IBM Information >>> On Demand 2007

Advanced Technical Support
TECHNICAL SALES SUPPORT AMERICAS

IBM

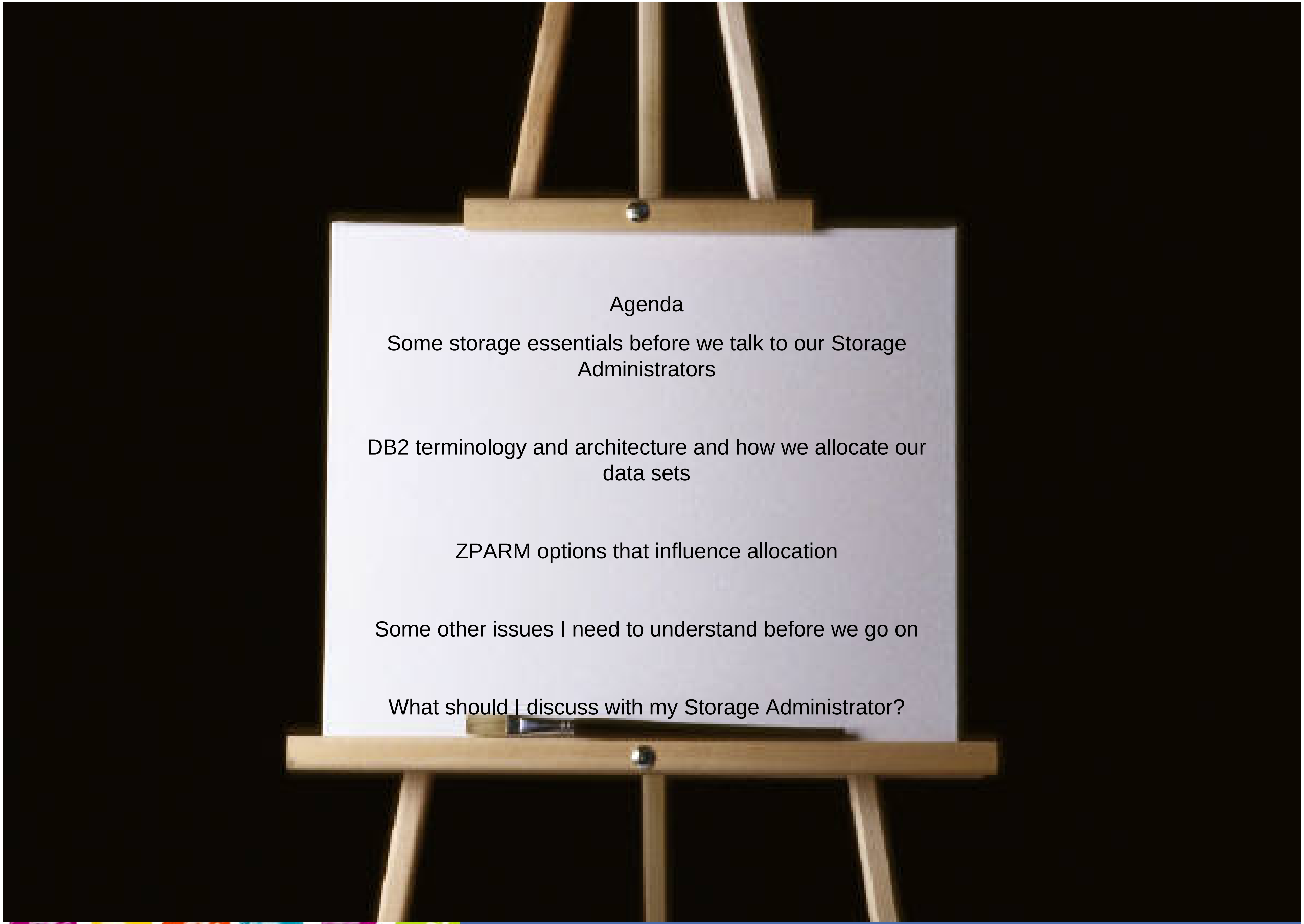
Help! My Storage Administrator Doesn't speak DB2! What Can I Do? Session 1015

John Iczkovits

iczkovit@us.ibm.com

Act.Right.Now.

IBM INFORMATION ON DEMAND 2007
October 14 – 19, 2007
Mandalay Bay
Las Vegas, Nevada



Agenda

Some storage essentials before we talk to our Storage Administrators

DB2 terminology and architecture and how we allocate our data sets

ZPARM options that influence allocation

Some other issues I need to understand before we go on

What should I discuss with my Storage Administrator?

Redpaper - Disk storage access with DB2 for z/OS

http://www.redbooks.ibm.com/redpieces/pdfs/redp4187.pdf

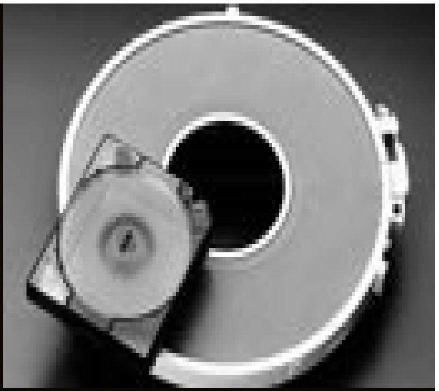
Want to understand

how DB2 works with tape?

Learn about DB2/tape

best practices:

http://www.ibm.com/support/techdocs /atsmastr.nsf/WebIndex/PRS2614



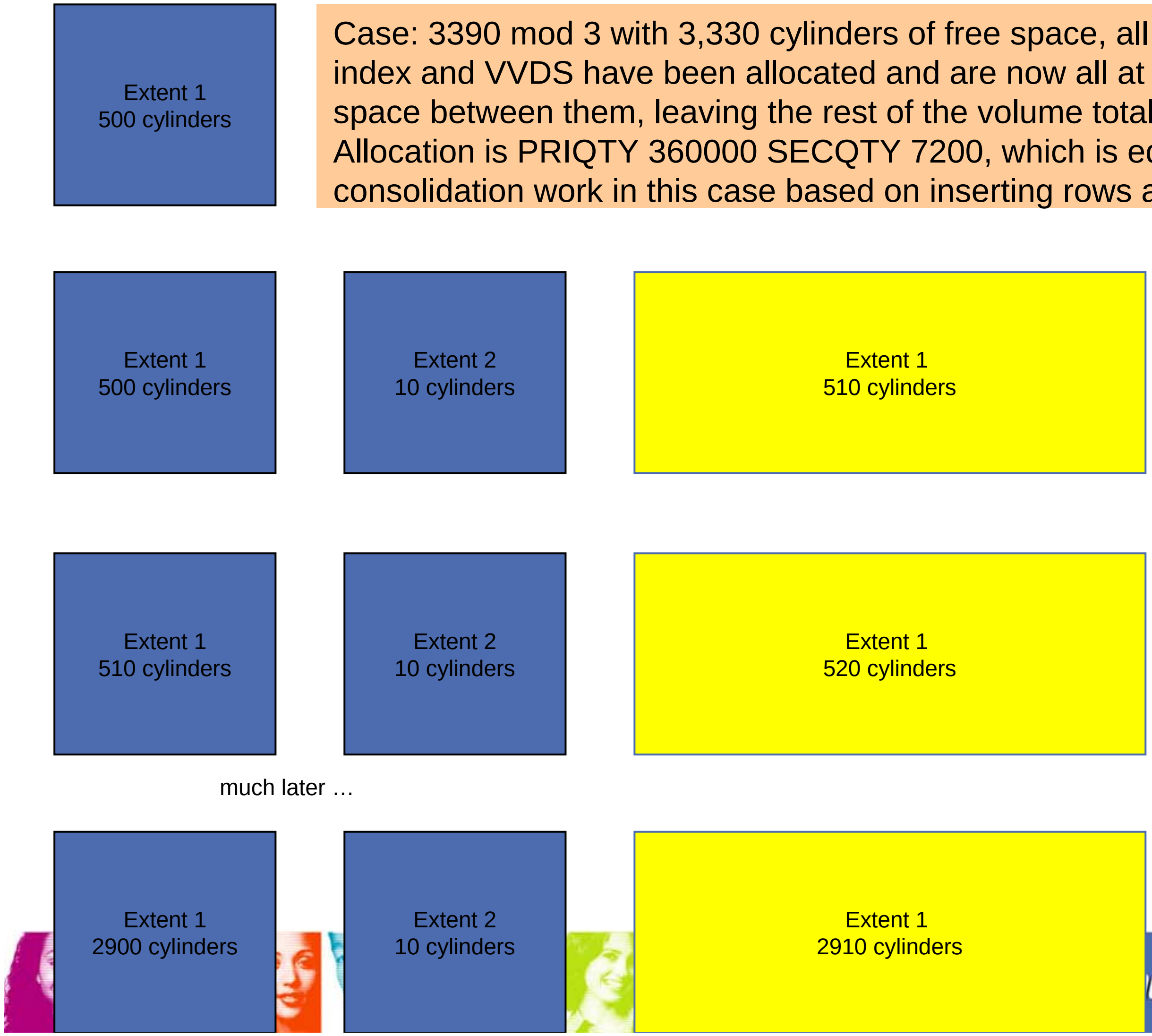
Some storage essentials before we talk to our Storage Administrators

Extent Consolidation – z/OS 1.5 plus

- Consolidates adjacent extents for DB2 LDS - VSAM data sets when extending on the same volume.
- Automatic and requires no action on your part
- If the extents are adjacent, the new extent is incorporated into the previous extent.
  - 1st extent tracks (seen on ISPF 3.4 Data set level listing) will show allocations more than the primary when extents are adjacent to the primary (see example in two pages).
- LDS must be SMS managed.
- There is a dfp limit of 123 extents per volume. Extent consolidation continues even when an LDS exceeds what would have been 123 extents. For example, my allocation is PRIQTY 720 SECQTY 720 which is 1 cylinder primary and secondary and we have an open volumes so that we do not run into another data set, if you insert 190 cylinders worth of data, you will not stop at 123 cylinders for the volume, rather allocate the 190 cylinders and have 1 extent.
- When extent consolidation is in effect in z/OS V1.5 the secondary space allocation quantity can be smaller than specified or calculated by the sliding scale based on the physical extent number. PTF UQ89517 for APAR PQ88665 corrects this problem by using the logical extent number together with the sliding scale.

Extent Consolidation on Empty Volumes

Case: 3390 mod 3 with 3,330 cylinders of free space, all in one chunk of space. The VTOC, VTOC index and VVDS have been allocated and are now all at the beginning of the volume with no free space between them, leaving the rest of the volume totally empty.
Allocation is PRIQTY 360000 SECQTY 7200, which is equivalent to CYL(500,10). How will extent consolidation work in this case based on inserting rows and increasing the space required?



Extent 1
500 cylinders

Extent 1
500 cylinders

Extent 2
10 cylinders

Extent 1
510 cylinders

Extent 1
510 cylinders

Extent 2
10 cylinders

Extent 1
520 cylinders

much later …

Extent 1
2900 cylinders

Extent 2
10 cylinders

Extent 1
2910 cylinders

In our case this segmented table space has a DB2 limitation of 2GB, which is a little bit more than 2,900 cylinders. The end result of going to the 2GB limit is the data set is allocated at over 2,900 cylinders with only 1 extent.

