ibm.com


# DB2 UDB for z/OS Version 8 Performance Topics Overview

*ITSO zSeries Tour 2005*
*pbruni@us.ibm.com*
*Leif.Pedersen@dk.ibm.com*

Redbooks Workshop
IBM ITSO - International Technical Support Organization

DB2 Information Management Software

© Copyright IBM Corp. 2005. All rights reserved.

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

| | | | | |
|---|---|---|---|---|
| IBM @server | Redbooks (logo)™ | CICS® | DB2 Connect™ | DB2® |
| | DFS™ | DFSMShsm™ | DRDA® | ECKD™ |
| ibm.com® | z/Architecture™ | z/OS® | AIX® | ETE™ |
| FlashCopy® | Footprint® | FICON® | IBM® | IMS™ |
| MVS™ | Notes® | Perform™ | PAL® | Redbooks™ |
| RACF® | RMF™ | S/370™ | WebSphere® | |

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.



© Copyright IBM Corp. 2005. All rights reserved.

# Notices

This information was developed for products and services offered in the U.S.A.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
*IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.


© Copyright IBM Corp. 2005. All rights reserved.

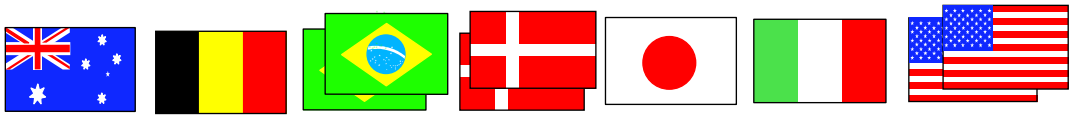
# The teams

## Redbook

Yoshio, Tsumugi, Doracilio, Mary, Paolo, Jan, Michael

## Workshop

Paolo, Glenn, Kirsten, Leif




© Copyright IBM Corp. 2005. All rights reserved.

## Agenda

- Part 1 - Overview and introduction
- Part 2 - SQL performance
- Part 3 - Subsystem performance
- Part 4 - Availability
- Part 4 - Availability continued
- Part 5 - Utilities
- Part 6 - Networking and e-business
- Part 7 - Data sharing
- Part 8 - Installation and migration


ibm.com/redbooks

© Copyright IBM Corp. 2005. All rights reserved.


# Part 1 - Introduction



© Copyright IBM Corp. 2005. All rights reserved.

# Objectives

Unicode support

64 bit addressing

Understand the performance enhancements in DB2 V8

- What are the performance enhancements in DB2 V8?
- What has been measured so far?
- How do I utilize the functions?
- What is the performance of new functions?

**Bigger**
- **table names**
- **SQL statements**
- **CI sizes**

**New SQL functionality**
multi-row fetch
MQTs
non-padded indexes

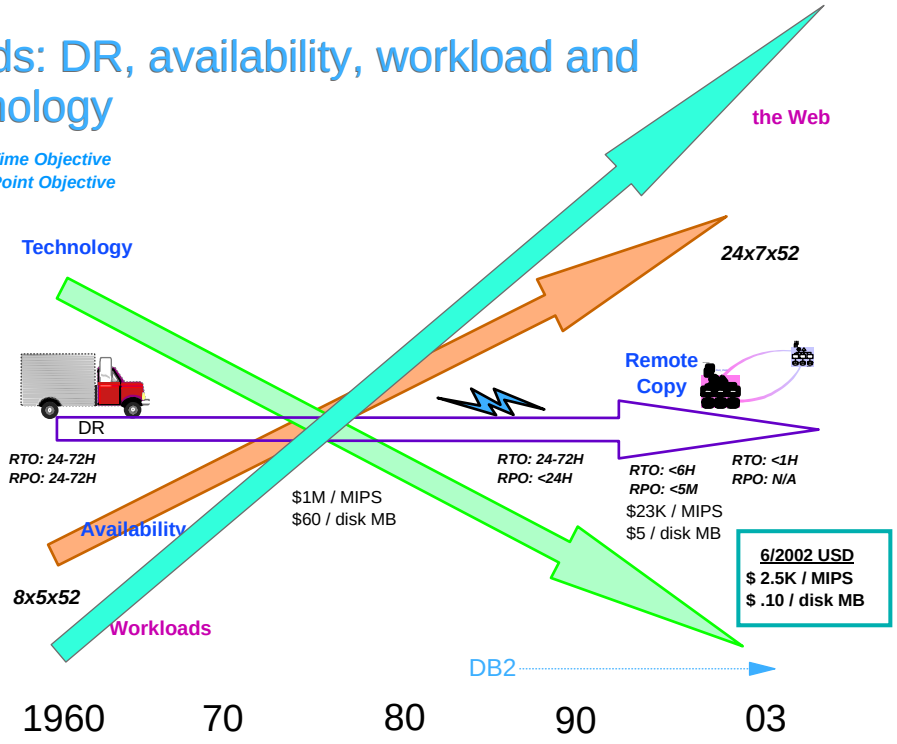
**Availability**
online schema
utilities

**Optimizer enhancements**
more stage 1
better filtering


© Copyright IBM Corp. 2005. All rights reserved.

# Trends: DR, availability, workload and technology

*RTO: Recovery Time Objective*
*RPO: Recovery Point Objective*



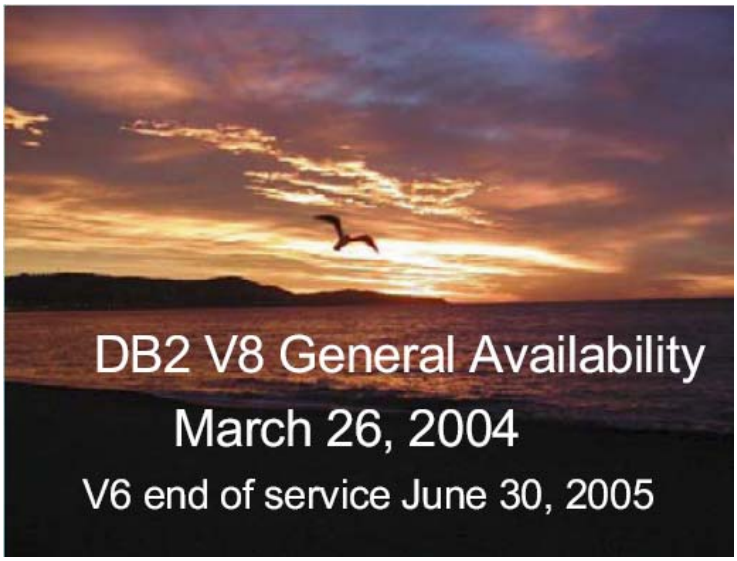


© Copyright IBM Corp. 2005. All rights reserved.

## Key performance factors

- **I/O**
- **CPU**
- **Locking**
- **Real storage**
- **Virtual storage**


© Copyright IBM Corp. 2005. All rights reserved.

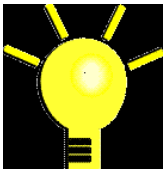



- **Requires z/OS 1.4 now**
  - z/OS 1.3 went out of service March 31, 2005
  - z/OS 1.5 is needed to exploit full DB2 V8 functionality
  - z/OS 1.6 adds a few more functions
  - z/OS 1.7 is here


© Copyright IBM Corp. 2005. All rights reserved.

## SQL enhancements in DB2 V8

- Multi-row SQL
- Materialized query tables
- Non-padded indexes
- More stage 1 predicates
- Backward index scan
- Multi-column sort merge join
- Parallel sort improvement
- Star join enhancements
- Better filtering using statistical inference on skew
- More derived predicates with IN-list
- Non-padded indexes
- Visual Explain




© Copyright IBM Corp. 2005. All rights reserved.

## Breaking the limits - Everything grows!

- 64 bit addressing
- Longer names for tables, columns, aliases and views
- Longer SQL statements
- More tables in joins
- Longer character literals
- Longer index keys
- More open data sets
- Number of active and archive logs
- More partitions in a table
- Larger VSAM CI sizes for larger page sizes


© Copyright IBM Corp. 2005. All rights reserved.

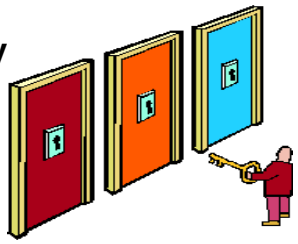
## CPU in DB2 V8

- **New version with new functionality, i.e. more code**
- **Longer names**
- **Longer and more complex SQL statements**
- **Unicode support**
- **New version with new functionality, i.e. more code**
- **Larger instructions due to 64 bit addressing**


© Copyright IBM Corp. 2005. All rights reserved.

## Security

- **Multilevel security with row-level granularity**
- **Encryption**



## Unicode

- **Unicode parsing**
- **Unicode catalog**
- **Multiple CCSIDs per SQL statement**



© Copyright IBM Corp. 2005. All rights reserved.

## Availability

- **Online schema changes**
- **Partition changes**
- **More partition independence**
- **System level point-in-time recovery**
- **Additional online DSNZPARMs**


© Copyright IBM Corp. 2005. All rights reserved.

## Utilities

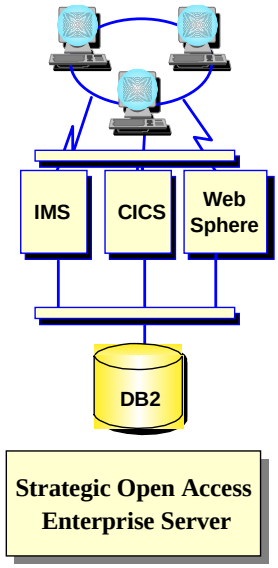
- **Online CHECK INDEX**
- **LOAD and REORG defaults**
- **RUNSTATS enhancements**
- **Cross loader**
- **BACKUP and RESTORE SYSTEM utilities**
- **Rebalancing partitions**



© Copyright IBM Corp. 2005. All rights reserved.

## Networking, e-business and DB2 family

- **DB2 Universal Driver for JDBC and SQLJ**
- **XML publishing functions**
- **Multi-row SQL**
- **Stored procedure enhancements**
- **Unicode support**
- **Better support for mismatched data types**

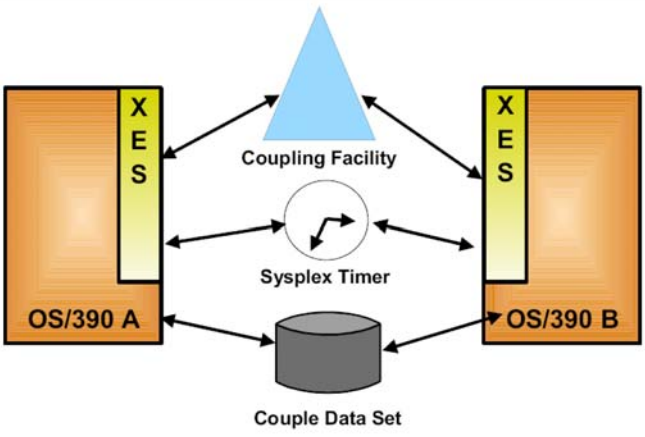




© Copyright IBM Corp. 2005. All rights reserved.

## Data sharing

- **Batching of CF requests**
- **Reducing XES contention**





© Copyright IBM Corp. 2005. All rights reserved.

IBM

# Part 2 - SQL

Redbooks Workshop

IBM ITSO - International Technical Support Organization

DB2 Information Management Software

© Copyright IBM Corp. 2005. All rights reserved.

## Multi-row SQL

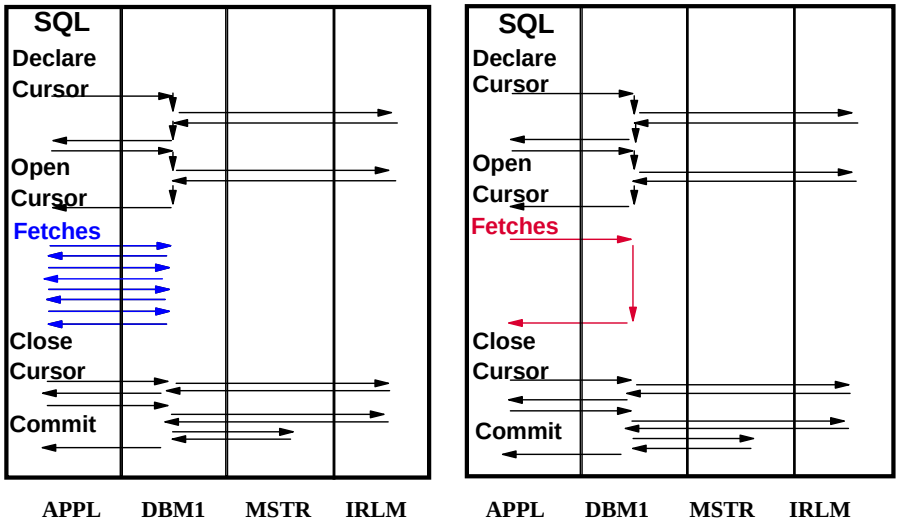
- **Process a number of rows at a time with *rowset* operations**
  - FETCH with a rowset cursor
  - Positioned UPDATE or DELETE using a rowset cursor
  - INSERT a rowset
- **Minimize number of API calls from application to DB2**
- **Host variable array (HVA) for each column**
  - HVA is supported in COBOL, PL/1, C and C++
- **GET DIAGNOSTICS provides the return codes for each row**



© Copyright IBM Corp. 2005. All rights reserved.

## Multi-row FETCH - API call reduction

Single-row Fetch vs. Multi-row Fetch





© Copyright IBM Corp. 2005. All rights reserved.

## Multi-row FETCH

Declare HVAs with 10 elements for each column

```
DCL COL1(10)  CHAR(8);
DCL COL2(10)  CHAR(8);
DCL COL3(10)  BIN FIXED(31);
```

Declare a CURSOR C1 and fetch 10 rows using a multi-row FETCH

```
EXEC SQL
    DECLARE  C1  CURSOR  WITH ROWSET POSITIONING  FOR
    SELECT  *  FROM TABLE1;

EXEC SQL   OPEN C1;

EXEC SQL
    FETCH NEXT ROWSET FROM C1   FOR 10 ROWS
    INTO  :COL1, :COL2, :COL3;
.....
```


© Copyright IBM Corp. 2005. All rights reserved.

# Multi-row UPDATE and DELETE

Update 10 rows using a positioned multi-row FETCH/UPDATE on a rowset cursor

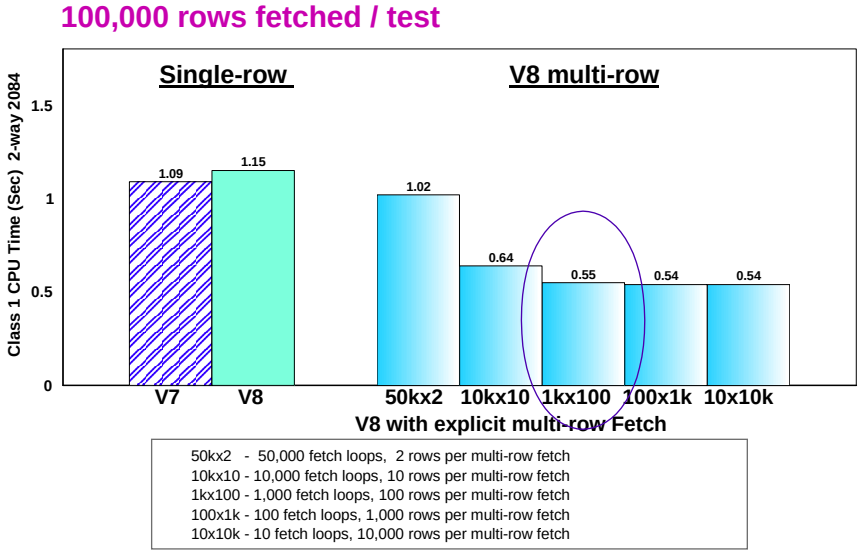
```
EXEC SQL
    UPDATE TABLE1
    SET COL3=COL3+1
    WHERE CURRENT OF C1;
```

Delete 10 rows using a positioned multi-row FETCH/DELETE

```
EXEC SQL
    DELETE FROM TABLE1
    WHERE CURRENT OF C1;
```


© Copyright IBM Corp. 2005. All rights reserved.

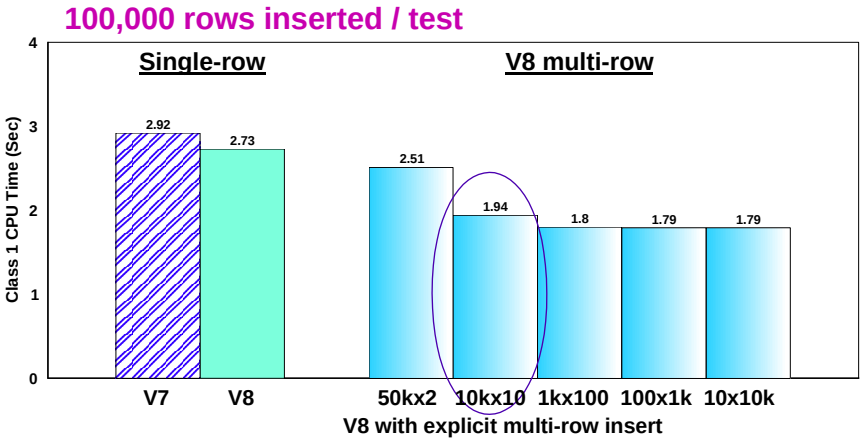
# Multi-row FETCH performance





© Copyright IBM Corp. 2005. All rights reserved.

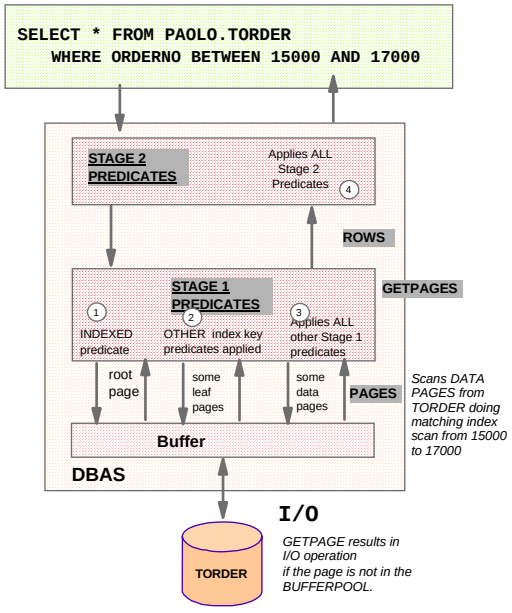
# Multi-row INSERT performance




50kx2 - 50,000 insert loops, 2 rows per multi-row insert
10kx10 - 10,000 insert loops, 10 rows per multi-row insert
1kx100 - 1,000 insert loops, 100 rows per multi-row insert
100x1k - 100 insert loops, 1,000 rows per multi-row insert
10x10k - 10 insert loops, 10,000 rows per multi-row insert


© Copyright IBM Corp. 2005. All rights reserved.

# Analysis of stage 1 and 2 predicates



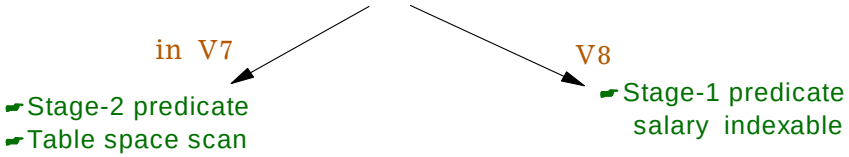


© Copyright IBM Corp. 2005. All rights reserved.

# Example of mismatched numeric types

```
employee ( name    character (20),
           salary   decimal (12,2),
           deptid   character (3) );
```

```
SELECT * FROM employee
    WHERE salary > :hv_float ;
```




- Stage-2 predicate
- Table space scan

- Stage-1 predicate
  salary indexable


© Copyright IBM Corp. 2005. All rights reserved.

# Example of mismatched string types

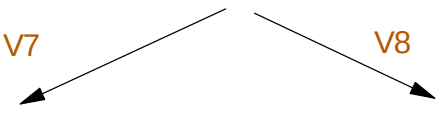
```
SELECT * FROM employee
   WHERE deptid = 'M60A' ;
```

CHAR(3)

or

```
SELECT * FROM employee
   WHERE deptid = :HV ;
```

:HV > 3 bytes




- Stage-2 predicate
- Tablespace scan

- Stage-1 predicate
- Could use index on deptid column


© Copyright IBM Corp. 2005. All rights reserved.

# Mismatched numeric data types

```
SELECT COUNT(*) FROM CV, PL
WHERE  PL.LP = 19861025
  AND CV.CS = PL. LP;
```

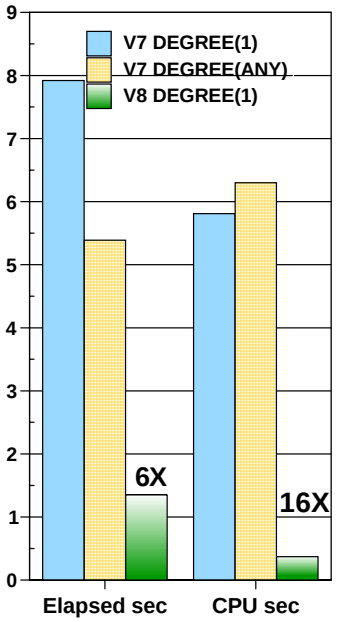
Data type of LP is DEC(8)
Index PLIX on LP column
Data type of CS is INT
Index CVIX on CS column

V7 PLAN_TABLE

| IXNAME | METHOD | MATCHCOL |
|---|---|---|
| CVIX | 0 | 0 |
| PLIX | 1 | 1 |

V8 PLAN_TABLE

| IXNAME | METHOD | MATCHCOL |
|---|---|---|
| PLIX | 0 | **1** |
| CVIX | 1 | **1** |





© Copyright IBM Corp. 2005. All rights reserved.

# Different length string data types

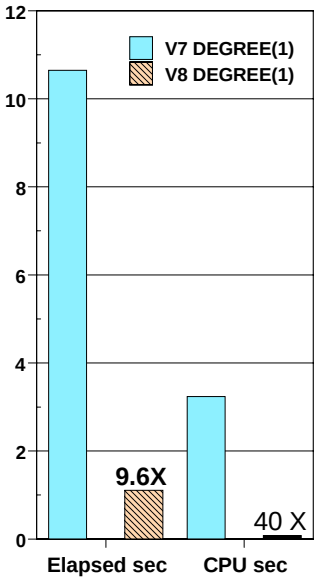
```
SELECT BRN.REG,PLC.POL,LEN,PLC.BRA,
       CMA.SUR,TRC,CST,COV.CVI,LPC,CSC,CVT
FROM BRN,PLC,COV,CIA,CMA,CLB,CCT,CAG
WHERE PLC.BRN = BRN.BRN
  AND PLC.POL = COV.POL
  AND PLC.POL = CIA.POL
  AND PLC.POL = CAG.POL
  AND COV.POL = CIA.POL
  AND COV.POL = CAG.POL
  AND COV.CVI = CIA.CVI
  AND CIA.CLI = CLB.CLI
  AND CAG.CMA = CMA.CMA
  AND COV.CVC = CCT.CVC
  AND COV.CST = 'MAILD  /* CHAR(5) <-> CHAR(6) */
  AND COV.CVT IN ('A ','B ','G ','P ')
ORDER BY  REG,PLC.POL,PLC.BRN,SUR;
```

V7 PLAN_TABLE

| TBNAME | IXNAME | ACCESS TYPE | MATCHCOL |
|---|---|---|---|
| COV | COVIX | I | 0 |
| PLC | PLCIX | I | 1 |

V8 PLAN_TABLE

| TBNAME | IXNAME | ACCESS TYPE | MATCHCOL |
|---|---|---|---|
| CCT | | R | 0 |
| COV | COVIM | I | 1 |





© Copyright IBM Corp. 2005. All rights reserved.

