IBM DB2 Utility Update: Las mejores prácticas (best practices)

Bryan F. Smith, bfsmith@us.ibm.com

TAKE BACK CONTROL

Abstract

- This presentation reviews the features of the DB2 Version 8 for z/OS Utilities, discusses the features of DB2 9 for z/OS Utilities, including the use of the System z9 Integrated Information Processor (IBM zIIP), reviews a set of performance, availability, and usability best practices when using the utility suite, and previews the futures of the IBM DB2 Utilities. Upon completion of this session, the attendee, whose skill level may range from low to high, will be able to identify new utility functions and apply them in their shop.

Agenda

- DB2 V8 for z/OS Utilities
- IBM's Unload Products
- Best practices
- A scenario of best practices
- zIIP (IBM System z9 Integrated Information Processor)
- DB2 9 for z/OS Utilities
- DB2 Utilities futures

DB2 V8 for z/OS Utilities

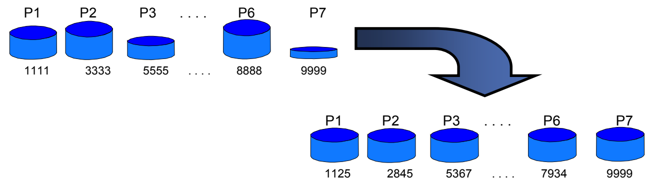
TAKE BACK CONTROL

Version 8 (1 of 3)

- New utilities BACKUP SYSTEM and RESTORE SYSTEM
- Delimited data support for LOAD and UNLOAD

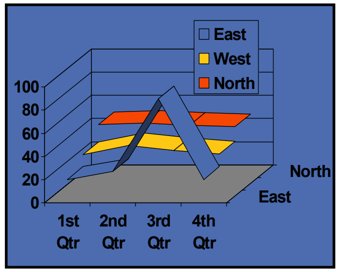
```
"Smith, Bob",4973,15.46
"Jones, Bill",12345,16.34
"Williams, Sam",452,193.78
```

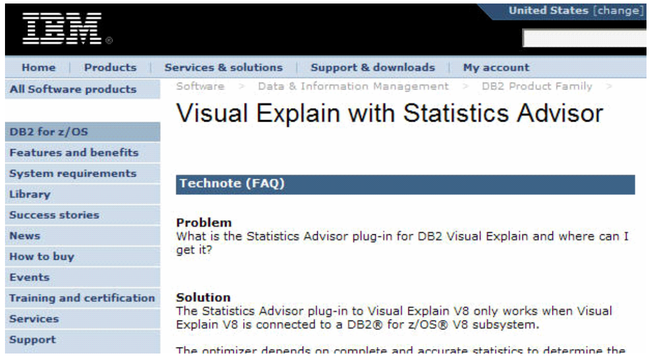
- New defaults for better "out of the box" performance
  - SORTKEYS for LOAD/REORG/REBUILD
  - SORTDATA for REORG
    - SORTDATA now allowed for 32K records with DFSORT
    - APAR PK18059 allows SORTDATA NO
- REORG SHRLEVEL NONE/REFERENCE REBALANCE



Version 8 (2 of 3)

- Non-uniform statistics on non-indexed columns
  - Current technique is separate DSTATS program
  - Significant performance improvements possible
  - Collected with the FREQVAL keyword on a specified group of columns (COLGROUP)





Visual Explain with Statistics Advisor

Technote (FAQ)

Problem
What is the Statistics Advisor plug-in for DB2 Visual Explain and where can I get it?

Solution
The Statistics Advisor plug-in to Visual Explain V8 only works when Visual Explain V8 is connected to a DB2® for z/OS® V8 subsystem.

The optimizer depends on complete and accurate statistics to determine the

- HISTORY statistics without updating main statistics -- “UPDATE NONE HISTORY ALL”
- REORG SHRLEVEL CHANGE allow DISCARD
- REORG SHRLEVEL REFERENCE catalog tables with links
- Online Concurrent Copy support for 32K pages

Version 8 (3 of 3)

- DFSORT
  - V8 removes the pre-req for an external sort product
  - Utilities only use the DFSORT package: SORT, MERGE functions only
    - Aim to improve reliability and performance
    - DFSORT code is part of a standard z/OS install
    - DB2 Utilities Suite has a licensed API to use DFSORT
    - Must have access to R14 DFSORT or V1R5 DFSORT (or higher) plus APAR PQ68263 applied
  - If your primary sort product is not DFSORT, then informational APAR II14047 (periodically updated) is <u>mandatory reading</u>

    http://www.ibm.com/support/docview.wss?rs=0&uid=isg1II14047

V8 Post GA

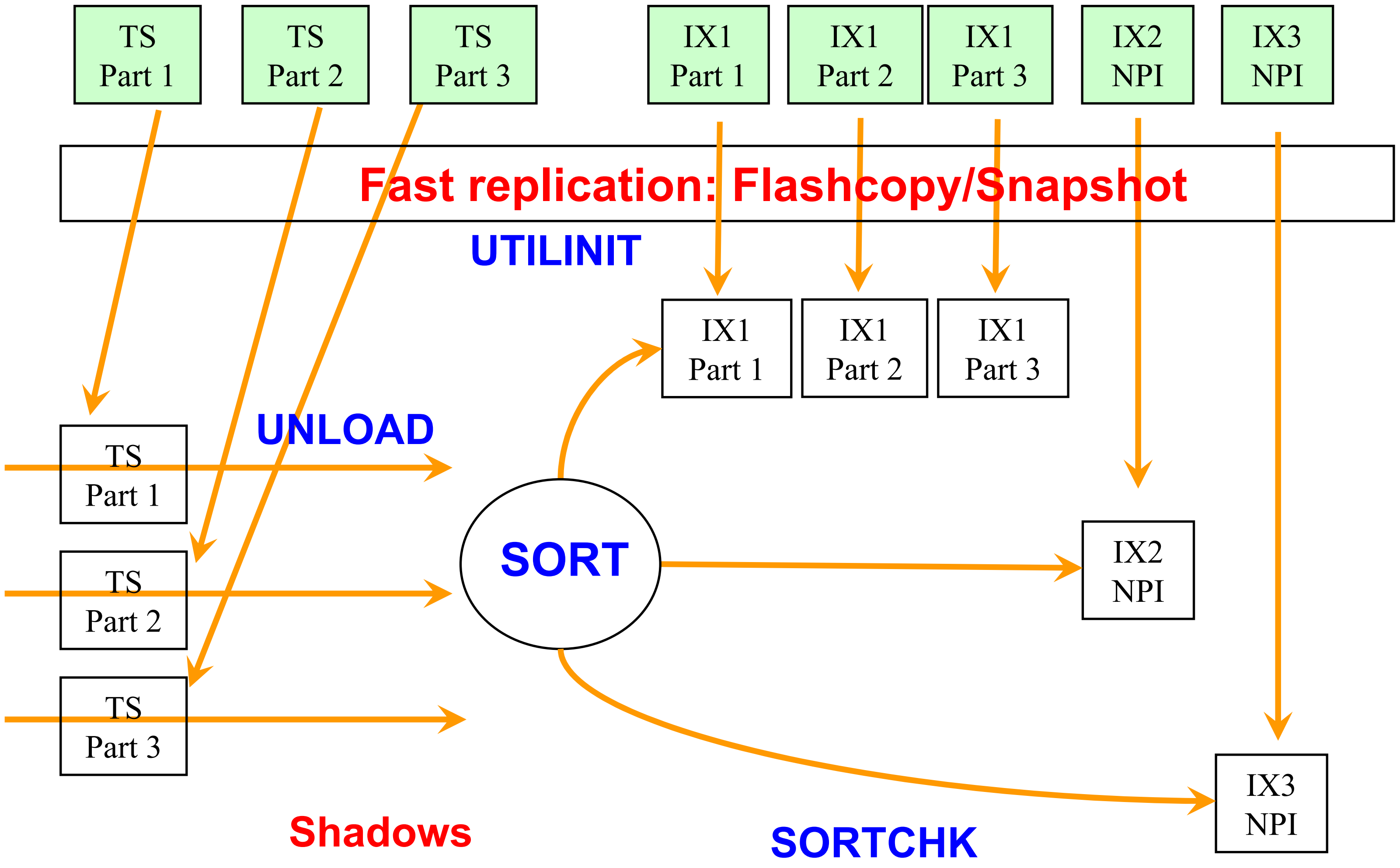
- *CHECK INDEX SHRLEVEL CHANGE*
  - APARs PQ92749 (DB2 base) and PQ96956 (Utility Suite) *(PTF avail for V8)*
- *Cross Loader support for > 32K LOBs*
  - APAR PQ90263 *(PTFs avail for V7/V8)*
- *LOB Handling for LOAD/UNLOAD using File Reference Variables*
  - APAR PK22910 *(PTFs avail for V7/V8)*
- *Automatically display CLAIMERS when REORG receives resource unavailable*
  - APAR PK00213 *(PTFs avail for V7/V8)*
- *zIIP-enablement for index maintenance*
  - See InfoAPAR II14219 *(PTFs avail for V8)*
- *LOAD via BATCHPIPE*
  - APAR PK34251 and z/OS APAR PK37032 *(PTF avail for V8)*
- *Switch to UTRW during UTILTERM for REORG SHRLEVEL CHANGE*
  - APAR PK34441 (PTF avail for V8)
- *Reduce switch phase time on table space with COPY NO indexes*
  - APAR PK35390 *(PTFs avail for V7/V8)*

Covered in subsequent slides

CHECK INDEX SHRLEVEL CHANGE

- Current CHECK INDEX causes data and indexes are unavailable for update for the duration
- CHECK INDEX SHR CHG different design than REORG SHR CHG
- Claim as reader for target data and indexes
- Create shadow datasets
  - Same dataset naming convention as REORG SHR CHG
  - Cannot run CHECK INDEX SHR CHG on two logical parts of NPI
- Drain writers for target data and indexes
- Copy datasets with DFSMSdss' ADRDSSU with FCNOCOPY to shadows
  - Uses dataset-level FlashCopy2 if available
  - Else, traditional media copy – still smaller r/o outage than SHR REF
- After logical complete for datasets,
- Dedrain target data and indexes
  - Run parallel check index on shadow data and indexes
    - Same parallel design as REBUILD INDEX
- At utilterm delete shadow datasets when DB2 managed

Checking all indexes in parallel



Cross Loader support for > 32K (rows with) LOBs

- Architectural limits within LOAD/UNLOAD did not allow for a record greater than 32K to be loaded or unloaded
- New buffering scheme for LOB values to bypass the 32K limit
- Will be constrained by region size
- Cross Load of 2GB LOBs will still not be possible
- Cross Loader will also allow for conversion between CLOBs and DBCLOBs (not currently supported when loaded from file)

LOB Handling for LOAD/UNLOAD using File Reference Variables

- Requirement is to move LOBs from one z/OS system to another z/OS system
- Need to support millions of rows
- Typical LOB sizes are 25K, 200K, 1MB
- Need to allow user to limit LOAD at target with WHEN clause
- LOB column values will be stored as separate PDS member, PDS/E member, or HFS directory member.
- LOB column values from each row will have identical member names in each PDS, PDS/E, or HFS
- Data set name stored in output record
- Design fits well with File Reference Variables where LOB values are in individual datasets

LOB Handling for LOAD/UNLOAD using File Reference Variables…

- LOAD is changed to allow an input field value to contain the name of file containing a LOB column value. The LOB is loaded from that file.

```
//SYSREC    DD *
"000001","UN.DB1.TS1.RESUME(AI3WX3JT)","UN.DB1.TS1.PHOTO(AI3WX3JT)"
"000002","UN.DB1.TS1.RESUME(AI3WX5BS)","UN.DB1.TS1.PHOTO(AI3WX5BS)"
"000003","UN.DB1.TS1.RESUME(AI3WX5CC)","UN.DB1.TS1.PHOTO(AI3WX5CC)"
"000004","UN.DB1.TS1.RESUME(AI3WX5CK)","UN.DB1.TS1.PHOTO(AI3WX5CK)"
```

```
 LOAD DATA FORMAT DELIMITED
      INTO TABLE MY_EMP_PHOTO_RESUME
           (EMPNO     CHAR,
            RESUME    VARCHAR CLOBF,
            PHOTO     VARCHAR BLOBF)
```

new syntax

LOB Handling for LOAD/UNLOAD using File Reference Variables…

- UNLOAD is changed to store the value of a LOB column in a file and record the name of the file in the unloaded record of the base table.

```
TEMPLATE LOBFRV1 DSN 'UN.&DB..&TS..RESUME' DSNTYPE(PDS) UNIT(SYSDA)
TEMPLATE LOBFRV2 DSN 'UN.&DB..&TS..PHOTO'  DSNTYPE(PDS) UNIT(SYSDA)
```

