


00101 01011000  
00000 01000101  
00010 01001001

**IDUG** North America

01000101 01011000 01010000 01000101 01010010  
01000100 01010101 01000111 00100001 00100000  
01001110 01000011 01000101 00100000 01001001

Experience IDUG

**Session: A12**  
**DB2 for z/OS Selected Performance Topics**  
Roger Miller  
*IBM*

 **IDUG**  
The Worldwide DB2 User Community

May 15, 2009 • 08:00 a.m. – 09:00 a.m.  
Platform: DB2 for z/OS

This talk will present some of the latest challenges and achievements for DB2 performance, noting some recent measurements and what can be achieved with the latest software and the latest hardware. The discussion will be centered on what you can do to help improve performance in your shop, particularly on the items that can help in DB2 9.

## DB2 for z/OS Selected Performance Topics



- Objectives and some recent measurements
- System performance tuning: what can be achieved across the subsystem
- Database design changes that can make significant improvements
- Application tuning: SQL, utilities, and other areas
- Sources and resources

IDUG'2009 North America 2

The first part of this presentation is system performance, looking at the subsystem and overall system, covering the aspects of objectives, metrics, managing, resources, tuning and System z. The second part is application performance, with sections for tuning, SQL, languages, utilities, environment, database design, and application design. We'll finish with pointers to many other sources and resources, providing a detailed roadmap of where to look for more information. We expect this information to be usable by many people with highly varied tasks. Performance professionals and database administrators who are concerned about performance will be able to use most of the content. Application designers, architects and programmers will primarily use the application performance part.

## Objectives

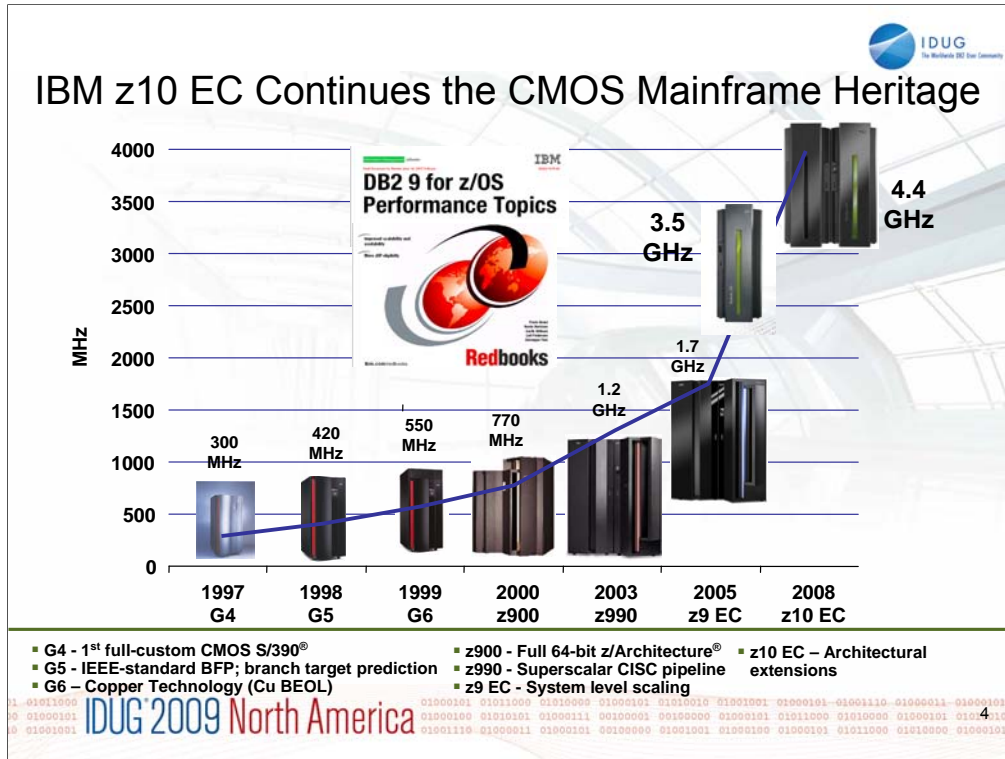
- Response time and elapsed time
- Resources and costs
  - Processing
  - Memory
  - I/O
  - Communication
  - Software
  - People
- Throughput
- System availability
- Data Integrity, System Integrity, Availability, Concurrency, Security, Privacy, and Productivity



Setting performance objectives is difficult. Setting realistic performance and service-level agreements can be almost impossible. The challenge is having the needed resources to meet the objectives. The business wants to have significant cost reductions, but improved service, something for nothing.

One of the primary concepts for performance and for capacity planning is the balanced computer system, with the objectives of avoiding performance problems and allowing the business to change. A balanced system design provides enough of all the computer resources (processor, memory, disk, and communication) to deliver the service without undue bottlenecks or constraints for a single resource. The balanced system optimizes system capacity and achieves the best performance and throughput in accordance with the goals. In one sense, we will be tuning our applications to match the configuration. Capacity planning is estimating and tuning our configuration to match the applications.

The next key concepts are service levels and service management. We want the performance objectives to be realistic, attainable, affordable, meaningful, understandable, and measurable. How you define good performance for your DB2 subsystem depends on your particular data processing needs and their priority.



The design of the IBM System z10™ processor chip is the most extensive redesign in over 10 years, resulting in an increase in frequency from 1.7 GHz (z9 EC) to 4.4 GHz on the z10 EC. The z10 BC processors run at 3.5 GHz.

It is designed for secure data serving, yet also was enhanced to provide improvement enhances for CPU intensive workloads. The result is a platform that continues to improve upon all the mainframe strengths customers expect, yet opens a wider aperture of new applications that can all take advantage of System z10s extreme virtualization capabilities, and lowest TCO versus distributed platforms.

See section 4.3.1 z10 performance in the latest updates of DB2 9 for z/OS Performance Topics, SG24-7473 for additional detail.

## IBM System z10 Benefits for DB2

- **Faster CPUs, more CPUs, more memory**
  - ▶ 50% more n-way performance “on average”
  - ▶ 62% more uniprocessor performance
  - ▶ 70% more server capacity (54->64 CPUs)
  - ▶ Up to 64 CPUs, z/OS 1.9 needed for 64-way in a single LPAR
  - ▶ Up to 1.5 TB, z/OS 1.8 needed for >256G in a single LPAR
- **Infiniband Coupling Facility links**
- **New OSA-Express3, 10 GbE for faster remote apps**
- **HiperDispatch**
- **Hardware Decimal Floating Point facility**
- **1MB page size (DB2 X plans to exploit)**
- **50+ instructions added to improve compiled code efficiency (DB2 X plans to use)**



IDUG'2009 North America 5

The new z10 has faster processors and more processors. DB2 measurements found most workloads having a range of 1.4 to 2.1 times faster with z10 compared to z9.

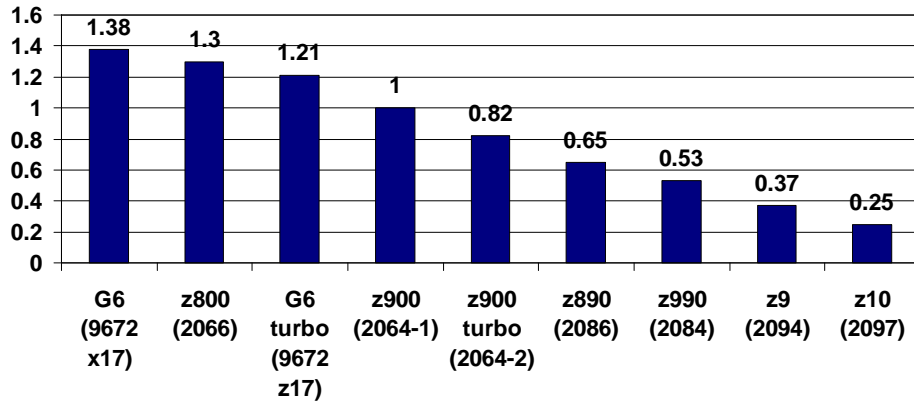
Larger memory: DB2 users can potentially see higher throughput with more memory used for DB2 buffer pools, EDM pools or SORT pools.

Improved IO: improvements in the catalog and allocation can make the large number of data sets much faster and easier to manage. Disk IO times and constraints can be reduced.

Substantial improvements in XML parsing can result from use of the zIIP and zAAP specialty engines. The z10 zIIP and zAAP engines are much faster at no additional cost. The zIIP processors can be used for XRC processing.

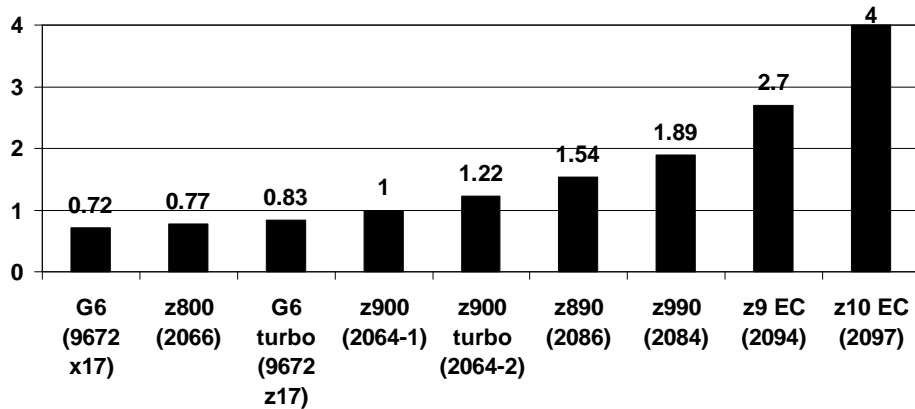
**HiperDispatch:** Only available on z10 EC. Combination of z/OS software and firmware. Minimum z/OS R1.7 + IBM zIIP Web Deliverable support for z/OS V1.7 to enable HiperDispatch. z10 EC Driver level 73G. Single HIPERDISPATCH=YES z/OS IEAOPTxx parameter dynamically activates HiperDispatch

## CPU Time Multiplier for some processor models



These are the improvements in performance from a single uniprocessor, expressed as a CPU multiplier. So if a process takes one second of CPU time on the z900, it runs .37 second on the z9 EC and roughly .25 second on the z10 EC or 4 times faster.

## Recent single-processor relative CPU speeds



These are the improvements in performance expressed as the power of the uniprocessors. The change from 1 to .25 does not seem as impressive as the improvement of 4 times faster processing. Of course, we also gain another factor of improvements from the larger numbers of engines that come with each processor. The System z9 EC processors can have as many as 54 engines, while earlier processors were limited to 12 or 20 engines. The z10 EC can have up to 64 processors, while the z10 BC can have up to 5 standard processors and 5 specialty engines.

## DB2 for z/OS & IBM zIIP value



Portions of DB2 V8 and **DB2 9 (blue)** workloads may benefit from zIIP\*:

**ERP, CRM, Business Intelligence or other enterprise applications**

- Via DRDA over a TCP/IP connection
- **DB2 9 for z/OS Remote native SQL procedures**
- **DB2 9 XML parsing**



**Data warehousing applications\*:** Large parallel SQL queries

**DB2 9 higher percentage of parallel queries eligible for zIIP**

**DB2 Utilities LOAD, REORG & REBUILD** maintaining index structures

IDUG 2009 North America

DB2 9 uses zIIP in two new ways, remote native SQL procedures and increased use of parallelism. See IDUG Europe 2007 presentation F06 by Terry Purcell, "Tuning your SQL to get the most out of zIIPs". This session can be obtained from IDUG online Technical Library by searching for code EU07F06.

The zIIP is designed so that a program can work with z/OS to have all or a portion of its enclave Service Request Block (SRB) work directed to the zIIP. The above types of DB2 V8 work are those executing in enclave SRBs, of which portions can be sent to the zIIP. Not all of this work will be run on zIIP. z/OS will direct the work between the general processor and the zIIP. The zIIP is designed so a software program can work with z/OS to dispatch workloads to the zIIP with no anticipated changes to the application – only changes in z/OS and DB2. IBM DB2 for z/OS version 8 was the first IBM software able to take advantage of the zIIP. Initially, the following workloads can benefit: SQL processing of DRDA network-connected applications over TCP/IP: These DRDA applications include ERP (e.g. SAP), CRM (Siebel), or business intelligence and are expected to provide the primary benefit to customers. Stored procedures and UDFs run under TCBs, so they are not generally eligible, except for the call, commit and result set processing. DB2 9 remote native SQL Procedure Language is eligible for zIIP processing. BI application query processing utilizing DB2 parallel query capabilities; and functions of specified DB2 utilities that perform index maintenance. For more, see <http://www.ibm.com/systems/z/ziip/>

The DB2 9 and z/OS System Services Synergy Update paper discusses recent XML benchmark measurements and performance information.

<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101227>

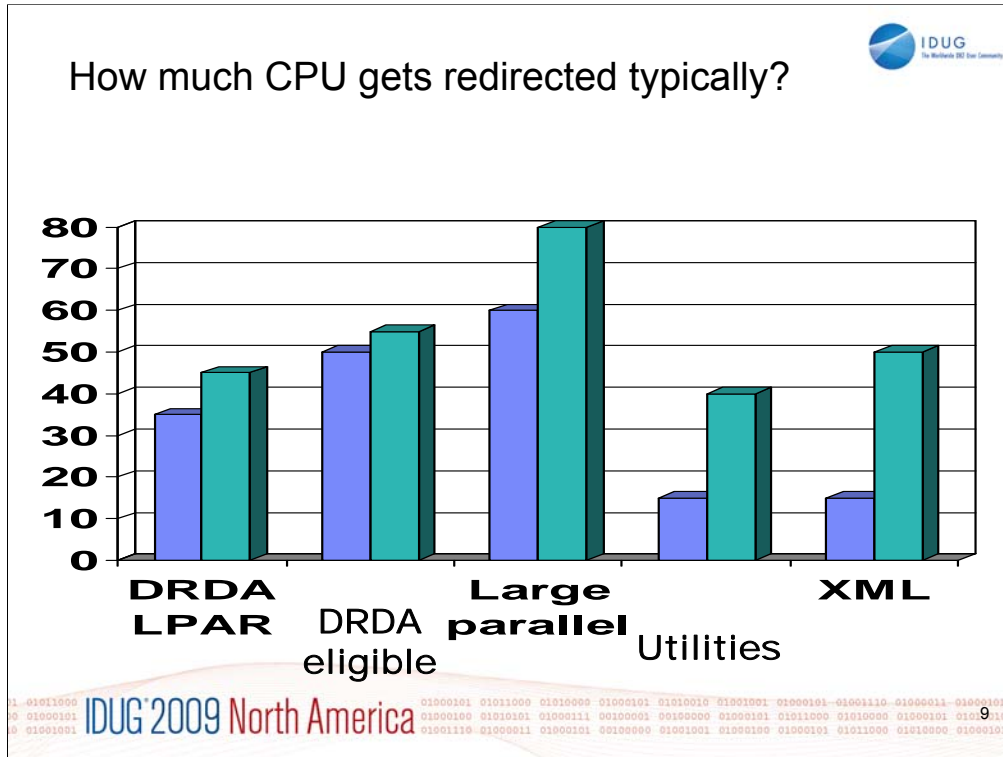
<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101387>

DB2 9 for z/OS remote native SQL procedures are described in this paper, showing scalability up to 3193 transactions per second for SQL procedures and redirect to zIIP of over 40%

<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD104524>

\* zIIP allows a program working with z/OS to have all or a portion of its enclave Service Request Block (SRB) work directed to zIIP. Above types of DB2 work are those running in enclave SRBs, of which portions can be sent to zIIP.





The range of processing redirected to zIIP and zAAP ranges widely. For some of the example workloads tested, this chart shows common ranges. With different workloads, your numbers will vary. The largest redirection is for large parallel queries, where as much as 80% of the CPU time is redirected. Some large parallel queries cannot be processed entirely in parallel, so the redirect percentage drops. When measuring the full LPAR, with work that cannot be redirected, such as operating system and performance monitors.