## Stage 1 and indexable predicates - Conclusions

- **Predicates comparing mismatched string data types are stage 1 and indexable with minor exceptions**
- **Predicates comparing numeric data types are stage 1 and indexable with minor exceptions**
- **Be careful with casting - No longer necessary and can dictate a suboptimal access path**
- **Rebind plans and packages to get better access paths**
- **Consider creating an index for the mismatched columns to enable index access**
- **Apply the fix for APAR PK06270: in V8 Compatibility Mode, predicates with special registers may incorrectly be marked as Stage 2 non indexable**


© Copyright IBM Corp. 2005. All rights reserved.

## Sort with backwards index scan

Index on (Col1 ASC, Col2 DESC)
Forward index scan on (Col1 ASC, Col2 DESC)
Backward index scan on (Col1 DESC, Col2 ASC)
No other combination can use the index

| Query | DB2 Version | Access type in PLAN_TABLE | Notes |
|---|---|---|---|
| **SELECT EMPNO FROM EMP ORDER BY EMPNO ASC** | V7 | I | Base case |
| **SELECT EMPNO FROM EMP ORDER BY EMPNO DESC** | V7 | I | Sort needed |
| **SELECT EMPNO FROM EMP ORDER BY EMPNO ASC** | V8 | I | Same as V7 |
| **SELECT EMPNO FROM EMP ORDER BY EMPNO DESC** | V8 | **I** | Sort no longer required |


© Copyright IBM Corp. 2005. All rights reserved.

## Better filter factor estimation with multiple predicates

- **Queries which contain two or more predicates referencing a table may get better access paths**
- **If frequency and correlation statistics are not available, the optimizer will try to infer them**
- **Improved query selectivity estimation using statistical inference on column skew and correlation**
- **Significant performance improvement for some complex queries**


© Copyright IBM Corp. 2005. All rights reserved.

## Multi-column sort merge join

- **Use of multi-column predicates in sort merge join**
  - In V7
    - Only the first join column is considered, other join predicates were applied at a later stage
  - In V8
    - Access path selection also based on multi-column predicates
    - Allows for better filtering in the join phase
    - Mismatched predicates on one or more char columns are allowed
- **Cost-based sort of new table**
- **Query parallelism is enabled for multi-column sort merge join**


© Copyright IBM Corp. 2005. All rights reserved.

## Parallel sort enhancement

- **Parallel sort is enabled for single and multiple-table sort composite**
  - In V7 only single table composite could use parallel sort
- **Cost-based decision**
- **Thresholds to disable parallel sort for short running queries**
  - Total sort data size (<2 MB, 500 pages)
  - Sort data size per parallel degree (<100 KB, 25 pages)

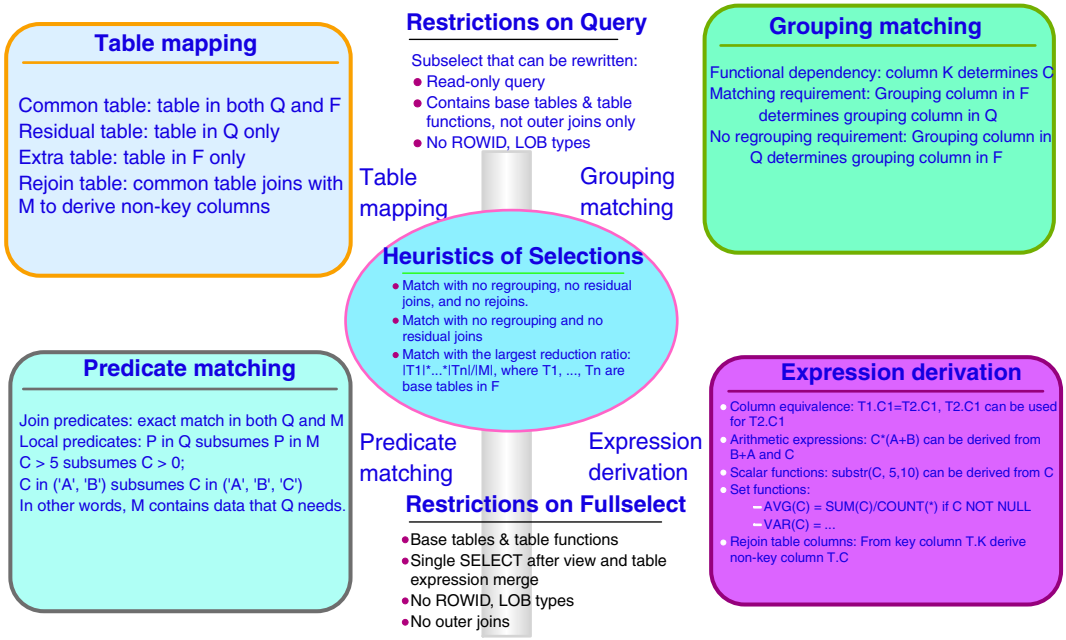

© Copyright IBM Corp. 2005. All rights reserved.

## Materialized query table (MQT)

- **Materialization of data derived from other tables with a fullselect**
  - Improve response time
  - Avoid work of scanning, aggregation and joins of detailed base tables
  - Simplify SQL
- **Source tables can be**
  - Base tables and views
  - Table expressions and user-defined table expressions
- **MQTs can be chosen by optimizer through automatic query rewrite**
  - For dynamic SQL
  - Static SQL must address MQT directly


© Copyright IBM Corp. 2005. All rights reserved.

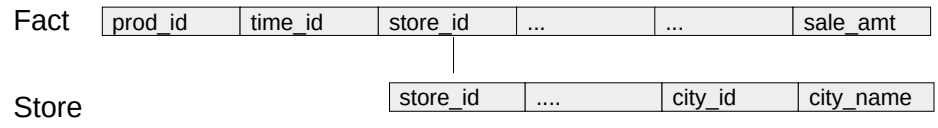
# Automatic query rewrite





© Copyright IBM Corp. 2005. All rights reserved.

# Query rewrite example




RefQ ( reference query)

```
select   S.city_id,
         sum(F.sale_amt)
from     Fact F, Store S
where    F.store_id = S.store_id
group by S.city_id
```

MQT1

```
select   S.city_id,
         sum(F.sale_amt) as sm_sale
from     Fact F, Store S
where    F.store_id = S.store_id
group by S.city_id
```

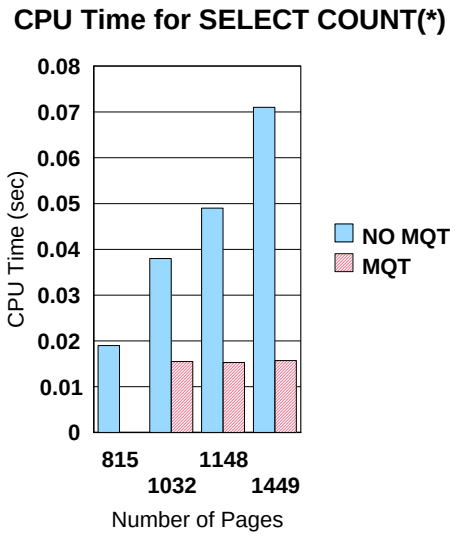
QRW (query rewrite)

```
select      A.cityid,
            A.sm_sale
from        MQT1 A
```


© Copyright IBM Corp. 2005. All rights reserved.

# MQTs and short running queries

- CPU time for SELECT COUNT(*) from tables with various numbers of 4K pages
- MQTs are not used for short running query scanning 815 page table (0.019s CPU time)
- MQTs are used for query scanning 1032 page table (0.015s cpu, 0.037s w/o MQT)

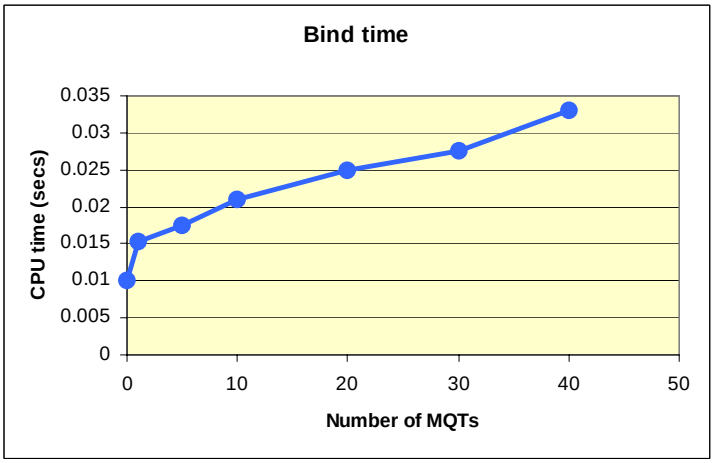




© Copyright IBM Corp. 2005. All rights reserved.

# Increase in BIND cost

- BIND cost for query (select count(*)...) without MQT is 0.01 s
- BIND cost for query using one MQT is 0.015 s
- Roughly 0.4 ms to examine each qualified MQT





© Copyright IBM Corp. 2005. All rights reserved.

# Query rewrite for a simple query

```
CREATE TABLE MQT091 (BRN,POL,COUNT) AS (
SELECT BRN,POL,COUNT(*) FROM EVE
GROUP BY BRN, POL
HAVING COUNT(*) > 50)
DATA INITIALLY DEFERRED
REFRESH DEFERRED
MAINTAINED BY USER
ENABLE QUERY OPTIMIZATION;
```

V7 PLAN_TABLE

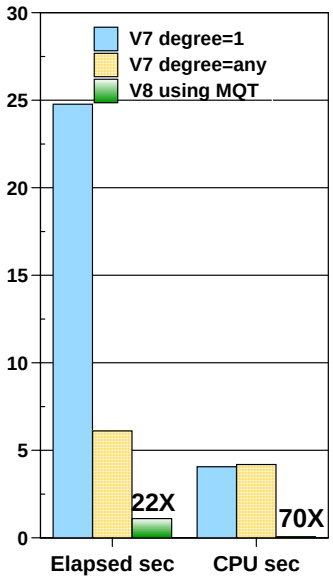
| TNAME | TABLE_TYPE | ACCESSTYPE |
|---|---|---|
| EVE | T | R |

~ 1 million rows in EVE

V8 PLAN_TABLE

| TNAME | TABLE_TYPE | ACCESSTYPE |
|---|---|---|
| MQT091 | M | R |

1388 rows in MQT





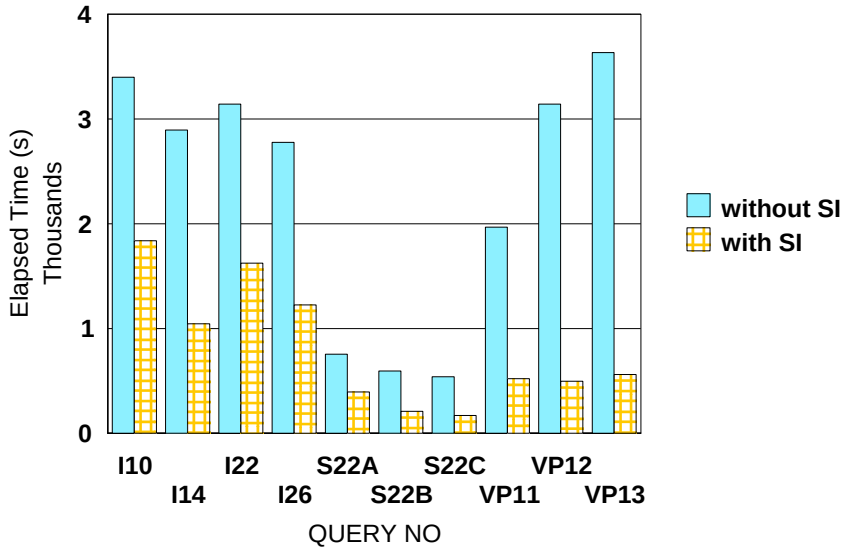
© Copyright IBM Corp. 2005. All rights reserved.

# Star join processing enhancements

- **In-memory work files (IMWF)**
  - For composite tables of outside-in process and materialized snowflakes
  - Reduces CPU and workfile I/O
  - Reduces contention on the work file buffer pool
  - Created in dedicated pool above the bar
- **Better filtering estimates for dimensions**
- **Access path enhancements**
- **Sparse indexes (SI) on work files for materialized snowflakes**
  - Already in use through APAR PQ61458 in V7, enhanced in V8
  - Binary search for key in memory, scans few rows in workfile
  - Enables use of nested loop join in inside-out phase
  - Eliminates need for sort of large composite fact table


© Copyright IBM Corp. 2005. All rights reserved.

# Sparse index (SI) performance

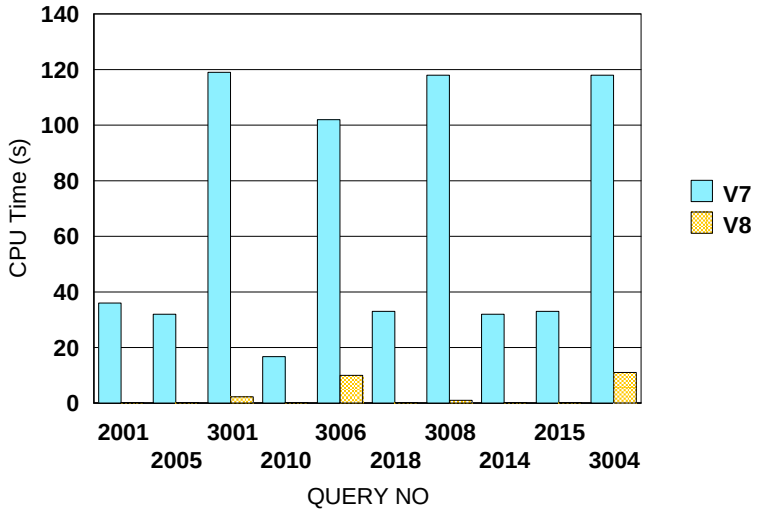




© Copyright IBM Corp. 2005. All rights reserved.

# V8 star join access path

- Large CPU reduction from better access path

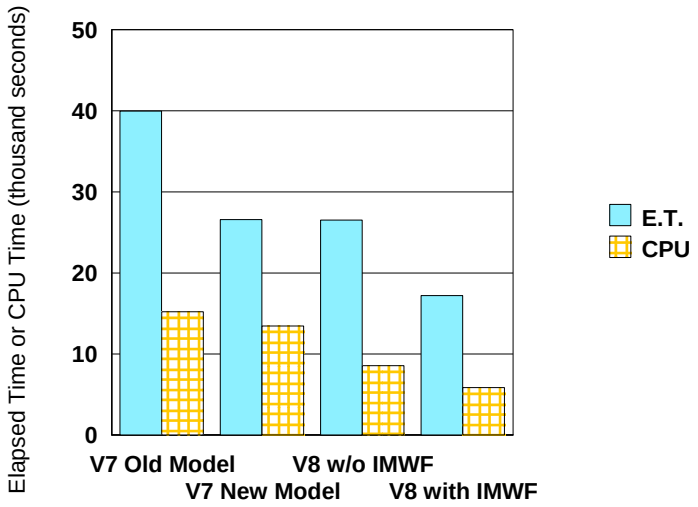




© Copyright IBM Corp. 2005. All rights reserved.

# Star join workload

Total elapsed time and total CPU time for 79 queries on highly normalized warehouse workload





© Copyright IBM Corp. 2005. All rights reserved.

# ORDER BY in SELECT INTO with FETCH FIRST ROW ONLY

- **In V7 if you wanted the first row only from a specific ordering**

```
DECLARE CURSOR C1 FOR SELECT .... ORDER BY Col1
OPEN C1
FETCH C1
CLOSE C1
```

- **With V8 the SQL is reduced to**

```
SELECT INTO.... ORDER BY Col1 FETCH FIRST ROW ONLY
```

- **Performance measurements have shown up to 30% reduction in CPU time for queries**
  - With multiple rows in the result set
  - No sort required


© Copyright IBM Corp. 2005. All rights reserved.

# Dynamic scrollable cursors

- **In V7 only static scrollable cursors**
  - Materialization of result set in declared temporary table
  - Insensitive or sensitive
  - No visibility to inserts in base tables
- **In V8 dynamic scrollable cursors**
  - Direct access to the base tables
  - Inserts to the base tables are visible, size of result set might change
  - Insensitive dynamic not allowed
  - Can be used with rowsets
- **Two new options for cursor declaration**
  - SENSITIVE DYNAMIC
  - ASENSITIVE - DB2 determines sensitivity of the cursor
    - Insensitive static when the query is read-only
    - Sensitive dynamic when the query is not read-only


© Copyright IBM Corp. 2005. All rights reserved.

# Performance of read cursor

**Read cursor - static vs. dynamic**

| Trace data | Static | | | Dynamic | | |
|---|---|---|---|---|---|---|
| | Fetch 50k Open 1M | Fetch 10 Open 1M | Fetch 50k Open 100k | Fetch 50k Open 1M | Fetch 10 Open 1M | Fetch 50K Open 100k |
| **Class 1 ET / CPU** | 17.33 / 12.57 | 16.76 / 11.91 | 5.73 / 2.57 | 0.973 / 0.879 | 0.072 / 0.011 | 0.971 / 0.885 |
| **Class 2 ET / CPU** | 17.21 / 12.44 | 16.75 / 11.90 | 5.61 / 2.43 | 0.849 / 0.743 | 0.069 / 0.009 | 0.849 / 0.748 |
| **Class 3 suspend** | 4.29 | 4.4 | 3.02 | 0.058 | 0.054 | 0.059 |


© Copyright IBM Corp. 2005. All rights reserved.

# Performance of update cursor

**Update cursor - static vs. dynamic**

| Trace data | Static | | | Dynamic | | |
|---|---|---|---|---|---|---|
| | UPDATE or DELETE 1k Open 1M | UPDATE or DELETE 10 Open 1M | UPDATE or DELETE 1k Open 50k | UPDATE or DELETE 1k Open 1M | UPDATE or DELETE 10 Open 1M | UPDATE or DELETE 1k Open 50k |
| **Class 1 ET / CPU** | 16.44 / 11.51 | 15.75 / 11.16 | 4.95 / 1.33 | 0.601 / 0.148 | 0.218 / 0.016 | 0.508 / 0.144 |
| **Class 2 ET / CPU** | 16.43 / 11.49 | 15.75 / 11.16 | 4.93 / 1.32 | 0.589 / 0.135 | 0.215 / 0.014 | 0.496 / 0.131 |
| **Class 3 suspend** | 4.5 | 4.31 | 3.56 | 0.454 | 0.195 | 0.364 |

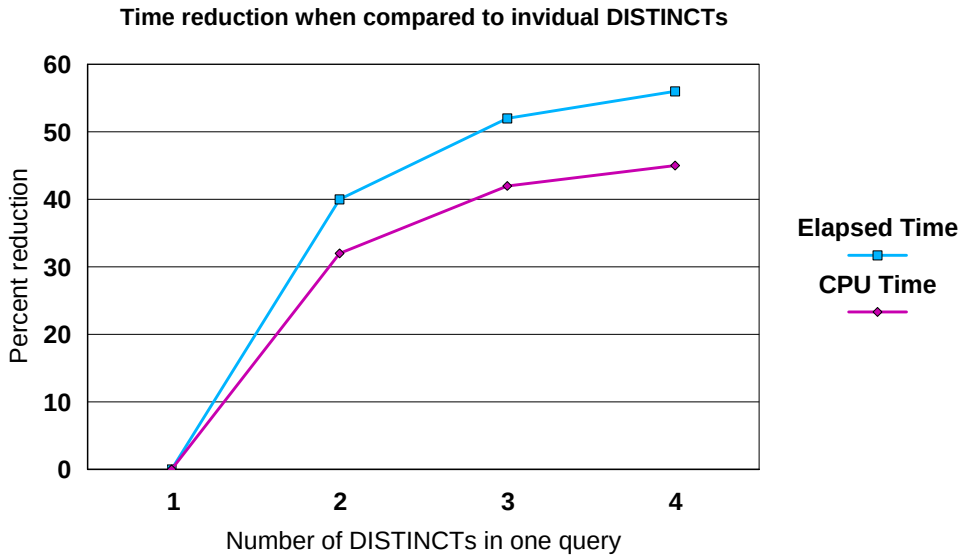

© Copyright IBM Corp. 2005. All rights reserved.

# Multiple DISTINCT clauses

- **Prior to V8 only one DISTINCT clause per subselect**
  - Except for multiple DISTINCTs on the same column, e.g. SELECT COUNT(DISTINCT(Col1)), AVG(DISTINCT(Col1)) FROM T1 GROUP BY Col1;
- **In V8 multiple DISTINCT clauses**
  - SELECT COUNT(DISTINCT Col1), AVG(DISTINCT Col2) FROM T1 GROUP BY Col1;
  - Less complex SQL
  - Some queries can now be written as one, whereas in V7 one per distinct clause was necessary
  - DB2 family compatibility
- **Not supported in V8**
  - SELECT COUNT(DISTINCT(Col1, Col2)) FROM T1 GROUP BY Col1;


© Copyright IBM Corp. 2005. All rights reserved.

# Multiple DISTINCT performance





© Copyright IBM Corp. 2005. All rights reserved.

# IN-list processing enhancements

- **V7 Enhancements to IN list processing**
  - Predicate pushdown for IN list predicates (V7 APAR PQ73454)
  - Correlated subquery transformation enhancement (V7 APAR PQ73749)
  - Activated by DSNZPARM INLISTP (default value 0 in V7 = disabled)
- **By default enabled in V8 (INLISTP=50)**
  - IN list predicate pushdown into nested table expression (NTE) or materialized view (MV)
    - Better filtering inside NTE and MV; fewer resulting rows
    - Potential index usage on columns when resolving NTE or MV
- **Correlated subquery transformation enhancement**
  - IN list predicates generated by predicate transitive closure "pulled up" to parent query block
    - Filtering done at parent level, i.e. fewer invocations of subquery executions
    - IN list predicate on parent query block may be indexable


© Copyright IBM Corp. 2005. All rights reserved.

