Information Management

IBM

What's New in DB2 10.1 for Linux, UNIX, and Windows

Filip Slánička

DB2/SAP technical professional, Community of Practice, CEE

slanicka@cz.ibm.com

19. června 2013

DB2 10.1 for LUW - High Performance, Low Costs

Low Operational Cost
Parallel processing, deep compression, and automation

Ease of Development
SQL compatibility, native XML and graph stores, and cloud support

Reliability
High availability, fast recovery, and online utilities

"With each release of DB2, ***I experience faster results with less CPU.”***
--Martin Hubel, President, Martin Hubel Consulting Inc.

Building On the Pillars of DB2

Low Operational Costs

- Adaptive compression
- Multi-Temperature Data Management
- Faster query response
- Improved index mgmt
- Real-time data warehousing

Ease of Development

- Temporal capabilities
- Row and Column Access Control
- SQL compatibility enhancements
- NoSQL graph store

Reliability

- DB2 pureScale enhancements
- Workload management enhancements
- HADR support extended to multiple standby servers

DB2 10.1 Adaptive Compression

Low Operational Costs

- **Adaptive compression is an advanced row compression technique that uses two levels of compression dictionaries (table-level and page-level) to improve compression ratios, particularly as data changes**

- **How it will help you**
  - Lower costs
    - Postpone upcoming storage purchases
    - Lower ongoing storage needs
      - Better compression rates = increased storage savings
    - Easier administration with reduced need for table reorganization
    - Compression rates remain very high, even as data grows and changes
    - Table reorganization not required to maintain high compression rates
  - Higher performance
    - Faster queries for I/O-bound environments
    - Faster backups

DB2 10.1 Adaptive Compression (cont.)

Low Operational Costs

DB2 9.7

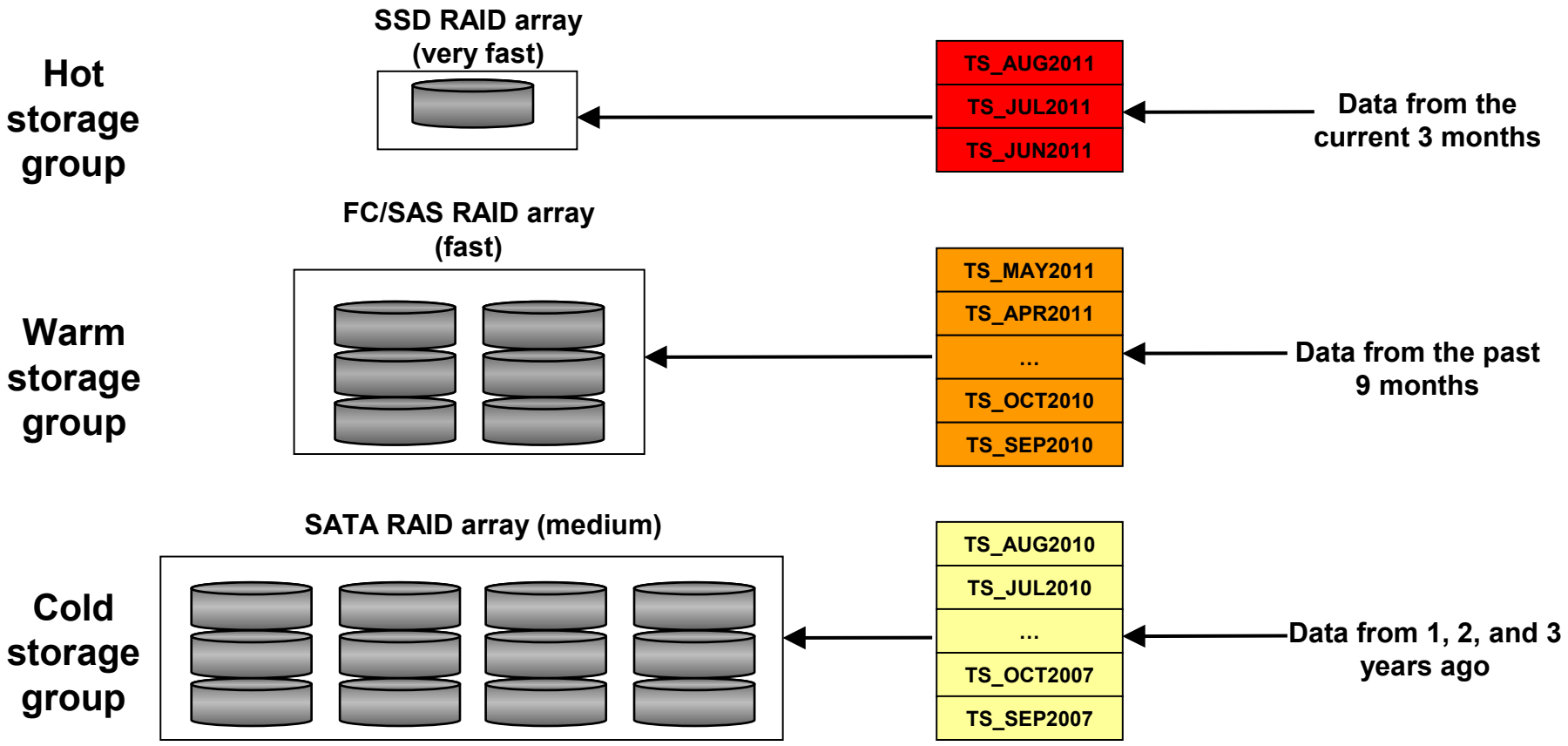
- Uses a single, static compression dictionary
- Compresses data based on recurring patterns that appear in the table
- A classic table reorganization is necessary to improve compression ratios if a significant number of records in a table have been updated, or if a large amount of new data has been inserted

DB2 10.1

- Multiple page-level dictionaries in addition to a single table-level dictionary
- Compression dictionary contains locally frequent patterns, with one dictionary stored on every page
- When a page becomes full, page-level compression is applied, immediately freeing up more storage in that page
- Reduced need for table reorganization

Multi-Temperature Data Management (cont.)

Low Operational Costs



Multi-Temperature Data Management (cont.)

Moving Data Between Storage Tiers

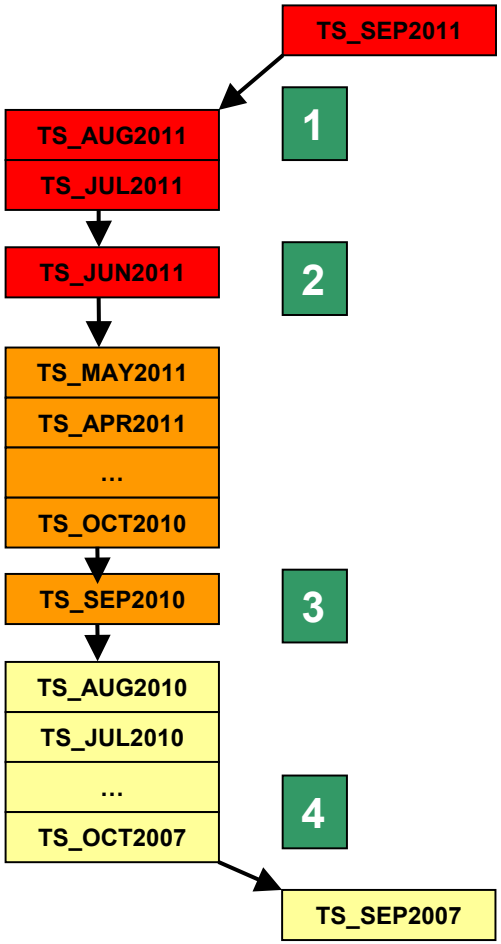
Low Operational Costs

1 ALTER TABLE sales
ATTACH PARTITION TS_SEP2011
STARTING FROM ('9/1/2011') ENDING ('9/30/11')

2 ALTER TABLESPACE TBSP_JUN2011
USING STOGROUP SG_WARM

3 ALTER TABLESPACE TBSP_SEP2010
USING STOGROUP SG_COLD

4 ALTER TABLE sales
DETACH PARTITION TS_SEP2007
INTO TS_SEP2007_DETACHED



Multi-Temperature Data Management (cont.)

Integration With the DB2 Workload Manager

Low Operational Costs

- **DB2 WLM work class and threshold DDL have been extended to support the new data tag attribute**
- **Work class sets are predictive based on compilation information**
  - Sometimes there isn't enough information at compile time (for example, queries with parameter markers) to predict which table spaces will be touched. For this reason…
  - Activities can be mapped to a service subclass based on the data expected to be touched before the activity starts to run
- **Data tag thresholds are reactive and use information that is available at runtime**
  - Activities can be remapped to a service subclass based on the data touched by the activity as it is running

Multi-Temperature Data Management (cont.)

Optimizes Your Storage Use

Low Operational Costs

- **Higher performance**
  - Improved ability to meet SLAs
- **Lower costs**
  - Gracefully extend lifespan of current storage

*"Multi-temperature data management helps in **breaking the I/O thresholds** during intensive log writings in high OLTP environments."*
-- Ahtesham Akhtar, Information Management Consultant SBM.