Allocate 10 cylinder primary for VSAM object DSNB, due to volume fragmentation largest cylinder chunks available are 4,3,3. What happens to extents:

Prior to allocation:

Cylinder 15 head 0 – cylinder 18 head 14 (4 cylinders)	Cylinder 19 head 0 – cylinder 21 head 14 (3 cylinders)	Cylinder 22 head 0 – cylinder 30 head 14 (9 cylinders)	Cylinder 31 head 0 – cylinder 33 head 14 (3 cylinders)
extent	extent	extent	extent
Free space	*Free space*	DSNA	*Free space*

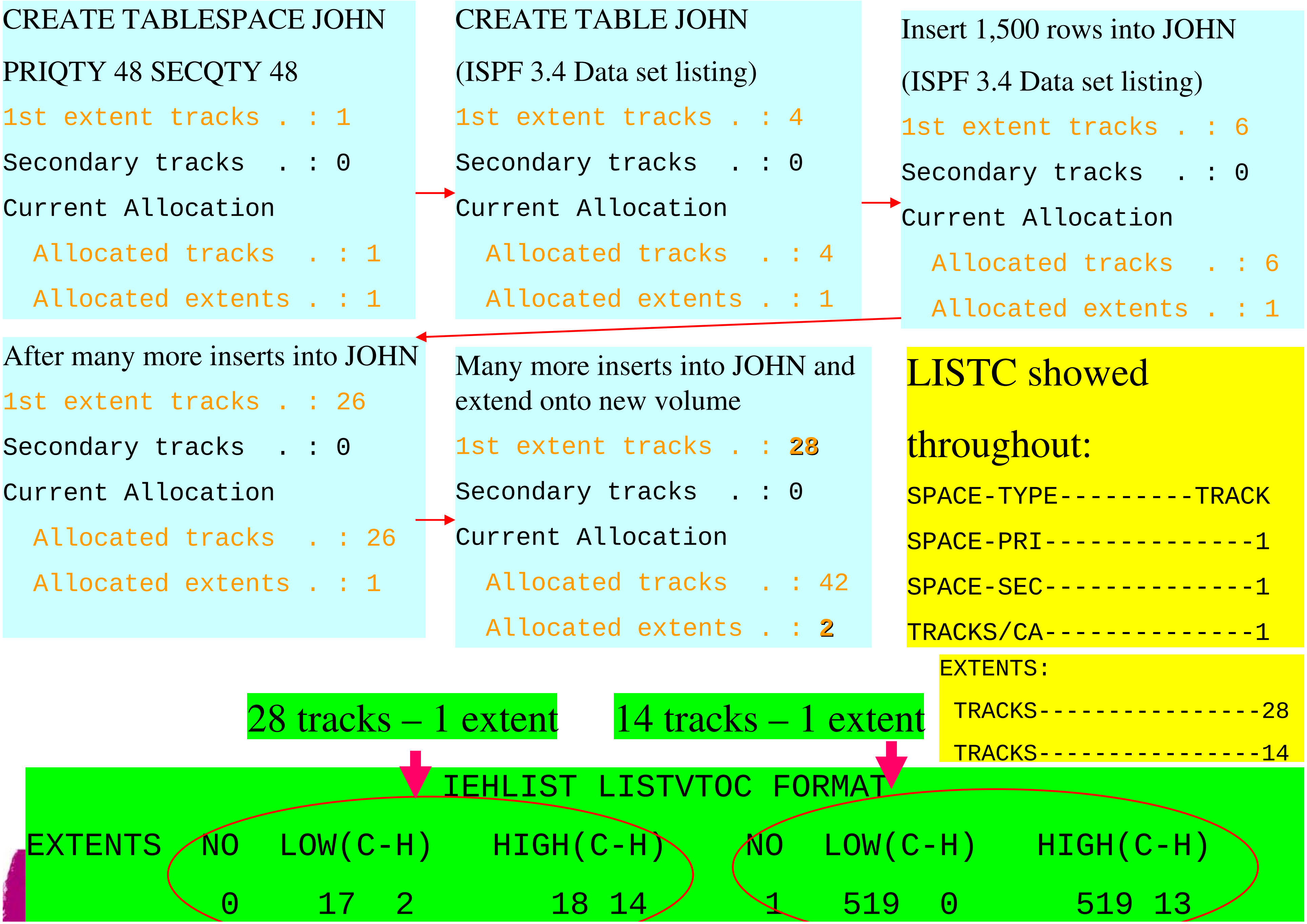
Non SMS post allocation:

Cylinder 15 head 0 – cylinder 18 head 14 (4 cylinders)	Cylinder 19 head 0 – cylinder 21 head 14 (3 cylinders)	Cylinder 22 head 0 – cylinder 30 head 14 (9 cylinders)	Cylinder 31 head 0 – cylinder 33 head 14 (3 cylinders)
extent	extent	extent	extent
DSNB	DSNB	DSNA	DSNB

SMS z/OS 1.5 and after, post allocation:

Cylinder 15 head 0 – cylinder 18 head 14 (4 cylinders)	Cylinder 19 head 0 – cylinder 21 head 14 (3 cylinders)	Cylinder 22 head 0 – cylinder 30 head 14 (9 cylinders)	Cylinder 31 head 0 – cylinder 33 head 14 (3 cylinders)
extent 1 for DSNB		extent	extent 2 for DSNB
DSNB	DSNB	DSNA	DSNB

Example – extend a data set until you must jump over another data sets extent



CREATE TABLESPACE JOHN

PRIQTY 48 SECQTY 48

```
1st extent tracks . : 1
Secondary tracks  . : 0
Current Allocation
  Allocated tracks  . : 1
  Allocated extents . : 1
```

CREATE TABLE JOHN

(ISPF 3.4 Data set listing)

```
1st extent tracks . : 4
Secondary tracks  . : 0
Current Allocation
  Allocated tracks  . : 4
  Allocated extents . : 1
```

Insert 1,500 rows into JOHN

(ISPF 3.4 Data set listing)

```
1st extent tracks . : 6
Secondary tracks  . : 0
Current Allocation
  Allocated tracks  . : 6
  Allocated extents . : 1
```

After many more inserts into JOHN

```
1st extent tracks . : 26
Secondary tracks  . : 0
Current Allocation
  Allocated tracks  . : 26
  Allocated extents . : 1
```

Many more inserts into JOHN and extend onto new volume

```
1st extent tracks . : 28
Secondary tracks  . : 0
Current Allocation
  Allocated tracks  . : 42
  Allocated extents . : 2
```

LISTC showed

throughout:

```
SPACE-TYPE---------TRACK
SPACE-PRI--------------1
SPACE-SEC--------------1
TRACKS/CA--------------1
```

```
EXTENTS:
 TRACKS----------------28
 TRACKS----------------14
```

28 tracks – 1 extent

14 tracks – 1 extent

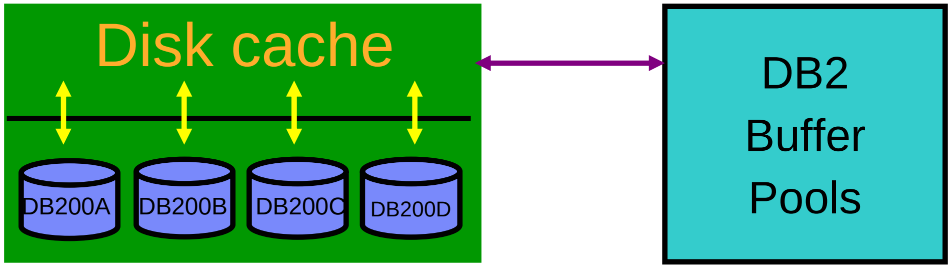
```
                 IEHLIST LISTVTOC FORMAT
EXTENTS   NO   LOW(C-H)     HIGH(C-H)      NO   LOW(C-H)     HIGH(C-H)
           0     17   2        18 14        1     519   0        519 13
```

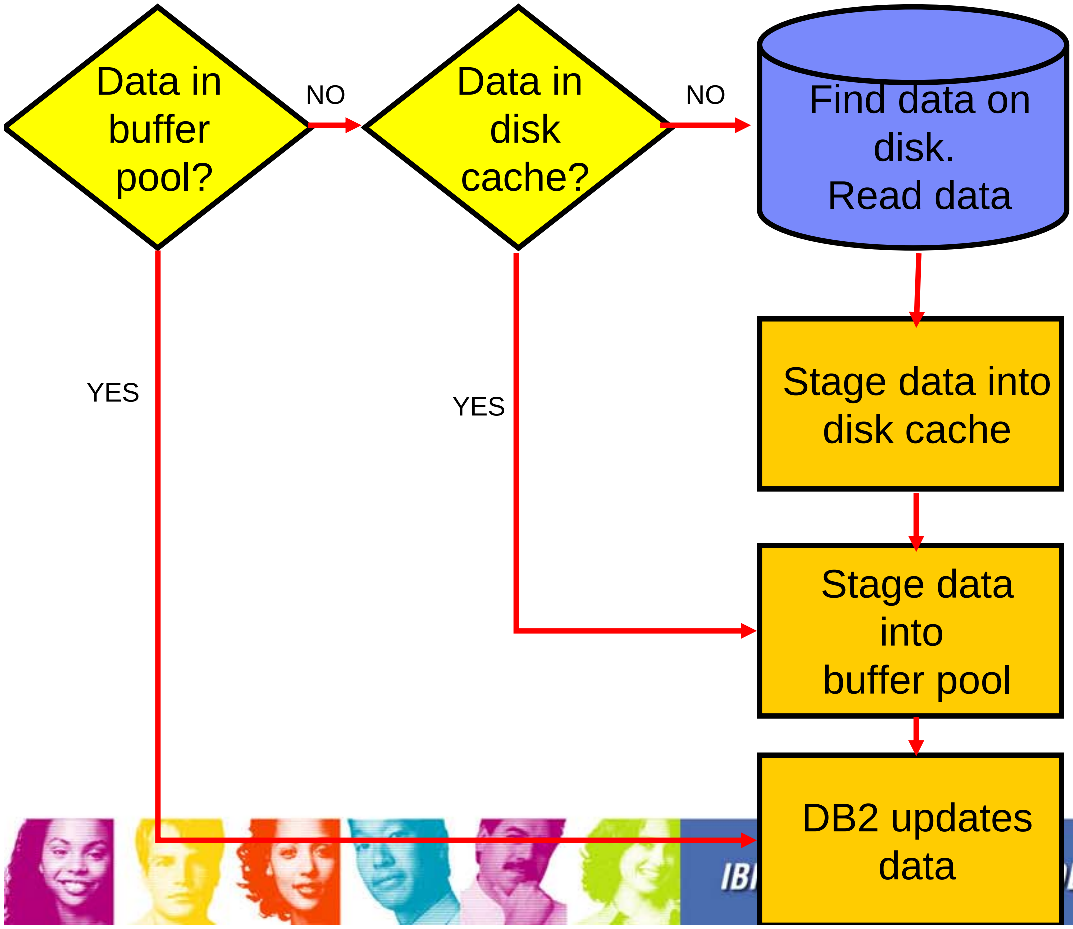
Disk Cache - Why Storage Administrators think we are unfriendly! page 1 of 2

Illustration does not include Data Sharing with GBP dependent data, disk perspective is still the same.

- at a very high level –

```
UPDATE IBM.JOHN
SET SALARY = SALARY + 5000
WHERE NAME='JOHN';
```




Data in buffer pool?
NO
Data in disk cache?
NO
Find data on disk. Read data
YES
YES
Stage data into disk cache
Stage data into buffer pool
DB2 updates data

- Based on DB2 checkpoint or buffer pool thresholds:
  - DB2 destages updated data back down to cache.
  - DB2 sees this as a disk write, even though the write is to cache.
  - Data at this time may or may not remain in the buffer pool depending on why data was destaged.
  - Data is also written to the NVS (Non Volatile Storage) part of the disk controller that is battery backed. If the controller crashes, data is not lost.
- Based on disk thresholds, cache destages data back down to disk.

Disk Cache - Why Storage Administrators think we are unfriendly! page 2 of 2

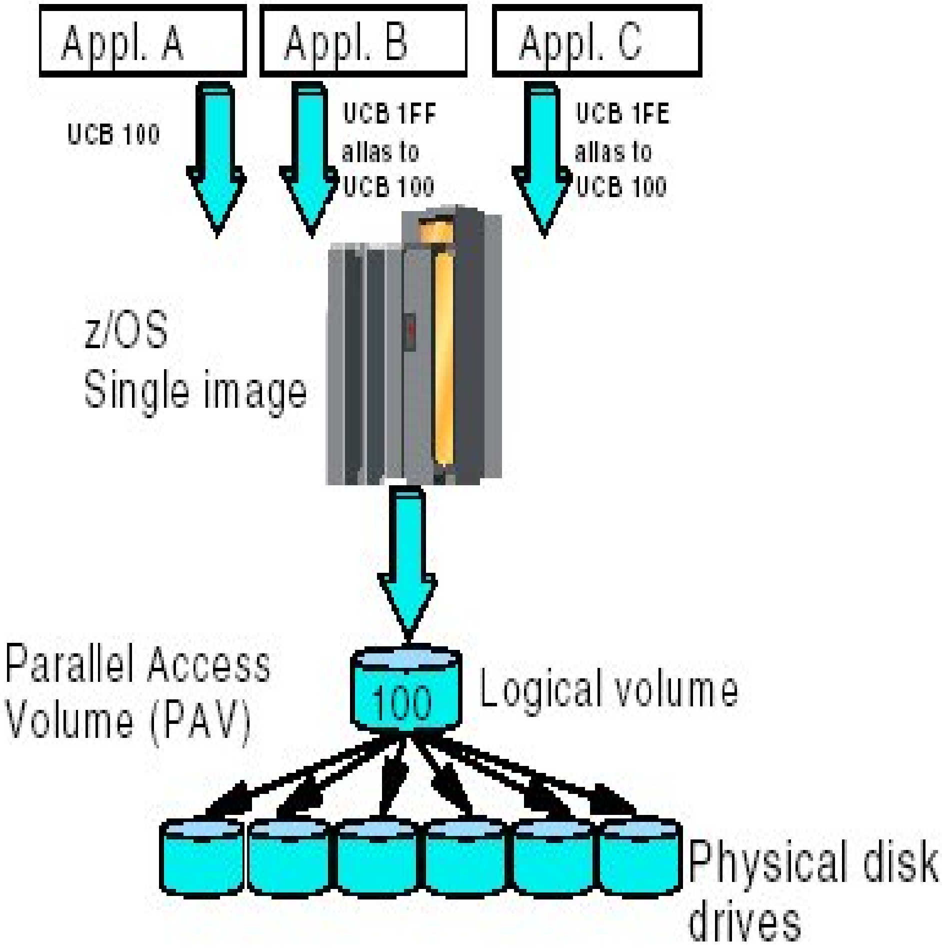
- SELECT * FROM IBM.JOHN;
  - Read only operations do not require data to be destaged to cache.
- From a conceptual point, the function of cache and buffer pools are similar.
  - From a storage perspective, DB2 data is typically considered “unfriendly” because of the relatively low reuse of data in cache.
  - DB2 will use the data residing in the buffer pool when available. It may not require the data in disk cache at all.
- Read from cache is exponentially faster than from disk.
  - No need to go to disk, find the data, and bring it back through cache.
- Just because your buffer pool casts out data, it does not mean that it is no longer retained in cache.
  - Newer disk controllers have very large cache sizes and can retain data for longer periods.
  - In DB2 V8. You can allocate very large buffer pools, as long as they are backed by real storage.

