



Maximizing IBM DB2 V8 using the DB2 Utilities

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

TAKE BACK **CONTROL**



Abstract

- Since IBM released DB2® V8, many new functions have been added to enhance it. Significant improvements were first delivered in DB2 utilities in Version 8, and some of these functions were delivered in the service stream after Version 8 became generally available. With all these enhancements to efficiency and availability, now is the time to make sure you are taking full advantage of DB2 V8 so that your system is performing optimally.
- Join us for this complimentary teleconference and learn about DB2 V8 features, including best practices for performance, availability and usability. We'll also discuss the use of the IBM System z9™ Integrated Information Processor (zIIP). Bring your questions and be prepared to find new ways of using DB2 utilities that you may never have considered.
- You'll come away with a good understanding of how to:
 - Achieve higher availability and better performance during data maintenance
 - Identify new utility functions and apply them in your enterprise
 - Benefit from greater productivity of system programmers and DBAs



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 V8 for z/OS Utilities

TAKE BACK **CONTROL**



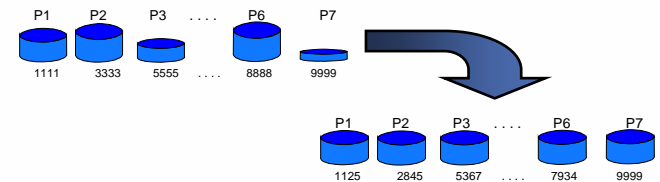
Version 8 (1 of 3)

- 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

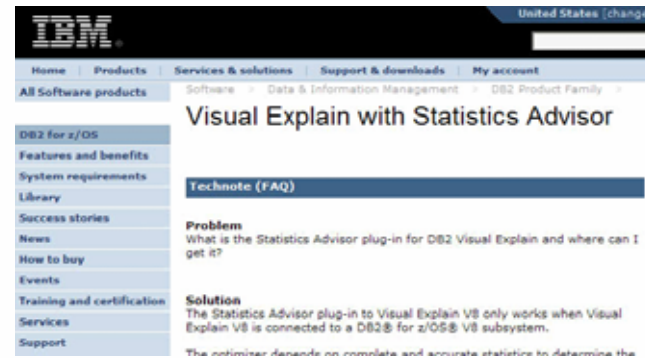
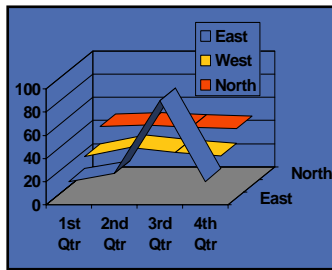
- New defaults for better "out of the box" performance
 - SORTKEYS for LOAD/REORG/REBUILD
 - SORTDATA for REORG
 - SORTDATA now allowed for 32K records with DFSORT
 - APAR PK18059 allows SORTDATA NO

- 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)



- 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 mandatory reading

<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 AP34251 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