*"The ability to prioritize access to hot data with DB2 10 Workload Manager and Multi-temperature Storage means a **significant storage cost savings** for our customers."*
-- Radu Parvu, Senior Systems Analyst – Solution Specialist, Accenture, Finnland.

Up to 3x Faster Query Performance

Low Operational Costs

Increase Ability to Meet SLAs. Postpone Hardware Upgrades

- **Multi-core parallelism enhancements**
- **Performance improvements for**
  - Queries over star schemas
  - Queries with joins and sorts
  - Queries with aggregation
  - Hash joins
- **Higher performance**
  - Up to 35% faster out-of-the-box performance
  - Up to 3x faster when using new features
- **Lower costs**
  - Postpone hardware upgrades

JOIN-CHEER

*"DB2 10.1 performance improvements helped us in achieving nearly **5X faster query response** times."*
-- Bin Ma, Division Manager, Beijing Join-Cheer Software Co. Ltd

Index Management Redefined

Low Operational Costs

Increase Ability to Meet SLAs. Lower Administration Costs

- **Jump scan**
  - Reducing the number of indexes needed
- **Smart index prefetching**
  - Faster index access and fewer index reorganizations
- **Smart data prefetching**
  - Faster index scans and fewer index reorganizations
- **Predicate evaluation avoidance**
  - Faster index scans
- **Higher performance**
  - Faster index performance
- **Lower costs**
  - Fewer indexes to maintain
  - Dramatic reduction in index reorganization

"Jump Scan ***optimizes buffer usage by 75 to 80%,*** *resulting in very good* ***improvement in overall performance and saving the CPU cycles****."*
-- Shanmukhaiah D, Cognizant Technology Solutions.

Time Travel Query – Temporal Tables

Ease of Development

- **System-period temporal table**
  - DB2 tracks the period when a row is valid (beginning when it is inserted, and ending when it is either updated or deleted)
  - All currently valid rows are stored in one table
  - When a row is no longer currently valid, it is automatically moved to an associated history table that is transparent to users and applications
  - DB2 will automatically query the history table and return applicable rows based on the SQL executed against the base table
- **Application-period temporal table**
  - The user or application updates the beginning and end of the period in which the information is valid
  - All data is kept in a single table and DB2 will automatically split rows based on SQL activity against the existing rows
- **Bi-temporal table**
  - Combines characteristics of both types of temporal tables

Row and Column Access Control

Ease of Development

Data Centric Security

Ease of Compliance with Privacy and Sensitive Data Requirements

- **Fine-grained access control**
  - Hide rows from unauthorized users
  - Mask the value of columns for unauthorized users
- **Data-Centric security**
- **Policy-driven security, with flexible policies**
- **Does not require classification**

Account	Name	Income	Branch
1111-2222-3333-4444	Ana	22,000	A
2222-3333-4444-5555	Bob	71,000	B
3333-4444-5555-6666	Celia	123,000	B
4444-5555-6666-7777	Dinesh	172,000	C

Teller Amy sees

Account	Name	Income	Branch
2222-3333-4444-5555	Bob	71,000	B
3333-4444-5555-6666	Celia	123,000	B

Telemarketer Pat sees

Account	Name	Income	Branch
xxxx-xxxx-xxxx-4444	Ana	22,000	A
xxxx-xxxx-xxxx-5555	Bob	71,000	B
xxxx-xxxx-xxxx-6666	Celia	123,000	B
xxxx-xxxx-xxxx-7777	Dinesh	172,000	C

DB2 10 PL/SQL Compatibility

Ease of Development

Average PL/SQL Compatibility Moves Above 98%

*"The total cost of ownership with DB2 running on IBM systems is almost **half the cost of Oracle** Database on Sun systems."*
--- Reliance Life Insurance

*"We switched from Oracle Database to IBM DB2 and **cut our costs in half, while improving performance and reliability of business applications**."*
Sandro Reátegui Banco de Crédito del Peru

- *Moved from Oracle Database to IBM DB2*
- *Used "compatibility features"*
- ***3-30x faster query performance***
- ***200% improvement in data availability***

-- JSC Rietumu Banka

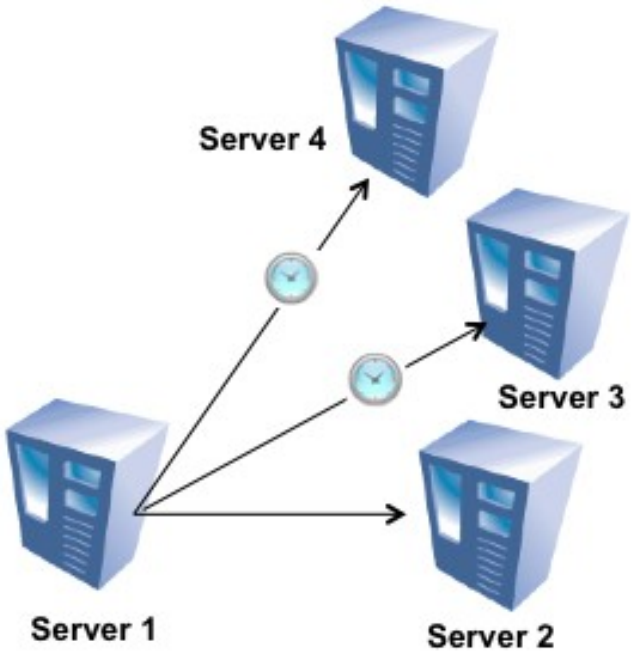
9.7.1	SUB STRB	Increase compatibility
9.7.1	UDF Parameters: INOUT	Increase compatibility
9.7.1	FORALL/BULK COLLECT	Increase compatibility
9.7.1	Improve BOOLEAN	Increase compatibility
9.7.1	Conditional Compilation	Enhancement
9.7.1	Basic DPF Support	Broaden coverage
9.7.1	OCI Support	Broaden coverage
9.7.2	UDF Parameters: DEFAULT	Increase compatibility
9.7.2	Obfuscation	Enhancement
9.7.2	NCHAR, NVARCHAR, NCLOB	Increase compatibility
9.7.3	NUMBER Performance	Performance
9.7.3	Runtime "purity level" Enforcement	Increase compatibility
9.7.3	RATIO_TO_REPORT Function	Increase compatibility
9.7.3	RAISE_APPLICATION_ERROR	Increase compatibility
9.7.3	Small LOB Compare	Increase compatibility
9.7.4	Multi-action Trigger & Update Before Trigger	Increase compatibility
9.7.4	Autonomous Tx Improvements	Increase compatibility
9.7.4	LIKE Improvements, LISTAGG	Increase compatibility
9.7.4	ROW & ARRAY of ROW JDBC Support	Increase compatibility
9.7.5	Pro*C Support	Increase compatibility
9.7.5	Nested Complex Objects	Increase compatibility
10	Local Procedure Definitions	Increase compatibility
10	Local Type Definitions	Increase compatibility
10	PL/SQL Performance	Performance

HADR Supports Multiple Standby Servers

Reliability

Increase Ability to Meet SLAs. Disaster Recovery

- **HADR now supports more than one stand-by server**
  - If primary server fails, principle standby takes over
  - If principle standby then fails, can switch to auxiliary standby
  - Auxiliary standby can provide complete offsite availability, while maintaining speed of local standby
- **Time delay apply available for the standby**



DB2® 10.5

with BLU Acceleration

New

<Speaker Name Here>
<Speaker Title Here>

DB2® 10.5

with BLU Acceleration

New

Multi-workload database software for the era of Big Data

***BLU Acceleration** – Extreme performance and storage savings, leveraging dynamic "in-memory" and columnar technologies, for analytic processing*

***DB2 pure Scale** – High availability, extreme scalability, and application transparency for OLTP workloads*

***Mobile -** Rich capabilities to support mobile devices*

***NoSQL** – Continue to support the next generation of applications*

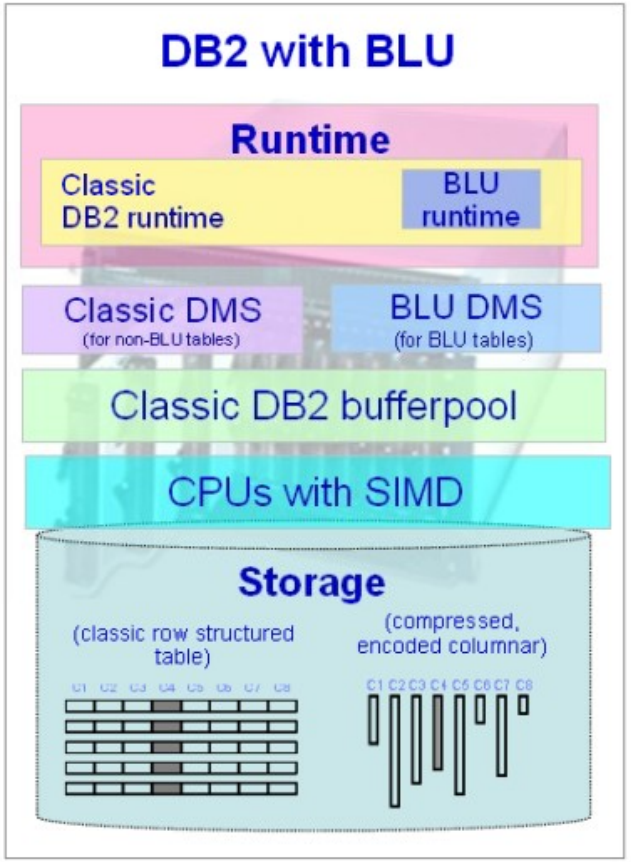
***Oracle Application Compatibility** – Continue to reduce the cost and risk associated with migrating Oracle applications to DB2*

***Enhanced Tooling** - Reducing the total cost of ownership with DB2 and making the adoption, management, monitoring, and maintenance very simple*

DB2 10.5 with BLU Acceleration

What is DB2 with BLU Acceleration?

