



Information Management – DB2 Data Server

## How Viper2 Can Help You!

December 6, 2007  
Matt Emmerton  
DB2 Performance and Solutions Development  
IBM Toronto Laboratory  
[memmerto@ca.ibm.com](mailto:memmerto@ca.ibm.com)



## How Can Viper2 Help DBAs?

- **By putting intelligence and automation into the software**
  - Intelligence
    - Able to react to changes in environment and workload
  - Automation
    - Must have features to simplify the DBA's job
      - Make the simple tasks automatic
      - Make the hard tasks easier
      - Make the impossible tasks possible



## How can Viper2 help DBAs?

- **Application Development Environment**
  - Rich set of support languages
  - Native XML storage and query capabilities
- **Database failover**
  - HADR
- **General Robustness**
  - Read Retry
  - Storage Keys
- **Ability to react to changing workloads**
  - Autonomic Capabilities
  - Self-Tuning Memory Manager (STMM)
  - Workload Management (WLM)
- **Ease of administration**
  - Automatic Storage
  - Table Compression and Large RID
  - Table Partitioning
  - Single-Image System Backup and Flash Copy
  - Roles



## Application Development - Languages

- **Application developers want to use what is familiar to them:**
  - C/C++, Java
  - Web “scripting” languages (PHP, Perl)
- **Having native DB2 interfaces for all of these languages eliminates the need for:**
  - Additional third-party products to interface between applications and DB2
  - Extra training, support and testing of these products



## Application Development - XML

- **Many applications are designed around “rich” data**
- **There is a desire to store and query this “rich” data natively in a data server**
- **DB2 has rich XML support:**
  - Supports X/Query and SQL/XML
  - Specialized data types, indexes, join operators and access mechanisms tailored to XML
  - Supports relational and XML data in the same database



## How can software help DBAs?

- **Application Development Environment**
  - Rich set of support languages
  - Native XML storage and query capabilities
- **Database failover**
  - HADR
- **General Robustness**
  - Read Retry
  - Storage Keys
- **Ability to react to changing workloads**
  - Autonomic Capabilities
  - Self-Tuning Memory Manager (STMM)
  - Workload Management (WLM)
- **Ease of administration**
  - Automatic Storage
  - Table Compression and Large RID
  - Table Partitioning
  - Single-Image System Backup and Flash Copy
  - Roles



# Database Failover

## High Availability (HA) and Disaster Recovery (DR)

- **HADR is a warm-standby setup for DB2**
  - Primary and secondary servers
  - Logs are shipped from the primary to the secondary and replayed on the secondary, which keeps it “warm”
  - When the primary fails, all remaining logs are shipped to the secondary and replayed and the secondary takes over
  - Clients will be rerouted automatically when a failure occurs
  - Degree of “warm”-ness can be controlled by the DBA
    - Affects the amount of data to be transferred and replayed
    - Affects the amount of downtime between primary failure and secondary takeover



# Database Failover

## High Availability (HA) and Disaster Recovery (DR)

### ▪ **Restrictions**

- OS and DB2 must be same version (including patches) but hardware can be different
  - Allows smaller hardware to be used for standby

### ▪ **Administration (pre-Viper2)**

- Required lots of scripting to manage failover properly
- Any changes (eg: add/drop tablespace container, add/drop node) had to be replicated on the secondary and scripts updated appropriately

### ▪ **Administration (Viper2)**

- TSA (Tivoli System Automation) bundled with HADR
- Completely integrated with DB2
  - Manages all schema and configuration changes automatically
  - Manages all failover operations automatically





## How can software help DBAs?

- **Application Development Environment**
  - Rich set of support languages
  - Native XML storage and query capabilities
- **Database failover**
  - HADR
- **General Robustness**
  - Read Retry
  - Storage Keys
- **Ability to react to changing workloads**
  - Autonomic Capabilities
  - Self-Tuning Memory Manager (STMM)
  - Workload Management (WLM)
- **Ease of administration**
  - Automatic Storage
  - Table Compression and Large RID
  - Table Partitioning
  - Single-Image System Backup and Flash Copy
  - Roles



