index access
    – Index is less clustered
    – Result set is not too small
▪ Multiple index access path
  f Index ANDing or Index ORing
▪ List Prefetch Operation
  f Index scanned in usual way
  f Qualifying RIDs extracted
  f RIDs sorted in RID sequence
  f Access to data pages is via sorted RID list
# Sequential Detect

- **Benefit**: Improve I/O efficiency when accessing data or index in sequential or skip-sequential pattern

- **How does it work?**
  - Monitor data and index access at run time
  - Trigger Dynamic Prefetch when pages are accessed in sequential or skip-sequential pattern

**Get Page**

```
2 4 5 8 12 16 18 20 . . .
```

**Prefetch**

```
20 - 35
```
Deferred Writes

- Each VDWQ is managed in LRU (Least Recently Updated)
- If VDWQT=0, write 32 LRU pages when the number of changed pages reaches 40
Writing Data

- Writing is triggered by
  - System Checkpoint
  - A high count of updated pages for a data set
  - A low percentage of AVAILABLE buffers
  - An infrequently updated page found on top of the LRU chain
- Before closing a data set
- Write I/Os are handled by DB2 system tasks
  - Pages are sorted to improve I/O efficiency
  - Up to 32 pages for each write I/O
  - All pages on a single I/O are within a disk cylinder
Benefits of Deferred Writes

- Increase the probability of batching I/Os
- Minimize the number of write I/Os for frequently updated pages
- Deferred Write reduces I/O per row updated
- Batched updateds increase I/O efficiency
- Maximize I/O concurrency by scheduling multiple "write engines"
  - CPU time is charged as SRB time in DBM1
  - Running under zIIP in DB2 10
Page Latching

• Page Latches serialize physical changes to pages
  • With Row Level Locking, we can't depend on page locking for this anymore
  • Page latches are inexpensive
    • No Deadlock Detection
  • S and X latches are supported
  • Reader - Hold S-Latch
  • Updater - Hold X-Latch
  • Write engine – Acquire S-latch to quiesce updaters
  • Held for very short duration (while object is accessed)
  • Locks ensure logical consistency (committedness)
64-bit Buffer Pools

- Max BP size – up to 1TB

- PGFIX = YES option to long-term page fix buffers in real storage (i.e. virtual = real)
  - Use where I/O rate is high
  - Must have real storage available to back the pool
  - Up to 10% CPU saving

- PGFIX = NO (which is the default)
  - Needs to do page fix/free for each I/O or each GBP operation
ALTER BUFFERPOOL Command

- **VPSIZE**
  - Online to alter Buffer Pool size
  - PGSTEAL – LRU or FIFO
    - NONE – option for in-memory tables/indexes (DB2 10)
- **VPSEQT** – default 80%
  - Threshold used to prevent prefetch data from monopolizing BP
    - = 0 – disable prefetch
- **DWQT/VDWQT** – Deferred Write thresholds
- **VPPSEQT/VPXPSEQT** – limit buffers used by prefetch done by parallel queries
DISPLAY BUFFERPOOL Command

- DISPLAY BPOOL LSTATS allows users to monitor BP and I/O activities at the dataset level
- DBNAME and SPACENAM keywords
  - Allow users to limit the dataset level BP and I/O activities for a specified set of tablespaces/indexes
    - DBNAME: dbnames, name1:name2, name*, *
    - SPACENAM: spacenames, name1:name2, name*, *
- IFCD 199 records to report dataset level BP Statistics and I/O activities
  - Average and Maximum I/O response time
  - Number of Pages and CHANGED pages in BP
  - Defined in Statistics Class 8 and Monitor Class 1
DISPLAY BUFFERPOOL Command ...

DSNB450I - TABLESPACE = DSNDB07.DSNTMP03, USE COUNT = 6, GBP-DEP = N

DSNB452I - STATISTICS FOR DATASET 1 -

DSNB453I - VP CACHED PAGES -
  CURRENT = 2 MAX = 7847
  CHANGED = 1 MAX = 1510

DSNB455I - SYNCHRONOUS I/O DELAYS -
  AVERAGE DELAY = 3 MAXIMUM DELAY = 6
  TOTAL PAGES = 36198

DSNB456I - ASYNCHRONOUS I/O DELAYS -
  AVERAGE DELAY = 2 MAXIMUM DELAY = 60
  TOTAL PAGES = 37988 TOTAL I/O COUNT = 2883
Automatic buffer pool management

- Only the size attribute of the buffer pool.
- Can be enabled or disabled at the individual buffer pool level.

Automatic management entails the following:

- DB2 Registers the BPOOL with WLM
- DB2 provides sizing information to WLM
- DB2 communicates to WLM each time allied agents encounter delays
- DB2 periodically reports BPOOL size and random read hit ratios to WLM
DB2 Registers BPOOL to WLM

**IWM4MREG Service**

- **Trigger**
  - `ALTER BPOOL AUTOSIZE(YES)`
  - BPOOL allocation
  - Automatic management set ON (DB2 deregisters when deallocated or altered OFF)

<table>
<thead>
<tr>
<th>Name</th>
<th>BP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curren t</td>
<td>800MB</td>
</tr>
<tr>
<td>Min</td>
<td>600MB</td>
</tr>
<tr>
<td>Max</td>
<td>1GB</td>
</tr>
</tbody>
</table>
Periodic reporting

Data Collection exit
(one for each pool)

DB2

BP0  BP1  BP2  BP7

DB2 Periodic Report

Buffer Pool Sizes
Hit Ratio for Random Reads

WLM

1 Plots size and hit ratio over time.
2 Projects effects of changing the size
Buffer Pool adjusting

- If the buffer pool is adjusted, the result will be just as though an ALTER BUFFERPOOL VPSIZE command had been issued.
  - The new size is stored by DB2 in the BSDS.
- If the buffer pool is deallocated (e.g. because DB2 is being stopped) it will subsequently be reallocated at its most recently allocated size.

Example
  - If BPOOL is adjusted from 800 MB to 900 MB
  - Then DB2 is stopped and restarted
  - BPOOL will be subsequently allocated at 900 MB