Data IBM disk controllers bring into cache:

- Data access is determined through the controllers adaptive cache.
- Random read –
  - – record or block staging (page)
  - – partial track staging (from the page requested until the end of the track)
  - – full track staging
  - – Adaptive cache will determine which of the three will be used based on previous use of data whereby anything from as small as one page, or as large as from the page until the end of the cylinder plus the next cylinder will be read in.
- Sequential prefetch – full track staging – prefetch from the page up to the end of the cylinder plus the next cylinder (extent boundary). Actual operation is done on extent boundaries or stage group (an internal construct, depends on rank dimensions).
- See ZPARM section for SEQCACH.

Parallel Access Volumes (PAV) - ESS "Shark" and DS8000

- Multiple UCBs per logical volume
- PAVs allow simultaneous access to logical volumes by multiple users or jobs from one system.
- Reads are simultaneous
- Writes to different domains are simultaneous
- Writes to same domain are serialized
- Eliminates or sharply reduces IOSQ
- High I/O activity, particularly to large volumes (3390 mod 9, 27, and 54) greatly benefits from the use of PAV.
- WLM GOAL mode management of dynamic PAVs. Static PAVs do not require WLM. Dynamic PAVs and Priority I/O Queueing recommended.
- Multiple paths or channels for a volume is nothing new, but multiple UCBs (MVS addresses) for a volume is.



Dynamic PAV (Parallel Access Volumes)

aka “WLM-managed PAV”

- ✓ PAVs reduce or avoid IOSQ time. PAVs are essential for large volumes.
- ✓ Each PAV is assigned one of 64K UCB addresses. The number of UCB addresses limits the number of PAVs in a sysplex.
- ✓ PAVs are pooled by LCU (Logical Control Unit)
- ✓ PAVs are dynamically assigned by Workload Manager to a base volume
- ✓ The assignment must be coordinated throughout the sysplex
    - Lots of WLM overhead
- ✓ Different systems compete for PAVs if the systems share the same LCU
- ✓ The number of PAVs needed to avoid IOSQ time far exceeds the maximum number of concurrent I/Os, especially in a sysplex environment

Hyper PAV

- PAVs are assigned to a base volume only for the duration of the I/O
- No coordination required across the sysplex
- Different systems won’t compete against each other
- PAVs are still pooled by LCU
- Advantages:
  - 64K UCB address constraint is relieved
  - No WLM overhead
  - The total number of PAVs needed for an LCU is equal to the maximum number of concurrent I/Os
  - The same alias address can be used by different systems for different volumes
- System Requirements: z/OS 1.8, Release 2.4 ucode for the DS8000, upgradeable for existing DS8000 control units
  - PTFs to be supplied for z/OS 1.6 and 1.7

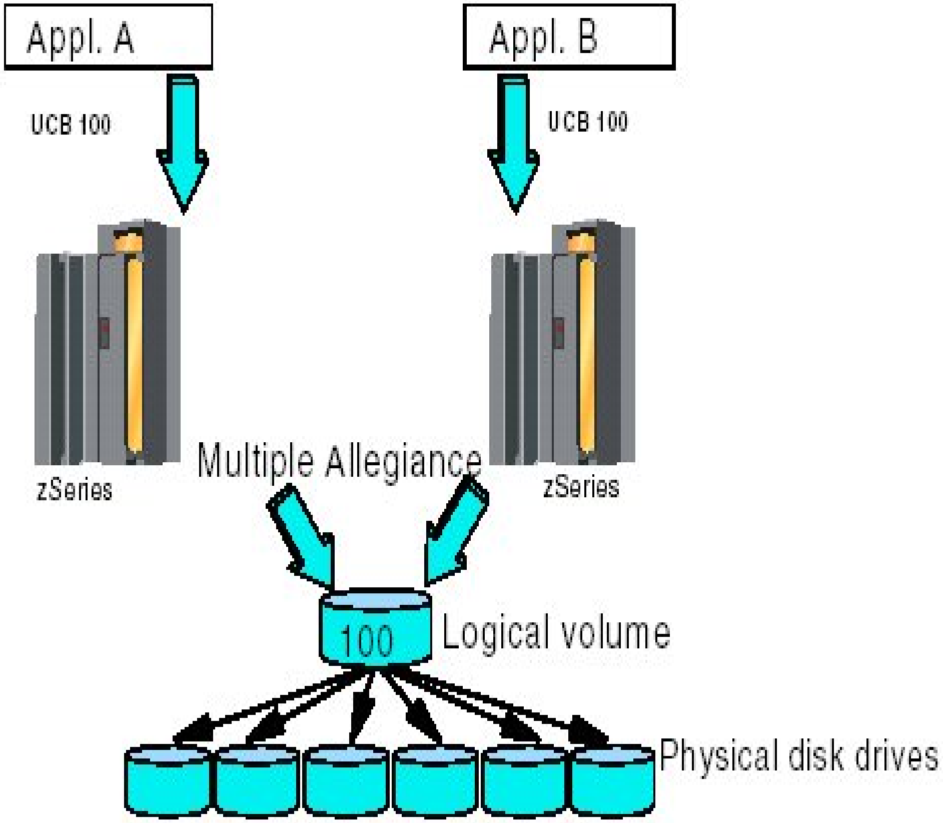
Hyper PAVs

Workloads that benefit most from UCB constraint relief

- ✓ PPRC (Synchronous Peer to Peer Remote Copy)
  - Eliminating IOSQ time allows reads to be serviced during writes and also allows more writes to be service in parallel.
- ✓ If the cache hit ratio is poor, PAVs may only shift queue time from IOSQ to disconnect time, because of DDM contention. However, shifting the queues into the control unit prevents cache hits from queueing behind cache misses.
- ✓ Different hosts access the same LCUs, but with different device skews.

Multiple Allegiance (MA) - ESS "Shark" and DS8000

- **Similar to PAV, however for more than one LPAR. Unlike PAV, MA is automatically turned on with your IBM disk.**
- **Incompatible I/Os are queued in the ESS/DS8000**
- **Compatible I/O (no extent conflict) can run in parallel**
- **ESS/DS8000 boxes guaranty data integrity**
- **No special host software required, however: Host software changes can improve global parallelism (limit extents)**
- **Improved system throughput**
  - Different workloads have less impact on each other
- **Reduces PEND time (device busy)**



Useful for Data Sharing!

VSAM Data Striping - ESS, DS8000, and some RVA

- Spreads the data set among multiple stripes on multiple control units (this is the difference from hardware striping which is within the same disk array)
- An equal amount of space is allocated for each stripe
- Striped VSAM data sets are in extended format (EF) and internally organized so that control intervals (CIs) are distributed across a group of disk volumes or stripes.
  - DB2 V8 now allows striping for all page types, while V7 only allows striping for 4K pages. See APAR PQ53571 For issues with objects greater than 4K pages prior to DB2 V8.
- Greater rate for sequential I/O
- Recommended for DB2 active log data sets
- I/O striping is beneficial for those cases where parallel partition I/O is not possible. For example: segmented tablespace, work file, non partitioning indexes (NPIs, NPSIs), DPSIs, LOBs, etc.

Sequential Data Striping

- Recommendations –
  - Stripe all possible objects that are input or output for utilities when the associated table space is striped. For example (when the associated table space is striped):
    - Stripe the image copy data sets
    - Stripe input files for LOADs, etc.
- Only disk data sets can be striped.
- You can now stripe your disk archive log data sets beginning in DB2 9.
  - Archive log data sets can be compressed as well starting in DB2 9.

Large Sequential Data Sets – z/OS 1.7 and above – see DB2 notes.

- Removes the size limit of 65535 tracks (4369 cylinders) per volume for sequential data sets
  - BSAM, QSAM, and EXCP
    - Data sets do not have to be in SMS managed and in extended format
  - 16 extents is still the limit
  - Architectural limit is 16,777,215 tracks
  - JES2/JES3 spool can now be larger than 64K tracks
    - Must still be a single extent
- Changed APIs supports all sequential and partitioned data sets
- DFSMShsm support of migration/recall, backup/restore and ABACKUP/ARECOVER of large format data sets
- Addresses limits on capacity in customers’ systems
  - Direction is to support large capacity devices
  - Systems support up to 64,512 (63K) devices

Specifying Large Sequential Data Sets

- DSNTYPE DD statement, dynamic allocation text unit, TSO ALLOCATE keyword support 4 new values:
  - LARGE
    - If the data set is sequential or DSORG is omitted, then large format data set
  - EXTREQ
    - If the data set is VSAM or sequential or DSORG is omitted, then extended format data set is required
  - EXTPREF
    - If the data set is VSAM or sequential or DSORG is omitted, then extended format data set is preferred
      - If not possible, then neither extended format, nor large format
  - BASIC
    - If the data set is VSAM or sequential or DSORG is omitted, then neither extended format nor large format
- Data class can provide the new DSNTYPE
  - New &DSNTYPE variable for ACS routines with values of either LARGE or BASIC

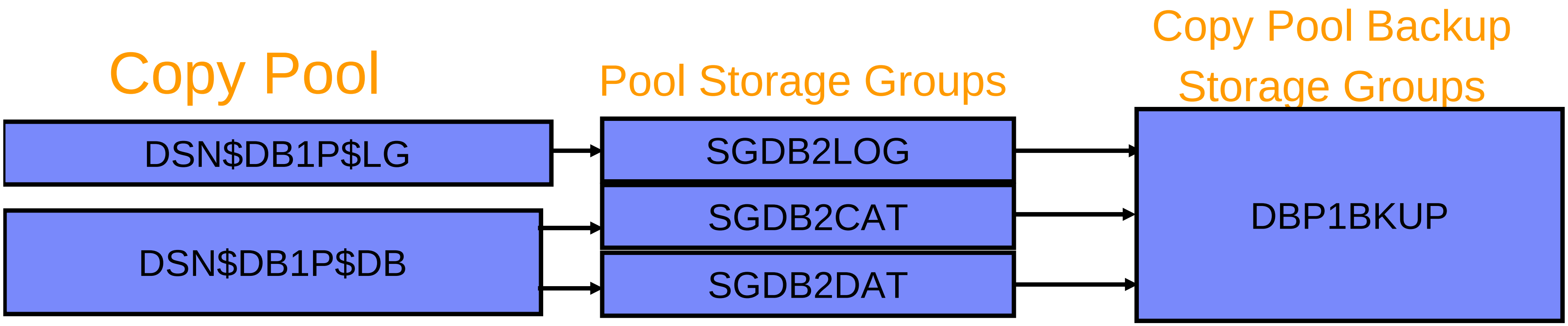
DB2 9 and Large Sequential Data Sets

- Can be non SMS managed.
- In DB2 9 NFM supported for:
  - Utility input data sets.
  - Utility output data sets when coded in the SMS Data Class or DD includes DSNTYPE=LARGE.
  - Archive log data sets can be allocated on one disk as 4 GB -1 byte. Archive data sets can be created on disk only in NFM or ENFM modes, however, DB2 tolerates reading them in CM.