- **New technology for analytic queries in DB2 LUW**
  - DB2 column-organized tables add columnar capabilities to DB2 databases
    - Table data is stored column organized rather than row organized
    - Using a vector processing engine
    - Using this table format with star schema data marts provides significant improvements to storage, query performance, ease of use, and time-to-value
  - New unique runtime technology which leverages the CPU architecture and is built directly into the DB2 kernel
  - New unique encoding for speed and compression
    - This new capability is both main-memory optimized, CPU optimized, and I/O optimized



How Fast Is It?
Results from the DB2 10.5 Beta

Customer	Speedup over DB2 10.1
Large Financial Services Company	46.8x
Global ISV Mart Workload	37.4x
Analytics Reporting Vendor	13.0x
Global Retailer	6.1x
Large European Bank	5.6x

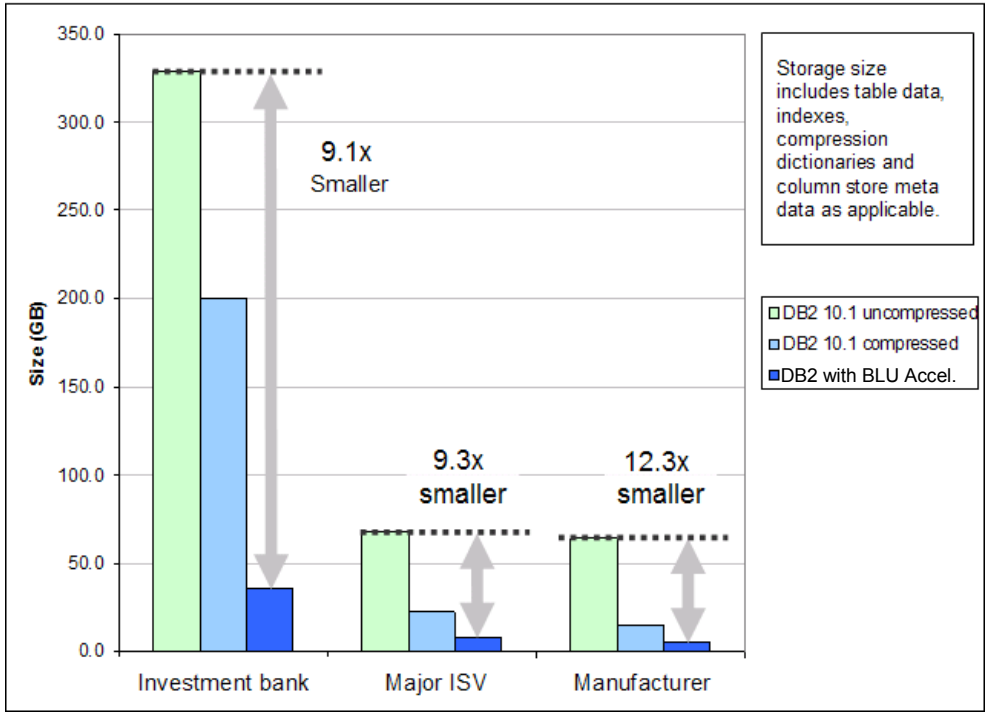
10x-25x
Improvement is common

BNSF RAILWAY

"It was amazing to see the faster query times compared to the performance results with our row-organized tables. The performance of four of our queries improved by over 100-fold! The best outcome was a query that finished 137x faster by using BLU Acceleration."
- Kent Collins, Database Solutions Architect, BNSF Railway

Storage Savings

- **Multiple examples of data requiring substantially less storage**
  - 5% of the uncompressed size
  - Fewer objects required
- **Multiple compression techniques**
  - Combined to create a near optimal compression strategy
- **Compression algorithm adapts to the data**



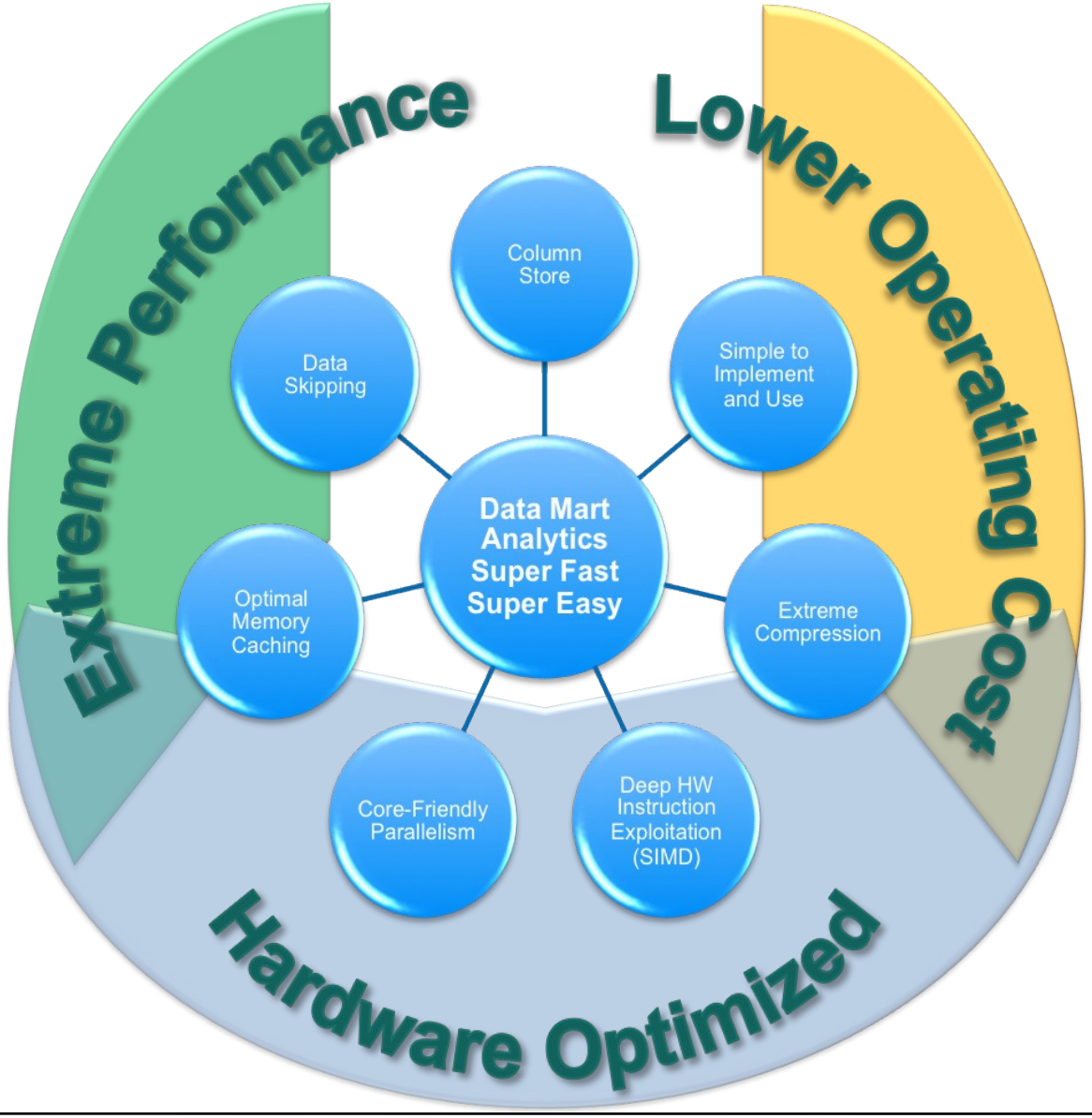
Analytic Database Management Complexity

Database design and tuning

- Decide on partition strategies
- Select compression strategy
- Create table
- Load data
- Create auxiliary performance structures
  - Materialized views
  - Create indexes
    - B+ indexes
    - Bitmap indexes
- Tune memory
- Tune I/O
- Add optimizer hints
- Statistics collection