# IN-list performance example

- **IN-list predicate push down to table expression**
- **Better filtering, fewer resulting rows**
- **Access type become IN list index access (N)**

| | V7 | V8 | Delta |
|---|---|---|---|
| Elapsed secs | 9.71 | 0.1 | -99% |
| CPU secs | 3.76 | 0.02 | -99.5% |
| Workfile getpages | 12654 | 18 | -99.9% |
| Data getpages | 33411 | 56 | -99.8% |
| Index getpages | 9712 | 11 | -99.9% |
| # of seq. prefetch | 906 | 3 | -99.9% |

```
SELECT X.DISCOUNT, SUM(X.QUANTITY), ...
FROM (SELECT *
        FROM LITEM_V1
          UNION  SELECT *
      FROM LITIEM_V2 ) AS X
WHERE X.QUANTITY <= 10  AND  X.ORDERKEY IN (1,234, 506)
```

```
SELECT X.DISCOUNT, SUM(X.QUANTITY), ...
FROM (SELECT *
        FROM LITEM_V1  WHERE X.ORDERKEY IN (1.234.506)
          UNION  SELECT *
      FROM LITIEM_V2 WHERE X.ORDERKEY IN (1.234.506)
) AS X
WHERE X.QUANTITY <= 10 AND X.ORDERKEY IN (1.234.506)
```

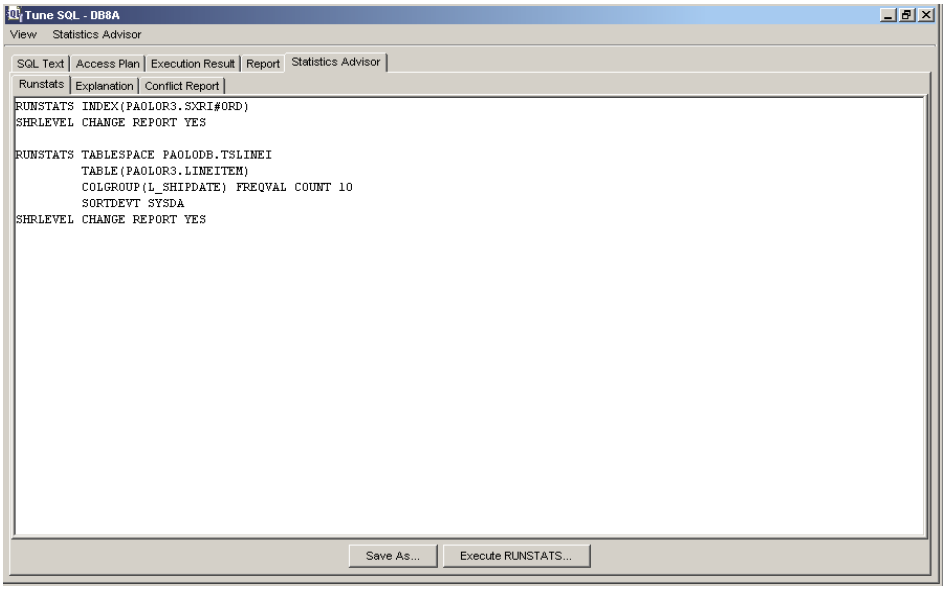

© Copyright IBM Corp. 2005. All rights reserved.

# Visual Explain - Prerequisites

- **DB2 Connect V8.1PE**
- **Stored procedures**
  - DSNWZP
    - For browsing subsystem parameters
    - Service SQL
  - DSN8EXP
    - EXPLAIN with stored procedure, supplied through APAR UQ92323
  - DSNUTILS
    - Enabling execution of the runstats statements generated by Statistics Advisor from Visual Explain
- **APARs**
  - UQ89372: Enable EXPLAIN STMTCACHE ALL keyword for explaining all statements that have been cached
  - UQ91075: Explainable flag not always set properly


© Copyright IBM Corp. 2005. All rights reserved.

## A simple query - Statistics Advisor





© Copyright IBM Corp. 2005. All rights reserved.

## SQL performance in V8 - Key points

- **Multi-row SQL is a significant performance booster in both distributed and local environments. Limiting the number of API calls may reduce CPU consumption with up to 50%**
- **MQTs offer an efficient way of speeding up long running queries by pre-computing the query and saving it a table. Especially relevant in data warehousing environments on queries with lots of aggregation**
- **More stage 1 and indexable predicates, primarily on mismatched data types. Useful in distributed environment because of limitations of columns types in Java and C/C++**
- **Lots of internal optimizer enhancements resulting in better access path selection and better performance**


© Copyright IBM Corp. 2005. All rights reserved.

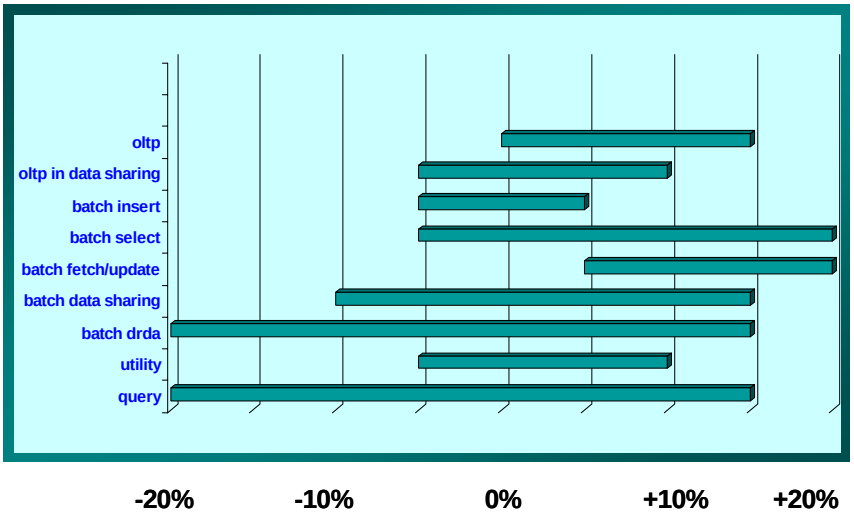
# Part 3 - DB2 subsystem

Redbooks Workshop
IBM ITSO - International Technical Support Organization

DB2 Information Management Software

© Copyright IBM Corp. 2005. All rights reserved.

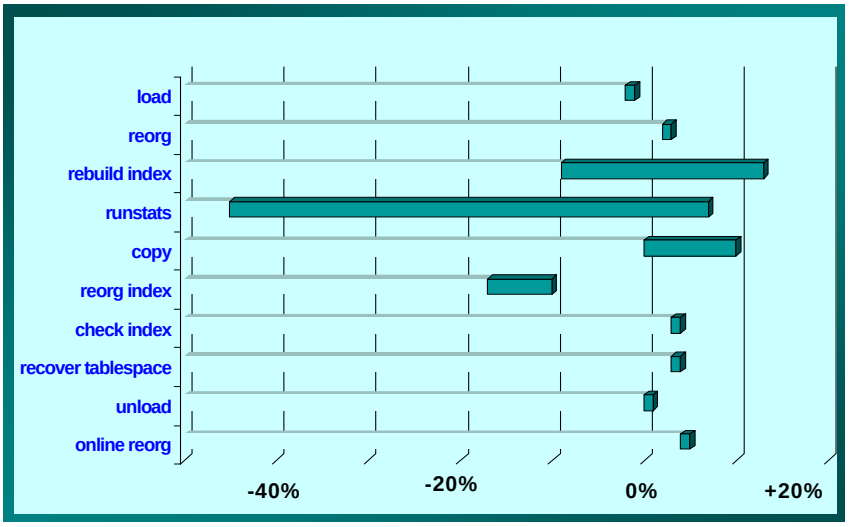
## CPU range by workload





© Copyright IBM Corp. 2005. All rights reserved.

## V7 vs. V8 Utility CPU measurements





© Copyright IBM Corp. 2005. All rights reserved.

## DB2 CPU conclusion

- **Average regression of CPU from DB2 V7 to DB2 V8 is 0 - 10%**
  - Moved storage above the 2 GB bar
  - Long names
    - CHAR(8) to VARCHAR(18) or VARCHAR(128)
  - Long index keys
    - 255 to 2000 bytes
  - Longer and more complex SQL statements
    - 32 KB to 2 GB
  - Extended support for Unicode
    - Multiple CCSIDs in the same SQL statement
- **Average regression of CPU for utilities is -5% to +10%**


© Copyright IBM Corp. 2005. All rights reserved.

## Tuning for CPU usage in DB2 V8

- **Increase the size of buffer pools and other pools that was reduced in V7**
- **Rebind critical plans and packages with RELEASE(DEALLOCATE)**
  - In data sharing this is less necessary in V8
- **Turn on thread reuse again**
- **Consider turning off DSNZPARMs EDMBFIT, CONSTOR and MINSTOR**
- **Long term page fix your buffer pools (can save 8%CPU)**
- **Use multi-row FETCH and INSERT (coding required in local)**


© Copyright IBM Corp. 2005. All rights reserved.

## DB2 storage pools

- **DB2 private virtual storage can be associated to three types of storage pools:**
  - Fixed storage (pools and control blocks)
    - Buffer pools
    - Buffer pools control blocks
    - EDM pool
    - Compression dictionaries
    - Data space lookaside buffers (DB2 V7)
    - Data space buffer pool control blocks (DB2 V7)
    - Hiper pool control blocks (DB2 V7)
  - Variable storage
    - Used for every active thread (thread activity related)
  - Stack storage
    - Calling modules save areas (thread and DB2 agents related)


© Copyright IBM Corp. 2005. All rights reserved.

## Virtual storage constraint before DB2 V8

- **Each address space has an addressing range of 2 GB based on 31 bit addressing**
  - Maximum of 16 MB available "below the line"
  - Maximum of 2032 MB available "above the line"
- **Practical maximum available to DB2 below and above the line is much less**
  - Approximate 1.3 - 1.6 GB
- **Storage is allocated into different subpools which have unique characteristics e.g., SP229**
  - Storage acquired via GETMAIN
  - Storage released by FREEMAIN


© Copyright IBM Corp. 2005. All rights reserved.

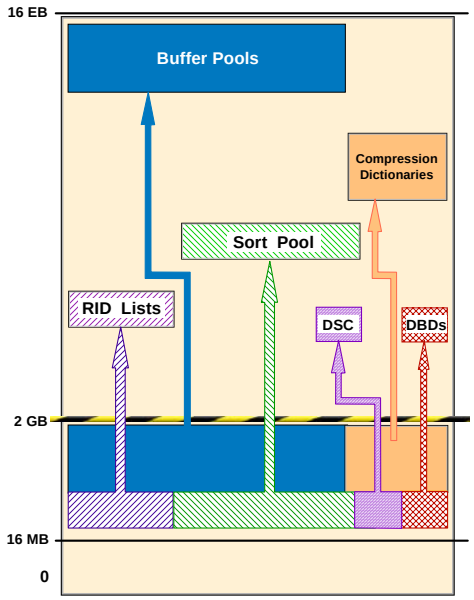
## 64 bit evolution

- 64 bit zSeries, z/Architecture *hardware* support (z800, z900, z990)
- Operating system support
  - 64 bit *Real storage* support (OS/390 V2R10 ESAME mode +)
    - Large amounts of real storage (for performance)
    - Improve **all** versions of DB2
    - V6 data space advantages (outperforms hiperpool)
  - 64 bit *Virtual storage* support (z/OS V1.2 +)
    z/OS 64-bit Virtual Storage Roadmap
    - Exploited by **DB2 V8** (requires z/OS 1.3)
      - Move large memory areas above the bar
    - Exploited by **IRLM V2.2** (that ships with DB2 V8)
      - Locks always in private storage above the bar (PC=YES enforced)
    - Improve scalability, availability, ease
    - Hiperpool/dataspace no longer needed


© Copyright IBM Corp. 2005. All rights reserved.

## Pools moved above the bar

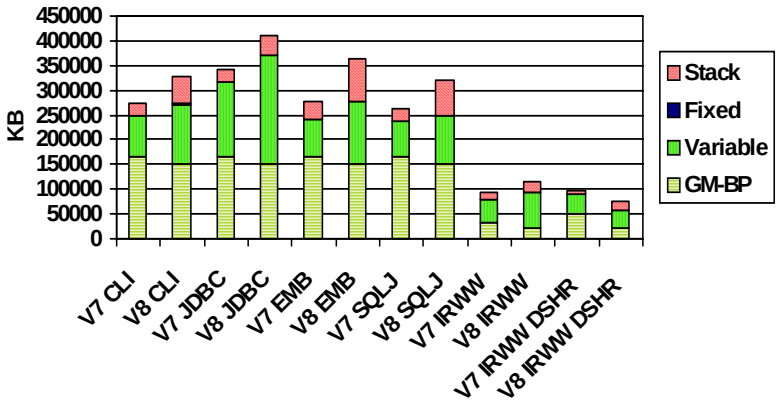
- Buffer pools (and CBs)
- DBD in the EDM pool
- DSC
- Compression dictionaries
- Sort pool
- RID pool
- Castout buffers
- Global dynamic statement cache
- (LOBs)





© Copyright IBM Corp. 2005. All rights reserved.

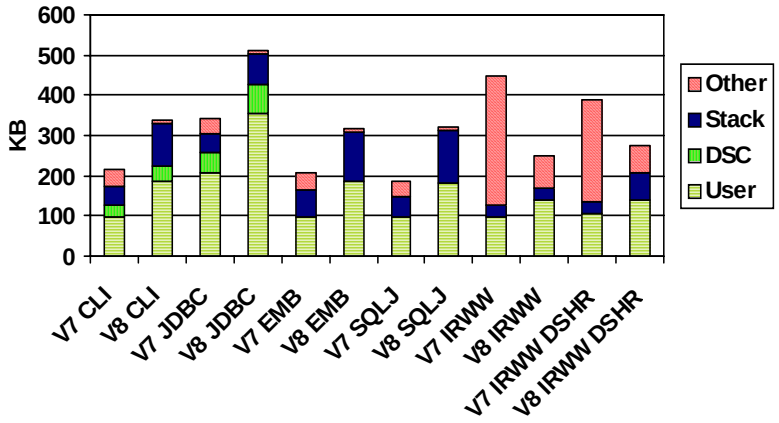
## DBM1 below 2 GB virtual storage comparison





© Copyright IBM Corp. 2005. All rights reserved.

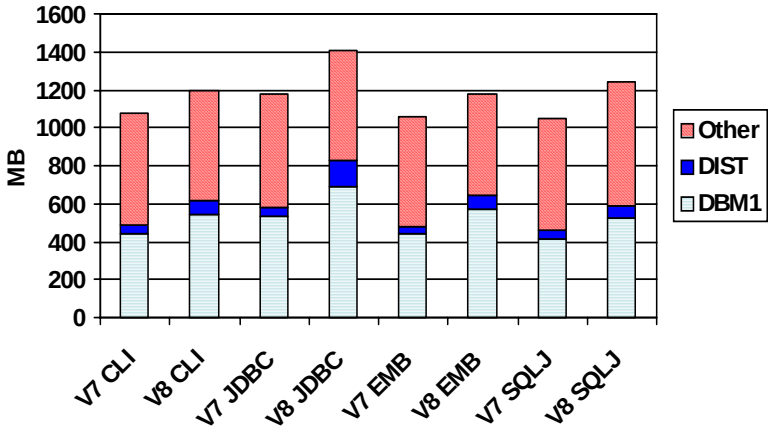
# DBM1 below 2 GB - User thread storage comparison





© Copyright IBM Corp. 2005. All rights reserved.

# V7 and V8 **real** storage comparison





© Copyright IBM Corp. 2005. All rights reserved.

## Real storage - Conclusion

- **Real storage usage has been growing at about 20 to 30% per year since 1985 to support performance and scalability**
  - More and bigger buffer pools, other pools, threads
- **V8 continues a similar trend by removing bottlenecks which would have prevented the exploitation of bigger real storage**
- **If everything under user control is kept constant, 1 to 20% increase in real storage is typically observed**
  - Smaller % increase for larger DB2 subsystem
  - Larger % increase for smaller DB2 subsystem


© Copyright IBM Corp. 2005. All rights reserved.

## Influences on VSCR

- **Using data spaces (max 8 M pages each)**
  - If you have allocated buffer pools and global dynamic statements cache (DSC) in data spaces
    - Then the VSCR would be less
- **Using hiper pools (max 8 GB total)**
  - As cache for buffer pools
    - Then the VSCR would be less
- **LOBs use data spaces with versions prior to V8**
  - Storage goes now above the bar in the DBM1 address space
    - They are not influential with respect to VSCR


© Copyright IBM Corp. 2005. All rights reserved.

## VSCR - Recommendations

- **If you have not measured the amount of storage per thread, here are some rough estimates:**
  - Low end: 200 to 400 KB - Static
  - Mid range: 500 KB to 2 M - Static / Dynamic
  - High end: 3 MB to 10 MB or higher - Dynamic, heavy sort, parallelism applied, long running persistent thread WFI, CICS Protected Entry Threads with many packages with RELEASE(DEALLOCATE) or (RELEASE(COMMIT) with CONTSTOR=NO)
- **APAR PQ96772 (UK0091)**
  - Provides significant virtual storage relief for large, dynamic SQL cache environments by moving the DSC control blocks above the bar
- **APARs II04309 and II10817 for DB2 storage issues and z/OS suggestions**
  - APAR PK01336 (UK01034/3) for DIST asid


© Copyright IBM Corp. 2005. All rights reserved.

## VSCR - Conclusions

- **DB2 V8 64-bit support does not totally eliminate VSC below the 2 GB bar in the DBM1 address space**
- **V8 provides valuable additional relief, but the relief varies by installation**
- **DB2 V8 uses more storage for each thread in the DBM1 address space due to:**
  - Thread structures increased in size to support larger names in DB2
  - The RDS OP pool storage that was used at bind time in V7, is now acquired from the thread's variable storage pool
  - Some DB2 structures must increase in size to support 64-bit addresses

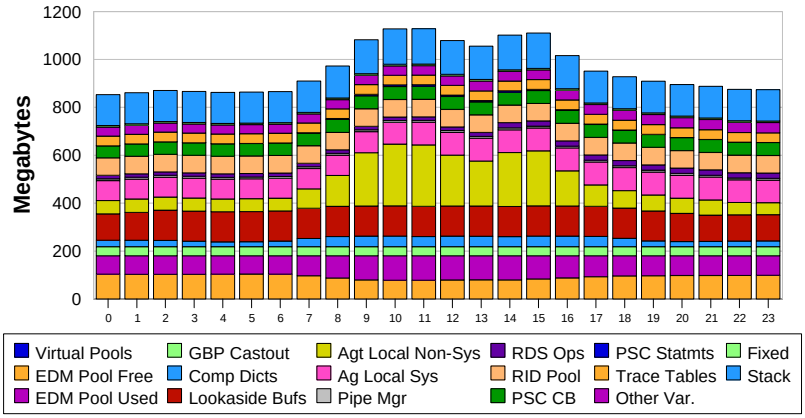

© Copyright IBM Corp. 2005. All rights reserved.

## Monitoring DBM1 storage

- **New messages: DSNB536I and DSNB610I**
- **IFCID 225, now in Statistics Class 6**
  - -STA TRACE(S) CLASS(1,3,4,5,**6**)
- **PQ99658 (UK04394/3) adds 225 IFCID records to Statistics Class 1 for both V7 and V8, records are also cached during the interval, the default goes from 30 to 5 mins.**

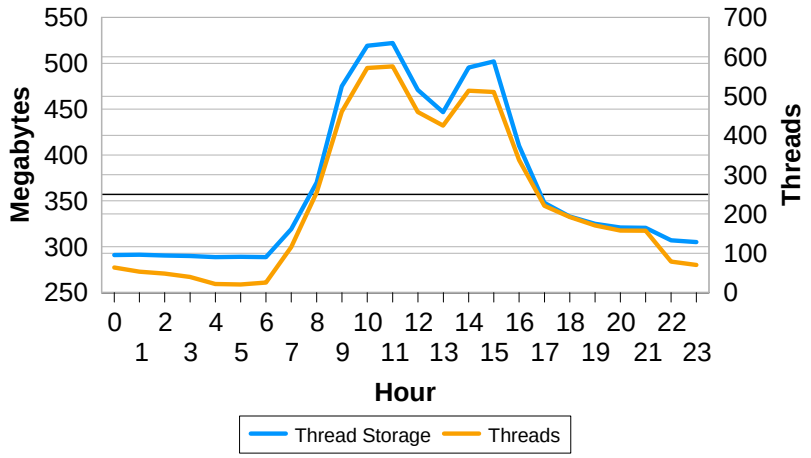
Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

## V7 DBM1 very detailed virtual storage usage




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

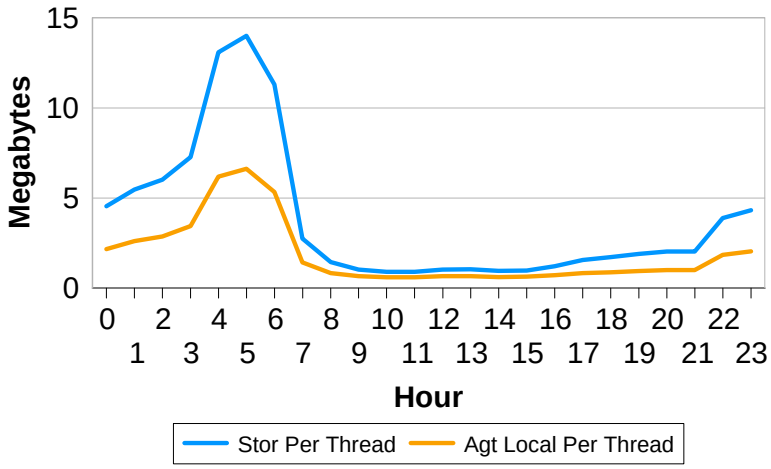
# V7 DBM1 thread memory - Headline numbers





© Copyright IBM Corp. 2005. All rights reserved.

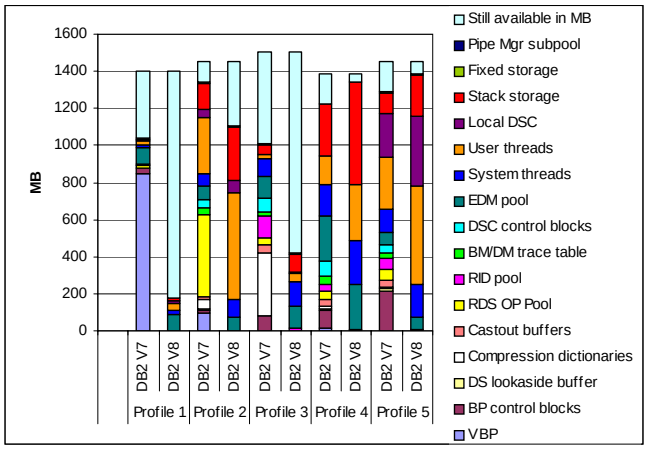
# V7 DBM1 thread memory - Per thread profile





© Copyright IBM Corp. 2005. All rights reserved.

# DB2 V7 vs. V8 DBM1 storage profile





© Copyright IBM Corp. 2005. All rights reserved.

# DB2 V7 vs. V8 DBM1 storage estimates

The graph shows the expected storage used in DBM1 address space after migration to DB2 V8 based on the IFCID 225 record from DB2 V7.

Customer who already have moved everything into data spaces and are close to be or are constrained in DB2 V7 will probably still have problem in DB2 V8.

- USER THREAD storage is expected to grow with 40 - 90%. Distributed thread will be in the high end and non distributed will be in the low end.
- SYSTEM THREAD storage is expected to grow by 40%.
- STACK storage may increase by 100%.
- LOCAL STATEMENT CACHE may increase by 60% - remember that Local statement cache is only used when packages are bound with KEEPDYNAMIC(YES)
- Buffer pools, Control blocks for buffer pools, DS lookaside buffer, DBD from the EDMPOOL, compression dictionaries, castout buffers, RDS OP pool will be moved above the bar.


© Copyright IBM Corp. 2005. All rights reserved.

## DBM1 storage - Conclusions

- **Many DB2 systems will get virtual storage relief in DBM1**
- **Few DB2 systems will not get virtual storage relief in DBM1**
- **IFCID 225 - Storage area which will increase**
  - User threads
  - System threads
  - Stack storage
  - Local statement cache (KEEPDYNAMIC YES)


© Copyright IBM Corp. 2005. All rights reserved.