dfp Enhancements and DB2

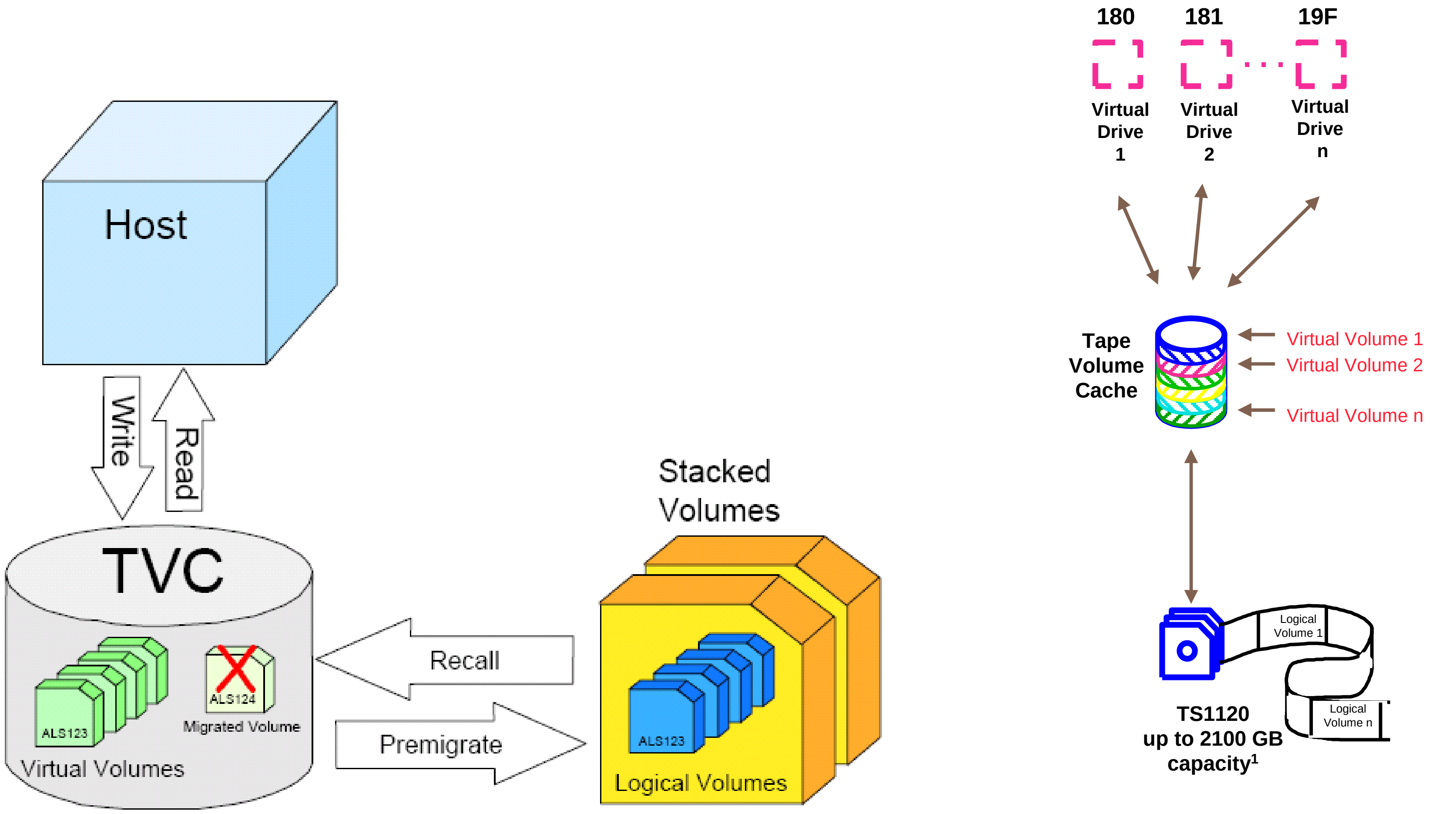
- As with DB2, dfp introduces enhancements on a periodic basis. These enhancements may effect data sets used by DB2, however, keep in mind that many times a DB2 APAR is required before DB2 can take advantage of the dfp enhancement.
  - For example, the dfp enhancement to increase the VSAM data set maximum to 7,257 extents required some DB2 APARs in order to utilize the enhancement. Another example is when dfp introduced large sequential data sets in z/OS 1.7, DB2 could not take advantage of utilizing this enhancement for such things as archive log data sets and image copies until the introduction of DB2 9.
- Verify if any DB2 maintenance is required to take advantage of dfp enhancements.

Copy Pools - BACKUP Command (DB2 V8)



- A copy pool is a defined set of pool storage groups that contain data that DFSMShsm can backup and recover collectively, using fast replication.
- DFSMShsm manages the use of volume-level fast replication functions, such as FlashCopy and SnapShot.
- Provides point-in-time copy and recovery services.
- DSN$locn-name$cp-type, DSN and $ are required, locn-name is the DB2 location name, cp-type is the copy pool type. DB is for database. LG is for logs. For example: DB2 DB1P would have copy pools named DSN$DB1P$DB for the database copy pool and DSN$DB1P$LG for the log copy pool.
- BACKUP command records entry in BSDS, as well as the DFSMShsm BCDS.

TS7700 Virtualization Engine Operation



[1] assumes 3:1 compression

Some key differences between ATLs and virtualized tape

- ATLs do not contain disk drives to house data, therefore no virtualization is possible.
- ATL VOLSERs are the actual physical tape VOLSER, while only the logical tape is known for virtualized tape.
- ATL tape UCBs are the physical tape drives, while virtualized tapes will only externalize the logical drive.
- Virtual tape allows for copying of data to another location via VTS PtP or the TS7700 grid system, ATLs do not allow for this type of copy mechanism.
- ATLs do not automatically stack data sets on tape as virtualized tapes do. Software is required to stack tapes on ATLs.
- Only virtualized tape will emulate an IBM 3490 Enhanced Capacity (3490E) tape drive of a specific size: 400, 800, 1000, 2000 or 4000 MB. When using ATLs, you are not restricted by these sizes, rather the size of the physical tape.

Tape Issues

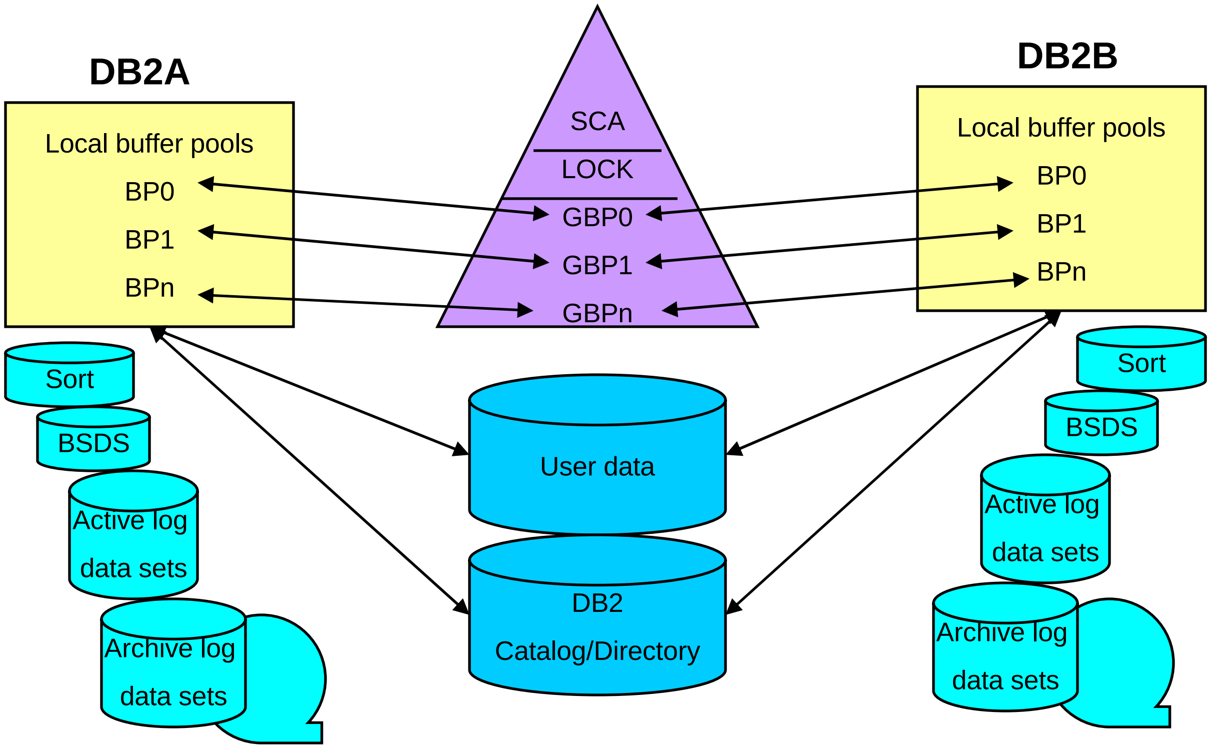
- Tape is an excellent medium to place your DB2 related data on. Consider tape for your:
  - Archive logs
  - Image copies
  - Very large Sortwork data sets
  - Other assorted data sets
- There are some challenges using tape in a DB2 environment:
  - VSAM data sets, PDSs and PDSEs must reside on disk
  - No concurrency or sharing within a data set or volume, therefore parallelism does not does not exist.
  - Tapes perform best using pure sequential access. Other access types may result in some performance degradation.
  - Data sets residing on tape cannot be striped
- Discuss your tape and data requirements with your Storage Administrator. Understand your company's tape solutions and how to best manage your DB2 related data.

DB2 terminology and architecture

and

how we allocate our data sets

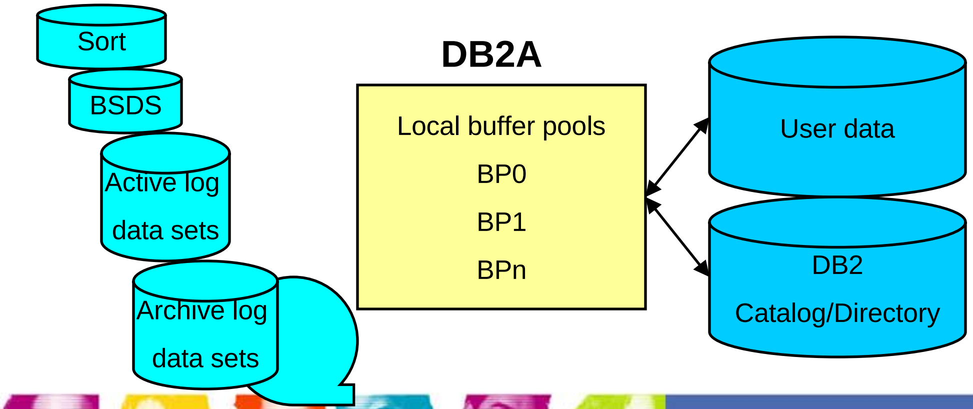
DB2 Member vs. subsystem



DB2 **Data Sharing group**

With **members**

DB2A and DB2B



DB2 **subsystem**

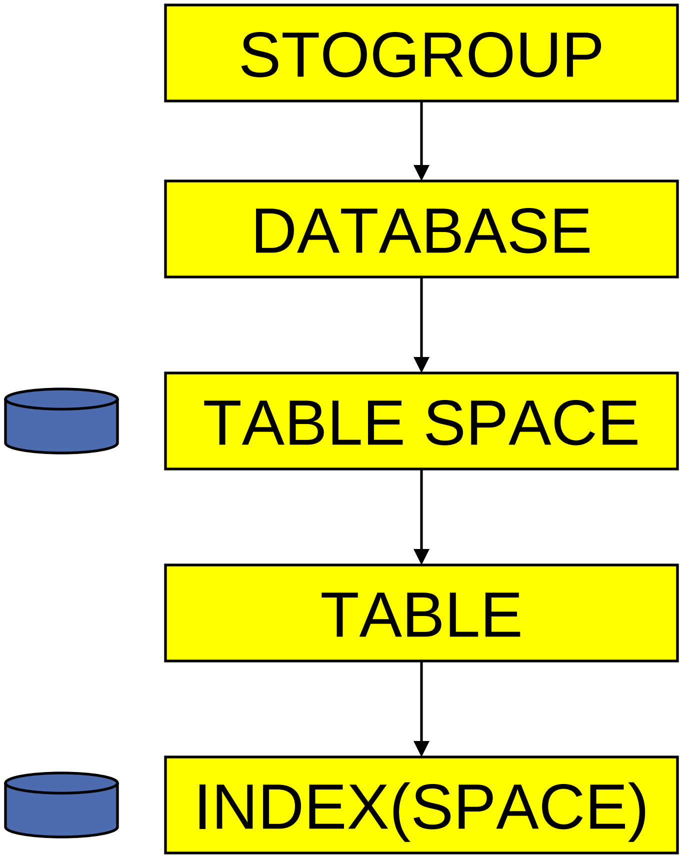
DB2A

(no Data Sharing!)

Important Physical DB2 Data Sets

- Table and index spaces – VSAM LDS. Indexes are LDSes as well with a cluster and data component.
- Image copy – PS data sets. Table space and/or index backup using the DB2 COPY utility. DB2 Copies data by page.
- DB2 catalog and directory – VSAM LDS. Houses internal DB2 information for objects.
- BSDS (Boot Strap Data Set) – VSAM KSDS. Houses internal information for such things as the active and archive log data sets, etc. Should be software (DB2) dual copied.
- Active log data sets – VSAM LDS. Logs object updates. Should be software (DB2) dual copied.
- Archive log data sets – PS data sets. Archive of used active log data sets. Should be software (DB2) dual copied.
- Most other DB2 data sets are PDSE, PDS, and sequential data sets.