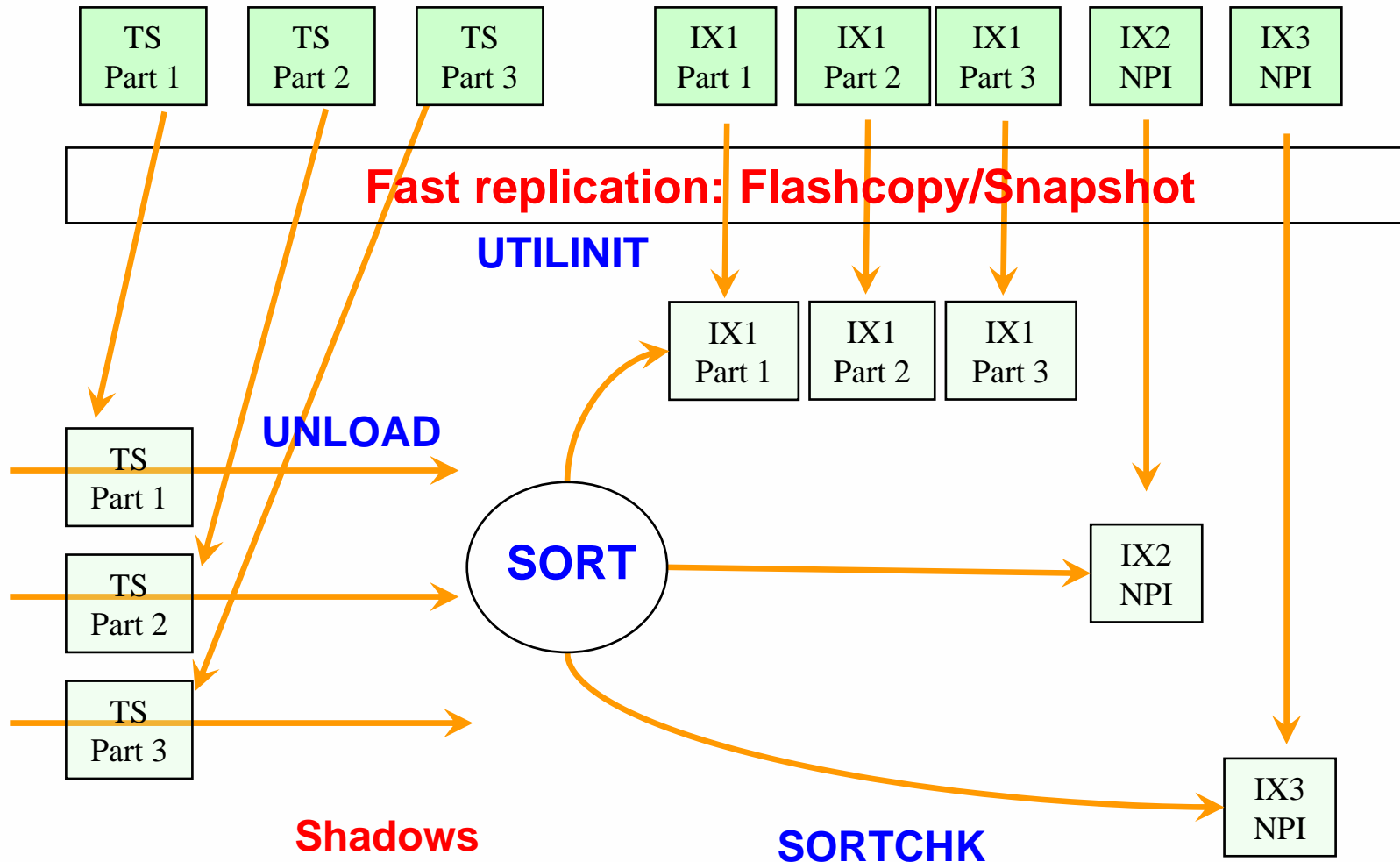


CHECK INDEX SHRLEVEL CHANGE

- 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)
```

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

Creates PDS with member for each LOB

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
```