# Robustness

## ▪ Read Retry Logic

- Any kind of network-mounted (NFS) or network-attached (SAN/NAS) storage can suffer from transient failures due network outages
- Prior to v9 (Viper), any read error would be considered fatal and the instance would terminate
- In v9 (Viper) and later, we retry failed reads up to 10 times to allow transient network failures to correct themselves before terminating
  - Greatly improved uptime when using network-backed storage



## Robustness

### ■ **Storage Keys**

- This is a form of hardware memory protection on POWER6
- First used in Viper2 to protect bufferpool memory against rogue memory corruption
  - From internal DB2 coding errors
  - From external UDF coding errors



## How can software help DBAs?

- **Application Development Environment**
  - Rich set of support languages
  - Native XML storage and query capabilities
- **Database failover**
  - HADR
- **General Robustness**
  - Read Retry
  - Storage Keys
- **Ability to react to changing workloads**
  - Autonomic Capabilities
  - Self-Tuning Memory Manager (STMM)
  - Workload Management (WLM)
- **Ease of administration**
  - Automatic Storage
  - Table Compression and Large RID
  - Table Partitioning
  - Single-Image System Backup and Flash Copy
  - Roles



# Changing Workloads

## Automatic RUNSTATS

- **Statistics / RUNSTATS**

- Proper statistics are essential for proper query plans
- Statistics need to be updated periodically to reflect changes in data (quantity and distribution)
- A manual process managed by DBAs

- **Automatic RUNSTATS**

- Query engine compares estimated cost with actual cost
- Once the difference in costs exceeds some threshold, we will execute RUNSTATS to update statistics
- Throttling infrastructure is used to ensure minimal impact



# Changing Workloads

## Automatic RUNSTATS

- **Viper2 introduces two new features**
- **Just-In-Time Statistics (JITS)**
  - If a high number of queries can benefit from updated statistics right now, RUNSTATS will be run immediately if it can be done without impacting performance
- **Statistics Fabrication**
  - We can estimate key statistics (cardinality, index key distributions) just by looking at overall object statistics (number of pages in a table, number of keys in each level of the index btree structure, etc)
  - These statistics are often better than what is currently being used by the optimizer



# Changing Workloads

## Automatic Backup and Reorg

### ■ **Automatic Backup**

- Performs a full backup once:
  - The last backup is > X hours old
  - More than Y log files have been written

### ■ **Automatic Reorg**

- Tables and indexes can be automatically REORGed in the background
  - Proper statistics are essential here!
- Using throttling infrastructure to ensure that performance is not impacted
- Reduces the need for large REORG operations during maintenance window