Storage Related DB2 Object Creation Flow



- DB2 STOGROUP and SMS Storage Group are similar in concept, with the SMS Storage Group being more sophisticated.
- Table space and Index (space) objects are physical, while the others are logical.

⛁ = physical data set

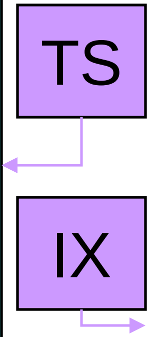
Creation of DB2 Data Sets

- Table spaces and index spaces are VSAM LDS data sets, which can be:
  - DB2 or user managed
    - DB2 managed data sets are created using DDL
    - User managed data sets are created outside of DB2, typically through IDCAMS
  - SMS or non-SMS managed
    - When SMS managed, the LDS can be defined by using the DB2 STOGROUP, SMS Storage Group, or both.
      - Recommendation – when SMS managing your LDSes, for the DB2 STOGROUP use VOLUMES(“*”) and let SMS do its job. With DB2 9, For the DB2 STOGROUP, you can specify SMS DATACLAS, STORCLAS, or MGMTCLAS and not specify the VOLUMES attribute.

User and DB2 Managed Allocations

User managed – data sets created by users, typically through IDCAMS. Users are responsible to maintain the data sets including creating new data sets when size limit is reached.

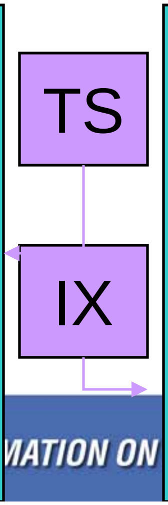
```
DEFINE CLUSTER                                -
   (NAME(HLQ.DSNDBC.DBNAME.TSNAME.I0001.A001) -
      LINEAR REUSE VOLUMES(DB200A)            -
       CYL(100 10)                    -
       SHAREOPTIONS(3 3))                -
 DATA                             -
    (NAME(HLQ.DSNDBD.DBNAME.TSNAME.I0001.A001))
```



```
DEFINE CLUSTER                                -
   (NAME(HLQ.DSNDBC.DBNAME.IXNAME.I0001.A001) -
      LINEAR REUSE VOLUMES(DB200B)       -
       CYL(30 3)                      -
       SHAREOPTIONS(3 3))                -
 DATA                             -
    (NAME(HLQ.DSNDBD.DBNAME.IXNAME.I0001.A001))
```

DB2 managed – data sets created by DB2 using DDL. DB2 is responsible to maintain the data sets including creating new data sets when size limit is reached.

```
CREATE TABLESPACE TSNAME
  IN DBNAME
  USING STOGROUP STGNAME
          PRIQTY 72000
          SECQTY 7200
          ERASE NO
  LOCKSIZE PAGE LOCKMAX SYSTEM
  BUFFERPOOL BP2
  CLOSE NO
  CCSID EBCDIC;
```



```
CREATE INDEX AUTHID.IXNAME
  ON authid.table
     (MGRNO   ASC)
  USING STOGROUP stgname
    PRIQTY 21600
    SECQTY 2160
    ERASE NO
    BUFFERPOOL BP3
    CLOSE NO;
```

How is space allocated for DB2 user data sets?

- User managed data sets:
  - – normal space parameters used in IDCAMS
- DB2 managed data sets – prior to DB2 V8:
  - – DDL through PRIQTY and SECQTY
    - PRIQTY and SECQTY are in KB. Although a track can hold 56KB, DB2 data sets will only allocate and use 48KB of the 56KB. For one track specify PRIQTY 48, one cylinder specify 720 (48 (KB) * 15 tracks)
    - If SECQTY=0 then no secondary is allowed
    - Through a combination of sliding secondary and ZPARM values for TSQTY, IXQTY

Where can I find allocation or extent error messages for DB2 LDSes?

- User managed data – typically in the IDCAMS job for the allocation. Extent and extend error messages can be found in the DB2 MSTR and/or DBM1 STCs.
- DB2 managed data sets – create, extent, and extend error messages can be found in the DB2 MSTR and/or DBM1 STCs.

DB2 LDS Overview

- Row – logical DB2 record. VSAM records are made of one or more DB2 rows stored in pages.
- Page - a unit of storage within a table space (4 KB, 8 KB, 16 KB, or 32 KB) or index space (4 KB). In a table space, a page contains one or more rows of a table.
- Table – contains pages for one or more common objects.
- Table space - a page set that is used to store the records in one or more tables. Table spaces are VSAM LDS data sets.
- Index (optional) – a set of pointers that are logically ordered by the values of a key.
- index space (optional) - a page set that is used to store the entries of one index. Index spaces are VSAM LDS data sets.

Note – because DB2 indexes are LDSes, they are a separate cluster and data component from the table space LDS cluster and data component. DB2 indexes are not VSAM Index components.

Types of table spaces created

- Simple – for multi table (can be single) – table spaces, each row on a page can owned by any table. Limit: 64 GB total = 2GB per LDS * 32 data sets. LLQ denotes space allocated, i.e. first data set 2GB LLQ=A001, next data set 2GB LLQ=A002, next data set 2GB LLQ=A003. No longer available in DB2 9.
- Segmented (most common) – for multi table (can be single) – table spaces, each page can only contain data from a specific table. Limit: 64 GB total = 2GB per LDS * 32 data sets. LLQ denotes space allocated, i.e. first data set 2GB LLQ=A001, next data set 2GB LLQ=A002, next data set 2GB LLQ=A003.
- Partitioned – only one table possible. Each partition ends at a specific range. Limit: Up to 128TB total, however each partition can not exceed a specific limit depending on the number of partitions. Each partition regardless of the total number of partitions can not exceed 64GB. LLQ is the partition number. For example, have 5 partitions with ranges of data starting with 0-50:
  - Partition 1 (LLQ A001) – from 0-10
  - Partition 2 (LLQ A002) – from 11-20
  - Partition 3 (LLQ A003) – from 21-30
  - Partition 4 (LLQ A004) – from 31-40
  - Partition 5 (LLQ A005) – from 41-50
- LOBs (Large Objects) - a table space in an auxiliary table that contains all the data for a particular LOB column in the related base table. Can be a partition as well. Useful for such things as housing movies.

NOTE – there are several different types of indexes as well, which we do not go into detail in this presentation.

Types of table spaces created starting with DB2 9

- XML Table spaces – similar to LOB table spaces, however used for XML data.
- UTS (Universal Table Spaces) – combines the features of segmented and partitioned table spaces to create a new type of DB2 data set. UTS can be created as partition by growth (PBG) or partition by range (PBR). UTSes contain one table per table space. Limit: 128 TB, similar restrictions to partitions.

DDL options that influence space for DB2 managed data sets

- Both table spaces and indexes:
  - DSSIZE (for table spaces only, its index will match the value) – up to 64GB. Only valid for partitions and LOBs. Requires EA/EF for objects greater than 4GB.
  - LARGE (for table spaces only, its index will match the value) – 4GB only. Only valid for partitions and LOBs. Does not require EA/EF.
  - PRIQTY – Primary quantity specified in KB
  - SECQTY – Secondary quantity specified in KB
  - USING STOGROUP – Determines DB2 STOGROUP
  - FREEPAGE – Number of free pages after a page with data. May be useful, but only after a REORG or LOAD
  - PCTFREE – Percentage of a page to leave free after a REORG or LOAD
  - BUFFERPOOL – For table spaces determines the size of a page (4K, 8K, 16K, and 32K) as well as the buffer pool. For indexes, determines the buffer pool. Currently indexes are allocated as 4K only.
  - DEFINE – Determine if object should be physically created at the time of execution or deferred until data in inserted. It is possible to have an entry for a table and/or index space in the DB2 catalog, but not have the LDS created until data is inserted.

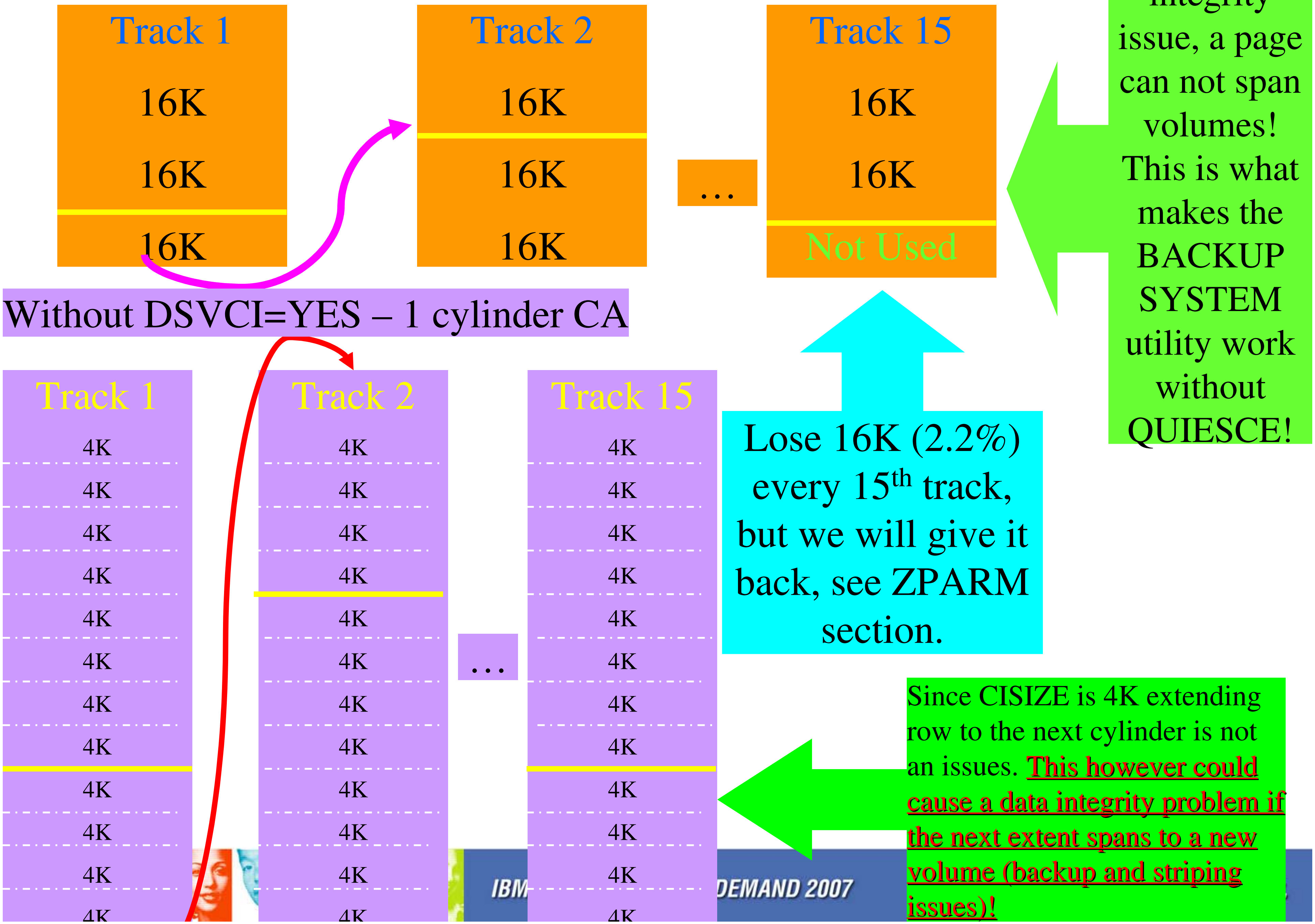
DDL options that influence space for DB2 managed data sets

- Table spaces only:
  - LOB – for Large OBjects
  - NUMPARTS – Used for partitioning
  - COMPRESS – Compress data in a table space. Compression is only done after LOAD or REORG, not as rows are inserted.
- Index only:
  - CLUSTER – Cluster table in a specific order
  - PARTITONED (and others similar) – Used for partitions
  - DEFER - whether the index is built during the execution of the CREATE INDEX statement. Primarily used for objects created with the UNIQUE keyword. It is possible to have an entry for an index space in the DB2 catalog, but not have the LDS created until data is inserted.
  - PADDED – If columns should be padded if not full
  - PIECESIZE - Specifies the maximum addressability of each piece (data set) for a secondary index. This can be in K, M, or G. Creates a new data set with an advancing LLQ for new pieces.