## Buffer pool long term page fixing

- **The operating system normally "fixes" the storage temporarily during the life of the I/O operation**
  - This extra CPU cost can be as high as 10% for I/O intensive profiles
- **New ALTER BUFFERPOOL parameter PGFIX**
  - It fixes the buffer pages once in memory and keeps them fixed in real storage
  - DB2 will allow up to 80% of the real storage of the z/OS LPAR to be fixed
    - This limitation is in place to protect other workloads on the LPAR from being "squeezed" by DB2


© Copyright IBM Corp. 2005. All rights reserved.

# Long term page fixing

Non-data sharing

| PGFIX | NO | YES | Delta (YES / NO) |
|---|---|---|---|
| DBM1 (msec / commit) | 0.455 | 0.322 | -29% |
| Total DB2 CPU time (msec / commit) | 2.436 | 2.238 | -8% |
| ITR (msec / commit) | 587.22 | 601.42 | +6% |

Data sharing

| PGFIX | | | | Delta (YES / NO) |
|---|---|---|---|---|
| DBM1 (msec / commit) | 0.560 | 0.563 | 0.456 | 0.457 | -19% |
| Total DB2 CPU time (msec / commit) | 3.318 | 3.306 | 3.109 | 3.357 | -2% |
| Group ITR (msec / commit) | 890.70 | | 897.00 | | +1% |


© Copyright IBM Corp. 2005. All rights reserved.

# Long term page fixing - Candidates

I/O intensity = pages accessed/#buffers

**One example with 100,000 buffers total (400MB)**

| | #buffers | Pages read or written | Buffer i/o intensity |
|---|---|---|---|
| BP0 catalog/directory | 2000 | 10000 | 5 #2 |
| BP1 workfile | 10000 | 20000 | 2 #3 |
| BP2 in-memory index or data | 30000 | 100 | 0.003 |
| BP3 other index | 53000 | 100000 | 1.9 #4 |
| BP4 other data | 5000 | 100000 | 20 #1 |


© Copyright IBM Corp. 2005. All rights reserved.

# IRLM V2.2

- **DB2 locks always reside above the 2 GB bar**
- **Maximum number of locks is 100 million - around 16 time more the IRLM V2.1**
- **Locks no longer reside in ECSA, so PC=YES is always used**
- **New parameters**
  - MLMT - Max storage for locks
  - PGPROT - Page protect
- **PC and MAXCSA parameters are no longer used in the IRLM V2.2. However, you must still specify them in the IRLM JCL for compatibility reasons**


© Copyright IBM Corp. 2005. All rights reserved.

# IRLM CPU - V2.1 vs. V2.2

| DB2 | V7 | | V8 | | V8 | | Delta for P1 % (V8-V7) / V7 | Delta for P1 % (V8-V7) / V7 |
|---|---|---|---|---|---|---|---|---|
| **IRLM (PC=YES)** | 2.1 | | 2.2 | | 2.2 | | | |
| **Lock Protocol** | N/A | | 1 | | 2 | | | |
| **Group ITR (commits/sec)** | 823.56 | | 804.69 | | 933.41 | | -2.29% | +13.34% |
| **IRLM CPU msec/commit** | 0.330 | 0.371 | 0.335 | 0.379 | 0.014 | 0.014 | +1.85% | -96.01% |

- If 2% of the overall CPU time is consumed by IRLM 2.2
- The overall impact of having PC=YES is 0.2%, (10% * 2%). This is probably not noticeable
- In DB2 V7, the availability benefits of moving from PC=NO to PC=YES outweigh the very slight performance benefits of PC=NO in most cases

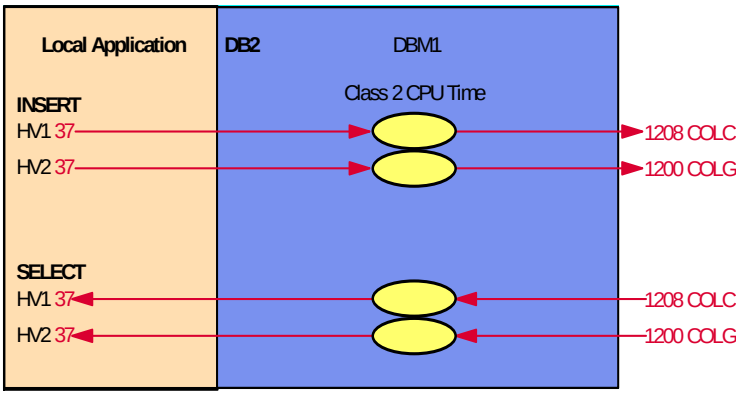
Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

## DB2 V8 Unicode support

- **Conversions generally done by Conversion Services and helped by operating system (z/OS 1.4) and zSeries hardware**
- **EBCDIC and ASCII CCSIDs must be specified in DSNHDECP**
  - And they cannot be changed (from DB2 V8)
- **SQL statements are parsed as Unicode (UTF-8) even in CM**
- **Can combine different encoding schemes (CCSIDs) in same SQL statement in NFM**


© Copyright IBM Corp. 2005. All rights reserved.

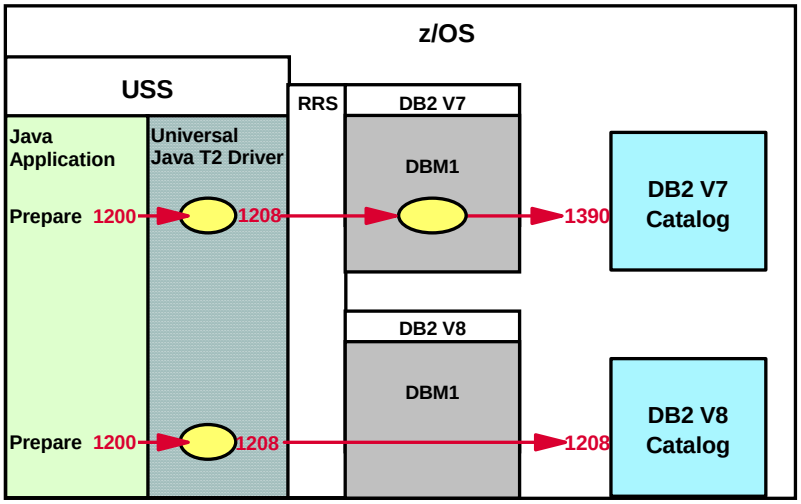
## Class 1 or class 2 CPU time




➡ Local Java clients only affect class 1 CPU time


© Copyright IBM Corp. 2005. All rights reserved.

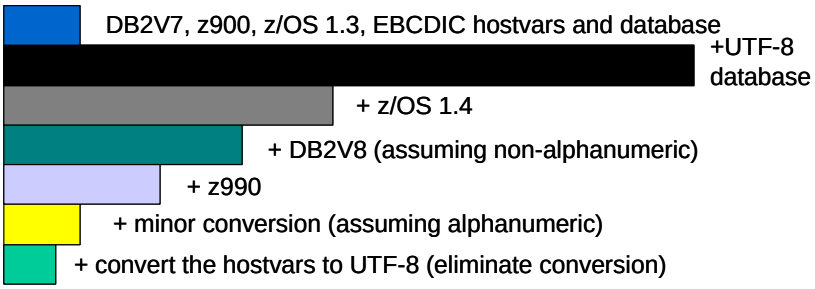
# Local prepare - Universal Java Type 2 driver





© Copyright IBM Corp. 2005. All rights reserved.

# Conversion performance overview





© Copyright IBM Corp. 2005. All rights reserved.

## Unicode - Conclusions

- **Enhancements in DB2 V8, z/OS Conversion Services and the z990 hardware, dramatically improve Unicode conversion performance**
- **Prior to z/OS Conversion Services improvements (HC3) Unicode to EBCDIC was always slower than ASCII conversion to EBCDIC**
  - Now is about 20% faster
  - Major Unicode conversion is about 12% faster
- **The cost of conversion can effect the total CPU time**
  - Table space scan of 1 million rows with no predicates or indexes
    - Takes 18.6 CPU seconds without conversion
    - Takes 63.3 CPU seconds with conversion
    - 67% of the CPU time was spent doing conversion


© Copyright IBM Corp. 2005. All rights reserved.

## Data encryption - 1

- **DB2 V8**
- **ICSF (Integrated Cryptographic Service Facility)**
- **Pre-z990**
  - Cryptographic Coprocessor is required
- **On z990**
  - CPACF (CP Assist for Cryptographic Function) instruction
  - PCIXCC (Peripheral Component Interconnect Extended Cryptographic Coprocessor) card, needed for
    - DRDA encryption
    - IBM Data Encryption Tool


© Copyright IBM Corp. 2005. All rights reserved.

## Data encryption - 2

- **New built-in functions**
  - ENCRYPT_TDES (or ENCRYPT)
  - DECRYPT_BIN
  - DECRYPT_CHAR
  - GETHINT
- **SET ENCRYPTION PASSWORD**
  - Specify a password as a key to encryption
    - Can be different for each row in a table
- **Columns must be "for bit data"**
- **LOAD and UNLOAD do not support DB2 encryption**


© Copyright IBM Corp. 2005. All rights reserved.

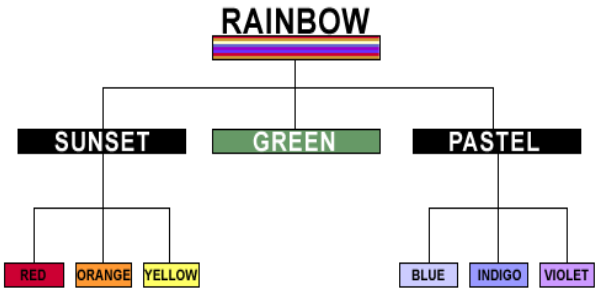
## Data encryption calculation

- **DB2 encryption cost on z990**
  - Approximately 2.3 microseconds per column
  - 0.0086 microseconds per byte encrypted
  - Fetch 100k rows with two 32 byte columns encrypted
    - 2x(2.3 + 32x0.0086)x100,000 = 520,000 microseconds
- **IBM Encryption Tool cost on z990**
  - 1 microsecond per row
  - 0.006 microseconds per byte encrypted
  - Fetch 100k rows with row size of 204 byte
    - (1 + 204x0.006)x100,000 = 220,000 microseconds


© Copyright IBM Corp. 2005. All rights reserved.

## Row level security with row granularity

- **In RACF**
  - Set up a security hierarchy (SECLEVEL) and categories (CATEGORY)
  - The SECLABEL class is active
  - Assign security labels to users





© Copyright IBM Corp. 2005. All rights reserved.

## Row level security - Performance

- **At a very rough level, the performance impact of row level security can be considered in the same ballpark as DB2 data compression with less than 5% impact for a typical online transaction but higher for CPU-bound sequential scans.**
- **The CPU increase caused by Row level security is because DB2 must now access the SECLABEL column in the table, evaluate the user's SECLABEL against the SECLABEL of the data, manage a cache of SECLABEL values and periodically consult RACF.**

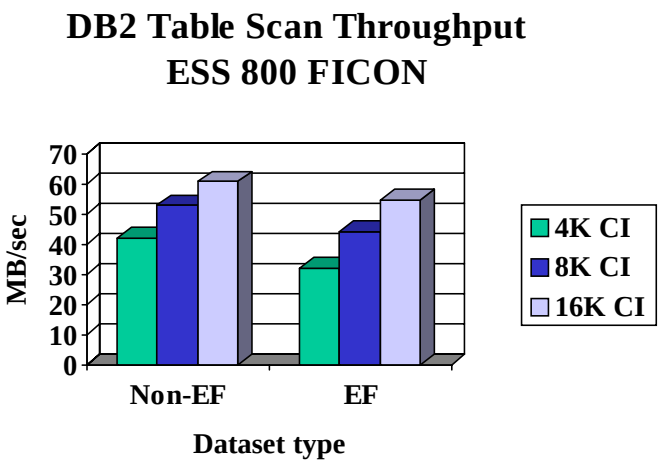

© Copyright IBM Corp. 2005. All rights reserved.

# Larger VSAM CI size in DB2 V8

- **DB2 V7 used only CI size of 4 KB**
  - Page size 8, 16 or 32 KB
    - Chain of 2, 4 or 8 CIs and requires VSAM only to allocate CIs in 4 KB
- **DB2 V8 support CI size of 4, 8, 16 and 32 KB**
  - Activated in NFM by DSNZPARM DSVCI
    - Table spaces will be converted by LOAD REPLACE or REORG
  - Indexes only use CI size of 4 KB


© Copyright IBM Corp. 2005. All rights reserved.

# Larger VSAM CI size





© Copyright IBM Corp. 2005. All rights reserved.

IBM

# Part 4 - Availability


DB2 Information Management Software

© Copyright IBM Corp. 2005. All rights reserved.

## Online schema overview

- **Ability to make the following changes while minimizing impact on availability:**
  - Change data types and lengths (char, numeric and graphic)
    - Including columns referenced within a view
    - Including columns that are part of an index
  - Add columns to indexes
  - Drop the partitioning index
  - Change the clustering index
  - Alter index VARCHAR columns to be non padded
  - Alter identity column attributes (except data type)
  - Add partition to end of table
  - Rotate partitions
  - Automatically rebalance partitions


© Copyright IBM Corp. 2005. All rights reserved.

## Online schema performance

- **Performance measurements were taken for the following:**
  - Change data types and lengths
  - Add columns to indexes
  - Create or alter an index to have true varying length character columns
  - Rotate and rebalance partitions

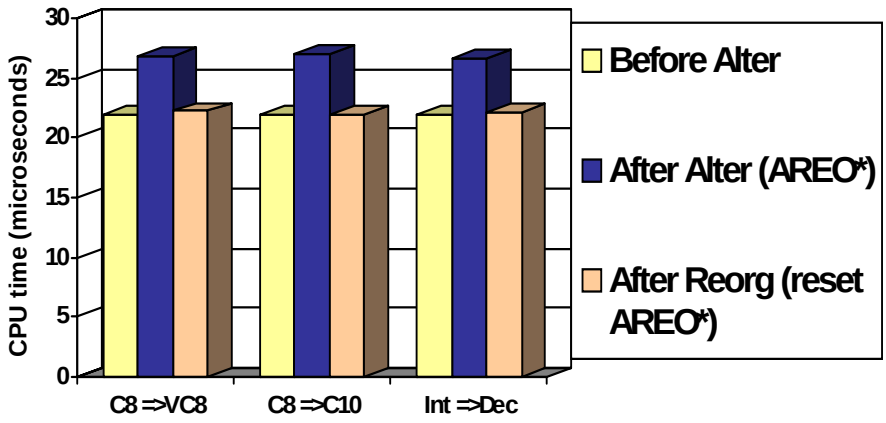

© Copyright IBM Corp. 2005. All rights reserved.

## Versioning support in V8

- **Altering existing objects may result in multiple data formats**
  - All data and image copies for an object may not be changed immediately to match new format
  - Support for multiple versions of objects needed
- **Versioning used to track object definitions at different times during the object's life**
  - Allows data access, index access, recovery to current, and recovery to a point-in-time while maximizing data availability
  - Individual rows use new version upon INSERT, UPDATE
  - Entire object converted to new version upon LOAD REPLACE, REORG TABLESPACE or REBUILD INDEX
- **Up to 256 active versions for table space, up to 16 for indexes**
  - Minimize number of active versions for best performance


© Copyright IBM Corp. 2005. All rights reserved.

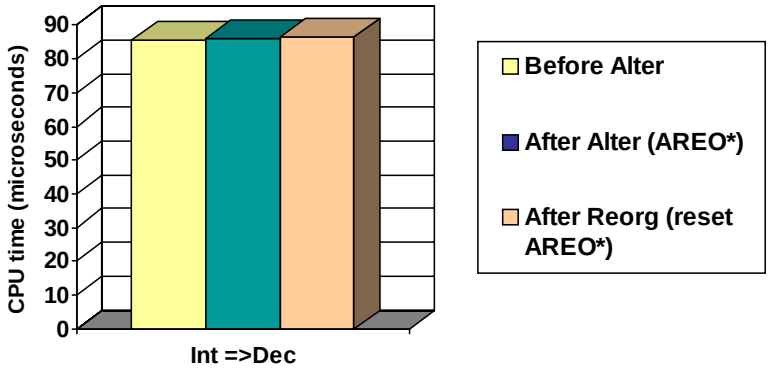
# Alter column data types - FETCH CPU times





© Copyright IBM Corp. 2005. All rights reserved.

# Alter column data types - INSERT CPU times

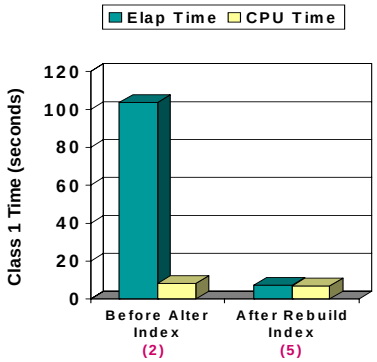




© Copyright IBM Corp. 2005. All rights reserved.

# Alter index add column measurements

SELECT Elapsed Time and CPU Time




1. DDL Create Index
   - CREATE INDEX ix1 ON tab1(c1) ...
2. SELECT c1, c2 ... WHERE c2 = 'xx'
   - Column not indexed, table space scan
3. ALTER INDEX ix1 ADD COLUMN (c2)
   - Index in RBDP
4. Rebuild Index
5. SELECT c1, c2 ... WHERE c2 = 'xx'
   - Non matching index only scan
   - 10+ times Elapsed time improvement

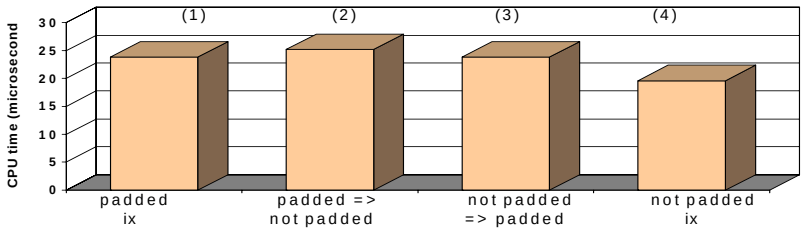

© Copyright IBM Corp. 2005. All rights reserved.

# Alter index VARCHAR columns to non padded

- **Prior to V8, VARCHAR columns in indexes are padded to the maximum length**
- **New default index option is NOT PADDED**
  - VARCHAR columns no longer padded to maximum length in index keys
- **Index-only access now available for VARCHAR data**
  - With no padding to maximum length at SELECT time
- **Index can be altered between PADDED and NOT PADDED but will be placed in RBDP state**


© Copyright IBM Corp. 2005. All rights reserved.

# Fetch CPU time - Alter index padding - 1

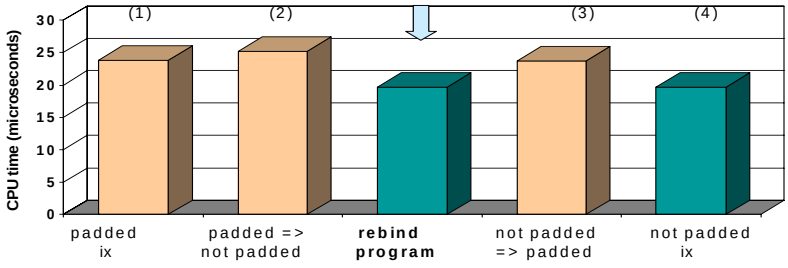



- **Fetch CPU time comparison**
  - **(2) vs. (1) : +6%**
  - **(3) vs. (1) : No difference**
  - **(4) vs. (1) : -17%**

| | (1) | (2) | (3) | (4) |
|---|---|---|---|---|
| Index only access | N | N | N | Y |


© Copyright IBM Corp. 2005. All rights reserved.

# Fetch CPU time - Alter index padding - 2



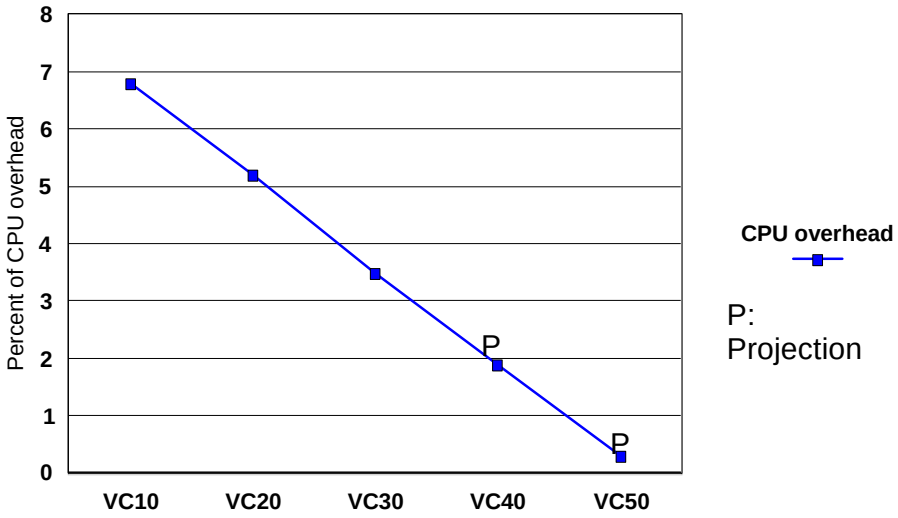

**Fetch CPU time reduced after application program rebind - access path changed to index-only access**


© Copyright IBM Corp. 2005. All rights reserved.

# Average CPU overhead for 15 Queries

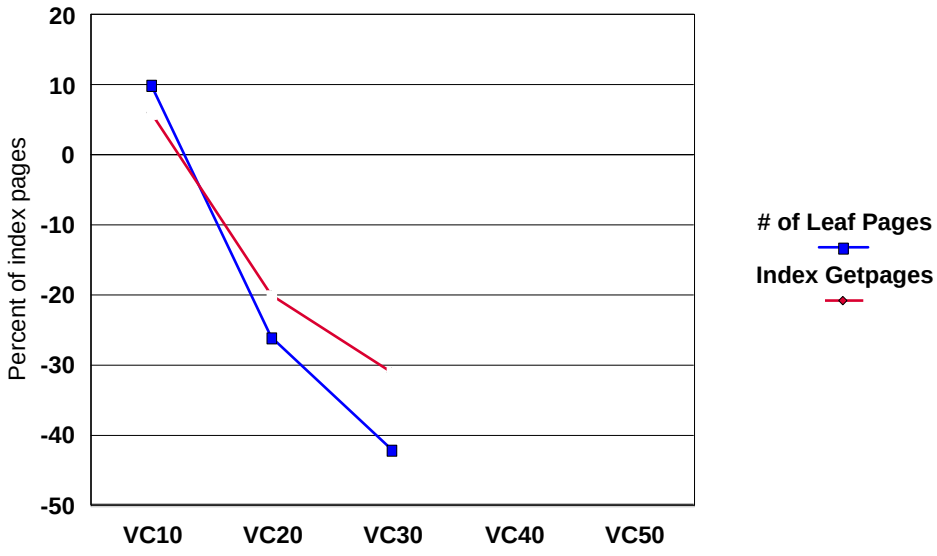
Query CPU overhead using NOT PADDED vs. PADDED keys

Two varchar keys in two indexes are altered from padded to not padded



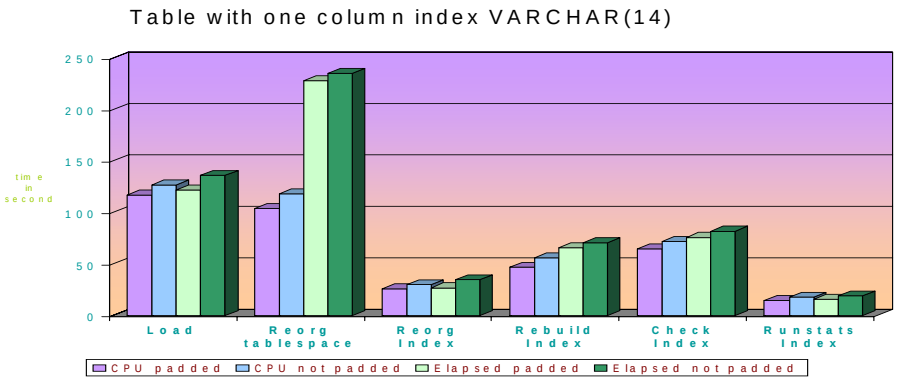

Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

# Avg. Index Leaf Pages and Getpages for 15 Queries




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

# NOT PADDED index impact on Utilities





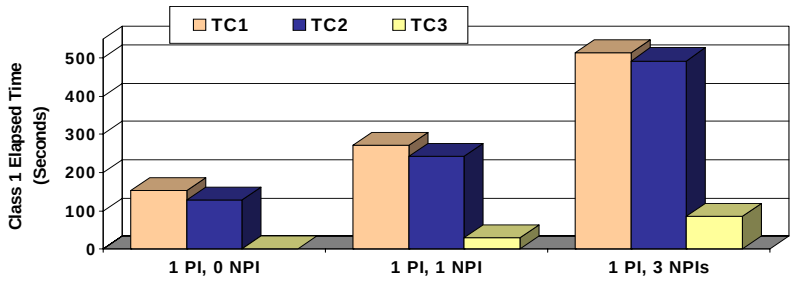
© Copyright IBM Corp. 2005. All rights reserved.

