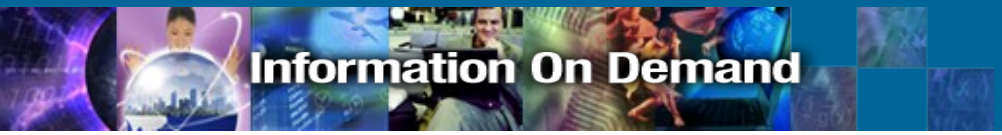


DB2 9

Platform: Linux, Unix, Windows

ON DEMAND BUSINESS™

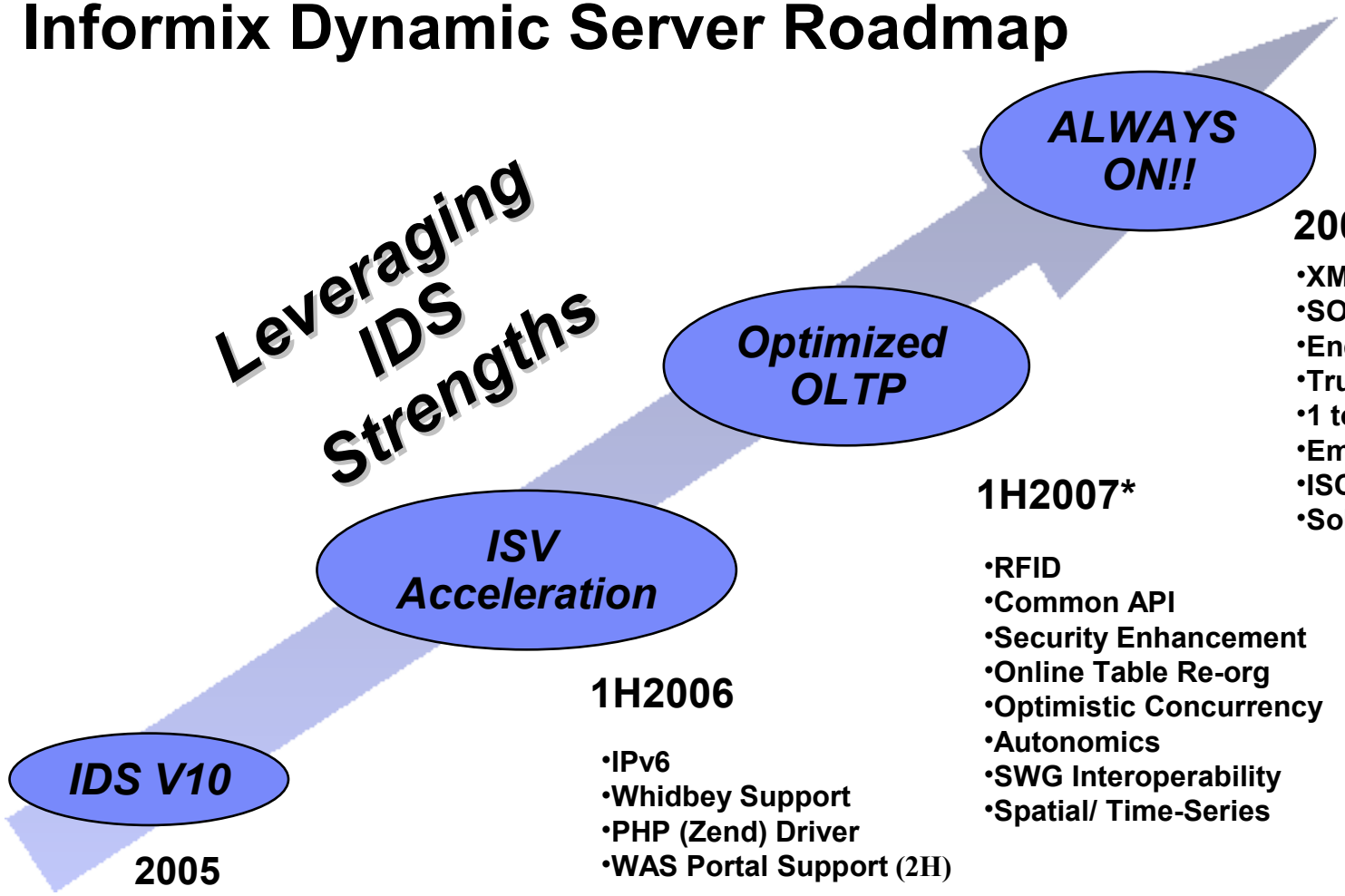
Agenda

- The IBM Database portfolio roadmap and directions
- DB2 9 Great new stuff ….
  - Compression
  - Large RID
  - Table Partitioning
  - Autonomics
  - XML (covered in detail next session)
- Summary

IBM Database Portfolio Highlights

- ***IDS v10***
  - largest release since IDS 9.x in 1996.
  - Fastest ever: 13-20% faster than IDS 7.31.
  - Strong V10 acceptance - 70+ clients & partners in 1st 60 days of availability.
- ***DB2 9***
  - No limits, highly available.
  - Embedding of DB2 as default database for SAP.
  - Full support of XML as backbone to SOA.
- ***UniVerse and UniData (U2)***
  - Full-featured, high-performance MultiValue databases.
  - Large network of U2 Business Partners
  - Large customer install base in SMB

Informix Dynamic Server Roadmap



2005

- IDS Express & Runtime
- Wal-Mart
- CISCO
- MQ series support
- Oracle-to-IDS MTK

1H2006

- IPv6
- Whidbey Support
- PHP (Zend) Driver
- WAS Portal Support (2H)

1H2007*

- RFID
- Common API
- Security Enhancement
- Online Table Re-org
- Optimistic Concurrency
- Autonomics
- SWG Interoperability
- Spatial/ Time-Series

2008*

- XML Enablement
- SOA
- Encrypted Key Management
- Trusted Context
- 1 to N HDR Chaining
- Embeddability Kit
- ISC/Advisors
- Solutions Install