DB2 and VSAM sizes

DB2 page size	VSAM CI size V7/V8	VSAM physical block size V7/V8	blocks per track V7/V8	DB2 pages per tracks
4	4	4	12	12
8	4/8	4/8	12/6	6
16	4/16	4/16	12/3	3
32	4/32	4/16	12/3	1.5

DB2 V8 with DSVCI=YES for 32K pages - 1 cylinder CA



No data integrity issue, a page can not span volumes! This is what makes the BACKUP SYSTEM utility work without QUIESCE!

Lose 16K (2.2%) every 15th track, but we will give it back, see ZPARM section.

Since CISIZE is 4K extending row to the next cylinder is not an issues. This however could cause a data integrity problem if the next extent spans to a new volume (backup and striping issues)!

DB2 Data Set Naming Conventions

- ***catname*.DSNDB*x*.*dbname*.*psname*.*y*000*z*.*lnnn***
  - ***catname*** – HLQ
  - ***x* -** C (for VSAM cluster) or D (for VSAM data components)
  - ***dbname* -** DB2 database name (see next foil for some names)
  - ***psname* -** Table or index space name
  - ***y000z* -** I or J, which indicates the data set name used by REORG with FASTSWITCH.
    - Starting with DB2 9, the number (z) can be 1 or 2, making the qualifier either I0001, I0002, J0001, or J0002.
  - ***lnnn* -** l is A, B, C, D, or E, ***nnn*** is the data set number beginning with 001. Non partition data sets always start with A001, for partition data sets:
    - A001-A999 for partitions 1 through 999
    - B000-B999 for partitions 1000 through 1999
    - C000-C999 for partitions 2000 through 2999
    - D000-D999 for partitions 3000 through 3999
    - E000-E096 for partitions 4000 through 4096

dbname for DB2 cat/dir, temp, and sort

- DSNDB01 – DB2 directory. The DB2 system database that contains internal objects such as database descriptors and skeleton cursor tables.
- DSNDB04 – default database if specific database is not requested. This should very rarely be used. In many shops data sets containing this qualifier are considered throw away data sets. Verify with your DB2 group first if you want to delete these data sets.
- DSNDB06 – DB2 catalog. A collection of tables that contains descriptions of objects such as tables, views, and indexes.
- DSNDB07 – work file database used for DB2 functions that require sorts. This name is typically used for non Data Sharing environments. For Data Sharing environments one member can use this qualifier, however what is more typical is a name tied to a member, for example WORKDB1A as a work file for member DB1A only.

Typical naming conventions for DB2 BSDS, active and archive log data sets

- BSDS
  - Non Data Sharing: *ssid*.BSDS01 and *ssid*.BSDS02
  - Data Sharing: *dsgrp.ssid*.BSDS01 and *dsgrp.ssid*.BSDS02
- Active log data sets
  - Non Data Sharing: *ssid*.LOGCOPY1.DS*xx* and *ssid*.LOGCOPY2.DS*xx*
  - Data Sharing: *dsgrp.ssid*.LOGCOPY1.DS*xx* and *dsgrp.ssid*.LOGCOPY2.DS*xx*
- Archive log data sets (BSDS created first, then archive log data set)
  - Active log to archive:
    - Non Data Sharing: *ssid*.ARCGLOG1.**.A*xxxxxxx* and *ssid*.ARCGLOG2.**.A*xxxxxxx*
    - Data Sharing: dsgrp.*ssid*.ARCGLOG1.**.A*xxxxxxx* and *dsgrp.ssid*.ARCGLOG2.**.A*xxxxxxx*
  - BSDS (Boot Strap Data Set):
    - Non Data Sharing: *ssid*.ARCGLOG1.**.B*xxxxxxx* and *ssid*.ARCGLOG2.**.B*xxxxxxx*
    - Data Sharing: *dsgrp.ssid*.ARCGLOG1.**.B*xxxxxxx* and *dsgrp.ssid*.ARCGLOG2.**.B*xxxxxxx*

Which DB2 data sets should reside on the same disk?

- Table spaces and indexes can reside on the same disk so long as there are enough PAVs available. Otherwise separate table spaces from indexes.
- DB2 sort data sets (DSNDB07 equivalent) can reside on the same disk with table spaces and/or indexes so long as there are enough PAVs available. Otherwise separate DSNDB07 data sets to their own volumes.
- DB2 catalog and directory data sets should be placed on their own volumes. You can mix their table spaces and indexes so long as there are enough PAVs available.
- BSDS and active log data sets can reside on the same volume. Separate copy 1 and 2 of the BSDS and active log data sets onto separate disk, preferably on different disk controllers.
- Archive log and image copy data sets should be separated from all volumes above. Depending on your recovery situation you may need to separate archive logs from image copy data sets. Otherwise they can be placed together on the same volume.
- Most other DB2 data sets, PDSE, PDS, and sequential data sets can typically reside on the same volumes.
- It is generally a good idea to have data from one subsystem or Data Sharing group reside on separate volumes from other subsystem or Data Sharing groups.

DB2 Sizes and Limitations

- DB2 LDS user data sets CA do not exceed 1 cylinder.
  - If PRIQTY > 1 cylinder then CA is 1 cylinder
  - If PRIQTY<1 cylinder then CA is in tracks
- Maximum number of managed volumes at one time in a DB2 STOGROUP - 133 (DB2 limitation)
- Maximum size of a VSAM data set is 4GB unless it is defined with a data class that specifies extended format with extended addressability (dfp limitation). However:
  - Maximum simple or segmented data set size - 2 GB
  - Largest simple or segmented data set size - 64 GB (32 data sets * 2 GB size limit)
- Maximum size of a partition or LOB created with the LARGE keyword - 4 GB
- Maximum size of a partition or LOB created with the DSSIZE keyword - 64 GB (depending on page size). EF/EA data sets.
- CREATE INDEX with the PIECESIZE keyword can allocate smaller data sets
- Allocate objects on cylinder boundary
  - Improved SQL SELECT and INSERT performance
  - 2x improvement relative to track allocation

RBA and LRSN

- RBA (Relative Byte Address) – refers to the RBA of the active or archive log data sets, not of a table or index space. Typically used for non Data Sharing.
- LRSN (Log Record Sequence Number) – value derived from the store clock timestamp and synchronized across the members of a Data Sharing group by the Sysplex Timer. Provides the ability to work across Data Sharing members resulting in a common recovery point spanning the different RBAs of different members.

ZPARM options that influence allocation

MGEXTSZ - Sliding Secondary Allocation Size (DB2 V8) - page 1 of 7

- **Applies to DB2 managed pagesets**
- **Tries to avoid VSAM maximum extent limit errors**
  - Can reach maximum dataset size before running out of extents. Beware of heavily fragmented volumes, which may impede this feature. This is less of an issue in z/OS 1.7 where an LDS can have 7,257 extents. The sliding secondary rule can exceed 255 extents when ‘Extent Constraint Removal’ is turned on in z/OS 1.7.
  - Calculation for next extent is based on total space and not total extents because of such factors as extent consolidation.

- **Uses cylinder allocation**
  - Default PRIQTY
    - 1 cylinder for non-LOB table spaces and indexes
    - 10 cylinders for LOB table spaces

- **Can be used for:**
  - New pagesets: No need for PRIQTY/SECQTY values
  - Existing pagesets: Execute SQL to ALTER PRIQTY/SECQTY values to -1 plus schedule a REORG

MGEXTSZ - Sliding Secondary Allocation Size (DB2 V8) - page 2 of 7

- **Two sliding scales will be used depending on the maximum dataset size. The first 127 extents are allocated in increasing size, and the remaining extents are allocated based on the initial size of the data set:**
  - For 32 GB and 64 GB data sets, each extent is allocated with a size of 559 cylinders.
  - For data sets that range in size from less than 1 GB to 16 GB, each extent is allocated with a size of 127 cylinders.
- **Maximum dataset size determined based on DSSIZE, LARGE and PIECESIZE and defaults.**
- **BEWARE! If your Storage Administrator has set up a “LARGE” DB2 Storage Group, using this technique will probably not work well, unless you explicitly specify a large enough PRIQTY instead of DB2 implicitly specifying one.**
- **Advantages:**
  - Minimizes the potential for wasted space by increasing the size of secondary extents slowly at first
  - It prevents very large allocations for the remaining extents, which would likely cause fragmentation.
  - It does not require users to specify SECQTY values when creating and altering table spaces and index spaces.
  - It is theoretically possible to always reach maximum data set size without running out of secondary extents.
  - Particularly helpful for users of ERP/CRM vendor applications, which have many small data sets that can grow rapidly

4 Rules for Sliding Secondary

1. If PRIQTY is specified by the user, the PRIQTY value will be honored, otherwise the new default value as determined by either TSQTY, TSQTY*10 or IXQTY will be applied: 1 cylinder for non-LOB table spaces and indexes, and 10 cylinders for LOB table spaces.
2. If no SECQTY is specified by the user, the actual secondary quantity allocation will determined by maximum of 10% of PRIQTY, and the minimum of calculated secondary allocation quantity size using the slide scale methodology and 559 (or 127) cylinders depending on maximum DB2 dataset size. When a pageset spills onto a secondary dataset, the actual secondary allocation quantity will be determined and applied to the primary allocation. The progression will then continue. Prior to DB2 Version 8, the PRIQTY would have been used.
3. If SECQTY is specified by the user as 0 to indicate do not extend, This will always be honored. This condition will apply to DSNDB07 work files where many users set SECQTY to 0 to prevent work files growing out of proportion.
4. If SECQTY specified by the user is greater the 0, the actual secondary allocation quantity will be the maximum of the minimum of calculated secondary allocation quantity size using the slide scale methodology and 559 (or 127) cylinders depending on maximum DB2 dataset size, and the SECQTY value specified by the user. When a pageset spills onto a secondary dataset, the actual secondary allocation quantity will be determined and applied as the primary allocation. The progression will then continue. Prior to DB2 Version 8, the PRIQTY would have been used.

Rules 1, 2 and 3 above apply regardless of the ZPARM MGEXTSZ setting.

Rules 1, 2 and 3 apply to data sets created prior to DB2 Version 8.

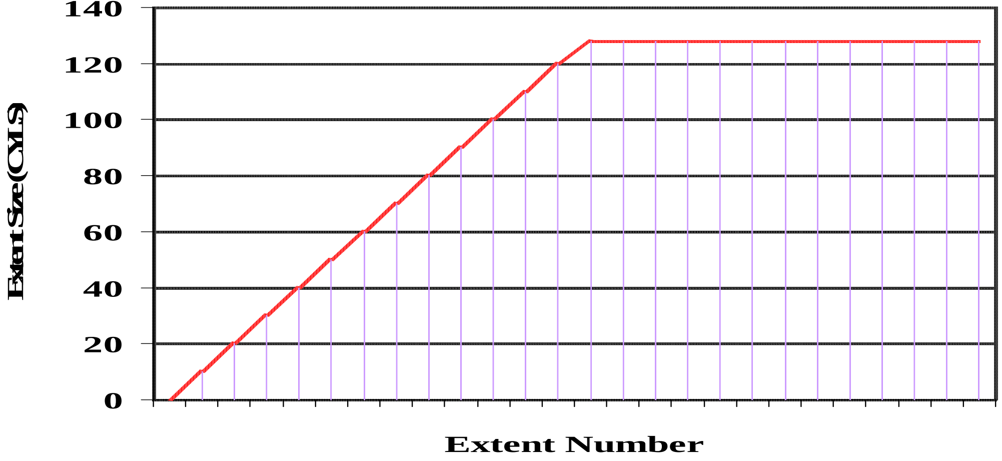
Rule 4 only applies when ZPARM MGEXTSZ is set to YES. The actual secondary allocation quantity applied will not be reflected in the Catalog. The primary allocation quantity and the actual secondary allocation quantities will never exceed DSSIZE and PIECESIZE.