Creates PDS with member for each LOB

```
UNLOAD DATA
FROM TABLE DSN8910.EMP_PHOTO_RESUME
(EMPNO  CHAR(6),
 RESUME VARCHAR(255)  CLOBF  LOBFRV1,
 PHOTO  VARCHAR(255)  BLOBF  LOBFRV2) DELIMITED
```

new syntax

Output:

```
"000001","UN.DB1.TS1.RESUME(AI3WX3JT)","UN.DB1.TS1.PHOTO(AI3WX3JT)"
"000002","UN.DB1.TS1.RESUME(AI3WX5BS)","UN.DB1.TS1.PHOTO(AI3WX5BS)"
"000003","UN.DB1.TS1.RESUME(AI3WX5CC)","UN.DB1.TS1.PHOTO(AI3WX5CC)"
"000004","UN.DB1.TS1.RESUME(AI3WX5CK)","UN.DB1.TS1.PHOTO(AI3WX5CK)"
…
```

Automatically display CLAIMERS when REORG receives resource unavailable

- Before

```
DSNUGUTC -  REORG TABLESPACE DB1.TS1 COPYDDN(SYSCOPY) SHRLEVEL NONE STATISTICS
DSNUGBAC - RESOURCE UNAVAILABLE
           REASON 00C200EA
           TYPE 00000200
           NAME DB1.TS1
```

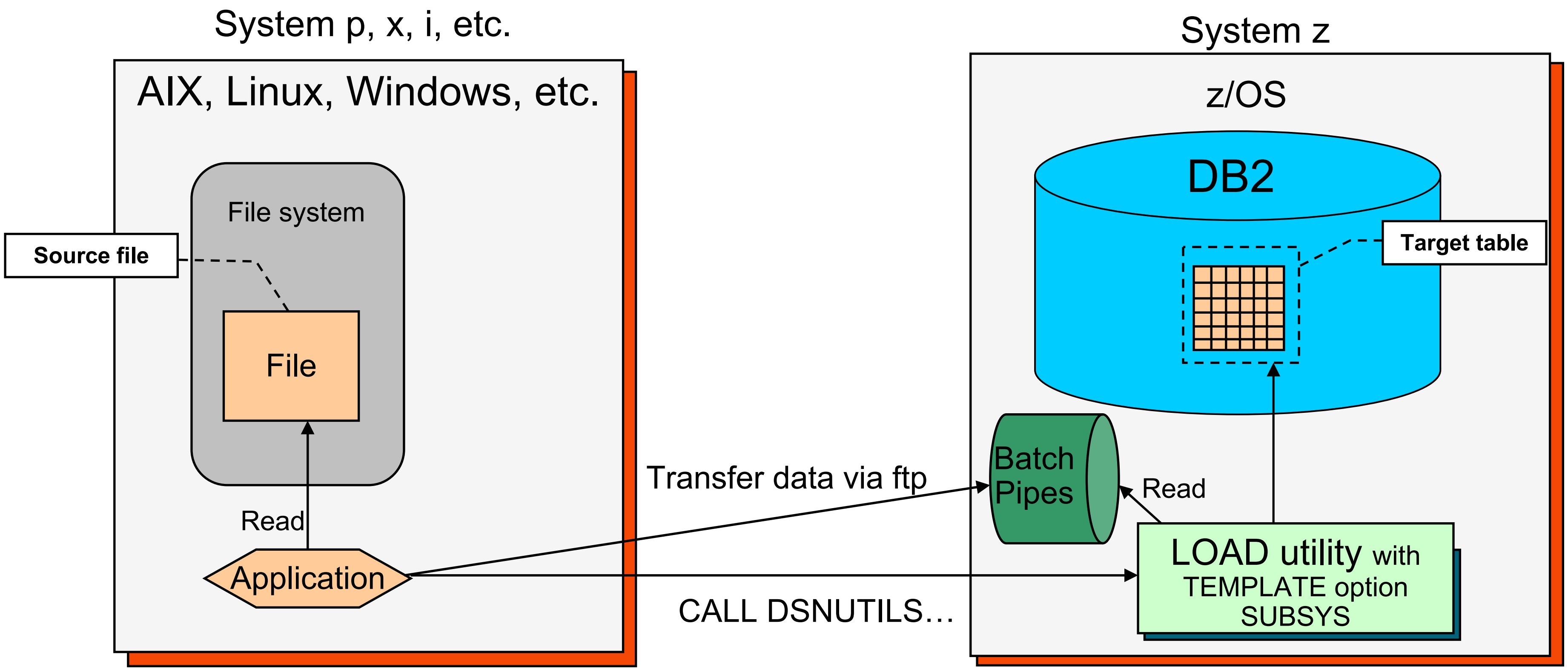
- After

```
DSNUGUTC -  REORG TABLESPACE DB1.TS1 COPYDDN(SYSCOPY) SHRLEVEL NONE STATISTICS
) ***********************************
) * DISPLAY DATABASE SUMMARY
  *   GLOBAL CLAIMERS
) ***********************************
…
NAME     TYPE PART  STATUS            CONNID   CORRID       CLAIMINFO
-------- ---- ----- ----------------- -------- ------------ --------
TS1      TS         RW,UTRO           BATCH    CAPP          (WR,C)
    -                  AGENT TOKEN 10
    -                  MEMBER NAME V91A
TS1      TS         RW,UTRO           BATCH    CAPP          (RR,C)
    -                  AGENT TOKEN 10
    -                  MEMBER NAME V91A
******* DISPLAY OF DATABASE DB1 ENDED **********************
) DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
  DSNUGBAC - RESOURCE UNAVAILABLE
             REASON 00C200EA
             TYPE 00000200
             NAME DB1.TS1
```

Culprits

LOAD via BATCHPIPE

- Can be used locally by specifying pipe on TEMPLATE
- …or remotely:



Look for article on this in the next IDUG Solutions Journal!

IBM's Unload Products

IBM’s Unload Products

- Two products from IBM
  - DB2 UNLOAD Utility (in the IBM DB2 Utilities Suite)
  - DB2 High Performance Unload (HPU) Utility
- HPU was delivered before the UNLOAD utility – had this not been the case, we would never have used the words “High Performance”
- In elapsed time, both are comparable (sometimes UNLOAD is faster, sometimes HPU is faster)
- In CPU time, HPU consumes approximately half the CPU in many situations (but not always)
- It’s OK (and we have client examples) to have both and use both
  - UNLOAD is geared towards user of DB2 Utilities (Utilities interface)
  - HPU is geared towards application developers (SQL interface)
- Don’t expect significant new functions in the UNLOAD utility beyond supporting DB2 functionality

Best Practices

TAKE BACK CONTROL

COPY/RECOVER/QUIESCE Best Practices

- COPY
  - PARALLEL keyword provides parallelism for lists of objects (including partitions)
  - CHECKPAGE YES
  - Incremental copy rule-of-thumb: Consider using incremental image copy if
    - <5% of pages are randomly updated (typically means less than 1% of rows updated)
    - <80% of pages are sequentially updated such that updated pages are together and separated from non updated pages
  - Copy indexes on your most critical tables to speed up recovery
- MERGECOPY – Consider using it
- RECOVER
  - PARALLEL keyword provides parallelism for lists of objects (including partitions)
  - Compressed pagesets result in faster restore phase
  - Enable Fast Log Apply (which can use dual-copy logs) and PAV
  - <=10 jobs/member with LOGAPSTG=100MB, 20-30 objects per RECOVER
- QUIESCE
  - WRITE NO is less disruptive (no quiescing of COPY=NO indexes)
  - Use TABLESPACESET
- Large BUFNO – the default is optimal in our lab
  - Anecdotal evidence of improved performance with a large BUFNO (e.g., BUFNO=100) but we have not seen this in our benchmarks – we suspect that this helped in cases where I/O configuration was not well tuned

LOAD Best Practices

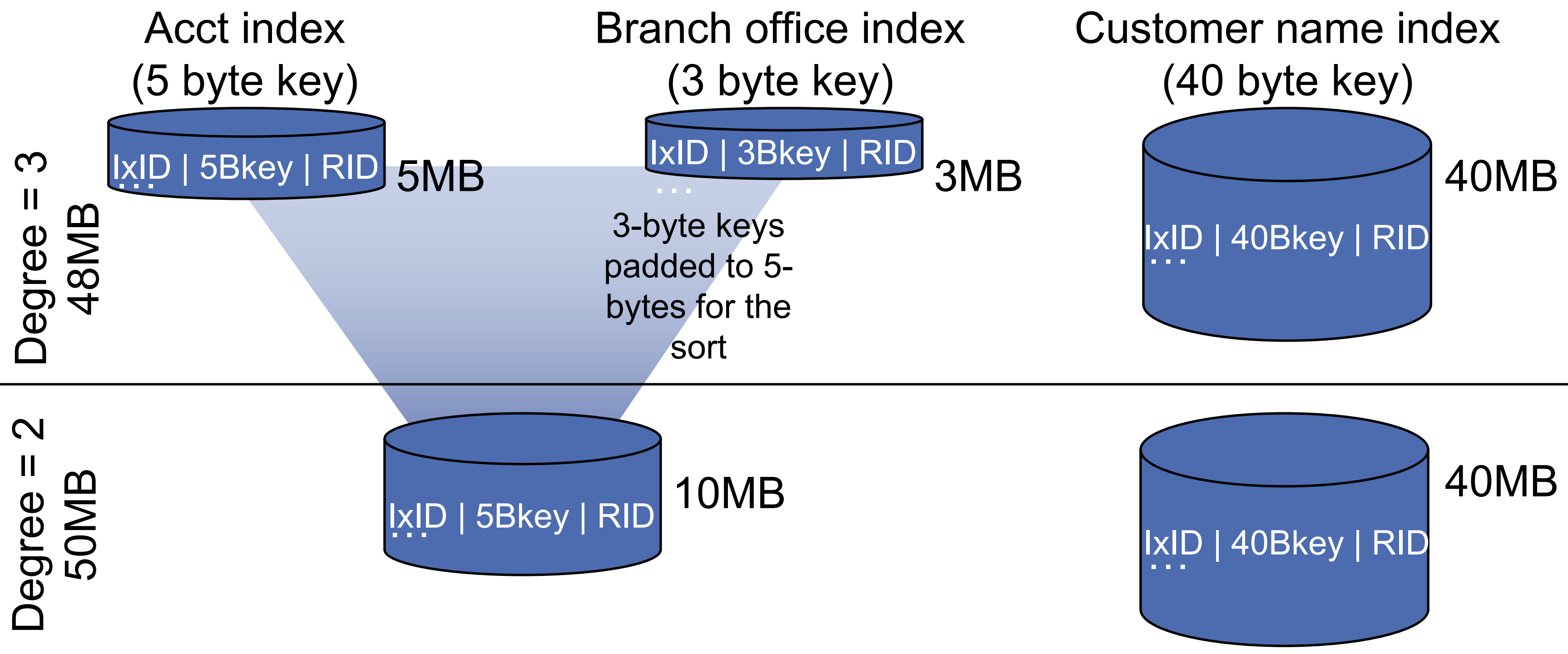
- LOAD
  - LOG NO reduces log volume; if REPLACE, then take inline copy
  - KEEPDICTIONARY (track dictionary effectiveness with history statistics PAGESAVE)
  - Load Partition Parallelism (V7)
    - Not individual LOAD part level jobs
    - Enable Parallel Access Volume (PAV)
  - Inline COPY & Inline STATISTICS
  - Index parallelism (SORTKEYS)
    - On LOAD, provide value for SORTKEYS <u>only</u> when input is tape/PDS mbr
    - Remove SORTWKxx and use SORTDEVT/SORTNUM
  - Use REUSE to logically reset and reuse DB2-managed data sets without deleting and redefining them (affects) elapsed time
  - When using DISCARD, try to avoid having the input on tape
    - Input is re-read to discard the errant records
  - Avoid data conversion, use internal representation if possible
  - Sort data in clustering order (unless data is randomly accessed via SQL)
  - LOAD RESUME SHRLEVEL CHANGE instead of batch inserts