UniData Roadmap

UniData:
Performance,
Security, Standards

2004 — 6.1

GA 8/10/2004

- UniObjects for .NET
- XML Schema
- XML/DB with GUI Mapping Tool
- XML DOM API
- SOAP Client API
- JDBC - Scrollable Cursors & SSL
- UCI Performance
- Table Relationship Mapping for ODBC

2005 — 7.1

eGA 6/30/2005

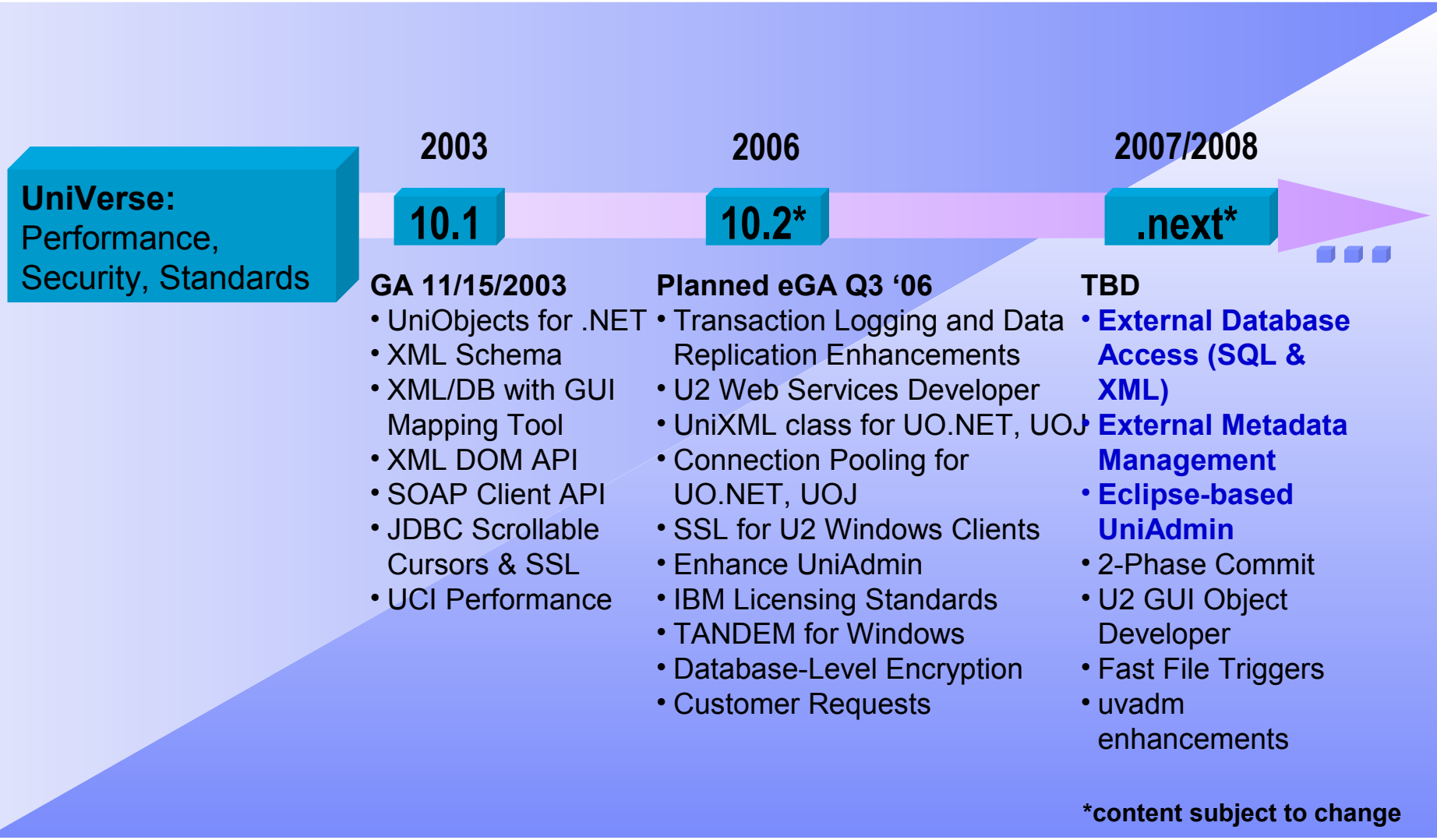
- External Database Access
- External Metadata Mgt
- RFS/TP for Windows
- Non-RFS-based TP
- UniXML Class for UO.NET, UOJ
- IBM Licensing Standards
- Connection Pooling for UO.NET, UOJ
- BUILD.INDEX ONLINE
- SSL for U2 Windows Clients
- SSL Client Configuration Tool

2007/2008 — .next*

TBD

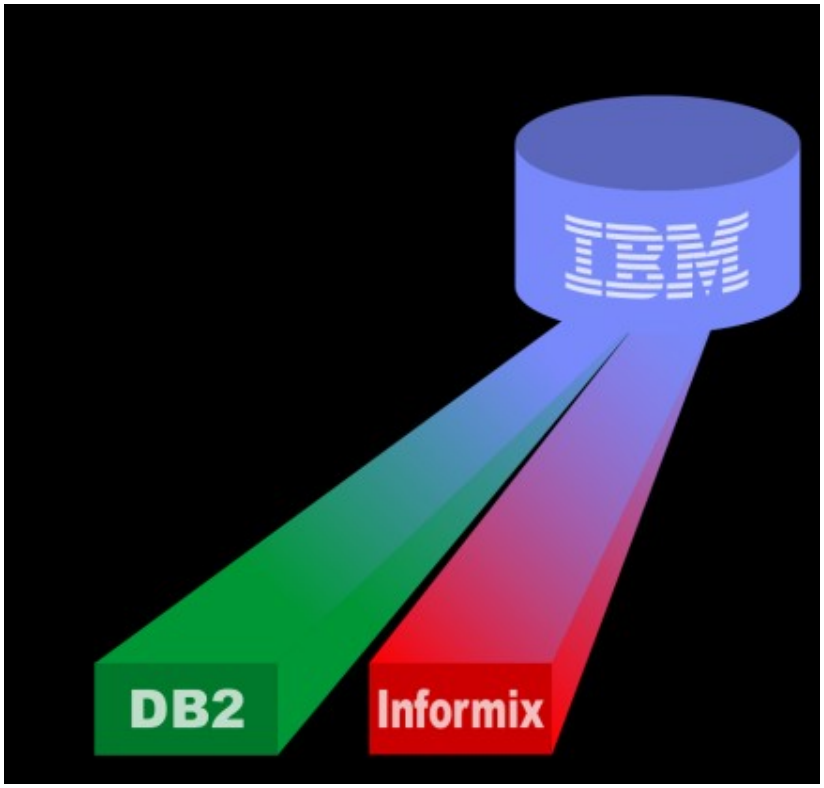
- **Integration with IBM Portfolio**
- **External Database Access (XML)**
- RESIZE ONLINE
- Interoperability Management
- **U2 Soap Server**
- U2 GUI Object Developer
- 2-phase commit
- **Database-level Encryption**
- Customer Driven Enhancements

