

DB2[®] 10.5

with BLU Acceleration



George Baklarz, PhD
IBM DB2 Worldwide Technical Evangelist



DB2® 10.5

with BLU Acceleration

**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

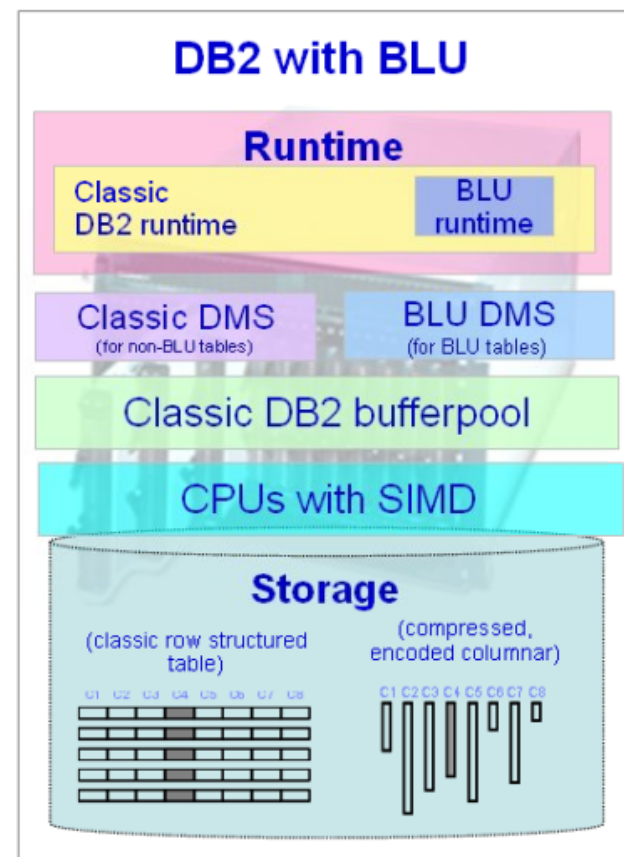
DB2 10.5 with BLU Acceleration



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

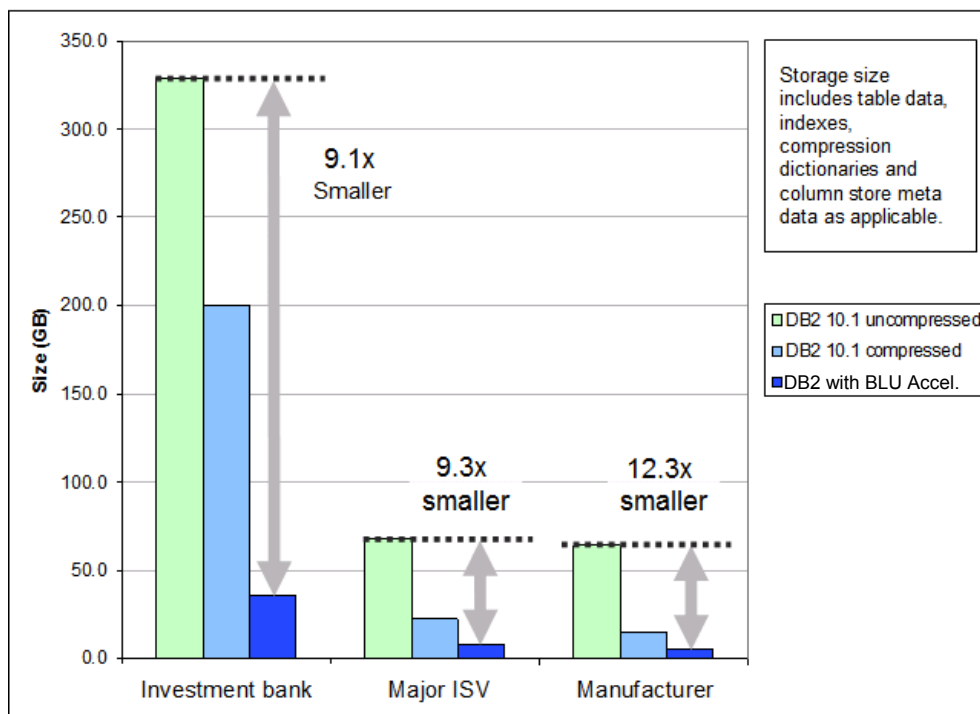


“It was amazing to see the faster query times compared to the performance results with our row-organized tables. The performance of four of our queries improved by over 100-fold! The best outcome was a query that finished 137x faster by using BLU Acceleration.”

- 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**



Seamless Integration into DB2

- **Built seamlessly into DB2 – Integration and coexistence**
 - Column-organized tables can coexist with existing, traditional, tables
 - Same schema, same storage, same memory
 - Integrated tooling support
 - Optim Query Workload Tuner (OQWT) recommends BLU Acceleration deployments

- **Same SQL, language interfaces, administration**
 - Column-organized tables or combinations of column-organized and row-organized tables can be accessed within the same SQL statement

- **Dramatic simplification – Just “Load and Go”**
 - Faster deployment
 - Fewer database objects required to achieve same outcome
 - Requires less ongoing management due to it's optimized query processing and fewer database objects required
 - Simple migration
 - Conversion from traditional row table to BLU Acceleration is easy
 - DB2 Workload Manager (WLM) identifies workloads to tune
 - Optim Query Workload Tuner recommends BLU Acceleration table transformations
 - Users only notice speed up; DBA's only notice less work!
 - Management of single server solutions less expensive than clustered solutions

Super fast, Super Easy – Create, Load, and Go!

Database Design and Tuning

1. Decide on partition strategies
2. Select Compression Strategy
3. Create Table
4. Load data
5. Create Auxiliary Performance Structures
 - Materialized views
 - Create indexes
 - B+ indexes
 - Bitmap indexes
6. Tune memory
7. Tune I/O
8. Add Optimizer hints
9. Statistics collection