REORG Best Practices

- REORG
  - LOG NO reduces log volume; requires an image copy (inline is a good choice)
  - KEEPDICTIONARY (track dictionary effectiveness with history statistics PAGESAVE)
  - On V7, SORTDATA to use table space scan and then sort
  - NOSYSREC to avoid I/O (forced for SHRLEVEL CHANGE)
    - Take full image copy before REORG SHRLEVEL NONE
  - Inline COPY & Inline STATISTICS
  - Index parallelism (SORTKEYS)
    - Remove SORTWKxx and use SORTDEVT/SORTNUM
  - Use REUSE to logically reset and reuse DB2-managed data sets without deleting and redefining them (affects) elapsed time

Parallel Index Build (triggered by SORTKEYS on LOAD, REORG, REBUILD)

- The higher the degree of parallelism, the less disk space is required to perform the sort of the index keys
- Example with three indexes on a table:



Degree = 1 would require 120MB of disk space due to needing to pad all index keys to the longest key to sort

REORG SHRLEVEL CHANGE Best Practices

- REORG SHRLEVEL CHANGE (sometimes called online REORG)
  - TIMEOUT TERM frees up the objects if timeouts occur in getting drains
  - DRAIN ALL (better chance of successful drain)
  - MAXRO = IRLMRWT minus 5-10 seconds (to prevent timeouts)
  - DRAIN_WAIT = IRLMRWT minus 5-10 seconds (to prevent timeouts)
  - RETRY = utility lock timeout multiplier… i.e, 6
  - RETRY_WAIT = DRAIN_WAIT*RETRY

  - Enable detection of long running readers (zparm) and activate IFCID 0313 (it's included in STATS CLASS(3))
    - This will report readers that may block command and utilities from draining
    - It includes "well-behaved" WITH HOLD cursors which a drain cannot break-in on

REBUILD/CHECK/RUNSTATS Best Practices

- REBUILD
  - Index parallelism (SORTKEYS)
  - remove SORTWKxx and use SORTDEVT/SORTNUM
  - Inline STATISTICS
  - CREATE INDEX DEFER followed by REBUILD INDEX
    - As of V8, dynamic SQL will not select the index until it is built
- CHECK DATA
  - If large volumes of delete data (e.g. after REORG DISCARD)
    - LOG NO to avoid log archive and log latch contention
    - Image COPY will be required
- CHECK INDEX
  - SHRLEVEL CHANGE
    - Uses dataset-level FlashCopy2 if available
    - Else, traditional media copy – still smaller r/o outage than SHR REF
- RUNSTATS
  - SHRLEVEL CHANGE for availability
  - Collect only column stats on columns used in SQL predicates
    - Use the Statistics Advisor to detect which stats to collect
    - SAMPLE reduces CPU time when gathering column stats
  - KEYCARD provides valuable info for little processing cost (see next slide)

KEYCARD

- Collects all of the distinct values in all of the 1 to *n* key column combinations for the specified indexes. *n* is the number of columns in the index. For example, suppose that you have an index defined on three columns: A, B, and C. If you specify KEYCARD, RUNSTATS collects cardinality statistics for column A, column set A and B, and column set A, B, and C.
- So these are cardinality statisics across column sets... if we had a 3-column index that had these values:

Col1	Col2	Col3
A	B	C
A	B	D
A	B	E
A	B	E
A	C	A
A	C	A
A	D	A
B	B	B

then these stats would be collected:

  - Col1 cardinality = 2
  - Col1 and Col2 cardinality = 4
  - Col 1, Col2, and Col3 cardinality = 6

Sorting w/DFSORT Best Practices

- Remove SORTWKxx and use SORTDEVT/SORTNUM
  - This will use dynamic allocation
  - To direct datasets to storage group, use ACS (see DFSMSrmm SMS ACS Support reference on *References* slide)
- DFSORT installation options (see APAR II14047)
  - Memory usage
    - Leave the default for SIZE set to MAX
    - Don't bother with changing TMAXLIM (initial storage for each sort)
    - If you have to turn a knob…. DSA (Dynamic Size Adjustment)
      - R14 default is 32M; V1R5 default is 64M
      - You could set this to 128M, but then look to see if DFSORT ever uses this much (needs REGION > 128M!)
      - Follow DFSORT tuning recommendation to use hiperspaces, data spaces, etc. (if not on 64-bit LPAR)
  - Default number of sort datasets is 4 – change this to handle 80-90% of jobs
- >64K track datasets for DFSORT supported in z/OS 1.7

Number of DFSORT sort work datasets for DB2 Utilities (SORTNUM on LOAD/REORG/REBUILD)

- Background on max size for a sort work dataset
  - Prior to z/OS V1R7 -> 64K tracks (on 3390 devices, this is ~3.4GB)
  - z/OS V1R7 -> ~million tracks (on 3390 devices, this is ~=52.7GB)
  - Large volume sizes can reduce the number of sort work datasets
- Dynamic allocation of sort work datasets
  - SORTNUM>=2 for each sort, and it applies to each sort in the utility
  - If 3 indexes; SORTKEYS is specified; there are no constraints limiting parallelism, and SORTNUM is specified as 8, then a total of 24 sort work datasets will be allocated for a REBUILD INDEX job, for example
  - For REORG tsp, SORTNUM applies to data sort(s) and index sort(s)
  - If no SORTNUM, default value for SORT is used (DFSORT default=4)
  - There is a danger in specifying too high a value for SORTNUM
    - Each sort work dataset consumes both above the line and below the line virtual storage, and if you specify too high a value, the utility may decrease the degree of parallelism due to virtual storage constraints, possibly decreasing the degree down to one, meaning no parallelism.

Number of DFSORT sort work datasets for DB2 Utilities (SORTNUM on LOAD/REORG/REBUILD)

- Example
  - Given 40GB of data to sort (so we'll need 48GB of disk space, using the 1.2 X factor)
  - If the sort device is a 3390 Model 9 (whose volume capacity is 8.4GB) and we're running z/OS V1R5 (where sort work datasets cannot exceed 64K tracks, ~=3.4GB), then at least 15 sort work data sets would be needed to satisfy the sort (the limiting factor being DFSORT's limit of 64K tracks prior to z/OS V1R7)
  - If instead, the sort device is a 3390 Model 27 (whose volume capacity is around 27GB) and we're running z/OS V1R7, then at least 2 sort work data sets would be needed to satisfy the sort (the limiting value being the size of the 3390 Mod 27 volume).
  - In both of these cases there must be enough space for each of the datasets to extend to their maximum size.

A scenario of best practices

TAKE BACK CONTROL

Reorg scenario

- Reorg tablespaces: ADHTSTDB.ADHTSTTS and ADHTSTDB.ADHTSTTT
- Invocation:

```
REORG TABLESPACE ADHTSTDB.ADHTSTTS
REORG TABLESPACE ADHTSTDB.ADHTSTTT
```

- Best practices followed above? No. Here, we apply the best practices…

```
LISTDEF APPL1 INCLUDE ADHTSTDB.ADHTSTT*
TEMPLATE BACKUP1 DSN &DB..&TS..D&DA..T&TI. UNIT=SYSALLDA
REORG TABLESPACE LIST APPL1
LOG NO
KEEPDICTIONARY
SORTDATA          -- Default in V8
NOSYSREC          -- Take an image copy first!
COPYDDN(BACKUP1)
STATISTICS
   TABLE (T1) SAMPLE 25 COLUMN (C1, C2)  -- Only cols in SQL
   TABLE (T2) SAMPLE 25 COLUMN (C5, C12) -- Only cols in SQL
SORTKEYS
SORTDEVT          -- SORTNUM defaults to sort component's parms
```

Provide logic to routine maintenance

- Leverage the ability to invoke utilities programmatically via stored procedures
  - DSNUTILS for EBCDIC parameters
  - DSNUTILU for UNICODE parameters

Provide logic to routine maintenance

- Example (using REXX):

```
/* REXX */
…
ADDRESS DSNREXX "CONNECT DB2P"
IF SQLCODE ¬= 0 THEN CALL SQLCA

Uid='';Restart=''; Utstmt=,
'REORG TABLESPACE ADHTSTDB.ADHTSTTS',
  'LOG NO KEEPDICTIONARY',
  'SORTDATA SORTKEYS SORTDEVT'
  'STATISTICS',
    'TABLE (T1) SAMPLE 25 COLUMN (C1, C2)',
    'TABLE (T2) SAMPLE 25 COLUMN (C5, C12)'
Utility='REORG TABLESPACE'
CopyDSN1='DSN.FIC.ADHTSTTS.VERSION(+1)'
CopyDEVT1='SYSDA'
CopySpace1=1
```

```
ADDRESS DSNREXX "EXECSQL" ,
"CALL DSNUTILS(:UID, :RESTART,                 " ,
"              :UTSTMT,                        " ,
"              :RETCODE,                       " ,
"              :UTILITY,                       " ,
"              :RECDSN  ,:RECDEVT  ,:RECSPACE  ," ,
"              :DISCDSN ,:DISCDEVT ,:DISCSPACE ," ,
"              :PNCHDSN ,:PNCHDEVT ,:PNCHSPACE ," ,
"              :COPYDSN1,:COPYDEVT1,:COPYSPACE1," ,
"              :COPYDSN2,:COPYDEVT2,:COPYSPACE2," ,
"              :RCPYDSN1,:RCPYDEVT1,:RCPYSPACE1," ,
"              :RCPYDSN2,:RCPYDEVT2,:RCPYSPACE2," ,
"              :WORKDSN1,:WORKDEVT1,:WORKSPACE1," ,
"              :WORKDSN2,:WORKDEVT2,:WORKSPACE2," ,
"              :MAPDSN  ,:MAPDEVT  ,:MAPSPACE  ," ,
"              :ERRDSN  ,:ERRDEVT  ,:ERRSPACE  ," ,
"              :FILTRDSN,:FILTDEVT ,:FILTRSPACE)"
IF SQLCODE < 0 THEN CALL SQLCA
…
```

Provide logic to routine maintenance

- Rich logic can be provided to
  - Take an image copy before running REORG with NOSYSREC
  - Examine statistics (from RUNSTATS or the Real-time statistics) to determine when to run a utility
  - Examine a control table to determine windows when maintenance can or cannot be run
  - …

- You have full control without needing individual threshold keywords on each utility

- But, maybe you don't want to write <u>or maintain</u> this type of logic yourself… that where products like the DB2 Automation Tool for z/OS come into play

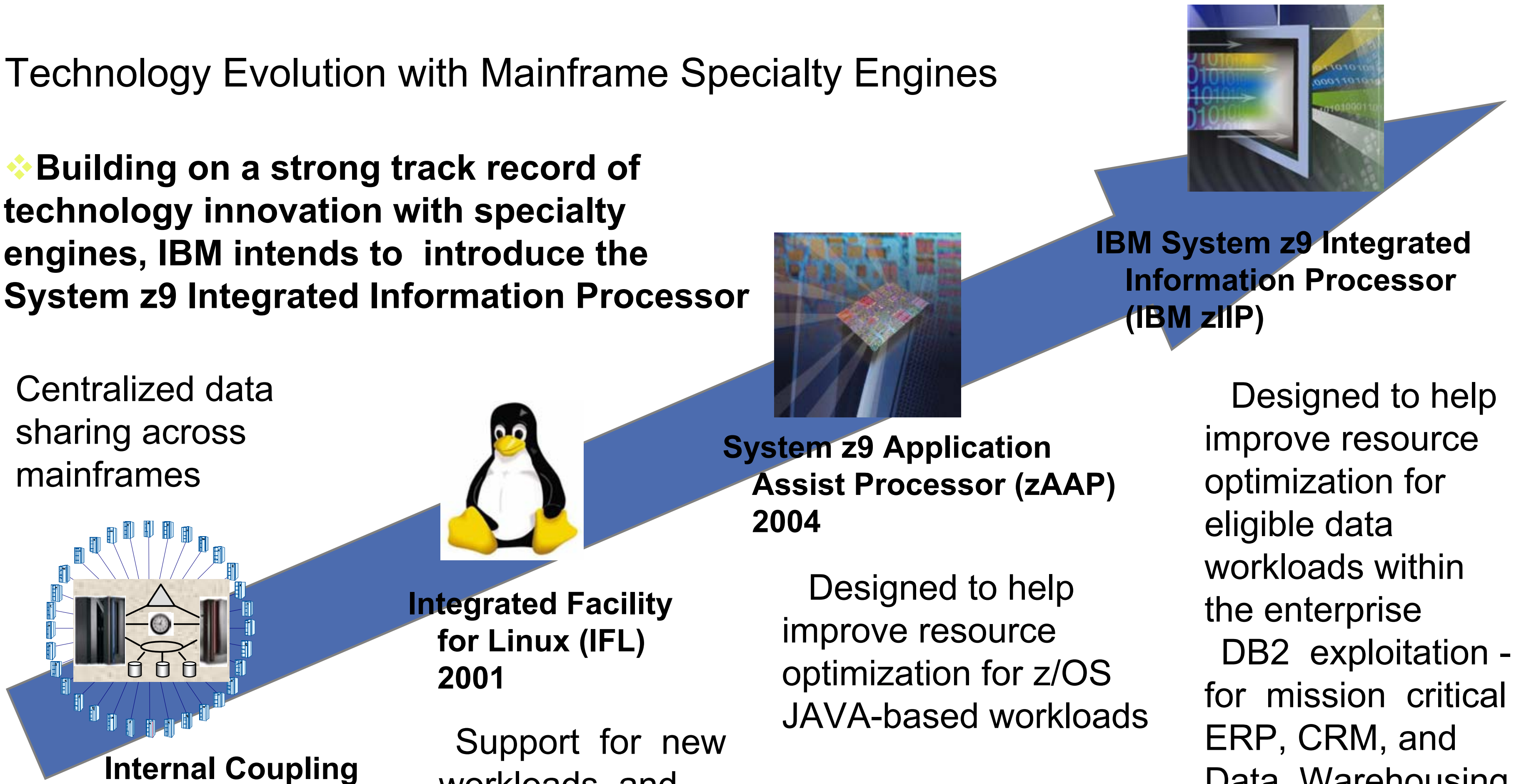
zIIP - (IBM System z9 Integrated Information Processor)