MGEXTSZ - Sliding Secondary Allocation Size (DB2 V8) - page 4 of 7

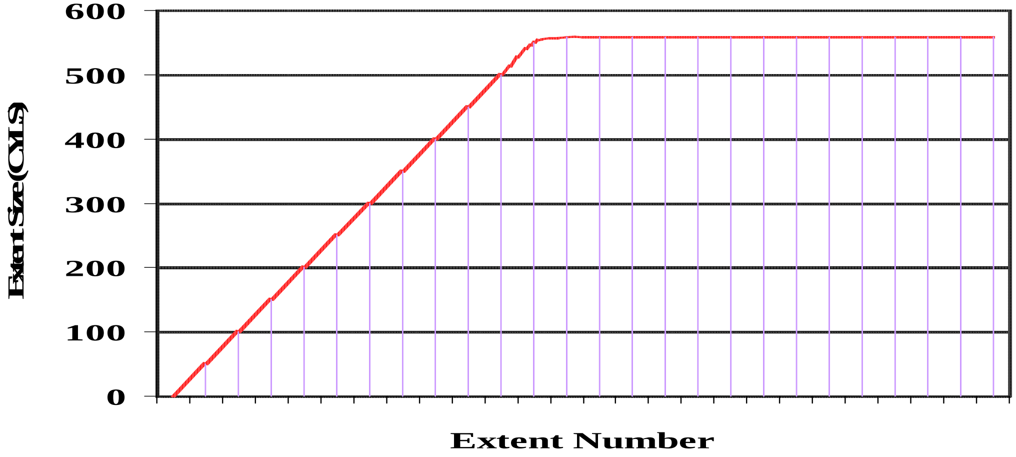
- If PRIQTY and/or SECQTY is not specified by the user then -1 will be recorded in the associated Catalog columns:
  - PQTY and SQTY in SYSTABLEPART
  - PQTY and SQTY in SYSINDEXPART.
- With MGEXTSZ NO/YES:
  - **YES – when allocating a new data set for a new piece (after A001) the primary and secondary allocation will be last size calculated by the sliding secondary of the previous piece. The maximum allocation will not exceed 127 cylinders for objects 16GB or below, and 559 cylinders for objects 32 or 64GB.**
    - ***For example, A001 hits the 2GB limit and DB2 now needs to create the A002 data set. A001's last allocation based on sliding scale was 79 cylinders, A002 will be created with a primary and secondary allocation of 79 cylinders.***
  - **NO – When specifying a value for PRIQTY and SECQTY the new data set (created after A001) will have the same allocation as A001 for primary and secondary and therefore take on the characteristics of A001's PRIQTY and SECQTY. If PRIQTY and SECQTY were not specified then the allocation works as documented in YES above.**

MGEXTSZ - Sliding Secondary Allocation Size (DB2 V8) - page 5 of 7

Sliding Scale for less than 1 GB to 16 GB



Sliding Scale for 32 GB and 64 GB



Maximum allocation of secondary extents

Max DS size in GB	Max Alloc in Cylinders	Extents to reach full size
1	127	54
2	127	75
4	127	107
8	127	154
16	127	246
32	559	172
64	559	255

Sample Allocations for A001 Segmented (assuming TSQTY=0 and IXQTY=0)

With MGEXTSZ=NO

```
PRIQTY           48
SECQTY           48

1st extent tracks . : 1
Secondary tracks  . : 1
```

With MGEXTSZ=YES

```
PRIQTY           48
SECQTY           48

1st extent tracks . : 1
Secondary tracks  . : 15
```

With MGEXTSZ=YES and DSVCI=YES. For 32K pages only -
we add 2.2% to the allocation:

```
PRIQTY           48
SECQTY           48

1st extent tracks . : 2
Secondary tracks  . : 16
```

With MGEXTSZ=NO or YES

```
PRIQTY           48
```

no secondary

```
1st extent tracks . : 1
Secondary tracks  . : 15
```

no primary

```
SECQTY           48

1st extent cylinders: 1
Secondary cylinders : 1
```

no primary and no secondary

```
1st extent cylinders: 1
Secondary cylinders : 1
```

```
PRIQTY           72000
```

no secondary

```
1st extent cylinders: 100
Secondary cylinders : 10
```

no primary

```
SECQTY           72000

1st extent cylinders: 1
Secondary cylinders : 100
```

```
PRIQTY           72000
SECQTY           0

1st extent cylinders: 100
Secondary cylinders : 0
```

Sample Allocations for A001 Segmented (assuming TSQTY=3600 (5 cylinders) and IXQTY=3600)

With MGEXTSZ=NO

```
PRIQTY          48
SECQTY          48

1st extent tracks . : 1
Secondary tracks  . : 1
```

With MGEXTSZ=YES

```
PRIQTY          48
SECQTY          48

1st extent tracks . : 1
Secondary tracks  . : 15
```

With MGEXTSZ=YES and DSVCI=YES. For 32K pages only -
we add 2.2% to the allocation:

```
PRIQTY          48
SECQTY          48

1st extent tracks . : 2
Secondary tracks  . : 16
```

With MGEXTSZ=NO or YES

```
PRIQTY          48
```

no secondary

```
1st extent tracks . : 1
Secondary tracks  . : 15
```

no primary and no secondary

```
1st extent cylinders: 5
Secondary cylinders : 1
```

```
PRIQTY          72000
```

no secondary

```
1st extent cylinders: 100
Secondary cylinders : 10
```

```
no primary
SECQTY          72000

1st extent cylinders: 5
Secondary cylinders : 100
```

With MGEXTSZ=NO

no primary

```
SECQTY          48

1st extent tracks . : 75
Secondary tracks  . : 1
TRACKS/CA-------------1
```

See PK05644 – preformat up
to 2 cylinders even though allocation
is in tracks. .

With MGEXTSZ=YES

no primary

```
SECQTY          48

1st extent cylinders: 5
Secondary cylinders : 1
TRACKS/CA-------------15
```

Sliding Secondary FAQ

- When using sliding secondary, after REORG without REUSE, do allocations start as new on the sliding scale as it did during allocation prior to the REORG, or is allocation based on current size and extents?
  - Answer: After REORG without REUSE, allocations start from the beginning on the sliding scale. Size and extents are not factored in. If the goal is to reduce extents or size, then you must execute an ALTER to space prior to the REORG, which is the same operation were you not using sliding secondary.

TSQTY and IXQTY (DB2 V8) DB2 V6 and V7 by the PTF for APAR PQ53067

- **Specifies the amount of space in KB for the primary space allocation quantity for DB2-managed table spaces (TSQTY) and indexes (IXQTY) which are created without the USING clause.**
- **It uses cylinder allocation. The default values are set to 0, which means in DB2 V8:**
  - Default PRIQTY and SECQTY
    - 1 cylinder for non-LOB tablespaces and indexes
    - 10 cylinders for LOB tablespaces
- **Autonomic selection of data set extent sizes with a goal of preventing extent errors before reaching maximum data set size - use with sliding secondary allocation.**
- **Prevents heavy over-allocation and waste of excessive space.**
- **Can also result in better performance of mass inserts, prefetch operations, as well as LOAD, REORG and RECOVER utilities.**
- **PRIQTY honored if used.**

DSVCI - Support for VSAM Control Interval (CI) Greater than 4K (DB2 V8) - page 1 of 2

- **Support for CI sizes of 8, 16, and 32 K table spaces. For indexes, only 4K pages are supported. Starting with DB2 9, indexes can be 8, 16, or 32 K as well.**
- **Requires ZPARM value DSVCI in DSN6SYSP be set to YES, which is the default. It is set during the DB2 install in panel DSNTIP7 under VARY DS CONTROL INTERVAL.**
- **Supported for DB2 managed as well as user managed DB2 table spaces. DB2 install procedure provides JCL that will convert user defined DB2 catalog table spaces to proper CI sizes during ENFM. User managed table spaces require manual IDCAMS CI change.**
- **Activated in NFM for corresponding non LOB page sizes of table spaces, which will not take effect until after a LOAD or REORG of the table space. Note – this issue is resolved in DB2 9. Some alternatives prior to DB2 9 for LOBs:**
  - **If it is a LOB with LOG YES, one way to switch CI size is to use the RECOVER utility. For LOG NO LOBs, you can still use COPY and RECOVER after switching the LOB to access r/o and quiescing updaters.**
  - **STOP the LOB, IDCAMS EXPORT, then IMPORT with NEWNAME, change the names, then START the LOB.**
  - **DSN1COPY into a new object.**
- **New CI sizes:**
  - Reduce integrity exposures
  - Relieves some restrictions on concurrent copy (of 32K objects) and the use of striping (of objects with a page size > 4K).
  - Potentially reduce elapsed time for table space scans
  - I/O bound executions benefit with using larger VSAM CI sizes, including COPY and REORG.

DSVCI - Support for VSAM Control Interval (CI) Greater than 4K (DB2 V8) - page 2 of 2

- **Striped VSAM data sets are in extended format (EF) and internally organized so that control intervals (CIs) are distributed across a group of disk volumes or stripes. A CI is contained within a stripe.**
- **Increase your non 4K buffer pool sizes to accommodate new CI sizes.**
- **Some test results:**
  - 16K page measurement with 16K instead of 4K CI
    - +40% for non EF (Extended Format) datasets
    - +70% for EF datasets
    - EF getting nearly equivalent to non EF in data rate performance
  - Some table spaces may have negative results - test and verify
- **LISTCAT results (SPACE does not change for 8K and 16K table spaces, just 32K):**
  - after CREATE TABLESPACE to a 4K buffer pool with PRIQTY 72000:

```
•CISIZE--------------4096
•PHYREC-SIZE---------4096
•PHYRECS/TRK-----------12
•SPACE-TYPE------CYLINDER
•SPACE-PRI------------100
```

  - after CREATE TABLESPACE to a 32K buffer pool with DSVCI=YES:

```
•CISIZE-------------32768
•PHYREC-SIZE--------16384
•PHYRECS/TRK------------3
•SPACE-TYPE------CYLINDER
•SPACE-PRI------------103
```

PRIQTY 72000

Notice, DB2 adds the 2.2% overhead for you!

DB2 V7 – 32K,
PRIQTY 72000

```
CISIZE--------------4096
PHYREC-SIZE---------4096
PHYRECS/TRK-----------12
SPACE-TYPE------CYLINDER
SPACE-PRI------------100
```

SEQCACH

- Determine for prefetch if BYPASS or SEQ(uential) is used for cache. Although BYPASS is an acceptable ZPARM value, we no longer bypass cache for the DS8000, ESS, or RVA. 3990 was the last controller to honor BYPASS.
- BYPASS for DS8000, ESS, or RVA will use sequential detect while SEQ will use explicit command.
  - SEQ (explicit) puts tracks on the accelerated list and starts prestaging for next I/O. Operations are done on extent boundaries or stage group (an internal construct, depending on rank dimensions). Explicit reacts faster and can end sooner than using the detect mechanism (BYPASS).
  - BYPASS (sequential detect) stages data to the end of a stage group (an internal construct, depending on rank dimensions).
- Recommendation – change SEQCACH to SEQ. NOTE – BYPASS is the default.

SEQPRES

- Utility cache option.
- Recommendation – set to YES. Default is NO.

SVOLARC - Single VOLume ARChive - PQ49360

- **DB2 allocates up to 15 volumes for archive log data sets to allow for space extension onto other volumes.**
  - Problem: For some SMS users who use guaranteed space for archive log data sets, DB2 may request primary allocation of up to 15 volumes (which it may not need) thereby causing space related problems.
  - Symptoms include: Message DSNJ103I with ERROR STATUS=970C0000, SMS REASON CODE=00004336 or REASON CODE=00004379
  - Solution: Consider changing to SVOLARC=YES if you want SMS to only allocate one volume to avoid this situation when using guaranteed space.

UNIT and UNIT2 - for DB2 Archive Log Data Sets

- **It can be set to the same UNIT type. However, beware of the following:**
  - Problem: many installations set UNIT and UNIT2 to a disk unit, VTS or use TMM.
  - Storage Administrator sets ACS routines whereby:
    - ARCHLOG1 is allocated to a Storage Group that DFSMShsm “sweeps” hourly to ML1 or ML2.
    - ARCHLOG2 is allocated to a Storage Group that DFSMShsm migrates once every 24 hours.
  - If they stay on ML1 or eventually migrate to ML2, their final residence can land on the same volume which can be a single point of failure. When ML2 tapes are set to MOD (typical scenario):
    - Tape can tear and become unusable
    - Tape can be misfiled by Operator or Tape Librarian
    - Tape is maliciously lost or destroyed
  - Solutions:
    - Allow DFSMShsm to backup the archive data sets before migrating
    - Duplex your ML2 tapes – may be an issue after tapes are recycled, then may result in a single point of failure again down the line
    - Transmit a copy of at least one set to another site
    - Split UNIT and UNIT2 between 2 device types, e.g. one to disk, one to tape
    - Use ABARS to copy one of the archives from ML1 or ML2
  - NOTE: This can happen to any dual copied data set, including image copies. Dual copy data sets residing on VTS or when using TMM can also have a single point of failure. Discuss your requirements with your Storage Administrator,

Some other issues

I need to understand

before we go on

How did this happen? I have 90 cylinders primary, 12 cylinders secondary, 63 extents, but 25530 tracks allocated?