Repeat

The Seven Big Ideas of DB2 with BLU Acceleration



7 Big Ideas: 1 Simple to Implement and Use

- **LOAD and then… run queries**
  - No indexes
  - No `REORG` (it's automated)
  - No `RUNSTATS` (it's automated)
  - No `MDC`
  - No `MQTs` or Materialized Views
  - No partitioning
  - No statistical views
  - No optimizer hints

- **It is just DB2!**
  - Same SQL, language interfaces, administration
  - Reuse DB2 process model, storage, utilities

7 Big Ideas: 1 Simple to Implement and Use

- **One setting optimized the system for BLU Acceleration**
  - Set `DB2_WORKLOAD=ANALYTICS`
  - Informs DB2 that the database will be used for analytic workloads

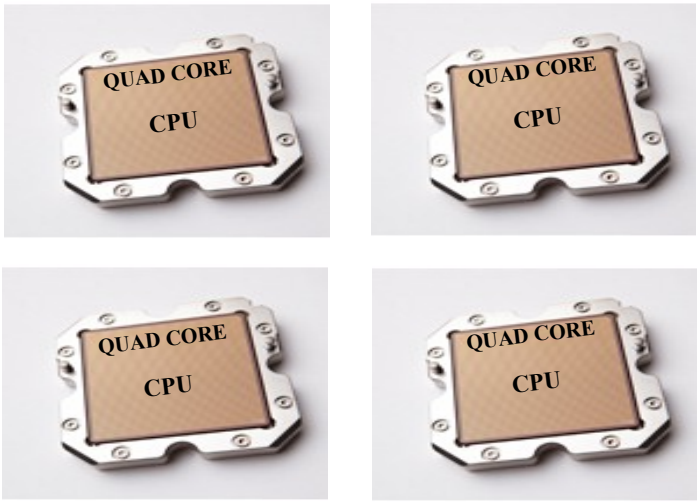
- **Automatically configures DB2 for optimal analytics performance**
  - Makes column-organized tables the default table type
  - Enables automatic workload management
  - Enables automatic space reclaim
  - Page and extent size configured for analytics
  - Memory for caching, sorting and hashing, utilities are automatically initialized based on the server size and available RAM

- **Simple Table Creation**
  - If `DB2_WORKLOAD=ANALYTICS`, tables will be created column organized automatically
  - For mixed table types can define tables as `ORGANIZE BY COLUMN` or `ROW`
  - Compression is always on – no options

- **Easily convert tables from row-organized to column-organized**
  - `db2convert` utility

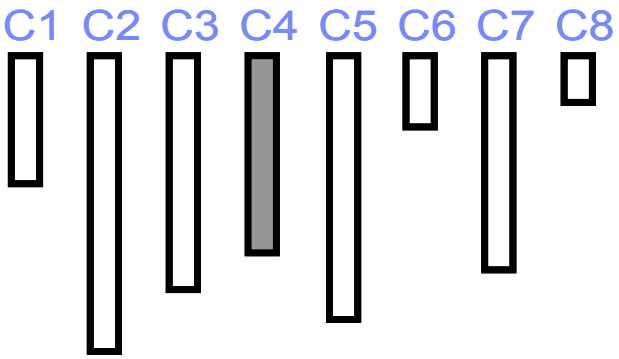
7 Big Ideas: 4 Core-Friendly Parallelism

- **Careful attention to physical attributes of the server**
  - Queries on BLU Acceleration tables automatically parallelized
- **Maximizes CPU cache, cacheline efficiency**



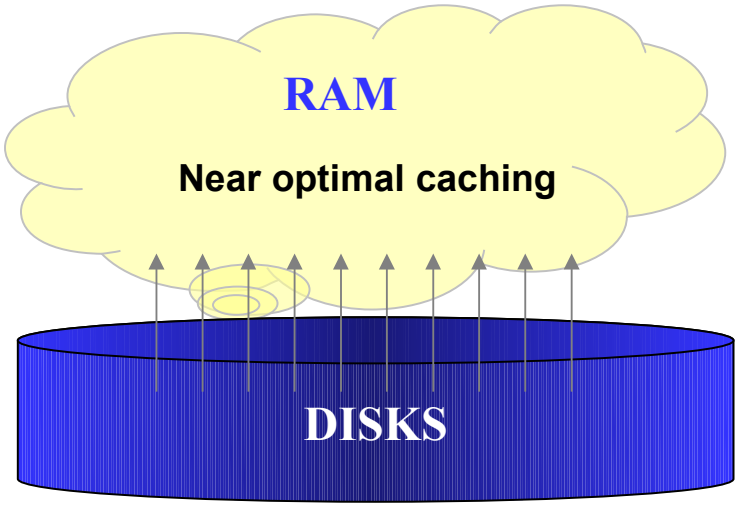
7 Big Ideas: 5 Column Store

- **Minimal I/O**
  - Only perform I/O on the columns and values that match query
  - As queries progresses through a pipeline the working set of pages is reduced
- **Work performed directly on columns**
  - Predicates, joins, scans, etc. all work on individual columns
  - Rows are not materialized until absolutely necessary to build result set
- **Improved memory density**
  - Columnar data kept compressed in memory
- **Extreme compression**
  - Packing more data values into very small amount of memory or disk
- **Cache efficiency**
  - Data packed into cache friendly structures



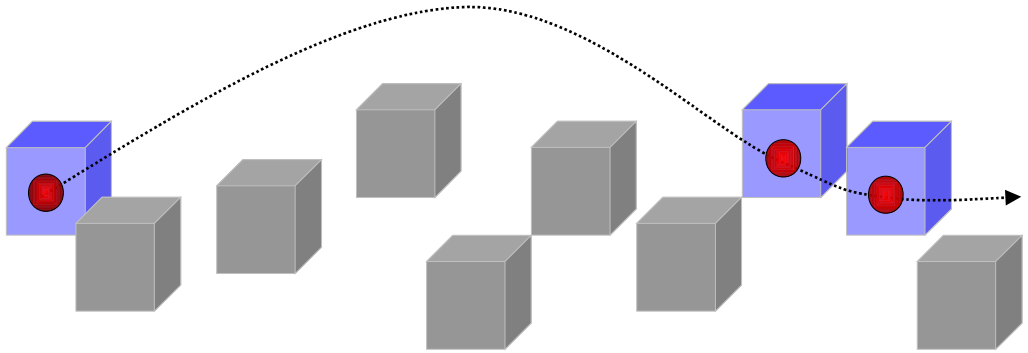
7 Big Ideas: 6 Scan-Friendly Memory Caching

- **New algorithms cache in RAM effectively**
- **High percent of interesting data fits in memory**
  - We leave the interesting data in memory with the new algorithms
- **Data can be larger than RAM**
  - No need to ensure all data fits in memory
  - Optimization for in memory and I/O efficiency



7 Big Ideas: 7 Data Skipping

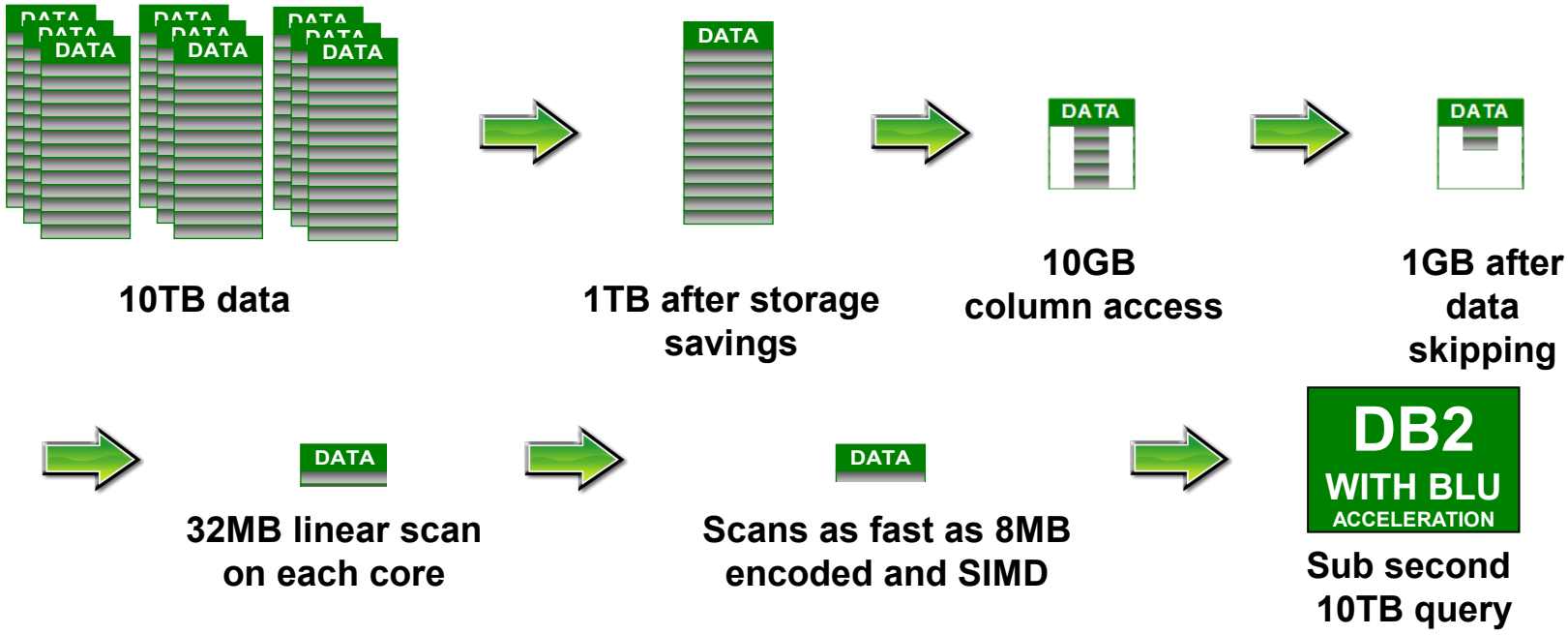
- **Automatic detection of large sections of data that do not qualify for a query and can be ignored**
- **Order of magnitude savings in all of I/O, RAM, and CPU**
- **No DBA action to define or use – truly invisible**
  - Persistent storage of min. and max. values for sections of data values



