May 6–10, 2007

San Jose Convention Center

San Jose, California, USA

Session: E11

A Sneak Peak at the Next Release of DB2 LUW : Part 2

(For Administrators)

IDUG® 2007 North America

Matt Huras
Aamer Sachedina
IBM

May 9, 2007 3:00 p.m. – 4:00 a.m.

Platforms: Linux, Unix, Windows

GoFurther

Obligatory Fine Print

The information in this presentation concerns new products and/or features that IBM may or **may not** announce or deliver. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The specification and/or details of the content described in this presentation **may change**.

REFERENCES IN THIS PUBLICATION TO IBM PRODUCTS, PROGRAMS, OR SERVICES DO NOT IMPLY THAT IBM INTENDS TO MAKE THESE AVAILABLE IN ALL COUNTRIES IN WHICH IBM OPERATES.

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.

SOME OF THE MATERIAL MAY REPRESENT IBM CONFIDENTIAL INTELLECTUAL PROPERTY WHICH IS NOT TO BE DIVULGED TO ANYONE WITHOUT PRIOR CONSENT FROM IBM.

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted Context
  - Auditing enhancements
- Unicode Enhancements

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted Context
  - Auditing enhancements
- Unicode Enhancements

Motivation: Integrated HA and DR Solution

- Automated HA and DR setup requires a cluster manager (TSA, Vertias).
- Specialized cluster management skills are required to do this.
- Regardless of how many scripts we provide in “samples”, in general, this some level of customization for each installation
- **Goal: Integrate automatic DB2 HA and DR setup with third party cluster managers.**

5

GoFurther

Fully integrated cluster manager into DB2.

Will be shipping Tivoli System Automation with DB2 on AIX and Linux – free 2 node TSA license included with DB2

Designed a “generic” interface that can be used to “plug-in” any cluster manager. We will provide plug-ins for IBM cluster managers and Veritas has one for VCS

Eventually replace “generic” interface with industry standard which we are driving force behind.

Once the cluster is setup, DB2 will maintain it when nodes are added/dropped or tablespaces added/dropped, …

From a purely HADR perspective, there is really no difference between replicating between machines which are physically next to one another as compared to replicating across machines separated by many 1000's of miles. This statement is arguably true for TSA as well, however, there are issues which in practise can arise which must be understood first. Most of these issues arise with the mechanism used to determine quorum. (Quorum is used by cluster managers to determine which node or set of nodes should own resources should various network or node failures occur). With TSA 2.1, there are three common methods for quorum determination: 1) shared disk (SDTB) 2) majority nodes tiebreaker (MNTB) 3) network tiebreaker (NetTB). Let's consider each of these quorum algorithms and the implications for a distributed environment.

For the NetTB to work; the following condition is required: IF sub domain A can communicate (ping) with the target and sub domain B can communicate (ping) with

Integrated HA and DR Solution

- **HA Cluster Manager Integration**
  - Coupling of DB2 and TSA on Linux and AIX, other platforms coming later.
  - DB2 interface to configure cluster.
  - DB2 to maintain cluster configuration, add node, add tablespace.
  - Exploitation of new vendor independent layering (VIL), providing support for any cluster manager.
  - Eventually replace VIL with industry standard, e.g. SAF?
- **NO SCRIPTING REQUIRED!**
  - One set of embedded scripts that are used by all cluster managers

Fully integrated cluster manager into DB2.

Will be shipping Tivoli System Automation with DB2 on AIX and Linux – free 2 node TSA license included with DB2

Designed a “generic” interface that can be used to “plug-in” any cluster manager. We will provide plug-ins for IBM cluster managers and Veritas has one for VCS

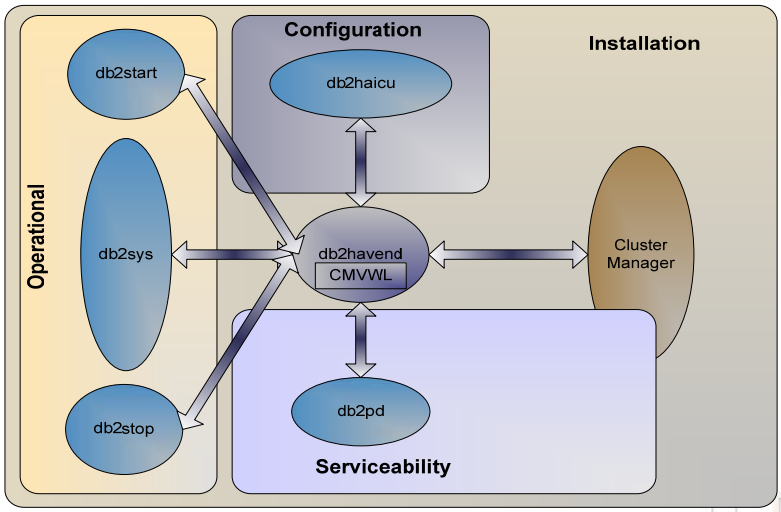
Eventually replace “generic” interface with industry standard which we are driving force behind.