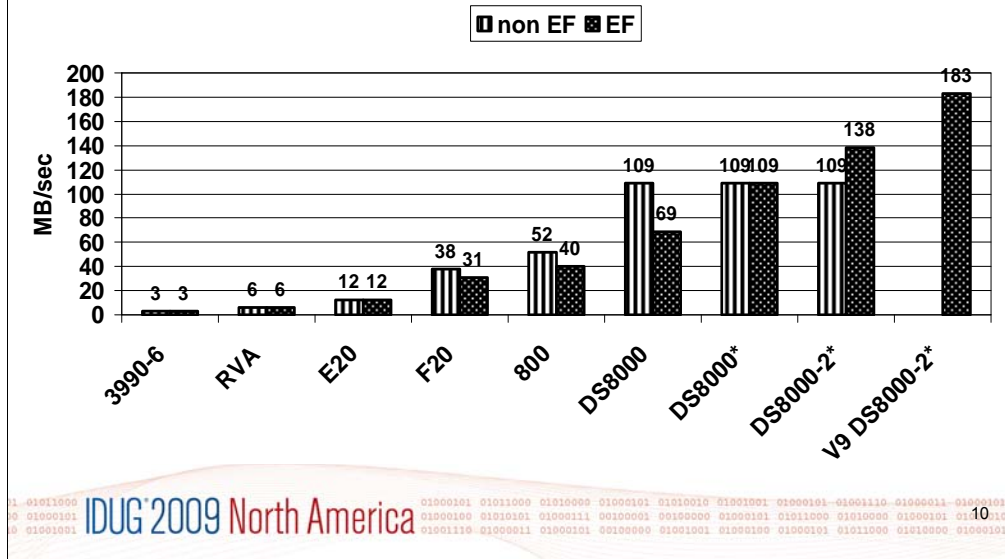
The range of utility redirection is large, depending upon the percentage of index processing. Most utility processing is not during the peak time, so this part of work is not as important. The utility CPU time is reduced in DB2 9, and the amount redirected is also reduced.

z/OS XML System Services consumes approximately 15% to 50% of total CPU time in measured XML insert or LOAD operations. This portion of CPU time is eligible to exploit zAAP redirection. The amount of CPU time for z/OS XML System Services will vary widely for other applications, based on the document size, its complexity, and number of indexes defined on XML tables.

See web resources and papers noted at the end of this presentation:

<ftp://ftp.software.ibm.com/software/data/db2/zos/presentations/overview/ziip-zaap-specialty-engines-idug-au-2008-miller.pdf>

## Disk performance for sequential read

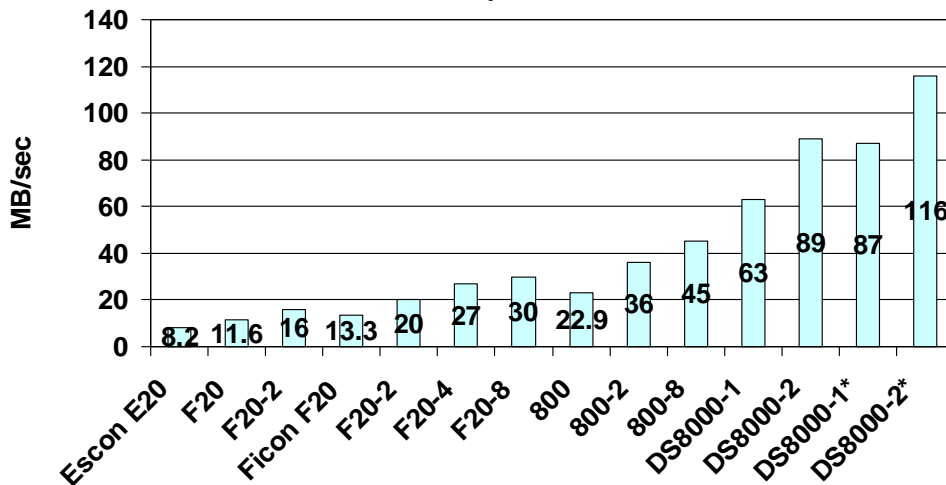


The recent performance for sequential read from disk has had a sharp upward trend. The 3390 and 3990-6 controller could read 3 MB per second as a data rate. Data rates climbed very slowly in prior generations of disk, but changes from a simple disk read to reading from a computer with similar amounts of memory and processing capability to the main computer have improved the speed by a factor of 61 as the processing use of memory and parallel processing bypasses the disk constraints.

The ESS models E20, F20 and 800 could deliver 12, 31 and 40 MB per second for the EF data sets, while non-EF data sets could read 12, 38 and 52 MB per second. Non-EF DS8000 reads reached 109 MB per sec, while the EF reads could only be 69 MB per second without the MIDAW. MIDAW with EF also reached 109 MB per second. With two stripes on the EF data set, DB2 achieved 138 MB per second. The changes in DB2 9 for larger prefetch quantity achieved 183 MB per second on the DS8000 with two stripes. The new DS8300 Turbo was over 200 MB per second.

### Maximum observed rate of active log write

- First 3 use Escon channel, the rest is Ficon.
- -N indicates N I/O stripes; \* MIDAW



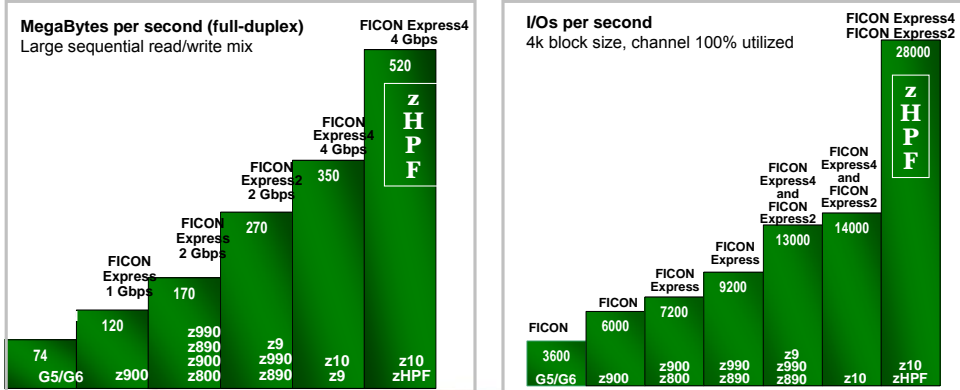
IDUG 2009 North America 11

The DS8000 and DS8000 turbo disks with channel performance improvements in fiber channels and in the System z9 processor have made big improvements in data rates for sequential reading. We have seen improvements from 40 MB / second on the ESS model 800, to 69 MB / second on the DS8000. Then use of the z9 and MIDAW improved that data rate to 109 MB / second. With two stripes, that configuration can reach 138 MB / second. DB2 9 changes in read quantity, write quantity and preformat quantity allow the same hardware to deliver 183 MB / second in reading and a similar speed for writing. With the MIDAW, the performance gap between Extended Format (EF) data sets and non EF data sets is practically eliminated. DS8000 Adaptive Multi-stream Prefetching (AMP) takes the next step in improved disk performance.

## z10 High Performance FICON for System z (zHPF)



- Optimization of storage area network (SAN) traffic using zHPF to improve performance
  - ▶ Maximum number of I/Os per second can be increased by up to 100%\*
  - ▶ For OLTP workloads (DB2, VSAM, PDSE, and zFS ) that transfer small blocks of fixed size data (4K blocks)
- Exclusive to System z10 - FICON Express4 and FICON Express2
- Requires IBM DS8000™ Release 4.1



IDUG 2009 North America

12

High Performance FICON and Solid State Disk more than doubles the random throughput per channel for the same amount of channel time.

Channel consolidation - fewer channels can manage more storage capacity. Actual throughput depends on the percentage of I/Os eligible for zHPF. DB2 prefetch I/Os are not yet eligible.

Improves SSD response time and throughput by 20%

zHPF requires DS8000 Release 4.1, z10 processor and z/OS 1.10 or SPE to z/OS 1.8

SSD technology speeds up data access and removes bottlenecks imposed by spinning disks

– Other important performance features include MIDAW, HyperPAV, AMP, High Performance FICON, 4 gbps FICON links

Optimization of storage area network (SAN) traffic using zHPF to improve performance

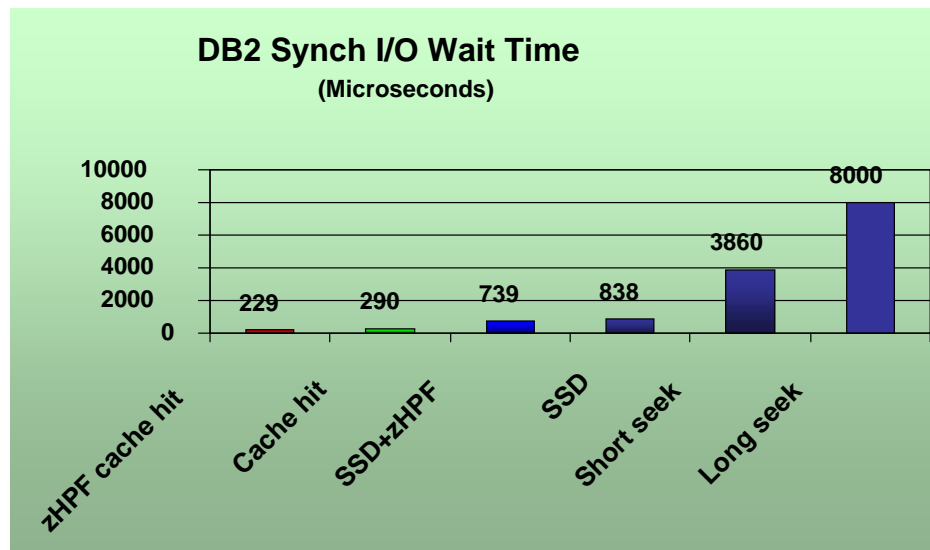
Maximum number of I/Os per second can be increased by up to 100%\*  
For OLTP workloads (DB2, VSAM, PDSE, and zFS ) that transfer small blocks of fixed size data (4K blocks)

Exclusive to System z10 - FICON Express4 and FICON Express2

Requires Control unit exploitation – IBM DS8000™ Release 4.1

z/OS V1.7 with the IBM Lifecycle Extension for z/OS V1.7 (5637-A01), V1.8, V1.9, or V1.10 with PTFs

## Solid state disk in storage pyramid



IDUG'2009 North America 13

Understanding where solid state disk fits into the storage hierarchy is very important. This device is very fast for random IO, but it's more expensive than magnetic disk.

Solid state disk provide much faster random IO times than hard disk. The disk cache is the fastest, at 229 to 290 microseconds. The solid state disk times are 739 to 838 microseconds. Getting data from the spinning disk requires roughly 4 to 8 milliseconds.

If you have very effective bufferpools and disk cache today, then solid state disk might not be as useful. If your IO is more random, so that your average random IO is 4 milliseconds or more, then the solid state disk may be able to hold enough of the highest IO density information to reduce the IO times significantly.

While solid state disk is higher than magnetic disk today, the cost is rapidly lowering, and new technology promises even more reductions. See Jeff Berger's presentations for more detail.

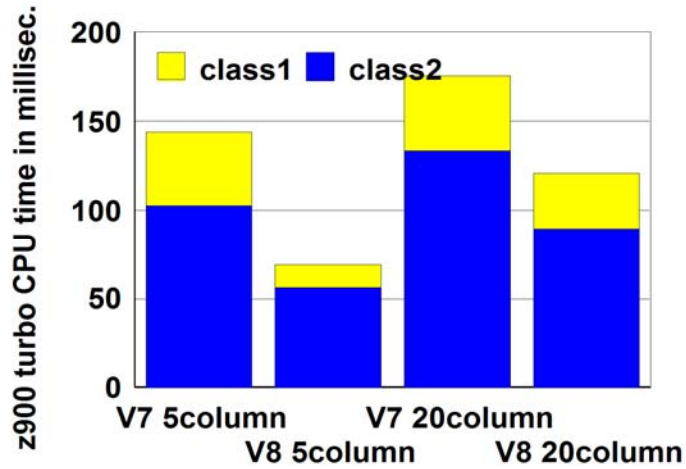
## Automatic use of multi-row Fetch

- DRDA even in CM
- DSNTTEP4 = DSNTTEP2 with multi-row fetch
  - ▶ Up to 35% CPU reduction in fetching 10000 rows with 5 and 20 columns
- DSNTTIAUL (sample unload application)
  - ▶ Up to 50% CPU reduction in fetching 10000 rows with 5 and 20 columns
- QMF with APAR

You can enhance the performance of your application programs by using multiple-row FETCH and INSERT statements to request that DB2 send multiple rows of data, at one time, to and from the database. Using these multiple-row statements in local applications results in fewer accesses of the database. Using these multiple-row statements in distributed applications results in fewer network operations and a significant improvement in performance.

In some situations, DB2 has already made those changes for you. DRDA is improved even in CM. Applications changes come in NFM, with DSNTTEP4, DSNTTIAUL and QMF.

## DSNTIAUL fetching 10000 rows with 5 and 20 columns



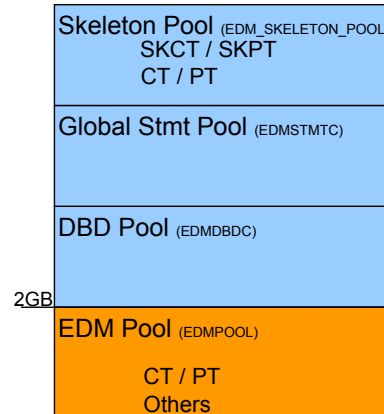
Chapter 9 of DB2 V8 Performance Topics discusses the DSNTPE4 and DSNTIAUL changes in detail. This chart shows the changes for small numbers of columns, such as 5 versus 20.

## DB2 9 64 bit Evolution (Virtual Storage Relief)



Virtual Storage Constraint is an important issue for many DB2 customers.

- EDMPOOL Changes:
  - ▶ V8 – DBD storage moved above 2GB bar.
  - ▶ DB2 9 – SKCT, SKPT, some CT, PT storage moved above 2 GB bar, when BIND occurs
  - ▶ DB2 9 approx. 60% reduction in EDMPOOL size observed for lab workloads
- Other changes:
  - ▶ Some storage acquired for distributed applications moved above 2GB bar.
  - ▶ Control blocks for table spaces and RTS move above the bar.
  - ▶ DSC statement text moved above the bar
    - SAP tests have shown almost 300 MB reduction in virtual storage below 2 GB bar



IFCID 217: detailed DBM1 virtual storage health

IFCID 225: consolidated DBM1 virtual storage health

IDUG 2009 North America 16

Objective was 10-15% relief. EDMPOOL – you can estimate DB2 9 EDM pool size from V8 stats as follows:  $(\#pgs \text{ for SKCT/SKPT}) * 0 + (\#pgs \text{ PT/CT}) * 70\%$ . Rough ROT is V8 EDM pool size can be reduced by 60%. Extensive use of SQL SPs can drive up EDM pool usage since these pkgs are larger. General recommendation, keep this the same in DB2 9. EDM\_SKELETON\_POOL – added in DB2 9.

For SAP SD transaction with 360 user threads, the total DBM1 virtual storage below 2GB was 1091MB in V8 and 819MB in DB2 9 for almost 300MB reduction. Most of that comes from local dynamic statement cache storage going down from 466MB to 172MB. Net of 1% real storage increase overall. For TPCE for example, 532 pages below 2GB were used for PT in V8. In DB2 9, 763 pages are used below 2GB and 316 pages are used above 2GB. So there is an increase in below 2GB virtual storage usage for PT in DB2 9. 1268 pages used for SKPT in V8 are all moved above 2GB in DB2 9 with a total of 1329 pages. EDM pool pages used below 2GB therefore dropped from 532+1268 to 763, or 58%.