# Dynamic partition management

- **Prior to V8 you had to drop and recreate table space to add partitions or rotate partitions**
- **ALTER TABLE ADD PARTITION statement adds a partition at the end**
  - New partition immediately available - no REORG needed
- **ALTER PARTITION ROTATE FIRST TO LAST statement logically makes first partition the last**
  - Second partition becomes the first
  - Can specify that data rows in new "last" partition are deleted
  - Emptied partition immediately available - no REORG needed
- **New REBALANCE keyword on REORG**
  - Automatically rebalances a range of partitions
  - Plans, packages and cached dynamic statements invalidated
  - Affected partitions placed in REORG Pending (REORP) state


© Copyright IBM Corp. 2005. All rights reserved.

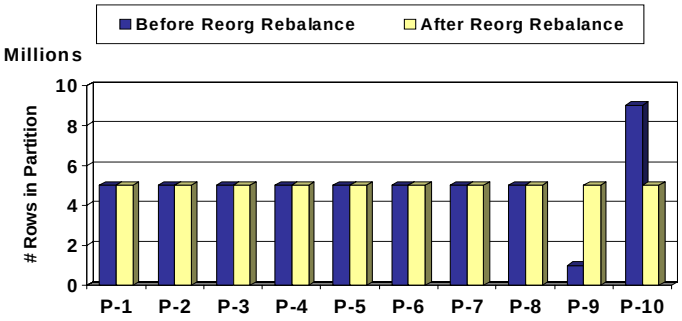
# Rotate partition - Elapsed time comparison




- TC1 = Delete Part 1 from Table
- TC2 = Alter Table Alter Part Rotate First To Last
- TC3 = 1. Load Replace Part 1 w/ Dummy Infile
  2. Alter Table Alter Part Rotate First To Last
- TC3 can be up to 100 times faster depending on the number of NPIs


© Copyright IBM Corp. 2005. All rights reserved.

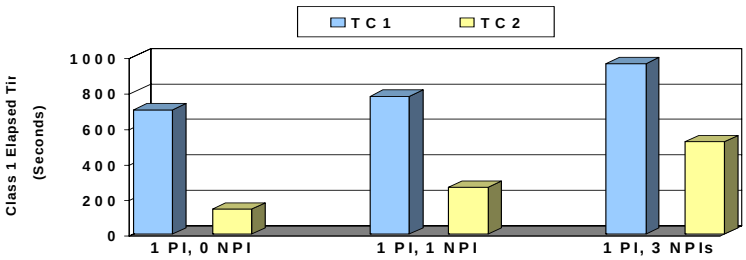
# REORG REBALANCE - Partition distribution




- Partition distribution : 8(5M) + 1(1M) + 1(9M)
- Resulted in even partition sizes


© Copyright IBM Corp. 2005. All rights reserved.

# REORG REBALANCE - Elapsed time




- Partition distribution : 8(5M) + 1(1M) + 1(9M)
- TC1 - reorg tsp rebalance all parts
  TC2 - reorg tsp rebalance parts 9 and 10
- 40% to 80% faster if Reorg Rebalance parts 9 and 10 only


© Copyright IBM Corp. 2005. All rights reserved.

# Index enhancements

- **Prior to V8, partitioning is controlled by partitioning index**
  - Only one index may be partitioned
    - Same index also is the partitioning and clustering index
  - All secondary indexes are non partitioned indexes (NPIs)
  - Contention issues on NPIs for utilities run at partition level
    - Requires BUILD2 phase for Online REORG of a partition
    - LOAD PART
- **V8 introduces table controlled partitioning**
  - No indexes required for partitioning
  - Partitioned, partitioning and clustering no longer intertwined
  - Secondary indexes may now be physically partitioned
    - Elimination of BUILD2 phase for Online REORG of a partition
    - Contention on LOAD PART is eliminated


© Copyright IBM Corp. 2005. All rights reserved.

## New classification of indexes

- **An index may / may not be correlated with the partitioning columns of the table**
  - Partitioning index (PI)
  - Secondary index
- **An index may / may not be physically partitioned**
  - Partitioned
  - Non-partitioned
- ***Clustering* index:**
  - Any index may be the clustering index!!
  - The clustering index can be unique / non-unique

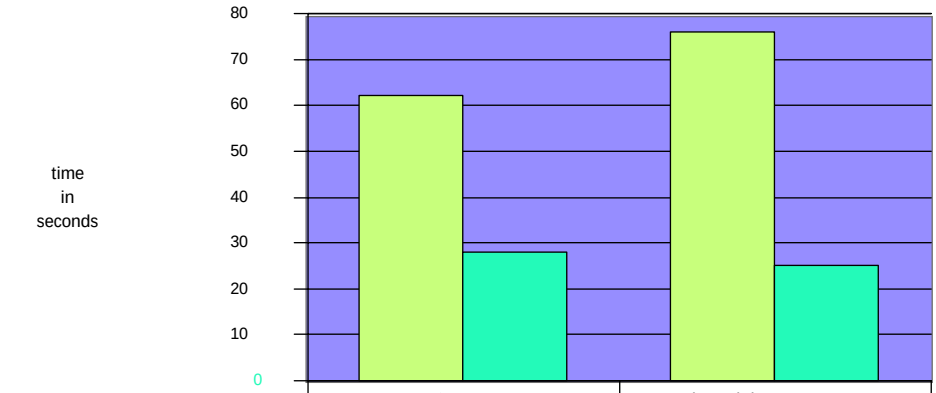

© Copyright IBM Corp. 2005. All rights reserved.

## Data partitioned secondary indexes

- **A secondary index that is physically partitioned is a data partitioned secondary index (DPSI)**
  - Created with PARTITIONED keyword
  - Data is physically partitioned according to partitioning columns of the table
- **Benefits of DPSIs are:**
  - Allow partition-level operation for REORG INDEX, RUNSTATS INDEX on secondary indexes
  - Contention on index pages for utilities operating on partition level is eliminated
  - Concurrent LOAD PART utilizes more efficient method to insert DPSI keys
  - Elimination of BUILD2 phase for Online REORG of a partition
  - Reduce data sharing overhead


© Copyright IBM Corp. 2005. All rights reserved.

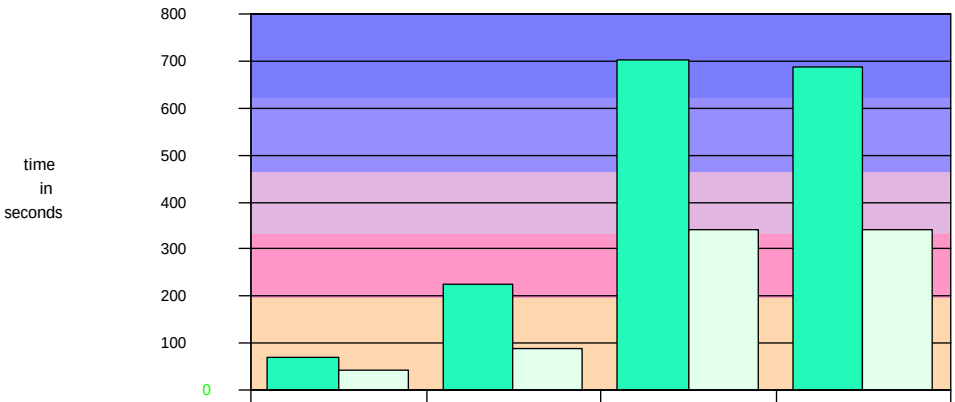
# LOAD a partition with 1 NPI vs. 1 DPSI
(table of 20M rows, 10 partitions)




| | CPU | Elapsed time |
|---|---|---|
| Table with 1 NPI | 62 | 76 |
| Table with 1 DPSI | 28 | 25 |

Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

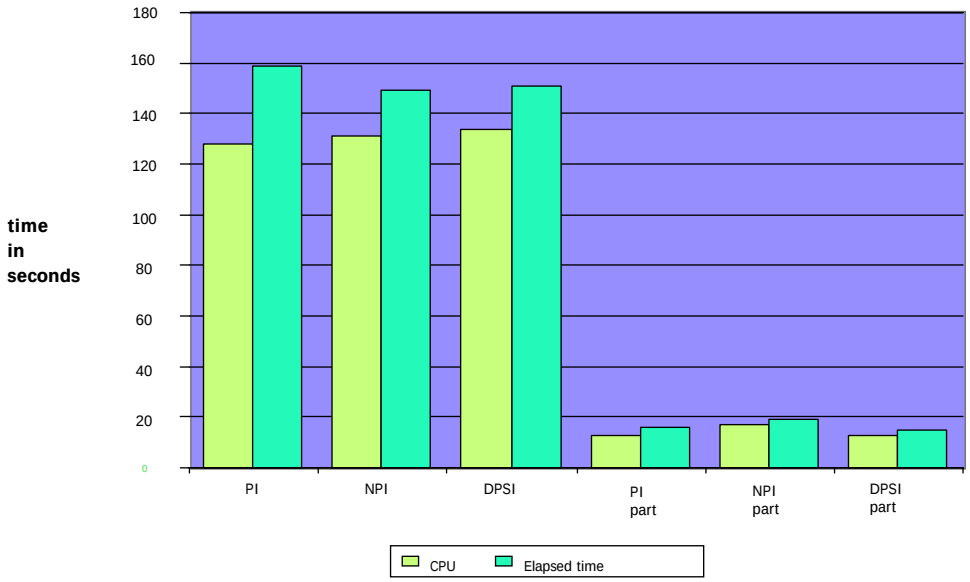
# LOAD table of 20M rows
(10 partitions with 5 NPI or 5 DPSI)




| | 5 DPSI single part | 5 NPI single part | 5 DPSI whole table | 5 NPI whole table |
|---|---|---|---|---|
| CPU | 69 | 225 | 703 | 686 |
| Elapsed time | 42 | 87 | 340 | 342 |

Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

## RUNSTATS Index a partition of PI, NPI and DPSI





© Copyright IBM Corp. 2005. All rights reserved.

## NPI vs. DPSI results - SELECT COUNT query

- **Measurements for SELECT COUNT(*) FROM CVR**
  - Reduction in CPU time with DPSI
  - Similar elapsed time reduction because the query is CPU bound

| | NPI | DPSI | % difference |
|---|---|---|---|
| **access path** | index only 2285 pages | index only 2302 pages | |
| **CPU (sec.)** | 0.759703 | 0.587190 | -23 |
| **Elapsed (sec.)** | 0.825141 | 0.668409 | -19 |

- **Enable parallelism for SELECT COUNT(*) FROM CVR query**
  - Elapsed time and CPU time for DPSI using parallelism are better than for NPI

| | NPI | DPSI | % difference |
|---|---|---|---|
| **access path** | index only 2312 pages | index only 2302 pages | |
| **CPU (sec.)** | 0.718432 | 0.657405 | -9 |
| **Elapsed (sec.)** | 0.637515 | 0.231345 | -64 |


© Copyright IBM Corp. 2005. All rights reserved.

# DPSI overhead when probing all partitions

- **The following query was executed:**
  - SELECT COUNT(*) FROM EVE,PLC WHERE EVE_DPSI >= PLC_DPSI AND PLC_DPSI = constant
- **Observations on measurements shown in the table**
  - DPSI overhead due to DB2 having to probe each DPSI partition in order to evaluate predicates
  - Access path remains the same in both test cases

| | NPI | DPSI | % difference |
|---|---|---|---|
| **Access path** | PLC - index only<br>EVE - index only | PLC - index only<br>EVE - index only | |
| **CPU (sec.)** | 0.021699 | 0.083861 | +281% |
| **Elapsed (sec.)** | 0.037188 | 0.305458 | +721% |
| **Index getpages** | 8 | 17,000 | |


© Copyright IBM Corp. 2005. All rights reserved.

# DPSI - Conclusions

- **Utility processing benefits from the use of DPSIs**
  - Use for partition level operations
  - Less contention
  - No BUILD2 phase for Online REORG of a partition
  - Less data sharing overhead
- **Query processing may be negatively impacted**
  - Queries using DPSIs need predicates on partitioning columns
    - Optimizer can then eliminate non qualifying partitions from the query
  - Don't use DPSI for queries that have to scan all the physical partitions
  - IX only queries show comparable or better performance, much better with query parallelism
  - When using host variables, DB2 determines qualifying partitions at execution time
    - REOPT(ALWAYS) or REOPT(VARS) not required


© Copyright IBM Corp. 2005. All rights reserved.

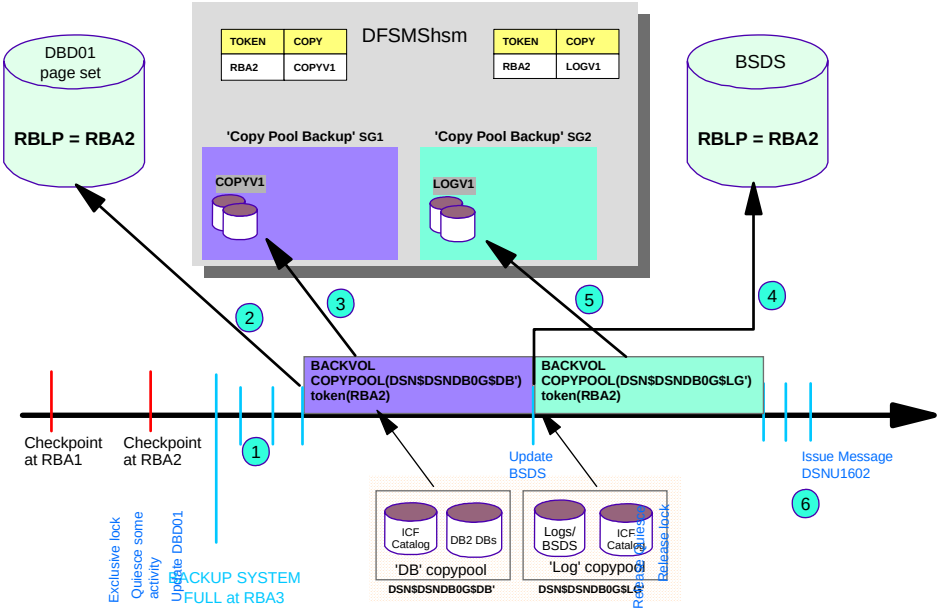
# System level point-in-time recovery

**New utilities and system functions for**

- **Backing up the system**
  - BACKUP SYSTEM
    - Less disruptive than -SET LOG SUSPEND
    - Not a consistent backup (like -SET LOG SUSPEND)
- **Restoring the system**
  - RESTORE SYSTEM
    - Restores consistency (no outstanding URs when done)
- **Where system is an entire**
  - DB2 subsystem
  - DB2 data sharing group
- **Support for DB2 Tracker site**


© Copyright IBM Corp. 2005. All rights reserved.

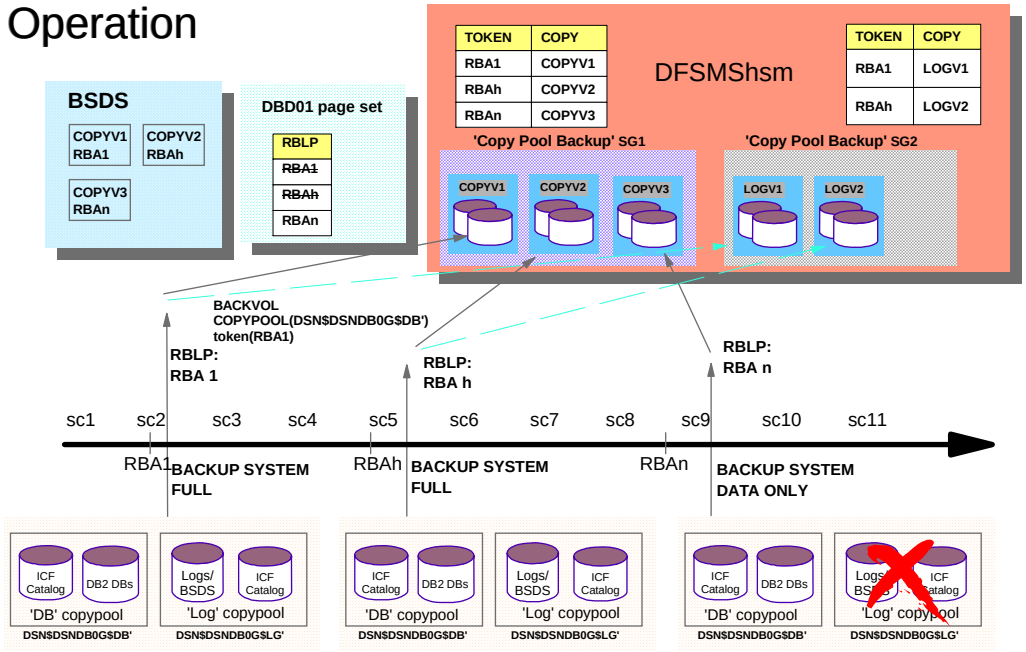
# BACKUP SYSTEM





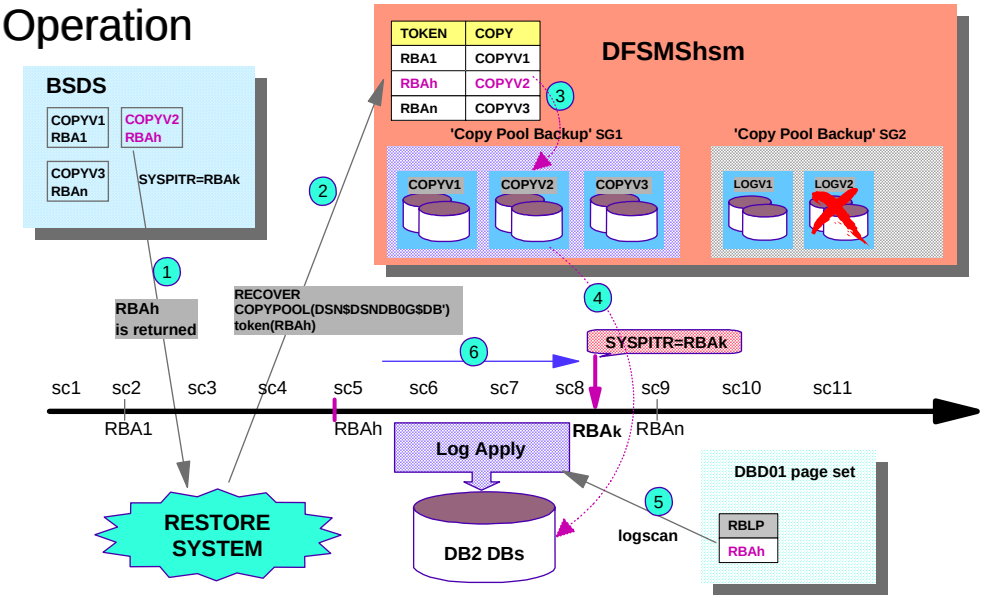
© Copyright IBM Corp. 2005. All rights reserved.

# BACKUP SYSTEM Operation




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

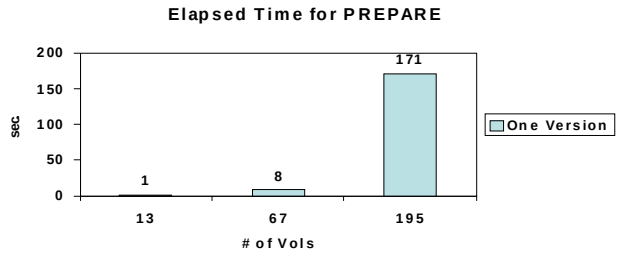
# RESTORE SYSTEM Operation

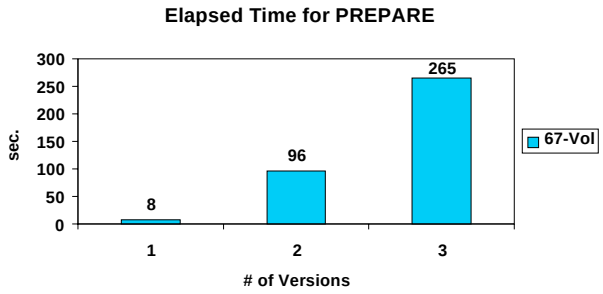



Note that only the database COPYPOOL is restored

Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

# FRBACKUP PREPARE measurements



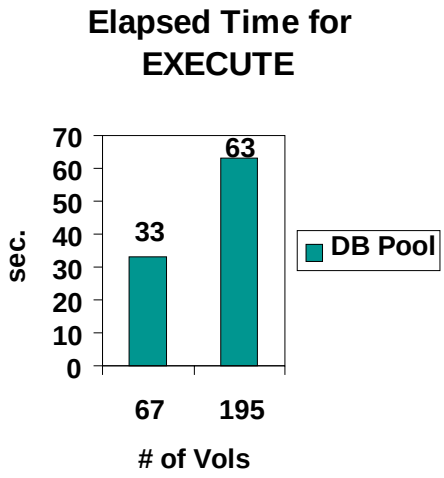




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

# BACKUP SYSTEM measurements

BACKUP SYSTEM DATA ONLY

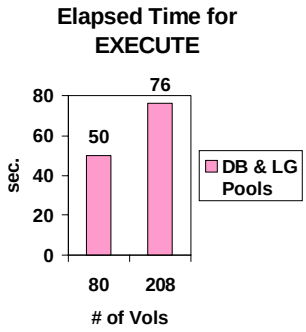
- Invokes FRBACKUP COPYPOOL(DSN$location-name$DB EXECUTE




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

# BACKUP SYSTEM measurements

- **BACKUP SYSTEM FULL**
  - **Invokes 'FRBACKUP COPYPOOL(DSN$location-name$DB) EXECUTE'**
  - **Then invokes 'FRBACKUP COPYPOOL(DSN$location-name$LG) EXECUTE'**

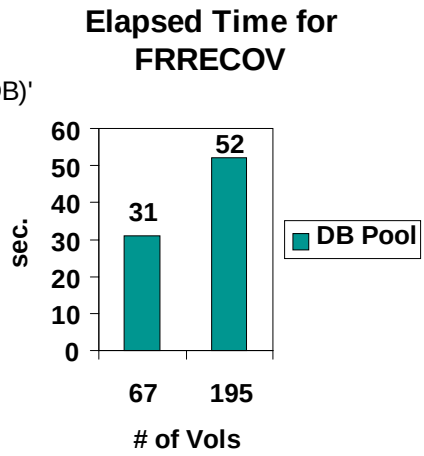




© Copyright IBM Corp. 2005. All rights reserved.