Once the cluster is setup, DB2 will maintain it when nodes are added/dropped or tablespaces added/dropped, …

From a purely HADR perspective, there is really no difference between replicating between machines which are physically next to one another as compared to replicating across machines separated by many 1000's of miles. This statement is arguably true for TSA as well, however, there are issues which in practise can arise which must be understood first. Most of these issues arise with the mechanism used to determine quorum. (Quorum is used by cluster managers to determine which node or set of nodes should own resources should various network or node failures occur). With TSA 2.1, there are three common methods for quorum determination: 1) shared disk (SDTB) 2) majority nodes tiebreaker (MNTB) 3) network tiebreaker (NetTB). Let's consider each of these quorum algorithms and the implications for a distributed environment.

For the NetTB to work; the following condition is required: IF sub domain A can communicate (ping) with the target and sub domain B can communicate (ping) with

Integrated HA and DR Architecture



7

The DB2 high availability feature provides a tighter coupling between cluster managers and DB2 UDB along multiple dimensions:

Installation: TSA is bundled and installed with DB2 as the default cluster manager on Linux and AIX.

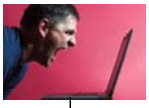
Configuration: The DB2 high availability feature includes a configuration utility to setup DB2 with cluster managers

Operation: DB2 high availability feature automatically updates the cluster manager when there are DB2 state changes significant to the cluster.

Serviceability: TSA FixPaks are installed as part of DB2 FixPaks.

DB2 interacts with cluster managers through a set of APIs externalized in this solution. These APIs can be implemented by cluster manager vendors or any other third-party and are represented by the cluster manager vendor wrapper library (CMVWL) in the figure above. The DB2 high availability feature includes wrappers for TSA and HACMP.

Clustering Setup Pre-Viper II



Overworked admin doing initial setup

Install cluster SW

Install DB2

Create DB2 instance & database

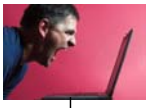
Add nodes / partitions to DB2 instance

Create cluster domain

Add each DB2 node's host to cluster domain

Design and create 'resource models'

Take multiple courses on cluster management



Overworked admin adding a new file system for DB2 (tablespace container or storage path)

Add container or storage path to DB2 (ALTER)

Remember to modify cluster resource model to account for new file system

Refer to cluster management course material to figure out how to do this

BRING OUT THE CONSULTANTS!

Clustering Setup Post-Viper II

Relaxed admin doing initial setup

- Install Viper II
- Create DB2 instance & database
- Run DB2 HA config. tool – db2haicu
- Add nodes / partitions to DB2 instance

Relaxed admin adding a new file system for DB2 (tablespace container or storage path)

- Add container or storage path to DB2 (ALTER)

Integrated HA and DR Solution

- **Automate HADR failover**
  - Exploit HA cluster manager integration previously described.
  - Exploit arbitrator instead of shared disk, as this is really for DR.
    - Both systems must have connectivity to arbitrator.

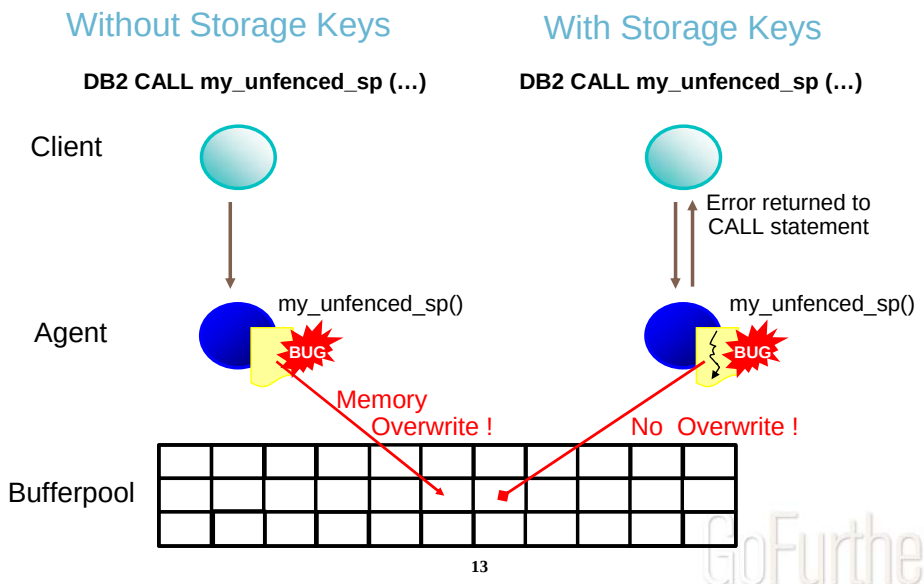
Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware memory protection exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted Context
  - Auditing enhancements
- Unicode Enhancements

Hardware Memory Protection Exploitation

- Built-in machine instructions to allow/disallow access to specified memory region(s)
- Execute very quickly
  - Can invoke frequently without significant performance penalty
- DB2 Viper II exploits this capability
- Benefits:
  - Increased toleration to bugs in unfenced stored procedures
  - General increase in memory resiliency

Hardware Memory Protection Exploitation



Hints / Tips / Notes

- Enable via: DB2_MEMORY_PROTECT=YES
- Does not protect against all types of bugs in user-written code
  - Eg. code which erroneously closes the wrong file descriptor
  - Not a license to be less careful about making user routines unfenced

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted Context
  - Auditing enhancements
- Unicode Enhancements

Serviceability Improvements: First Occurrence Data Capture (FODC)

- FODC is a new facility to capture information on the first occurrence of a problem.
- A “package” of information is created when a problem occurs on the DB2 server.
- FODC package provided by customer to DB2 support.
- db2support tool still used to collect the package.
- Helps maximize the chances that DB2 support will be able to help resolve the problem after the first occurrence.