Repeat

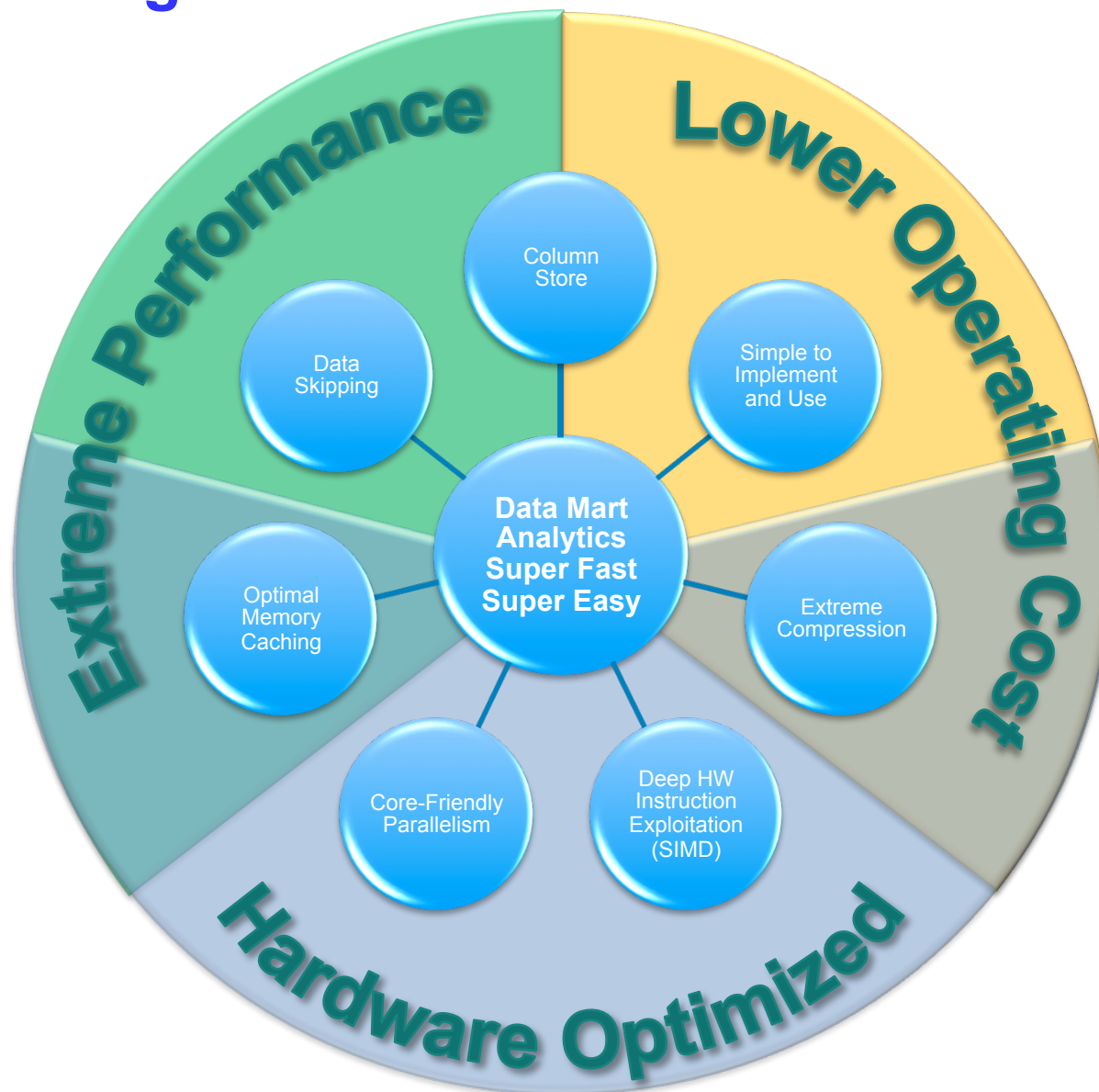


DB2 with BLU Acceleration

1. Create Table
2. Load data



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: 2 Compute Friendly Encoding and Compression

- **Massive compression with approximate Huffman encoding**
 - More frequent the value, the fewer bits it takes
- **Register-friendly encoding dramatically improves efficiency**
 - Encoded values packed into bits matching the register width of the CPU
 - Fewer I/Os, better memory utilization, fewer CPU cycles to process

LAST_NAME Encoding

Johnson	
Smith	
Smith	
Smith	
Smith	
Johnson	
Smith	
Gilligan	
Sampson	
Smith	

Packed into register length













7 Big Ideas: 2 Data Remains Compressed During Evaluation

- **Encoded values do not need to be decompressed during evaluation**
 - Predicates and joins work directly on encoded values

```
SELECT COUNT(*) FROM T1 WHERE LAST_NAME = 'SMITH'
```

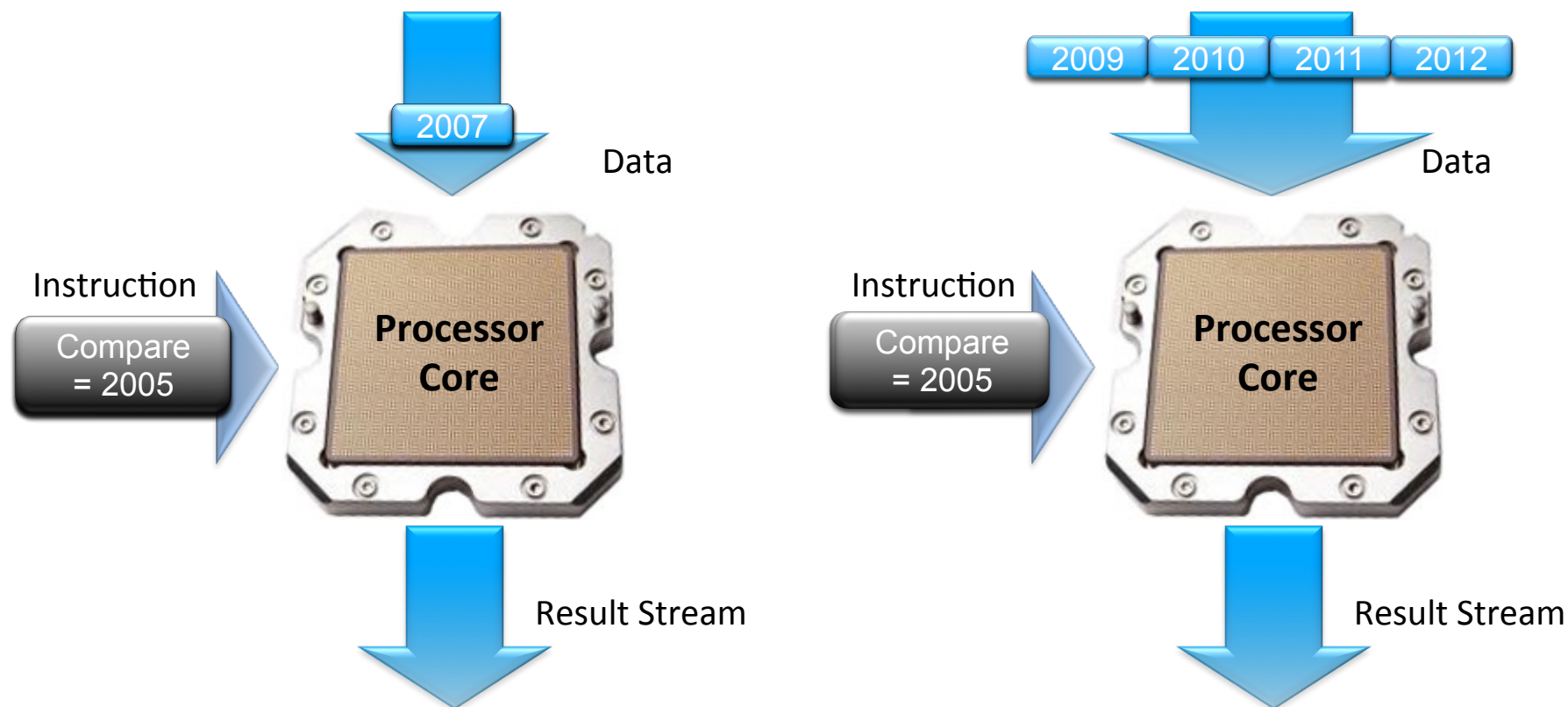
LAST_NAME Encoding

Johnson	
Smith	
Smith	
Smith	
Smith	
Johnson	
Smith	
Gilligan	
Sampson	
Smith	



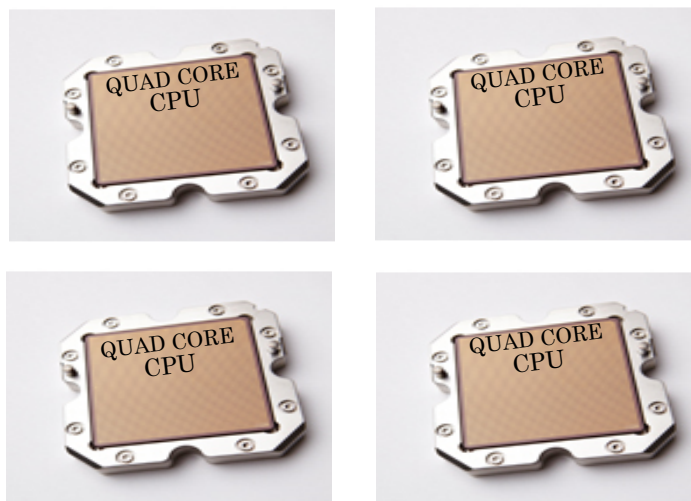
7 Big Ideas: ③ Multiply the Power of the CPU

- Performance increase with Single Instruction Multiple Data (SIMD)
- Using hardware instructions, DB2 with BLU Acceleration can apply a single instruction to many data elements simultaneously
 - Predicate evaluation, joins, grouping, arithmetic



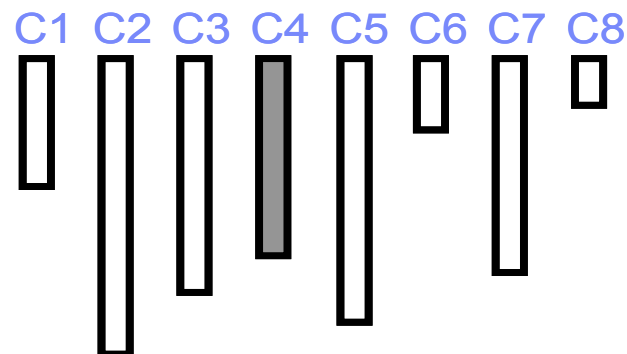
7 Big Ideas: ④ Core-Friendly Parallelism