EDM LRU latch contention relief (LC24). Previously one LRU chain in EDM pool with only one corresponding latch. Now split into three pools each with 3 latches: 1) SKCT/SKPT above 2GB 2) CT/PT above 2GB 3) CT/PT below 2GB

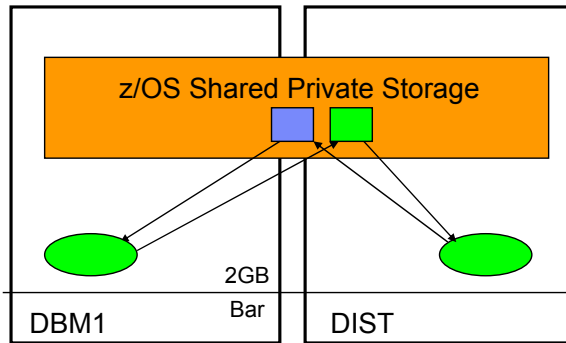
DBM1, the following are moved above the bar in DB2 9

- Parse trees peak below-the-bar storage for full prepare reduced 10%
- EDM fixed pools: V8 customer dumps show as much as 50MB will be moved. Allows larger above the bar EDM pools
- SKPTs / SKCTs (primarily static SQL). Also part of the CTs/PTs; New EDM pool for skeletons; Savings in below the bar 10MB to 300MB
- Page set blocks, RTS blocks: up to 10s of MB savings
- Local SQL statement cache: rough ROT: about 60% moves above bar
- Thread-related storage: Certain Run Time Structures, space block DMTR 10s of MB or more in savings
- Average V8 customer may see another 200 MB of relief: RID Pool, Sort Pool, compression dictionaries, castout buffers and others already above the bar (V8).



## 64-bit DDF – Shared Private with DBM1

- DDF address space runs in 64-bit addressing mode
  - Shared 64-bit memory object avoids cross memory moves between DBM1 and DDF and improves performance
- Shared memory: new virtual storage type allowing multiple address spaces to share storage.
- Similar to ECSA – always addressable, avoids AR and XM moves.
- Different from ECSA – only available to those address spaces registering with z/OS to share this storage.
- Reduces data formatting and data movement
- Reduces virtual storage
  - It exists once, instead of in each address space



The DDF address space now has 64 bit addressing, sharing above the bar space with DBM1. Performance is the primary improvement for distributed work, saving movement of storage across the address spaces. This technique requires the 128 GB virtual space, but this space is not backed unless used, and the amount of storage is reduced. DB2 requires a minimum of 128GB configured or DB2 9 will not start. Set by HVSHARE in Parmlib

(Notes continued from prior page)

Skeletons above the bar: For customers that use heavy package and plan activity such as banks, this is the most significant DBM1 below the bar storage relief in DB2 9. For customers that use very few or small packages, such as SAP environments, the savings is smaller.

LI702 – move spaceblk (SPA) above the bar. SPA to be split into 2, 1 above, 1 below. Only a few, non-complex RTs are being considered for DB2 9. Simple insert, delete Expected results will vary by SQL mix. (-5 to 30%)?

Has storage key & fetch protection

Defaults to 2TB size

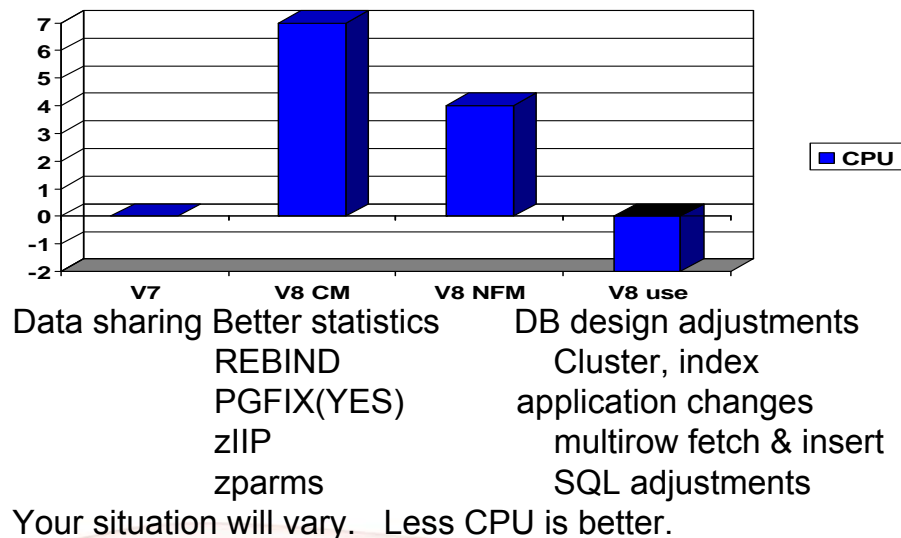
## System tuning

- New versions and service
- Parameter changes
- Buffer pools and other storage
- Use of zIIP and zAAP
- Thread reuse
- Enough performance monitoring,  
not too much

The subsystem tuning options can make a significant difference, although the database administration and application programming options generally provide larger improvements.

New versions and service can make a significant difference in performance. Some of the subsystem parameters can help, particularly when there is a bottleneck, such as memory. Use of zIIP and zAAP engines can help with costs. Thread reuse can help, when used appropriately.

## V8 best practice performance plan example scenario



Your situation and mileage will vary, but this is a common shape for a V8 performance plan, starting with zero for the V7 base line. When you move to V8, CPU time generally increases from 5% to 10%, shown here as 7. Start with long term page fix for buffer pools with high numbers of pages read and written. Reorg and collect improved statistics for non-uniform distribution of data on non-indexed columns. The V8 CM performance plan REBINDs the primary packages, and adjusts DSNZPARMs. The CM REBIND process provides most of the improved access paths. Data sharing batching helps in CM. During CM, a zIIP is added if your peak work load includes DRDA SQL, parallel query or LOAD, REORG and REBUILD.

In moving to NFM, some additional DSNZPARMS are adjusted and REBIND all plans and packages. Database designs start taking advantage of new clustering & indexing options, such as NOT PADDED for large varchar indexed columns. After making the design changes, REORG the data; REORG or REBUILD the indexes; get improved statistics & REBIND. The data sharing group is quiesced, and protocol 2 locking is used.

V8 use takes more advantage of the V8 performance improvements: MQTs, DPSI, more not-padded indexes, multi-row Fetch, cursor Update, cursor Delete, & Insert. Use other SQL improvements to reduce V8 CPU, less than V7. The work may grow, but some of the growth uses the zIIP.


## DB2 for z/OS V8 Performance Overview

- Performance / Scalability Enhancements
  - ▶ Improved partitioning scale and flexibility
  - ▶ Many index improvements
- Query / Access Path Performance Enhancements
- Multirow fetch and insert
- Synergy with new hardware: zIIP, MIDAW, DS8000, ...




DB2 for z/OS V8 has many key performance and scalability improvements. The biggest CPU saver for most customers is the ability to insert and select multiple rows, reducing the CPU for crossing address spaces and making a bigger improvement for distributed cases. Optimization improvements, with many new options for partitions and indexes, improved optimization techniques and better information for the optimizer are important.

Synergy with both processor and IO hardware is extended substantially. The ability to use zIIP improves costs. The memory structure expansion to 64 bit addressing helps in many ways. Faster disk speeds keep the system balanced.



V8 queries and data warehouses

DB2 UDB for z/OS  
Version 8



- Optimization Improvements
  - ❑ Materialized Query Tables
  - ❑ Improved optimization techniques
  - ❑ Enhanced data for optimizer
  - ❑ Visual Explain, Optimization Service Center
- Enhanced index options
- New Partitioning options
- QMF improvements
- SQL enhancements, multirow fetch & insert

IDUG 2009 North America

Queries and data warehousing are improved a lot in V8. Optimization improvements provide a performance boost and make the job simpler. Improved optimization techniques like ability to use indexes more, star join and scale improvements allow reduced work for computers and for people. Enhanced data helps get the best access path. Visual Explain improves the ability to analyze and resolve any problems.

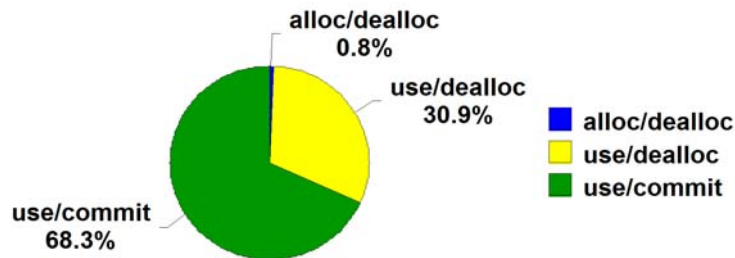
The many improvements for indexes, materialized query tables and partitioning can save space and add new options for improved performance and availability, even while simplifying the process. Not padded, clustering, longer and backward scans help indexes. Being able to add, rotate and rebalance partitions improve partitioning options. QMF enhancements build upon these strengths and add new function to reporting, dash boards, and a new platform in WebSphere. SQL enhancements on this page and the next improve portability of the SQL, improve the ability to express queries, and help with performance.

## Distribution stats on single and multiple columns

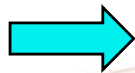
- Top N highest, and/or lowest, frequency of values and cardinality

Bind option

Acquire /  
release  
example



**SELECT FROM A, SYSIBM.SYSPLAN B WHERE B.ACQUIRE='A'  
AND B.RELEASE='D' ...**



**Better join sequence from more precise filter  
factor estimation of combined predicates**

To run efficiently, data warehousing, data mining, and ad hoc query applications need statistics on columns that are in predicates, regardless of whether they are leading columns of an index. In addition, distribution statistics on non-leading index columns or non-indexed columns let DB2 make better access path decisions when data is asymmetrically distributed. In Version 8, you can use the RUNSTATS utility to collect the following additional statistics:

- Frequency distributions for non-indexed columns or groups of columns
- Cardinality values for groups of non-indexed columns
- Least-frequently occurring values, most-frequently occurring values, or both, for any group of columns

## Thread Reuse

**Thread reuse for 5 to 20% CPU time reduction for light transactions**

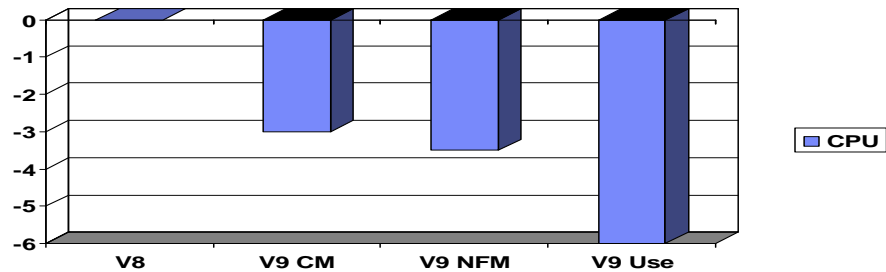
| NORMAL TERMINATION | AVERAGE | TOTAL  |
|--------------------|---------|--------|
| NEW USER           | 1.00    | 174752 |
| DEALLOCATION       | 0       | 0      |
| RESIGNON           | 0       | 0      |
| INACTIVE           | 0       | 0      |

➤ All except DEALLOCATION indicate successful thread reuse.

Thread reuse is very effective for very small, light transactions with a high rate. Thread reuse is important for a case with three SQL statements running 100 tps. When the transaction has 10, 20 or 100 SQL statements, the value is diminished sharply. If the transaction rate is less than 10 per second or the number of SQL statements per transaction is more than 20, then the value of thread reuse is minimal.

When the value of thread reuse is minimal, then the costs of keeping the threads – in memory and CPU – generally exceeds the value.

## DB2 9 z10, z9, z890 & z990 performance plan example scenario



|                      |                       |
|----------------------|-----------------------|
| Utilities            | DB design adjustments |
| Histogram statistics | Index improvements    |
| REBIND               | application changes   |
| DSNZPARMS            | native SQL procedures |
|                      | SQL adjustments       |

Your situation will vary. Less CPU is better.

z800 and z900 expect +5% to +10% CPU

If you have a z9, z990 or z890, this is expected to be a common shape for a DB2 9 performance plan, starting with zero for the V8 baseline. When you first move to DB2 9, total DB2 CPU time generally decreases from 0% to 5% for z9, z890 and z990 customers, shown here as a first step -3%. Utility CPU reductions help immediately. Some work will be about the same (+/-3%). Start with reorgs and collect improved histogram statistics when useful. The DB2 9 CM performance plan REBINDs the primary packages and adjusts DSNZPARMS. The REBINDs provide most of the improved access paths. On z800 or z900 the initial CPU expectation is +5 to +10% regression, more if there are many columns, so making adjustments is more important.

In moving to NFM, some additional DSNZPARMS are adjusted and all plans and packages are rebound. The DB2 9 use line takes wider advantage of DB2 9 performance improvements. Database designs start taking advantage of new indexing options, such as compression, index on expression and larger pages. After making the design changes, REORG the data and REORG or REBUILD the indexes, get the improved statistics and REBIND. Native SQL procedures, added use of zIIP, and improved SQL continue the improvements in this phase.

Scenario: Customer mix of DB2 CPU time is 30% in utilities, 70% in SQL access. With 10% improvement for the utilities, we get a -3% net, assuming that SQL is the same as before. With optimization improvements, another -1/2% improvement shows up in DB2 9 NFM. Then as design adjustments, reorgs and rebinds are performed, we get improvements from varchar improvements, native SQL procedures and improved SQL, another -3%.



## DB2 9 for z/OS Performance Overview

- Significant CPU time reduction in most utilities
- Synergy with new hardware: zIIP, MIDAW, DS8000, ...
- Performance / Scalability Enhancements
  - Especially Insert, Update & Delete
- Query / Access Path Performance Enhancements
- Other performance enhancements: LOBs, varchar, native SQL procedure, index compression
- Improved virtual storage usage below 2GB DBM1

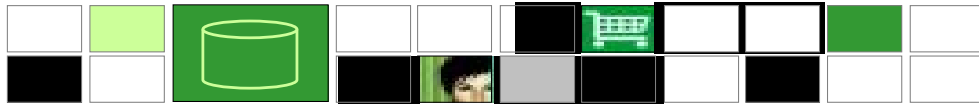


The key performance improvements in DB2 9 are reduced CPU time in many utilities, deep synergy with System z hardware and z/OS software, improved performance and scalability, especially for insert, update and delete, better LOB performance and scalability, improved optimization for SQL, zIIP processing for remote native SQL procedures, index compression, reduced CPU time for data with varying lengths and better sequential access. This version also improves virtual storage use below the 2 GB bar.

The optimization improvements include more function to optimize, improved information for optimization, better optimization techniques and a new approach to providing information for tuning. V8 SQL procedures were not eligible to run on the zIIP, but changing to use the native SQL Procedure Language on DB2 9 will make the work eligible for zIIP processing. Varying length data can improve substantially if there are large numbers of varying length columns. Several improvements in disk access can reduce the time for sequential disk access and improve data rates.

## Most consumable DB2 9 improvements

- CM very little to no action:
  - Utility CPU reductions
  - Logging improvements
  - Improved index page split
  - Larger prefetch, write & preformat quantities
  - LOB performance
  - DDF VSCR
  - Optimization Service Center, Opt. Expert, & Data Studio
- Changed online REORG
- Improved RUNSTATS
- Optimization improvements, EDMPOOL VSCR
- More parallel, use of zIIP
- NFM
  - LOB lock avoidance
  - Reordered row format
  - Index: larger page sizes, compression, index on expression



Here are some highlights for items that deliver the most quickly and easily: Very little to no action is required for the utility CPU reductions, logging improvements, improved index page split, larger prefetch, write & preformat quantities, some LOB performance, DDF virtual storage constraint relief. The first group delivers in CM. The next items require some work. Changed online REORG and other utility improvements require process changes and use of SHRLEVEL(CHANGE). Improved RUNSTATS statistics needs some analysis to determine where the value is greater than the cost of gathering the new statistics. Optimization improvements are automatic for dynamic SQL, but require work to REBIND for static SQL. In both cases, we need baselines to check for regression. EDMPOOL virtual storage constraint relief also requires a REBIND. Optimization Service Center takes some learning, but should be fast for those who have used Visual Explain in the past. See the book, SG24-7421, DB2 9 for z/OS: New Tools for Query Optimization. LOB lock avoidance requires NFM and APAR PK62027 to avoid group quiesce. Reordered row format requires a REORG in NFM and varying length columns. Index improvements for larger page sizes, compression, index on expression require database design work to determine where they are applicable. ALTERs, REORGs and creation of new indexes are needed.