7 Big Ideas: How DB2 with BLU Acceleration Helps

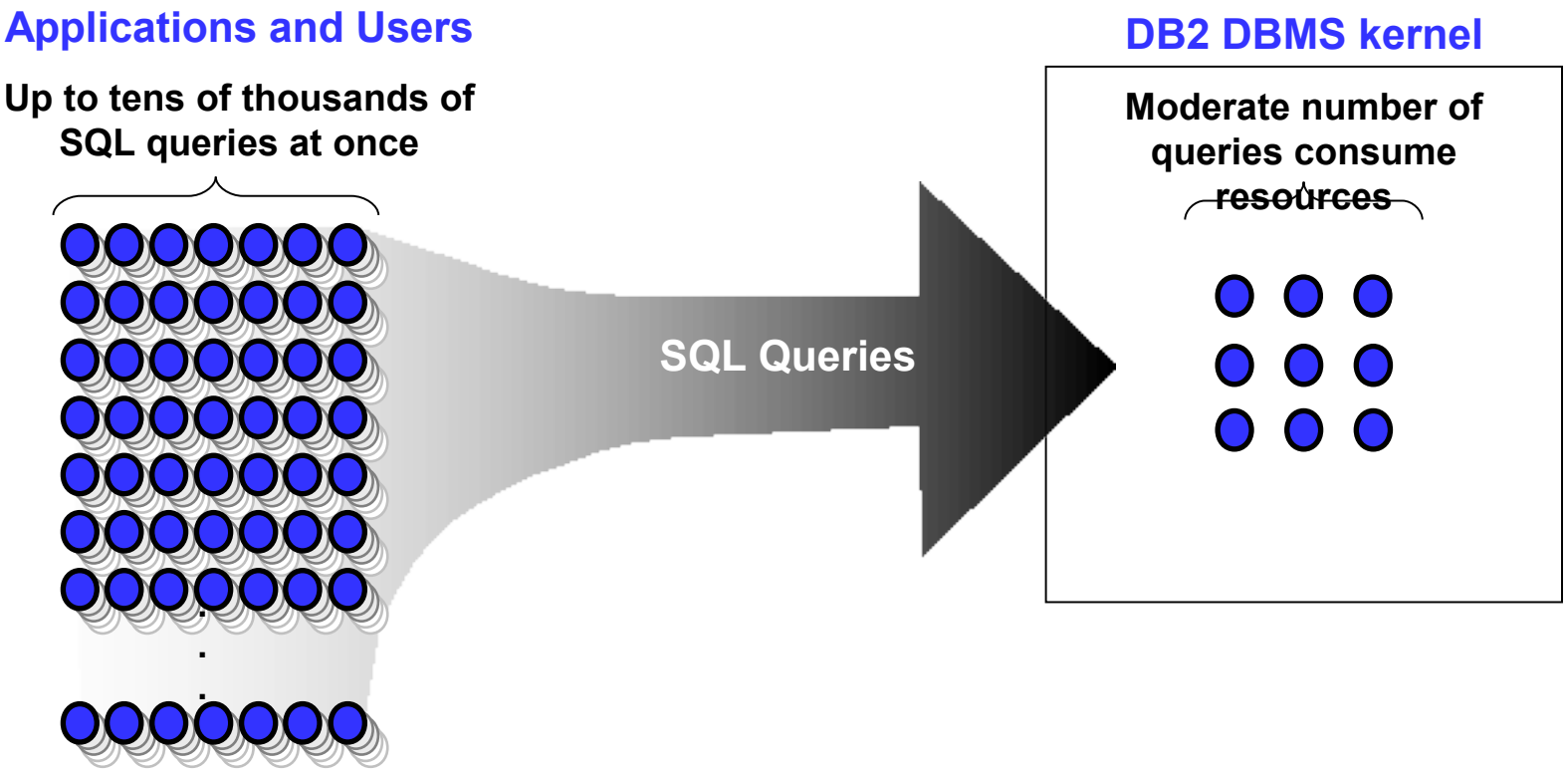
~Sub second 10TB query – An Optimistic Illustration

- **The system – 32 cores, 10TB table with 100 columns, 10 years of data**
- **The query: `SELECT COUNT(*) from MYTABLE where YEAR = '2010'`**
- **The optimistic result: sub second 10TB query! Each CPU core examines the equivalent of just 8MB of data**



Unlimited Concurrency with "Automatic WLM"

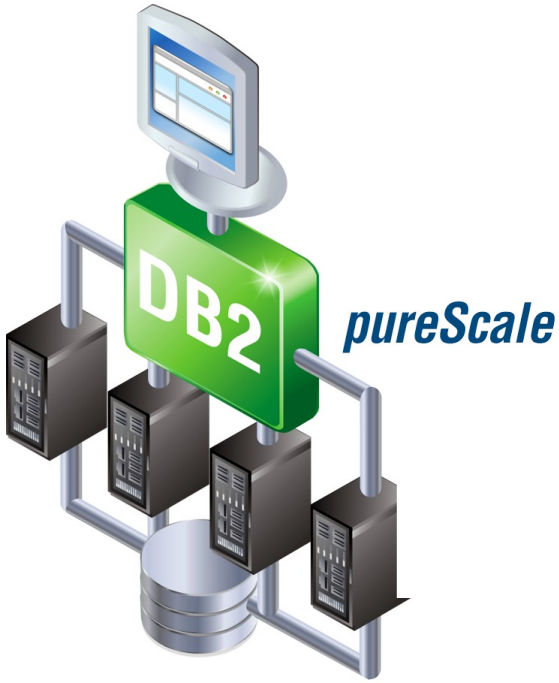
- DB2 10.5 has built-in and automated query resource consumption control
- Every additional query that runs naturally consumes more memory, locks, CPU, and memory bandwidth. In other database products more queries means more contention
- DB2 10.5 automatically allows a high level of concurrent queries to be submitted, but limits the number that consume resources at any point in time
- Enabled automatically when `DB2_WORKLOAD=ANALYTICS`



DB2 10.5 pureScale

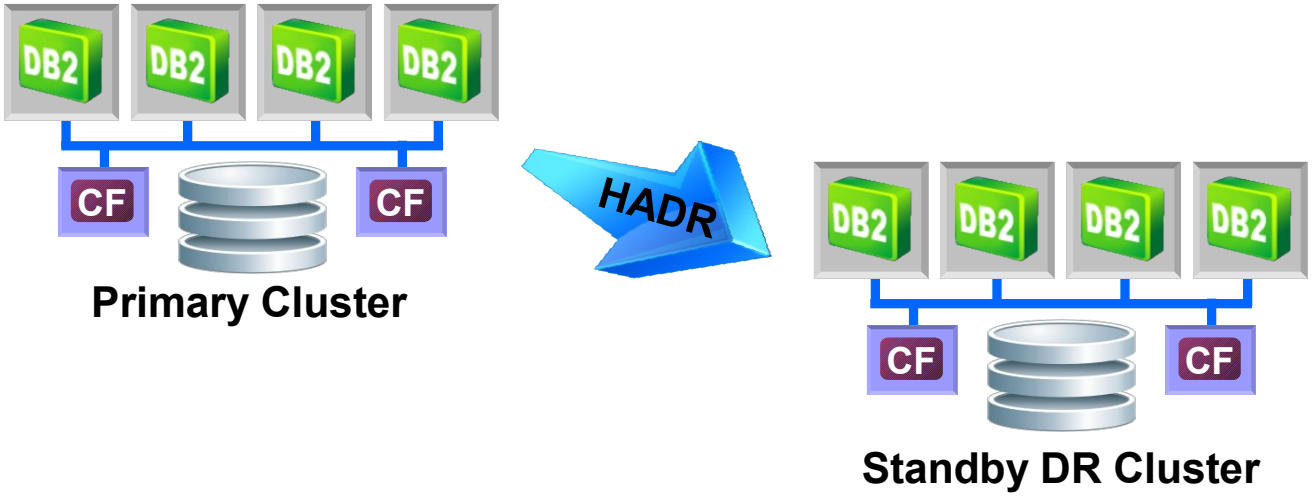
DB2 10.5 Delivers 'Always Available' Transactions
99.999% Up Time, Optimized for OLTP Workloads