- **CREATE TABLESPACE PRIQTY 64800 SECQTY 8640**
- **After some time and space ... ALTER TABLESPACE SECQTY 20000**
- **Add more rows, trip extents**
- **No REORG afterwards**
- **DB2 information correct**
- **MVS information is not!**
- **CYL(90,12) with 63 extents**
  - should be 12510 tracks
  - exception - fragmentation
  - BEWARE Storage Administrators! What you see is not always what you get!

```
  ISPF 3.4 listing                    Tracks  XT  Device
---------------------------------------------------------
RI1.DSNDBD.A020X7KR.CKMLPR.I0001.A001  25530  63  3390

LISTCAT output:
ALLOCATION
        SPACE-TYPE------CYLINDER     HI-A-RBA------1254850560
        SPACE-PRI-------------90     HI-U-RBA------1113292800
        SPACE-SEC-------------12

primary+(secondary*(extents-1))=space
90+(12*(63-1))=834 cylinders, 12510 tracks, not 25530!
```

This case is not a disk fragmentation issue. After the ALTER, DB2 knows the allocation converted to CYL(90,28). However, MVS still thinks it is CYL(90,12) until redefined by a process such as REORG without a REUSE. PRIQTY and SECQTY is actually what DB2 uses, not CYL(90,12).

I understood the last chart, but how did I actually get LESS space then I requested?

- **When your Storage Administrator has set up a Data Class with the following attributes:**
  - **The space constraint relief attribute on, and with the request for a percentage for space reduction, your data set allocated can actually be less than requested . This can happen if your volume does not have enough space.**
  - E.g., 4K object, created with PRIQTY 72000 (100 cylinders), the Data Class space constraint was set up to allow 10% reduction, you had one volume with 92 cylinders remaining. Results:
    - The DB2 catalog will still show the equivalent of PRIQTY 72000.
    - The actual MVS allocation will be 90 cylinders or the equivalent of PRIQTY 64800.

My Storage Administrator told me they recreated my DB2 data sets with high extents. They are now as low as 1 extent per data set. Are there any issues?

- **Storage Administrators have a number of ways of causing extent reduction, thereby potentially bringing back a data set in 1 extent, among them:**
  - DFSMShsm MIGRATE and RECALL functions
  - DFSMSdss COPY, or DUMP and RESTORE functions
  - DEFRAG with the CONSOLIDATE keyword
- **Using Such functions as DFSMSdss COPY may be much faster than running REORGs.**
- **Do you have SQL that uses the DB2 catalog to report on extents? This can potentially be a problem. You may be redoing REORGs unnecessarily since the move was done outside of DB2 and DB2 does not know about it. *** NOTE – the EXTENTS column in the RTS will only be updated once an update or applicable utility is run for the object. A simple start after extent reduction or a read based on SELECT will not update the EXTENTS column (same issue as the catalog).**
- **Do you use high extents as a tool to review issues with clustering indexes? This can potentially be a problem. Review CLUSTERRATIOF more closely.**

- ***** NOTE – Using procedures outside of DB2 for consolidation of space does not remove pseudo deleted data. You must use DB2 utilities in order to remove pseudo deleted data.**

Thinking about SMS managing all of your DB2 volumes (good idea!) and for STOGROUP specifying VOLUMES(“*”) (another good idea!)?

- Creating a STOGROUP and specifying VOLUMES(“*”) allows SMS to choose a volume for allocation.
- Here is the issue at hand – at times a Storage Administrator sets up the SMS ACS routines to check for specific volume types for allocation. Much of the time an * is not provided for the test. Without the ACS routine having an * as part of a valid test, your allocation will fail.
- If you will be creating STOGROUPs with VOLUMES(“*”) make sure your Storage Administrator adds an * to their test of device allocations in the ACS routines.
- With DB2 9, For the DB2 STOGROUP, you can specify SMS DATACLAS, STORCLAS, or MGMTCLAS and not specify the VOLUMES attribute.

DB2 9 and SMS Classes

- Starting in DB2 9, you can use SQL with CREATE STOGROUP or ALTER STOGROUP to specify a DATACLAS, STORCLAS, and/or MGMTCLAS.
- VOLUME clause is now optional: it can be omitted if any of the DFSMS classes are specified.

Differences Between DB2 STOGROUP and SMS Storage Group

DB2 STOGROUP	SMS Storage Group
Different STOGROUPs can share the same disk volume(s)	One disk volume can only belong to one SMS Storage Group
VOLSERs are specific	Recommendation - code VOLUMES("*") to allow for SMS management. Avoid Guaranteed Space and specific VOLSERs where possible since this defeats the purpose of SMS.
SYSIBM.SYSVOLUMES has a row for each volume in column VOLID	When created with VOLUMES("*") SYSIBM.SYSVOLUMES has an * for column VOLID
Limited to management of 133 volumes	No volume limit
Volume selection based on free space	Volume selection based on SMS algorithms

DB2 9 – Indexes - Size and Compression

- Starting with DB2 9, indexes can now be 4K, 8K, 16K, or 32K.
- Dictionaryless, software managed index compression at the page level.
- Indexes are compressed at write time, decompressed at read time. They are uncompressed in the buffer pools.
- Compression of indexes for BI workloads
  - Indexes are often larger than tables in BI
- Solution provides page-level compression
  - Data is compressed to 4 KB pages on disk
  - 32/16/8 KB pages results in 8x/4x/2x disk savings
  - No compression dictionaries – compression on the fly
- DSN1COMP can be used for indexes as well starting with DB2 9.

What should I discuss with my Storage Administrator?

Issues I should discuss with my Storage Administrator about my DB2 environment:

- Who is the disk manufacturer for the volumes I use?
- Do I use ESCON or FICON and at what data transfer speed?
- What models of disk do I use?
- What type of 3390 do I emulate?
- How much disk do I have for each environment (# of volumes, and total GB)?
- How much cache does each disk box contain?
- if we buy more disk cache, how much performance will I gain?
- What type of RAID is my disk and what does it mean to me?
- What features are turned on for my disk?
  - PAV (if available, dynamic, static, or hyper)
  - FlashCopy (version 1 or 2)

ht.Now.

Issues I should discuss with my Storage Administrator about my DB2 environment:

- Do you VSAM stripe my heavily sequential DB2 data sets (e.g. active logs)?
- Do you sequential stripe my applicable disk data sets that support my VSAM striped data sets?
- Do you sequential stripe my disk archive log data sets if on DB2 9?
- Do you compress my archive log data sets if on DB2 9?
- How often do you DEFRAG my volumes and based on what criteria?
- How well are my disks doing when reviewing RMF data during peak periods?
- For what DB2 LDSes do I have EF turned on (be careful if using FICON!)?
- If I am on z/OS 1.7 or above, in the SMS Data Class, is ‘Extent Constraint Removal’ set to YES to exceed the 255 extent limit? Does it make sense to increase the limit?
- Is space constraint relief being used for my data sets? Include the advantage of 5 extent rule for allocations.

Issues I should discuss with my Storage Administrator about my DB2 environment:

- Are my data sets created with the Guaranteed Space attribute? If so, why? If you do have the attribute on, mention to your Storage Administrator that although on multi volume data sets the primary allocation is propagated to additional volumes, for DB2 managed data sets the secondary is propagated not the primary.

- Do you migrate any of my DB2 or related data sets, and if you do what are the Management Class attributes?

- Do I use the SMS Management Class to expire any of my data, such as my archive logs or image copies?
- If I archive my DB2 LDSes, what is my ZPARM value for RECALL and RECALLD?

- Do you full volume backup my volumes, and if yes, why?

- Do you incrementally backup my volumes, and if yes, why?

- In the SMS Storage Group, what are the values for HIGH and LOW and why were they set to those numbers?

- Do my DB2 data sets use extended Storage Groups or overflow volumes? If so, what are the VOLSERs? If we FlashCopy our volumes, we need to make sure these are included as well.

Issues I should discuss with my Storage Administrator about my DB2 environment:

- If I have more than one disk controller, are my active log data sets segregated onto separate controllers for availability? Do you use the SMS Separation profile to accomplish this?

- For my archive log data sets, what units do I write to?
- How long do I keep them for? Is this a realistic number?
- What unit types do I use for my archive data sets?
- If I write to tape, what type of tape do I use? What does this tape type really mean to me? Also, what is the tape capacity?
- Is there a better technology to write my archive log data set to?

- How do you guarantee that my dual copied BSDS and archive log data sets do not wind up on the same volumes? Same question if you dual software copy your image copy data sets.

- What Storage Groups do I write to?
- What are their names?
- How many volumes do I have in each Storage Group?

Issues I should discuss with my Storage Administrator about my DB2 environment:

- Are the specs different for my different Storage Groups?

- How can I tell how much space I have available in each of my Storage Groups?

- Do you make sure I have enough space in my Storage Groups or do I?

- Are you seeing write bursts in the RMF Cache Volume Detail report? If so, I can probably relieve this situation.

- Are you using some type of disk migration tool such as Piper, TDMF, or FDR/PAS?

- How are we handling excessive extents? Will you run processes to reduce them or do I?

Issues I should discuss with my Storage Administrator about my DB2 environment:

- If my data sets are migrated, does it get migrated to disk or tape? If tape, how long will it take to retrieve a data set? Does the tape contain multi data sets? Is the tape manually mounted or automatic (robotic)?
- How often do the Storage Administrators and/or performance team review RMF data for DB2 disk? Is it often enough? Can we sit down and review the information together and have the reviewer explain what may be a potential problem?
- Be specific when discussing ‘the catalog’. Your Storage Administrator is thinking about the ICF catalog, not the DB2 catalog.
- For your DB2 LDSes, make sure your Storage Administrator does not set a value for "Volume Count“ or "Add'l Volume Amount“ in the Data Class panel of ISMF. We want DB2 to be able to allocate LDSes to dfp limitations, not artificial limitations set in SMS.

What should I discuss with my Storage Administrator?

- Who manufactures the tape units I use and what models do I create my tapes on?
- Do I create my tape data sets on an ATL or virtualized tape?
- If I use virtualized tape, are all of my tapes being created at one site or automatically copied via VTS/PtP or VE/Grid to an alternate site?
- For virtualized tape or ATL, do you tape copy any of my data sets from one site to another using an alternative method for immediate availability in case of a disaster? For example, do you XMIT any of my tapes data sets, or duplex my data sets residing on hsm tapes and send them offsite?
- I understand that for virtualized tape I can create logical tape volumes in sizes 400, 800, 1000, 2000 or 4000 MB. How do I specify the different SMS Data Classes to use the different sizes?
- What is the physical tape size that I use?

What should I discuss with my Storage Administrator?

- If I am not using virtualized tape, do you stack my data sets on tape? If so, what mechanism do you use?
- If I am not using virtualized tape and when I create my data sets I only use a small portion of the tape. What happens to the rest of the tape? Do I get charged for the entire tape?
- How many tape drives are available for me to use at any one time? Are these tape drives always available?
- If I use tape virtualization, is the disk and cache I write to as an intermediate stage large enough for my requirements?
- If I use tape virtualization, are my tape data sets created as Preference level 0 or 1? Have a discussion with your Storage Administrator regarding how long your data should stay on the TVC.

What should I discuss with my Storage Administrator?

- Do you physically pool my volume? If you do, what do you pool and why?

- Do you selectively Dual Copy some of my data sets? If you do, which ones and why?

- When I create dual software copies of tape data sets, how can I be sure they will not wind up on the same physical tape?
- Which flavor of Copy Management is active?

- Where does it make most sense to create your DB2 related objects? Virtualized tape or non, perhaps the use of TMM or hsm ML2? Are these tapes copied to another site or duplexed?

Things that influence my DB2 allocation

- DB2 or user managed data sets
- Volume fragmentation
- z/OS 1.5 and above – extent consolidation
- z/OS 1.7 and above – ability to exceed 255 extents for LDS
- Size of CA
- Use of parameters such as PIECESIZE, LARGE, and DSSIZE
- Use of ZPARM for:
  - DSVCI
  - MGEXTSZ (sliding secondary)
  - Use of TXQTY/IXQTY

Things that influence my DB2 allocation

- Use of Extended Format (EF) data sets
- Use of SMS Data Set Separation Profile
- Use of SMS Data Class for:
  - Attributes
  - Space constraint relief
  - Multi volume allocations
- Use of SMS Storage Class for:
  - Guaranteed Space attribute
- Use of SMS Management Class for:
  - Expiration date for data sets
  - Migration and backup of data sets

Things that influence my DB2 allocation

- Use of SMS Storage Group for:
  - – HIGH and LOW values
  - – Use of extend or overflow Storage Groups
  - – Reasons for allocation on volume – primary, secondary, tertiary, or rejected

Storage related items that effect my DB2 performance

- Type of disk
- Use of FICON or ESCON as well as channel transfer rate
- PAV (this also depends on if static or dynamic)
- Priority I/O Queueing
- Size of disk cache
- Number of paths for devices
- Disk volume/address architecture. Not “front loading the box”
- VSAM data striping
- Sequential data striping
- Allocation in tracks instead of cylinders (track penalty)
- MIDAW
- CI size (ZPARM DSVCI)
- Migration/recall of data
- Use of data in cache (ZPARMs SEQCACH, SEQPRES)
- FlashCopy V2 for CHECK INDEX utility
- Write bursts and use of buffer pools
- Utilities and usage of archive log data sets on tape are dependent of hardware and software which influence run times.