DB2 Data Management Software

IBM

e business software

Chapter 9: Storage Management

IBM DB2 Universal Database V8.1

Database Administration Certification Preparation Course

Maintained by Clara Liu

IBM Software Group

Objectives

- In this section, we will cover:
  - DB2 Process Model
  - DB2 Memory Model
  - DB2 Table Spaces
  - Managing Table Spaces
  - Performance Considerations

DB2 Data Management Software

IBM

e business software

Chapter 9: Storage Management

DB2 Process Model

DB2 Memory Model

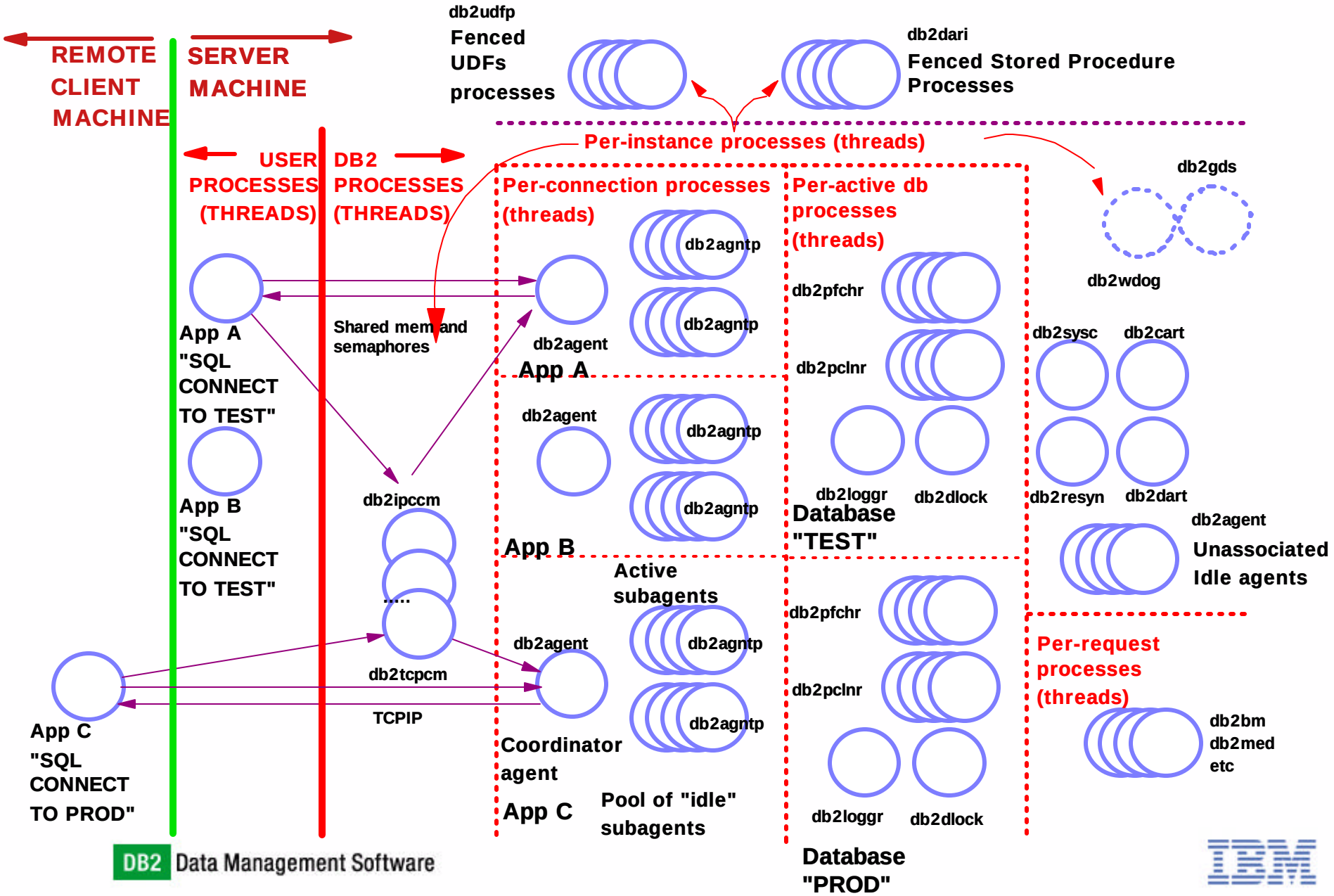
DB2 Table Spaces

Managing Table Spaces

Performance Considerations

IBM Software Group

DB2 Process Model



DB2 Process Model

- Connection-level processes:
  - db2agent - coordinator agent performs database requests on behalf of clients
  - db2agntp - subagents receive requests from coordinator agent
- Database-level processes:
  - db2tcpcm, db2snacm, db2ipccm, et al. - communication listeners that provide connection support for local and remote clients
  - db2pfchr - I/O prefetchers
  - db2pclnr - buffer pool page cleaners
  - db2loggr - manipulates log files to handle transaction processing and recovery
  - db2dlock - deadlock detector
- Instance-level processes:
  - db2sysc - system controller
  - db2resyn - resync agent that scans the global resync list
  - db2gds - global daemon spawner to start new processes (UNIX-only)
  - db2wdog - watchdog process to handle abnormal terminations (UNIX-only)
  - db2udf - fenced UDFs run outside of DB2's address space
  - db2dari - fenced stored procedures run outside of DB2's address space

DB2 Agents

- Communication between the database manager and client and client applications
  - Unix: processes
  - Intel: threads
- Processes/threads
  - Logical agent
    - Represents a connected application to the database manager
    - Contains information and control blocks required by an application
    - Controled by MAX_LOGIAGENTS DBM configuration parameter
  - Worker agent
    - Carries out application requests
    - No permanent attachment to any particular application
    - Contains all information and control blocks require to complete actions
    - 4 types of worker agents
      - Active coordinator agents
      - Subagents
      - Inactive agents
      - Idle agents