# RESTORE SYSTEM measurements

- RESTORE SYSTEM
  - Invokes 'FRRECOV COPYPOOL(DSN$location-name$DB)'
  - Followed by LOGAPPLY phase
- RESTORE SYSTEM LOGONLY
  - Enters (FAST) LOGAPPLY phase directly





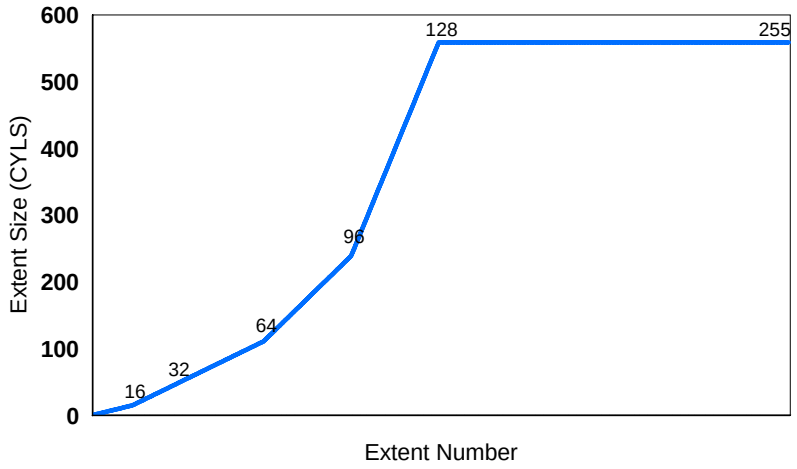
© Copyright IBM Corp. 2005. All rights reserved.

## Automated space management

- **New sliding secondary quantity for DB2 managed page sets**
- **Controlled by zparm MGEXTSZ (YES or NO - default is NO)**
- **Secondary quantity not specified at CREATE time**
- **Adjusts secondary value so max allowable size of data set is reached before max number of extents (255) is reached**
- **Initial secondary allocations are small, and later secondary allocations are larger**
- **Reduces need to alter secondary quantity size**
- **Sliding quantity automatically calculated for each data set**


© Copyright IBM Corp. 2005. All rights reserved.

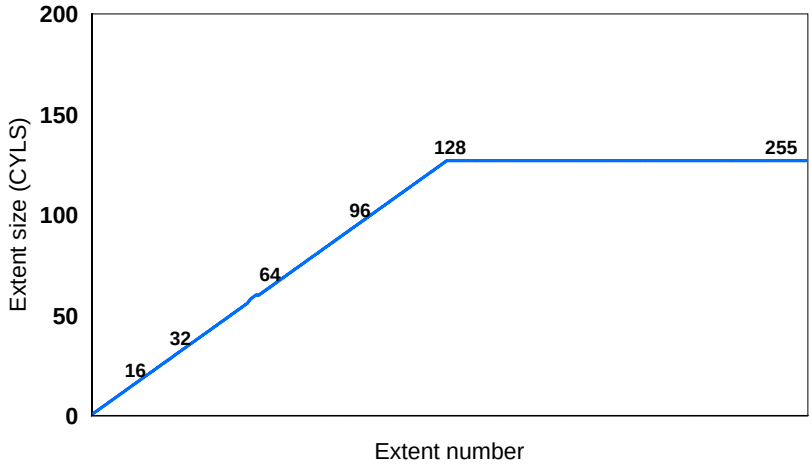
## Sliding scale for 64 GB data set





© Copyright IBM Corp. 2005. All rights reserved.

## Sliding scale for 16 GB data set





© Copyright IBM Corp. 2005. All rights reserved.

## Volatile tables

- **DB2 V6 introduced parameter NPGTHRSH to favor index access for small tables**
  - Did not account for tables whose statistics were very volatile (frequently changing)
  - Also operates at a subsystem level
- **DB2 V8 introduces VOLATILE keyword on table DDL**
  - Specified on CREATE and ALTER
  - Operates at the table level
  - Operator favors index access for tables specified as VOLATILE
  - Beneficial for cluster tables used in ERP applications


© Copyright IBM Corp. 2005. All rights reserved.

IBM

# Part 5 - Utilities

Redbooks Workshop

IBM ITSO - International Technical Support Organization

DB2 Information Management Software

© Copyright IBM Corp. 2005. All rights reserved.

## What's new in DB2 Version 8

- Delimited data support for LOAD and UNLOAD
- New defaults for better "out of the box" performance
- SORTKEYS for LOAD/REORG/REBUILD
- SORTDATA for REORG
- SORTDATA now allowed for 32 KB records with DFSORT
- REORG will use implicit clustering index
- REORG SHRLEVEL CHANGE allow DISCARD
- REORG SHRLEVEL NONE/REFERENCE allow REBALANCE
- REORG SHRLEVEL REFERENCE catalog tables with links
- Online Schema Support (e.g., REPAIR VERSIONS)
- Improved usability (SCOPE PENDING)
- Non-uniform statistics and on non-indexed columns
- HISTORY statistics without updating main statistics
- New utilities BACKUP SYSTEM and RESTORE SYSTEM

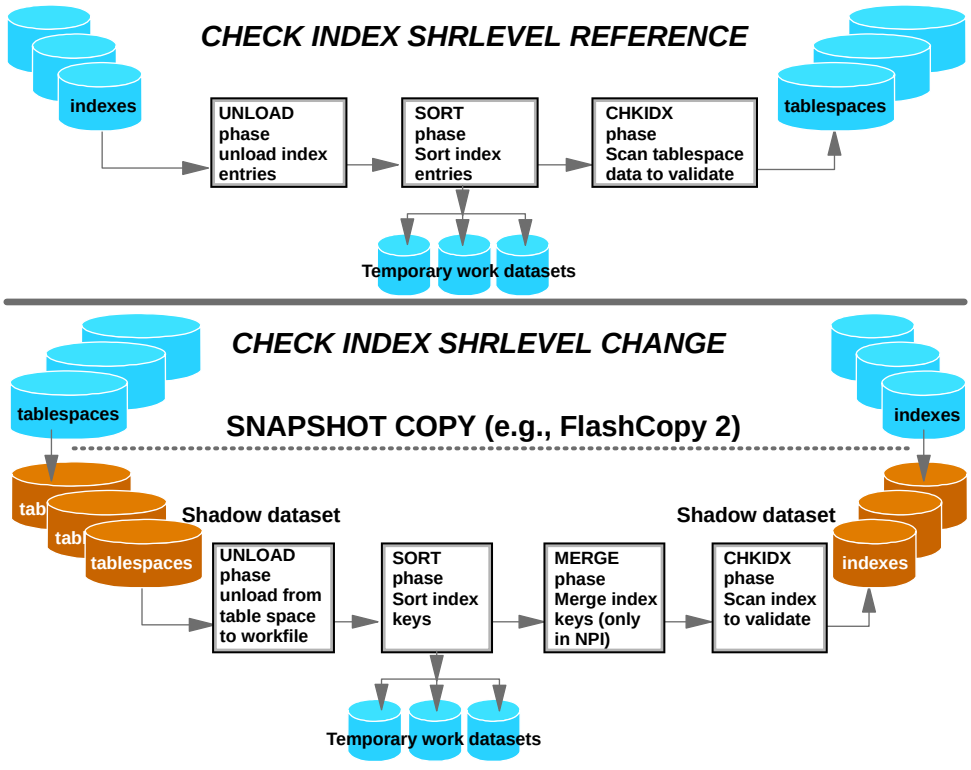


© Copyright IBM Corp. 2005. All rights reserved.

# CHECK INDEX SHRLEVEL CHANGE

- CHECK INDEX is an online utility checking that an index is consistent with its data.
- Useful after a conditional restart or point-in-time recovery. It is also recommended before CHECK DATA, for RI, especially if you specify DELETE YES.
- CHECK INDEX has now two options:
  - read-only behavior as SHRLEVEL REFERENCE
  - and the new SHRLEVEL CHANGE
- New capability, added by APARs PQ92749 (UK04683) and PQ96956 to V8.
- Meant to provide longer read-write access and reduce the read-only time of applications during CHECK INDEX.
- In addition CHECK INDEX provides a parallel processing structure similar to REBUILD INDEX.


© Copyright IBM Corp. 2005. All rights reserved.




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

# SHRLEVEL REFERENCE and CHANGE

**REFERENCE**

1. UTILINIT (drain and flush buffers)
2. UNLOAD (IX entries)
3. SORT (IX entries)
4. CHKIDX (SCAN data)
5. UTILTERM(release writers)

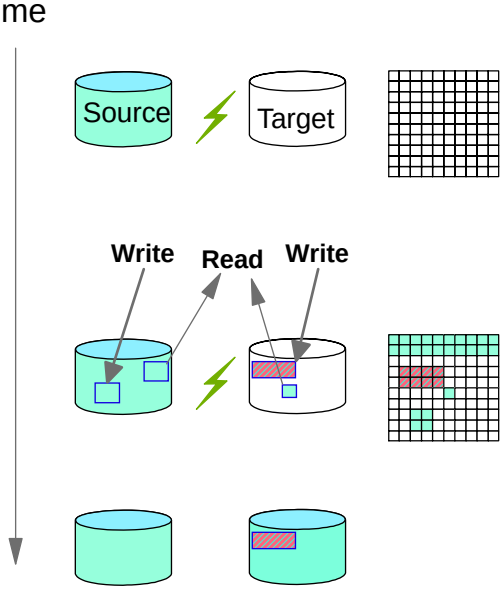
**CHANGE**

1. UTILINIT (drain writers + create shadows + release writers)
2. UNLOAD (data)
3. SORT (IX Keys)
4. CHKIDX (IX Keys)
5. MERGE (if NPI)
6. UTILTERM


© Copyright IBM Corp. 2005. All rights reserved.

# FlashCopy in action

Time




**FlashCopy V2 requested**

**1- FlashCopy relationship is established**

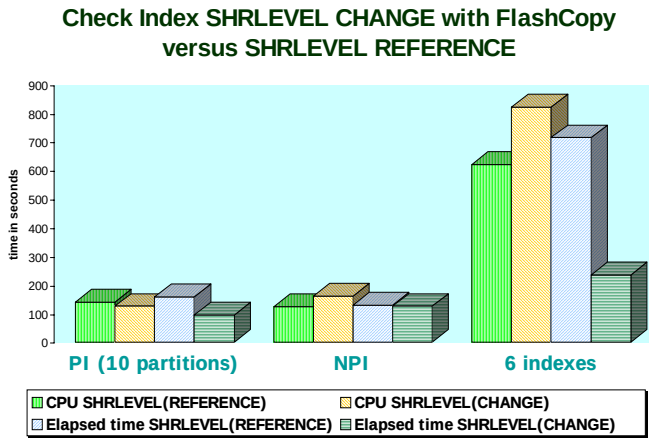
**Both source and target volumes immediately available**

**2- Read and write to both source and target volumes possible**

**3- When copy is complete, relationship between source and target ends**

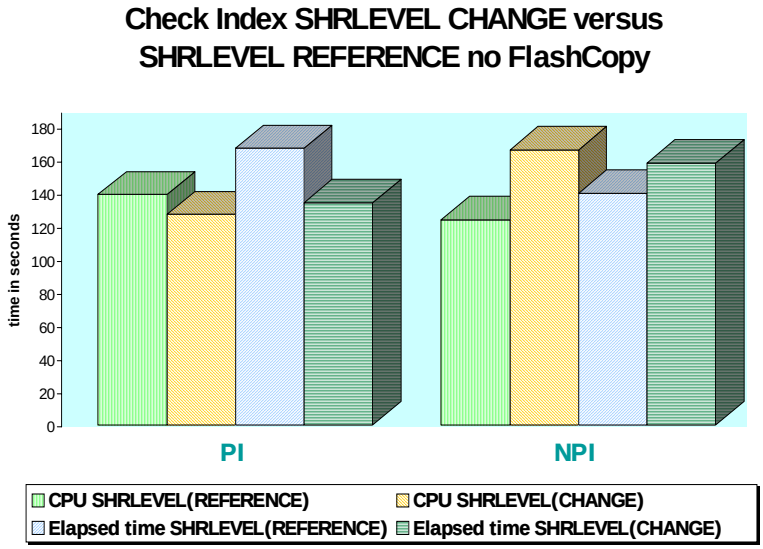

© Copyright IBM Corp. 2005. All rights reserved.

# Using FlashCopy V2




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

# Without FlashCopy



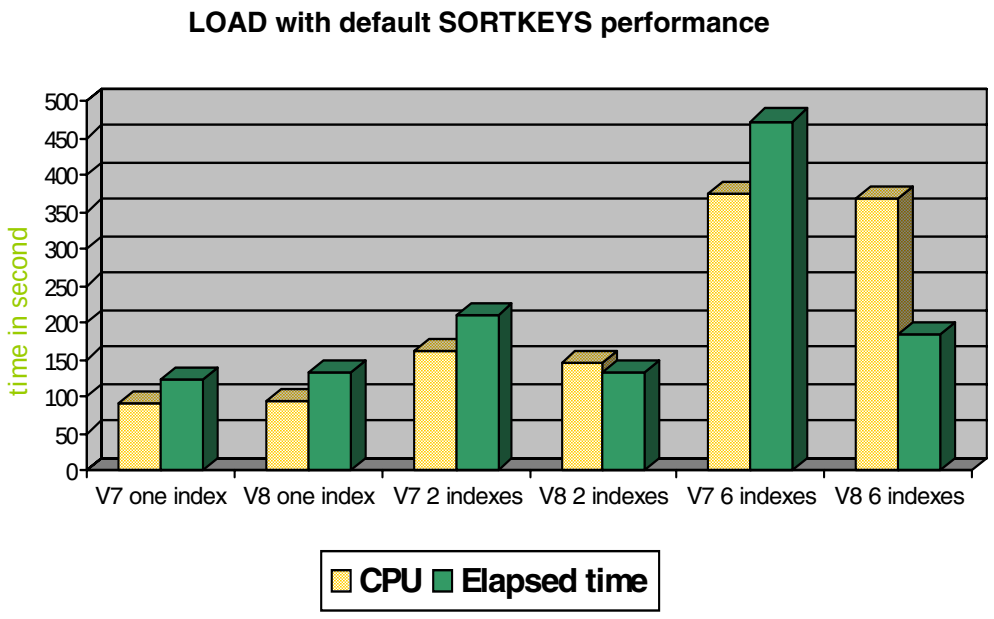

Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

## LOAD and REORG

- **SORTKEYS for LOAD and REORG, and SORTDATA for REORG are keywords which generally contribute to better performance.**
- **In V7 SORTKEYS and SORTDATA were not the default for SHRLEVEL REFERENCE and NONE, so you had to specify explicitly these parameters to activate the methods providing better performance.**
- **In V8, SORTKEYS in the LOAD utility is the default and SORTDATA and SORTKEYS are default in the REORG utility.**
- **The SORT phase is enforced in LOAD and REORG in V8.**


© Copyright IBM Corp. 2005. All rights reserved.

## LOAD - NEW DEFAULT SORTKEYS





© Copyright IBM Corp. 2005. All rights reserved.

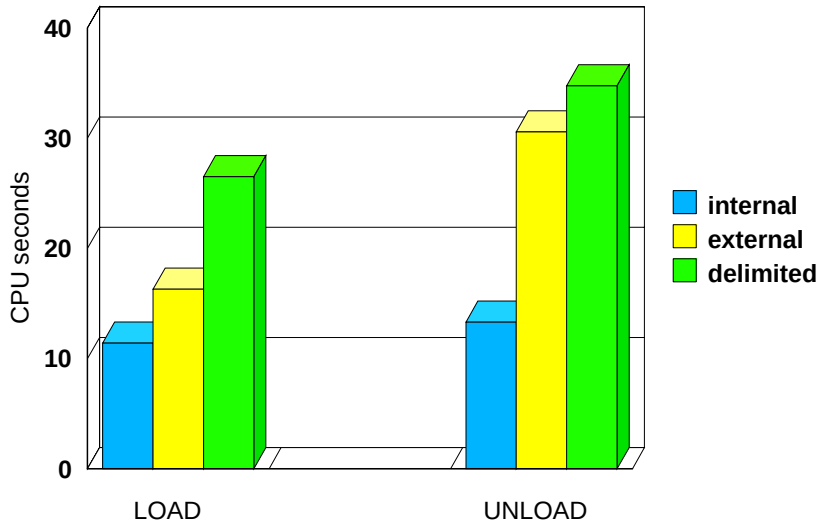
## LOAD and UNLOAD delimited

LOAD DATA INTO TABLE tablename
FORMAT DELIMITED

- Data and delimiters can be un/loaded in UNICODE, EBCDIC or ASCII
- Can load spread sheet data previously saved in comma separated value (CSV) format
- Can easily load varchar


© Copyright IBM Corp. 2005. All rights reserved.

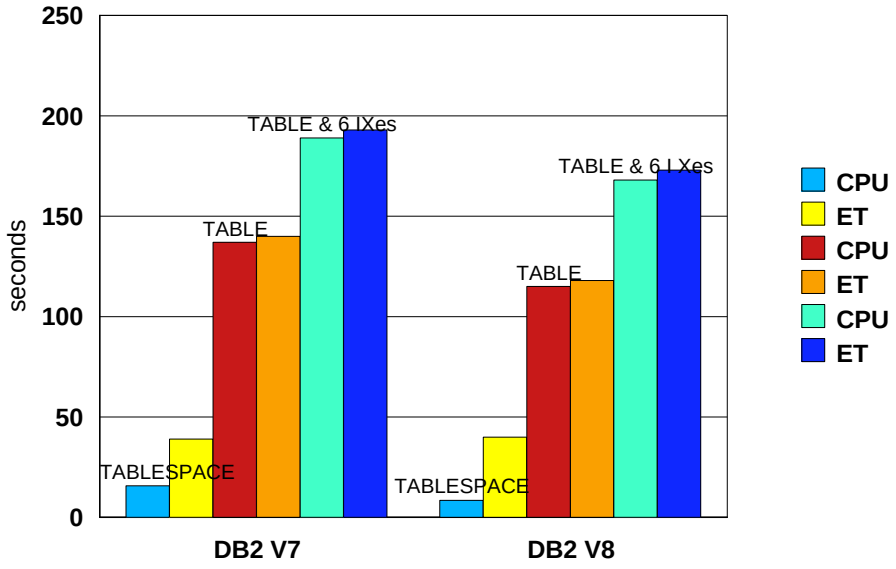
## LOAD and UNLOAD delimited





© Copyright IBM Corp. 2005. All rights reserved.

## RUNSTATS improvements




Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

## New RUNSTATS option - DSTATS

- **Collect distribution statistics on any column, or group(s) of columns, indexed or non-indexed, specified at the table level.**
- **Frequency distributions for non-indexed columns or groups of columns**
- **Cardinality values for groups of non-indexed columns**
- **LEAST frequently occurring values, along with MOST frequently occurring values, for both index and non-indexed column distributions (DB2 V7 only gathers the most frequently occurring value.)**

Redbooks Workshop
IBM ITSO - International Technical Support Organization
ibm.com/redbooks
© Copyright IBM Corp. 2005. All rights reserved.

## DSTATS

- **New keywords COLGROUP, LEAST, MOST and BOTH enable the collection of distribution statistics on any table column and on a group of columns**
- **SORTDEVT and SORTNUM keywords for using dynamic allocation of the sort work data sets**
- **ST01WKnn and STATWK01 in DD statements**
- **APARs PQ88375, PQ87509 and PQ90884**


© Copyright IBM Corp. 2005. All rights reserved.

## Cross Loader - 1

- **Function: single LOAD job to transfer data from one location to another location or from one table to another table at the same location**
- **Local server or any DRDA-compliant remote server as a data input source**
- **Performance APAR: PQ84160**
- **PTFs are available for:**
  - **V7**: UQ91416
  - **V8**: UQ91422


© Copyright IBM Corp. 2005. All rights reserved.

## Cross Loader - 2

- **APAR - PQ90263: Improved LOB handling for V7 and V8**
- **The processing of LOB columns is changed to bypass the architectural limit of 32767 bytes**
- **LOAD with INCURSOR stores only 8 bytes per LOB column within the 32767 byte buffer.**
- **CrossLoading tables with LOB columns is limited by memory available above the 16MB line.**
- **DSNU1178I message is changed, increasing the region size should result in successful execution of the LOAD utility.**
- **Must restart DB2 for the fix to become active**

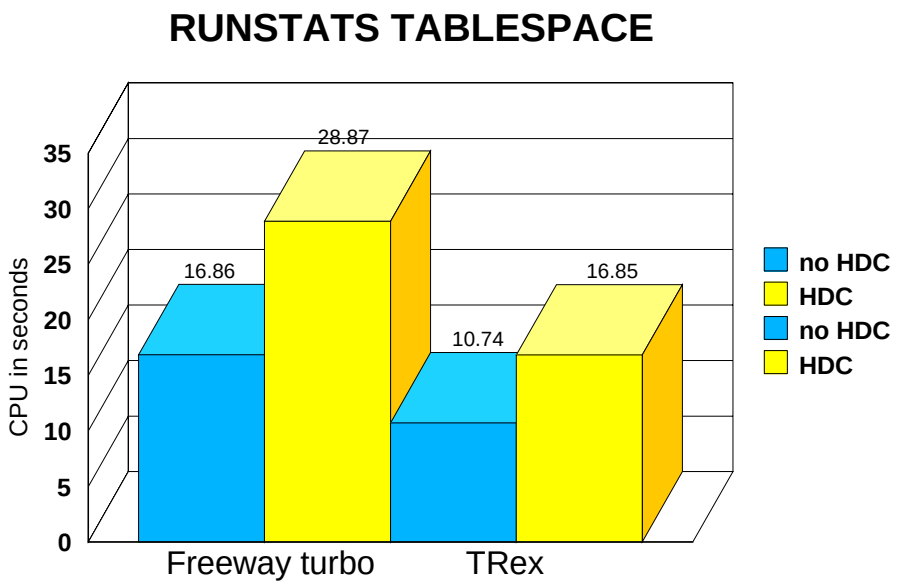

© Copyright IBM Corp. 2005. All rights reserved.

## Hardware compression

- **Hardware data compression (HDC) on the IBM z900 (Freeway) has been enhanced to provide significant performance gains over IBM 9672 G6 processors. Based on our DB2 V7 set of measurements for both SQL and Utility, we observed up to 5.5 times improvement for compression and up to 6 times for expansion. For both expansion and compression, the longer the record length, the higher the performance gain. With expansion, records with lower compression ratios have higher performance gains.**
- **Similar improvements have been provided to the IBM z800 (Raptor).**
- **HDC has been measured in line with processors speed for z900 turbo models.**
- **HDC on the IBM z990 (TRex) has shown, over IBM z900 (Freeway) Turbo processors, 35 to 45% improvement for compression and 80 to 100% improvement for expansion, in line with the processor speedup, with some improvement measured on expansion.**


© Copyright IBM Corp. 2005. All rights reserved.