UniVerse Roadmap



IBM Database Technology Strategy

- Continued Focus on Performance, Scale, Availability
- Reduce TCO and Accelerate Time-to-value
- Support for New Data Types
- Deep Cross-middleware Integration



Commonality Across DB Servers

TCO - Total Cost of Ownership

When you have the same old stuff …

DB2 9 Strategy and Key Investment Areas

- XML Support
- Reducing the Total Cost of Ownership
- Expanding Database Capacity and Removing Limits
- Security & Data Compression

- Upgrading to DB2 9 should be fast/simple
- If it isn't broken, don't fix it …
- Significant effort has gone into ensuring a smooth transition
  - New capabilities are available but not turned on by default
  - Performance is expected to be approximately equal on most platforms, improved on Linux

Compression

- Dictionary based - symbol table for compressing/decompressing data records
  - Lempel-Ziv (LZ) based algorithm
- Dictionary per table stored within the permanent table object (~74KB)
- Applies to base table data
- Example:

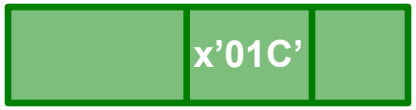
```
CREATE TABLE <table name> … COMPRESS YES

REORG TABLE <table name> … RESETDICTIONARY
```

Row Compression

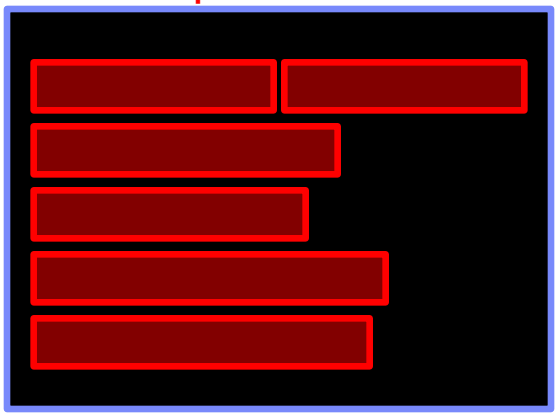
Uncompressed Row

Compressed Row

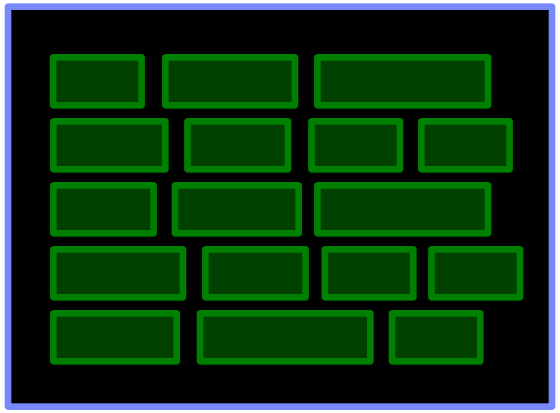


Common sequences of consecutive bytes in row replaced with 12 bit symbol

Data page with uncompressed rows



Data page with compressed rows



Compression – benefits

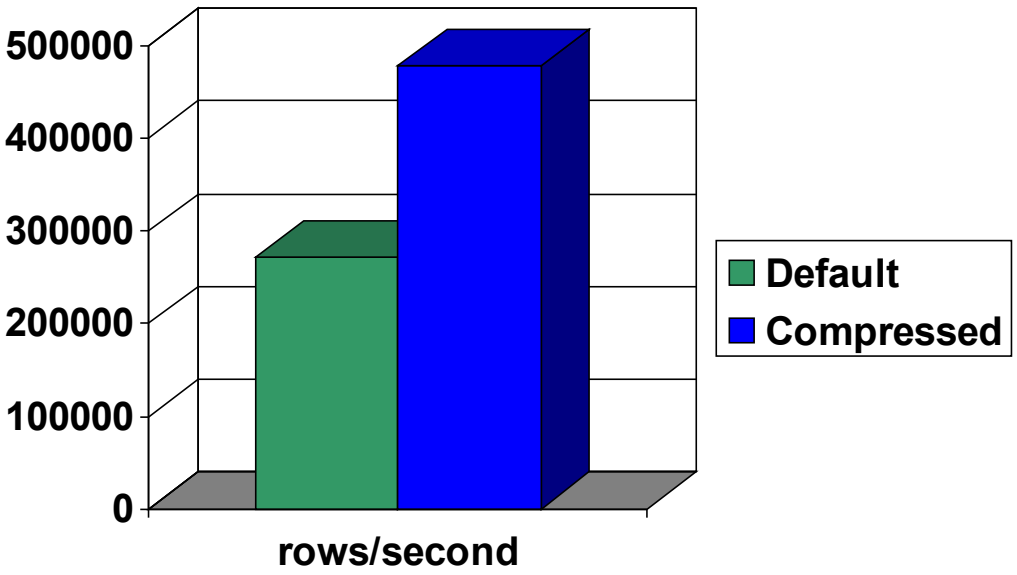
- Compression saves you significant $$ in storage
- How much depends on what percentage of your data is base table data
  - i.e index, LOB, XML data is not compressed
- Unless you have Large RIDs you are still limited to 255 rows/page max – use compression with Large RIDs!
- Degree of compression depends on data characteristics
  - DSS – 47% - 68%
  - OLTP– 20% - 25%
  - Sample Customer data – 68%-78%