## DB2 9 Utilities Performance Improvements

### CPU reductions in LOAD, REORG, and REBUILD

- Reductions mostly due to improved index processing (\* with exceptions)
- 10 to 20% in Image Copy\* (even with forced CHECKPAGE YES)
- 5 to 30% in Load, Reorg, Reorg Partition, Rebuild Index
  - ▶ Except REORG TABLESPACE SHR CHG PART with NPSIs
- 20 to 40% in Load
- 20 to 60% in Check Index
- 35% in Load Partition
- 30 to 40% in Runstats Index
- 40 to 50% in Reorg Index
- Up to 70% in Load Replace Partition with dummy input

We have seen substantial performance improvements in the utilities, with some early customers noting as much as 20% to 30% overall reductions in utility CPU time. If utilities are 30% of the total DB2 CPU time and you average 20% improvements, then the net would be a 6% reduction in total DB2 CPU. We anticipate that most of the savings would not be for the peak processing time.

The utility improvements are broad-based, but not all processing improves. The primary improvements are in index processing, so you will probably have larger improvements if you have more indexes. Many of the CPU savings are in index processing, so the zIIP redirection for DB2 9 utilities is generally less than V8.

The utilities that process only indexes see the most dramatic improvement. For example, REORG INDEX sees a 40-50% improvement while REORG table space is only 5-30%. That's because during REORG TABLESPACE there is a lot of data row processing so that index key processing is proportionally less of the entire job than for REORG INDEX.

COPY is improved with less data movement for pages than in previous releases. This is seen even with some additional CPU needed for the default CHECKPAGE YES option.

For LOAD REPLACE of a part with no data rows, the NPIs must be updated to delete logical keys. The interface to index manager was improved so that the delete of these keys is save up to 70% of the CPU.

## DB2 9 Query Enhancements

- SQL enhancements: INTERSECT, EXCEPT, cultural sort, caseless comparisons, FETCH FIRST in fullselect, OLAP specifications: RANK, ROW\_NUMBER, ...
- pureXML integration and text improvements
- Index improvements
  - ▶ Index on expression            Larger index pages
  - ▶ Index compression            Improved page split
- Improved Optimization statistics: Histogram
- Optimization techniques & REOPT(AUTO)
  - ▶ Cross query block optimization
  - ▶ Generalize sparse index & in-memory data cache method
  - ▶ Dynamic Index ANDing for Star Schema
- Analysis: instrumentation & Optimization Service Center

Query enhancements improve data warehousing and reporting. Today's complex applications include both transactions and reporting, so performing both well is imperative. More queries can be expressed in SQL with new SQL enhancements. The set operators INTERSECT and EXCEPT clauses make SQL easier to write. OLAP extensions for RANK, DENSE\_RANK and ROW\_NUMBER add new capabilities. Other SQL statements improve consistency with the DBMS industry. DB2 9 continues the progress in SQL, with many new functions, statements and clauses. The biggest changes are in XML. New SQL data manipulation statements are MERGE and TRUNCATE. New data types with DECIMAL, FLOAT, BIGINT, BINARY and VARBINARY. Improvements in LOBs provide new function, more consistent handling and improved performance. Security is improved with network trusted context and roles. Data definition consistency and usability are improved. DB2 9 is another big step in DB2 family consistency and in the ability to port applications to DB2 for z/OS.

Indexes have many improvements in DB2 9. The key items are the ability to have an index on an expression instead of on a column, compression to save disk space, larger index pages and an improved page split to improve the insert rate.

Data sizes continue to increase while the SQL grows more complex. The SQL enhancements provide more opportunities for optimization, and DB2 9 adds optimization enhancements to improve query and reporting performance and ease of use. Improved data is provided for the optimizer, with improved algorithms and a rewritten approach to handling performance exceptions. Histogram statistics provide better information about non-uniform distributions of data when there are many values skewed, rather than just a few. Improved algorithms widen the scope for optimization. When exceptions occur, guidance and support are made easier with improved instrumentation, the Optimization Service Center and the Optimization Expert.

## Histogram Statistics - RUNSTATS

- Non-uniform distribution with a high cardinality
- Histogram statistics addresses skews across ranges of data values
- Summarizes data distribution on an interval scale
- DB2 uses equal-depth histograms
  - Each quantile has about the same number of rows
  - Example - 1, 3, 3, 4, 4, 6, 7, 8, 9, 10, 12, 15 (sequenced), cut into 3 quantiles

| Seq No | Low Value | High Value | Cardinality | Frequency |
|--------|-----------|------------|-------------|-----------|
| 1      | 1         | 4          | 3           | 5/12      |
| 2      | 6         | 9          | 4           | 4/12      |
| 3      | 10        | 15         | 3           | 3/12      |

## RUNSTATS

Maximum 100 quantiles for a column

Same value columns WILL be in the same quantile

Quantiles will be similar size but:

Will try and avoid big gaps between quantiles

A column high value and low value may have separate quantiles

Null WILL have a separate quantile

If less than 100 column values, reverts to Distribution Stats

Not supported with LOAD and REORG

Supports column groups as well as single columns

## Accounting Class 1 and 2

| AVERAGE            | CLASS1 | CLASS2  |
|--------------------|--------|---------|
| ELAPSED TIME       | 233ms  | 19ms    |
| CPU TIME           | 2.95ms | 2.71ms  |
| WAIT TIME          |        | 14.76ms |
| NOT ACCOUNTED TIME |        | 1.31ms  |

For most cases

- Class 1 for application + DB2 time
- Class 2 for DB2 time only
- CICS without TS 2.2 or later threadsafe option
  - Class 1 CPU for task switch + DB2 time
  - Class 2 for DB2 time only

### Accounting report (not trace) by connection type most useful for initial analysis

**Omegamon DB2 Performance Expert ACCOUNTING REPORT LAYOUT(LONG) ORDER(CONNTYPE) EXCLUDE(PACKAGE(\*)) to group by thread connection type such as TSO, CICS, DB2CALL, RRS, IMS, DRDA, etc. for the period of interest.**

**Also STATISTICS REPORT LAYOUT(LONG) for the corresponding period extremely desirable**

Resolving performance challenges is much simpler with the needed information, so that it's easy to tell if the problem is inside DB2 or not. In most situations, that means needing both accounting class 1 and 2. For CICS transactions that are not using the threadsafe option, class 1 is often adequate.

## High NOT ACCOUNTED time – 2 most likely causes

- ❑ CPU wait under high CPU utilization, especially with lower dispatching priority
  - E.g. goal mode with low priority for DB2 address space compared to DDF enclave, CICS, WebSphere address space, or DDF enclave with SYSOTHER (discretionary)

- ❑ Excessive detailed online tracing with vendor tools

Other causes are much less frequent and widely varied

Some events not being captured by DB2, but more events are being captured in newer versions

Details on the web:

<http://www.ibm.com/support/docview.wss?rs=64&context=SSEPEK&uid=swg21045823>

Online support document: <http://www.ibm.com/support/docview.wss?rs=64&uid=swg21045823>

What is DB2 Accounting Class 2 Not Accounted Time? The following simple formula defines DB2® Class 2 Not Accounted Time:  $DB2\ Class\ 2\ Not\ Accounted\ Time = DB2\ Class\ 2\ Elapsed\ time - (DB2\ Class\ 2\ CPU\ time + DB2\ Class\ 3\ suspension\ time)$

Usually the DB2 Class 2 Not Accounted time is very small or negligible. It represents time that DB2 is unable to account for. If you see significant DB2 Class 2 Not Accounted time, it could be due to one of the following reasons. In some cases, high DB2 Class 2 Not Accounted time is caused by too much detailed online tracing or bugs in some vendor performance monitors. Reduce the level of tracing or stop the vendor performance monitor to help reduce the Not Accounted time to an acceptable level. This situation is usually the primary cause of high not-accounted-for-time on systems that are not CPU-constrained.

In a non-data sharing environment, it could be due to running on a very high CPU utilization environment and waiting for CPU cycles, especially with lower dispatching priority in Work Load Manager goal mode. A non-dedicated LPAR can be interrupted by another LPAR and lose the processor for some time. It could be due to running in a high MVS™ paging environment and waiting for storage allocation. If DB2 gets swapped out by losing control of the processor or waiting for a processor, this increased time is the result.

In a data sharing environment, prior to V7, it could be due to asynchronous coupling facility requests. For example, group buffer pool requests for > 4KB pages, long running coupling facility commands such as 'Read Directory Info' and 'Delete Name under mask', or conversion of synchronous requests to asynchronous requests due to a coupling facility subchannel busy condition. In V7 or later, the coupling facility suspensions due to asynchronous requests are shown under a new category called 'Asynch IXL Requests' in DB2 performance monitor. If the asynchronous write to the secondary group buffer pool (GBP) does not complete before synchronous write to the primary GBP with GBP duplexing, not accounted time can be the result.

Not accounted time can be the wait time for return from requests to be returned from VTAM® or TCP/IP. Instrumentation Facility Interface (IFI) log read can cause not accounted time.

If the environment is very I/O intensive, the Media Manager might be running out of request blocks.

z/OS® events can cause not accounted time, such as SRM timer pops, for example when an SMF SRB is triggered to collect open data set statistics. Waiting for package accounting can result in not accounted time. Not accounted time can be accounted for by a PREPARE which is not found in the Dynamic Statement Cache. Waiting for a page being moved from VP to HP can result in not accounted time. DD consolidation (z/OS parameter DDCONS=YES DETAIL) has overhead that is not accounted. See DDCONS informational APAR I107124. Data set open contention related to PCLOSET being too small can cause time that is not accounted. Time for RMF interval data set statistics gathering can cause not accounted time. DB2 internal suspend and resume can cause this not accounted time by looping when several threads are waiting for the same resource, but this case is very rare.

## Accounting Class 3

| SUSPENSIONS               | TOTAL TIME     | #EVENTS      |
|---------------------------|----------------|--------------|
| LOCK/LATCH                | 0.11ms         | 0.3          |
| SYNC DATABASE I/O         | 8.73ms         | 8.86         |
| SYNC LOG WRITE I/O        | 1.64ms         | 0.49         |
| OTHER READ I/O            | 2.64ms         | 0.76         |
| OTHER WRITE I/O           | 0.004ms        | 0.00         |
| SERVICE TASK              | 1.60ms         | 0.47         |
| .....                     |                |              |
| <b>TOTAL CLASS 3 WAIT</b> | <b>14.76ms</b> | <b>10.88</b> |

- Class 3 acctg strongly recommended: Negligible overhead except when high DB2 latch contention, eg over 10000 per second

**Lock/Latch wait = Lock wait + IRLM latch wait + internal DB2 latch wait**

**In the rare case of over 10000 per second, disabling class 3 may significantly bring down class 1 and 2 CPU time.**

**Sync I/O wait = wait for read or write I/O by this application agent**

**Avg time = 8.73ms/8.86 = 0.985ms**

**Other read I/O wait = wait for read I/O by another application agent or prefetch engine**

**Other write I/O wait = wait for write I/O by another application agent or write engine, may include some time waiting for log write-ahead**

Accounting class 3 is strongly recommended for nearly every customer. The overhead is very low other than in the exceptional case of very high latch contention (over 10 thousand per second, as a rule of thumb).



## I/O wait time tuning

- **Buffer pool tuning**
- **Compression (data and index)**
- **Synchronous read → prefetch**
- **I/O configuration tuning**
  - **Make sure of sufficient I/O resources**
  - **Faster device, such as DS8000 as needed**
  - **Parallel Access Volume (PAV) beneficial if I/O contention with high IOSQ time in RMF**
  - **I/O striping**

The first step in tuning I/O is to see if the I/O can be avoided or reduced by using buffer pool tuning and compression. If large numbers of synchronous I/Os are needed, changing to asynchronous prefetch can often improve the time.

If the response times for a single page synchronous reads and 32 page prefetches are in the 1 to 2 ms range, then there is little to gain from IO tuning. Sometimes reorganization can help a lot. If the I/O is very random, then the disk access takes longer. If your times are larger, then I/O configuration tuning can help.


Do you have fast devices where needed? I/O tuning often means looking at both disk performance information and at DB2 performance information.

## Database design tuning

- Index changes: many new options
  - Ability to use index
  - Clustering
  - Compression
  - Index on expression, XML, ...
- Universal table space

Both DB2 V8 and DB2 9 have large changes to index design options. With more effective options, indexes can be used in situations that were not possible in prior releases. The improvement can be orders of magnitude.

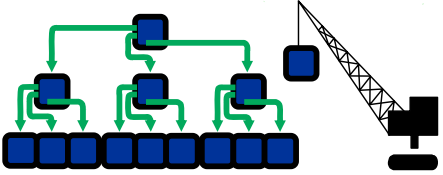
The biggest change for table spaces since Version 2 is the universal table space. It is both partitioned and segmented in the space maps. It can be partitioned even without a partitioning key. This is the table space structure for the future.




## Index: DB2 for z/OS V8

### Index Improvements

- Variable length index keys
- Index-only access for varchar data
- Maximum index key 2000 bytes
- Predicates indexable for unlike types
- Backward Index Scan
- Partitioning separate from clustering
- Data-partitioned secondary indexes (DPSI)
- Create index online during select, insert
- Add column to index





DB2 V8 provides many new opportunities for improving index processing, rebuilding the architecture for indexes.

We are able to use indexes more effectively, reducing the space in variable-length indexes, being able to have index-only access with variable-length data and being able to use the index when the predicates do not match.

In some cases, such as backward index scans or partitioning, we will be able to work as efficiently with one less index. Being able to eliminate an index will improve the insert, delete, LOAD, REORG and update processing.

We have more flexibility in indexes, with longer index keys, the ability to partition secondary indexes and the ability to have more effective clustering.


## More Indexable Predicates

- **For column comp-op value with unlike type or length**
  - ▶ **4 byte char column = 8 byte host variable**
  - ▶ **Integer column = decimal host variable**
  - ▶ **Stage 2 and non indexable in V7**
  - ▶ **Stage 1 and indexable in V8**
    - **So index on char or integer column here can be used in V8 but not in V7**
  - ▶ **Also useful where a programming language does not support SQL data types. For example,**
    - **No decimal type by C/C++, no fixed-length char by Java**

The most common mismatches for data types come with languages like Java, C++ and C and decimal data. Often the comparison is from a floating point host variable to a decimal column.

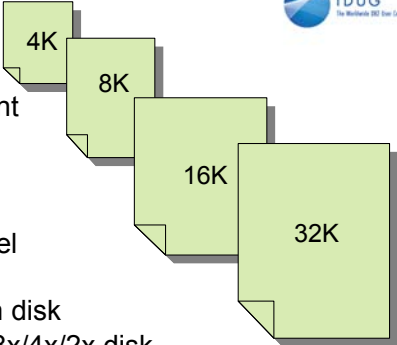
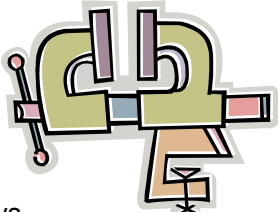
A second type of mismatch that is very common is to have a literal or host variable with a character column length greater than that of the column.

For both of these cases, the result was often poor performance because of the inability to use an index. While there are still some restrictions, performance is expected to improve substantially for many customers.



## Indexing Enhancements

- Larger index pages allow more efficient of storage
  - ▶ Fewer page splits for long keys
  - ▶ More key values per page
- Index compression provides page-level compression
  - ▶ Data is compressed to 4K pages on disk
  - ▶ 32K/16K/8K pages results in up to 8x/4x/2x disk savings
  - ▶ No compression dictionaries
    - Compression on the fly
    - No LOAD or REORG required
- Rebuild Index SHRLEVEL CHANGE
- Index on expression
- Define RANDOM index keys to avoid hot spots with multiple processes inserting sequential keys

IDUG 2009 North America 37

Indexing improvements contribute to the overall improvements in query performance. Specific improvements include index compression, index on expression, index key randomization, and larger index page sizes.

Larger index pages allow for more efficient use of storage, with fewer page splits for long keys and more key values per page.

Multiple processes inserting sequential keys can create hot spots on indexes. Randomized index keys avoid hot spots. Application insert throughput improved via avoidance of locking conflicts, but retrieval of sequential rows is likely to be slower.