## Runstats and compression





© Copyright IBM Corp. 2005. All rights reserved.

## Maintenance additions

- **See the informational APAR II14047 for useful information on the use of DFSORT by DB2 utilities.**
- **PK04076 (UK03983): SORT is bypassed during LOAD SHRLEVEL NONE when input is sorted and only one index is defined.**
- **PK01510: RUNSTATS CLUSTERRATIO for NPI**
- **PK03469: RUNSTATS with multiple COLGROUPS**
- **LOB support**


© Copyright IBM Corp. 2005. All rights reserved.

IBM

# Part 6 - Networking and e-business

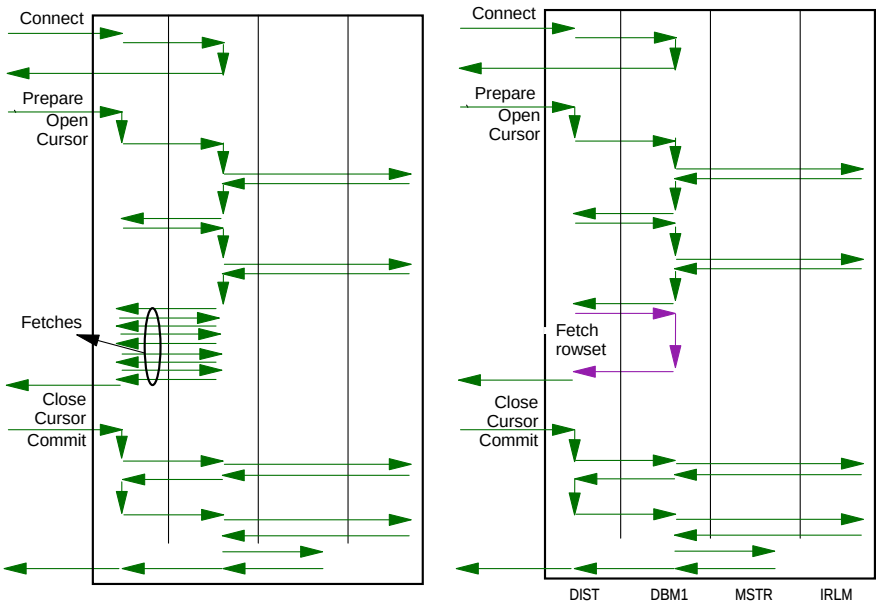
Redbooks Workshop

IBM ITSO - International Technical Support Organization

DB2 Information Management Software

© Copyright IBM Corp. 2005. All rights reserved.

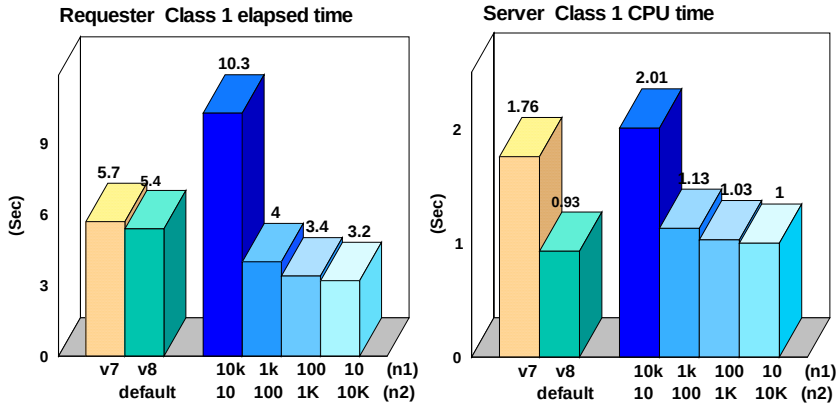
## FETCH distributed DB2 V7 vs. DB2 V8






© Copyright IBM Corp. 2005. All rights reserved.

# FETCH - Distributed - Host to host

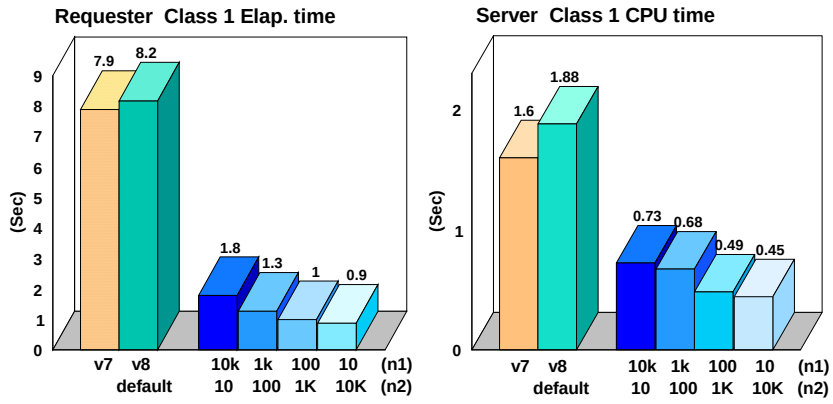



V8 default = V8 without explicit multi-row FETCH
V8 n1 x n2 = V8 with explicit multi-row FETCH
n1= # of FETCH
n2= # of rows per MR FETCH


© Copyright IBM Corp. 2005. All rights reserved.

# INSERT - Distributed - Host to host

**Total rows inserted : 100,000 rows**




V8 default = V8 without explicit multi-row Insert
V8 n1 x n2 = V8 with explicit multi-row Insert,
n1= # of Insert SQL calls
n2= # of rows per MR Insert SQL call


© Copyright IBM Corp. 2005. All rights reserved.

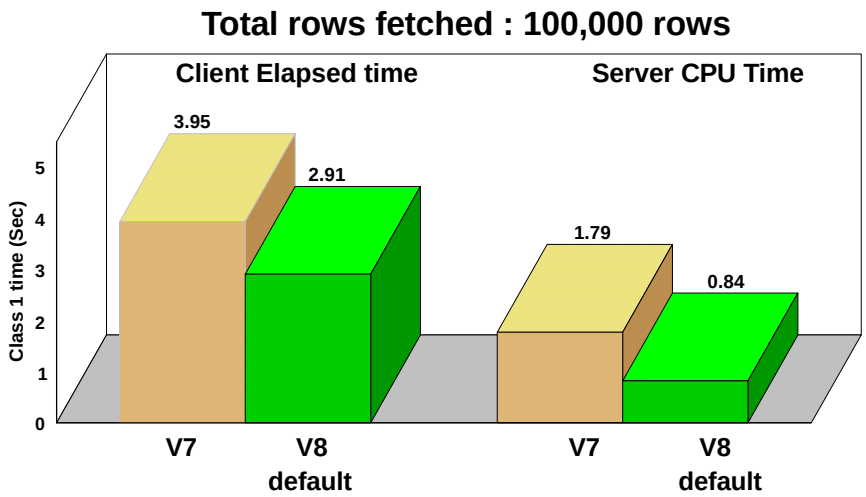
# DB2 Connect evaluation DB2 V7 vs. DB2 V8 - DRDA larger query block

| | Elapsed time (sec.) | Transfer rate (MB/sec.) |
|---|---|---|
| DB2 V7 | 7.111 | 4.28 |
| DB2 V8 | 6.176 | 4.93 |

The retrieval of large result sets from DB2 V8 is 15.2% faster than from DB2 V7 when using the new 64 KB query block size - RQRIOBLK.


© Copyright IBM Corp. 2005. All rights reserved.

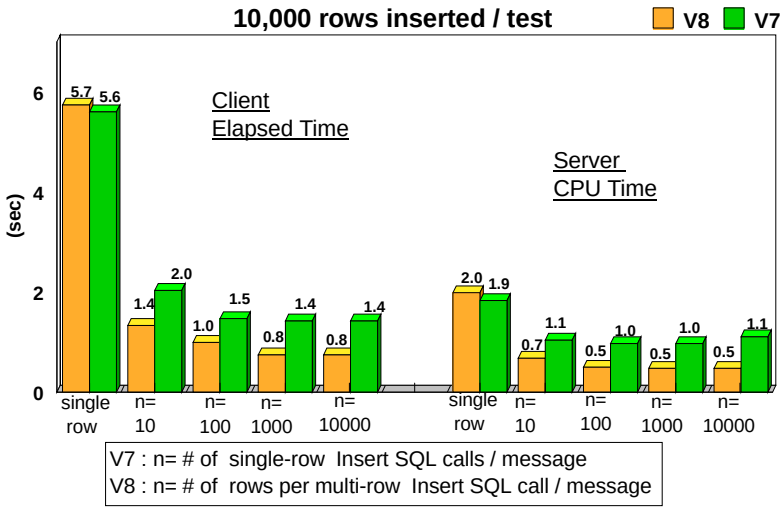
# FETCH performance - DB2 client to host





© Copyright IBM Corp. 2005. All rights reserved.

# INSERT performance - DB2 client to host

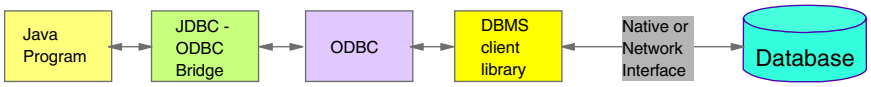




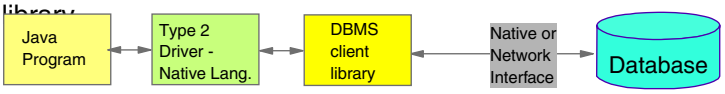
© Copyright IBM Corp. 2005. All rights reserved.

# Summary of JDBC Driver Types

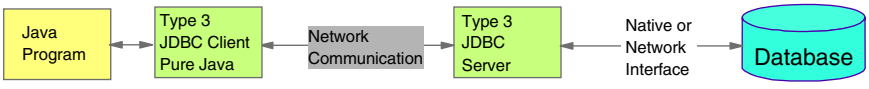
**Type 1**: JDBC requests are delegated to the ODBC client library



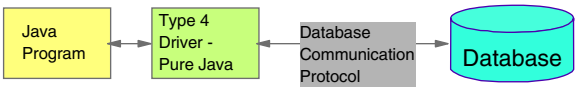

**Type 2** (native API): JDBC functionality in Java, built on top of the DB2 client library




**Type 3** (native communication protocol): JDBC requests are delegated to a remote JDBC server - 100% Java client library




**Type 4** (database protocol): 100 % pure Java client library





© Copyright IBM Corp. 2005. All rights reserved.

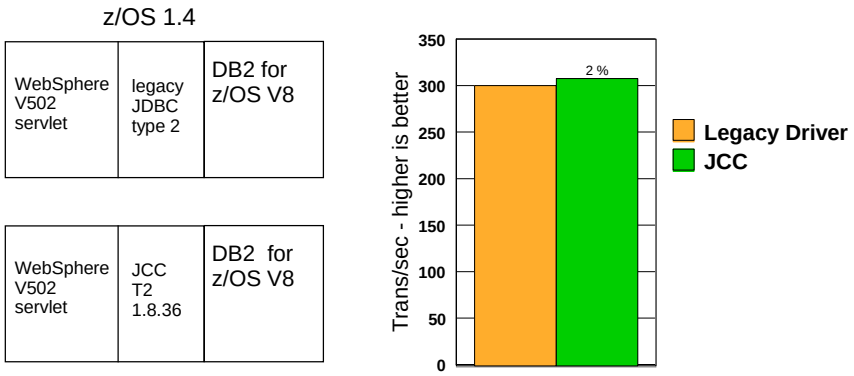
# DB2 Universal Java Driver (JCC)

- **Type 2 and Type 4 Driver**
  - T2 needs Gateway or DB2 Client to use DRDA to a remote DB2
  - T4 talks directly DRDA to a remote DB2
- **Single code base across the DB2 family**
  - Portability and compatibility
  - Development and service
- **Java API enhanced for SQLJ/JDBC 3.0 standard compliance**
- **Improved authentication and error handling**
- **Improved monitoring and tracing**


© Copyright IBM Corp. 2005. All rights reserved.

# WebSphere and DB2 for z/OS V8 Servlet workload - 1

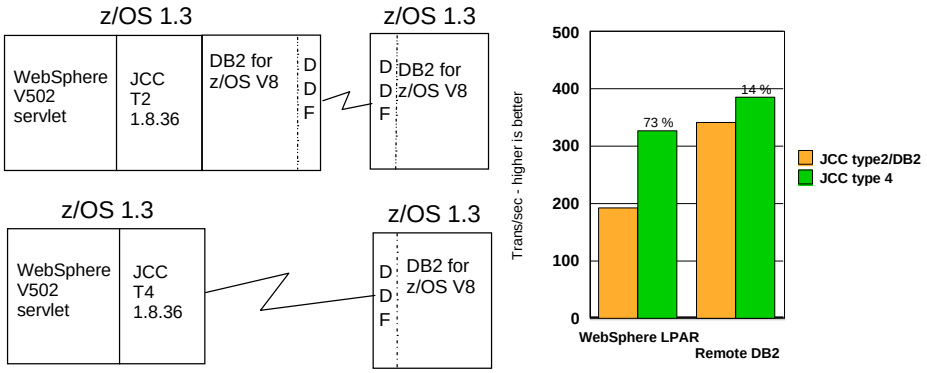
## Legacy JDBC 2.0 driver vs. JCC type 2





© Copyright IBM Corp. 2005. All rights reserved.

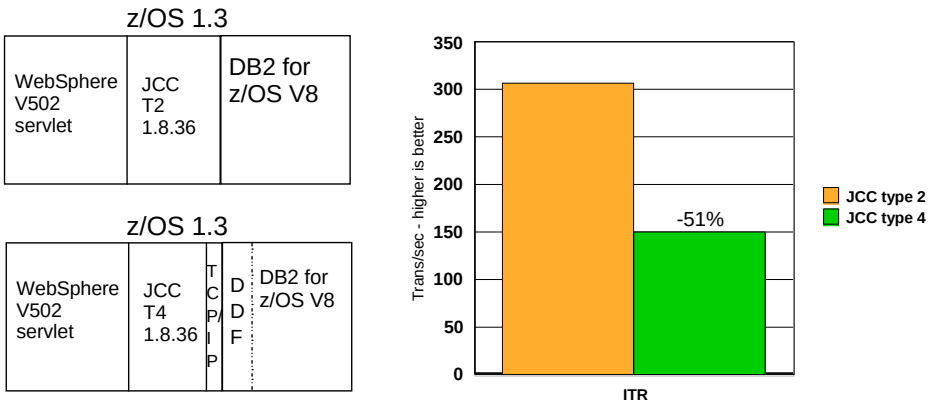
# JCC type 2 vs. JCC type 4 - Remote environment





© Copyright IBM Corp. 2005. All rights reserved.

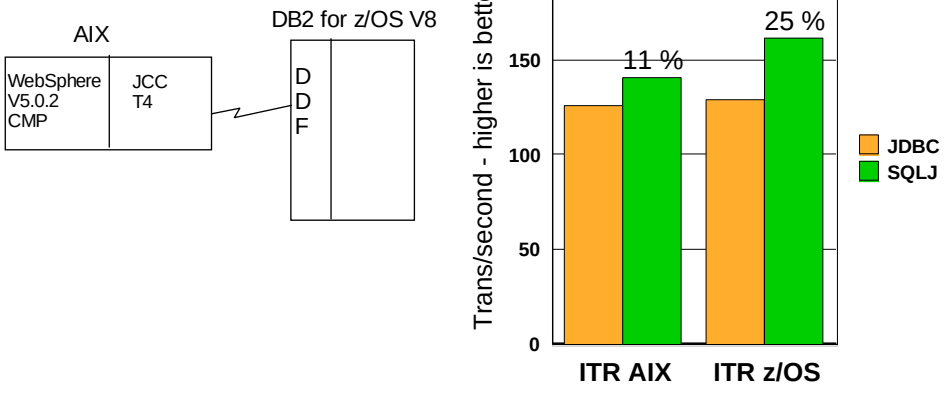
# JCC type 2 vs. JCC type 4 - Local environment





© Copyright IBM Corp. 2005. All rights reserved.

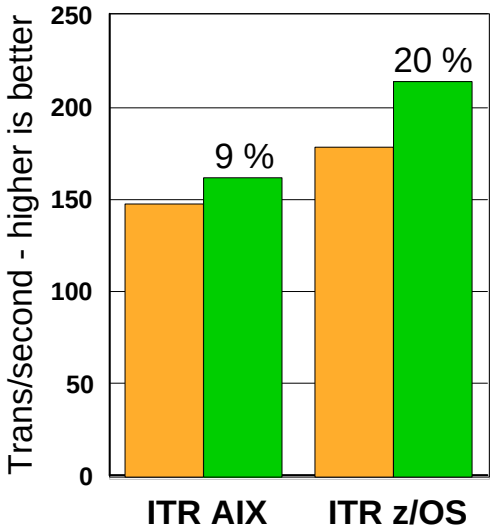
# JCC type 4 FP4 - JDBC vs. SQLJ





© Copyright IBM Corp. 2005. All rights reserved.

# KEEPDYNAMIC YES




SpecJApp2002
WebSphere/AIX V502
DB2 for z/OS V8
ITR = tps/cpu%*100


© Copyright IBM Corp. 2005. All rights reserved.

# Part 7 - Data sharing


DB2 Information Management Software

© Copyright IBM Corp. 2005. All rights reserved.

## CF request batching

- **CF request batching allows DB2 V8 to reduce traffic to and from the CF for both writes to and reads from the group buffer pool for castout processing**
- **Reduces data sharing overhead for most workloads.**
- **Most beneficial for large numbers of updates to GBP-dependent objects (e.g. batch workloads)**
- **Potential to reduce CPU used by DBM1 address space**
  - Fewer messages passed to and from the CF
- **CF utilization may be slightly higher as message service times increase**
  - CF has to perform more work to service each batched request
- **CF link utilization may decrease slightly due to fewer messages passed**



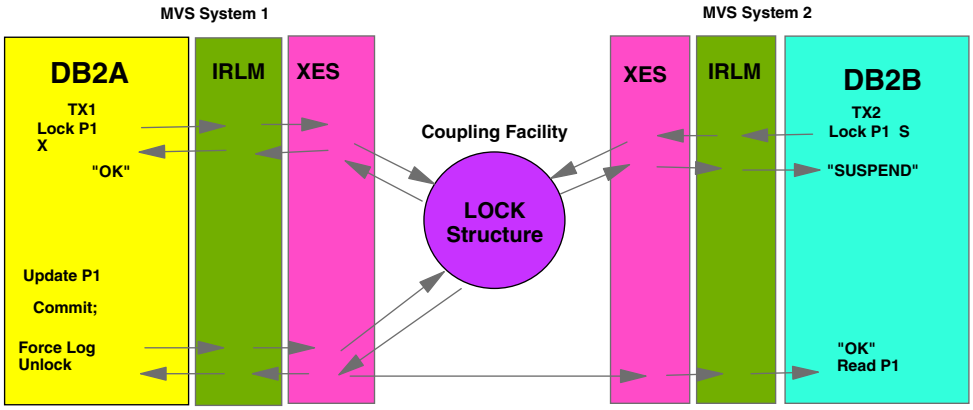
© Copyright IBM Corp. 2005. All rights reserved.

# Locking protocol - Data sharing review

- **Data sharing use two types of locks:**
  - Physical locks (P-locks)
    - Page set P-locks: tracks inter-DB2 read-write interest of a page set
    - Page P-locks: ensures physical consistency of a page across members
    - P-locks are global (apply to the group)
  - Logical locks (L-locks)
    - Serializes access to data to ensure data consistency
    - L-locks can be local (apply to a member) or global (apply to the group)
- **Explicit hierarchical locking**
  - Locks organized in a parent/child relationship
    - Child locks for page and row locks
    - Parent locks for table space and partition locks
  - Only the most restrictive lock needs to be propagated


© Copyright IBM Corp. 2005. All rights reserved.

# Lock contention




- When "no contention", global lock granted synchronously for execution of the transaction
- No need to "suspend" the transaction task (measured in microseconds)
- "Lock contention" is detected quickly


© Copyright IBM Corp. 2005. All rights reserved.

## Lock contention before V8

- **Lock information held in three different components:**
  - IRLM
  - XES
  - Lock structure in the CF
- **z/OS cross-system extended services (XES)**
  - Uses a negotiation process to control access
  - Child locks propagated to XES and CF based on inter-DB2 interest on parent lock
- **Three types of global contention**
  - False contention - Lock table hashing algorithm provides same hash value for two different resources; they share one lock table entry
  - XES contention - XES detects a contention because of incompatible lock modes for same resource; IRLM may find locks compatible
  - Real contention - Caused by normal IRLM lock incompatibility between two members


© Copyright IBM Corp. 2005. All rights reserved.

## Locking protocol level 2

- **New data sharing locking protocol**
  - Designed to avoid the cost of **global contention** processing whenever possible
- **Prior to V8 it is common for parent L-locks to cause XES contention**
- **V8 remaps parent IX L-locks from XES-X to XES-S locks**
- **XES can grant IX parent L- locks locally when only IS or IX L- locks are held on the object**


© Copyright IBM Corp. 2005. All rights reserved.

## Locking protocol level 2

- **Reduces global contention for table space L-locks**
  - Reduced XES-level contention across members
  - Improved data sharing performance, especially for OLTP
  - RELEASE(DEALLOCATE) may not be needed
  - LOCKPART YES is 'forced'
- **New protocol takes effect after first group-wide shutdown and restart in New Function Mode (NFM)**
  - Do not have to delete lock structure from CF to build new lock structure
- **New mapping takes effect after the restart of the first member, after successful quiesce of all members in the group**
  - Group-wide outage required to enable this feature


© Copyright IBM Corp. 2005. All rights reserved.

## Locking protocol level 2 - Benchmarks

- **Benchmarks have been run against a two member data sharing group to evaluate the impact of the new locking protocol.**
- **V7 test uses default of LOCKPART NO.**
- **V8 behaves as if table spaces are defined with LOCKPART YES.**
- **If the workload does not require all partitions to be GBP dependent, V8 would need to propagate more locks to the CF than the same workload running in V7.**
- **For these tests, all partitions were GBP dependent**
  - Impact of LOCKPART(NO) in V7 compared with V8 should be minimal


© Copyright IBM Corp. 2005. All rights reserved.

## Protocol level 2 - Statistics/Accounting Report extract

| Suspends per commit | DB2 V7 | | DB2 V8 | |
|---|---|---|---|---|
| IRLM Global Contention | 0.01 | 0.01 | 0.01 | 0.01 |
| XES Contention | 0.54 | 0.52 | 0.00 | 0.00 |
| False Contention | 0.13 | 0.11 | 0.00 | 0.00 |

| Class 3 suspend time (msec / commit) | DB2 V7 | | DB2 V8 | |
|---|---|---|---|---|
| Lock / Latch (DB2 + IRLM) | 6.219 | 5.708 | 0.080 | 0.109 |
| Global Contention | 8.442 | 7.878 | 0.180 | 0.197 |


© Copyright IBM Corp. 2005. All rights reserved.