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



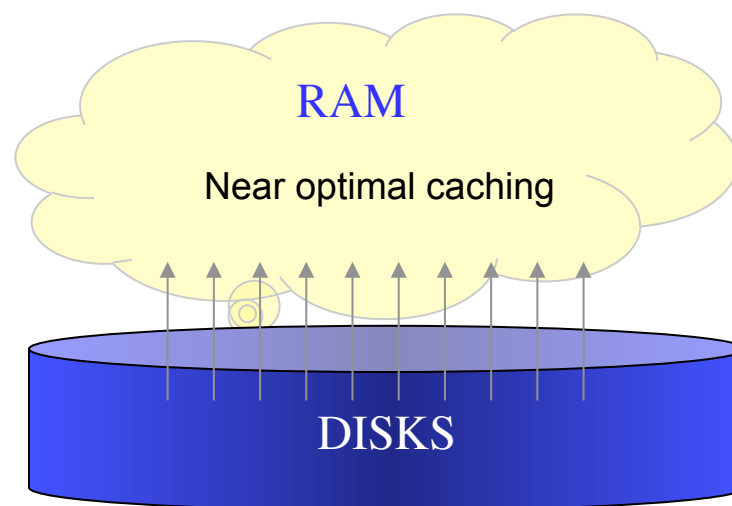
7 Big Ideas: 5 Column Store

- **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



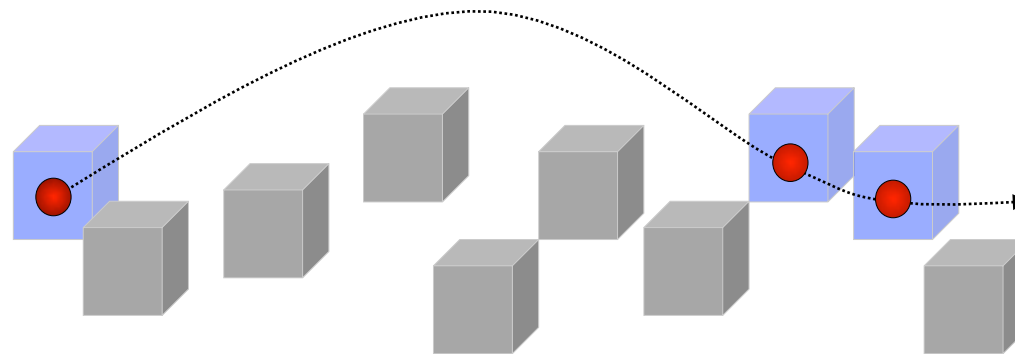
7 Big Ideas: ⑥ Scan-Friendly Memory Caching

- **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

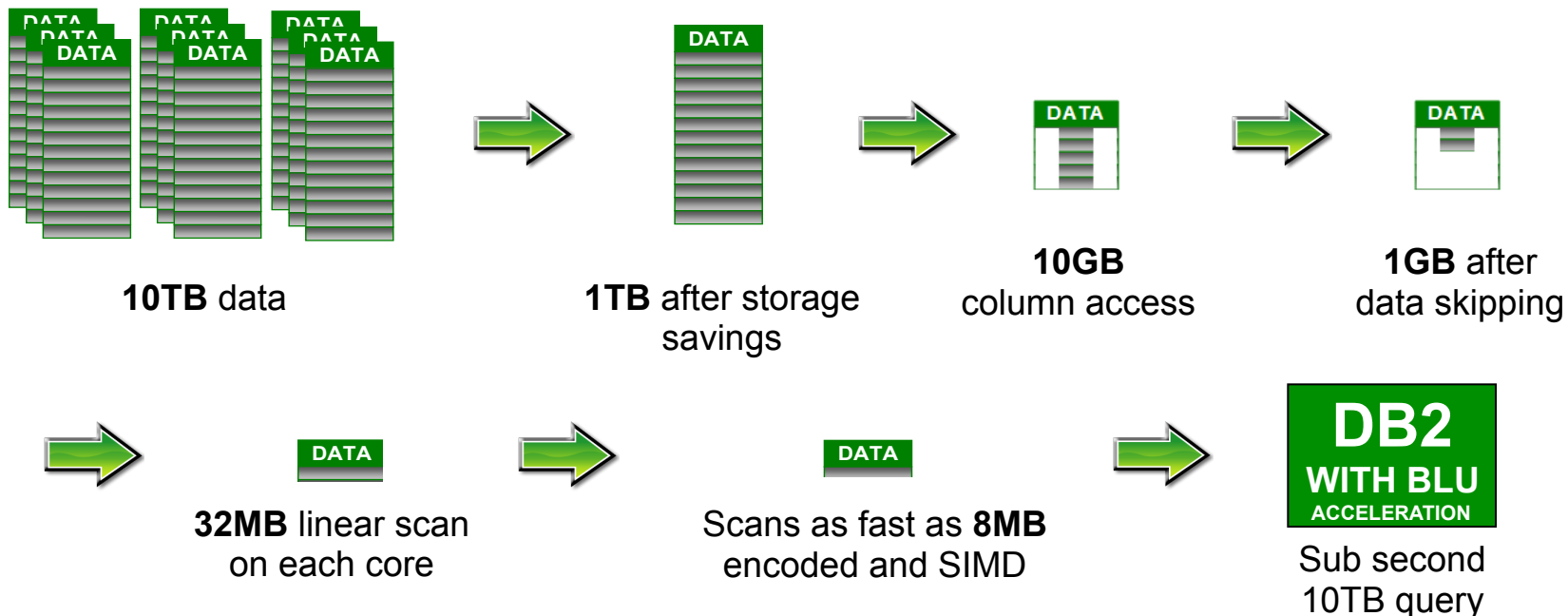
- Automatic detection of large sections of data that do not qualify for a query and can be ignored
- Order of magnitude **savings in all of I/O, RAM, and CPU**
- **No DBA action to define or use – truly invisible**
 - Persistent storage of min. and max. values for sections of data values



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

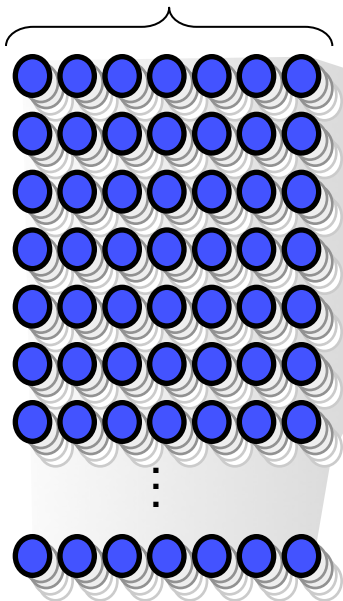


Unlimited Concurrency with “Automatic WLM”