First Occurrence Data Capture

- Automatic FODC
  - A FODC package is automatically generated in the following cases:
    - Traps
    - Panics
    - Physical data corruption detected on disk (e.g. CBIT errors, bad page)
    - Database marked bad.
- Manual FODC
  - Administrators can manually cause an FODC package to be generated by invoking the “db2fodc” tool if they observe a hang in the DB2 server.

```
db2fodc -hang
```

First Occurrence Data Capture

- **FODC package dumped in a subdirectory of DIAGPATH**
  - FODC_<OutageType>_<Timestamp>
  - Outage types:
    - **Trap, Panic, DataCor, DBMarkedBad**
  - **eg. FODC_Trap_2007-02-25-22.45.12.054138**
- **Contents of FODC package:**
  - Trap files
  - Binary dump files
  - Output of the symptom-specific call-out scripts
  - Memory reports
  - BSU event logs
  - Component-specific diagnostic files
  - Core files

First Occurrence Data Capture

- New registry variable “DB2FODC” controls FODC behaviour

DB2FODC sub-option	Description	Default
DUMPCORE: ON | OFF	Core file generation	ON
DUMPDIR: path to directory	Specifies absolute pathname of the directory where core file or shared memory dump will be created.	Default diagnostic directory
CORELIMIT:size	The maximum size of core files created.	?
DUMPSHMEM: ON | OFF	Shared memory dump during outage	OFF
MEMSCAN: ON | OFF	Memory scan on outage	OFF

FODC: Hints / Tips / Notes

- Set DUMPDIR to be a separate file system.
  - Shared memory dumps and core files can consume A LOT of space and you don't want them filling up sqllib.
- This maximizes the chances that the FODC package will allow DB2 support to resolve the problem on the first occurrence.

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted context
  - Auditing enhancements
- Unicode Enhancements

Viper II Data Redistribution – Key Benefits

- Substantial improvement in performance over previous releases.
  - Anywhere from 3x to 5x end to end!
- Allows user to specify subset of tables and order of tables to be redistributed.
- Inline space compaction and statistics collection can eliminate need for reorg and runstats after redistribute.
- Incremental and rebuild options for index maintenance without additional table scan.
- Substantially improved monitoring.
- Ability to set a “STOP AT” time for the end of a batch window and continue the reorg in the next batch window.

Pre Viper II, incremental was the only option for rebuild. Although, customers probably frequently dropped indexes prior to redistribute.

Viper II Data Redistribution - Details

- Redesigned from the ground up to improve performance.
- Sending partitions send multiple records in a batch transfer buffer instead of row by row inserts.
- Receiving partitions modify table by formatting pages at the 'data manager' level and write them directly to disk via. efficient big block I/O.
- Inter and intra table parallelism.
- Minimal logging – recovery through a 'consistency point' algorithm.

Data Redistribution Table Specification

- New “TABLE” option added to command.
- List of tables to be redistributed can be provided on command.
- Only a subset of tables can be redistributed if desired.
- Table order can be specified in command.

```
Example 1A: Redistribute only tables tab1, tab2, tab3:
   DB2 REDISTRIBUTE DATABASE PARTITION GROUP MYDBPAGRP TABLE (tab1, tab2,
   tab3) ONLY

Example 1B: Redistribute the rest of the tables:
   DB2 REDISTRIBUTE DATABASE PARTITION GROUP MYDBPAGRP CONTINUE

Example 2: Redistribute tab2 first and then rest of the tables in any
   order
   DB2 REDISTRIBUTE DATABASE PARTITION GROUP MYDBPAGRP TABLE (tab2) FIRST
```

Data Redistribution Parallelism Specification

- Intra-table parallelism is always present.
  - Table parts (DAT, INX, LOB/LF) redistributed by different threads.
- Inter-table parallelism is specifiable as an option.
  - Default inter-table parallelism is determined automatically by DB2 if not specified.

```
DB2 REDISTRIBUTE DATABASE PARTITION GROUP
  –db-partition-group- PARALLEL TABLE n
```

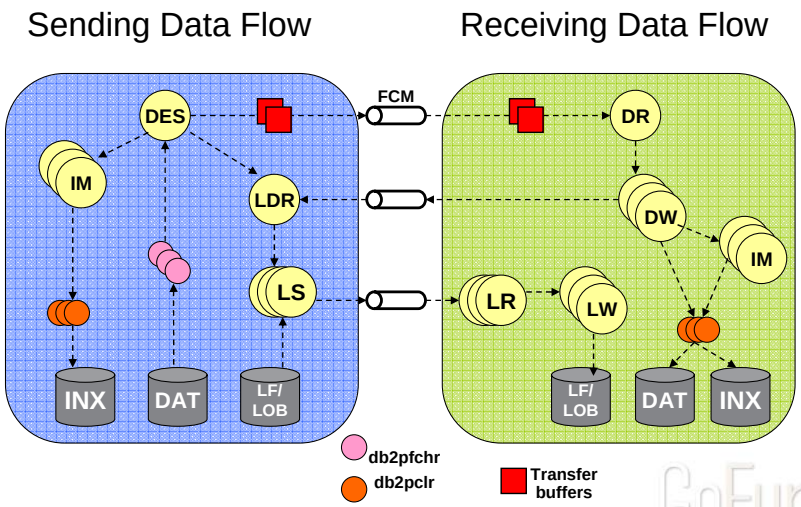

All operations on a given table are done within a single transaction. While a table is being redistributed, the corresponding table space for the table will be put into the "backup pending" state. All the tables in that table space will become read-only until the tablespace is backed-up, which can only be done when all tables in the tablespace have finished being redistributed.