Bigger index page: 4K, 8K, 16K, or 32K page → Up to 8 times less index split

Good for heavy inserts to reduce index splits. Especially recommended if high latch class 6 contention in data sharing.  
Two forced log writes per split in data sharing

Or high latch class 254 contention in non data sharing shown in IFCID 57

## Index Compression

### Difference between data and index compression

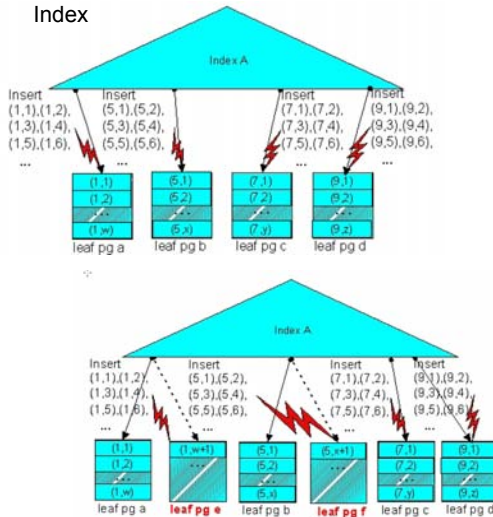
|                     | Data       | Index         |
|---------------------|------------|---------------|
| Level               | Row        | Page (1)      |
| Comp on disk        | Yes        | Yes           |
| Comp in Buffer Pool | Yes        | No            |
| Comp in Log         | Yes        | No            |
| Comp Dictionary     | Yes        | No (2)        |
| Average Comp Ratio  | 10% to 90% | 25 to 75% (3) |

Index compression can be a very important way to save disk space, especially if your indexes take more space than your compressed data. Index compression is very different from data compression, so your use will change significantly. Index compression does have some overhead in CPU time and in memory to save disk space. Index compression does not use a dictionary, so index data can be compressed as data is inserted. You can choose whether you want to use index compression by specifying `COMPRESS YES` or `COMPRESS NO` on the `CREATE INDEX` or the `ALTER INDEX` statements in DB2 9. See the Redpaper for much more.

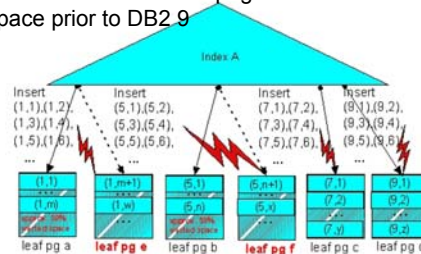
Index Compression DB2 9, REDP4345,  
<http://www.redbooks.ibm.com/abstracts/redp4345.html>

## Asymmetric Index Page Splits

Multiple Sequential Insert Patterns on an Index



Sequential inserts into the middle of an index resulted in some pages with 50% free space prior to DB2.9



New algorithm dynamically accommodates a varying pattern of inserts



### Asymmetric index page split depending on an insert pattern

Instead of 50-50 split

Up to 50% reduction in index split

-20% class 2 CPU, -31% elapsed time, -50% log write I/O and async CF requests in one data sharing measurement

2 log write I/O's per split in data sharing

-10% CPU, -18% elapsed time, -20% index Getpage and Buffer Update in one non data sharing measurement

Allowing index pages to split asymmetrically can improve space utilization and reduce contention that results from frequent page splits in an index with sequential insert patterns in the middle of the index. An index page size that is greater than 4 KB can also relieve contention by accommodating more index keys per page, which reduces the frequency of page splits in indexes. You can use the INDEXBP option on both CREATE DATABASE and ALTER DATABASE statements to specify 4-KB, 8-KB, 16-KB, or 32-KB index buffer pools, and the BUFFERPOOL keyword on the CREATE INDEX statement to specify 8-KB, 16-KB, and 32-KB buffer pools.

The performance improvement can be as much as a 50% reduction in index split time, 20% savings in class 2 CPU time and 31% elapsed time reduction for data sharing.

## Index on Expression

- Simple indexes can contain concatenated columns  
**create index totalComp on W2\_TABLE(salary, bonus)**
- Index on expression
  - ▶ Value of the index has been transformed
  - ▶ May not be the value of any of the columns that it is derived from
  - ▶ Optimizer can use this index

| W2_TABLE |         |         |
|----------|---------|---------|
| name     | salary  | bonus   |
| Gary     | 20,000  | 500     |
| Paul     | 40,000  | 5,000   |
| Matt     | 400,000 | 10,000  |
| Rex      | 40,000  | 2,000   |
| Rob      | 200,000 | 210,000 |

Create index totalComp on  
 W2\_TABLE(salary+bonus)

Query / Order on Total Compensation

Index on expression provides a new type of index. DB2 9 lets you create an index on a general expression. Query performance can be enhanced if the optimizer chooses that index. When you use an index on an expression, the results of the expressions are evaluated during insertion time or during an index rebuild and are kept in the index. If the optimizer chooses to use that index, the predicate is evaluated against the values that are stored in the index. As a result, run-time performance can be improved dramatically.

Test query WITH PAYTOT(FIRSTNME, LASTNAME, TOTALPAY) AS (SELECT FIRSTNME, LASTNAME, SUM(SALARY) + SUM(BONUS) FROM DBA032.EMP GROUP BY LASTNAME, FIRSTNME) SELECT FIRSTNME FROM PAYTOT WHERE TOTALPAY=(SELECT MAX(TOTALPAY) FROM PAYTOT)

applied to this example is

WITH PAYTOT(name, totalpay)

AS (select name, SUM(salary) + SUM(bonus) from W2\_TABLE GROUP BY name)  
 SELECT name from PAYTOT WHERE totalpay=(SELECT MAX(totalpay) from PAYTOT)



## DB2 9 Scalability

- Insert performance APPEND INDEX LOG  
 INDEX on expression, 8K, 16K, 32K, split  
 Randomized index key, larger preformat  
 Log Latch contention & spin relief, archiving  
 Not logged table space
- Partitioned table with segmented space
- Memory improvements 64 bit address space



Insert performance increases substantially, through a wide range of improvements. Logging performance is improved with latching improvements and striped archiving. The newer disk and channel changes (DS8000 Turbo, 4 Gb per second channels, MIDAW, & AMP), improve the data rates substantially. Indexes are improved, with larger page sizes to reduce the number of page splits and also a better page split. Where performance should be optimized for inserts, rather than for later retrieval, the append option can be used. If the data need to be randomized to avoid insert hot spots, the new randomized index key is useful.

The segmented space structure is more efficient in many situations, so adding that space structure for the large partitioned table spaces helps DB2 scale more efficiently.

Memory improvements continue the work from V8, with memory shared above the bar between the DDF and DBM1 address spaces. The shared memory can be used to avoid moving data from one address space to the other. More data structures from the EDMPOOL and dynamic statement cache are moved above the bar.

## Universal Table Spaces

- ▶ Combination of segmented with partitioning options
  - Better space management
  - Support of mass deletes / TRUNCATE
- ▶ If partitioned
  - Still must be one table per table space
  - Can choose **Range Based partitioning** (as before: PBR)
  - Can choose **Partitioned By Growth** (PBG)
- ▶ DROP / CREATE to migrate existing page sets
- ▶ Simple table spaces can not be created
  - Default table space is now Segmented
- Reordered Row Format applicable to all table space types

|        |                   |                    |                     |
|--------|-------------------|--------------------|---------------------|
| Prefix | Fixed Length Cols | Varchar Indicators | Varying Length Cols |
|--------|-------------------|--------------------|---------------------|

The segmented space structure is more efficient, so adding a universal table space structure for large partitioned table spaces helps DB2 scale.

There are two types of universal table space:

**Partition By Range** A partitioned segmented table space.

Partitioning column required. One table per table space.

Single-table table space, where each partition contains a segmented page set (allows segmented to increase from 64GB to 16TB or 128 TB with 32K pages)

Eliminates need to define partitioning key and assign key ranges

Partitions are added on demand. A new partition is created when a given partition reaches DSSIZE. DSSIZE defaults to 64G. Up to MAXPARTITIONS

Retains benefits of Utilities and SQL parallelism optimizations for partitioned tables. SEGSIZE defaults to 4 & LOCKSIZE defaults to ROW.

Single-table table space Need PBR for query partition elimination

No LOAD PART, ALTER ADD PART, or ROTATE PART All indexes are NPSIs. Automatic repositioning of Variable columns to end of row

Length attributes replaced with indicators positioned after fixed length columns.

**Partition By Growth** A partitioned segmented table space.

No partitioning column required. One table per table space.

## DB2 9 Varchar Performance Improvement

- Remember the tuning recommendation for rows with many columns with any varchar present?



### DB2 9 implements this recommendation **and more**

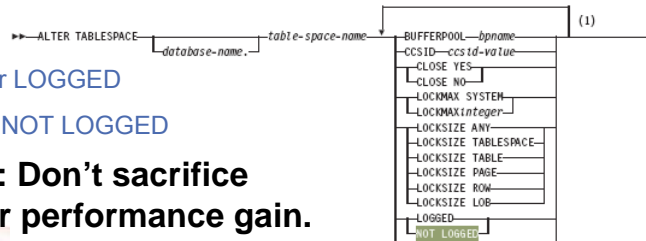
- 2 times or more improvement observed when many rows with many varchars are scanned and/or fetched using many predicates
- <5% improvement for a typical online transaction
- No difference if no varchar
- Reorg rebuilds compression dictionary if varchar columns when migrating to DB2 9 (PK41156)

Improved performance for varying-length rows: If you store a value that is shorter than the length of a column in a varying-length column, the data is not padded to the full length of the column. As a result, columns that follow varying-length columns are at variable offsets in the row. Prior to DB2 9 NFM, when you need to locate and access such a column, you must scan the columns sequentially after the first varying-length column. In DB2 9, the format in which a row that contains varying-length columns is stored in the table has been changed to facilitate locating columns within the row for data retrieval and predicate evaluation. As a result, you no longer need to run a sequential scan, and the performance improves for access to data in tables that store rows with varying-length columns. The improvement is often small, but can be large for many varying length columns in a row.

Impact on log record: log size can be bigger or smaller. Reordered row format (RRF) is not an option. RRF is only for data, not for indexes. When you REORG in NFM, the data is changed to RRF.

## NOT LOGGED table spaces

- Is actually NOT LOGGED tables spaces, tables, indexes, LOB, XML
- ALTER / CREATE a TABLESPACE as NOT LOGGED
  - ▶ ALTER not allowed if in same UOW with an update to the table space
- Indexes, LOB, and XML inherit the logging attribute of the base
  - ▶ These are considered “*Linked*” objects
- Effects the UNDO / REDO records
  - ▶ Control information is still logged
- LOB continue to log system pages & auxiliary indexes
- Unit of Recovery (UR) is still created



LOG YES is a synonym for LOGGED

LOG NO is a synonym for NOT LOGGED

**Recommendation: Don't sacrifice recovery for minor performance gain.**

Cannot explicitly specify for XML & Index objects

LOBs can be set independent of the base table. However, if a LOB is LOGGED, the base must also be logged. This “*dissolves the link*” with the base. Not compatible with CHANGE DATA CAPTURE attribute Applies to any tables in the table space Not allowed for DSNDB06 or Work File database

SYSCOPY activity

ALTER LOGGED to NOT LOGGED creates a **recoverable point**

ALTER NOT LOGGED TO LOGGED marks the object **COPYP** for the base table space

Frequent ALTERing may require more SYSCOPY space

A FULL COPY should be taken

Just before ALTERing to NOT LOGGED

Just after ALTERing to LOGGED

If changes are made while NOT LOGGED

The space is marked ICOPY

An ALTER to LOGGED will set COPYP

Image copies can be SHRLEVEL NONE or REFERENCE Full or incremental

Be careful with any ROLLBACK or CANCEL command that impacts a thread acting on NOT LOGGED objects:

CANCEL, ROLLBACK, LOAD RESUME failures, and Restart Can cause the object (and XML space) to end up in a **RECP** state and in the **LPL**. Indexes often end up in **RBDP** & in the **LPL**

## Application tuning

- Query tuning
- Dynamic to static SQL
- Multirow fetch & insert
- Stored procedures
- REORG Statistics REBIND
- Minimize processing needed

Multirow fetch and insert provide the largest opportunity for improvement, so we'll discuss this topic thoroughly and point to more resources.

Stored procedures can be a performance improvement in some situations and can be a challenge in others.

The process of reorganization, collecting statistics, and then rebinding applications is very important.

Some of the basic rules for minimizing processing need to be followed for the best performance.

Dynamic SQL to static SQL is now a range with many options. Choosing the right ones for your applications can save time.

## Query tuning improved in DB2 9

- Package BIND Stability
- Histogram Statistics
- Page Range Processing
- Global Query Optimization
- Generalized sparse index & in-memory data cache
- Dynamic Index ANDing
- Indexing Enhancements
- Optimization Service Center

See IOD presentations from Patrick Bossman on using OSC and Terry Purcell on optimization improvements in DB2 9 for much more depth on this topic.

Also check the IBM Redbooks publications at the end of this presentation.

## Dynamic SQL ← → Static SQL

- **Reduce dynamic bind frequency via**
  - **Dynamic statement caching with CACHEDYNAMIC YES**
  - **REOPT(ONCE) in V8 REOPT(AUTO) in DB2 9**
  - **Improved monitoring in V8 Visual Explain**
  - **Next step in Optimization Service Center (DB2 9 and V8)**
  
- **Incremental bind in accounting**
  - **Static plan/package with VALIDATE(RUN) and bind time failure**
  - **Static SQL with REOPT(ALWAYS), or referencing Declared Temp Table, or private protocol in requestor**

Once upon a time we could say that dynamic SQL was slow and static SQL was fast. Now the picture is much less black and white, with many shades of grey, often around the REOPT option. We have made dynamic SQL more static with several versions of dynamic statement caching. We have made static SQL more dynamic when a single access path cannot handle all of the needed options.

There is a wide spectrum of SQL, and performance improvements come from making dynamic SQL more static as well as from making static SQL more dynamic. REOPT has more options in V8 and more again in DB2 9. Programming options still matter as well.

For the program that should be bound, static SQL, one common mistake causes extra CPU time. When the package is bound with VALIDATE(RUN) and the object does not exist at BIND time, then incremental BINDs are performed, using extra CPU time. Other incremental bind causes are use of static SQL with REOPT(ALWAYS), referencing a declared temporary table, and distributed private protocol. Monitor the accounting reports for incremental binds to see if they can be avoided.

## JDBC/SQLJ

- Use CACHEDYN YES for JDBC, or better yet use SQLJ or best choice is pureQuery
- Select/Update/Insert required columns only
  - More important in JDBC/SQLJ environment
- Store numeric as smallint or int to minimize conversion and column processing cost
  - Relative cost: Integer (lowest) -> Float -> Char -> Decimal -> Date/Time -> Timestamp (highest)
- Match Java and DB2 data type
  - V8 enhancement for non-matching data type

For applications using dynamic SQL, regardless of the technique, dynamic statement caching tends to make a substantial reduction in CPU use. Some techniques, like use of parameter markers, rather than literals, will help with the dynamic statement cache hit ratio. Sometimes this technique will interfere with the best optimization, since the literal value is not known at run time. Using REOPT(ONCE) then allows the first run time value to be used. REOPT(AUTO) tries to improve upon that technique, reoptimizing when needed.

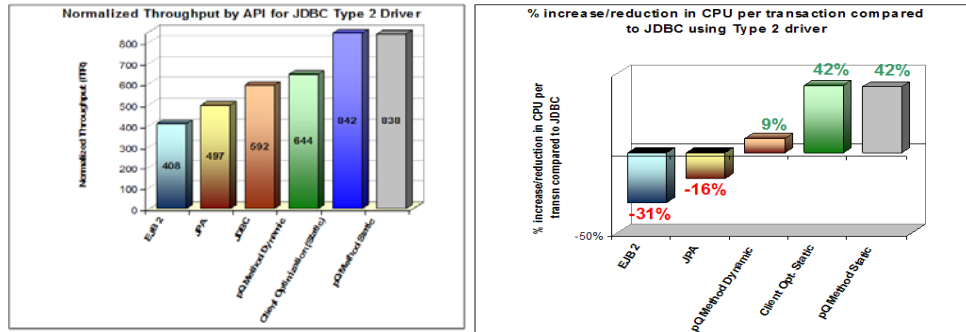
Within the Java world, pureQuery is the recommended approach for the best performing SQL.