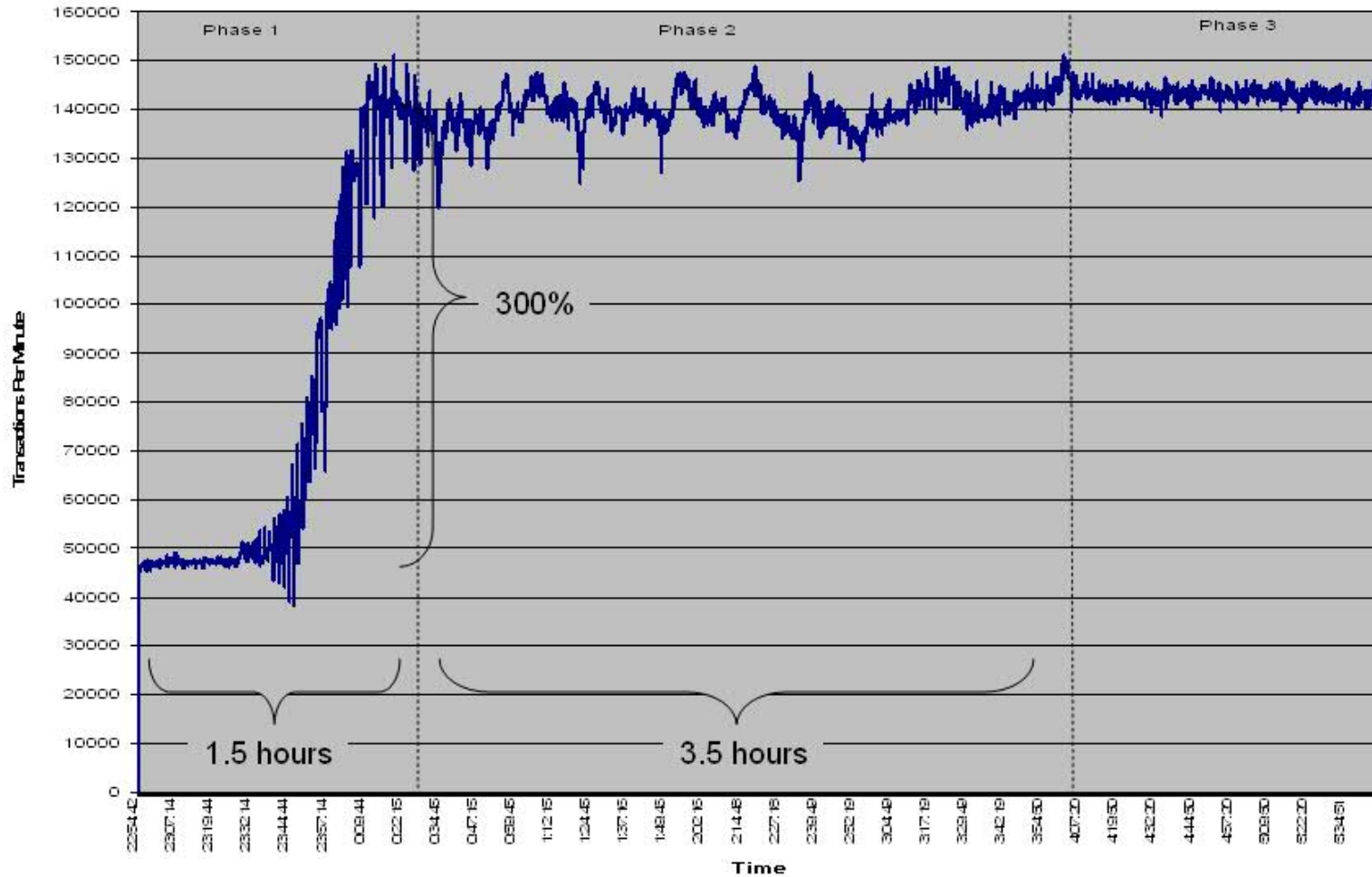


## Changing Workloads Self-Tuning Memory Manager

- **DB2 has lots of memory areas to manage**
  - Bufferpools, LOCKLIST, SORTHEAP, Package Cache
- **How to determine the “optimal” amount of memory to put in each pool, especially with a changing workload?**
  - Trial and error
  - Workload analysis
  - Use STMM
    - performs cost/benefit analysis on moving memory between memory areas
    - applies beneficial changes at runtime



# Changing Workloads Self-Tuning Memory Manager





## Changing Workloads Workload Management

- **In the past, Query Patroller was the only way to control various services classes**
- **Now we are introducing workload management (WLM) capabilities directly into DB2**
  - Allow SQL-based monitoring and reporting of the current system state
  - Does not require extensive logging and data mining
  - Can work alongside Query Patroller
  - Integrates with AIX WLM features



## How can software help DBAs?

- **Application Development Environment**
  - Rich set of support languages
  - Native XML storage and query capabilities
- **Database failover**
  - HADR
- **General Robustness**
  - Read Retry
  - Storage Keys
- **Ability to react to changing workloads**
  - Autonomic Capabilities
  - Self-Tuning Memory Manager (STMM)
  - Workload Management (WLM)
- **Ease of administration**
  - Automatic Storage
  - Table Compression and Large RID
  - Table Partitioning
  - Single-Image System Backup and Flash Copy
  - Roles

## Ease of Administration - Automatic Storage

- **Automatic Storage automates tablespace administration**
  - Type of tablespaces (DMS, SMS) chosen automatically at CREATE TABLESPACE time depending on tablespace type
  - Resizes tablespaces as necessary at runtime
    - Never get a midnight page for “out of space” again!
  - Threshold for resize and amount of resize can be controlled by DBA