Data Redistribution Indexing Mode Specification

- Specifies how indexes are maintained during redistribute.
- Default is AUTOSELECT.
  - DB2 will pick between REBUILD and INCREMENTAL based on the amount of data being moved.

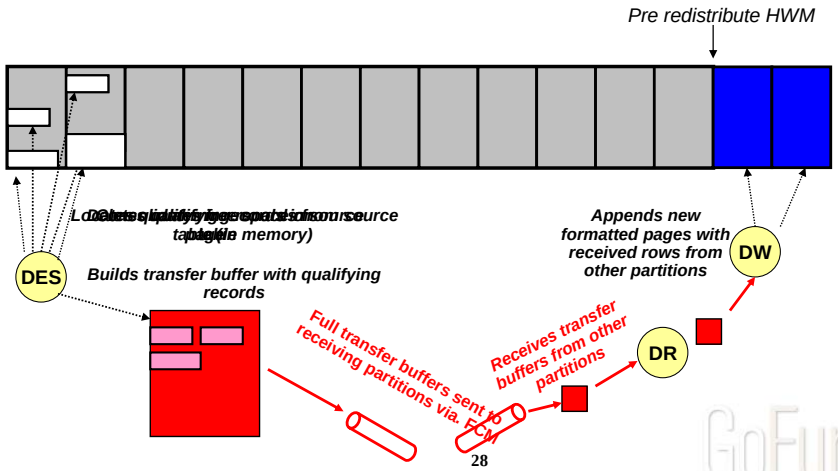
```
DB2 REDISTRIBUTE DATABASE PARTITION GROUP
  –db-partition-group- INDEXING MODE
  AUTOSELECT | INCREMENTAL | REBUILD |
  DEFERRED
```

Data Redistribution Flow



27

Data Redistribution Processing – Part 1



End of redistribute has a gap filling phase that fills up empty pages left by the DES with pages at the new end of the table

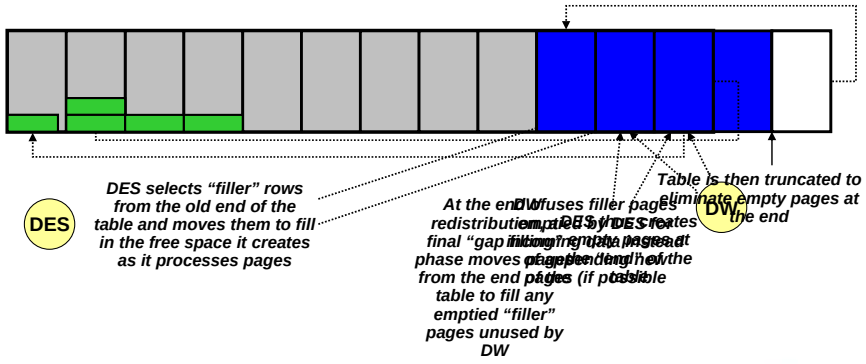
Table Compaction (COMPACT option)

- Table compaction is ON by default.
- PCTFREE is maintained.
- Effectively a space reclaiming table REORG as part of the redistribute.
- Moves ‘filler’ rows from original end of table, cell or range partition to fill free space in page.
- Table is truncated (if necessary) after a gap filling phase at the end of redistribute.

29

On the sending database partitions, Redistribute will fill holes on data pages as records are being redistributed to target database partitions. For regular tables, records will be taken from the logical end of the table and the table will be truncated at the end to free up spaces. For MDC tables, records will be taken within the same cell for holes filling and any empty blocks will be freed up for reuse. For Range Partitioned tables, records will be taken from the logical end of a range and truncation will take place on every range to free up spaces. The default value is ON.

Data Redistribution Processing – Part 2 (Table Compaction)



End of redistribute has a gap filling phase that fills up empty pages left by the DES with pages at the new end of the table.

Data Redistribute – STOP AT Specification

- When specified, redistribute will check the time before beginning work on each table.
- If STOP AT time is reached before tables are completed, command will complete with a warning message.
- Redistribute can be reissued with the CONTINUE option.

```
DB2 REDISTRIBUTE DATABASE PARTITION GROUP
  –db-partition-group- STOP AT yyyy-mm-dd-
  hh.mm.ss.nnnnnn
```


When this option is specified, Redistribute will compare the *local-isotime* with the local current timestamp before start working on every table. If the specified *local-isotime* is equal to or smaller than local current timestamp, Redistribute will stop with a warning message. As a result, all the previously done tables will be using the rebalancing distribution map and the remaining tables will continue to use the distribution map defined in the database partition group. The remaining tables can be redistributed using the CONTINUE option. This *local-isotime* value is specified as a time stamp, a 7-part character string that identifies a combined date and time. The format is yyyy-mm-dd-hh.mm.ss.nnnnnn (year, month, day, hour, minutes, seconds, microseconds) expressed in local time.