## Data Studio pureQuery Runtime for z/OS

- In-house testing shows double-digit reduction in CPU costs over dynamic JDBC

<http://www.ibmdatastatemag.com/story/showArticle.jhtml?articleID=208802229>



- IRWW – an OLTP workload, **Type 2 driver (local call)**
- Cache hit ratio between 70 and 85%
- 42% reduction in CPU per transaction** over dynamic JDBC

See the article, IBM Data Studio pureQuery Runtime for z/OS Performance

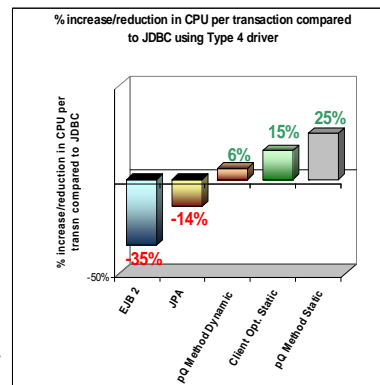
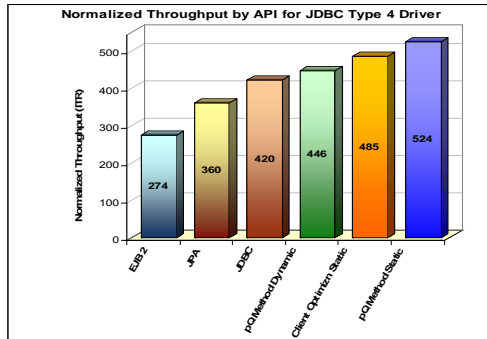
<http://www.ibmdatastatemag.com/story/showArticle.jhtml?articleID=208802229>

pureQuery improves application throughput on DB2 for z/OS by making it easy to code and deploy static SQL for Java. The new client optimization feature in Version 1.2 extends the benefits of static SQL to any existing Java application without changing or recompiling Java applications.

Your performance will probably vary from this laboratory measurement.

## Data Studio pureQuery Runtime for z/OS

- In-house testing shows double-digit reduction in CPU costs over dynamic JDBC  
<http://www.ibmdatabasemag.com/story/showArticle.jhtml?articleID=208802229>



- IRWW – an OLTP workload, **Type 4 driver**
- Cache hit ratio between 70 and 85%
- 15% - 25% reduction on CPU per transaction** over dynamic JDBC

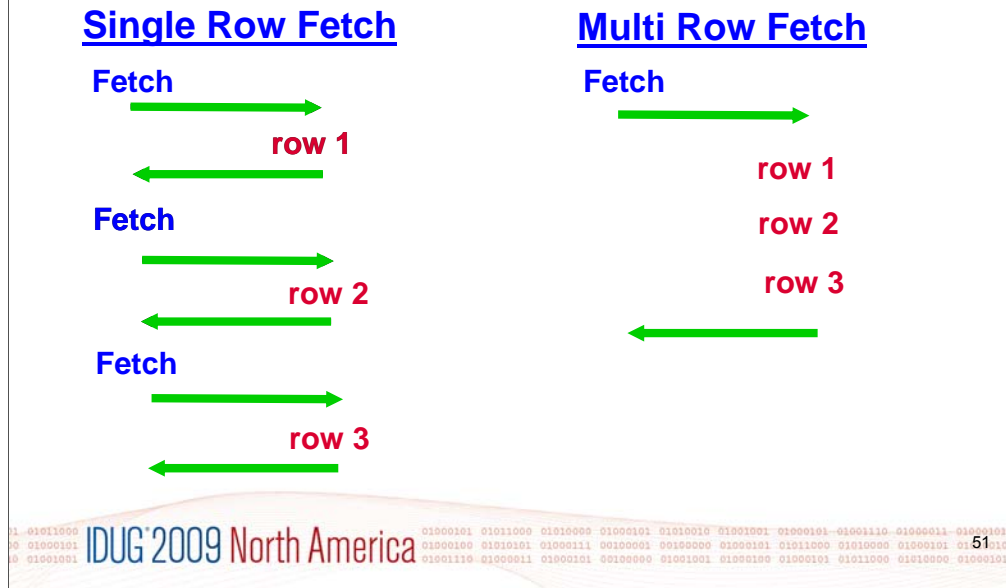
This is also the article, IBM Data Studio pureQuery Runtime for z/OS Performance, showing a type 4 driver, instead of type 2.

<http://www.ibmdatabasemag.com/story/showArticle.jhtml?articleID=208802229>

pureQuery improves application throughput on DB2 for z/OS by making it easy to code and deploy static SQL for Java. The new client optimization feature in Version 1.2 extends the benefits of static SQL to any existing Java application without changing or recompiling Java applications.

Your performance will probably vary from this laboratory measurement.

## Multi-row Fetch



You can enhance the performance of your application programs by using multiple-row FETCH and INSERT statements to request that DB2 send multiple rows of data, at one time, to and from the database. Using these multiple-row statements in local applications results in fewer accesses of the database. Using these multiple-row statements in distributed applications results in fewer network operations and a significant improvement in performance. This figure illustrates the difference between a series of single fetches and a single, multiple-row fetch operation.

For techniques and use with COBOL, see the ftp page

<ftp://ftp.software.ibm.com/software/data/db2/zos/presentations/application-development/>

And get these files. Note that the .wmv files are 25 – 30 MB.

[v8-multi-row-fetch-static-sql-cobol-betaworks-2007.pdf](#)

[v8-multi-row-fetch-static-sql-cobol-betaworks-2007.wmv](#)

[v8-multi-row-fetch-static-sql-cobol-betaworks-2007.zip](#)

[v8-multi-row-fetch-dynamic-sql-cobol-betaworks-2008.pdf](#)

[v8-multi-row-fetch-dynamic-sql-cobol-betaworks-2008.wmv](#)

[v8-multi-row-fetch-dynamic-sql-cobol-betaworks-2008.zip](#)

## Multi-row Fetch - continued

- **FETCH NEXT ROWSET FROM cursor FOR N ROWS INTO hva1, hva2, hva3**
- **Up to 50% CPU time reduction by avoiding API (Application Programming Interface) overhead for each row fetch (100 rows)**
  - ▶ **% improvement lower if more columns and/or fewer rows fetched per call**
    - **Higher improvement if accounting class 2 on, CICS without OTE, many rows, few columns**
  - ▶ **See later foils for distributed**

When using multi-row fetch operations, available in NFM, you normally fetch a set of rows, or a rowset in a single fetch operation, as shown here in the case of 10 rows:

```

FETCH NEXT ROWSET FROM my-cursor FOR 10 ROWS
INTO :hva1, :hva2, :hva3
  
```

Since DB2 now fetches 10 rows in a single Application Program Interface (API) crossing (going from the application to DB2 and back), instead of 10 API crossings, one for each row fetched without rowsets, multi-row fetch reduces the multiple trips between the application and the database engine. This reduction produces even more benefits in distributed applications.

Chapter 3.1 of DB2 V8 Performance Topics discusses the numbers in detail.

## Multi-row Insert

- **INSERT INTO TABLE FOR N ROWS  
VALUES(:hva1,:hva2,...)**
- **Up to 40% CPU time reduction by avoiding API  
overhead for each row insert**
  - ▶ **% improvement lower if more indexes, more  
columns, and/or fewer rows inserted per call**
- **Similar improvement for multi-row cursor  
Update and Delete**

Chapter 3.1 of DB2 V8 Performance Topics discusses the numbers in detail. With 10 rows, the improvement was 35%. Since the insert uses more processing than fetch, the resulting percentage improvement is a little smaller.

See the next slide for a distributed environment, where the improvement is generally larger.

## Multi-row in distributed environment

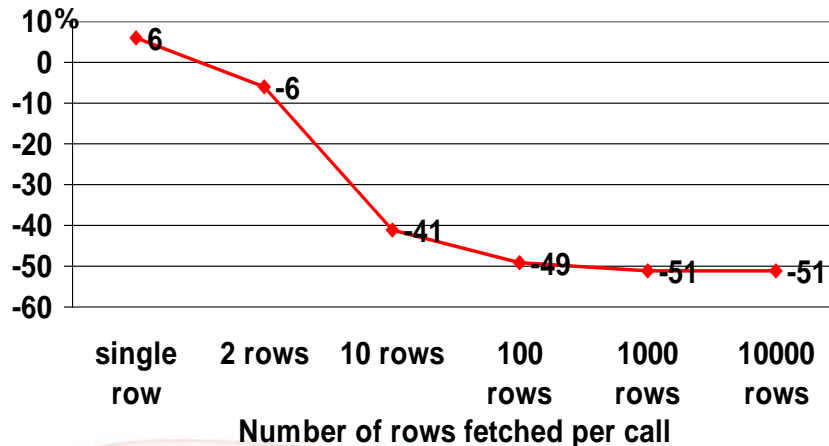
- **Fetch, insert, update & delete**
- **Dramatic reduction in network traffic and response time possible**
  - ▶ **by avoiding message send/receive for each row in**
    - Fetch when not [read-only or (CURRENTDATA NO and ambiguous cursor)]
    - Update and/or Delete with cursor
    - Insert
  - ▶ **Up to 8 times elapsed time reduction observed (up to 4 times CPU time reduction)**

DB2 V8 introduces multi-row fetch and insert. This function is by default enabled for DRDA requests and client application array requests, so the server savings for fetch are provided even without application changes.

The insert improvements do need application changes and can make a huge difference in CPU and elapsed times.

Chapter 7 of DB2 V8 Performance Topics discusses the distributed performance in detail.

## 20 column 100000 row Fetch CPU Time %change in V8 acctg class1 cpu time vs V7



IDUG'2009 North America 55

The graph clearly shows that the percentage improvement goes up as more rows are fetched per Fetch call.

With 1 row fetch, V8 CPU is 6% higher than V7.

However, with 2 row fetch, V8 becomes faster by 6%.

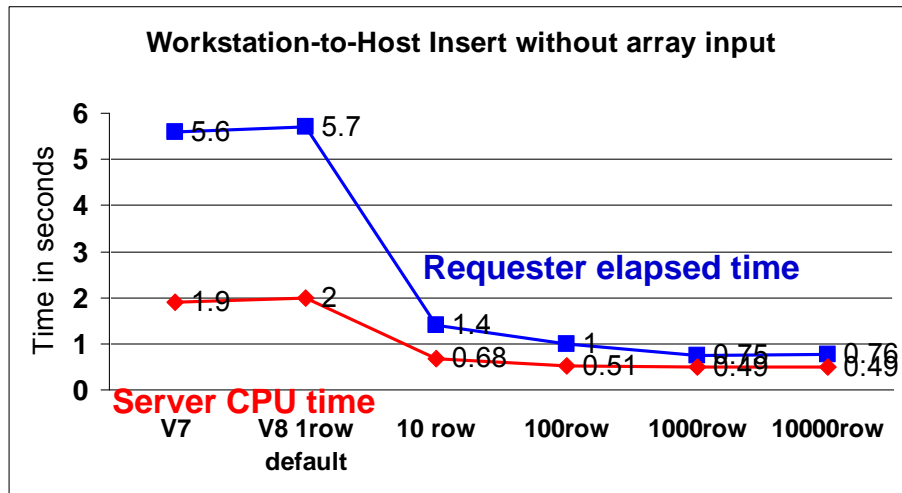
Beyond 100 rows, about 50% improvement continues.

Similarly for elapsed time and class 2 CPU time.

The measurement shown is for a very simple fetch via table space scan fetching 20 columns

Less %improvement for more complex Fetch involving join, sort, index access, more than 20 column fetch

More %improvement for less than 20 column fetch



DB2 for z/OS V8 acting as a DRDA application server,  
 accessed from a DB2 Connect Client running on  
 Linux/Unix/Windows as a DRDA application requestor

10000 20-column rows inserted

10 rows / Insert call

- 76% elapsed time and -63% CPU time compared to V7

- 30% elapsed time and -38% CPU time compared to V7 array input

100 rows / Insert call

- 82% elapsed time and -63% CPU time compared to V7

- 33% elapsed time and -49% CPU time compared to V7 array input



## Distributed/Stored Procedure

- **Stored procedure to avoid DRDA overhead for multiple SQL statements**

- **Example: 10 Select, Insert, Update, and/or Delete calls in stored procedure**

- **Results in 600us instead of 2100us overhead (10 SQL calls \* 210us per SQL call) on z900 (2064-1) processor**

- **Also faster response time because of as low as 1 rather than 10 message send/receive**

Stored procedures can provide a substantial performance improvement when they are used to avoid the DRDA overhead for multiple SQL statements. The example shown changes from 10 distributed SQL statements to a single stored procedure call containing those statements. The result is a substantial reduction in CPU time.

As the number of distributed SQL statements is reduced, the savings are also reduced. For a single SQL statement, the overhead for a stored procedure is very substantial. Stored procedures use WLM processing in another address space, so the overhead is more like a new transaction than a new subroutine call.

See DB2 9 Stored Procedures, SG24-7604,  
<http://www.redbooks.ibm.com/abstracts/SG247604.html>

DB2 V8 Stored Procedures, SG24-7083,  
<http://www.redbooks.ibm.com/abstracts/SG247083.html>

## Minimize SQL Calls to Reduce API Overhead

- **Filter out unnecessary rows by adding predicates rather than by application program checking**
- **Use DB2 column functions rather than application program code**
- **Example: find how many employees make more than \$10,000/month**
  - 1 **Select, fetching all 100000 employee rows**
  - 2 **Select Where Salary>10000, fetching 1000 rows**
  - 3 **Select Count Where ..., fetching 1 row**
    - **100 times CPU time reduction possible from API elimination**
    - **Watch out for VSAM programmers, IO modules (stage 3 predicates)**

Bonnie Baker would call the first bullet using stage 3 predicates. So if your application programmers are using VSAM techniques or generalized IO modules, then you may see an extra 5 x to 10 x CPU time. In the worst cases, the additional CPU time can be 100 times more than using application programming techniques.

Fetch First N Rows Only in V7, in subquery DB2 9

Limits the number of rows fetched to avoid fetching unwanted rows

Singleton Select (or SELECT INTO) can be used with Fetch First 1 Row even if multiple rows qualify. This technique avoids -811 SQLCODE. V8 supports ORDER BY for more meaningful query. Bigger improvement is possible for CICS attach.

UPDATE without cursor is more efficient than OPEN, FETCH, cursor UPDATE, CLOSE. Up to 30% (possibly more if CICS) CPU time saving possible from singleton Select or Update compared to cursor operation

Reducing #SQL calls improves API path length and Processor MIPS for row processing. Up to 2 to 3 times processor MIPS improvement possible from high-speed processor cache hit by repeated execution of a small set of modules / instructions and reduction in data moves.

V8 multi-row operation can significantly reduce the number of SQL calls issued by up to 50% CPU reduction for simple (short-running) local Fetches, more for distributed.

## Minimize Columns and Host Variables Referenced in SQL Calls

- **Avoid unnecessary columns**
  - **Doubled CPU time possible with 100 additional columns/host variables**
  
- **Increasing order of cost**
  - **Local EBCDIC least -> ASCII or UNICODE or DRDA**  
 -> **Single byte conversion -> Double byte conversion**
  - **Integer/char least and date/time/timestamp most expensive**

Avoiding unnecessary columns for both input and output is important for high performance, especially with programs which return millions or billions of rows each night. The CPU cost roughly doubles for each extra 100 additional columns. Beware the SELECT \* used, when only a few columns are needed.

Conversions are costly compared to simple movement. The highest costs are generally for date, time and timestamp.