DB2 Data Management Software

IBM

e business software

Chapter 9: Storage Management

DB2 Process Model

DB2 Memory Model

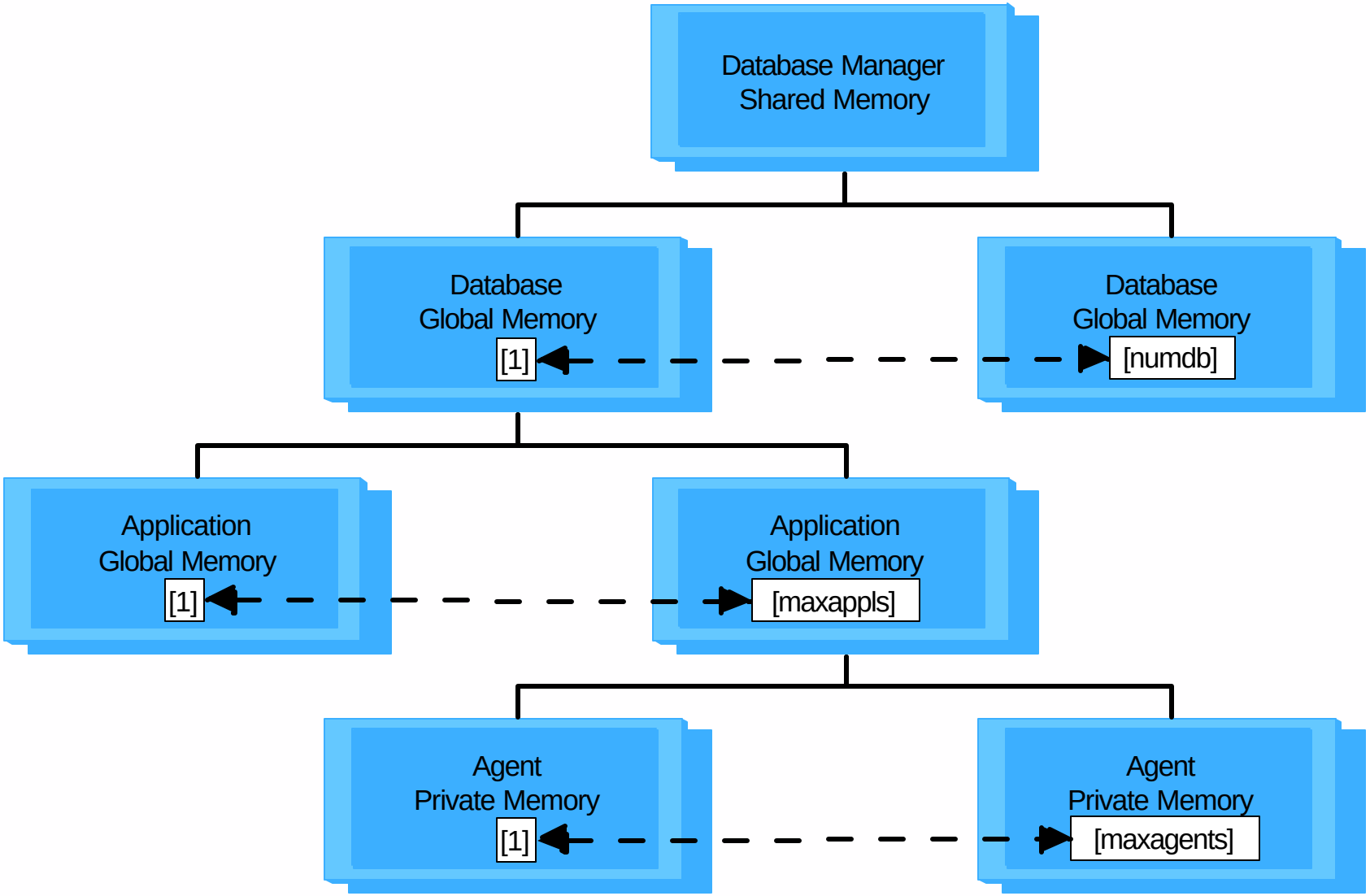
DB2 Table Spaces

Managing Table Spaces

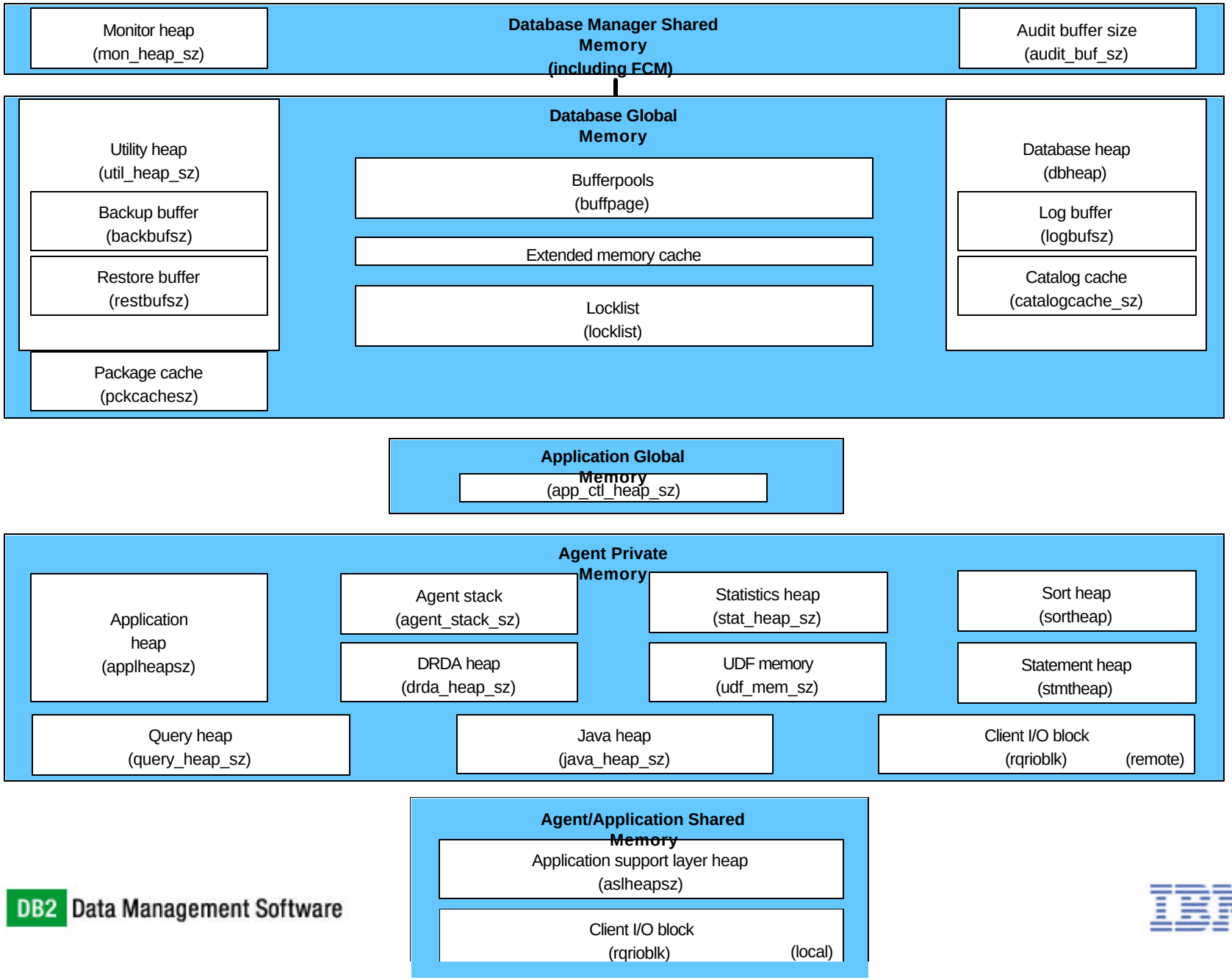
Performance Considerations

IBM Software Group

DB2 Memory Model



DB2 Memory Model



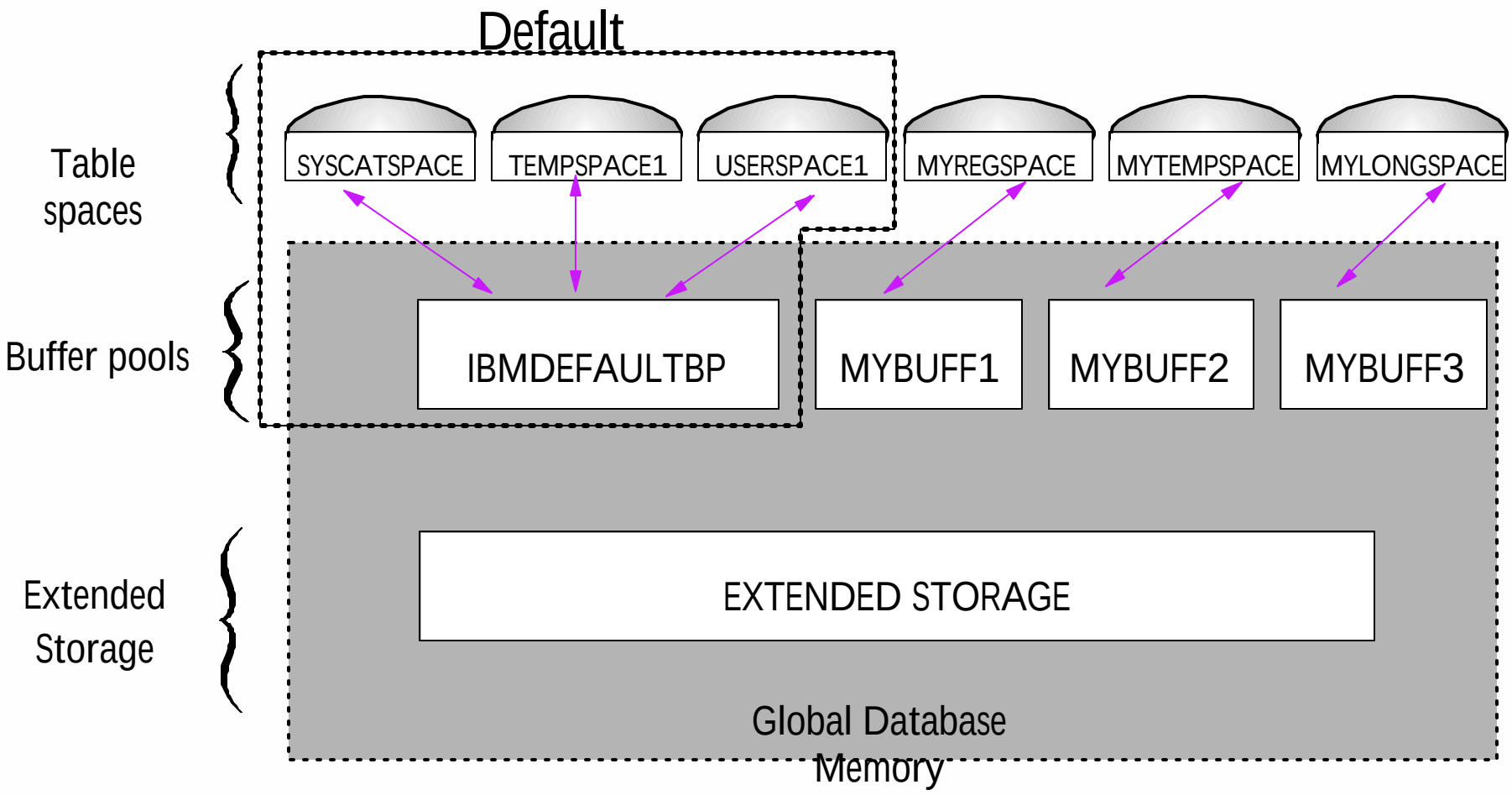
DB2 Shared Memory Sets

- Database Manager Shared Memory Set
  - Stores all relevant information for a particular instance, such as lists of all active connections and security information
- Database Shared Memory Set
  - Stores information relevant to a particular database, such as package caches, log buffers, and bufferpools
- Application Shared Memory Set
  - Stores information that is shared between DB2 and a particular application, primarily rows of data being passed to or from the database
- Agent Private Memory Set
  - Stores information that is used by DB2 to service a particular application, such as sort heaps, cursor information, and session contexts

Buffer Pools

- Used to buffer data in memory to reduce the number of I/O operations to the physical database
- Keep often requested data/index pages in memory
- Keep infrequently accessed tables (e.g. random access into very large table) out of main memory
- Most data manipulation takes place in buffer pools, except for large objects and long field data
- IBMDEFAULTBP is the default bufferpool created with every database
- Ability to keep large number of pages in extended storage cache

Buffer Pool Overview



Management of Buffer Pools

- Memory for the buffer pool is allocated when the database is activated or when the first application connects to the database
- Command to create buffer pool

  CREATE BUFFERPOOL bpname IMMEDIATE SIZE sz PAGESIZE pgsz

  CREATE BUFFERPOOL bpname DEFERRED SIZE sz PAGESIZE pgsz
- IMMEDIATE
  - The buffer pool will be created immediately. If there is not enough reserved space in the database shared memory to allocate the new buffer pool, a warning (SQLSTATE 01657) is returned, and the statement is executed DEFERRED
- DEFERRED
  - The buffer pool will be created when the database is reactivated (all applications need to be disconnected from the database)
- Information stored in SYSIBM.SYSBUFFERPOOLS

Management of Buffer Pools

- Use the ALTER BUFFERPOOL command to increase the size of the buffer pool, memory is allocated as soon as the command is committed if the memory is available.
- If the memory is not available, the changed occurs when all applications are disconnected and the database is reactivated.
- If the size of the buffer pool is decreased, memory is deallocated at commit time
- Alter size of a buffer pool

  ALTER BUFFERPOOL bpname IMMEDIATE SIZE sz

  ALTER BUFFERPOOL bpname DEFERRED SIZE sz

DB2 Data Management Software

IBM

e business software

Chapter 9: Storage Management

DB2 Process Model

DB2 Memory Model

DB2 Table Spaces

Managing Table Spaces

Performance Considerations

IBM Software Group

Table Spaces

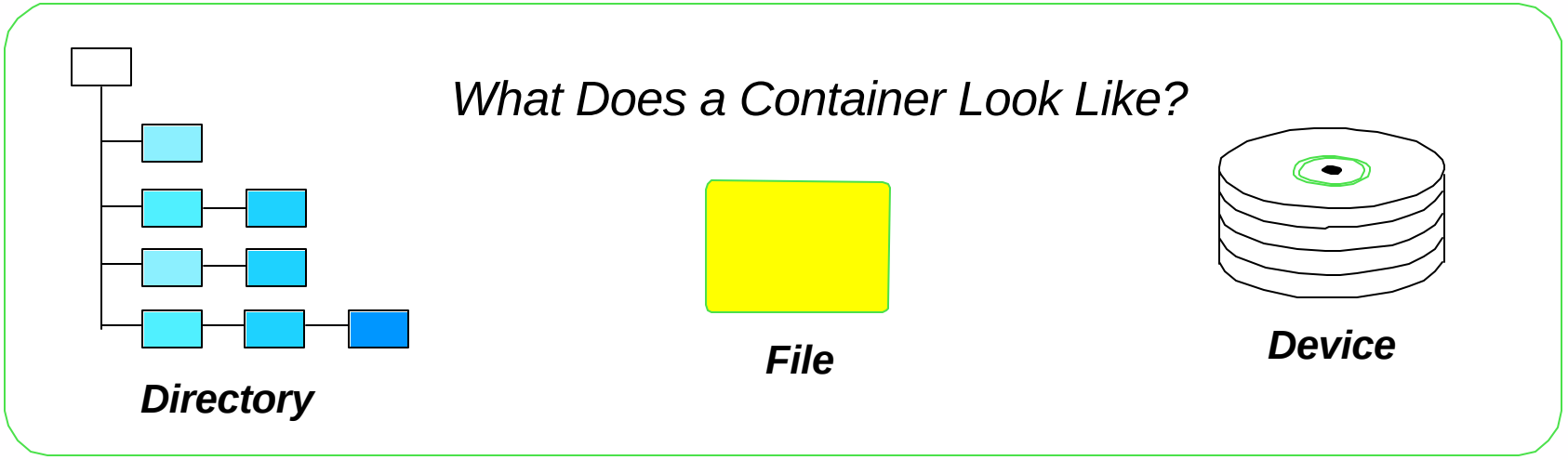
- All database objects are stored within table spaces
- Three types of table space:
  - REGULAR
  - LARGE
  - TEMPORARY
- REGULAR table space stores all data except for temporary tables
- LARGE table space stores long or LOB data, it must be a DMS table space
- TEMPORARY table space, two types:
  - SYSTEM TEMPORARY table space
    - A work area used by the database manager to perform operations such as sorts or joins
    - A database must have at least one SYSTEM TEMPORARY table space
    - A default SYSTEM TEMPORARY table space is created at database creation time
  - USER TEMPORARY table space
    - Stores declared global temporary tables
    - No user temporary tablespaces exist when a database is created