- 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`

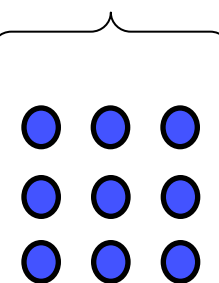
Applications and Users

Up to tens of thousands of SQL queries at once



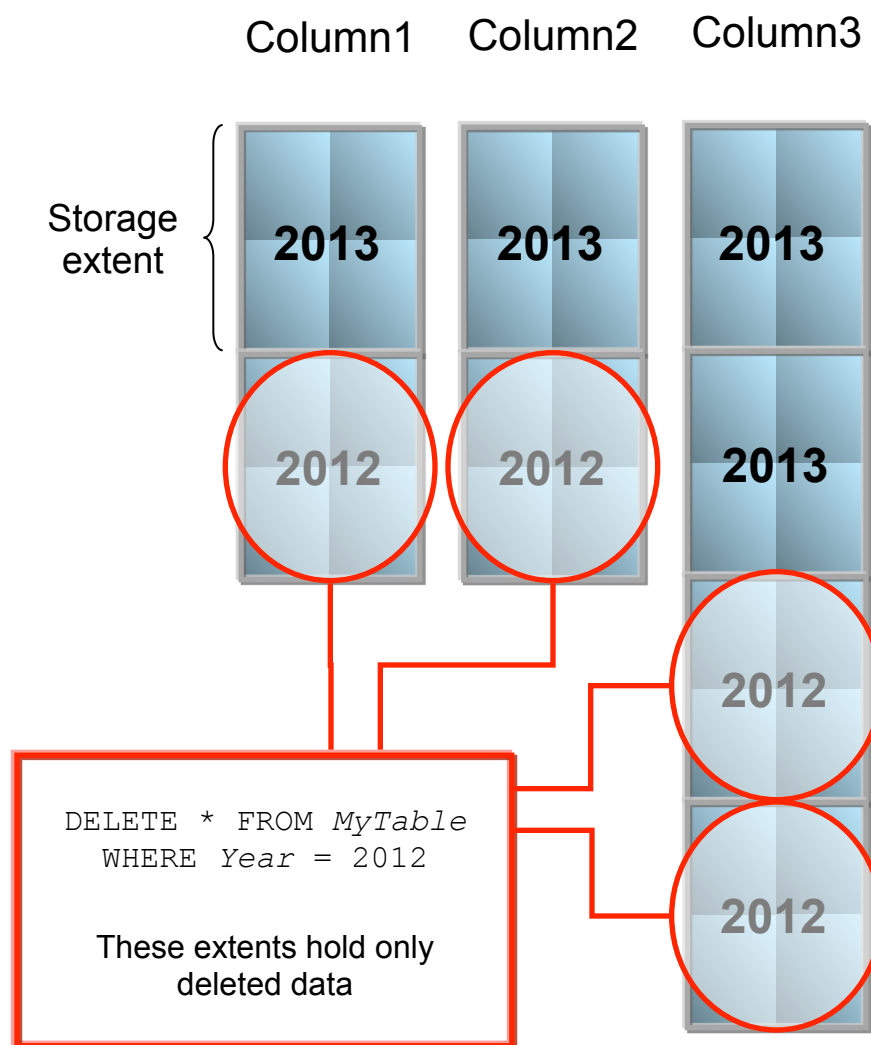
DB2 DBMS kernel

Moderate number of queries consume resources



Automatic Space Reclaim

- **Automatic space reclamation**
 - Frees extents with no active values
 - The storage can be subsequently reused by any table in the table space
- **No need for costly DBA space management and REORG utility**
- **Enabled out-of-the box for column-organized tables when DB2_WORKLOAD=ANALYTICS**
- **Space is freed online while work continues**
- **Regular space management can result in increased performance of RUNSTATS and some queries**



DB2 with BLU Acceleration Early Customer Quotes



“Using DB2 10.5 with BLU Acceleration, our storage consumption went down by about 10x compared to our storage requirements for uncompressed tables and indexes. In fact, I was surprised to find a 3x increase in storage savings compared to the great compression that we already observed with Adaptive Compression on the DB2 10.5 server.”

- Kent Collins, Database Solutions Architect, BNSF Railway



*“One of the things I really like about BLU Acceleration is that it enables me to put **column-organized tables beside row-organized tables in the same database.** In our mixed environment, we realized an amazing **10-25x reduction in the storage requirements for the database when taking into account the compression ratios, along with all the things I no longer need to worry about: indexes, aggregates, and so on.**”*

-Andrew Juarez, Lead SAP Basis and DBA

DB2 10.5 pureScale

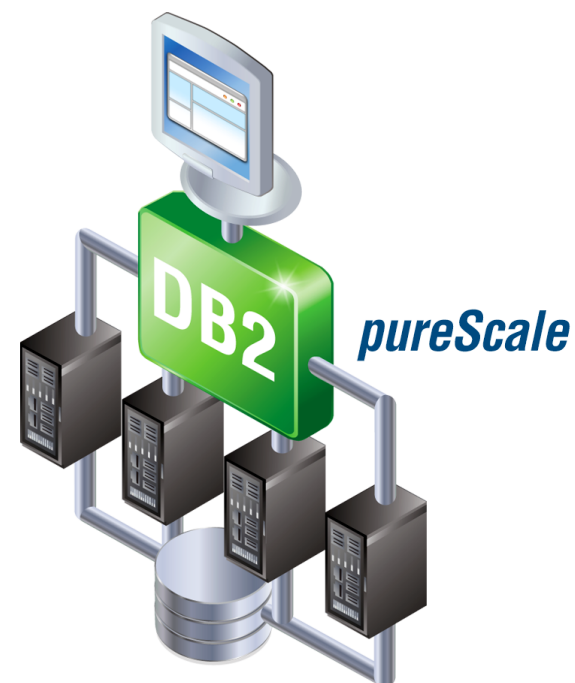


DB2 10.5 pureScale Enhancements

Enhanced availability, optimized for OLTP Workloads

- **DB2 pureScale**
 - Robust infrastructure for OLTP workloads
 - Provides improved availability, performance and scalability
 - Transparent scalability beyond 100 nodes¹
 - Leverages z/OS cluster technology

- **NEW pureScale enhancements**
 - Online member add
 - No planned downtime required for pureScale maintenance updates²
 - HADR designed to failover in seconds³



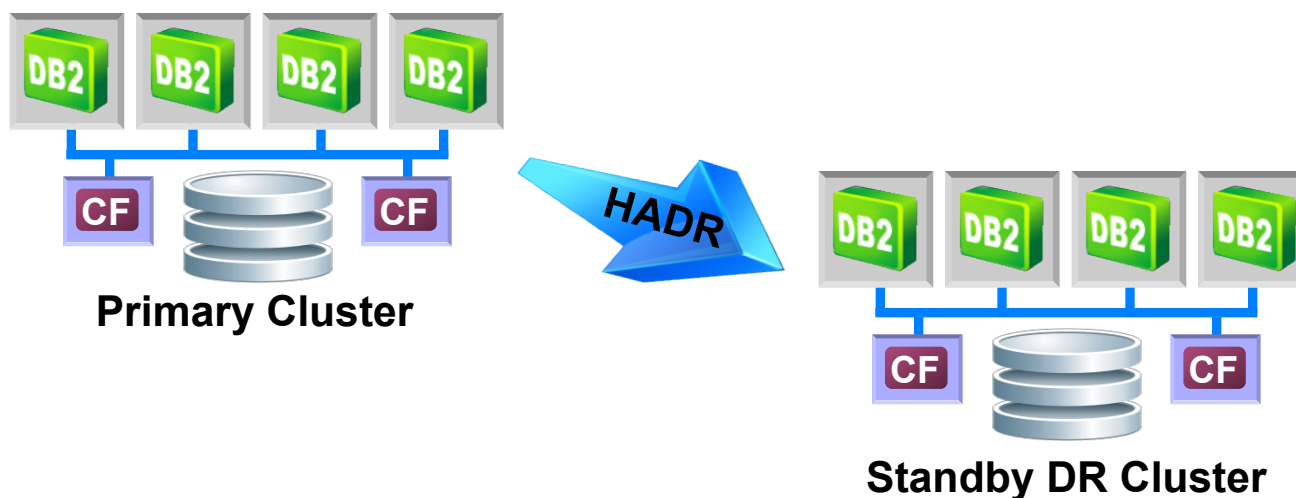
1. Available with DB2 Advanced Enterprise Server Edition.

2. Based on IBM design for normal operation with rolling maintenance updates of DB2 server software on a pureScale cluster. Individual results will vary depending on individual workloads, configurations and conditions, network availability and bandwidth.