## Minimize Predicates Evaluated

- Place most filtering predicates **first** in AND. (for predicates of the same type)

|                            |        |
|----------------------------|--------|
| WHERE HOME_STATE='MONTANA' | FF= 1% |
| AND HAIR='BROWN'           | FF=10% |
| AND SEX='MALE'             | FF=50% |

- Weighted average of 1.01 predicates evaluated
- If sequence of predicates is reversed, then the weighted average is 1.55, or 50% more predicate evaluation, which can lead to up to 20% CPU increase.
- Conversely, place most filtering predicates **last** in OR and IN-list without ACESSTYPE=N.  
 eg STATE IN ('NEW YORK','FLORIDA','MONTANA')

Place the most filtering predicates first in ANDed predicates.

Place the most filtering predicates last in OR and IN-list predicates.

These choices are generally less important than the numbers of rows and columns, but the difference in CPU time can be as much as 20%.

## Minimize SQL Statements in a Program Where Possible



```
DO ....  
  SELECT or INSERT or DELETE or UPDATE  
END  
  
      instead of  
SELECT, INSERT, DELETE, or UPDATE  
SELECT, INSERT, DELETE, or UPDATE  
SELECT, INSERT, DELETE, or UPDATE
```

- Reduces EDM pool and thread storage
- Reduces allocate/deallocate cost at SQL execution and commit or deallocation
- Better exploitation of sequential detection and index lookaside
  - Potentially fewer Getpages, Lock requests, and faster I/O
- See Redpaper 4424

IDUG'2009 North America 61

Large numbers of SQL statements in a package and large numbers of packages can increase the CPU time. Redpaper 4424 discusses two situations: when a small number of short-running SQL statements out of many SQL statements in a large package are executed and when a set of short-running SQL statements out of many different packages are executed. These two specific situations have become critical in some environments due to the performance difference after migration from DB2® V7 to V8. DB2 V8 has shown an increase in CPU associated with the major changes in functionalities. We show the measured CPU across V7, V8, and V9, highlight the improvement provided by V9, and provide some general tips on reducing CPU utilization by packages.

## LOB Improvements

- Progressive Streaming for LOB Locator Values
  - DB2 uses LOB size to determine whether to send LOB data to Java or DB2 CLI clients in one go (<32KB), in chunks (<1MB) or as LOB locator (>=1MB)
    - Transparent to application using LOB locators
- FETCH CONTINUE
  - Allows applications to retrieve LOB/XML data in pieces without the use of locators
- File reference variables
  - A file reference variable allows direct transfer of LOB data between DB2 and the file named in the variable
- Utility Changes
  - LOAD / Cross load LOB column lengths > 32KB supported
  - Logging for > 1GB LOBs
  - REORG LOB reclaim space
  - Online CHECK LOB and DATA
- Elimination of LOB locks for improved availability and performance

For Java and DB2 CLI programs that use locators with LOBs. Improves performance and less network traffic for LOBs that are less than 1MB Default behavior if using DB2 9 for z/OS Requires DB2 Connect 9.1 FP 1

No changes required to programs using locator values. DB2 Client and Type-4 driver manage progressive streaming of data to program. DB2 for z/OS determines whether to flow LOB values or Locators to client based on size thresholds for JDBC, SQLJ, and CLI. For small LOBs, (Default <= 32KB) the performance should approximate that of retrieving a VARCHAR column of comparable size. Medium size LOBs (Defaults > 32KB and <= 1MB) For large LOBs (Default over 1MB) locators are still used

Specific FETCH that contains LOB or XML columns used with programs that materialize LOBs. Application uses a buffer that might not be large enough to hold the entire LOB or XML value. If any of the fetched LOB or XML columns do not fit, DB2 returns information about truncated columns and the actual length.

Retrieve LOB or XML data in multiple pieces without use of locators. Must specify WITH CONTINUE on initial FETCH. Subsequent fetches use FETCH CURRENT CONTINUE Application must manage buffers & reassemble data. Not required to fetch entire object before moving to next SQLCA indicates whether data is truncated

LOAD / Cross load LOB column lengths > 32KB supported

Logging for > 1GB LOBs

REORG LOB reclaim space: SHRLEVEL(REFERENCE). Allows LOG NO.

Online CHECK LOB and DATA

Elimination of LOB locks: Now using LRSN & page latching for consistency checks

Prior to DB2 9, LOB locks were held until commit, Even for UR

Space search for LOB allocation: No LOB locks acquired for space search

Read LSN granularity improved to page level in LOB table space

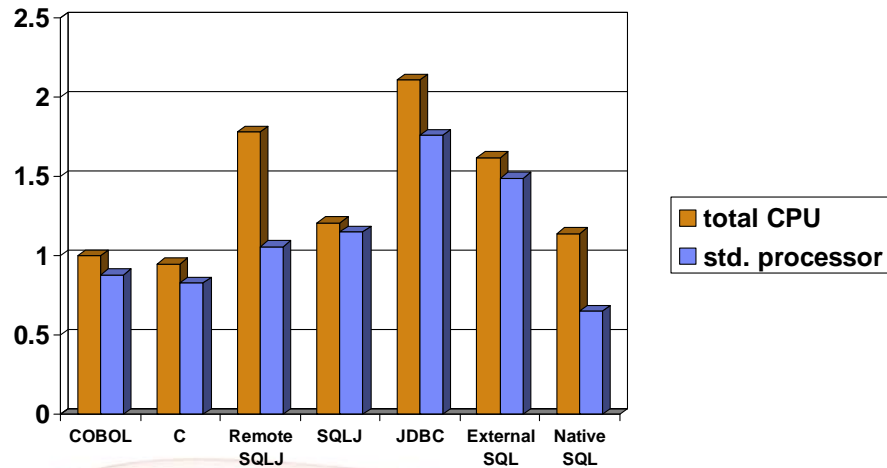
Improved availability & performance, particularly for UR readers

Requirements: NFM "Locking protocol 3" GBP changes Automatic in non-data sharing

Clean group-wide shutdown in data sharing once NFM enabled until PK62027

## Which language should I use?

Stored Procedures – relative cost of various languages with and without zIIP and zAAP for specific OLTP workload



IDUG 2009 North America 63

Many considerations are involved in choosing a language, such as available skills and a fit between the task and the objectives. This chart uses the performance numbers from a presentation on programming language considerations for use in stored procedures by Todd Munk and Gopal Krishnan.

The taller, orange bars show the total CPU used for an example workload, relatively light transactions while comparing various languages. The shorter blue bars show the amount of general purpose CPU used for remote stored procedures, removing the time which is redirected to zIIP and zAAP.

See the numbers and discussion on the next chart.

## Which language should I use? ...

Stored Procedures - Performance of different languages with and without zIIP and zAAP for specific OLTP workload

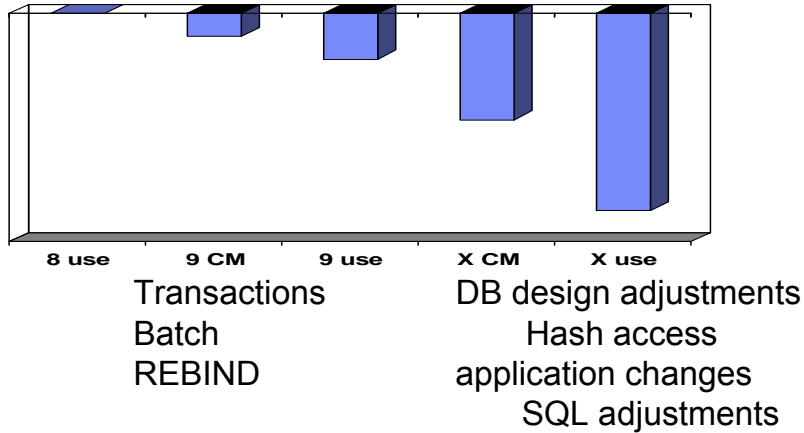
| Language                 | Base CPU  | GP CPU after zIIP and/or zAAP acceleration |
|--------------------------|-----------|--|
| COBOL stored proc        | 1X (BASE) | .88x                                       |
| C stored proc            | .95x      | .83x                                       |
| Remote SQLJ              | 1.78x     | 1.06x                                      |
| SQLJ stored proc         | 1.21x     | 1.15x (zIIP + zAAP)                        |
| JDBC stored proc         | 2.11x     | 1.76x (zIIP + zAAP)                        |
| External SQL stored proc | 1.62x     | 1.49x                                      |
| Native SQL stored proc   | 1.14x     | .65x                                       |

For this specific workload, COBOL and C have the lowest base CPU time. All of the other CPU times are calculated relative to the COBOL time.

When zIIP and zAAP time are removed, the general purpose CPU time is smallest for native SQL stored procedures. COBOL, C and external SQL stored procedures are also reduced by about 12% for the CALL, COMMIT, and stored procedure result set processing. Remote SQLJ is reduced more, and the Java options have reductions for both zIIP and zAAP.



DB2 X preliminary performance plan → significant CPU reductions, best with latest processors



Your situation will vary. Less CPU is better.

Processors z10, z9, z890, z990 & later      z/OS 1.10 & later

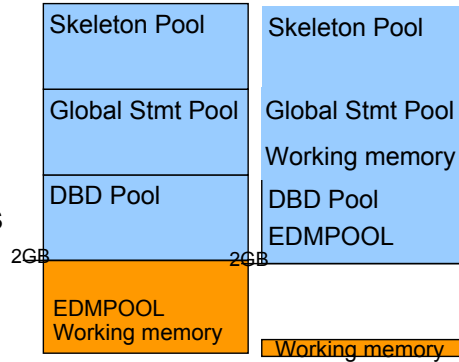
Reducing CPU from DB2 9 to DB2 X without significant administration or application changes is the primary thrust of the performance work.

This work is very preliminary, but the performance plan for DB2 X is much more aggressive than in any recent version. The last version which contained significant improvements for reducing CPU time in transactions and batch was Version 2 in 1988. Versions 3 to 9 made improvements in queries and in utility CPU time and provided many scalability improvements, but little reduction in transaction CPU time. We expect DB2 X to run only on z10, z9, z890, z990, and later processors, and to provide CPU reductions from the beginning, with improvements in CM, but more dramatic reductions for applications that can take advantage of the improvements in application design.

## 64 bit Evolution (Virtual Storage Relief)

**Virtual storage constraint is still an important issue for many DB2 customers.**

- DB2 9 helps (10% – 15%)
- DB2 X expects to move 80% - 90%+
  - ▶ More concurrent work
  - ▶ Reduce need to monitor
  - ▶ Able to consolidate LPARs
  - ▶ Reduced cost
  - ▶ Easier to manage
  - ▶ Easier to grow



The DB2 9 virtual storage objective was 10-15% relief. The DB2 X target is 80% to 90% of the DBM1 address space. We expect the result to be the ability to run much more concurrent work, with an early guess of 3 to 5 times more threads.

Storage monitoring should be drastically reduced. Customers are consolidating LPARs. Sometimes they need to have more than one DB2 subsystem on an LPAR, costing real storage and CPU. With these changes, work can run in one DB2 subsystem, rather than needing more members.

The net for this change is expected to be reduced cost, improved productivity, easier management, and the ability to grow DB2 use much more easily.

## Performance Sources and Resources

RTFW

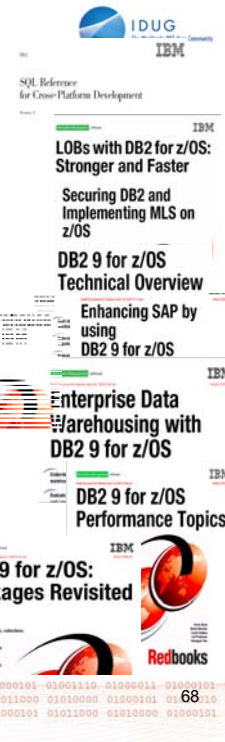
RTFW is the acronym for Read the Friendly Web. Let's take a short walk on the wild, wild web. The problem with the web is not too little information, but rather too much information. The experience is a bit like trying to take a drink from a fire hose. So I'd like to help a bit by narrowing the search with the twenty five cent tour of a few of my favorite DB2 web sites. A lot more information has been added in the past month or two, with many new books and web pages.

Let's start with the DB2 family. I'll generally show the short form or alias of the URL, omitting http://

Here are some tips for avoiding the 404. You don't need www in front of ibm.com in most situations. There is often something after www, such as the -306 in www-306 when you get the URL back from the browser. Remove the hyphen and number when you save the URL, since that number changes more quickly than the rest of the URL.

## DB2 9 in IBM Redbooks Publications

1. DB2 9 Technical Overview SG24-7330
2. DB2 9 Performance Topics SG24-7473
3. DB2 9 Stored Procedures SG24-7604
4. Index Compression with DB2 9 for z/OS redp4345
5. SQL Reference for Cross-Platform Development
6. Enterprise Database Warehouse, SG24-7637
7. 50 TB Data Warehouse on System z, SG24-7674
8. DB2 9 Optimization Service Center SG24-7421
9. LOBs with DB2 for z/OS SG24-7270
10. Deploying SOA Solutions SG24-7663
11. Enhancing SAP - DB2 9 SG24-7239
12. SAP Application on Linux z SG24-6847
13. Best practices SAP BI - DB2 9 SG24-6489-01
14. Data Sharing in a Nutshell, SG24-7322
15. Securing DB2 & MLS z/OS SG24-6480-01
16. Data Sharing: Distributed Load Balancing & Fault Tolerant Configuration redp4449
17. Considerations on Small & Large Packages redp444
18. Backup and Recovery Considerations redp4452
19. Powering SOA with IBM Data Servers SG24-7259
20. Packages Revisited, SG24-7688
21. Data Studio V2.1 Web Services redp4510



DB2 library more information <http://www.ibm.com/software/data/db2/zos/library.html>

Many IBM Redbooks publications, Redpapers and one cross-platform book on DB2 9 are published, in addition to the standard library, with more in the works. Check for updates.

1. DB2 9 Technical Overview, SG24-7330 <http://www.redbooks.ibm.com/abstracts/SG247330.html>
2. DB2 9 Performance Topics, SG24-7473, <http://www.redbooks.ibm.com/abstracts/SG247473.html>
3. DB2 9 Stored Procedures, SG24-7604, <http://www.redbooks.ibm.com/abstracts/SG247604.html>
4. Index Compression DB2 9, REDP4345, <http://www.redbooks.ibm.com/abstracts/redp4345.html>
5. Deploying SOA Solutions SG24-7663, <http://www.redbooks.ibm.com/abstracts/SG247259.html>
6. Cross-Platform Development Version 3, <http://www.ibm.com/developerworks/db2/library/techarticle/0206sqlref/0206sqlref.html>  
[ftp://ftp.software.ibm.com/ps/products/db2/info/xplatsql/pdf/en\\_US/cpsqlrv3.pdf](ftp://ftp.software.ibm.com/ps/products/db2/info/xplatsql/pdf/en_US/cpsqlrv3.pdf)
7. Enterprise Data Warehousing, SG24-7637, <http://www.redbooks.ibm.com/abstracts/sg247637.html>
8. LOBs with DB2 for z/OS SG24-7270, <http://www.redbooks.ibm.com/abstracts/SG247270.html>
9. Securing DB2 & MLS z/OS, SG24-6480-01, <http://www.redbooks.ibm.com/abstracts/sg246480.html>
10. Enhancing SAP, SG24-7239, <http://www.redbooks.ibm.com/abstracts/SG247239.html>
11. Best practices SAP BI, SG24-6489-01, <http://www.redbooks.ibm.com/abstracts/sg246489.html>
12. Optimization Service Center, SG24-7421, <http://www.redbooks.ibm.com/abstracts/sg247421.html>
13. Data Sharing in a Nutshell, SG24-7322, <http://www.redbooks.ibm.com/abstracts/sg247421.html>
14. DB2 9 for z/OS Data Sharing: Distributed Load Balancing and Fault Tolerant Configuration <http://www.redbooks.ibm.com/abstracts/redp4449.html>
15. DB2 for z/OS: Considerations on Small and Large Packages redp4424 <http://www.redbooks.ibm.com/abstracts/redp4424.html>
16. DB2 9 for z/OS: Backup and Recovery Considerations redp4452 <http://www.redbooks.ibm.com/abstracts/redp4452.html>
17. Powering SOA IBM Data Servers, SG24-7259 <http://www.redbooks.ibm.com/abstracts/SG247259.html>
18. DB2 9 for z/OS Packages Revisited, SG24-7688 <http://www.redbooks.ibm.com/abstracts/SG247688.html>
19. 50 TB Data Warehouse Benchmark on IBM System z <http://www.redbooks.ibm.com/redpieces/abstracts/sg247674.html>
20. SAP on DB2 9 for z/OS: Implementing Application Servers on Linux for System z <http://www.redbooks.ibm.com/redpieces/abstracts/sg246847.html>
21. IBM Data Studio V2.1: Getting Started with Web Services on DB2 for z/OS <http://www.redbooks.ibm.com/redpieces/abstracts/redp4510.html>
22. Parallel Sysplex Operational Scenarios <http://www.redbooks.ibm.com/redpieces/abstracts/sg242079.html>

zIIP [ibm.com/systems/z/ziip/](http://ibm.com/systems/z/ziip/) II14219 II12836 II10817

Home Products Services & solutions Support & downloads My account

Servers > Mainframe servers > Hardware >

Mainframe servers  
About IBM mainframe  
Hardware  
Software  
Operating systems  
Solutions  
Customer successes  
Literature  
FAQs  
Support  
Education  
News and events

zIIP – Overview

zIIP Getting started  
Overview About Resources

Today's corporate data  
Data is at the core of every business. But data management can be a big project. It needs to be available globally; it must be kept up in real time, and accessible 24x7. At the same time, it needs to be protected from unauthorized access and to comply with new regulations. The IBM System z™ mainframe family has a strong heritage in data serving, and its architecture is designed for massive data access, whether across the Internet, to storage devices, or to remote backup sites. Scalability, availability, security—these are all core competencies of the mainframe.