- **DB2 pureScale**
  - Clustered, shared-disk architecture
  - Provides improved availability, performance, and scalability
  - Complete application transparency
  - Scales to >100 members
  - Leverages z/OS cluster technology
- **New DB2 10.5 pureScale enhancements**
  - Rich disaster recovery capabilities with HADR
  - Rolling fix pack updates



HADR in DB2 pureScale

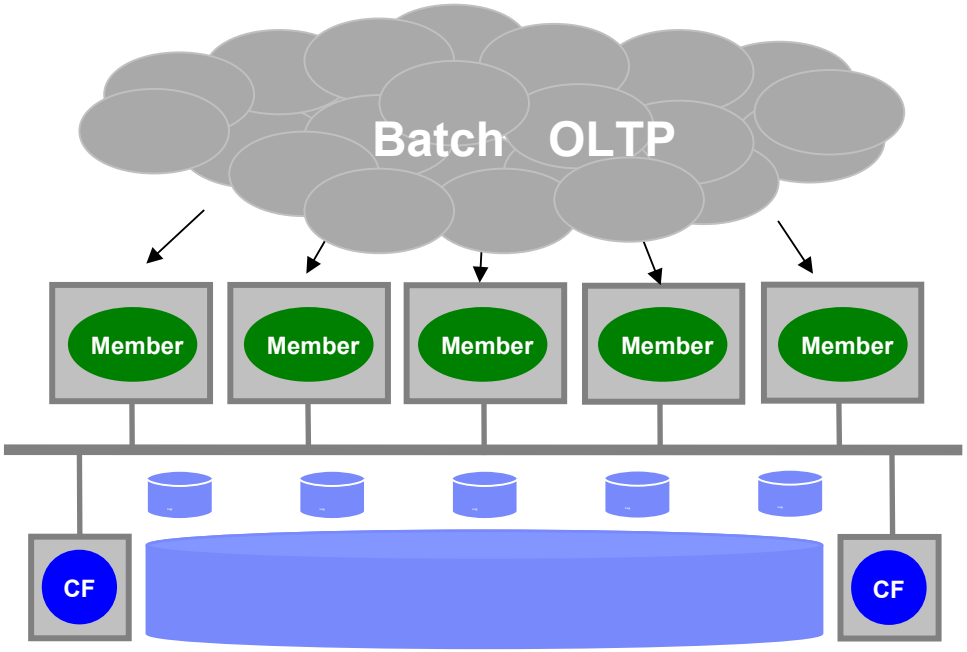
- **Integrated disaster recovery solution**
  - Simple to setup, configure, and manage
- **Support includes**
  - `ASYNC` and `SUPERASYNC` modes
    - `SYNC/NEARSYNC` under development
  - Time delayed apply
  - Log spooling
  - Both non-forced (role switch) and forced (failover) takeovers



Multi-Tenancy: Member Subsets

- **Previously, an application/tenant could only be configured to run**
  1. On one member (client affinity) or
  2. Across all members in cluster (workload balancing)

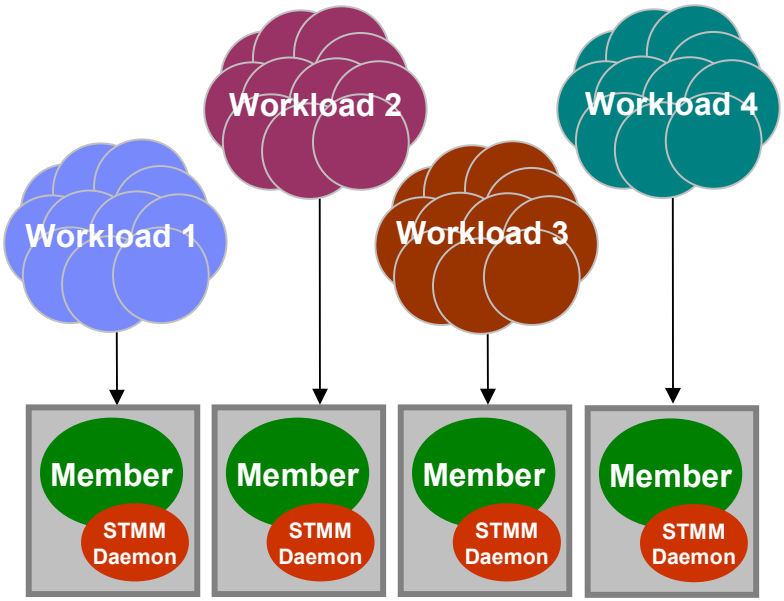
- **Can now point applications to subsets of members which enables**
  - Isolation of batch from transactional workloads
  - Multiple databases in a single instance to be isolated from each other



Multi-Tenancy: Self-Tuning Memory Management (STMM)

- **Prior DB2 pureScale STMM design**
  - Single tuning member makes local tuning decisions based on workload running on that member
    - Other member becomes tuning member in case of member failure
  - Broadcasts tuning decisions to other members
  - Works well in single homogeneous workload scenarios

- **DB2 pureScale now allows per-member STMM tuning**
  - Workload consolidation
  - Multi-tenancy
  - Batch workloads
  - Affinitized workloads



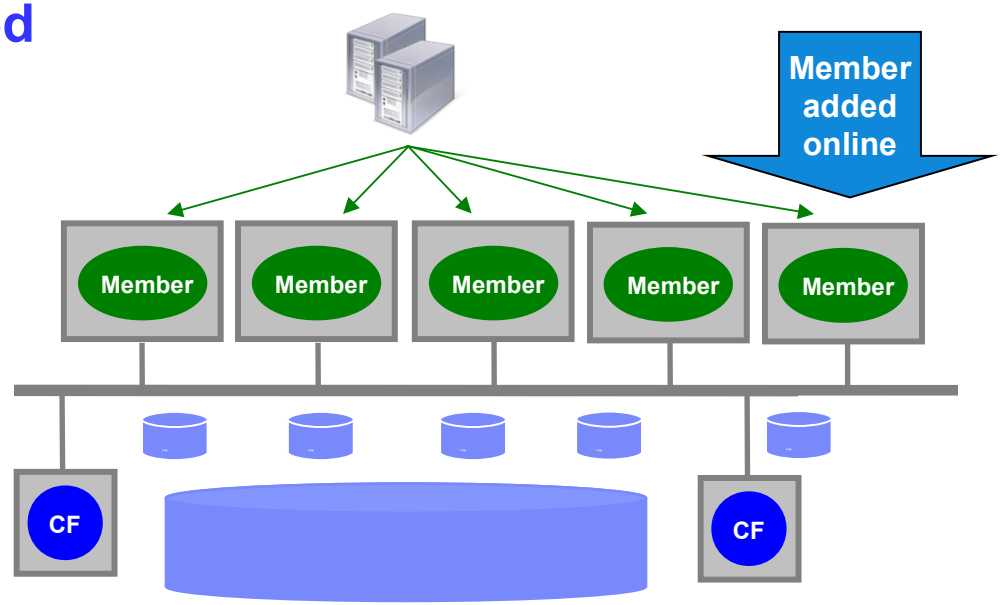
Online Add Member

- **New members can be added to an instance while it is online**
  - No impact to workloads running on existing members
  - Previously, required an outage of the entire instance to add a new member