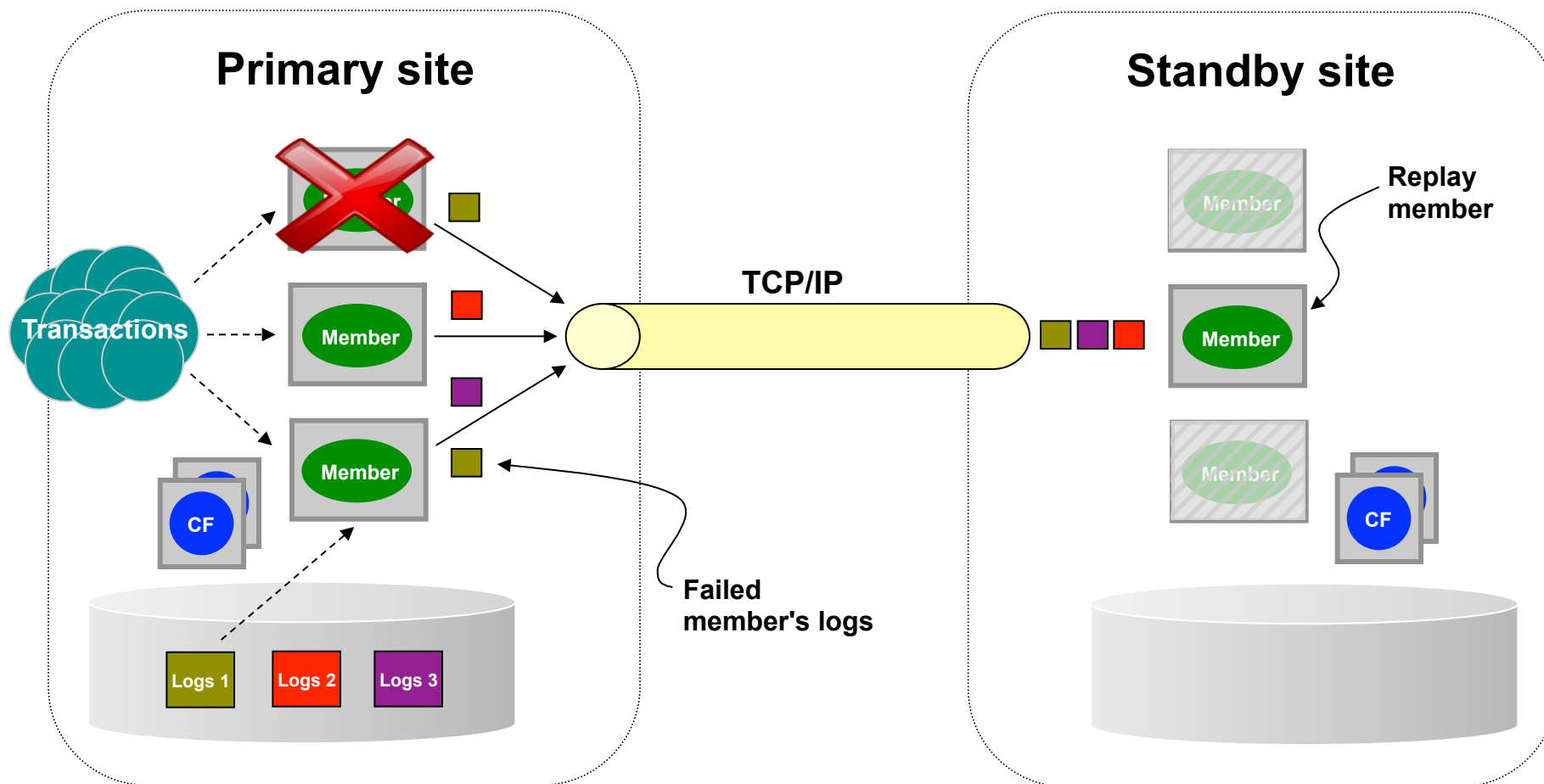
3. Based on IBM design for normal operation under typical workload using HADR and pureScale clusters. Individual results will vary depending on individual workloads, configurations, and conditions, network availability and bandwidth.

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



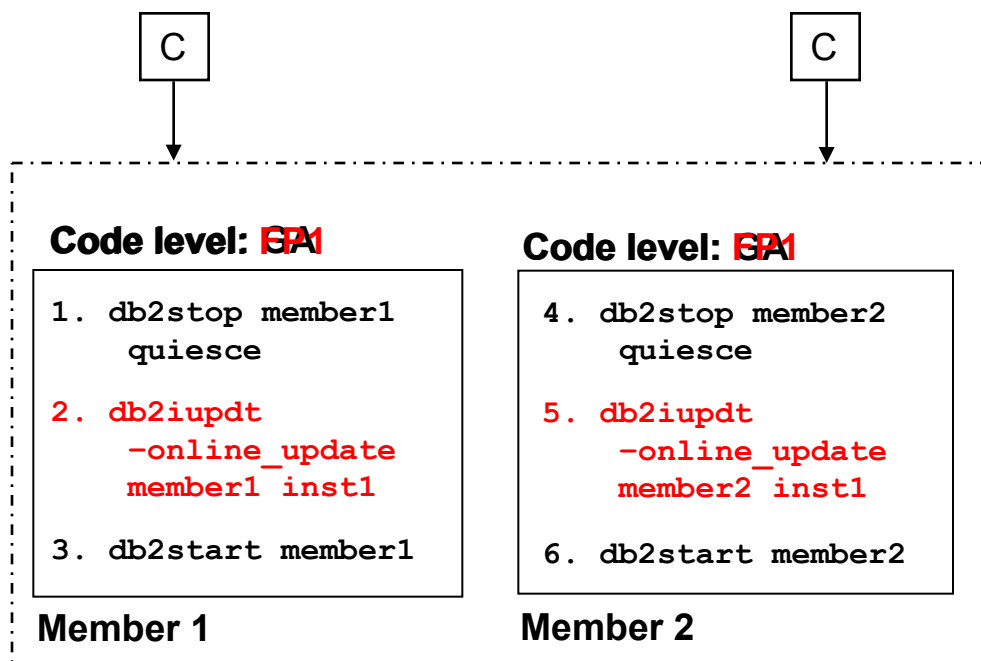
HADR in DB2 pureScale: Example



Rolling Fix Pack Updates

- **DB2 pureScale fix packs can be applied in an online rolling fashion**
 - Transparently install DB2 pureScale fix packs with no outage
- **New options for `db2iupdt` to do to online update, do a pre-commit check, and to subsequently commit the changes**
- **Includes updates of CFs and members**

Rolling Fix Pack Updates – Example



7. db2iupdt -commit_new_level inst1



Two member cluster (each at GA level) with clients (C) connecting into each member

1. Member 1 is quiesced – clients all move to Member 2
2. DB2 binaries updated on Member 1
3. Member 1 started again and a portion of the clients get rerouted to Member 1 to balance the workload
4. Member 2 is quiesced – clients all move to Member 1
5. DB2 binaries updated on Member 2
6. Member 2 started again and a portion of the clients get rerouted to member 2 to balance the workload

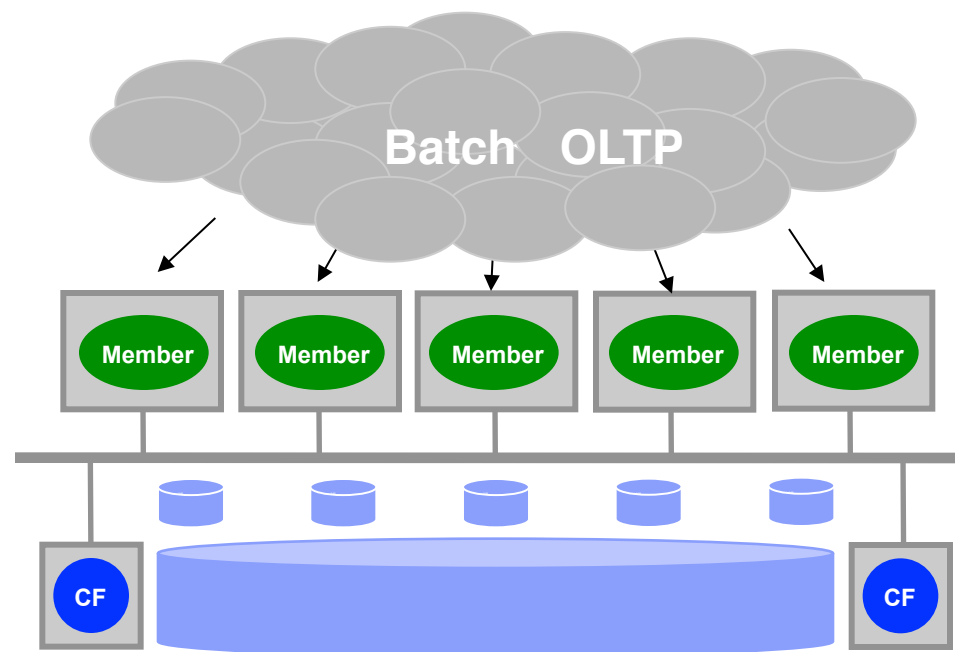
At this point, code is at FP1 level, but can't use any new FP1 features; can test stability and roll down to GA level if necessary

7. Updates are committed

The instance is now completely running at FP1 and new features can be used; cannot roll down to GA any longer.