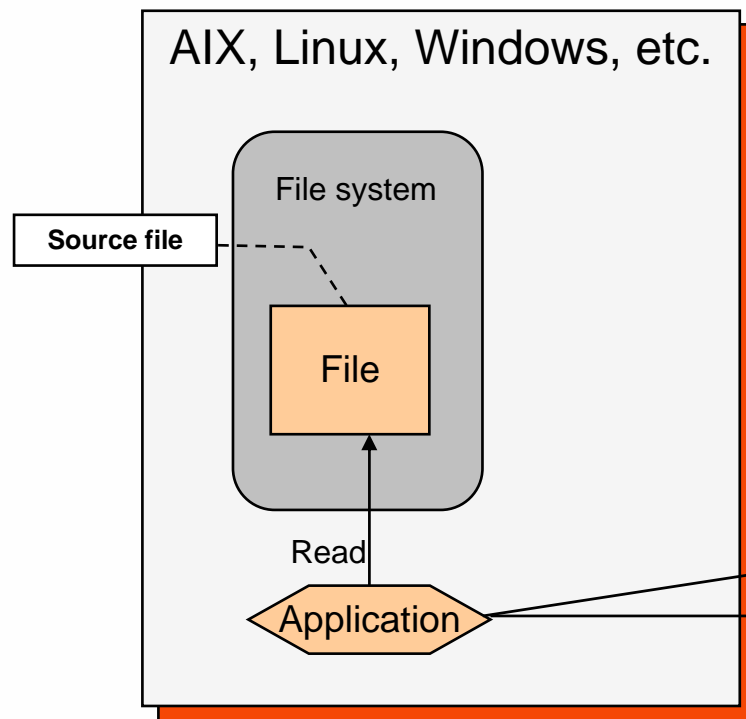
} Culprits



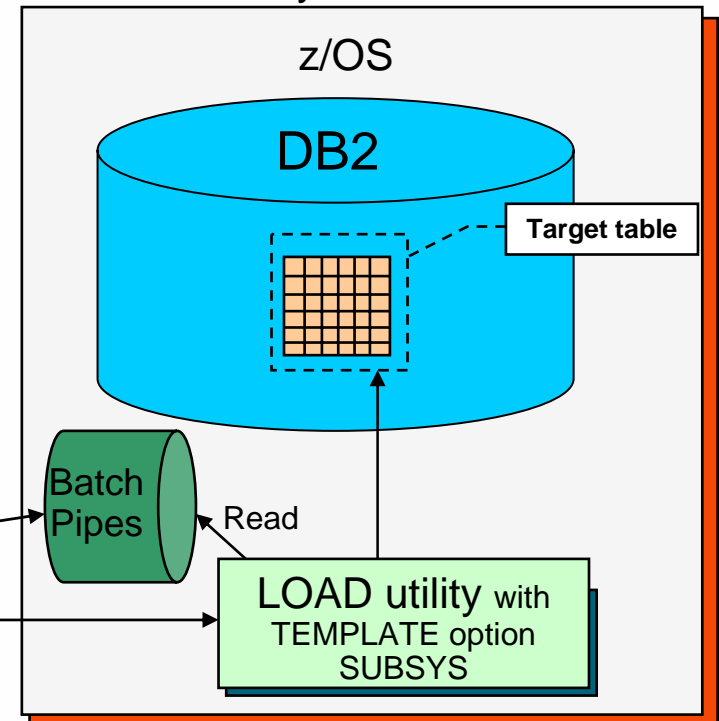
LOAD via BATCHPIPE

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

System p, x, i, etc.



System z



Transfer data via ftp

CALL DSNUTILS...

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





IBM's Unload Products

TAKE BACK **CONTROL**



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 only when input is tape/PDS mbr
 - Remove SORTWKxx and use SORTDEVT/SORTNUM
- 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