- **No change in add member command**

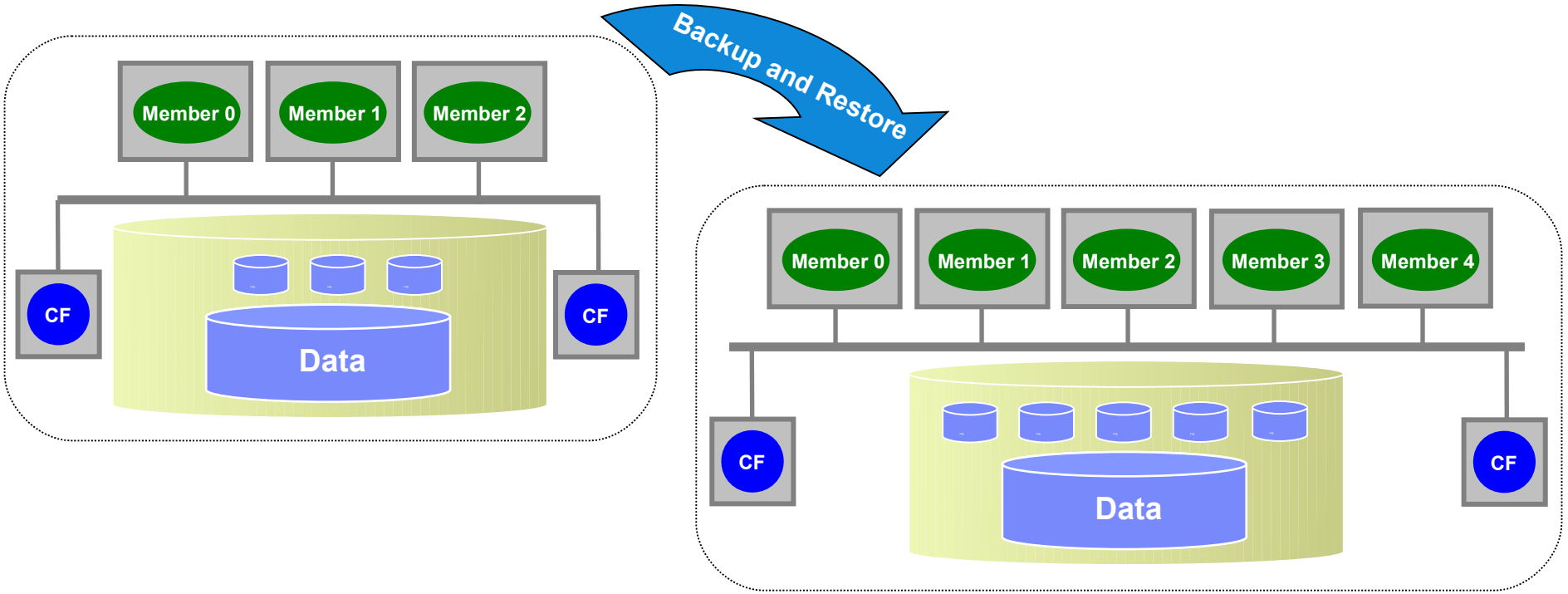
```
db2iupdt -add -m <newHost> -mnet <networkName> <instance>
```

- **Offline backup no longer needed after adding new members**



Topology-Changing Backup and Restore

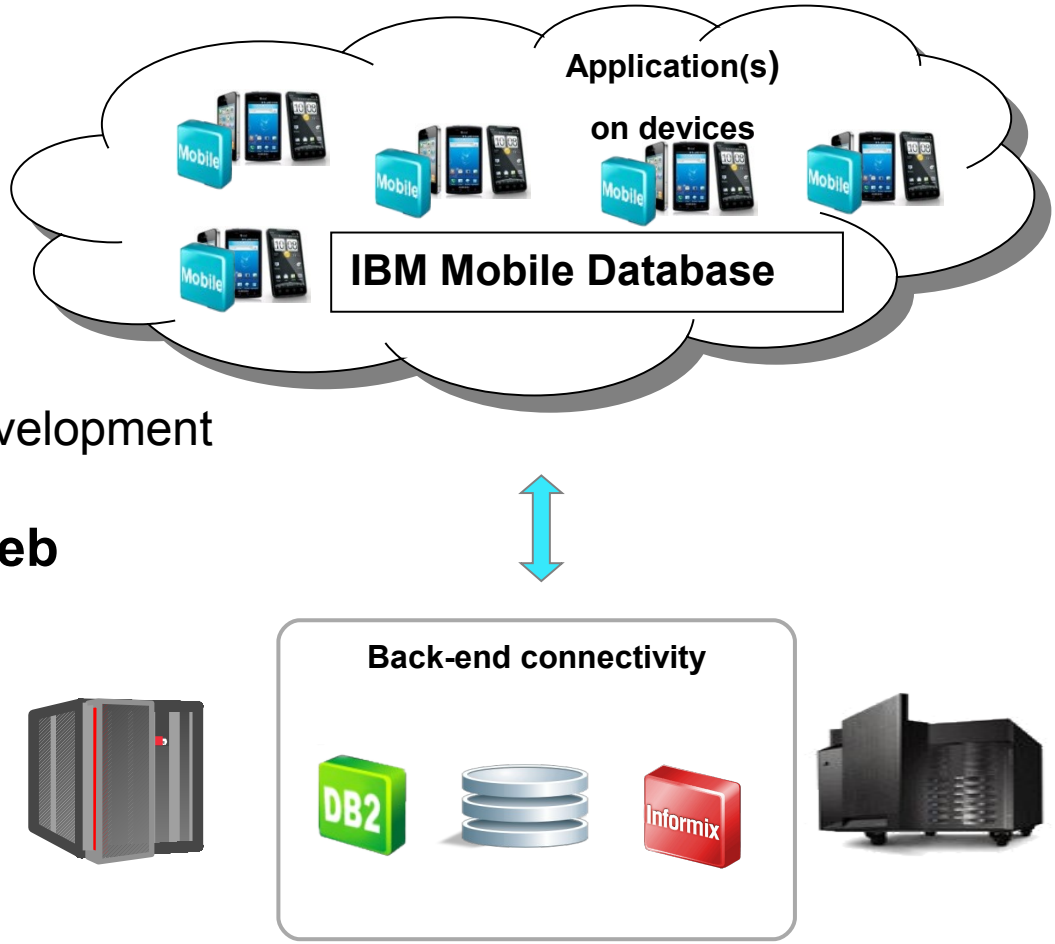
- Backup and restore between topologies with differing numbers of members
- Backup and restore from DB2 pureScale to non-DB2 pureScale (and vice-versa)



IBM Mobile Database

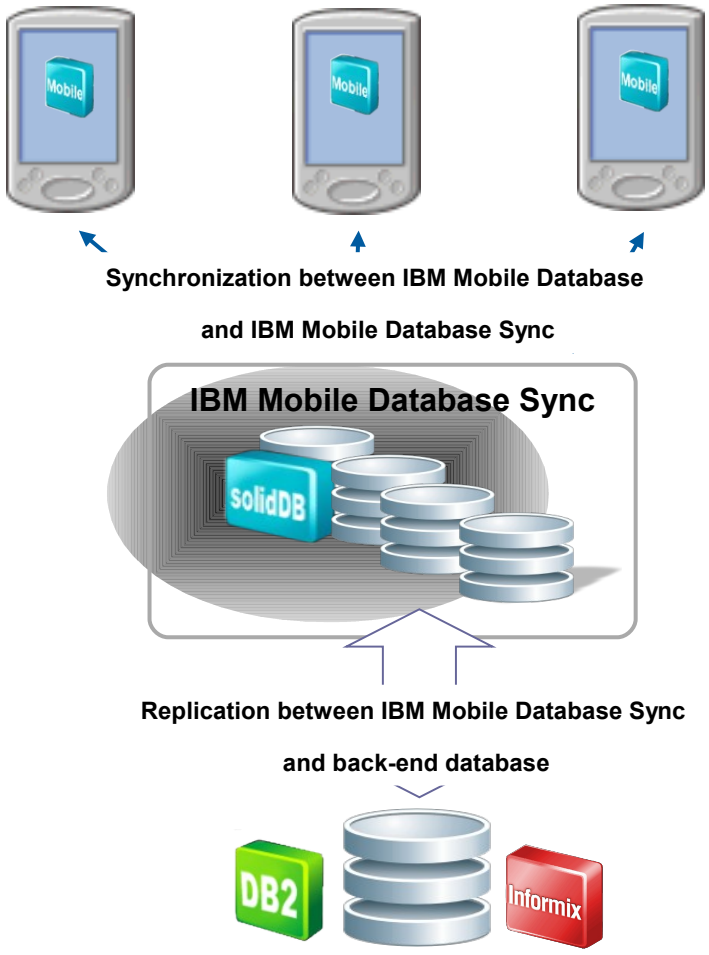
IBM Mobile Database

- **Full-featured, small footprint mobile data management solution**
  - Persistent data
  - Secure storage
  - Synchronization with back-end databases
- **Available on Android**
  - IOS and Windows Mobile in development
- **Free to download from the web**



Connectivity with Back-End Databases

- **IBM Mobile Database Sync gives rich synchronization capabilities for bidirectional communication between the IBM Mobile Database and enterprise databases**
  - IBM Mobile Database replicates data with IBM Mobile Database Sync
  - IBM Mobile Database Sync replicates data with back-end data servers
- **Back-end database can be IBM DB2 or IBM Informix**
- **Multiple solidDB systems can be used to scale the system for large number of devices**



DB2 10.5 Oracle Compatibility

Oracle Compatibility Built into DB2

Lower Transition Cost and Less Risk

Concurrency Control	→	Native support
Oracle SQL dialect	→	Native support
PL/SQL	→	Native support
PL/SQL Packages	→	Native support
Built-in package library	→	Native support
Oracle JDBC extensions	→	Native support
OCI	→	Native support
Oracle Forms	→	Through partners
SQL*Plus Scripts	→	Native support
RAC	→	DB2 pureScale