Compression - considerations

- If I/O bound
  - Significant I/O bandwidth savings
  - Elapsed time can decrease
- If CPU bound
  - CPU costs increase
    - Rows must be decompressed before being processed for evaluation
  - Elapsed time can increase

Compression Results

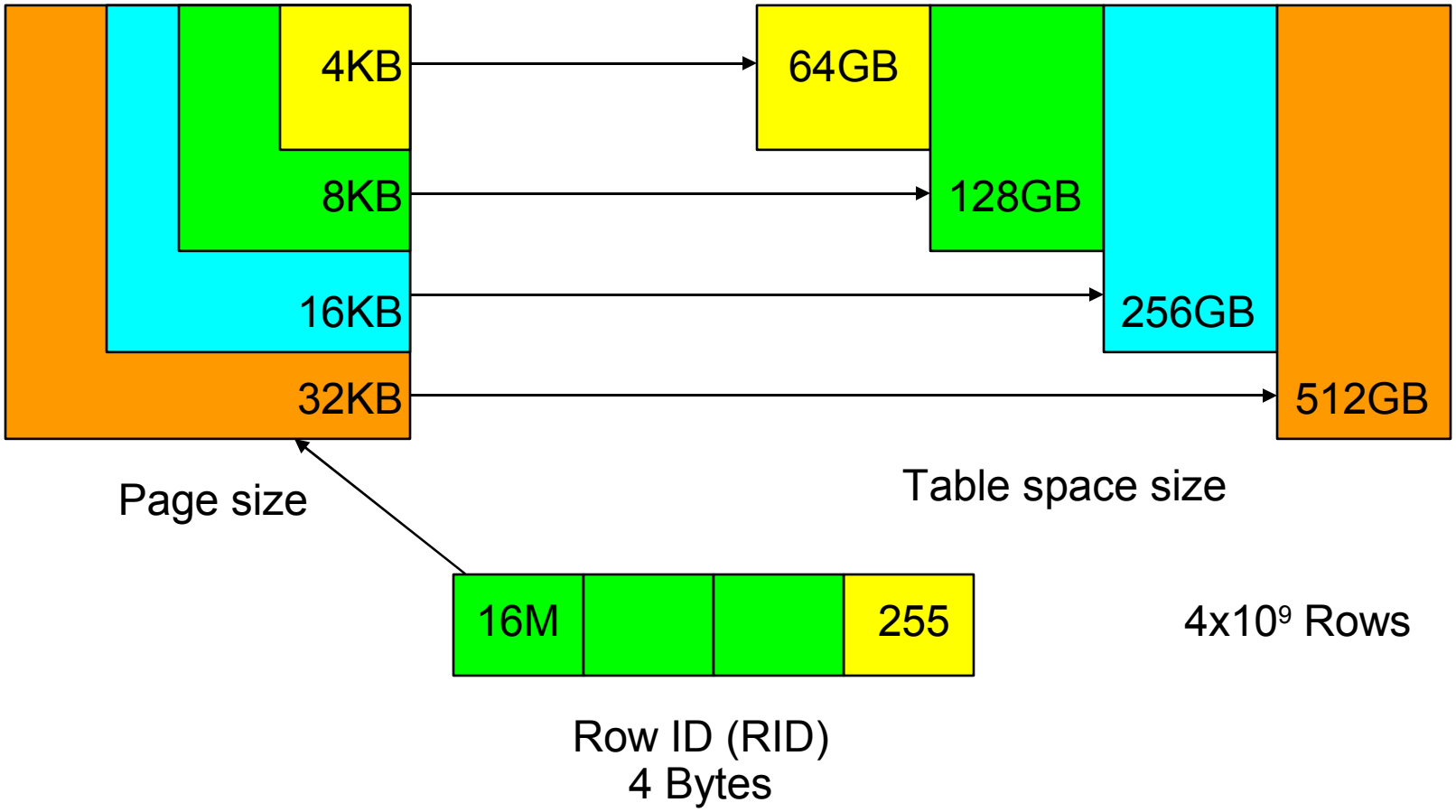
- I/O bound system
  - DSS system, table scan
  - 43% compression
  - 43% speed-up
- CPU bound system
  - OLTP system, compress largest table
  - Throughput impacted by ~8%