Table Spaces

- Two types of storage:
  - System Managed Space (SMS)
  - Database Managed Space (DMS)
- Table spaces are either 4K, 8K, 16K or 32K pages, 4K is default size
- Cannot mix page sizes within a table space
- Must be associated with a bufferpool in same page size
- Table space composed of one or more containers
- Data allocated by extents within containers
- Three table spaces created by default (all SMS)
  - SYSCATSPACE - system catalog tables
  - TEMPSPACE1 - temporary data
  - USERSPACE1 - default user data

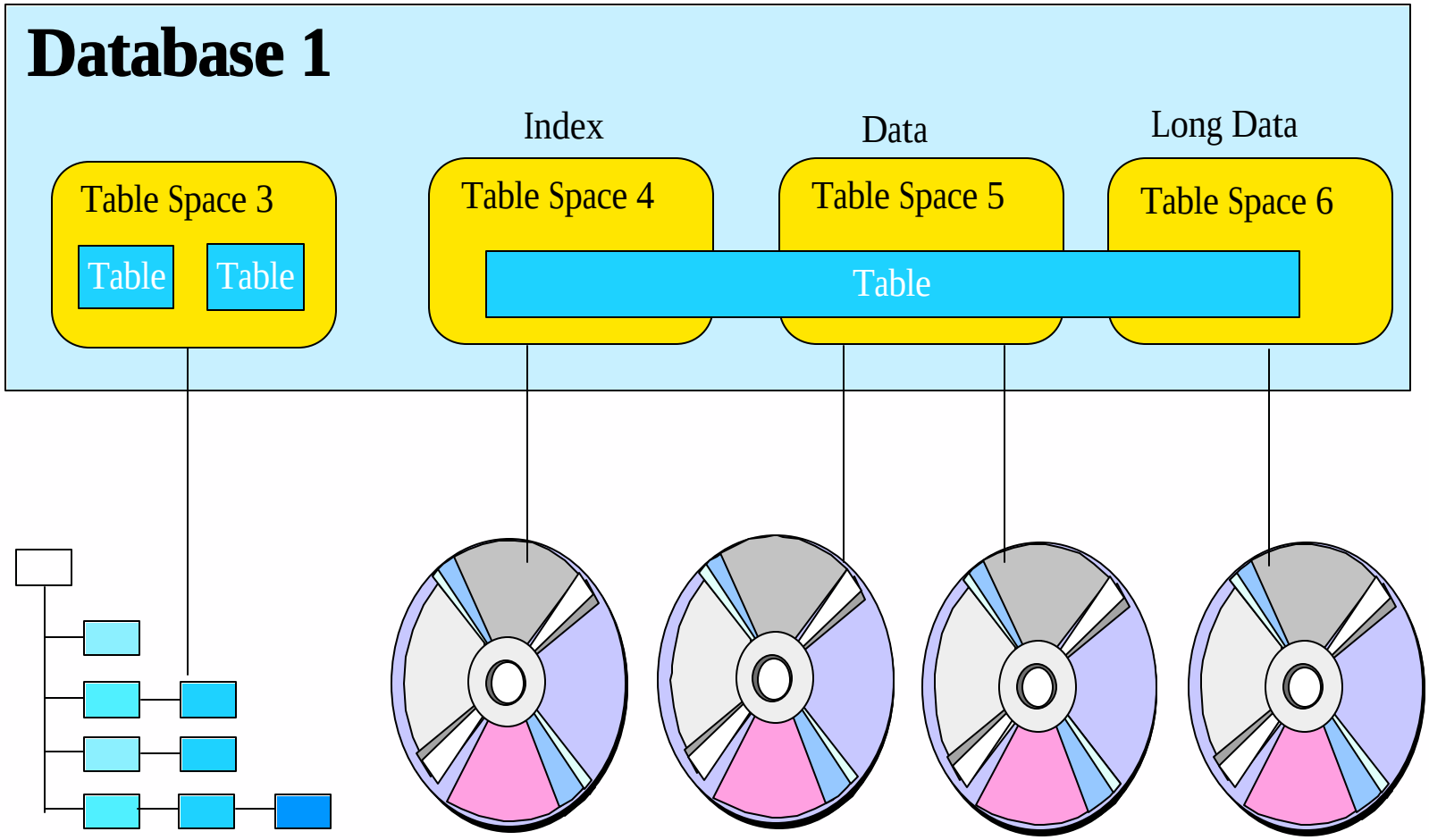
Containers

- Container is an Allocation of Physical Space



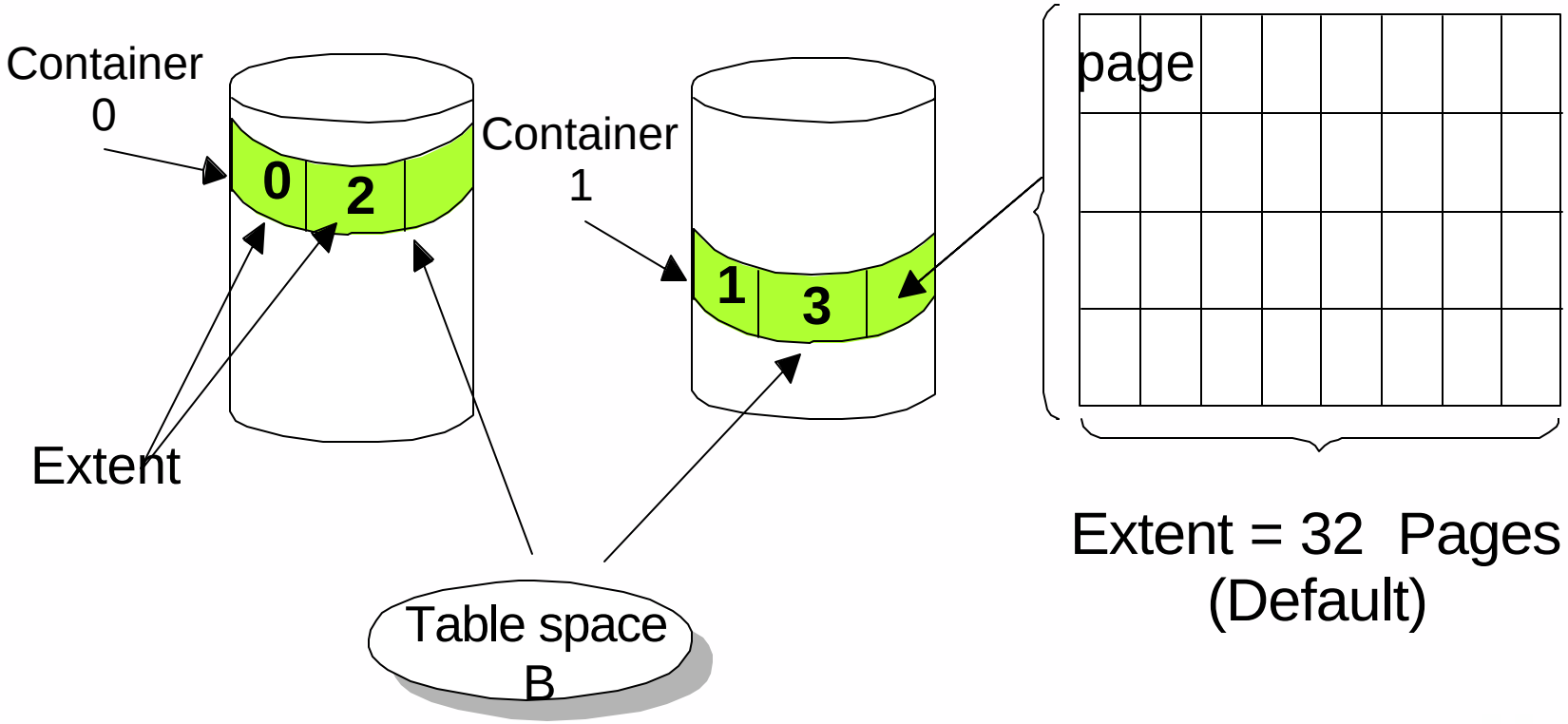
	File	Directory	Device
Intel	DMS	SMS	DMS
UNIX	DMS	SMS	DMS
Windows NT	DMS	SMS	DMS

Table Spaces and Containers



Containers and Extents

- DFT_EXTENT_SZ defined at database level
- EXTENTSIZE defined per table space
- Once a table space is created, EXTENTSIZE cannot be changed
- An extent consists of multiple pages
- Data written to containers based on the extent map

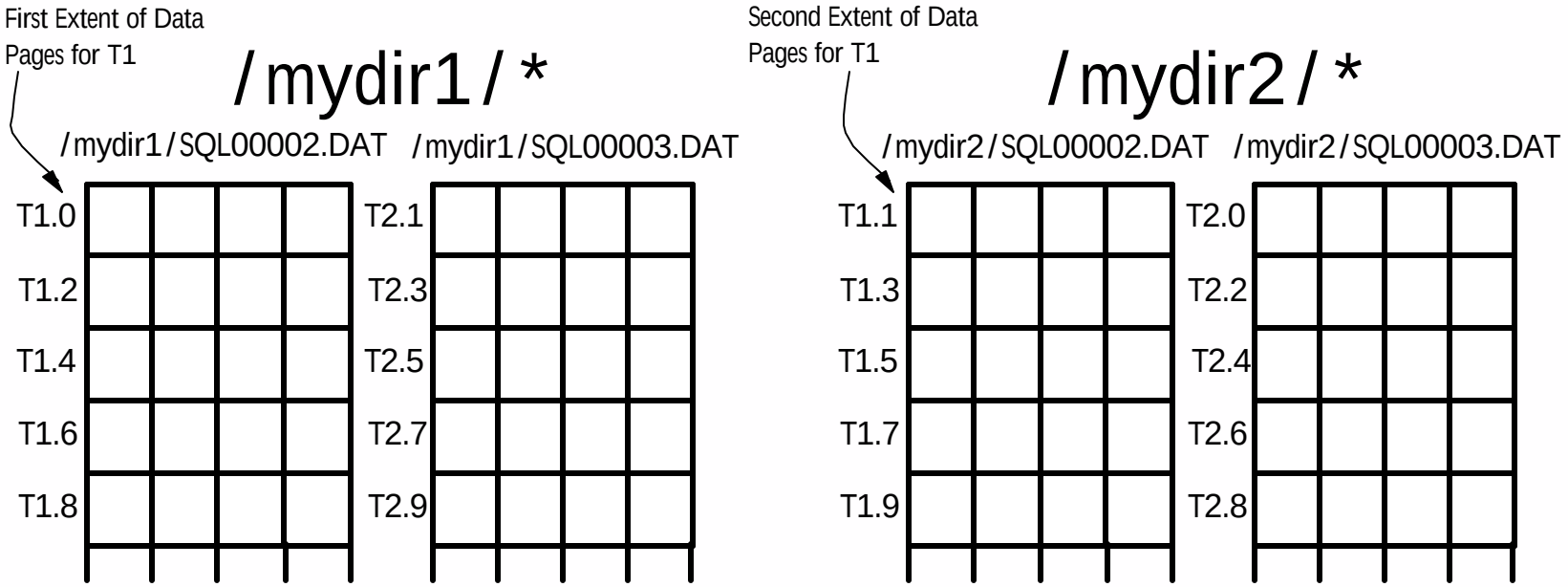


SMS Characteristics

- All table data and indexes share the same table space
- Each table in a table space is given its own file name used by all containers
  - The file extension denotes the type of the data stored in the file
- Dynamic file growth
- Upper boundary on size governed by:
  - Number of containers
  - Operating System limit on size of file system
  - Operating System limit on size of individual files
- If table space has more than one container, they are suggested to be in the same size (or close to same size)
  - When all space in a single container is allocated, the table space is considered full even if space remains in other containers
- New containers can only be added to SMS on a partition that does not yet have any containers
- UNIX: file system size may be increased
- Very easy to administer
- Recommended for TEMP table space