TAKE BACK CONTROL

zIIP (IBM System z9 Integrated Information Processor) -- 2006

Technology Evolution with Mainframe Specialty Engines

❖**Building on a strong track record of technology innovation with specialty engines, IBM intends to introduce the System z9 Integrated Information Processor**



IBM System z9 Integrated Information Processor (IBM zIIP)
System z9 Application Assist Processor (zAAP) 2004
Integrated Facility for Linux (IFL) 2001

Centralized data sharing across mainframes

Internal Coupling Facility (ICF) 1997

Support for new workloads and open standards

Designed to help improve resource optimization for z/OS JAVA-based workloads

Designed to help improve resource optimization for eligible data workloads within the enterprise

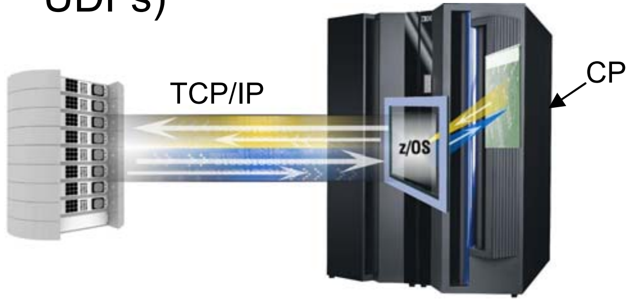
DB2 exploitation - for mission critical ERP, CRM, and Data Warehousing workloads

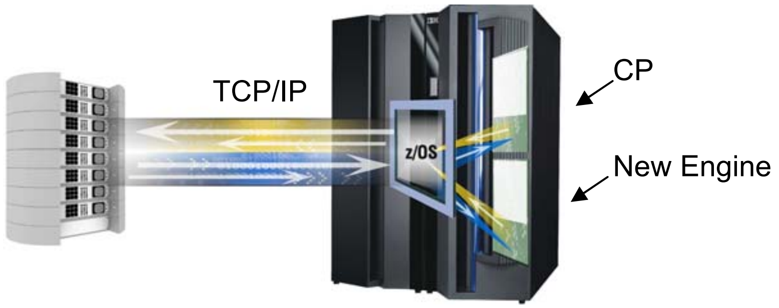
IBM zIIP Technical Details

- New specialty engine for the System z9 mainframe
  - Priced similar to other speciality engines
  - Does not affect IBM software license charges (same as all speciality engines)
  - Number of zIIPs per z9-109 not to exceed number of standard processors
  - z/OS manages and directs work between general purpose processors and the zIIPs
    - The zIIP is designed so that a program can work with z/OS to have all or a portion of it's enclave Service Request Block (SRB) work directed to the zIIP.
    - The enclave SRB interface is available upon request to non-IBM vendors as well.
- DB2 V8 for z/OS will be first exploiter of the zIIP with:
  - System z9 109
  - z/OS 1.6 or later
  - DB2 V8 for z/OS (with PTFs, listed in InfoAPAR II14219) or V9
- No changes required for DB2 V8 for z/OS applications
- We anticipate minimal overhead associated with enclave SRB redirect to zIIP
  - Workload is dispatched to enclave SRB/zIIP the same way it is dispatched on a general purpose processor

Types of DB2 for z/OS workloads that may benefit from zIIP

1. **ERP, CRM, Business Intelligence or other enterprise applications**
   - Via DRDA over a TCP/IP connection (enclave SRBs, not stored procedures or UDFs)





2. **Data warehousing applications***
   - **Requests that utilize parallel queries, including star schema**
3. **DB2 V8 for z/OS utilities***
   - **DB2 utility functions used to maintain index maintenance structures**

New DB2 accounting trace fields: IIP and IIPCP Time

```
 LOCATION: DSND81B          OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V3)
      GROUP: N/P                       ACCOUNTING REPORT - LONG
     MEMBER: N/P
  SUBSYSTEM: D81B                           ORDER: PLANNAME
DB2 VERSION: V8                             SCOPE: MEMBER

PLANNAME: DSNUTIL

AVERAGE       APPL(CL.1)  DB2 (CL.2)  IFI (CL.5)
------------  ----------  ----------  ----------
ELAPSED TIME  1:44.25600   53.465083         N/P
 NONNESTED    1:44.25600   53.465083         N/A
 STORED PROC    0.000000    0.000000         N/A
 UDF            0.000000    0.000000         N/A
 TRIGGER        0.000000    0.000000         N/A

CP CPU TIME    48.979165   18.069021         N/P
 AGENT         12.694310   10.249968         N/A
  NONNESTED    12.694310   10.249968         N/P
  STORED PRC    0.000000    0.000000         N/A
  UDF           0.000000    0.000000         N/A
  TRIGGER       0.000000    0.000000         N/A
 PAR.TASKS     36.284855    7.819053         N/A

 IIPCP CPU      3.432845         N/A         N/A

IIP CPU TIME   23.411692   23.411692         N/A
```

Work offloaded to zIIPs

zIIP-eligible on CP

Details on DB2 for z/OS utilities zIIP Offload

- Only the portions of DB2 utility processing related to index maintenance are redirected.
  - Only the BUILD portion of LOAD, REORG, and REBUILD
- Amount of workload eligible for zIIP will depend on:
  - How many indexes are defined on the table
  - How many partitions are in the table
  - If data compression is being used
  - Possibly other factors
- Lower amount eligible is expected with:
  - Tables with fewer indexes
  - Fewer partitions
  - Compression used
- Higher amount eligible is expected with:
  - LOAD and REORG with many indexes or many partitions
- CPU overhead incurred during execution unit switch from TCB to enclave SRB during Index Rebuild phase
  - Typically less than 10%
  - Eligible for offload

Possible reduction in Class 1 CPU:

•5 to 20% for Rebuild Index

•10 to 20% for Load/Reorg of a partition with one index only, or Load/ Reorg of entire tablespace

•40% for Rebuild Index of logical partition of non partitioning index

•40 to 50% for Reorg Index

•30 to 60% for Load or Reorg of a partition with more than one index

DB2 9 Utilities

DB2 9 Utilities

- Support for all new functions in DB2 Version 9 for z/OS product (universal table spaces, XML, not logged, etc.)
- More online utilities
  - Rebuild Index SHRLEVEL CHANGE
    - Great for building new non-unique indexes or when index is in RBPD
    - The only reason to rebuild the index is if it is in rebuild pending
    - Use REORG INDEX SHRLEVEL CHANGE to move indexes to different vols, not REBUILD INDEX SHRLEVEL CHANGE if availability is important
    - Table space must be in LOGGED state

DB2 9 Utilities

- More online utilities (cont…)
  - Reorg enhancements
    - Reorg LOB now supports SHRLEVEL REFERENCE (space reclamation)
    - Partition-level capabilities (not available with REBALANCE)
      - Partition parallelism (UNLOAD/RELOAD) in a single utility statement
        - » Parallel log apply for SHRLEVEL REFERENCE and CHANGE
        - » Multiple concurrent jobs no longer needed/supported – DSNU180I
      - Elimination of the BUILD2 phase outage
        - » NPSIs are also shadowed (higher disk requirement compared to when just logical part was reorganized), thus resulting in the NPIs also being reorganized -> therefore, don't follow with REORG of NPSIs.
        - » Higher availability results in increased resource consumption
        - » REORG SHRLEVEL REFERENCE now has a log phase since entire NPSI is shadowed and reorg'd

DB2 9 Utilities

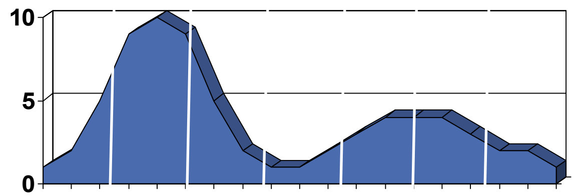
- More online utilities (cont…)
  - Check data, LOB and repair locate … SHRLEVEL CHANGE
  - Check index SHRLEVEL REFERENCE supports parallel for > 1 index
  - Load replace (SHRLEVEL CHANGE) via CLONE TABLE function

DB2 9 Utilities

- All utility messages have timestamp and julian date (day)

  DSNU000I   158 22:52:30.07 DSNUGUTC…

- Additional RUNSTATS statistics (Histograms)
  - An extension to frequency statistics for ranges of data values

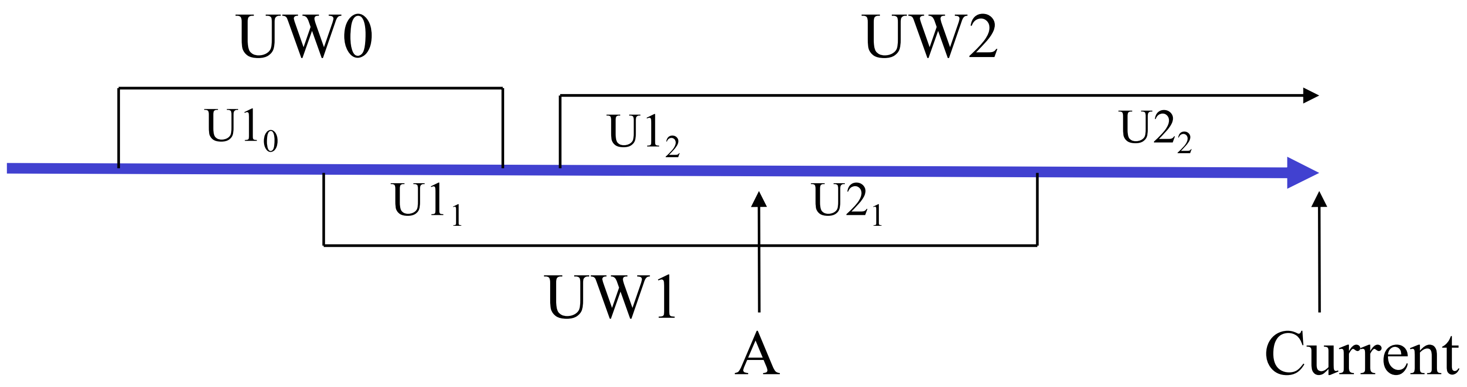


DB2 9 Utilities

- Always perform CHECKPAGE on the COPY utility
  - Prior to V9, CHECKPAGE was optional, with about ~5% CPU overhead, and if a broken page was encountered (DSNU441I for space maps or DSNU518I for others, both RC8), then copy-pending was set
  - Now, COPY always performs these checks (with reduced overall CPU!) and no longer sets copy-pending, so…. Check those RCs!
  - A new SYSCOPY record type is written if a broken page is detected to force a full image next since dirty bits may have already been flipped off in the space map pages
- The COPY utility includes SCOPE PENDING support to improve usability
- RECOVER RESTOREBEFORE x'rba/lrsn‘ directs RECOVER to use a recovery base prior to the rba/lrsn