## Protocol level 2 - RMF XCF Report extract

| | DB2 V7 | | DB2 V8 | |
|---|---|---|---|---|
| Req In (req / sec) | 2,300 | 2.200 | 17 | 18 |
| Req Out (req / sec) | 2,300 | 2,200 | 17 | 18 |
| CHPID Busy (%) | 97.43 | 86.53 | 22.27 | 12.74 |

## Protocol level 2 - RMF CF Activity Report extract

| Lock Structure Activity | DB2 V7 | DB2 V8 |
|---|---|---|
| Total Requests (/ sec) | 8,418.33 | 13,541.67 |
| Deferred Requests (/ sec) | 790.00 | 6.92 |
| Contention (/ sec) | 790.00 | 6.89 |
| False Contention (/ sec) | 115.00 | 2.58 |
| CF Utilization (%) | 5.4 | 6.9 |


© Copyright IBM Corp. 2005. All rights reserved.

## Protocol level 2 - CPU consumption

| | DB2 V7 | | DB2 V8 | | Delta (V8 / V7) |
|---|---|---|---|---|---|
| **IRLM (msec / commit)** | 0.330 | 0.317 | 0.012 | 0.015 | -96% |
| **Total DB2 CPU time (msec / commit)** | 3.447 | 3.708 | 3.109 | 3.357 | -10% |
| **Group ITR** | 823.56 | | 897.00 | | +9% |

## Protocol level 2 - RELEASE(DEALLOCATE) vs. RELEASE(COMMIT)

| | DB2 V7 | DB2 V8 |
|---|---|---|
| **Transaction CPU time** | -18% | -7% |
| **Global ITR** | +18% | +5% |


© Copyright IBM Corp. 2005. All rights reserved.

## Locking protocol level 2 - Conclusions - 1

- **Avoids cost of global contention processing whenever possible**
  - No need to wait for global lock contention processing to determine that a new parent IX or IS L-lock is compatible with existing parent IX or IS L-locks
- **New protocol dramatically reduces both XES and false contention**
  - Results in shorter lock and global contention suspension times
- **Less IRLM SRB time with an improved overall transaction throughput**
  - More synchronous XES requests to CF and more lock structure traffic with slightly higher CF utilization, though should not impact overall performance


© Copyright IBM Corp. 2005. All rights reserved.

## Locking protocol level 2 - Conclusions - 2

- **Reduction in performance gap between RELEASE(COMMIT) and RELEASE(DEALLOCATE) for data sharing workloads**
- **Change to LOCKPART parameter behavior may result in additional locks on individual partitions even though LOCKPART NO is specified**
- **We recommend you schedule a group wide restart after you enter NFM**
- **We recommend you reevaluate use of RELEASE(DEALLOCATE)**


© Copyright IBM Corp. 2005. All rights reserved.

## Change to IMMEDWRITE default BIND option

- **IMMEDWRITE(YES) solves application issue at expense of performance**
  - Forces each individual changed page immediately to the CF, ahead of commit
  - Each immediate write also writes to DB2 log, for as many changes as are made to same page
- **IMMEDWRITE(PH1) solves the issue for most customers**
  - Much less expensive that IMMEDWRITE(YES)
- **Little or no performance difference between NO and PH1**
- **V8 changes default from NO to PH1**
- **Options YES, NO and PH1 remain available**
  - NO now behaves like PH1


© Copyright IBM Corp. 2005. All rights reserved.

## IMMEDWRITE BIND change - Conclusions

- **IMMEDWRITE YES remains unchanged**
  - Changed pages written to GBP as soon as buffer updates completed
- **CPU cost of writing changed pages to GBP is now charged to the allied TCB**
  - Prior to DB2 V8, this was true only for PH1 and YES
  - For NO, CPU cost was charged to the MSTR SRB
- **Provides a more accurate accounting for all DB2 workloads**
  - DB2 is able to charge more work back to user who initiated the work
  - Customers who use allied TCB time for end user charge back may see additional CPU cost with this change
- **Immediately available when you migrate to DB2 V8**
  - You do not have to wait until new-function mode


© Copyright IBM Corp. 2005. All rights reserved.

## DB2 I/O CI limit removed

- **Prior to V8, DB2 did not schedule a single I/O for a page set with CIs that span a range of more than 180 CIs**
  - Restriction in place to avoid long I/O response times
- **With newer DASD technology, especially Parallel Access Volume (PAV) support, restriction can now be lifted**
- **V8 removes the CIs per I/O limit for list prefetch and castout I/O requests**
  - Limit still remains for deferred write requests
    - To avoid potential performance degradation during transaction processing
    - We don't want an online transaction to have to wait for a potentially longer deferred write I/O request to complete


© Copyright IBM Corp. 2005. All rights reserved.

## DB2 I/O CI limit removed

| Activity | With I/O CI Limit | | I/O CI Limit Removed | |
|---|---|---|---|---|
| **List Prefetch Requests** | 42,876 | 42,783 | 43,186 | 43,187 |
| **List Prefetch Reads** | 209,600 | 208,200 | 40,894 | 40,711 |
| **List Prefetch Pages Read / List Prefetch Reads** | 3.58 | 3.56 | 18.44 | 18.39 |
| **Pages Written per Write I/O** | 6.20 | 4.37 | 27.11 | 26.49 |

Statistics for the two DS members

## Data sharing CPU

| | With I/O CI Limit | | I/O CI Limit Removed | | Delta (Removed / With) |
|---|---|---|---|---|---|
| **DBM1 (msec / commit)** | 0.518 | 0.590 | 0.456 | 0.457 | -18% |


© Copyright IBM Corp. 2005. All rights reserved.

## DB2 I/O CI limit removed - Conclusions

- **Removal of DB2 CI I/O limits for list prefetch and castout I/O has potential to significantly reduce DBM1 CPU**
- **Data sharing environments can benefit through more efficient castout processing**
  - Prior to V8, castout processed a few pages on each request
  - Castout can now process more pages with each request and not be restricted to only pages within range of 180 CIs for the table space
- **CPU gains will vary significantly**
  - Dependent on application processing, access paths used and data organization and placement
  - Applications with list prefetch with a large distance between pages will benefit the most
  - OLTP environments with excessive unlock castout and delete namelist requests will also benefit


© Copyright IBM Corp. 2005. All rights reserved.

# Part 8 - Installation and migration



© Copyright IBM Corp. 2005. All rights reserved.

## Contents

- **DB2 catalog changes**
- **IVP sample programs:**
  - DSNTEP4
  - DSNTIAUL
- **DB2 Catalog Migration V7 - V8 CM**
- **Migration V8 CM to V8 NFM**


© Copyright IBM Corp. 2005. All rights reserved.

## New catalog table spaces

- **SYSEBCDC**
  - EBCDIC table space
  - Not used until NFM
  - SYSIBM.SYSDUMMY1 moved from SYSSTR to SYSEBCDC
- **SYSALTER**
  - New catalog table SYSIBM.SYSOBDS
  - To store version 0 information of ALTERed tables


© Copyright IBM Corp. 2005. All rights reserved.

## Buffer pool and group buffer pool changes

- **ALTER BUFFERPOOL command defaults have changed:**
  - DWQT - deferred write threshold is decreased from 50% to 30%
  - VDWQT - vertical deferred write threshold is decreased from 10% to 5%
- **ALTER GROUPBUFFERPOOL command defaults have changed:**
  - CLASST - class castout threshold is decreased from 10% to 5%
  - GBPOOLT - cast out threshold is decreased from 50% to 30%.
  - GBPCHKPT - time interval between group buffer pool checkpoints is lowered from 8 minutes to 4 minutes.


© Copyright IBM Corp. 2005. All rights reserved.

# IVP sample programs

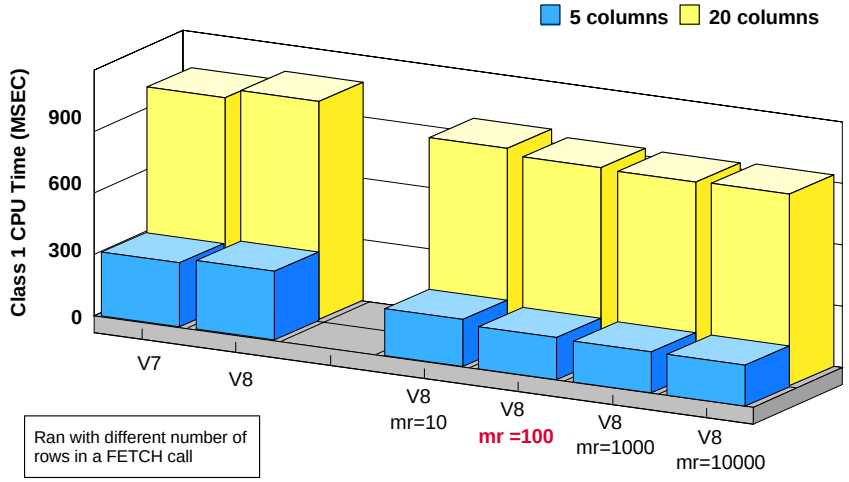
- **DSNTEP2**
  - GET DIAGNOSTICS
  - Larger SQL statement
  - Greater than 18 character table/column names
  - MAXERRORS to dynamically set the number of errors that DSNTEP2 will tolerate
  - SYSPRINT blocking changed
    - will speed up the rate in which DSNTEP2 outputs its results
- **Multi-row support for:**
  - DSNTEP4 - new sample program
  - DSNTIAUL


© Copyright IBM Corp. 2005. All rights reserved.

# Multi-row FETCH support of DSNTEP4

**10,000 rows fetched / test**

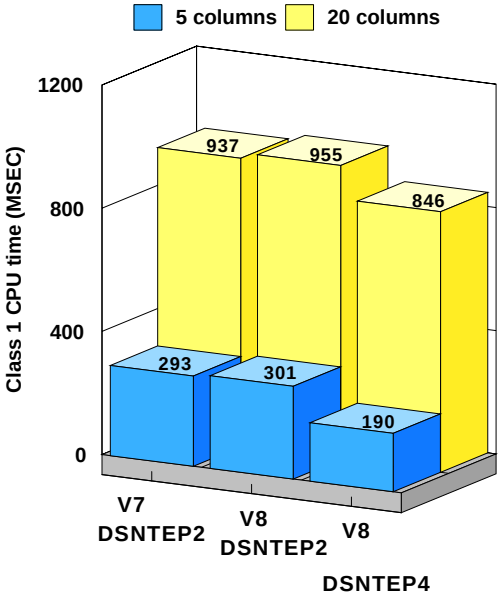
V8 w/o MR : +5% to +7%
V8 with MR : -5% to -35%




Ran with different number of rows in a FETCH call


© Copyright IBM Corp. 2005. All rights reserved.

# DSNTEP2 and DSNTEP4 - 10,000 rows fetched/test



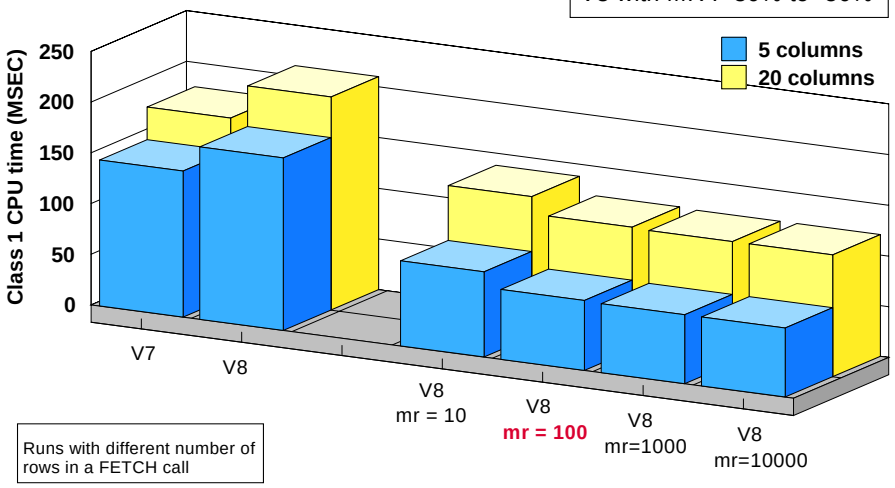

- **V8 DSNTEP4 :**
  - automatic exploitation of MR Fetch
  - default MR=100 rows
  - 10% to 35% CPU time reduction vs. V7 DSNTEP2
- **V8 DSNTEP2 :**
  - w/o MR exploitation
  - within 3% CPU time regression vs. V7 DSNTEP2


© Copyright IBM Corp. 2005. All rights reserved.

# Multi-row FETCH support of DSNTIAUL

**10,000 rows fetched / test**

V8 w/o MR : +20%
V8 with MR : -30% to -50%

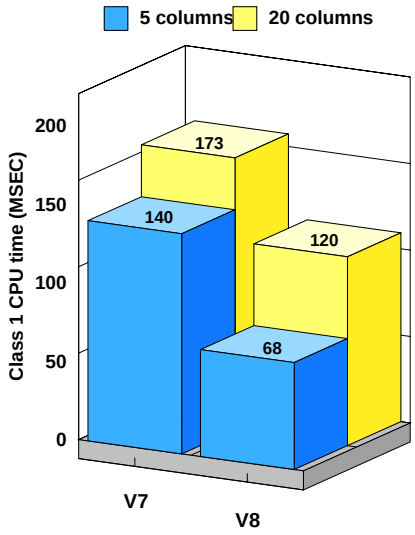



Runs with different number of rows in a FETCH call


© Copyright IBM Corp. 2005. All rights reserved.

# DSNTIAUL comparing V7 with V8

## 10,000 rows fetched / test




- V8 DSNTIAUL
  - automatic exploitation of Multi-row Fetch
  - default MR=100 rows
  - 30% to 50% CPU time reduction compared to V7


© Copyright IBM Corp. 2005. All rights reserved.

# Installation changes

- Cache dynamic SQL now enabled by default
- Fast log apply enabled by default
- Checkpoint frequency increased from 50,000 to 500,000
- Archive log block size reduced from 28,672 to 24,676
- Removed hiperpool definitions
- DDF panels have new terminology
  - "Inactive DBAT" instead of "Type 1 inactive thread"
  - "Inactive Connection" instead of "Type 2 inactive thread"
- DSNHDECP module must be defined by user
- Buffer pool BP8K0, BP16K0 and BP32K must be defined
- Data sharing - group buffer pool GBP8K0, GBP16K0 and GBP32K must be allocated
- Location name must be specified, even if DDF is not used
  - See APAR PQ91009/UQ90701
- Only WLM established stored procedures can be defined

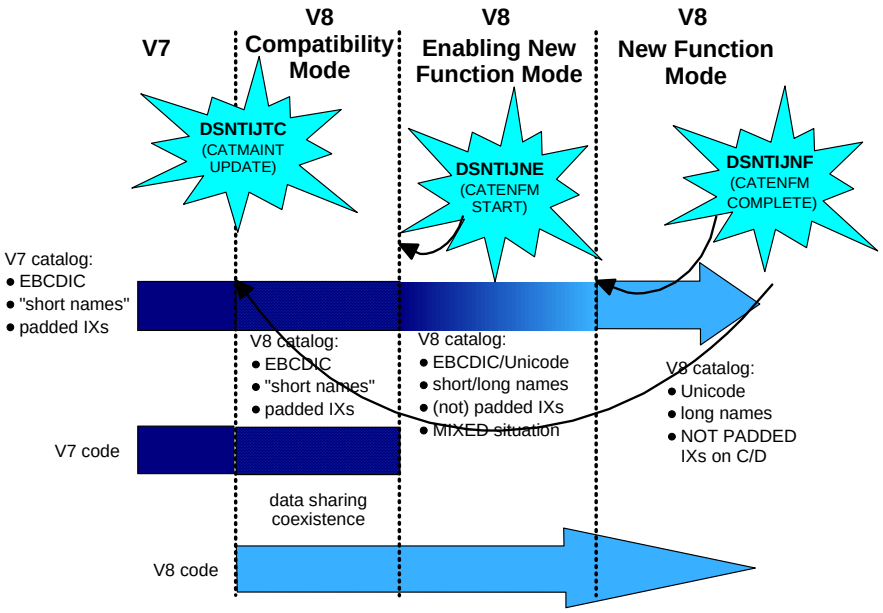

© Copyright IBM Corp. 2005. All rights reserved.

# Typical migration process today

- **Process built up over years of DB2 migration experience**
  - 1. Test install and migration/fallback on test system
  - 2. Rollout across strategic development environments
    - Verify compatibility of old functions on new release
  - 3. Finally migrate production to the new release
  - 4. Begin to use new functions
    - When satisfied with new release
- **Use no new function during migration time frame**


© Copyright IBM Corp. 2005. All rights reserved.

# Overview of V8 migration process





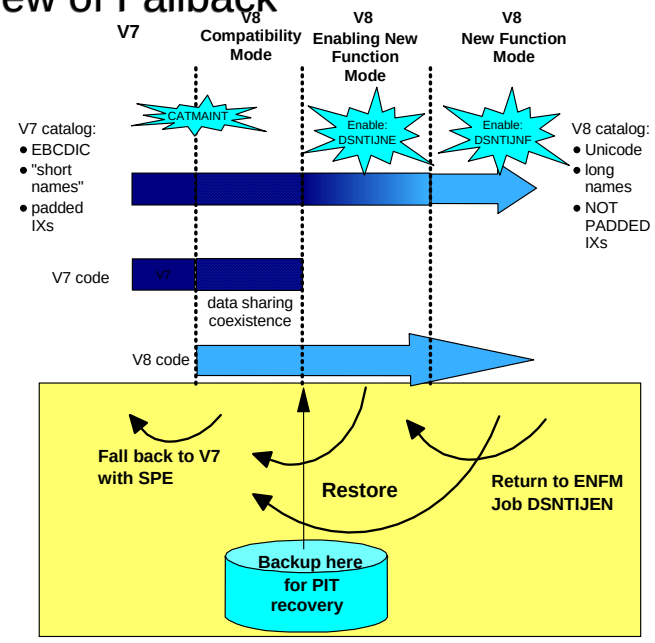
© Copyright IBM Corp. 2005. All rights reserved.

# Why a "New" migration process?

- **Significant enhancements to the product - largest release ever!**
  - Support for long names
  - Unicode catalog tables
- **More control over the migration process**
  - In line with system availability / maintenance window
  - Timing the introduction and use of the new functions
- **A tighter, more robust migration process**
  - Fewer migration and fallback errors
  - Forced SPE Installation
- **Converting of the DB2 catalog to Unicode**


© Copyright IBM Corp. 2005. All rights reserved.

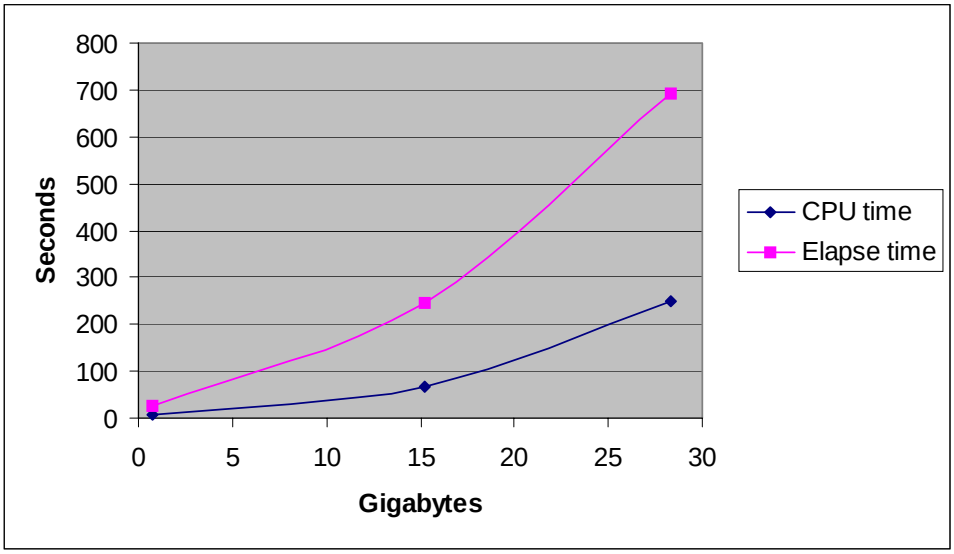
# Overview of Fallback





© Copyright IBM Corp. 2005. All rights reserved.

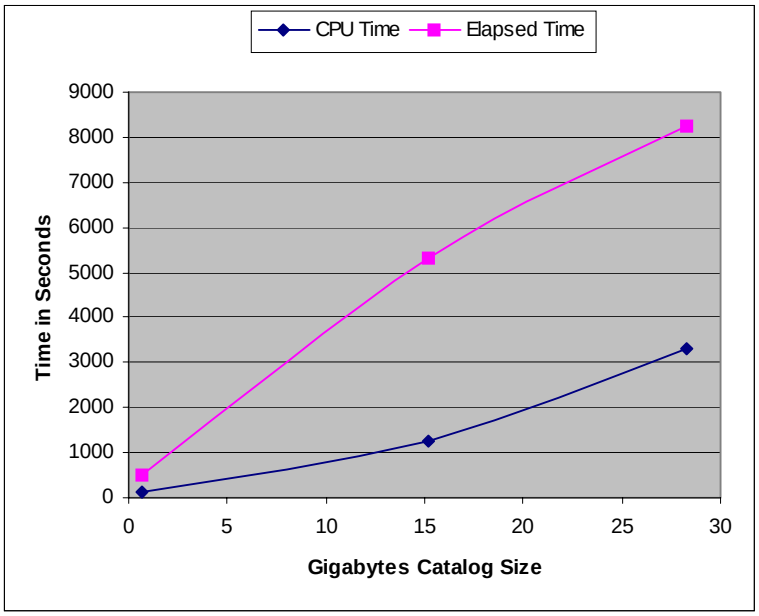
# DB2 catalog migration V7 - V8 CM





© Copyright IBM Corp. 2005. All rights reserved.

# DB2 catalog migration in V8 ENFM





© Copyright IBM Corp. 2005. All rights reserved.

## Bibliography: DB2 for z/OS V8 Redbooks

- DB2 UDB for z/OS Version 8 Performance Topics, SG24-6465
- Business Value of DB2 UDB for z/OS, SG24-6763
- DB2 for z/OS Stored Procedures: Through the CALL and Beyond, SG24-7083
- DB2 for z/OS and WebSphere: The Perfect Couple, SG24-6319
- DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, ... and More, SG24-6079
- Disaster Recovery with DB2 UDB for z/OS, SG24-6370
- DB2 Performance Expert for z/OS Version 2, SG24-6867-01
- z/OS Multilevel Security and DB2 Row-level Security Revealed, SG24-6480
- DB2 UDB for z/OS Version 8 Technical Preview, SG24-6871
- DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There, SG24-7088

## Bibliography: DB2 for z/OS V7 Performance Redbooks

- DB2 UDB Version 7 Selected Performance Topics, SG24-6894
- DB2 UDB Version 7 Performance Topics, SG24-6129


© Copyright IBM Corp. 2005. All rights reserved.

## DB2 for z/OS on the Web
## ibm.com/software/db2zos


- Full URL for DB2 UDB for z/OS home page
  http://www.ibm.com/software/data/db2/os390/index.html
- DB2 for z/OS Version 8 related material
  http://www.ibm.com/software/data/db2/zos/v8books.html
- Direct URL for Support page
  http://www.ibm.com/software/data/db2/os390/support.html
- Presentations page:
  http://www.ibm.com/software/data/db2/os390/presentations.html
- Migrate to DB2 for z/OS and OS/390 Version 7
  ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z14.pdf
- Migrate to DB2 for z/OS Version 8
  http://www.ibm.com/software/data/db2/os390/techdocs2/z01.pdf
- DB2 Developer Domain articles
  http://www7b.boulder.ibm.com/dmdd/


© Copyright IBM Corp. 2005. All rights reserved.