SMS Table Spaces

- What happens on disk during the following ?
  - CREATE TABLESPACE ts1
    MANAGED BY SYSTEM USING ('/mydir1', '/mydir2')
    EXTENTSIZE 4
  - CREATE TABLE t1 (c1 INT ...) IN ts1
  - CREATE TABLE t2 (c2 FLOAT ...) IN ts1



SMS Table Spaces

- Containers are operating system directories
  - Can increase table space capacity by enlarging underlying OS file system
- Data striped across container by extent
- Disk space allocated *on demand*
  - Allocate one page at a time (default)
  - Use *db2empfa* utility to enable multiple page allocation
  - Once *db2empfa* is run, the *multipage_alloc* database configuration parameter is set to YES
- Database objects (e.g. table data, indexes, large objects) are located by operating system file names
- Data, index, and large object data of a table must reside in the same table space

Associate each container (i.e. directory) with a different file system
... otherwise table space capacity limited to that of a single file system

Ensure containers have equal capacity (roughly)
... excess in larger containers isn't exploited

DMS Characteristics

- Space allocated at creation time
- Containers can be added or dropped (data is rebalanced)
- Automatic rebalancing
- Container size can be extended, reduced, or resized
- Capacity limited only by physical storage
- File system I/O used for DMS-file manipulation
- Direct I/O used for DMS-raw manipulation
- High performance potential (especially for OLTP)
- Flexible data placement
- Can split table objects (i.e. data, index, long field data) into different table spaces

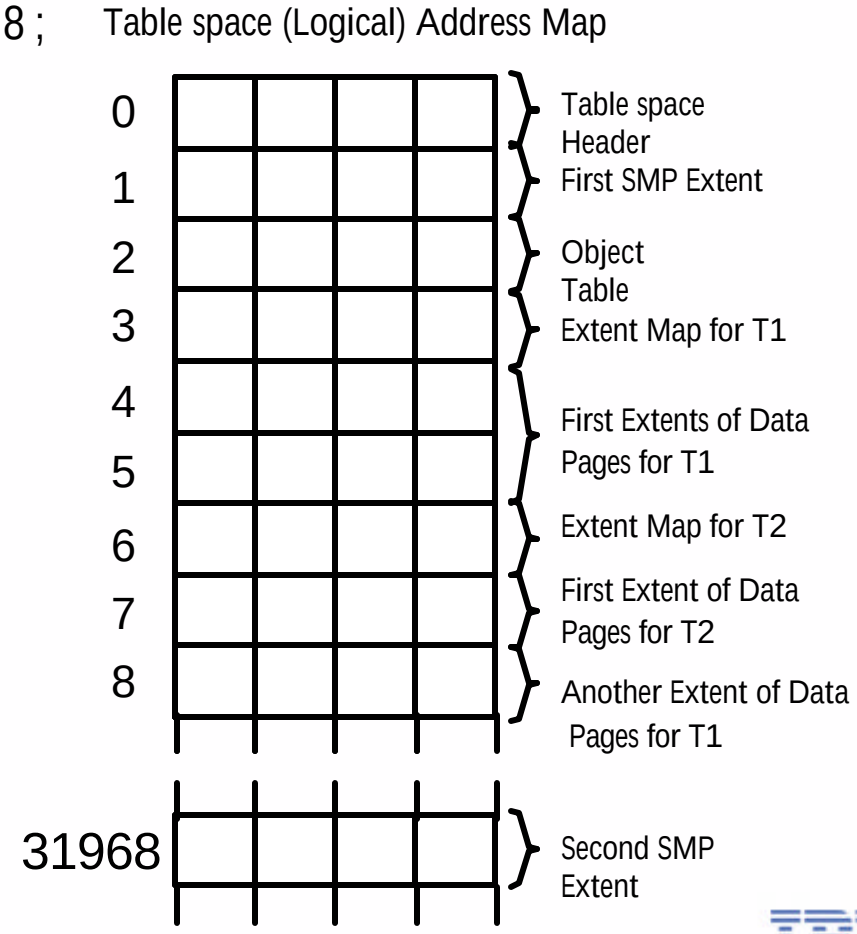
DMS Table Spaces

- Containers are either operating system files or raw devices
- Data striped across containers by extent
- Disk space allocated at table space creation
  - Space Map Pages (SMP) keep track of what extents are used and which are free
- Data and database objects are located by
  - Object table locates first extent in the object
  - Extent Map Pages (EMPs) for the object locates other extents in the object

Associate each container with a different disk(s)
... enables parallel I/O, larger table space capacity

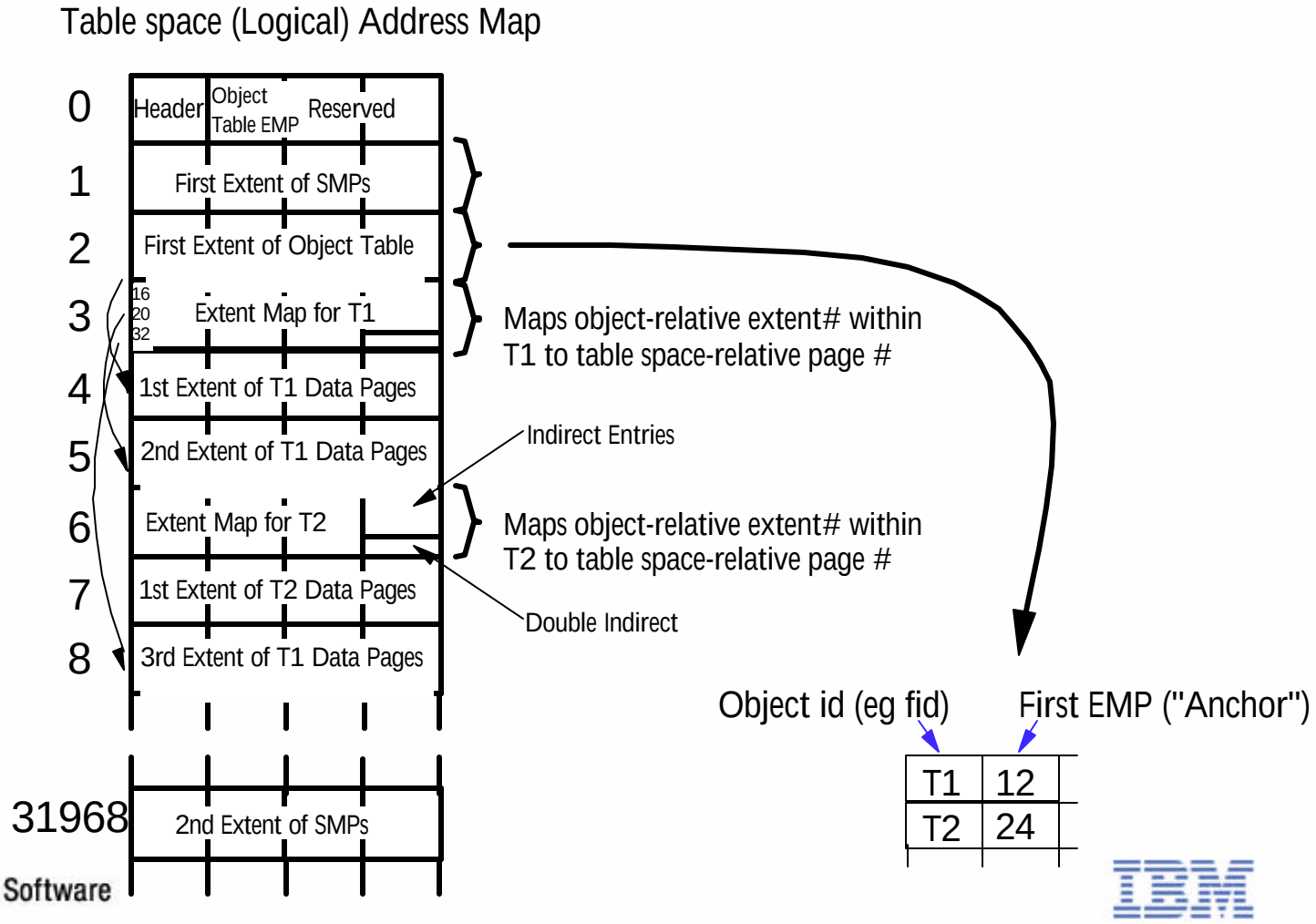
DMS Table Spaces

- What happens on disk during the following?
  - CREATE TABLESPACE ts2
    MANAGED BY DATABASE USING
    (FILE '/myfile' 1024, DEVICE '/dev/rhd7' 2048)
    EXTENTSIZE 4 PREFETCHSIZE 8 ;
  - CREATE TABLE t1
    (c1 INT ...) IN ts2 ;
  - CREATE TABLE t2
    (c1 FLOAT ...) IN ts2 ;



DMS Extent Maps

- Definition: A meta-data structure stored within a table space that records the allocation of extents to each object (table, index, etc.) in the table space
- Allocated an extent at a time