DB2 9 Utilities

- The ability to recover to any point in time with consistency
  - Uncommitted changes are backed out (UW1 and UW2 are backed out)
  - Significantly reduces (eliminates?) the need to run QUIESCE which can be disruptive to applications
  - Does not apply to RECOVER TOCOPY, TOLASTCOPY and TOLASTFULLCOPY using SHRLEVEL CHANGE copy (consistency is not ensured – use RBA/LRSN after COPY point)
  - Include all relevant objects in same RECOVER ensure data consistency from the application point of view
  - Requires NFM



UWn - Unit of Work number n

Un_m _ Update number n in Unit of Work m

DB2 9 Utilities

- The ability to recover to any point in time with consistency (contd)
  - Two new phases in RECOVER: LOGCSR and LOGUNDO (will each be executed by member)
  - RECOVER can be restarted in any LOG phase, LOGCSR will restart at beginning, others at last commit point
  - Fast Log Apply not used for LOGUNDO phase
  - Progress of LOGUNDO phase can be monitored by regular status message DSNU1555I showing current and target log RBA (e.g. to detect backout pf long running URs)
  - This function can greatly reduce your need to run QUIESCE
    - If you want to capture a log point non-disruptively
      - -DISPLAY LOG, or
      - QUIESCE against SYSEBCDC (or some other dummy tablespace), or
      - SELECT CURRENT_TIMESTAMP from SYSDUMMY1 and then convert it to an LRSN

DB2 9 Utilities

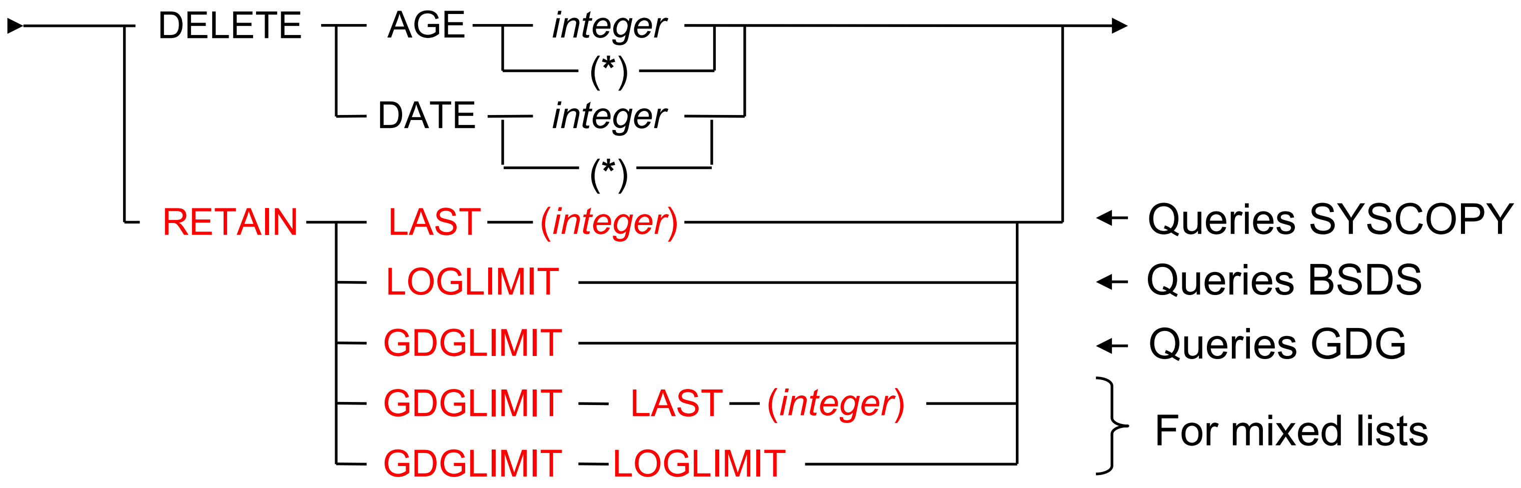
- Template switching (e.g., copy to tape if large; to disk if small)

```
TEMPLATE TEM2 DSN &DB..&TS..D&DA..T&TI. UNIT=TAPE
TEMPLATE TEM1 DSN &DB..&TS..D&DA..T&TI. UNIT=SYSALLDA LIMIT(20 CYL, TEM2)
COPY TABLESPACE SMALL.TS COPYDDN(TEM1)
COPY TABLESPACE LARGE.TS COPYDDN(TEM1)
```

  – Only supported for image and inline copies (COPYDDN, RECOVERYDDN); LIMIT is ignored if the TEMPLATE on which it is specified is used for other purposes (e.g.WORKDDN, PUNCHDDN)
  – Uses high formatted page of table space for full copy or 10% for incremental copy; short read claim during UTILINIT to determine size
  – No chaining of templates (only 1 switch); PREVIEW ignores switching
- Support large block interface (allows blocksize > 32,760 bytes) for tapes
  – Up to 40% reduction in elapsed time for COPY and restore of RECOVER
- Support large format datasets (allows > 65,535 tracks per DASD volume)
  – Handy for COPYing those large tablespaces with TEMPLATEs
  – Requires NFM for creating large format datsets

DB2 9 Utilities

- Modify Recovery simplification and safety



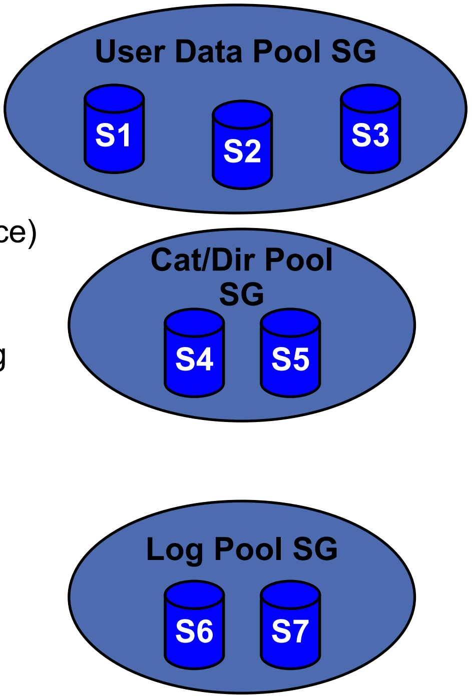
  – Syslgrnx records deleted even if no syscopy records deleted
  – Deletion works on dates not on timestamps, so more entries than requested might be kept

DB2 9 Utilities

- BACKUP/RESTORE SYSTEM
  - But first, a quick primer… DFSMS terms explained

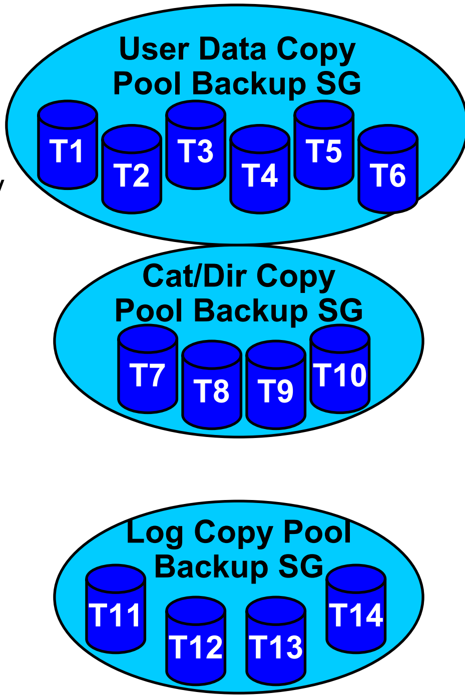
POOL Storage Groups

- Similar in concept to DB2 STOGROUPs
- A collection of (source) volumes
- Stores user data, catalog/directory, log data sets
- This example uses source volumes: S1 – S7



COPY POOL BACKUP Storage Groups

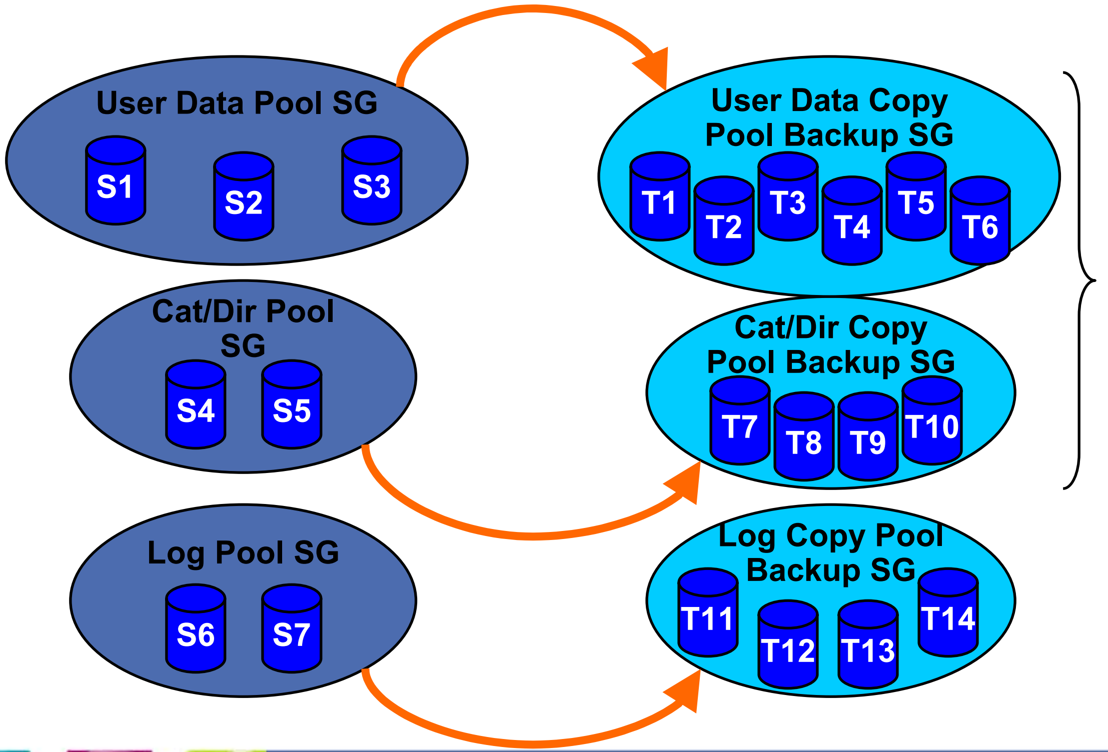
- A collection of volumes dedicated as FlashCopy targets for a source storage group
- Stores backups of user data, catalog/directory, log data sets
- Should contain a fixed multiple of volumes of the source storage group
- This example uses target volumes: T1 – T14



DB2 9 Utilities

- BACKUP/RESTORE SYSTEM
  - But first, a quick primer… Defining the FlashCopy target SG