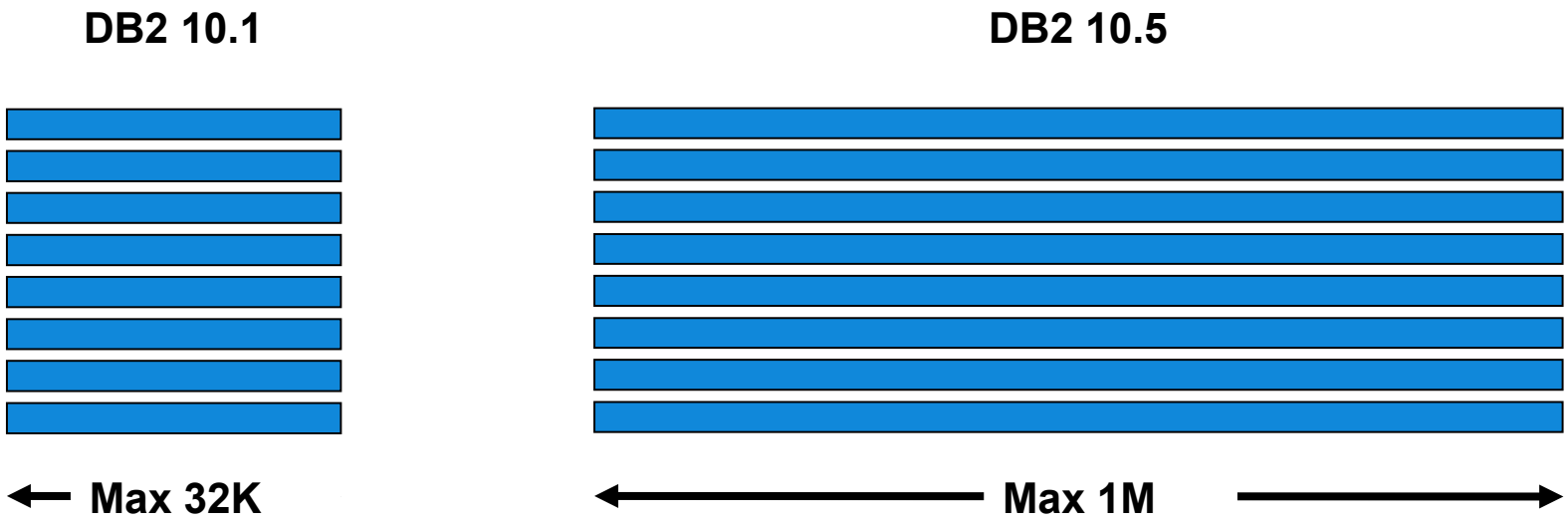
Changes are the exception. Not the rule.

Oracle Compatibility: Larger Row Widths

- **Accommodate larger strings**
  - Allow tables with up to 1MB wide rows

```
CREATE TABLE emp(name    VARCHAR(4000),
                 address VARCHAR(4000),
                 cv      VARCHAR(32000))
```

  - Allow large row `GROUP BY` and `ORDER BY` as long as key can sort



Oracle Compatibility: Additional Indexing

- **Function-based indexes**
  - Searching for computed values in a table instead of using Generated Columns
  - E.g. “Find employees without worrying about the case of their names”
    - `CREATE INDEX emp_name ON emp(UPPER(name));`
      `SELECT salary`
      `FROM emp`
      `WHERE UPPER(name) = 'MCKNIGHT';`

Name	Salary	Manages
McKnight	50000	Sales
Miller	25000	-
Van Gogh	45000	Finance
Chan	37000	-

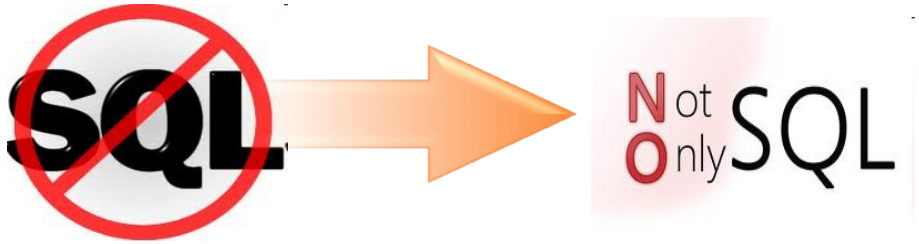
- **Indexes excluding `NULL` keys**
  - Enforce uniqueness only for `non-NULL` keys and exclude all `NULL` keys from Index
  - Compress index for `all-NULL` keys
  - Helps facilitate Oracle application migrations
    - `CREATE UNIQUE INDEX emp_manages`
      `ON emp(manages) EXCLUDE NULL KEYS`

- **Random key indexes**
  - Avoid hot index page for incrementally issued keys
    - `CREATE UNIQUE INDEX order_id ON order(id RANDOM);`

JSON Technology Preview

Background – What is NoSQL

- **A class of database management systems that depart from traditional RDBMSs**
  - Does not use SQL as the primary query language
  - Is “schema-less”
    - No rigid schema enforced by the DBMS
  - Programmer-friendly for adding fields to a document
  - Might not guarantee full `ACID` behavior
  - Often has a distributed, fault-tolerant, elastic architecture
  - Highly optimized for retrieve and append operations over great quantities of data



NoSQL DEFINITION: Next Generation Databases mostly addressing some of the points: being **non-relational, distributed, open-source** and **horizontally scalable.**

The original intention has been **modern web-scale databases.** The movement began early 2009 and is growing rapidly. Often more characteristics apply such as: **schema-free, easy replication support, simple API, eventually consistent / BASE** (not ACID), a **huge amount of data** and more. So the misleading term *"nosql"* (the community now translates it mostly with "**not only sql**") should be seen as an alias to something like the definition above. [based on 7 sources, 14 constructive feedback emails (thanks!) and 1 disliking comment . Agree / Disagree? Tell me so! By the way: this is a strong definition and it is out there here since 2009!]

LIST OF NOSQL DATABASES [currently 150]

Emergence of a growing number of non-relational, distributed data stores for massive scale data

Background - What is JSON?

- **JavaScript Object Notation**
  - Serialized form of JavaScript Objects
    - Lightweight data interchange format
    - Specified in IETF RFC 4627
    - http://www.JSON.org
- **Lightweight text interchange**
  - Designed to be minimal, portable, textural, and subset of JavaScript
    - Only 6 kinds of values!
    - Easy to implement and easy to use
- **Replacing XML as the de facto data interchange format on the web**
  - Used to exchange data between programs written in all modern programming languages
- **Self-describing, easy to understand**
  - Text format, so readable by humans and machines
  - Language independent, most languages have features that map easily to JSON

```
{
   "firstName": "John",
   "lastName" : "Smith",
   "age"      : 25,
   "address"  :
   {
      "streetAddress": "21 2nd Street",
      "city"        : "New York",
      "state"       : "NY",
      "postalCode"  : "10021"
   },
   "phoneNumber":
   [
      {
        "type"  : "home",
        "number": "212 555-1234"
      },
      {
        "type"  : "fax",
        "number": "646 555-4567"
      }
   ]
}
```

"Less is better: less we need to agree upon to interoperate, the more easily we interoperate"
JavaScript: The Good Parts, O'Reilly

JSON Technology Preview

- **Combine data from systems of engagement with traditional data in same DB2 database**
  - Best of both worlds
  - Simplicity and agility of JSON + enterprise strengths of DB2

- **Store data from web/mobile apps in it's native form**
  - New web applications use JSON for storing and exchanging information
  - It is also the preferred data format for mobile application backends

- **Move from development to production in no time!**
  - Ability to create and deploy flexible JSON schema
  - Gives power to application developers by reducing dependency on IT; no need to pre-determine schemas and create/modify tables
  - Ideal for agile, rapid development and continuous integration

DB2 10.5 Packaging Simplification

DB2 10.5 Simplifies Product Packaging

One Set of Editions for Both Transactional and Warehouse Workloads

	Departmental Market	Enterprise Market
Advanced function	**DB2 Advanced Workgroup Server Edition** • For small OLTP and analytic deployments • Primarily used in department environments within large enterprises or SMB/MM deployments • Limited by TB, memory, sockets and cores • Supports BLU, pS and DPF deployment models	**DB2 Advanced Enterprise Server Edition** • For Enterprise Class OLTP and/or analytic deployments • Targeting full enterprise/full data centre requirements • No TB, memory, socket or core limit • Supports BLU, pS and DPF deployment models
Core function	**DB2 Workgroup Server Edition** • Entry level offering • Single server for less intense workloads • Limited by TB, memory, sockets and cores • No support for BLU, pS or DPF deployment models	**DB2 Enterprise Server Edition** • Entry level offering • Single server for enterprise/more intense workloads • No TB, memory, socket or core limit • No support for BLU, pS or DPF deployment models
	Limited capacity	*Full capacity*

DB2 Developer Edition	DB2 CEO
DB2 Express and DB2 Express-C	DB2 Advanced CEO