SMS versus DMS

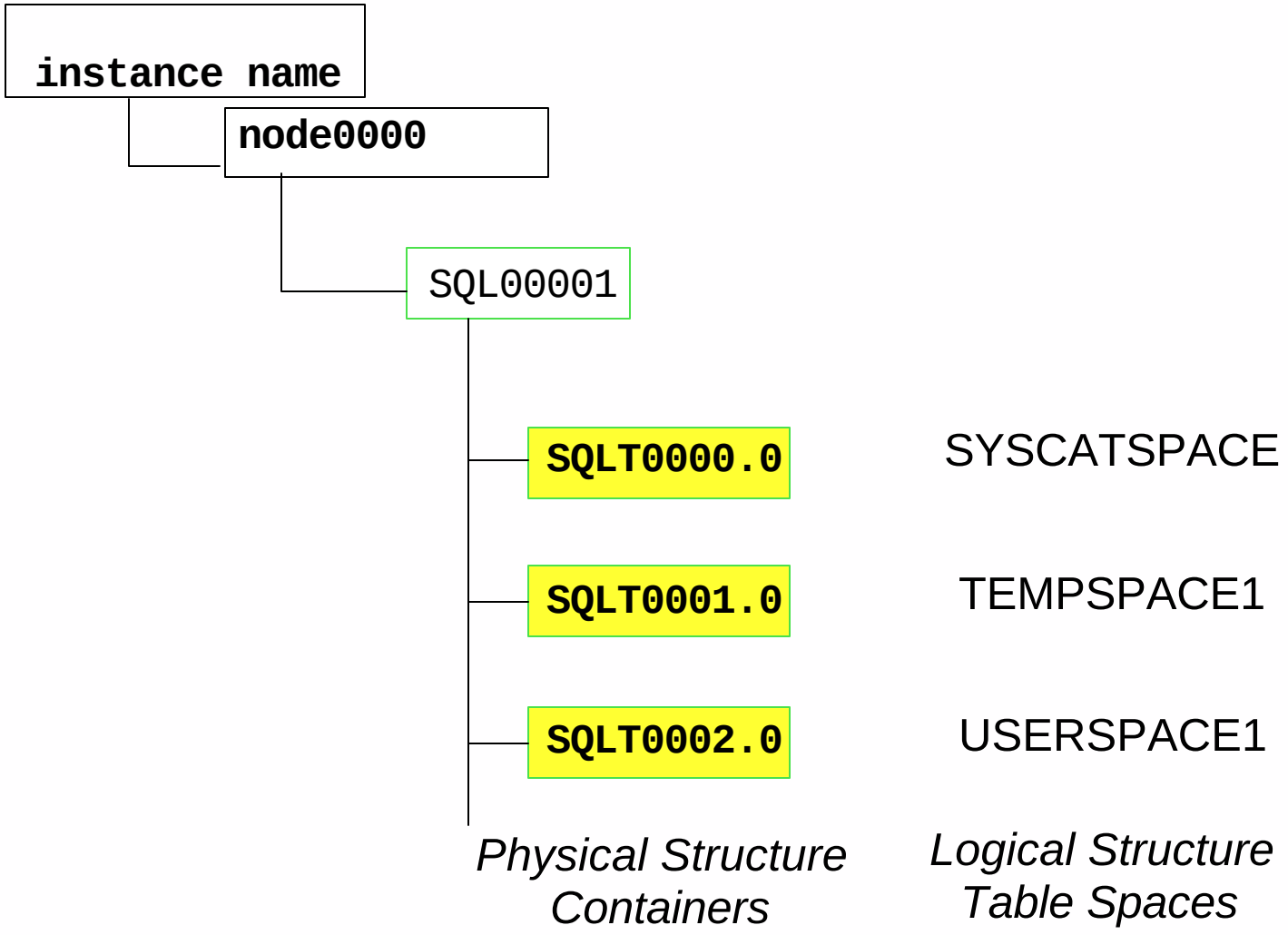
	SMS	DMS
Striping	**Yes**	**Yes**
Object Management	**Operating system** (via unique file names)	**DB2** (Object table and EMP extents)
Space Allocation	**Grows/shrinks on demand**	**Preallocated**
Ease Of Administration	**Best** . Little/no tuning required (e.g.. OS prefetching often very good) . Enlarge file system(s) associated with containers	**Good** . Some tuning required (e.g.. EXTENTSIZE PREFETCHSIZE) . Can enlarge table space via ALTER TABLESPACE ADD CONTAINER
Performance	**Very Good**	**Best** . Can achieve up to 5-10% advantage with raw containers. . Index, LOBs, Data for a single table can be spanned across table spaces.

- Table spaces can be renamed (does not include SYSCATSPACE)

  NOTE: renaming does update the minimum recovery time

Default Database Configuration

CREATE DATABASE ourdb ON path/drive



CREATE DATABASE Example

- CREATE DATABASE hrdb ON C
  USING CODESET codeset
  TERRITORY territory
  COLLATE USING SYSTEM
  CATALOG TABLESPACE
    MANAGED BY SYSTEM USING (' d:\db2\cattbsp')
    EXTENTSIZE 16
  USER TABLESPACE
    MANAGED BY DATABASE
    USING (FILE 'd:\db2\user.f1' 30M, FILE 'd:\db2\user.f2' 30M)
    EXTENTSIZE 64
    PREFETCHSIZE 128
  TEMPORARY TABLESPACE
    MANAGED BY DATABASE USING (DEVICE 'd:\db2\temp.f1' 10M) ;

Database Codeset, Territory, and Collating Sequence

- Codeset and Territory
  - Code set is mapped to the DB2 code page
  - Cannot be changed once the database is created
  - Refer to section 'Supported territory codes and code pages' in the DB2 Admin Guide
- Collating Sequence
  - Identifies the type of collating sequence to be used for the database
  - Cannot be changed once the database is created
  - Five types:
    - COMPATIBILITY
      - DB2 Version 2 collating sequence for back level support
    - IDENTITY
      - Strings are compared byte for byte
    - IDENTITY_16BIT
      - CESU-8 (Compatibility Encoding Scheme for UTF-16: 8-Bit) collation sequence
      - Can only be used when creating a Unicode database
    - NLSCHAR
      - Collating sequence based on the specific codeset and territory
    - SYSTEM
      - Collating sequence based on the current territory

CREATE TABLESPACE Examples

- CREATE TABLESPACE enterprise
  PAGESIZE 8K
  MANAGED BY SYSTEM
  USING ('/database/firstcnt'), ('/database/secondcnt'), ('/database/thirdcnt')
  EXTENTSIZE 16K
  PREFETCHSIZE 32
  BUFFERPOOL BP8K ;

- CREATE USER TEMPORARY TABLESPACE usertemp
  MANAGED BY DATABASE
  USING (DEVICE '/dev/rusrtmp1' 10M,DEVICE '/dev/rusrtmp2' 10M)
  OVERHEAD 24.1 TRANSFERRATE 0.9 ;

- CREATE LARGE TABLESPACE lobtbsp
  PAGESIZE 16K
  MANAGED BY DATABASE
  USING (DEVICE '/dev/rdb2lob1' 1000, DEVICE '/dev/rdb2lob2' 1000)
  BUFFERPOOL BP16K ;

DB2 Data Management Software

IBM

e business software

Chapter 9: Storage Management

DB2 Process Model

DB2 Memory Model

DB2 Table Spaces

Managing Table Spaces

Performance Considerations

IBM Software Group

Managing Table Spaces

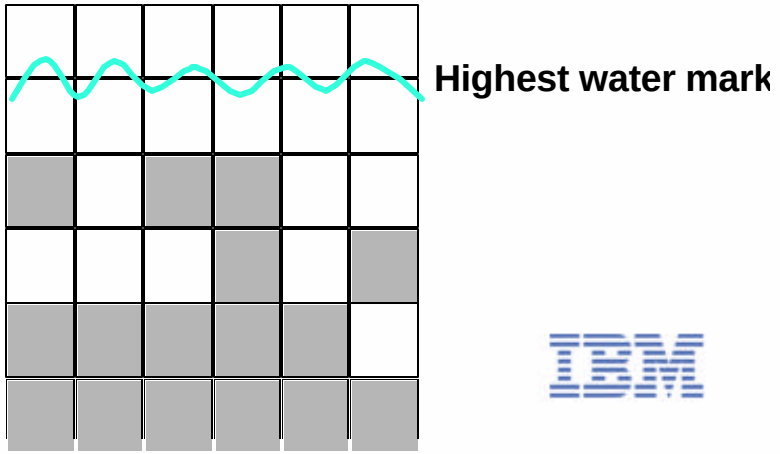
- List tablespaces [show detail]
  - Lists all table spaces in the database with optional detail on storage use
  - List status of table spaces
- List tablespace containers
  - Lists all containers for a table space
- Rename tablespace
  - Renaming a table space to a new name
  - Minimum recovery time of the table space is updated when the rename took place
  - **RENAME TABLESPACE userspace1 TO data2000 ;**
- Drop tablespace
  - Any dependent objects are deleted or marked as inoperative
    - All tables, indexes, keys (primary & foreign) and constraints are dropped
    - Views, Triggers & Packages are marked invalid
    - All catalog entries are removed
  - Table space will not be dropped if there is any table that spans on multiple table spaces, the table must be dropped first
  - Table space will not be dropped if there is any table that has the **RESTRICT ON DROP** attribute

Altering Table Space

- Modify PREFETCHSIZE, BUFFERPOOL, OVERHEAD, TRANSFERRATE
- Enable or disable DROPPED TABLE RECOVERY
- Switch table space ONLINE
- SMS Table Space
  - Add a container to the table space on a partition that currently has no containers
- DMS Table Space
  - Add or drop containers to or from the table space, examples:
    - ALTER TABLESPACE ts0
      ADD (FILE 'cont2' 2000, FILE 'cont3' 2000)
      ADD (FILE 'cont4' 2000) ;
    - ALTER TABLESPACE ts0
      DROP (FILE 'cont1' 2000) ;
  - Table space size can be extended, reduced, resized, examples:
    - ALTER TABLESPACE ts0
      EXTEND (FILE 'cont0' 100)
      RESIZE (FILE 'cont1' 3000) ;
    - ALTER TABLESPACE ts0
      REDUCE (ALL CONTAINERS 100) ;

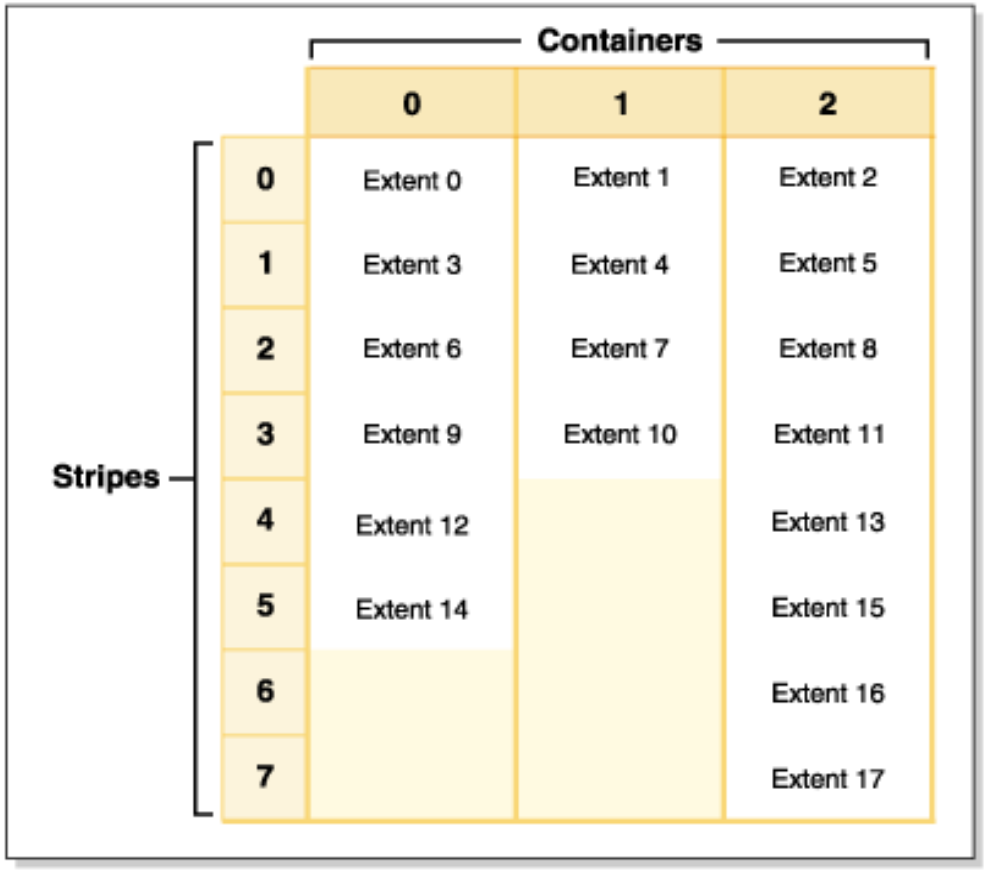
Add and Extend Container Size of a DMS Table Space

- When new containers are added to a table space or existing containers are extended, a rebalance of the table space data may occur.
- Access to the table space is not restricted during rebalancing (objects can be dropped, created, populated, and queried as usual) but there will be a significant impact on performance.
- The process of rebalancing when adding or extending containers involves moving table space extents from one location to another, and it is done in an attempt to keep data striped within the table space.
- The rebalancer starts at extent 0, moving one extent at a time until the extent holding the high-water mark has been moved.
- High-water mark is the page number of the highest allocated page in the table space, it can be obtained from the **LIST TABLESPACES SHOW DETAIL** command.
- If space is added above the high-water mark, rebalance will not occur.