## Ease of Administration - Large RID

- **What is a RID?**
  - A RID is a row identifier
  - “Small” RIDs are 4 bytes (3 byte page number and 1 byte slot)
    - Up to 16 million pages per tablespace, 255 rows per page
  - “Large” RIDs are 6 bytes (4 byte page number and 2 byte slot)
    - Up to 512 million pages per tablespace, 65,536 rows per page
- **What does this mean?**
  - Potential of hitting tablespace size limits greatly reduced
  - No need to redesign schema to get around these limits
    - DPF partitioning, UNION ALL views, Table partitioning are all ways to get around tablespace size limits
  - Allows any table to have more than 255 rows per page, but is essential for compression



# Ease of Administration

## Table Compression - Overview

### ■ How it works

- Dictionary (LZ-based) compression method (similar to WinZip)
- Data is compressed at the row level
- Data is stored in compressed form on disk and in memory; uncompressed at row access/modify time

### ■ What it buys you

- Reduced I/O cost at runtime
- Reduced storage hardware cost
- Improved bufferpool hit ratios (due to higher density of rows/page)

### ■ What restrictions exist

- XML and LOB data is not compressed
- Still limited to 255 rows/page unless you are using Large RIDs
- Degree of compression depends on workload:
  - Typical “customer” data: 68% - 78%
  - DSS: 47% - 68%
  - OLTP: 0% - 23%



## Ease of Administration Table Compression - Dictionary

### ■ **Viper**

- Compression dictionary not created by default
  - Some data must be populated
  - Run REORG to create dictionary and compress data
  - All subsequent operations will be compressed

### ■ **Viper2**

- Compression dictionary created by default



## Ease of Administration - Table Partitioning

- **What is it?**
  - Allows partitioning of tables by key ranges
  - Each range can be in a different tablespaces
  - Each range is completely independent
    - Access to a single range at runtime will not force accesses to other ranges
- **Easier Roll-In/Roll-Out**
  - New ATTACH operation for roll-in
  - New DETACH operation for roll-out
  - SET INTEGRITY is now online
- **Performance benefits**
  - Some BI-style queries can benefit due to range elimination
    - smaller joins, fewer rows to process, etc.
  - Data can be spread across multiple tablespaces
    - Increased I/O parallelism if tablespace containers are on different storage devices





## Ease of Administration Single Image Backup/Restore

- **Backing up a clustered database:**
  - Each node backed up separately (in parallel)
  - Catalogs backed up separately (serially)
  - Logs backed up separately (serially)
- **Keeping track of everything is difficult**
- **Viper2 now supports a single backup command for clustered database that does all three of these things together**



## Ease of Administration - Flash Copy

- **A painful process**
  - Find LUNs to use for flash copy
  - Suspend IO
  - <perform flash copy>
  - Unsuspend IO
  
- **Much simpler in Viper2**
  - BACKUP DB ... USE SNAPSHOT



## Ease of Administration - Roles

### ■ Before Viper 2:

- Privileges and authorities are per-object
- Multiple commands required to allow a user access to the objects that they need

```
GRANT SELECT ON TABLE ACCOUNTS TO USER GEORGE
```

```
GRANT SELECT ON TABLE CLIENT TO USER GEORGE
```

```
GRANT UPDATE ON TABLE ACCT_BAL TO USER GEORGE
```

- Granting or revoking access is tedious, error-prone and time-consuming
  - A single missed statement could be a security hole!



## Ease of Administration - Roles

- **After Viper 2:**

- Roles can be defined that encompass privileges and authorities on multiple 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_BAL TO ROLE TELLER
```

- Allowing or disallowing a user is now as simple as granting or revoking that role from a user

```
REVOKE ROLE TELLER FROM USER GEORGE
```

```
GRANT ROLE TELLER TO USER BILL
```



## Conclusion

- **Viper2 provides functionality ...**
  - to assist with workload migration and/or failover
  - to mitigate and/or limit failures
  - to minimize to amount of operator intervention for routine administration
  - to support a diverse application development environment
- **Which helps you ...**
  - Be more efficient
  - Do more with less



# Questions?