Data Redistribute – STATISTICS Specification

- This option specifies that redistribute should collect statistics for the tables that have a statistics profile.
- Specifying this option is more efficient than separately issuing the RUNSTATS command after the data redistribution is completed.

```
DB2 REDISTRIBUTE DATABASE PARTITION GROUP
  –db-partition-group- STATISTICS USE
  PROFILE | NONE
```

32

By default, the redistribute utility will update the statistics for those tables that have a statistics profile. For the tables without a statistics profile, it is recommended that you separately update the table and index statistics for these tables by calling the db2Runstats API or by issuing the RUNSTATS command after the redistribute operation has completed.

This option specifies that redistribute should collect statistics for the tables that have a statistics profile. Specifying this option is more efficient than separately issuing the RUNSTATS command after the data redistribution is completed.

USE PROFILE

Statistics will be collected for the tables with a statistics profile. For tables without a statistics profile, nothing will be done. This is the default.

NONE

Statistics will not be collected for tables.

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted context
  - Auditing enhancements
- Unicode Enhancements

Database Roles

- What is a “Role”?
  - Database object that groups together one or more privileges or authorities and can be assigned to users, groups, PUBLIC or to any other roles using a GRANT statement.
  - Can be assigned to a trusted context by using CREATE TRUSTED CONTEXT or ALTER TRUSTED CONTEXT.
- Grantable privileges and authorities.
  - All database privileges and authorities except SECADM.

By default, the redistribute utility will update the statistics for those tables that have a statistics profile. For the tables without a statistics profile, it is recommended that you separately update the table and index statistics for these tables by calling the db2Runstats API or by issuing the RUNSTATS command after the redistribute operation has completed.

This option specifies that redistribute should collect statistics for the tables that have a statistics profile. Specifying this option is more efficient than separately issuing the RUNSTATS command after the data redistribution is completed.

USE PROFILE

Statistics will be collected for the tables with a statistics profile. For tables without a statistics profile, nothing will be done. This is the default.

NONE

Statistics will not be collected for tables.

Database Roles - Advantages

- Roles can mirror job functions in an organization.
- Simplified management - No need to assign or revoke authorities and privileges to individual users.
- Modifications to a role affect all users assigned to that role.
- SECADM can delegate management of a role to others.

```
DB2 GRANT ROLE TELLER TO USER BOB WITH ADMIN OPTION
```

Example Scenario Without Roles

- Aamer and Matt work as bank tellers at the Acme Bank.
- To do their day to day duties they require privileges and authorities on some database objects.

```
GRANT SELECT ON TABLE ACCOUNTS TO USER AAMER, USER MATT
GRANT SELECT ON TABLE CLIENT TO USER AAMER, USER MATT
GRANT UPDATE ON TABLE ACCT_BALANCE TO USER AAMER, USER
   MATT
```

Example Scenario Without Roles

- One day, Aamer shows up to work in shorts and gets fired for violating the Acme Bank dress code.

```
REVOKE SELECT ON TABLE ACCOUNTS FROM USER AAMER
REVOKE SELECT ON TABLE CLIENT FROM USER AAMER
REVOKE UPDATE ON TABLE ACCT_BALANCE FROM USER AAMER
```

Example Scenario Without Roles

- Dale who is a substantially better dresser than Aamer gets hired as a teller to replace him.

```
GRANT SELECT ON TABLE ACCOUNTS TO USER DALE
GRANT SELECT ON TABLE CLIENT TO USER DALE
GRANT UPDATE ON TABLE ACCT_BALANCE TO USER DALE
```

Example Scenario With Roles

- Aamer and Matt work as bank tellers at the Acme Bank.
- To do their day to day duties they require privileges and authorities on some database objects.

```
CREATE ROLE TELLER
GRANT SELECT ON TABLE ACCOUNTS TO ROLE TELLER
GRANT SELECT ON TABLE CLIENT TO ROLE TELLER
GRANT UPDATE ON TABLE ACCT_BALANCE TO ROLE TELLER
```

```
GRANT ROLE TELLER TO USER AAMER,USER MATT
```

Single point of authority and privilege management for all members of the role

Example Scenario With Roles

- One day, Aamer shows up to work in shorts and gets fired for violating the Acme Bank dress code.

```
REVOKE ROLE TELLER FROM USER AAMER
```

- Dale who is a substantially better dresser than Aamer gets hired to replace him.

```
GRANT ROLE TELLER TO USER DALE
```

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted context
  - Auditing enhancements
- Unicode Enhancements

Trusted Contexts

- What is a trusted context?
  - A database object that identifies a trust relationship between a database and an external entity such as an application server.
- What attributes can a trust relationship be based on?
  - Authorization ID of the database connection.
  - IP address or domain name.
  - Data stream encryption

42

When a user establishes a database connection, the DB2 database system checks whether the connection matches the definition of a unique trusted context object in the database. When a match occurs, the database connection is said to be trusted.

A trusted connection allows the initiator of this trusted connection to acquire additional capabilities that are not available to it outside the scope of the trusted connection. The additional capabilities vary depending on whether the trusted connection is explicit or implicit.