Add and Extend Container of DMS Table Space - Example

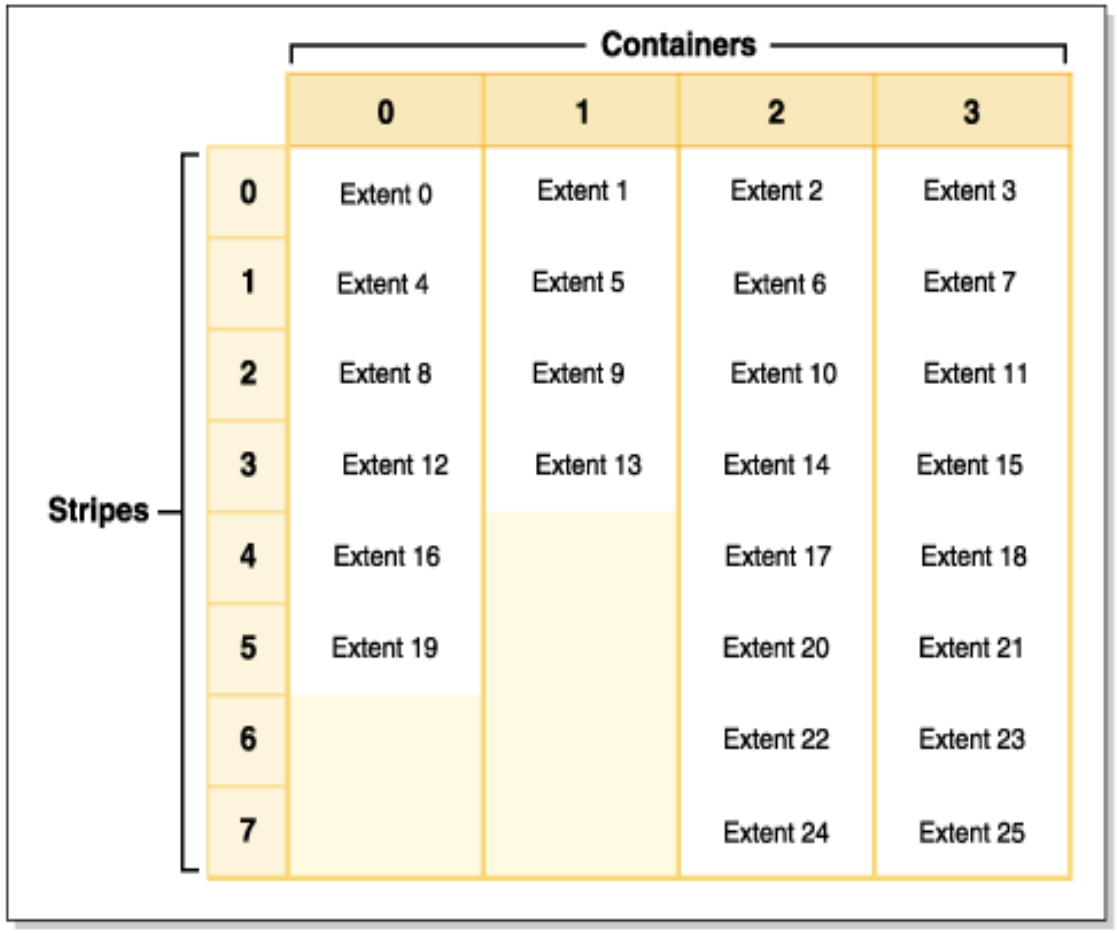
- Example 1: A DMS table space with three containers, extent size of 10, and the containers are 60, 40, and 80 pages respectively. Assume the high-water mark is at Extent 14.



- Stripes 0 to 7 belongs to stripe set 0

Add and Extend Container of DMS Table Space - Example

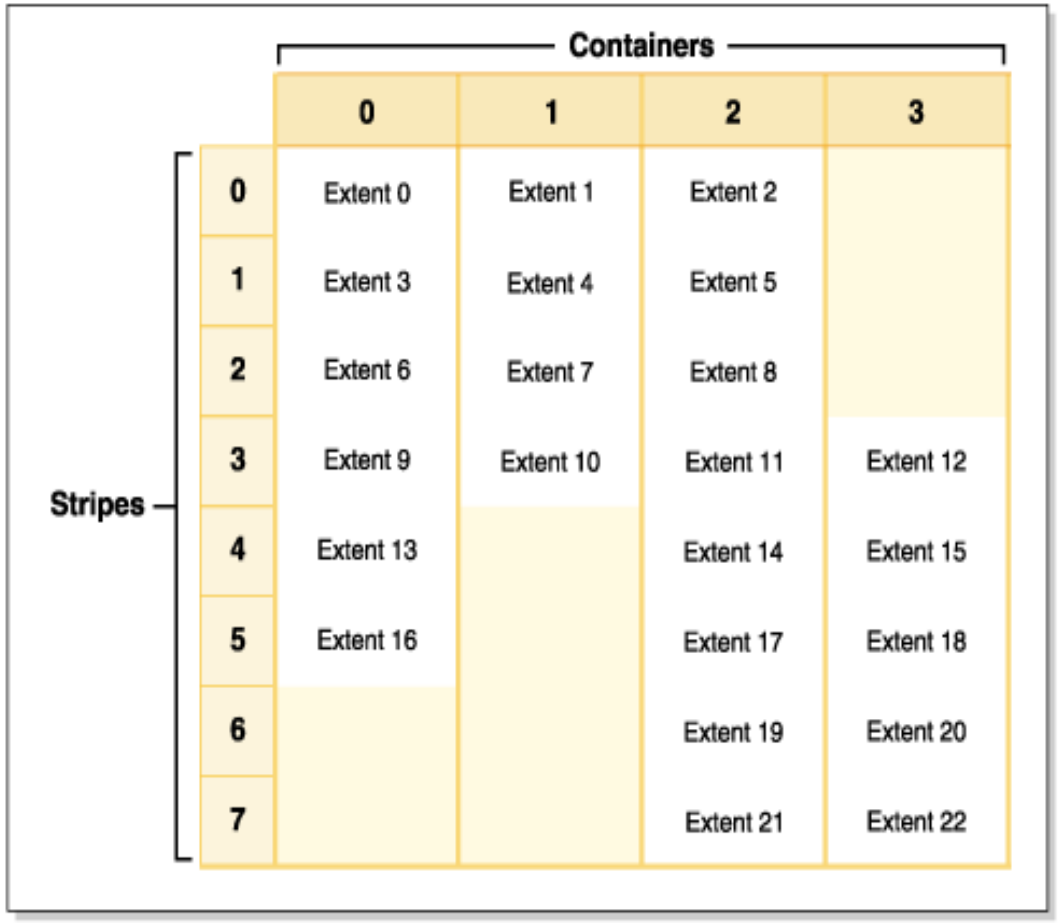
- Example 2: An 80-page container is added to the table space, the container is large enough to start in the first stripe (stripe 0) and end in the last stripe (stripe 7).



- Most of the extents are relocated, data was rebalanced
- Stripes 0 to 7 belongs to stripe set 0

Add and Extend Container of DMS Table Space - Example

- Example 3: A 50-page container is added to Example 1. The container is not large enough to start in the first stripe (stripe 0) and end (stripe 7), it is then positioned such that it ends at the last stripe.



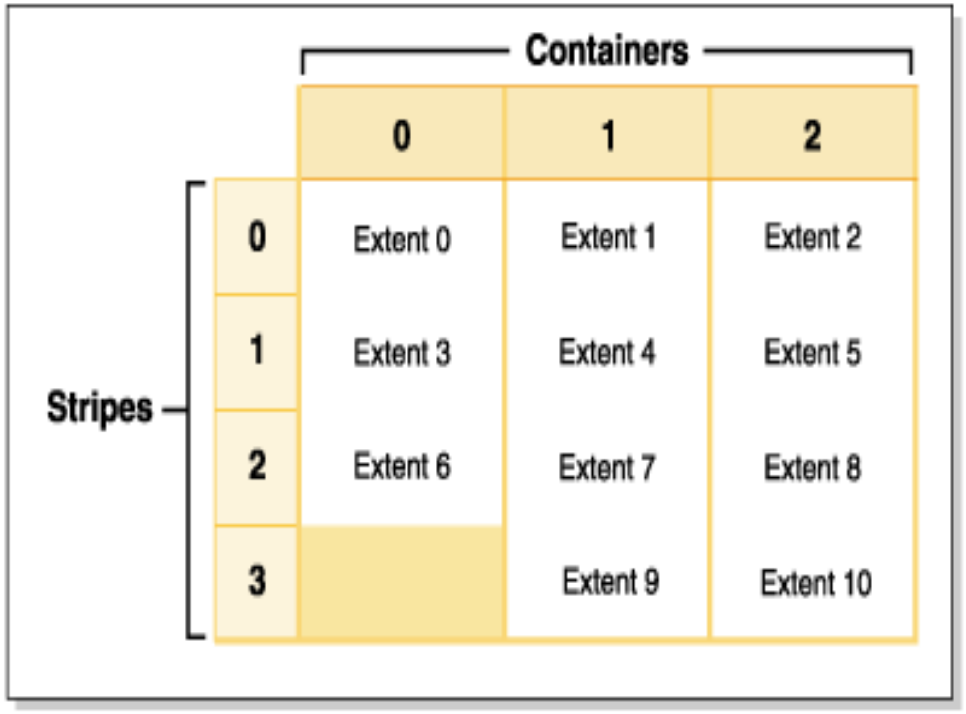
- Some of the extents are relocated, data was rebalanced
- Stripes 0 to 7 belongs to stripe set 0

Adding New Stripe Set

- Adding a container will almost always add space below the high-water mark, meaning rebalance is often necessary.
- There is an option to force new containers to be added above the high-water mark by adding a new *stripe set.*
- The existing containers in the existing stripe sets remain untouched, the new containers become part of a new stripe set
- New containers will be available for immediate use