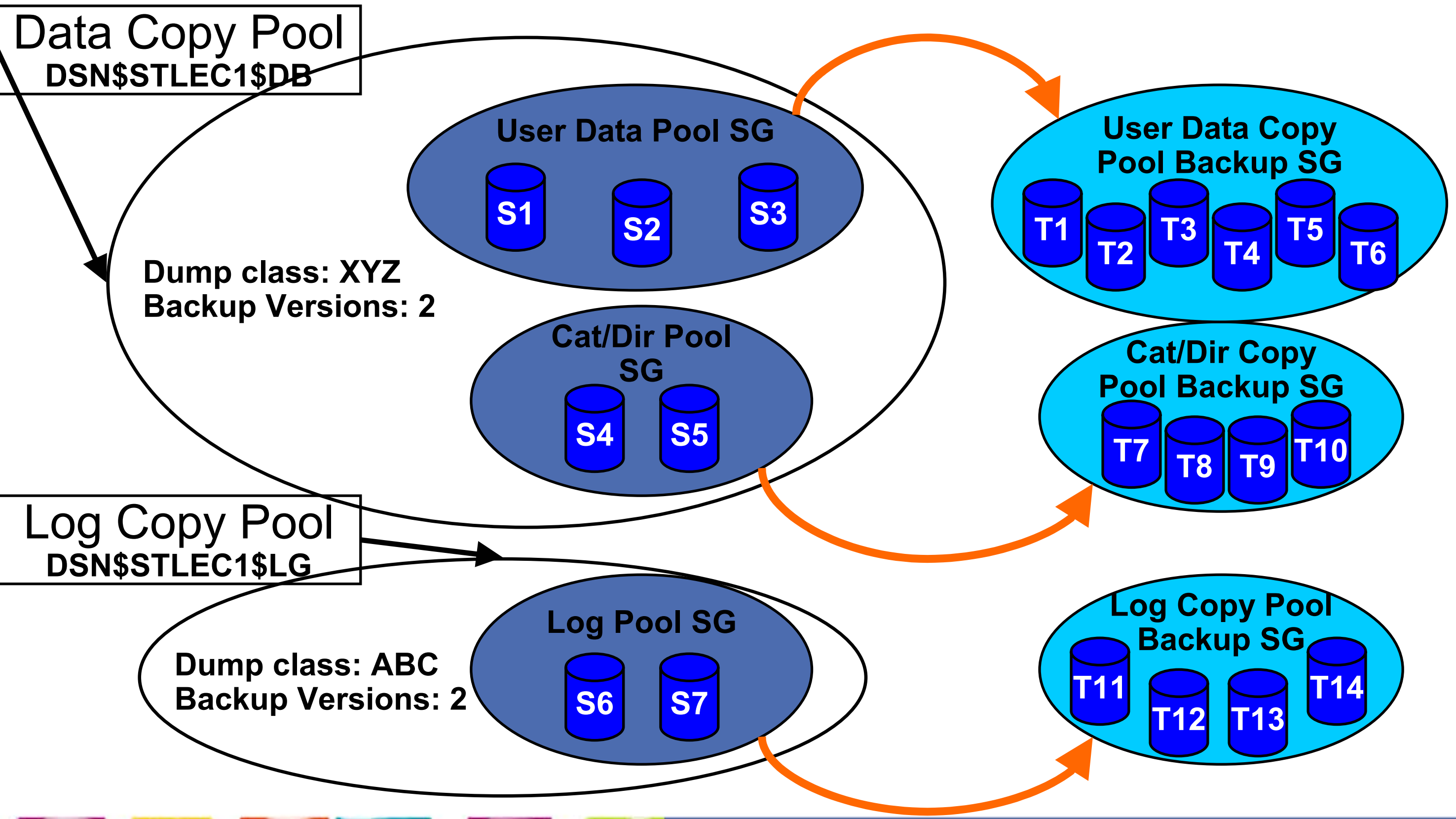
  The Copy Pool Backup SG is an attribute of the Pool SG



These volumes could be in one Copy Pool Backup SG

DB2 9 Utilities

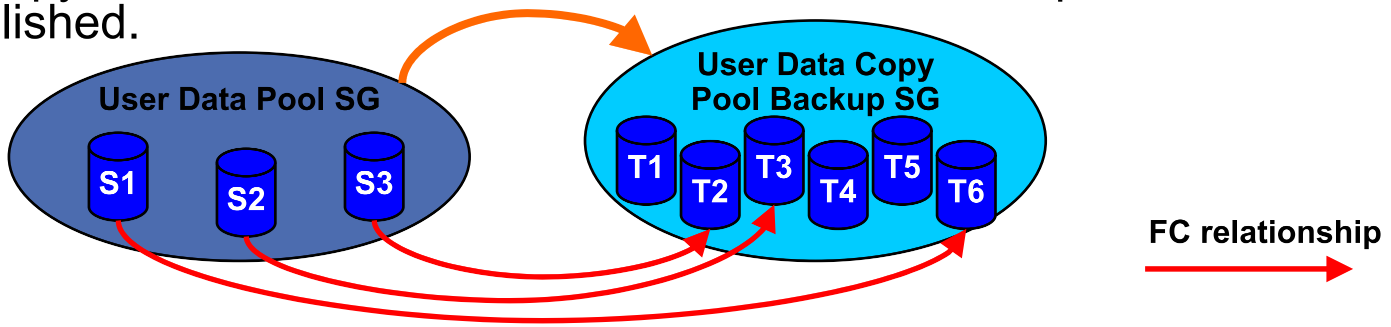
- BACKUP/RESTORE SYSTEM
  - But first, a quick primer… Copy Pools are a collection of source SGs



- A Copy Pool contains up to 256 (source) storage groups
- BACKUP SYSTEM/ RESTORE SYSTEM operate on Copy Pool Level

DB2 9 Utilities

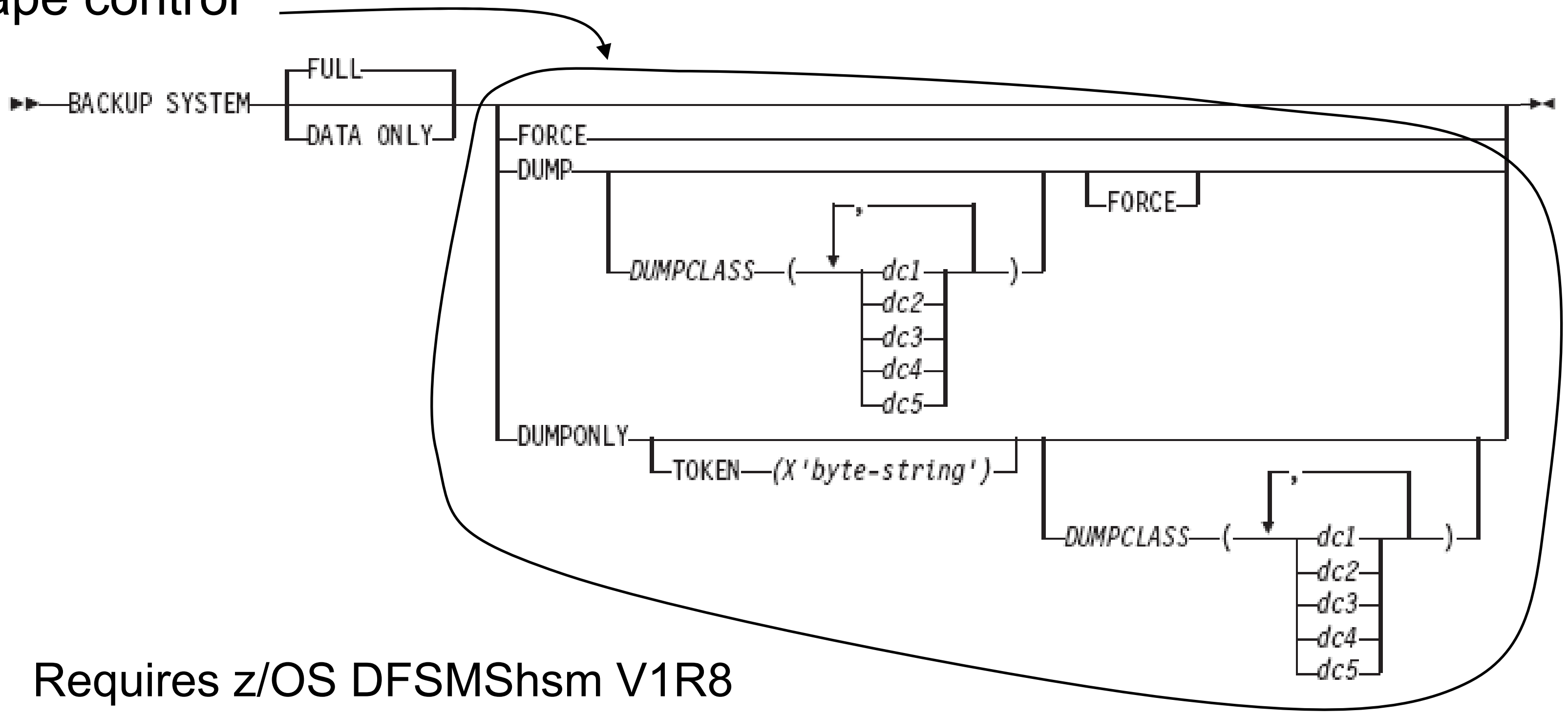
- BACKUP/RESTORE SYSTEM
  - But first, a quick primer… FlashCopy (via the BACKUP SYSTEM DATA ONLY utility statement)
    - A relationship is established between source volume and target volume
    - The copy is considered successful when the relationship is established.



    - A background copy is then started which will result in a target volume that looks like the source volume when the relationship was established.
      - This can result in a performance degradation because of the I/O
      - If a track on the source volume changes before the background copy is complete then that track will be copied to the target volume before the new contents of the track is written to the source volume.
    - The relationship goes away when the background copy completes.

DB2 9 Utilities

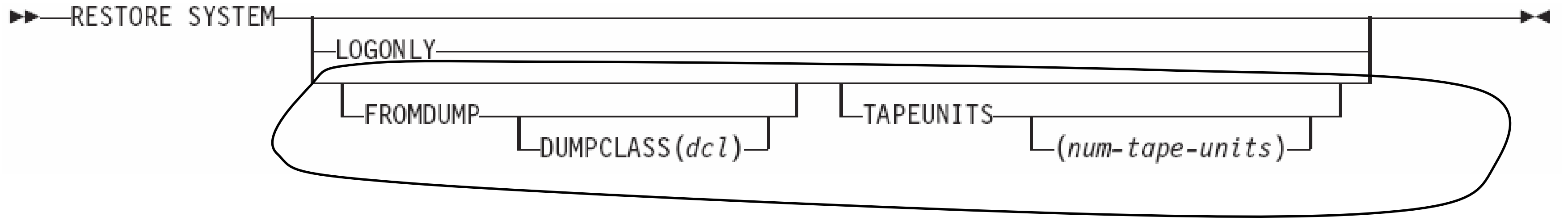
- BACKUP/RESTORE SYSTEM (what's new in DB2 9)
  - Tape control



- Requires z/OS DFSMShsm V1R8
- FORCE indicates that oldest copy can be overwritten even if dump to tape is in progress
- Use LIST COPYPOOL with DUMPVOLS option to verify dump status

DB2 9 Utilities

- BACKUP/RESTORE SYSTEM (what's new in DB2 9)



  - Restore from dumps will use parallelism; limited by number of distinct tape volumes that the dump resides on; capped by TAPEUNITS
  - ZPARMs to override tape options (DSNTIP6):

• FROMDUMP	←→	RESTORE/RECOVER FROM DUMP
• DUMPCLASS	←→	UTILS DUMP CLASS NAME
• TAPEUNITS	←→	RESTORE TAPEUNITS

DB2 9 Utilities

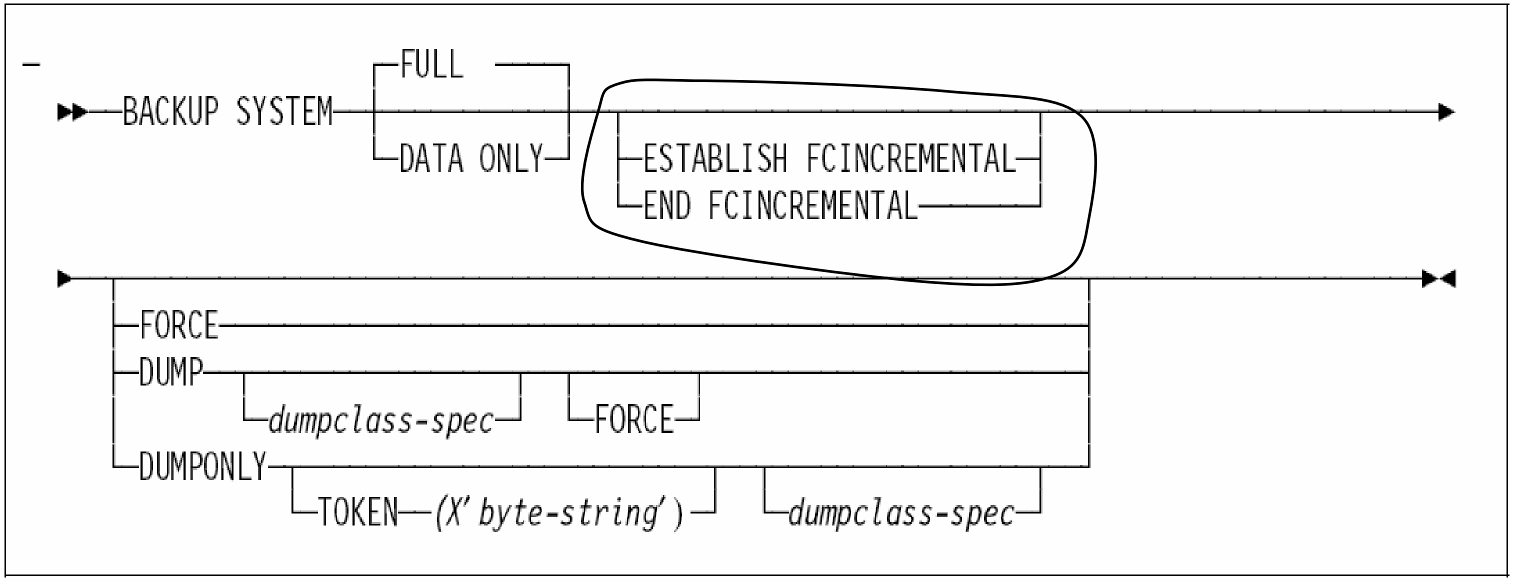
- BACKUP/RESTORE SYSTEM (what's new in DB2 9)
  - The ability to recover at the object level using system-level backups
    - RECOVER to point in time or to current
    - COPY YES indexes can be included in this recovery
    - Will restore from the previous system level backup or image copy
    - Some restrictions
      - If underlying datasets are deleted or moved to a different set of volumes, then object-level recovery is not possible
      - Therefore, take in-line image copies with REORG, and force object-level recoveries after moving DB2 pagesets
    - Requires z/OS V1R8
  - RTS COPY* columns also updated for BACKUP SYSTEM
  - Specify YES for SYSTEM-LEVEL BACKUPS on DSNTIP6
  - Support for Incremental FlashCopy (Post GA via APAR)