Large RID – the new default

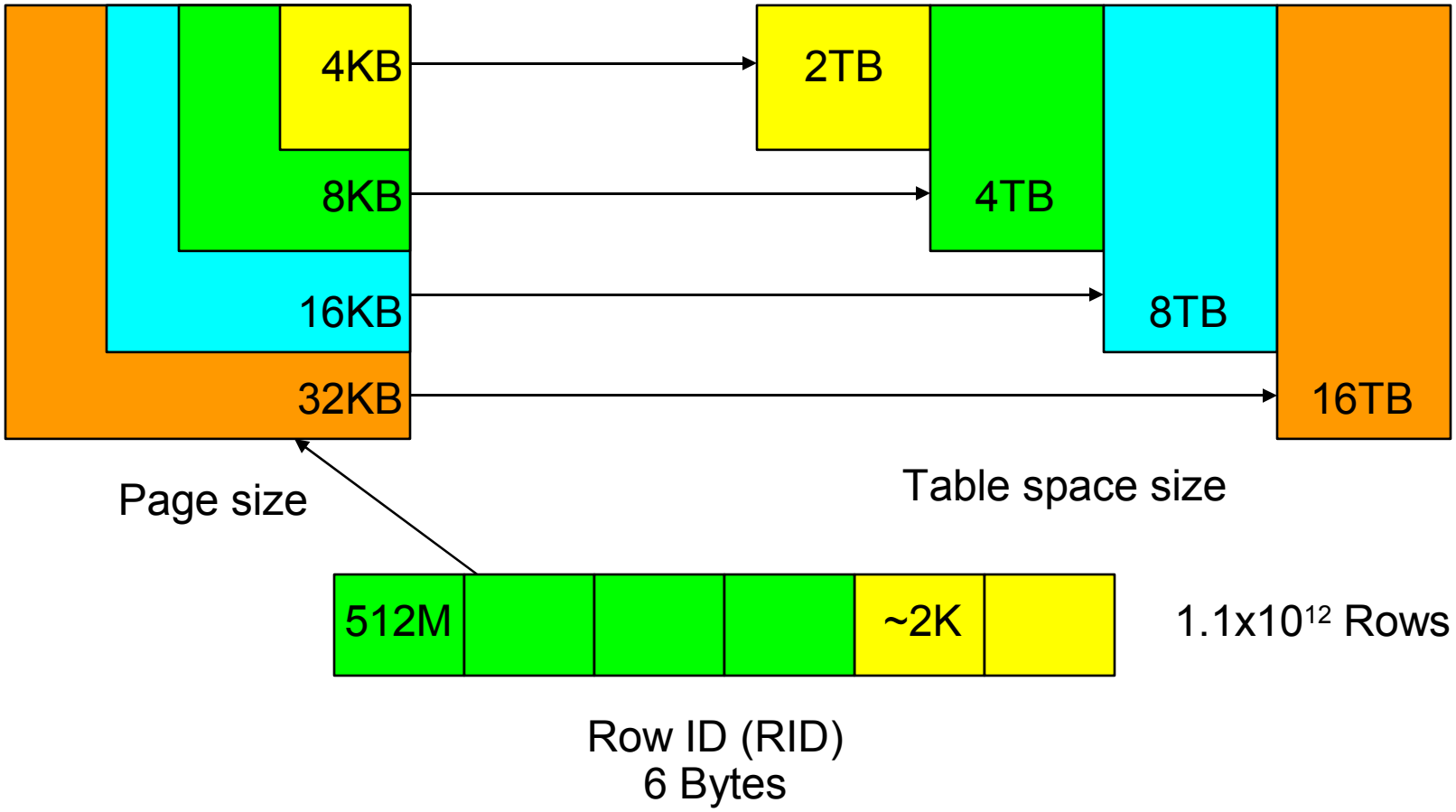
- RID – Row Identifier
  - A reference to the location of a row in a table
  - Contains the page number and the slot number
- Before DB2 9
  - RID is 4 bytes, 3 byte page number and 1 byte slot number
  - Default table space data type was REGULAR
    - CREATE TABLESPACE <tbspace-name> MANAGED BY [DMS | AUTOMATIC STORAGE | SMS]
  - Tables (data part) could not be placed in LARGE table spaces
- DB2 9
  - New 6 byte RID, 4 byte page number and 2 byte slot number
  - Infrastructure - runtime, sections, sort, log records, locks – all large RID
  - **Default table space data type for DMS table spaces is now LARGE**
    - CREATE TABLESPACE <tbspace-name> MANAGED BY [DMS | AUTOMATIC STORAGE]
  - **Tables can now be placed in LARGE table spaces**
  - **Indexes contain regular or large RIDs only, based on the table space type where the table data is stored; it has nothing to do with the type of table space where the index resides**

Previous Table Space Design



For tables in all table spaces (regular, temporary, DMS, SMS)

New LARGE and TEMPORARY Table Space Design



For tables in LARGE table spaces (DMS only)
Also all SYSTEM and USER temporary table spaces

Large RID/SLOT – benefits

- Capacity
  - No need to break apart tables due to size alone
    - Union all, DPF partitioning, range partitioning can all be combined with large RIDs
- Improved page utilization
  - Can put more than 255 rows per page
    - Particularly valuable with larger page sizes
    - Dovetails with DB2 row compression
- Manageability
  - Design flexibility, simplicity

Large RID – considerations

- Large RIDs are 50% bigger
  - Indexes grow, especially with short keys and many duplicates
    - 15% more index pages a good rule of thumb
- Using large RIDs when not strictly needed can cause a slight degradation in throughput
  - 1%-3% is a reasonable expectation
- Verify optimizer plan changes due to different statistics

Table Partitioning

- With table partitioning you can
  - Partition a table by range
  - Each range can be in a different tablespace
  - Ranges are independent
    - Access to one does not imply access to others
  - Use new ALTER ATTACH/DETACH statements for roll-in/roll-out

```
CREATE TABLE sales(sale_date DATE, customer INT, …)
 PARTITION BY RANGE(sale_date)
 (STARTING ‘1/1/2000’ ENDING ’12/31/2004’ EVERY 3 MONTHS);
```

Table Partitioning – considerations

- ALTER TABLE … ATTACH
  - Incorporates an existing table as a new range
  - Follow with SET INTEGRITY to validate data and maintain indexes
  - Data becomes visible all at once after COMMIT
  - Minimal interruption to other queries accessing table
- ALTER TABLE … DETACH
  - An existing range is split off as a stand alone table
  - Data instantly becomes invisible
  - Minimal interruption to other queries accessing table
- Key points
  - No data movement
  - Nearly instantaneous
  - SET INTEGRITY is now online

Table Partitioning / MDC / DPF

- 3 ways to spread data (can mix and match!)