Adding New Stripe Set - Example

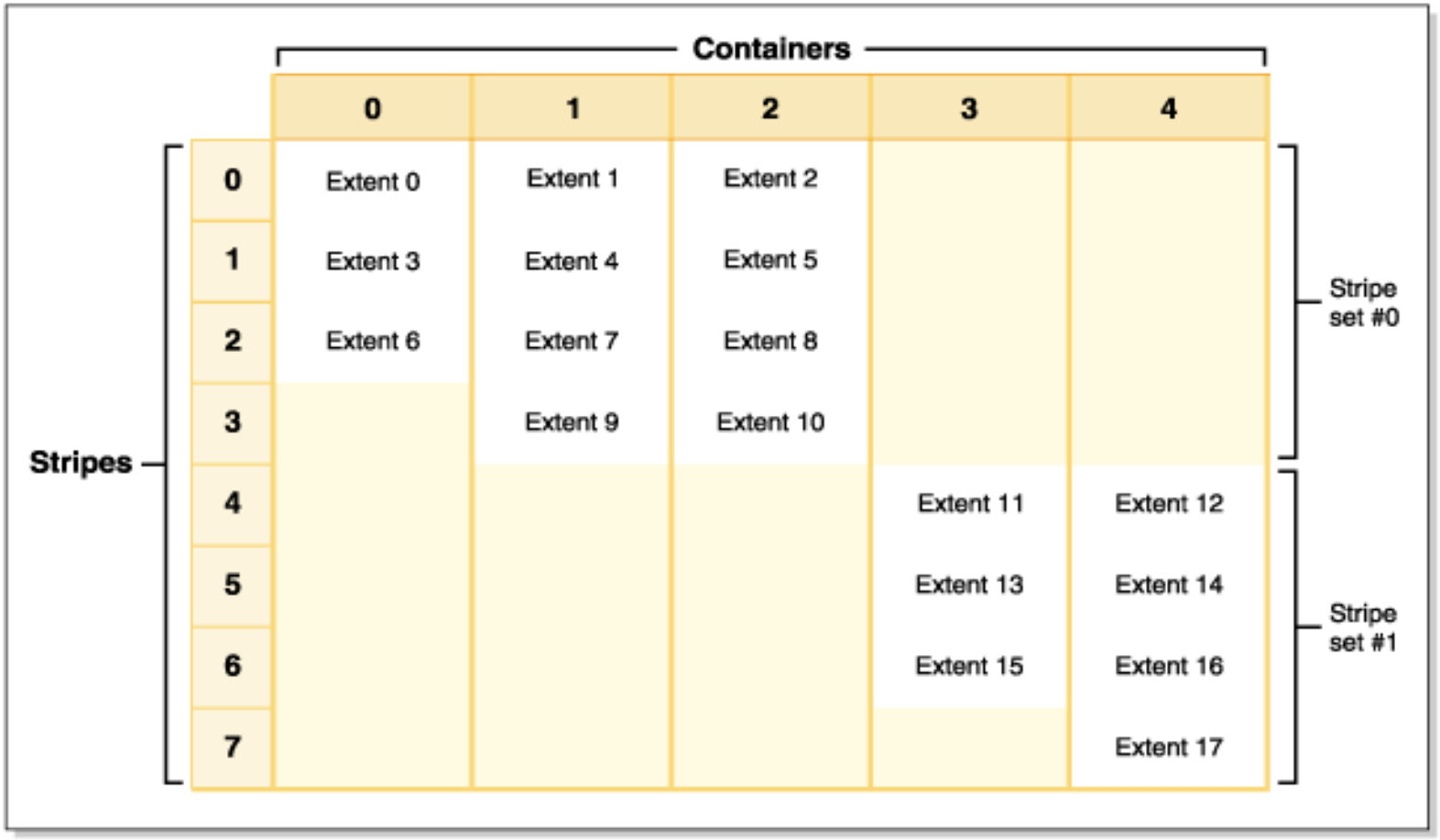
- Example 4: A table space with three containers, extent size of 10, and the containers are 30, 40, and 40 pages.



- Stripes 0 to 3 belongs to stripe set 0

Adding New Stripe Set - Example

- Example 5: Add two new containers that are 30 and 40 pages to Example 4 with the BEGIN NEW STRIPE SET option
  - **ALTER TABLESPACE abc BEGIN NEW STRIPE SET (FILE 'file1' 30, FILE 'file2' 40) ;**



Add Container to Existing Stripe Set

- Add new containers to any stripe set in the table space
- Must specify a valid stripe set
- To obtain the valid stripe set, get a table space map by taking a table space snapshot using the snapshot monitor
- Example:

Range Number	Stripe Set	Stripe Offset	Max Extent	Max Page	Start Stripe	End Stripe	Adj.	Containers
[0]	[0]	0	8	89	0	2	0	3 (0, 1, 2)
[1]	[0]	0	10	109	3	3	0	2 (1, 2)
[2]	[1]	4	16	169	4	6	0	2 (3, 4)
[3]	[1]	4	17	179	7	7	0	1 (4)

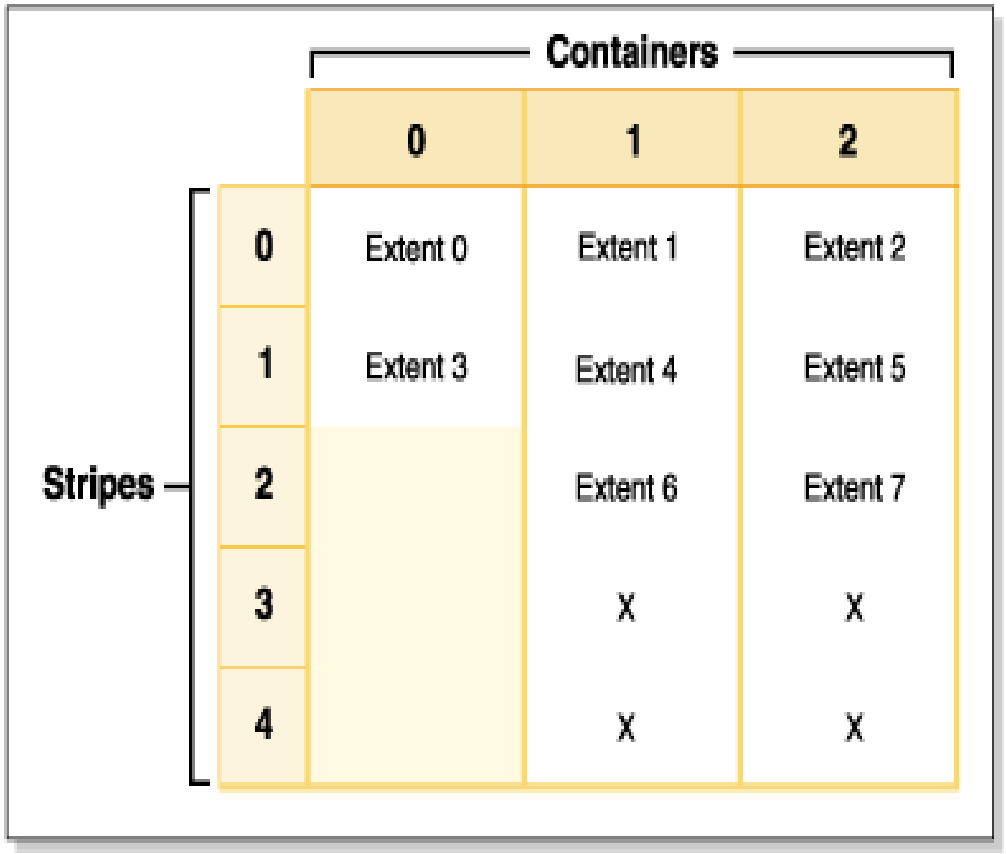
- Example:
  - ALTER TABLESPACE
    ADD TO STRIPE SET 1
    (FILE 'file1' 30, FILE 'file2' 30) ;

Drop and Reduce Container Size of a DMS Table Space

- Dropping or reducing a container will only be allowed if the number of extents being dropped is less than or equal to the number of free extents above the high-water mark of the table space.
- Data rebalancing may occur.
- The rebalancer starts with the extent that contains the high-water mark, moving one extent at a time until extent 0 has been moved.

Drop and Reduce Container of DMS Table Space - Example

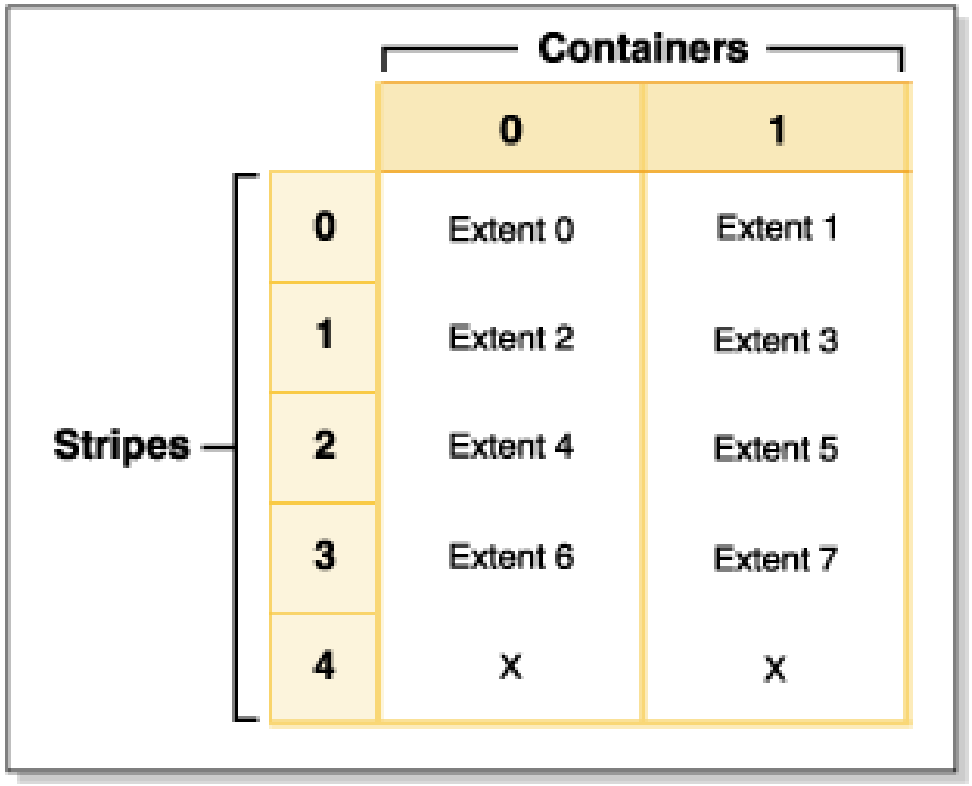
- Example 1: A table space with three containers, extent size of 10, and containers are 20, 50, and 50 pages. Assume high-water mark is at Extent 7



- An X indicates that there is an extent but they are empty (no data)

Drop and Reduce Container of DMS Table Space - Example

- Example 2: Drop container 0, which has two extents. The number of free extents above the high-water mark is four, therefore dropping container 0 is allowed. Rebalance will occur so that extents in container 0 are relocated to the other two containers. Noticed that the remaining containers are renumbered.