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



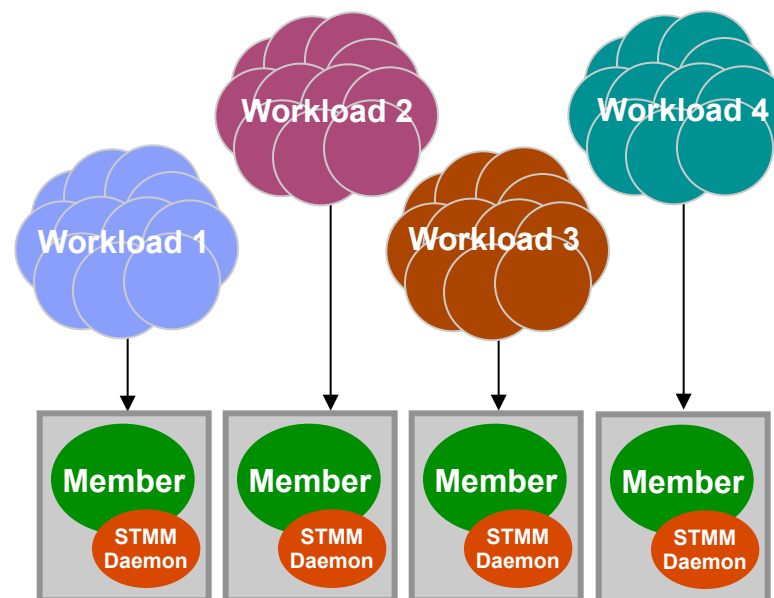
Multi-Tenancy: Self-Tuning Memory Management (STMM)

▪ 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

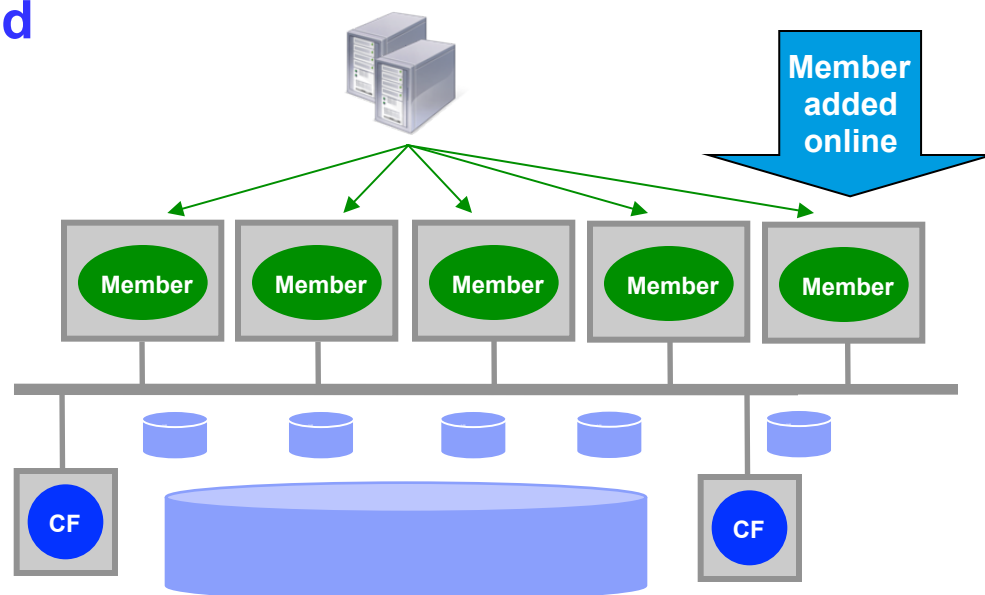


Online Add Member

- **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**

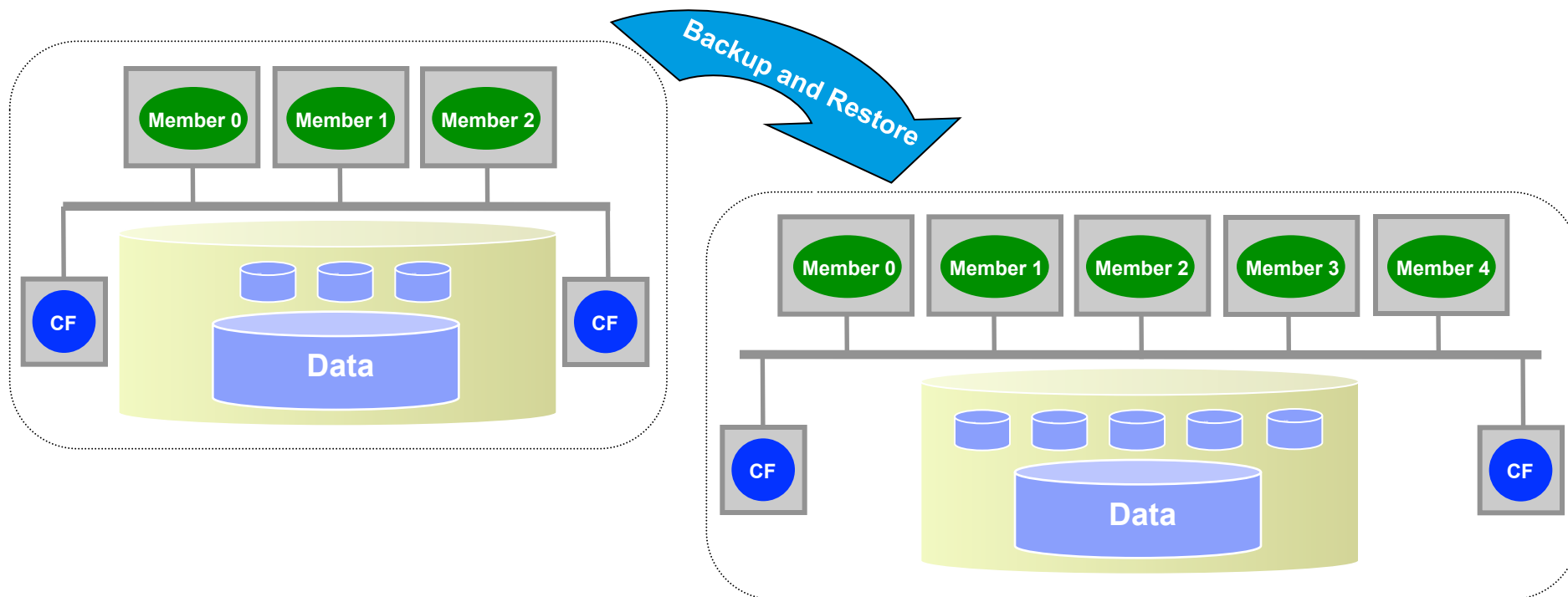
```
db2iupdt -add -m <newHost> -mnet <networkName> <instance>
```

- **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)



DB2 10.5 Oracle Compatibility



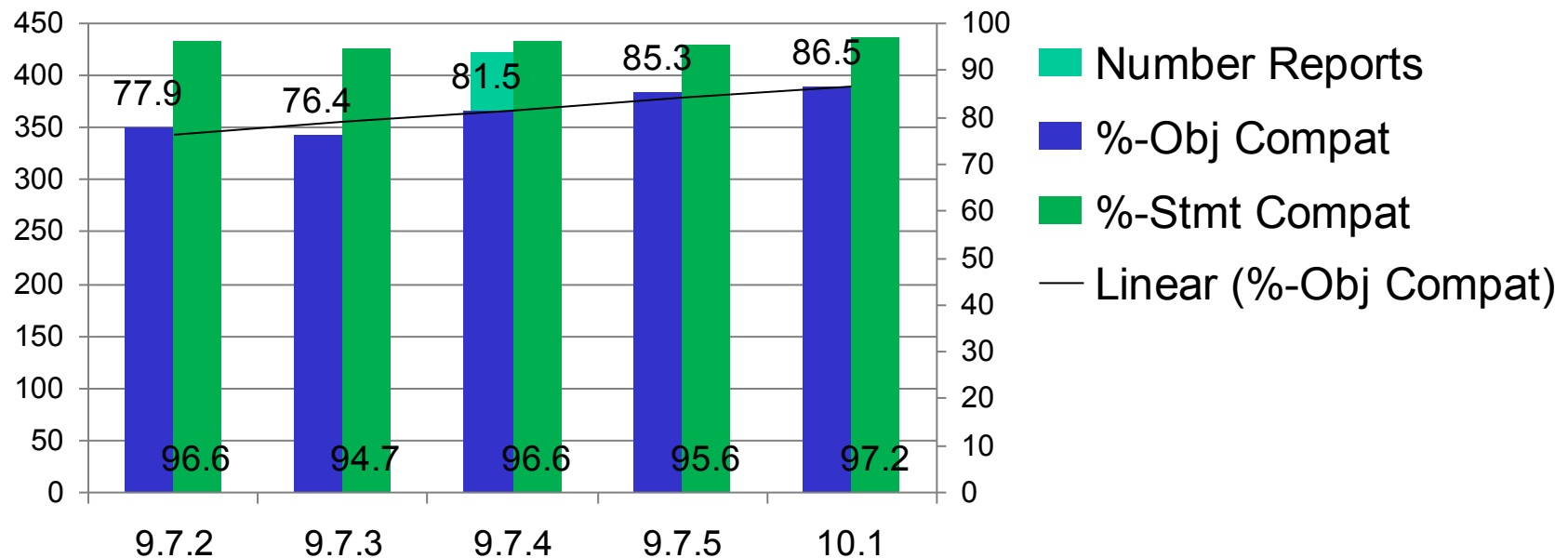
Oracle Compatibility Built into DB2

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.