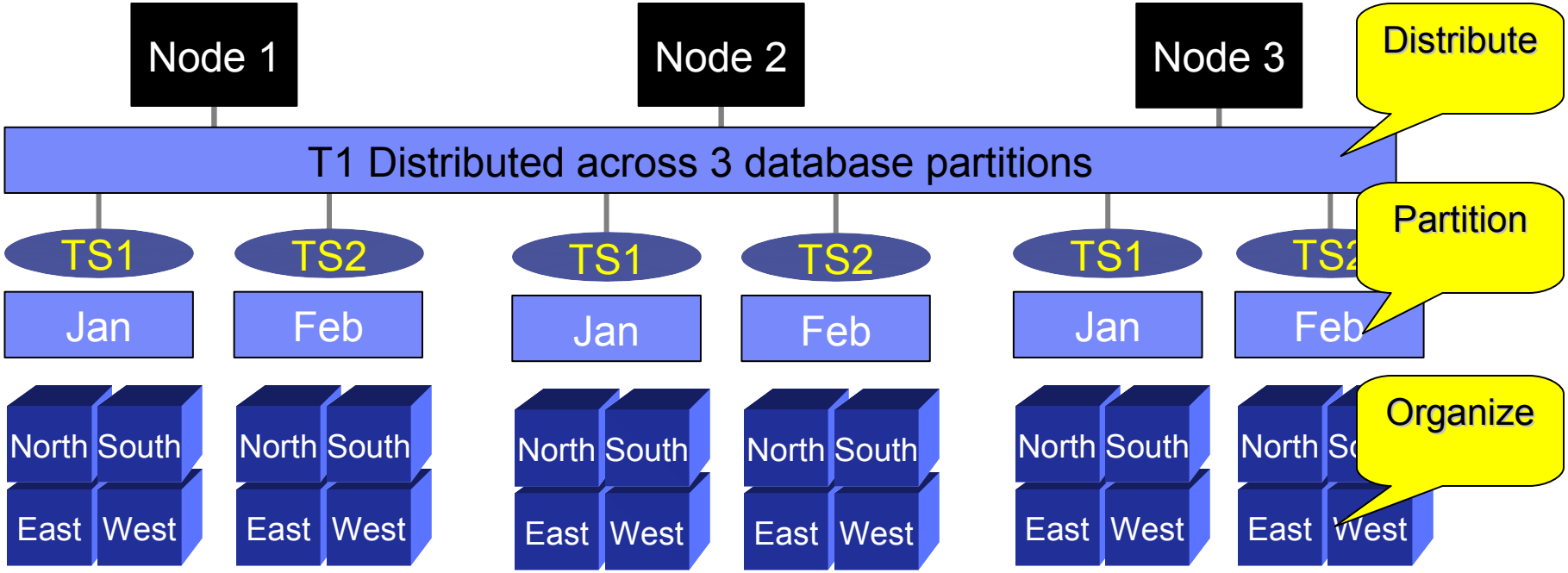


Table Partitioning – considerations

- Negligible overhead in determining the run-time partitions to touch
- Essentially linear overhead in managing data partitions
  - Select a “reasonable” number of partitions
- Similar performance to UNION ALL
- Combine with Large RID to support very large global indexes

Table Partitioning - # of Partitions

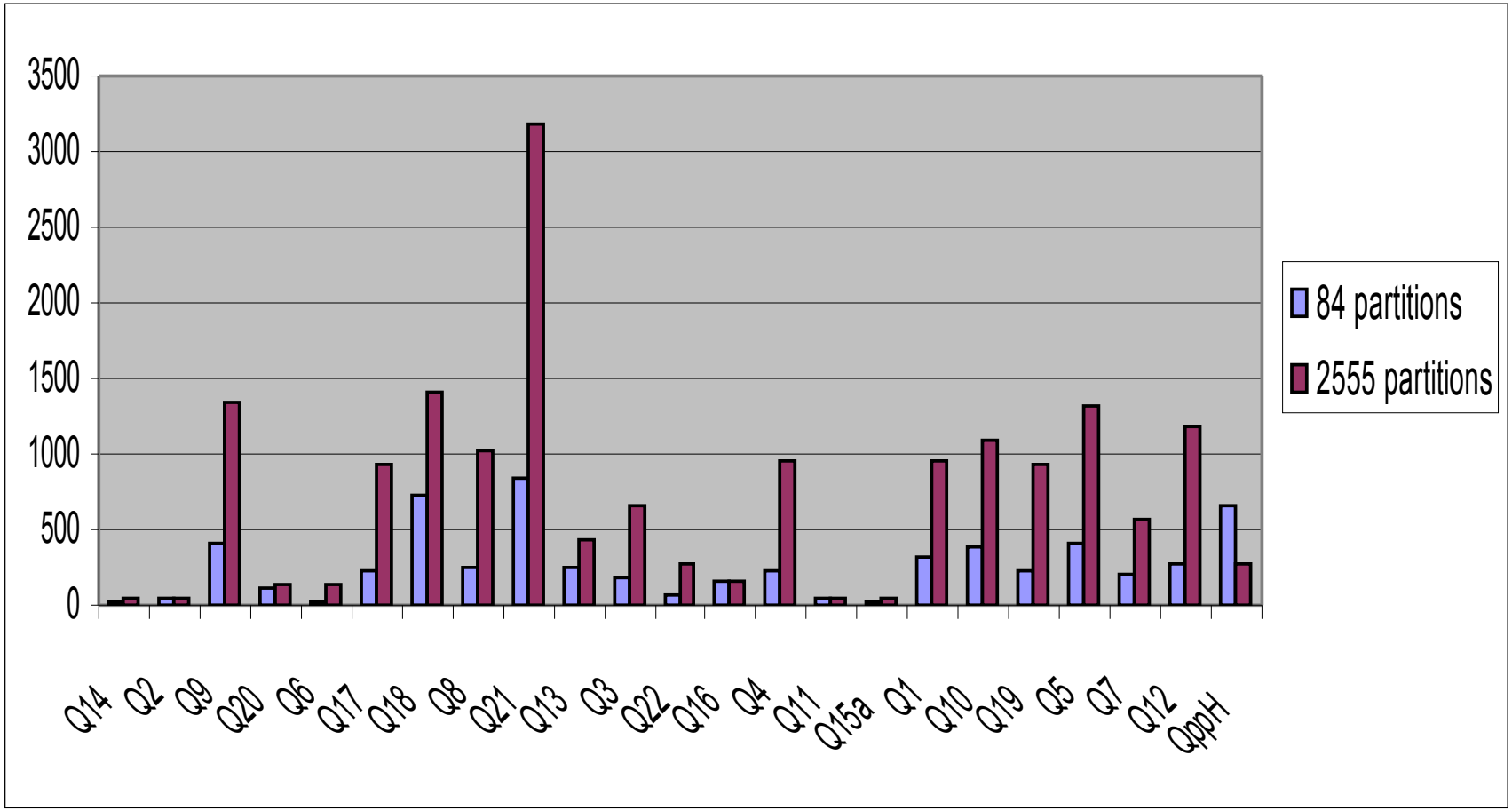
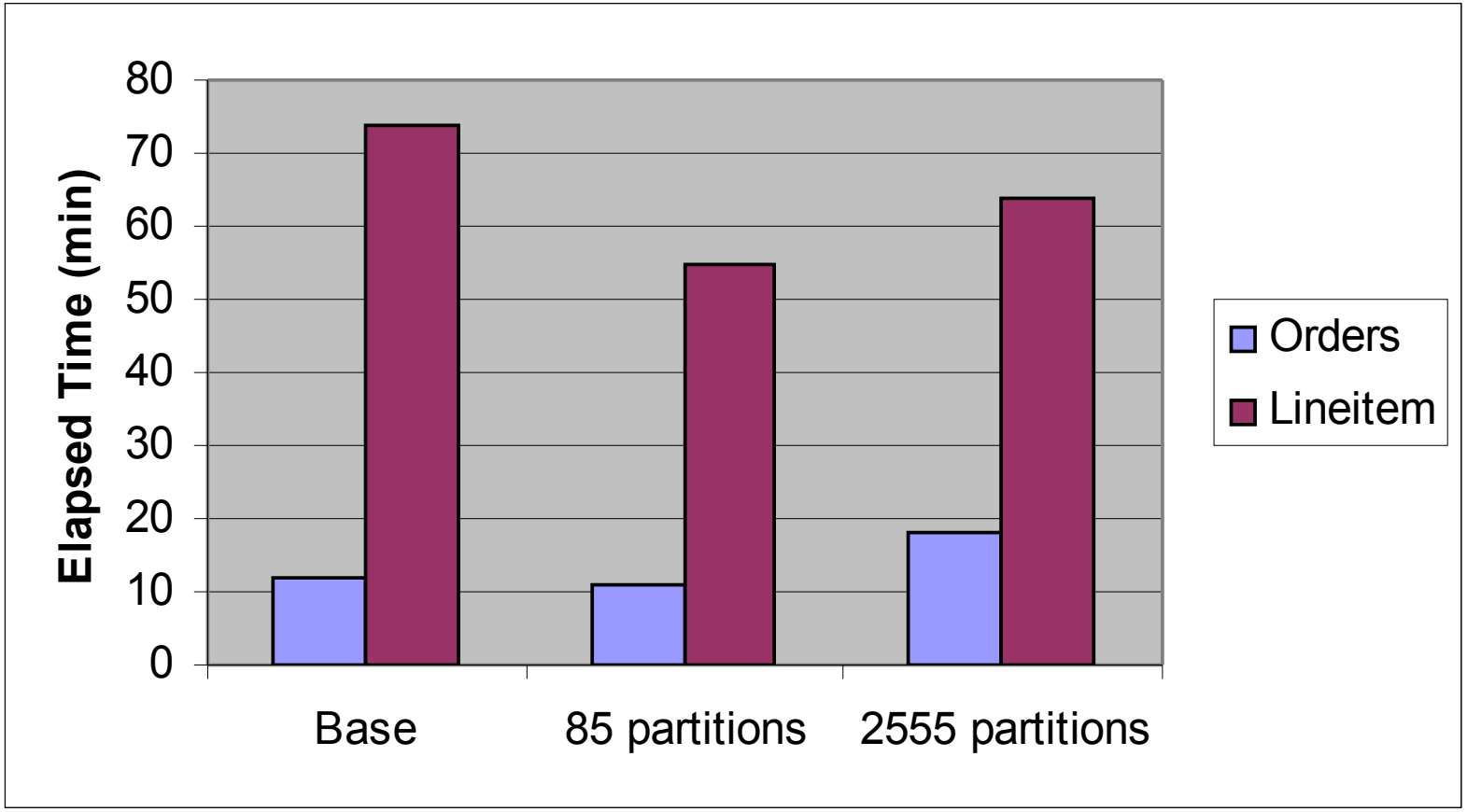