States of Table Spaces

- DB2 maintains information about the states of table spaces
- Common table space states:
  - 0x0 Normal
  - 0x1 Quiesced share
  - 0x2 Quiesced update
  - 0x4 Quiesced exclusive
  - 0x8 Load Pending
  - 0x10 Delete Pending
  - 0x20 Backup Pending
  - 0x80 Roll forward pending
  - 0x100 Restore pending
  - 0x4000 Offline and not accessible
  - 0x8000 Drop pending

DB2 Data Management Software

IBM

e business software

Chapter 9: Storage Management

DB2 Process Model

DB2 Memory Model

DB2 Table Spaces

Managing Table Spaces

Performance Considerations

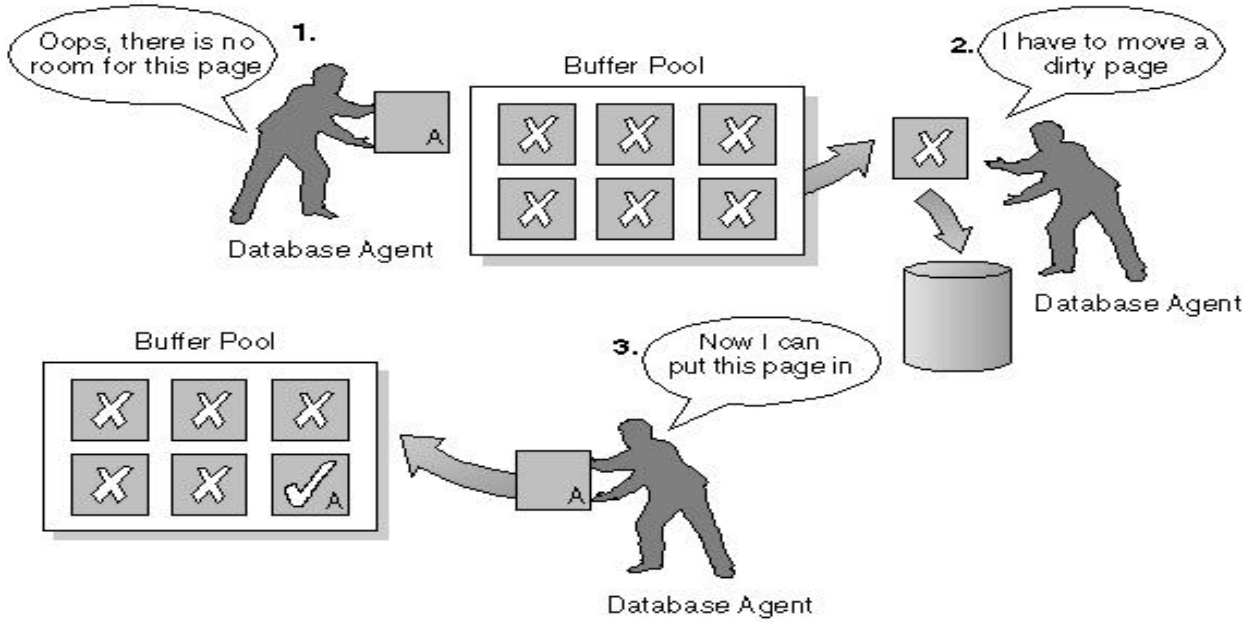
IBM Software Group

I/O Cleaners and Servers

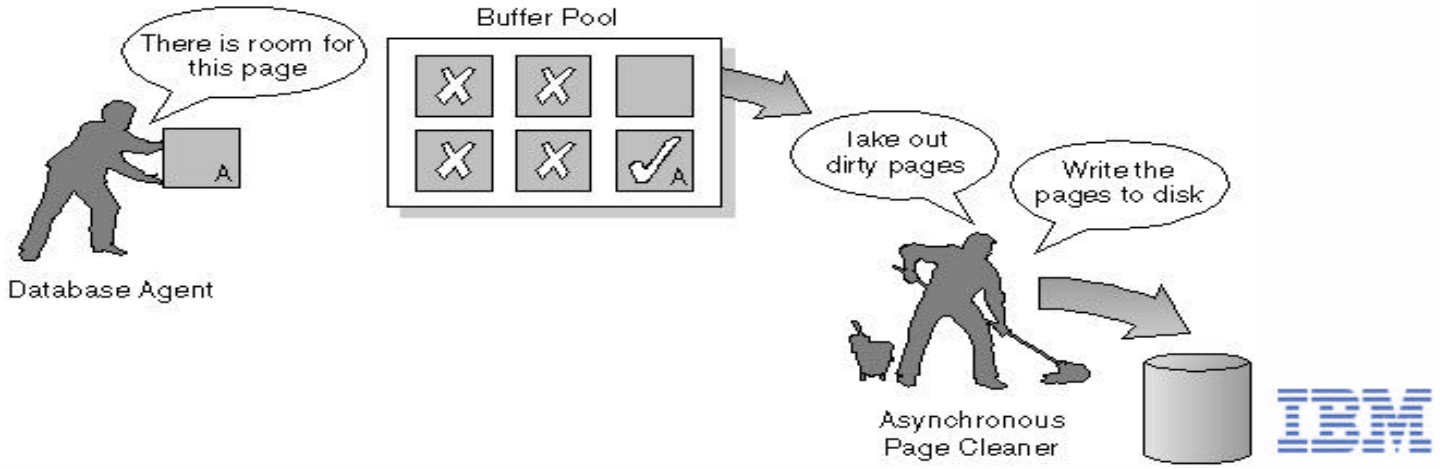
- Separate threads of control for fetching and writing pages to/from hard drives
- Can greatly enhance the performance of queries
- Parameters are NUM_IOCLEANERS and NUM_IOSERVERS
- Set num_iocleaners to be between one and the number of physical storage devices used for the database
- Parameters affect IO cleaners to be triggered:
  - CHNGPGS_THRESH - % of dirty pages in the buffer pool
  - SOFTMAX - influence # of logs needed to do crash recovery
- Set num_ioservers to one or two more than the number of physical devices on which the database resides

I/O Cleaners

Without Page Cleaners

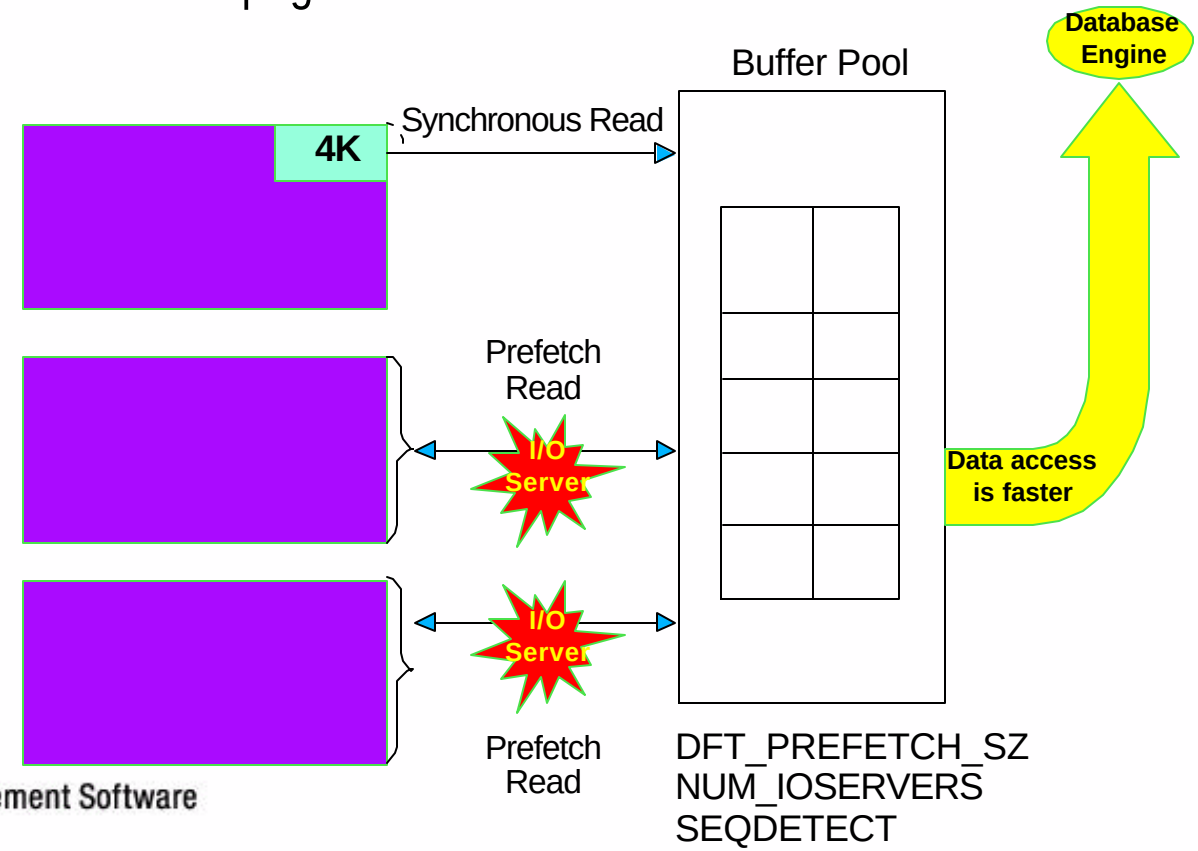


With Page Cleaners



I/O Prefetch

- Data is retrieved by I/O server tasks while previously retrieved data is processed by query tasks, thus reducing I/O bottleneck
- Prefetching, in order of best performance
  - Index prefetch occurs on index scans, RUNSTATs and REORG
  - Sequential prefetch of data pages: occurs on table scans
  - List prefetch of data pages: occurs on table scans of unclustered data



Storage Architecture: Best Practices

SMS or DMS ?

- For LOBs or LONG objects: use SMS or DMS with file containers
  - To benefit from OS file system caching
  - LOBs and LONG objects are not buffered in DB2 buffer pools
- For the catalog table space: use SMS or DMS with file containers and a small extent size (2 or 4 pages)
  - There are lots of relatively small tables in the catalog table space
  - DMS requires 2 overhead extents per table but SMS requires only 1 page
  - There are several LOB columns in the catalog tables
- For regular user data, it depends:
  - To maximize convenience and achieve very good performance, choose SMS
  - To maximize performance, choose DMS with raw containers
  - Raw containers generally outperform file containers because they avoid the path length penalty of the OS file system as well as unnecessary double buffering

Storage Architecture : Best Practices

Temporary table spaces

- SYSTEM TEMP table spaces are used for internal temporary tables, for examples:
  - Sorts (e.g.. resulting from an order by clause, create index)
  - Intermediate results
  - Table reorganization (one REORG option, "REORG with TEMP" creates the reorganized table in a TEMP table space, then copies it back to the target table space) ????
- Never create more than one TEMP table space for any pagesize
  - DB2 will likely always choose the one with the largest buffer pool anyway; the other's resources will be wasted
  - If all else is equal (i.e. the TEMPs have the same bufferpool, etc.), DB2 will round-robin between them
    - Result: you've fragmented your disk resources
- SMS is almost always the right choice for TEMP table spaces
  - Temporary table space demand is often transient (e.g.. month-end processing)
  - SMS allocates space on demand, and therefore allows disk to be used for other purposes during off-peak hours (unlike DMS)