An explicit trusted connection is a trusted connection that is explicitly requested. It allows the initiator of the explicit trusted connection the ability to:

Switch the current user ID on the connection to a different user ID with or without authentication

Acquire additional privileges that may not be available outside the scope of the trusted connection

An implicit trusted connection is a trusted connection that is not explicitly requested. The initiator of an implicit trusted connection can only acquire additional privileges that may not be available outside the scope of the trusted connection; they cannot switch the user ID.

Trusted Contexts

- What is a trusted connection?
  - A database connection whose attributes satisfy the definition of a **trusted context** is a **trusted connection**.
- A trusted connection allows an application to acquire special privileges that are not available to it outside the scope of the trusted connection.
- Additional capabilities depend on whether the trusted connection is **explicit** or **implicit**.

43

When a user establishes a database connection, the DB2 database system checks whether the connection matches the definition of a unique trusted context object in the database. When a match occurs, the database connection is said to be trusted.

A trusted connection allows the initiator of this trusted connection to acquire additional capabilities that are not available to it outside the scope of the trusted connection. The additional capabilities vary depending on whether the trusted connection is explicit or implicit.

An explicit trusted connection is a trusted connection that is explicitly requested. It allows the initiator of the explicit trusted connection the ability to:

Switch the current user ID on the connection to a different user ID with or without authentication

Acquire additional privileges that may not be available outside the scope of the trusted connection

An implicit trusted connection is a trusted connection that is not explicitly requested. The initiator of an implicit trusted connection can only acquire additional privileges that may not be available outside the scope of the trusted connection; they cannot switch the user ID.

Trusted Contexts

- **Explicit** trusted connection is explicitly requested and allows initiator to:
  - Switch the current user ID on the connection to a different user ID with or without authentication.
  - Acquire additional privileges that may not be available outside the scope of the trusted connection (based on trusted context).
  - Requires an application change.
- **Implicit** trusted connection is not explicitly requested and does not allow the initiator to switch to a different user ID.
  - Allows the initiator to acquire additional privileges that may not be available outside the scope of the trusted connection (based on trusted context).

When a user establishes a database connection, the DB2 database system checks whether the connection matches the definition of a unique trusted context object in the database. When a match occurs, the database connection is said to be trusted.

A trusted connection allows the initiator of this trusted connection to acquire additional capabilities that are not available to it outside the scope of the trusted connection. The additional capabilities vary depending on whether the trusted connection is explicit or implicit.

An explicit trusted connection is a trusted connection that is explicitly requested. It allows the initiator of the explicit trusted connection the ability to:

Switch the current user ID on the connection to a different user ID with or without authentication

Acquire additional privileges that may not be available outside the scope of the trusted connection

An implicit trusted connection is a trusted connection that is not explicitly requested. The initiator of an implicit trusted connection can only acquire additional privileges that may not be available outside the scope of the trusted connection; they cannot switch the user ID.

Trusted Contexts - Example

```
CREATE TRUSTED CONTEXT appserver BASED UPON
  CONNECTION USING SYSTEM AUTHID aamers
  ATTRIBUTES (ADDRESS '192.0.2.1') DEFAULT
  ROLE managerRole ENABLE
```

- Simply connecting to the database from IP address 192.0.2.1 as user ID *aamers* gives me an **implicit trusted connection.**
- Running within a trusted connection gives me all the privileges and authorities defined for the role *managerRole*.

45

When a user establishes a database connection, the DB2 database system checks whether the connection matches the definition of a unique trusted context object in the database. When a match occurs, the database connection is said to be trusted.

A trusted connection allows the initiator of this trusted connection to acquire additional capabilities that are not available to it outside the scope of the trusted connection. The additional capabilities vary depending on whether the trusted connection is explicit or implicit.

An explicit trusted connection is a trusted connection that is explicitly requested. It allows the initiator of the explicit trusted connection the ability to:

Switch the current user ID on the connection to a different user ID with or without authentication

Acquire additional privileges that may not be available outside the scope of the trusted connection

An implicit trusted connection is a trusted connection that is not explicitly requested. The initiator of an implicit trusted connection can only acquire additional privileges that may not be available outside the scope of the trusted connection; they cannot switch the user ID.

Trusted Contexts Help Eliminate Security Challenges

Security Challenge	How Trusted Contexts Help
Over granting of privileges to a single user id by application server	Explicit trusted connection allows application server to switch user IDs without authentication and without needing a new connection.
Loss of end user identity for audit purposes	End user identity preserved by switching user IDs
Diminished user accountability	User accountability is intact as end user identity is preserved when the application server performs database SQL on behalf of the end user

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted context
  - Auditing enhancements
- Unicode Enhancements

Auditing Enhancements

- Four main areas of enhancement:

1. Reduce amount of data gathered for auditing
   - Finer grained control of what needs to be audited.
2. Minimize performance impact.
   - Reduce overhead associated with auditing.
3. User specifiable location for audit log(s).
   - Better disks can be used than for the sqllib directory
   - Allows offline archiving without having to extract data from the audit log.
4. Enhanced security for audit log management.
   - SECADM not SYSADM will have authority over audit log configuration and contents.