Learn More  
→ What they are saying about zIIP  
→ Data Serving Press release  
→ Data Serving Roadmap z990  
→ z/OS website  
→ DB2 V9 for z/OS preview  
→ System z security

We're here to help  
IBM Mainframes just got even more cost competitive. Let us show you how with a customized quote.  
Request a quote  
or call us at 1-888-SHOP-IBM

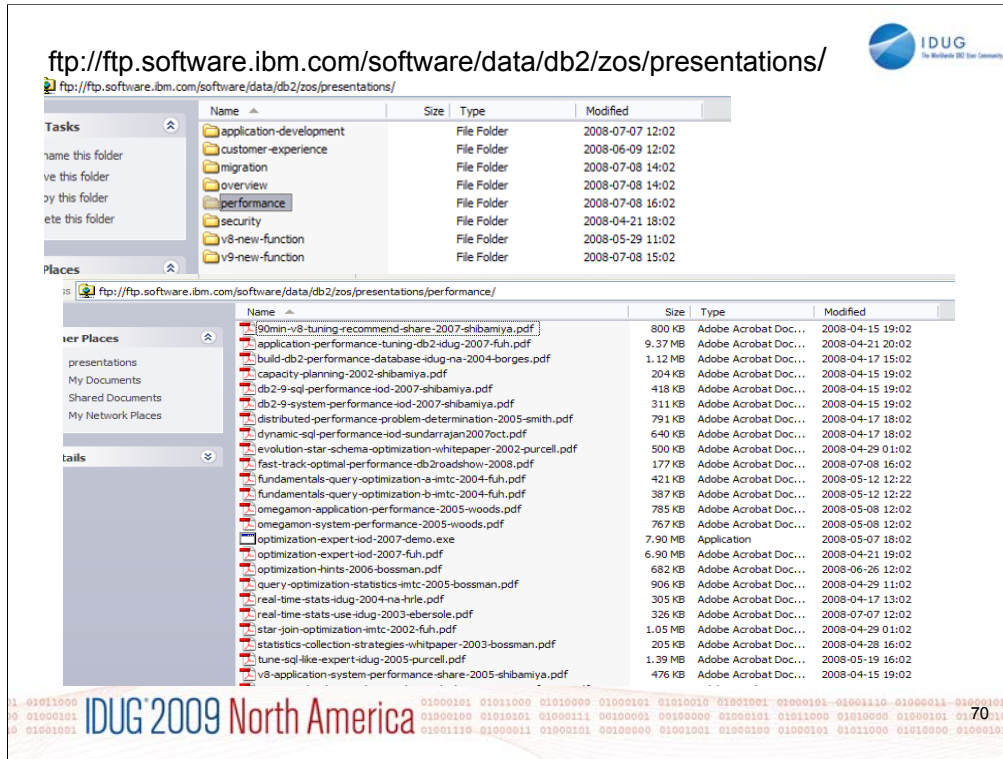
Exciting new features  
Earlier this year, IBM extended its mainframes data serving capabilities, delivering a new roadmap for the future of data serving and information on demand, previewing new DB2 function, and introducing a new specialty engine directed toward data serving workloads.  
The new specialty engine, the IBM System z9 Integrated Information Processor (IBM zIIP), is now available on the System z9 Enterprise Class (EC) and System z9 Business Class (BC) servers.

IDUG 2009 North America

See zIIP information on the web. This web site has the most current information, & pointers to more resources. The zIIP is for customers who are concerned about costs for growth. One big cost reduction is hardware cost, which is much less than a standard processor. The biggest cost reductions are in software, as IBM does not charge for software running on the specialty processors. The zIIP fits some customers very well, but will not apply for all. As a specialty processor, not all work can use the zIIP, which will only process work running under an enclave SRB. Most applications cannot run in SRB mode. The specifics of the software charging need to be considered. Customers must be current on hardware (System z9), current on software (z/OS 1.6 or later, DB2 V8 or later) and have a work load peak using the types of work eligible for zIIP:

Remote SQL processing of DRDA network-connected applications over TCP/IP: These DRDA applications include ERP (e.g. SAP or PeopleSoft), CRM (Siebel), and business intelligence running on other platforms. Remote SQL is expected to provide the primary benefits to customers, as it is commonly part of the peak load. Stored procedures and UDFs run under TCBS, so they are not generally eligible for zIIP, except for the call, commit and result set processing. DB2 9 remote native SQL Procedure Language is eligible for zIIP . Parallel queries: If the work comes in remotely over DRDA using TCP/IP, then the initial work is eligible as remote work. After the initial time, the parallel processing threads are eligible and can process on the zIIP. DB2 utility index processing: Functions of the LOAD, REORG and REBUILD DB2 utilities that perform index maintenance are eligible for zIIP. This is not a common peak capacity constraint, but could be useful in reducing CPU charges.

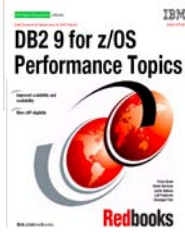

The best way to estimate the eligible work is to apply the needed z/OS and DB2 service, to run your work load and to take measurements. Use DB2 accounting with any product which can provide DB2 accounting reports, such as Omegamon.



The ftp server has moved and has been reorganized. Now, if you use the url above, you'll see categories for new function in V8, new function in DB2 9, application development, customer experience, migration, performance, security and overview (none of the above). If you move to one of the topics (DB2 9 new function in this example), you can check the individual presentations. Many have slides in pdfs and audio with the slides in .wmv files.

Get updated books February 2009  
<http://publib.boulder.ibm.com/infocenter/imzic/>

- ✓ Performance Guide
- ✓ Redbooks
- ✓ Administration Guide
- ✓ Data Sharing: Planning and Administration
- ✓ Utility Guide and Reference
- ✓ Installation Guide

IDUG '2009 North America

IDUG The Worldwide DB2 User Community

IBM

DB2 9 for z/OS Performance Topics

Redbooks

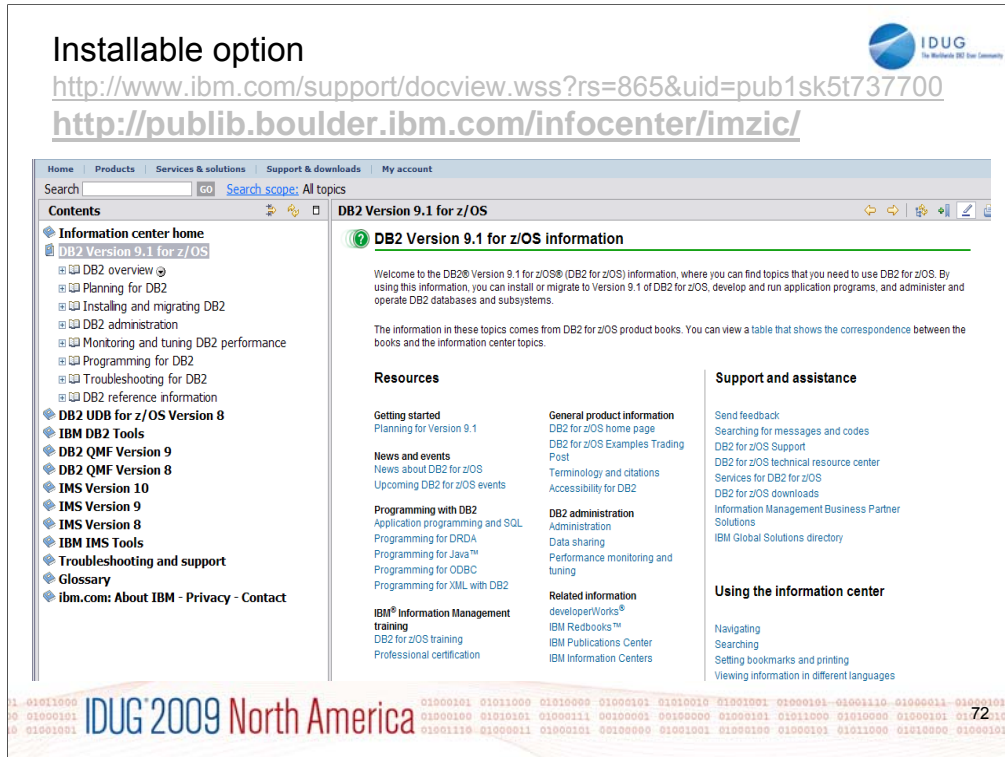
For performance, you need many books. Some are optional, for example the data sharing book is not needed if you don't use data sharing. You can get most of the books from the DB2 Technical References web page. The books were updated in December 2007 and February, March, June, and August 2008, with some coming later, so get the latest ones. Some of the Redbooks are important. You need books from the z/OS Library as well.

<http://www.ibm.com/support/docview.wss?rs=64&uid=swg27011656>

<http://www.ibm.com/support/docview.wss?rs=64&uid=swg27011658>

<http://www.ibm.com/systems/z/os/zos/bkserv/r9pdf/>

Be sure to use the latest information to save time and problems. Some of the IBM Redbooks publications have always been updated and added lately (next page).



This is the Information Center, with a wide spectrum of information and access to books for DB2 for z/OS, DB2 tools, QMF, IMS, IMS tools and more. You can get to this page from the Library page, by clicking Information Center. The Information Center provides information across the books and across multiple products.

If you click “Troubleshooting and Support”, then expand under “Searching knowledge base” and click “Web search:...”, you’ll find a helpful Web search page. From this page, you can search IBM support, DeveloperWorks, or even the whole Internet using Google.

The latest change in this area is an installable Information Center, so that you can use the facility even when the internet is not accessible.



This is the main DB2 for z/OS web page. You can get to the other DB2 for z/OS pages from here, so I often call this my home page. This page changes frequently, so look at the highlighted NEW items. Do you want to look in a DB2 book? Click on Library to see books on DB2 and QMF Version 8 (about 40), Version 7, 6 or even 5. V6 and V5 are out of service. You can check the latest changes by looking at the Information Updates or go to the Information Center. From this page, you can look for conferences (Events), specific classes (Education), or services. If you want to see the latest on DB2 9 or DB2 Version 8, click on the DB2 9 or the V8 link. If your primary concern is application development, the Developer Domain is for you. DB2 Magazine covers a broad range of topics about DB2. The latest machines System z9, z990 and z890 are on the System z page. Click DB2 and IMS Tools to see the wide range of help we provide.

**Data Studio**  
<http://www.ibm.com/software/data/studio/>

United States [ change ]

Home Solutions Services Products Support & downloads My IBM

Welcome Mr. Roger Miller [Not you?] [ IBM Sign in ]

Software > Information Management >

**IBM Data Studio**

The comprehensive data management solution to design, develop, deploy, and manage data-driven applications.

"With IBM Data Studio, we can discover faulty assumptions and get resolution from business sponsors early in the design process. This will help us reduce development costs as it's 50 to 75 percent more expensive to fix a design problem after we implement the changes"

Kevin Campbell, Application Architect, Univar USA

**Why IBM Data Studio**

IBM Data Studio is an Integrated Data Management Environment. Learn how IBM Data Studio can increase productivity and reduce development cost throughout the data lifecycle.

We're here to help

Easy ways to get the answers you need.

Call me now

E-mail us

or call us at  
**877-426-3774**  
 Priority code:  
**104CBW60**

Related links

- DB2 9
- Informix
- U2
- IMS
- IBM Balanced Warehouse
- DB2 for z/OS

Final! Interested

IDUG 2009 North America 74

See this page for all the changes in IBM Data Studio. Watch closely, as this area is changing fast. IBM Data Studio is an Integrated Data Management Environment. Learn how IBM Data Studio can increase productivity and reduce development cost throughout the data lifecycle.

IBM Data Studio is an Integrated Data Management Environment. Learn how IBM Data Studio can increase productivity and reduce development cost throughout the data lifecycle.

IBM Data Studio Developer: An Integrated Development Environment for creating and testing database and pureQuery applications.

IBM Data Studio pureQuery Runtime: A high-performance Java data access platform -- improves security and manageability of Java application connections to databases.

<http://www.ibm.com/software/data/studio/>

**DB2 Connect**  
 ibm.com/software/data/db2/db2connect/

**DB2 Connect**

**INTRODUCING DB2 9.5**  
 → Learn more

**DB2 Connect** makes your company's host data directly available to your personal computer and LAN-based workstations. It connects desktop and palm-top applications to your company's mainframe and minicomputer host databases for access to your enterprise information no matter where it is. **DB2 Connect** provides the application enablement and robust, highly scalable communication infrastructure for connecting Web, Windows, UNIX, Linux and mobile applications to z/OS and AS/400 data. **DB2 Connect** is included in many of the DB2 products.

**Information Management**

**We're here to help**

Easy ways to get the answers you need.

**Call me now**

**E-mail us**

or call us at  
**877-426-3774**  
 Priority code:  
**104CBW60**

**Related links**

→ What's new in DB2 Connect V9

IDUG'2009 North America 75

**DB2 Connect** makes your company's host data directly available to your personal computer and LAN-based workstations. It connects desktop and palm-top applications to your company's mainframe and minicomputer host databases for access to your enterprise information no matter where it is. **DB2 Connect** provides the application enablement and robust, highly scalable communication infrastructure for connecting Web, Windows, UNIX, Linux and mobile applications to z/OS and AS/400 data. **DB2 Connect** is included in many of the DB2 products.

<http://www.ibm.com/software/data/db2/db2connect/>

## Important Disclaimer

**THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.**

**WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.**

**IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.**

**IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.**

**NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:**

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

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.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in the operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information may contain 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.

**IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.**

**TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/400, Netfinity, RISC, RISC SYSTEM/6000, System i, System p, System x, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, System z,**

**The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, ODBC**

Session: A12



DB2 for z/OS Selected Performance Topics

# Questions?

Roger Miller

IBM Silicon Valley Lab

millerrl@us.ibm.com



IDUG'2009 North America 77

Thanks for coming.

Title – DB2 for z/OS technical evangelist, strategist, architect, designer, developer, writer, service, DB2 factotum (from the Latin for does everything or jack of all DB2 trades, master of several).

Current Projects - Roger is working to roll out DB2 9 for z/OS, to design the next improvements in DB2.

Technical accomplishments - Roger Miller is a DB2 for z/OS technical evangelist, architect and designer who worked on many facets of DB2, ranging from overall design issues to SQL, languages, install, security, audit, standards, performance, concurrency, and availability. He has worked for 30 years on DB2 development, product design and strategy. He often helps customers to use the product, answers many questions and presents frequently to user groups.

Fun facts - Roger likes hiking, bicycling, reading, Shakespeare, Yosemite and bears. He has learned to like working out with a personal trainer.