DB2 9 Utilities

- BACKUP/RESTORE SYSTEM (what's new in DB2 9)
  - Another primer, this time on Incremental FlashCopy:
    - A persistent relationship is established between two DASD devices
    - One is a source volume and the other is a target volume
    - DFSMShsm chooses the volumes
    - All tracks on the source volume are considered to be changed when the relationship is established so all tracks are copied.
    - Subsequent incremental copies will copy only the tracks that have changed on the source volume since the last copy was taken
    - A DASD volume can have only one incremental relationship
    - If a Copy Pool has more than 1 version then the remaining versions will be full backups

DB2 9 Utilities

- BACKUP/RESTORE SYSTEM (what's new in DB2 9)
  - Support for Incremental FlashCopy (Post GA via APAR)

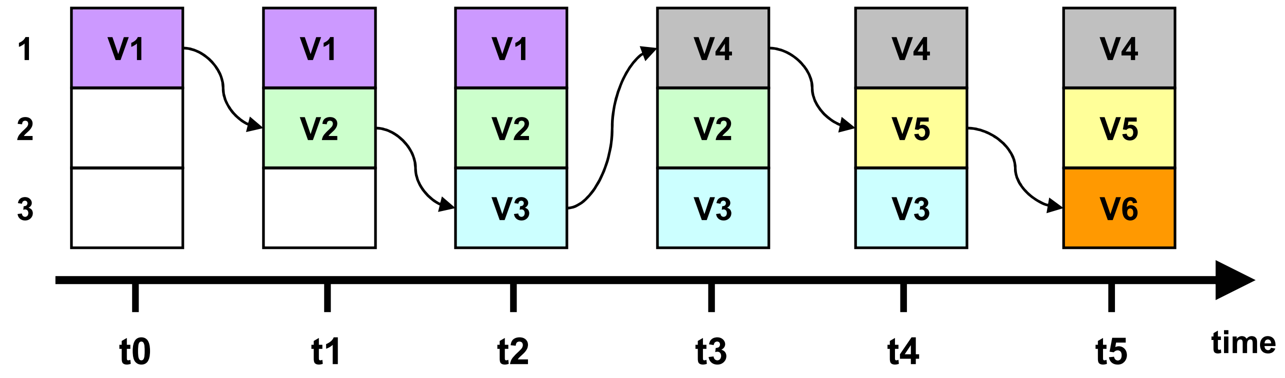


  - Incremental FlashCopy can also be activated using HSM commands; BACKUP SYSTEM will then use it

DB2 9 Utilities

- What are Fast Replication versions and generations?
  - A Copy Pool is defined with a number of *Versions* that can be maintained in parallel
  - *Version* numbers are increased with every FRBACKUP invocation (1..999)
  - If more versions are created than the Copy Pool can support then the oldest versions will be rolled off:

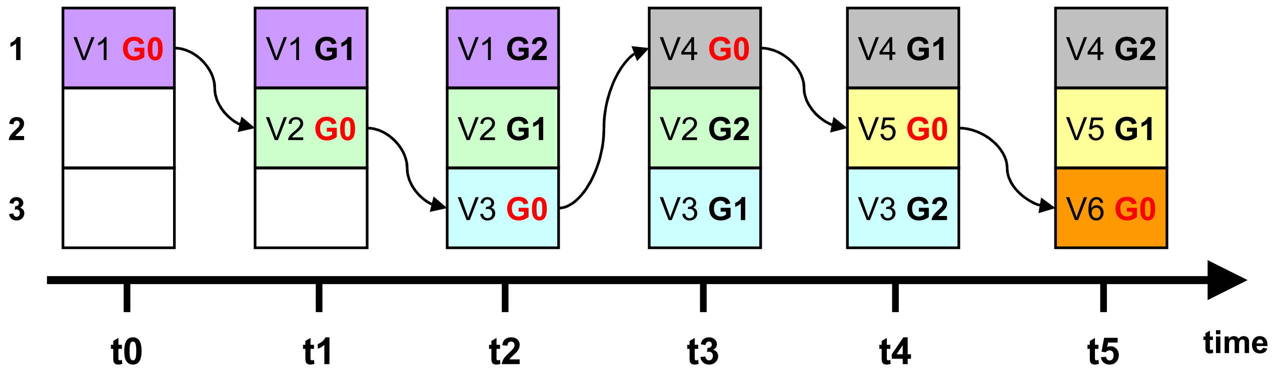
Copy Pool with 3 versions:



DB2 9 Utilities

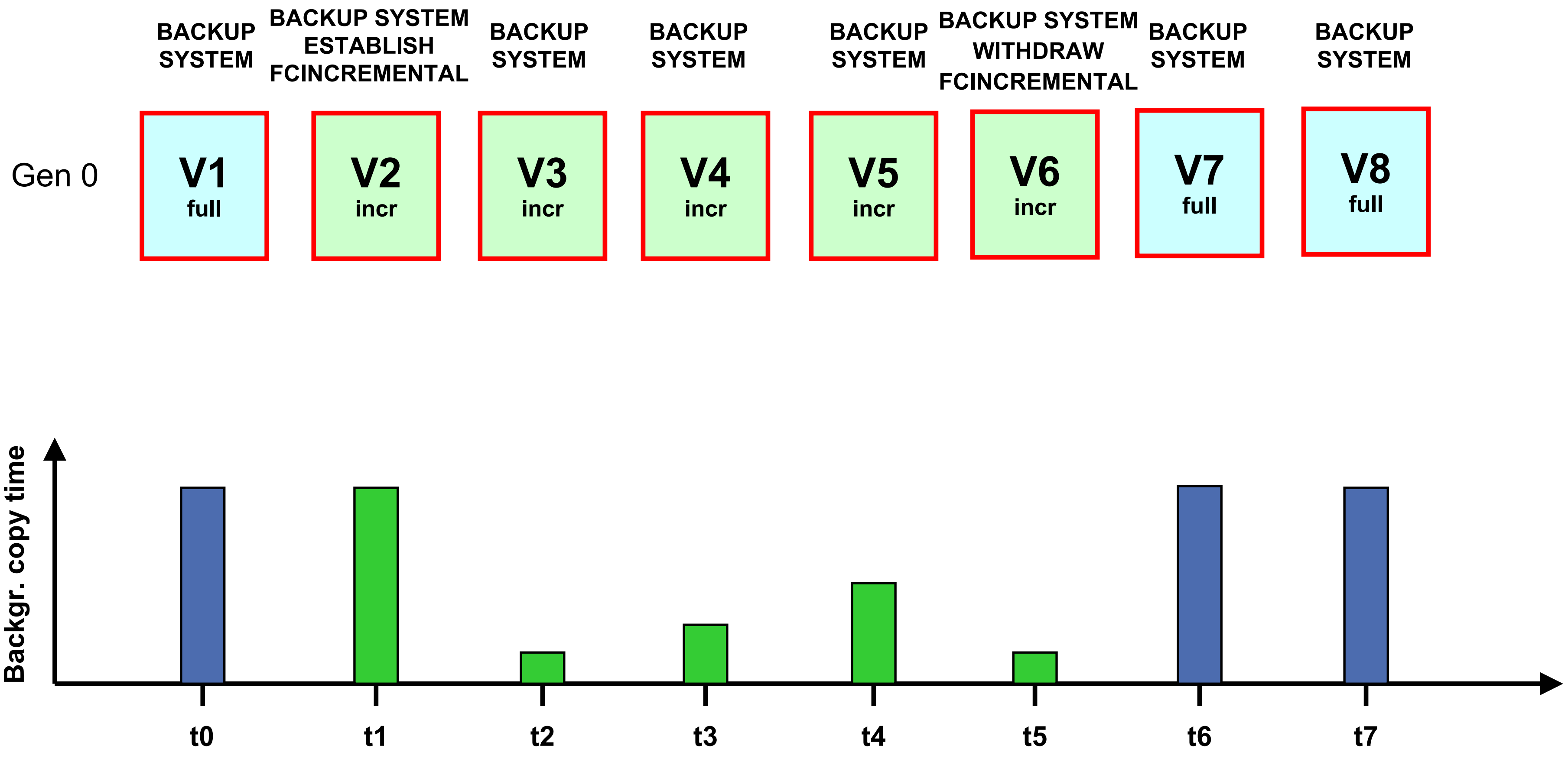
- What are Fast Replication versions and generations?
  - Generations are a relative way to address versions in the Copy Pool
  - Generation 0 always refers to the most current version, generation 1 refers to the second most current version, etc.
  - Same concept as for Generation Data Groups (GDG)

Copy Pool with 3 versions:



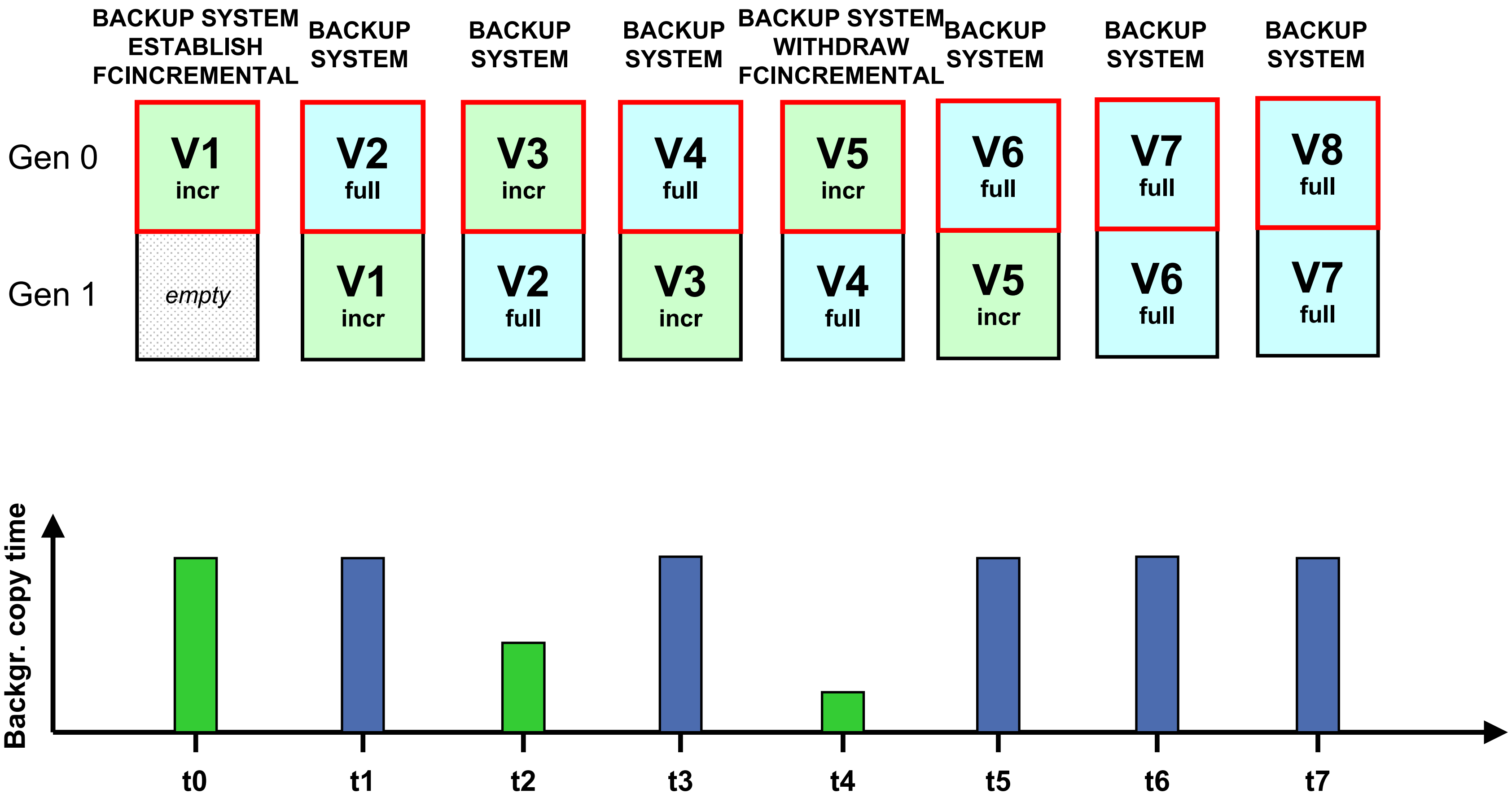
DB2 9 Utilities

- BACKUP/RESTORE SYSTEM (what's new in DB2 9)
  - Incremental Flashcopy with a single version:



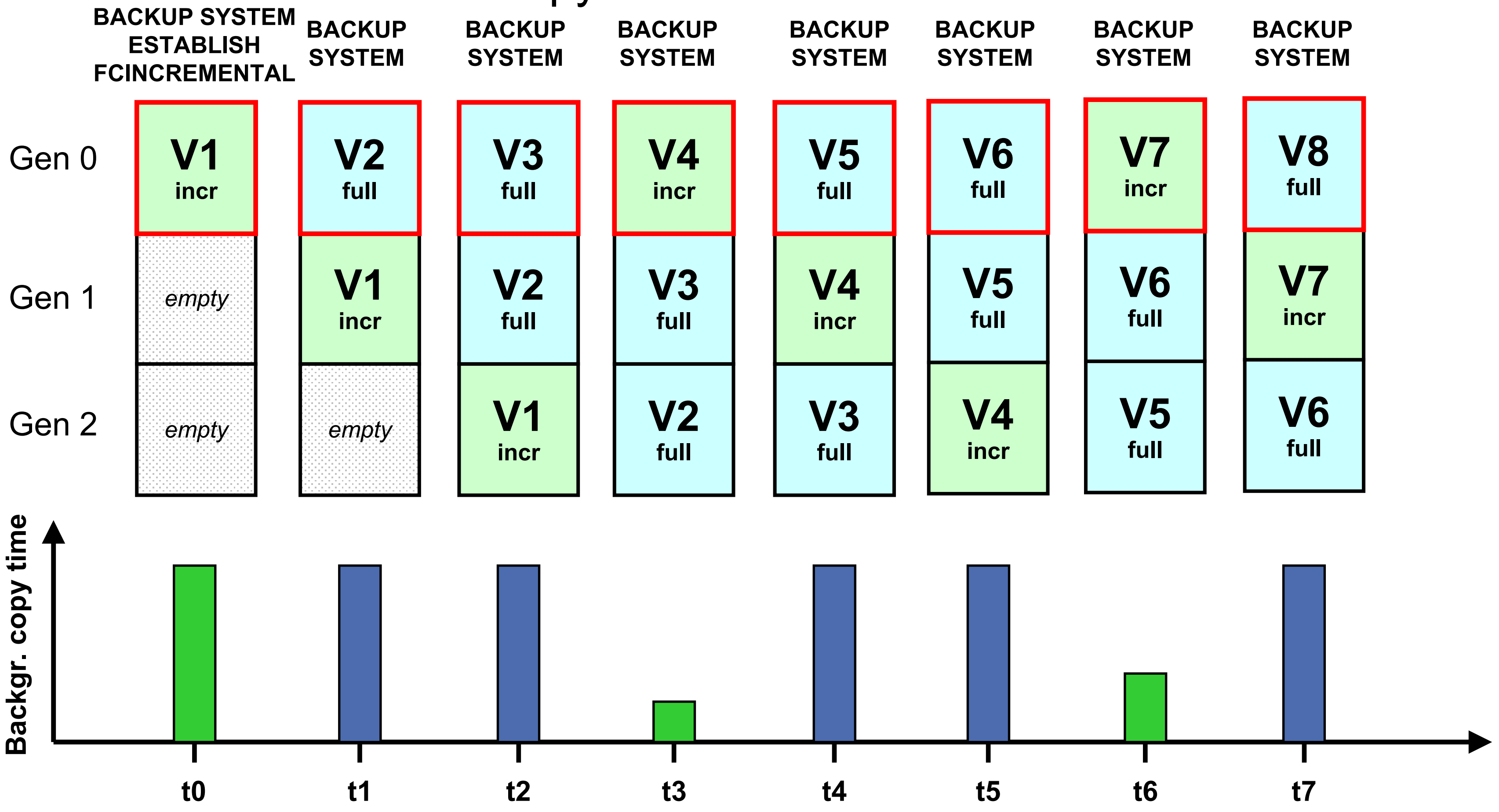
DB2 9 Utilities

- BACKUP/RESTORE SYSTEM (what's new in DB2 9)
  - Incremental Flashcopy with two FC versions:



DB2 9 Utilities

- BACKUP/RESTORE SYSTEM (what's new in DB2 9)
  - Incremental Flashcopy with three FC versions:



DB2 9 Utilities

- Progression of RECOVER log apply shown via -DIS UTIL
  - To help see progress and estimate elapsed time:

```
DSNU116I csect-name RECOVER LOGAPPLY PHASE DETAILS:
       STARTING TIME = timestamp
       START RBA = ss START LRSN = rr
       END RBA = ee END LRSN = nn
       LAST COMMITTED RBA = cc LAST COMMITTED LRSN = ll
       ELAPSED TIME = hh:mm:ss
```

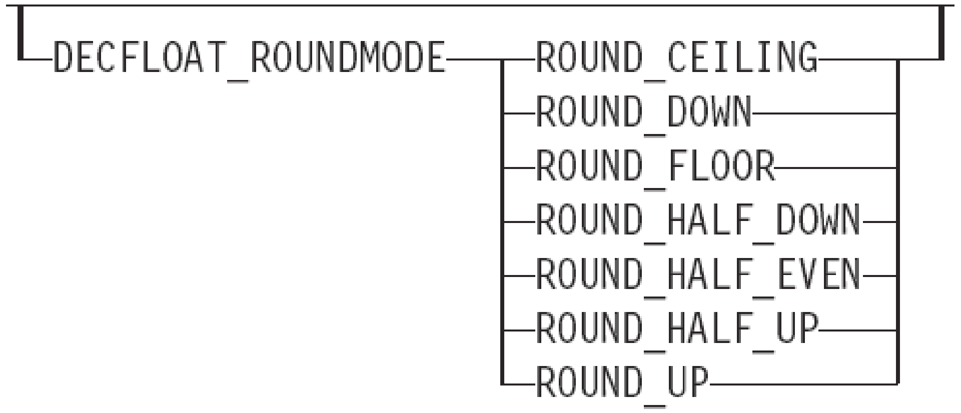
- COPY utility bufferpool usage uses MRU management of those pages read by the COPY utility
- Limit of 254 parts per REORG on a compressed table space lifted; storage consumption reduced

DB2 9 Utilities

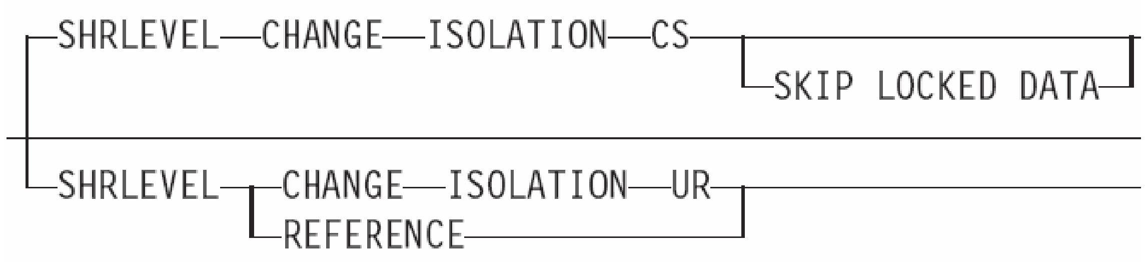
- CPU reductions - reductions mostly due to improved index processing (* with exceptions)
  - 5 to 20% in Recover Index, Rebuild Index, Reorg, and table space Image Copy* (even with forced CHECKPAGE YES)
    - Except REORG TABLESPACE SHR CHG PART with NPIs
  - 5 to 30% in Load
  - 35% in Load Partition
  - 20 to 60% in Check Index
  - 35% in Load Partition
  - 30 to 50% in Runstats Index
  - 40 to 50% in Reorg Index
  - Up to 70% in Load Replace Partition with dummy input

DB2 9 Utilities

- The LOAD and UNLOAD utilities support new data types and a new rounding mode



- The UNLOAD utility supports skipping rows that are locked for transaction updates



Utilities and Tools Futures

- Web-based administrative console
- Utilities companion product
- High priority requirements
  - Auto-stats
  - Auto-compression
  - Recover to a different table space in a different DB2 with consistency
  - Reduce RUNSTATS resource consumption
  - Dataset-level FlashCopy
  - Policy-based recovery
  - UTSERIAL elimination
  - REORG Enhancements (LOBs, etc.)
  - Eliminate SORTNUM specification
  - …

~~Manual invocation of~~
~~•RUNSTATS~~
~~•COPY/BACKUP SYSTEM~~
~~•QUIESCE~~
~~•MODIFY RECOVERY~~
~~•REORG~~

References

References

- DB2 UDB for z/OS home page
  http://www.ibm.com/software/data/db2/zos/index.html
- DB2 UDB for z/OS and OS/390 Version 7 Performance Topics, SG24-6129
- DB2 UDB for z/OS and OS/390 Version 7: Using the Utilities Suite, SG24-6289
- DB2 Magazine Fall 1998 - DB2 OS/390 Online Reorganization
  http://www.db2mag.com/db_area/archives/1998/q3/98fextra.shtml
- DB2 Magazine Quarter 2, 2003 - Programmer's Only - Programs vs. Utilities
  http://www.db2mag.com/db_area/archives/2003/q2/programmers.shtml
- Implementing Online Reorg in a Production Environment
  http://www.ibm.com/software/data/db2/os390/pdf/oreorg.pdf
- Moving Data Across the DB2 Family, SG24-6905
- Recommendations for Tuning Large DFSORT Tasks
  http://www.ibm.com/servers/storage/support/software/sort/mvs/tuning/index.html
- DFSMSrmm SMS ACS Support
  http://www.redbooks.ibm.com/abstracts/TIPS0530.html?Open

DB2 UDB for z/OS information resources

- Information center
  http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp
- Information roadmap
  http://ibm.com/software/db2zos/roadmap.html
- DB2 UDB for z/OS library page
  http://ibm.com/software/db2zos/library.html
- Examples trading post
  http://ibm.com/software/db2zos/exHome.html
- DB2 for z/OS support
  http://ibm.com/software/db2zos/support.html
- Official Introduction to DB2 for z/OS
  http://ibm.com/software/data/education/bookstore

Disclaimers & Trademarks*

Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion.

You should not rely on such information to make business plans. Any discussion of OEM products is based upon information which has been publicly available and is subject to change.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries: AIX, AS/400, DATABASE 2, DB2, OS/390, OS/400, ES/9000, MVS/ESA, Netfinity, RISC, RISC SYSTEM/6000, SYSTEM/390, SQL/DS, VM/ESA, IBM, Lotus, NOTES. The following terms are trademarks or registered trademarks of the MICROSOFT Corporation in the United States and/or other countries: MICROSOFT, WINDOWS, ODBC

Bryan F. Smith bfsmith@us.ibm.com

IBM