There are three primary concerns which customers have. The first is that the amount of data audited is currently too large. Customers rapidly fill up their disks, for some customers on the order of 1 gigabyte of data per hour. Much of this data is unnecessary to keep, but the current method of configuring audit forces it to be generated. This solution introduces many new methods of configuring audit such that a much finer granularity of control can be had, allowing a much smaller and more focused set of audit data to be collected. Individual databases will be able to have their own audit configuration, as well as particular objects within the database, such as tables, or even users, groups and roles. This solution will also introduce a new audit category called EXECUTE, which will allow customers to audit just the SQL statement that is being executed. Currently customers need to audit the CONTEXT event to capture this detail. However, the CONTEXT event encompasses much more than just the SQL statements, and is by far the largest contributor to the amount of data audited.

The second concern is performance of the DB2 server when auditing, which under the worst case can significantly slow down the database. The finer configuration mentioned above will greatly increase the performance as less data will be written to disk, which is the greatest contributor to the performance decrease. Additional internal changes will be made with the specific goal of increasing performance by reducing the overhead associated with auditing.

The third concern is with the location of the audit data. The current location in the sqllib/security directory is often limiting for customers, as it is typical to have the instance directory (sqllib) on a smaller, slower disk drive. This solution will introduce a method where the audit log can be moved to a customizable path, which allows a customer to place the auditing on a large

Audit Policies

- Audit configuration is done by the specification of an **Audit Policy.**
- Policies determine what is to be audited.
- Multiple policies can be created by SECADM.
  - For example one policy could to be related to sensitive data access (from a particular table for example), and another policy could be related to DBADM activities.

Audit Policies

- Audit policies can apply to subsets of database objects.
- Database objects that can have audit policies associated against them:
  - **The whole database.**
  - **Specific tables.**
  - **One or more trusted contexts.**
  - **Authorization IDs – representing users, groups and roles.**
  - **Specific authorities (SYSADM, SYSCTRL, SYSMAINT, SYSMON, DBADM, SECADM).**
- Audit configuration (policies) are per database, not per instance

50

For example, you can say ‘audit all activity by users with DBADM authority’, or ‘audit all access to this table’. In this manner a very narrow scope can be placed on what is audited which can drastically reduce the amount of data that is written to disk. New DDL statements are introduced in this solution to create, alter, drop and comment on audit policies, as well as to associate those policies with the auditable objects.

Before VIPER II audit configuration was applied to the entire instance.

EXECUTE Category

- Superior method of capturing SQL than CONTEXT category.
- Only captures events associated with execution of SQL.
- No need to consult the catalogs when trying to reproduce static SQL as with the CONTEXT category.
- **Data captured with EXECUTE WITH DATA allows the SQL to be replayed at a later date.**

Currently, in order to capture all of the SQL statements that a user issues, the CONTEXT category must be used. However, there are a number of problems with the CONTEXT category:

There are a lot of events captured that one may not be interested in if only the execution of SQL statements is of interest.

It is very difficult to log the SQL statement when using static SQL. The statement is not logged. The catalogs must be consulted and archived along with the audit logs in order to determine the statement.

Host variable and parameter markers are not logged, rather the character '?' takes their place.

The EXECUTE category is being introduced to alleviate these problems, and address the requirement of customers that we be able to easily capture just the SQL statements issued by users.

The EXECUTE category has two large advantages. The first is that the EXECUTE category can be used to accurately track what SQL statements a user issues without the use of other categories. This allows a smaller set of events to be captured reducing performance overhead. The other advantage of the EXECUTE category is that the data captured allows the SQL statement to be replayed at a later date. This requires the data to be restored to the state it was when the statement was issued at which point the statement can be replayed to view the results. For example, replaying the statement can be used to see exactly what rows a SELECT statement returned, given an accurate reproduction of the data at the time the statement was issued..

EXECUTE WITH DATA

- Before Viper II using the CONTEXT category might allow an auditor to see a dynamic SQL statement as follows:

```
SELECT ccv_num, social_sec_num,
  mothers_maiden_name, FROM accounts WHERE
  account_num = '?'
```

- EXECUTE WITH DATA category will show the actual value of the parameter / host variable as used at runtime:

```
SELECT ccv_num, social_sec_num,
  mothers_maiden_name, FROM accounts WHERE
  account_num = 210000121234
```

52

Currently, in order to capture all of the SQL statements that a user issues, the CONTEXT category must be used. However, there are a number of problems with the CONTEXT category:

There are a lot of events captured that one may not be interested in if only the execution of SQL statements is of interest.

It is very difficult to log the SQL statement when using static SQL. The statement is not logged. The catalogs must be consulted and archived along with the audit logs in order to determine the statement.

Host variable and parameter markers are not logged, rather the character '?' takes their place.

The EXECUTE category is being introduced to alleviate these problems, and address the requirement of customers that we be able to easily capture just the SQL statements issued by users.

The EXECUTE category has two large advantages. The first is that the EXECUTE category can be used to accurately track what SQL statements a user issues without the use of other categories. This allows a smaller set of events to be captured reducing performance overhead. The other advantage of the EXECUTE category is that the data captured allows the SQL statement to be replayed at a later date. This requires the data to be restored to the state it was when the statement was issued at which point the statement can be replayed to view the results. For example, replaying the statement can be used to see exactly what rows a SELECT statement returned, given an accurate reproduction of the data at the time the statement was issued..