Application Compatibility Over Time



- **Data is based on DCW (Database Conversion Workbench) DB2 reports in the database**
- **Compatibility is improved**
 - More and more complex applications
- **DB2 10.5 provides > 99% compatibility**

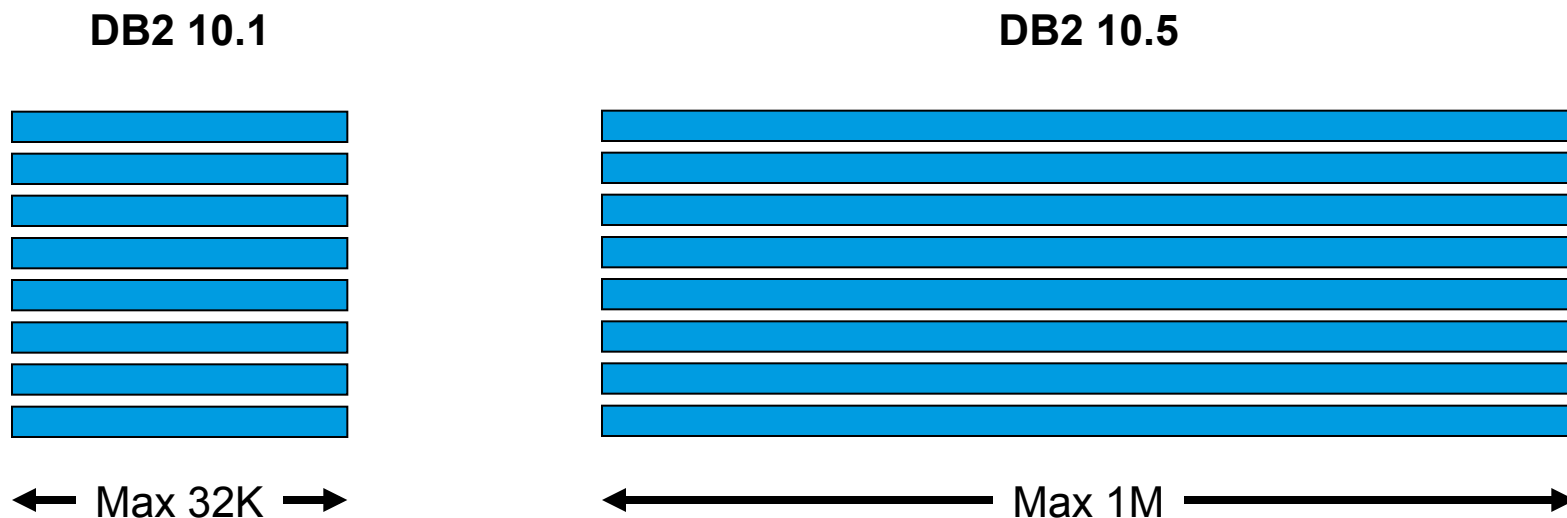
Oracle Compatibility: Larger Row Widths

- **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);
```

Oracle PL/SQL Compatibility

- **Create distinct type with weak type rules**
 - Removes limitation of existing distinct types not having weak typing
 - Optional check constraint
 - Optional `NOT NULL` constraint
 - Constraints enforced on assignment

- **Pipelined table function**
 - Introduce a new `PIPE` statement which returns a row to caller, but continues at next statement if caller wants another row
 - Incrementally produce a result set for consumption on demand

- **Ad-hoc federated table access**
 - Support ad-hoc reference to remote table using server in the identifier
 - Reach out to a table in a remote database

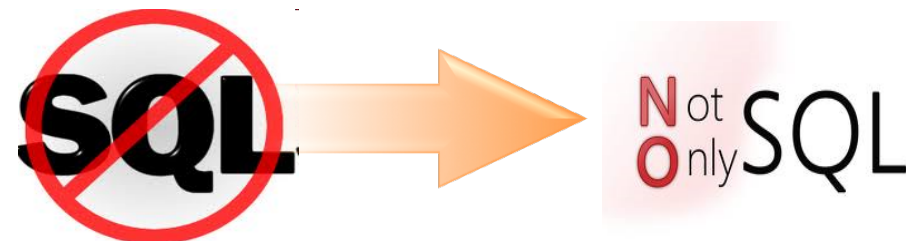
- **Function library extensions**
 - Updates to various built-in functions for improved compatibility support

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, textual, 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

The JSON-XML Shift

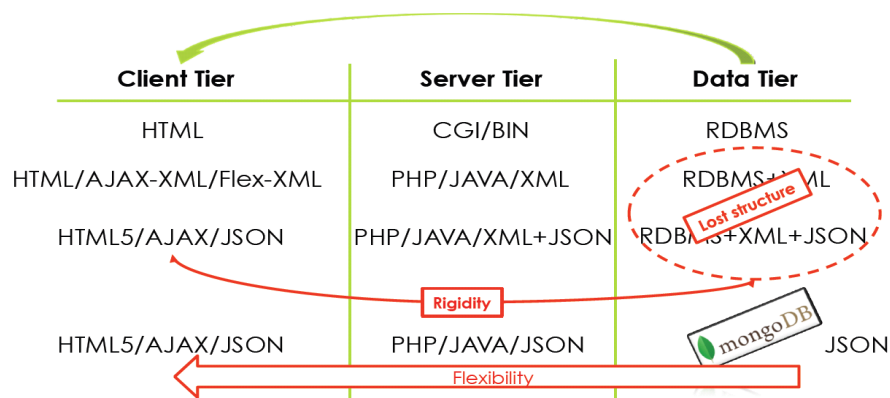
- **Developers find it easier to move data back and forth without losing information in JSON vs. XML**
 - XML is more powerful and more sophisticated than JSON
 - But JSON found to be 'good enough' → It makes programming tasks easier

- **By the time RDBMS world got very sophisticated with XML, developers had chosen JSON**
 - Application shift lead to emergence of database that store data in JSON (i.e., MongoDB)
 - JSON on the server side is appealing for developers using JSON on the client tier side



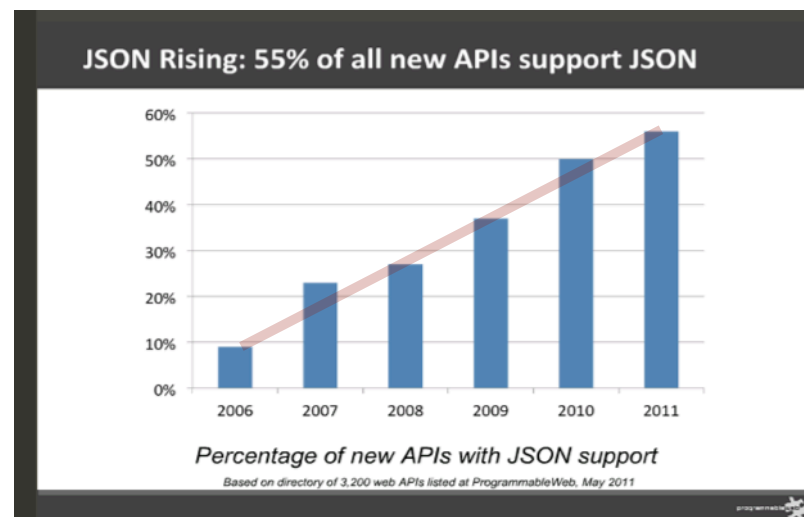
The Technology

- ▣ Each technology is awesome & powerful in itself But...