Table Partitioning – Create Index



Autonomics

- Autonomics uses built-in intelligence to automate repetitive onerous DBA tasks
- DB2 9 enables significant autonomic capability by default
  - **Automatic Storage**
    - takes the fuss out of container handling
  - **Autoconfigure for db/dbm config parms**
    - performs initial basic tuning on three dozen performance parameters
  - **Auto RUNSTATs**
    - updates table and index stats every day and only if needed.
  - **Self-tuning memory manager**
    - constantly adapt memory allocation to DB2
- DB2 Design Advisor continues to offer superb recommendations

Autonomics – benefits

- Autonomics simplifies database management which improves TCO and leads to productivity gains
  - Improved TCO by reducing/eliminating routine tuning maintenance
  - Improved performance in untuned or semi-tuned environments
  - Greater adaptability to varying workloads / operating environments

TCO - Total Cost of Ownership

Autonomics – considerations

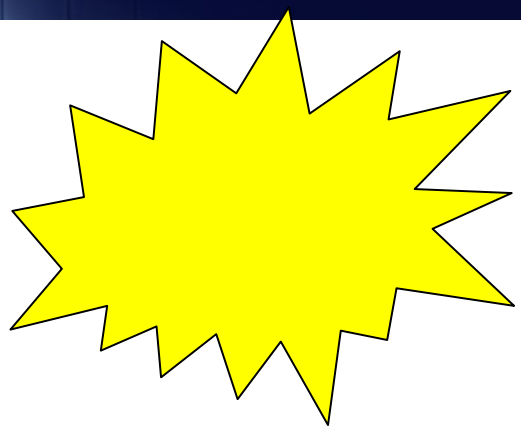
STSM: Self Tuning Memory Manager

- For a fully tuned environment
  - STMM overhead is ~2%
  - Aggregate autonomic overhead is ~3%
- For a semi-tuned environment
  - Benefits of STMM alone typically outweigh any overhead
- For an un-tuned environment
  - Leverage unused or under-utilized resources
  - 25% or greater improvement easily achieved