Audit Policies – Real World Examples

- The Acme Bank HR database contains some sensitive data in the “SALARIES” table.
- Corporate policy requires that access to all such data be audited.

```
CREATE AUDIT POLICY SENSITIVEDATAPOLICY
  CATEGORIES EXECUTE WITH DATA
  STATUS BOTH ERROR TYPE AUDIT

AUDIT TABLE SALARIES USING POLICY
  SENSITIVEDATAPOLICY
```

Audit Policies - Examples

- The Acme Bank corporate policy requires that all actions to the HR database by administrators and personnel managers be audited.
  - Assumes that a role PERSONNEL_MANAGER has been created with the “CREATE ROLE” statement.

```
CREATE AUDIT POLICY ADMINSPOLICY
   CATEGORIES
          EXECUTE WITH DATA STATUS BOTH,
          SYSADMIN STATUS BOTH ERROR TYPE AUDIT

AUDIT SYSADM,DBADM,SYSCTRL,SYSMAINT,SYSMON,SECADM,
   ROLE PERSONNEL_MANAGER
   USING POLICY ADMINSPOLICY
```

Agenda

- Scalability and Performance
  - Threaded architecture and memory/agent configuration simplifications
  - Optimistic locking
  - Optimized bulk deletion
  - LOB performance improvements
  - Decimal floating point
- TCO and Manageability
  - Further automated statistics collection improvements
  - Backup/recovery enhancements
  - Automatic compression enhancements
  - Automatic storage enhancements
- High Availability & Resiliency
  - Completely integrated HA and DR solution
  - Hardware Memory Protection Exploitation
  - Serviceability improvements
- Data Redistribution Enhancements
- Security
  - Database roles
  - Trusted context
  - Auditing enhancements
- Unicode Enhancements

Unicode is the Default Code Page for New Databases

```
CREATE DATABASE MYDB
```

- DB2 9 Behaviour:
  - Default code page is based on client / application code page where the CREATE DB is done from.
- **Viper II Behaviour:**
  - **Database will be Unicode REGARDLESS of the client /application code page.**
- To create a database with a specific code page in Viper II, you must explicitly specify CODESET and TERRITORY:

```
CREATE DATABASE MYDB USING CODESET ISO8859-1
   TERRITORY CA
```


Application needs to ask for blocking to benefit.

Motivation for Making Unicode the Default

- **DB2 9:**
  - The ACME Widget company located in Omaha, Nebraska has a database which stores its customer information (names and addresses).
  - The ACME DBA created the database after he installed DB2 9.

```
CREATE DB ACMEDB
```

  - ACMEDB defaults to 819, territory US – based on the locale of the client that the DBA issued the CREATE DB command from.
  - Later that year, ACME acquired the Guandong Widget company in Guandong, China.
  - **Unfortunately the Guandong Widget company's customer information tends not to fit very well in a single byte codepage:**

汉字 漢字 形聲字

57

Application needs to ask for blocking to benefit.

Motivation for Making Unicode the Default

- **DB2 Viper II:**
  - The ACME Widget company located in Omaha, Nebraska has a database which stores its customer information (names and addresses).
  - The ACME DBA created the database after he installed DB2 Viper II.

```
CREATE DB ACMEDB
```

  - ACMEDB defaults to Unicode.
  - Later that year, ACME acquired the Guandong Widget company in Guandong, China.
  - **The Guandong Widget company's customer information fits just fine in ACMEDB's Unicode codepage**

汉字 漢字 形聲字

Application needs to ask for blocking to benefit.

Default Collation for Databases

- Since we are defaulting to Unicode for CREATE DATABASE what is the default collation?
  - Binary collation as a default could cause problems.
- We've taken single byte weight tables and applied them to Unicode data.
- So if my operating system locale uses code page 819 then although the default code page of a database in Viper II will be UTF-8, we will use a weight table that matches 819 collation for the 256 characters that appear in code page 819.

Application needs to ask for blocking to benefit.

COLLATE USING Clause on CREATE DB

- CREATE DB accepts an optional COLLATE USING clause

```
CREATE DATABASE MYDB COLLATE USING SYSTEM_819_US
```

- In DB2 9 there were a limited number options that could be specified.
- Viper II adds 92 new collations.

Application needs to ask for blocking to benefit.

COLLATION_KEY_BIT Scalar Function for Culturally Correct Collation

- COLLATION_KEY_BIT(<column>,<collation name>)
- Used with ORDER BY clause:
  - e.g. to get German ordering:

```
ORDER BY COLLATION_KEY_BIT(name, 'UCA400R1_LDE')
```

- When used, the specified collation will be used instead of the database's collation.
- Collations that are culturally correct are supported as a parameter to COLLATION_KEY_BIT.
  - Such collations are not yet supported as default for the database.

Application needs to ask for blocking to benefit.

Larger list of character sensitive string functions over DB2 9:

- Scalar functions that are now character sensitive in Viper II that were not character sensitive in DB2 9:
  - INSERT
  - LEFT
  - OVERLAY
  - RIGHT
  - STRIP
  - TRIM
- New optional parameter:
  - OCTETS – byte based semantics
  - CODEUNITS16 – UTF-16 semantics
  - CODEUNITS32 – UTF-32 semantics

Application needs to ask for blocking to benefit.

Session: E11

A Sneak Peak at the Next Release of DB2 for Linux, Unix, Windows

Matt Huras
Aamer Sachedina

IBM
huras@ca.ibm.com
aamers@ca.ibm.com