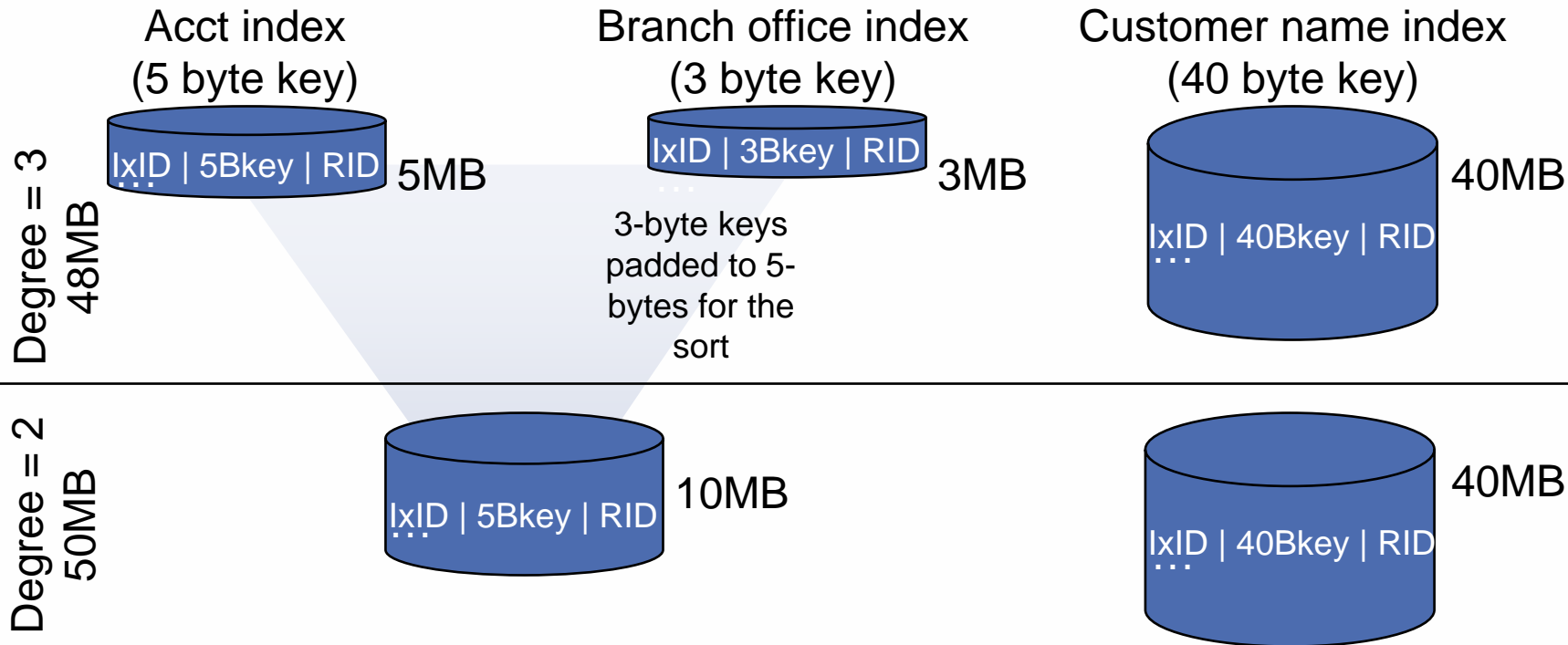
■ REORG

- 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



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 = lock timeout ZPARM/2 (30 seconds by default)
 - DRAIN_WAIT = IRLMRWT minus 5-10 seconds (to prevent timeouts)
 - RETRY = utility lock timeout multiplier (6 by default)
 - 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 statistics 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 hyperspaces, 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, \approx 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  $\neq$  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,:FILTRDEVT ,: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 or maintain 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**

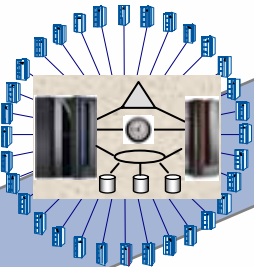


zIIP (IBM System z9 Integrated Information Processor) -- 2006

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**

Centralized data sharing across mainframes

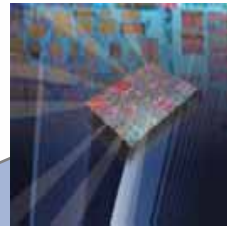


Internal Coupling Facility (ICF) 1997



Integrated Facility for Linux (IFL) 2001

Support for new workloads and open standards



System z9 Application Assist Processor (zAAP) 2004

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



IBM System z9 Integrated Information Processor (IBM zIIP)

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



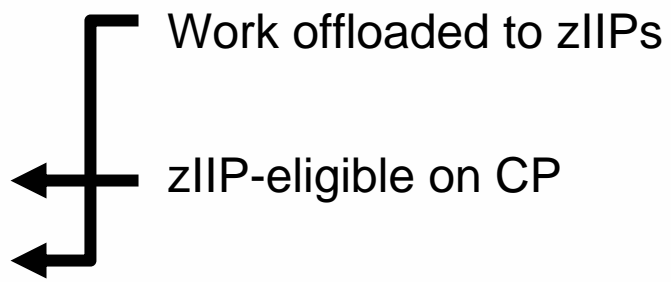
New DB2 accounting trace fields: IIP and IIPCP Time

```

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 PROC	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



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



References

TAKE BACK **CONTROL**



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



Thank You for Joining Us today!

Go to www.ibm.com/software/systemz to:

- ▶ Replay this teleconference
- ▶ Replay previously broadcast teleconferences
- ▶ Register for upcoming events