- ▣ ...Mixing technologies and solving integration issues are not



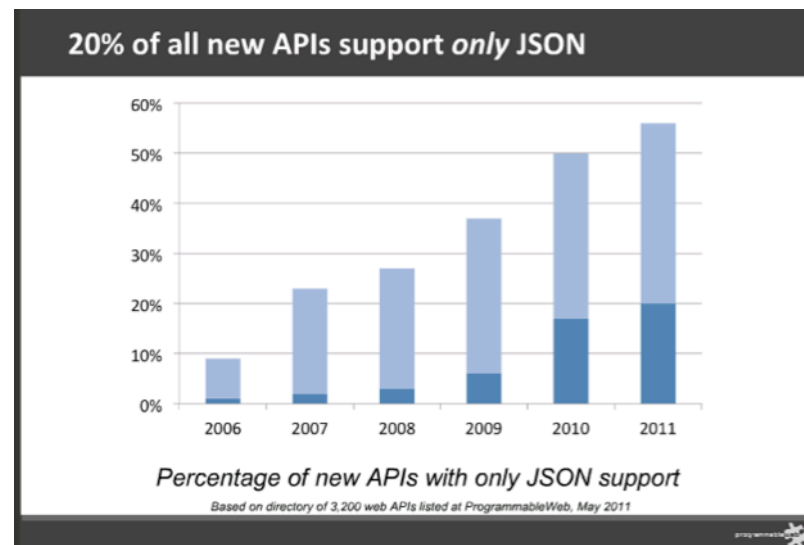
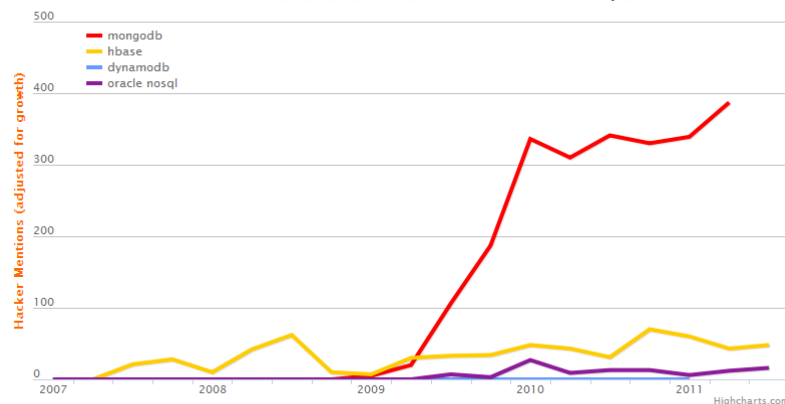
Open APIs State of the Market

- **JSON is the new cool**
 - XML declining: 5 years ago hardly any JSON
- **Why? JSON is**
 - Less verbose and smaller docs size
 - `<Mytag>vlaue</Mytag>` vs. `Mytag:value`
 - Tightly integrated with JavaScript which has a lot of focus
 - Most new development tools support JSON and not XML



HN Trends

See what hackers have been talking about.

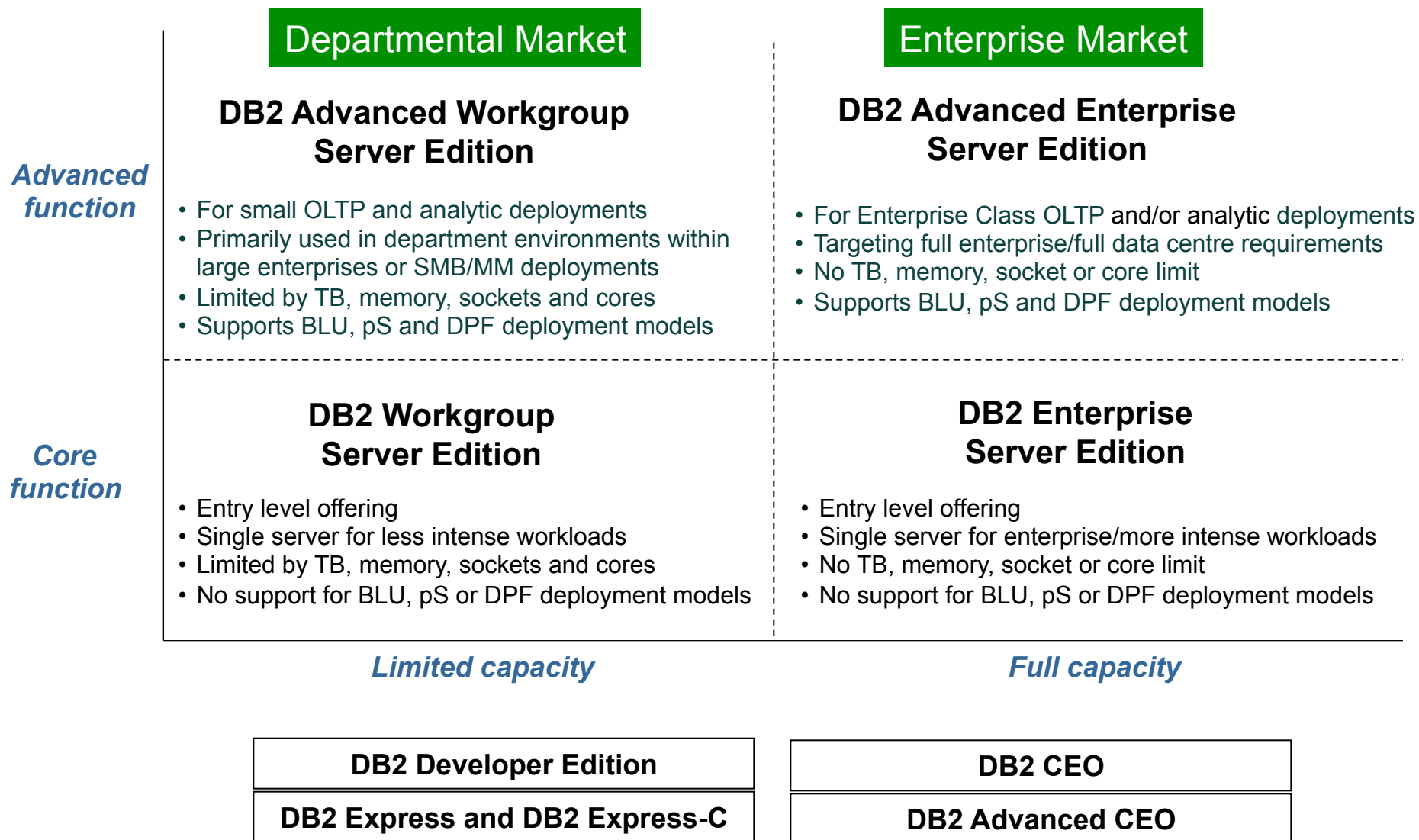


DB2 10.5 Packaging Simplification



DB2 10.5 Simplifies Product Packaging

One Set of Editions for Both Transactional and Warehouse Workloads



DB2® 10.5

with BLU Acceleration

Multi-workload database software for the era of big data

- **Always Available Transactions**

Disaster recovery of pureScale clusters over distances of 1000s km¹; means minimal downtime

- **Faster Analytics**

In-memory hybrid technology yields performance improvements ranging from 8-25x performance improvements², without costs or limits of in-memory only

- **Unprecedented Compatibility**

An average of 98% Oracle database application compatibility³

- **Future-Proofed Infrastructure**

NoSQL and Mobile database allows clients to expand and modernize their apps



“Before we made a final decision we benchmarked some of the key database management systems. That includes Oracle, SQL Server and DB2. We ended up choosing DB2 for several reasons. One was reliability, second was performance and perhaps the most important factor was ease of use”

– Bashir Khan, Director of Data Management and Business Intelligence

Thank You!



Disclaimer

© Copyright IBM Corporation 2013. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

IBM, the IBM logo, ibm.com, Information Management, DB2, DB2 Connect, DB2 OLAP Server, pureScale, System Z, Cognos, solidDB, Informix, Optim, InfoSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.