STMM and DATABASE_MEMORY

- STMM tunes DATABASE_MEMORY if it is set to AUTOMATIC or a numeric value
  - If set to AUTOMATIC, memory is taken from, and returned to, the OS if required by the database
    - DBA need not know how much memory to allocate to DB2
    - This is the default for newly created Viper databases
  - If set to a numeric value, memory is given to AUTOMATIC heaps up to the numeric value
    - Allows DBA to set total memory consumption for the database
    - DB2 will then distribute the memory to optimize performance
- If set to COMPUTED, no DATABASE_MEMORY tuning will occur
  - When database starts, memory requirements are computed based on the heap configuration
  - Once the database starts, the database shared memory set is allocated based on the computation
  - Version 8 AUTOMATIC behavior

Scenarios where STMM shines



Buffer pool tuning

- Difficult to tune memory when there are multiple buffer pools
- As number of buffer pools increases, possible configurations increases exponentially
- STMM works with multiple buffer pools regardless of page size
- Trades memory between buffer pools
  - Ensures that total memory doesn't change
    - 1 8k page becomes 2 4k pages in transfer
- Works so well that STMM is being used to tune benchmark systems in house

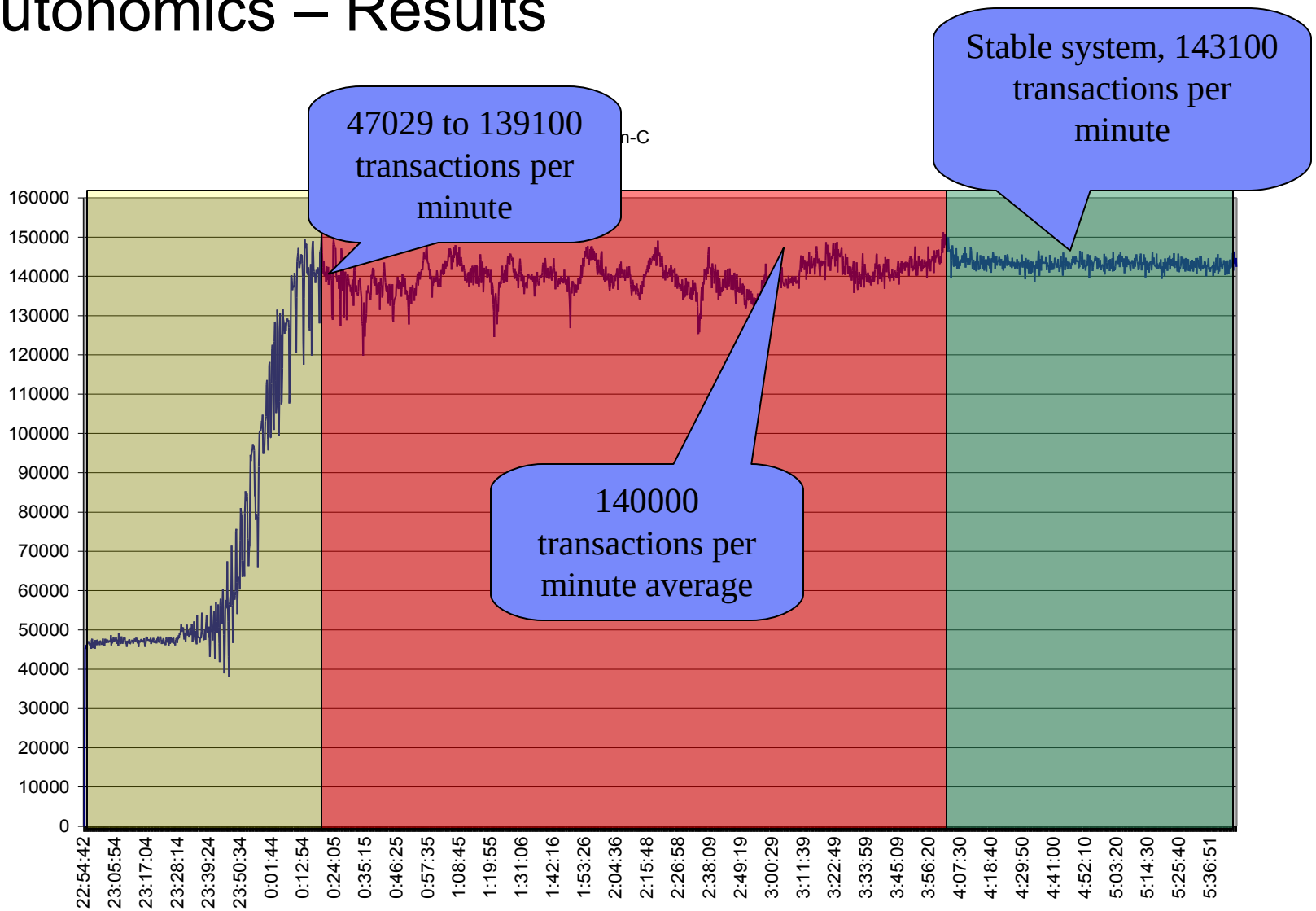
Memory varied workloads

- Some workloads have dramatically varied memory demands
  - Periods with high transaction throughput
  - Periods with long running transactions or online utilities
- STMM constantly re-evaluates the memory requirements
  - Can update the memory up to 60 times an hour
- Will optimize the memory usage based on the currently running workload
- Very difficult to perform similar tuning manually

Unknown memory requirements

- New workload with unknown memory requirements
  - Alternatively, new DB2 administrator unfamiliar with memory model
- STMM works deep down in DB2 and is able to sense workload memory requirements
- Tunes quickly enough to bring production systems from out of the box configuration to optimal in an hour or less
- Requires absolutely no DBA interaction once turned on
- Performs several weeks of manual (trial and error) tuning every hour
- Will stop tuning automatically when it reaches optimal configuration

Autonomics – Results



Evolution of the transaction rate over 8 hours

Summary

- DB2 performance leadership continues
- Core DB2 9 function is as fast or faster
- Significant new features available with DB2 9
  - Function and Performance
  - As you exploit new capabilities you will realize additional benefits
- Initial best practices are available
  - Evolution of best practices continue

XML and DB2…..

Don't go anywhere!!! ….The IBM "experts" will explain XML in the following session!


Thank
You