DB2® 10.5
with BLU Acceleration

Kevin Street
street@ca.ibm.com
**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
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

<table>
<thead>
<tr>
<th>Customer</th>
<th>Speedup over DB2 10.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large Financial Services Company</td>
<td>46.8x</td>
</tr>
<tr>
<td>Global ISV Mart Workload</td>
<td>37.4x</td>
</tr>
<tr>
<td>Analytics Reporting Vendor</td>
<td>13.0x</td>
</tr>
<tr>
<td>Global Retailer</td>
<td>6.1x</td>
</tr>
<tr>
<td>Large European Bank</td>
<td>5.6x</td>
</tr>
</tbody>
</table>

"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
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
BLU Acceleration Use Cases
Analytics Data Mart

*From Transactional Database*

- ERP or other transactional system
- Easily create and load a BLU Acceleration in-memory mart
- Transactional Database
- Line of Business Analytics Data Mart
- Multi-platform software flexibility
  - Instant performance boost
  - Handles terabytes of data
  - No indexes/aggregates to create and tune
  - Create tables, **Load and Go!**

Analytic Data Mart (BLU Tables)
Use Case – Enterprise Data Warehouse Offload

Data Mart Acceleration

EDW Application

Poor performing Oracle or Teradata warehouse

Cognos BI

with BLU Acceleration

OLAP Application

Easily create and load a BLU Acceleration in-memory mart

Create tables, Load and Go!
- Instant performance boost
- Handles terabytes of data
- No indexes/aggregates to create and tune
- Multi-platform software flexibility

Multi-platform software

© 2013 IBM Corporation
DB2 with BLU Acceleration Technology Internals
BLU Acceleration is Deeply Integrated With the DB2 Kernel

- **Client/Server**
  - BLU Acceleration uses DB2 client server infrastructure. Complete transparency to the application

- **Compiler**
  - BLU Acceleration uses the DB2 compiler to accept SQL, parse, perform semantic checking, and package creation

- **Process model – BLU Acceleration uses**
  - DB2 subagents
  - Prefetchers
  - TCB and Packed Descriptor for metadata

- **Memory**
  - BLU Acceleration uses DB2 bufferpool for storage allocation and caching
  - BLU Acceleration uses DB2 sort heap and package cache
  - OSS memory allocation for private work areas

- **Storage**
  - BLU Acceleration uses normal DB2 table spaces for storage allocations
  - Page sizes: 4K-32K

- **Utilities**
  - LOAD, BACKUP, RESTORE, EXPORT, SNAPSHOT, db2top, db2pd, etc.
The Seven Big Ideas of DB2 with BLU Acceleration

- Extreme Performance
  - Column Store
  - Data Skipping

- Lower Operating Cost
  - Simple to Implement and Use
  - Extreme Compression

- Hardware Optimized
  - Optimal Memory Caching
  - Core-Friendly Parallelism
  - Deep HW Instruction Exploitation (SIMD)

Data Mart Analytics Super Fast Super Easy
7 Big Ideas: 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

<table>
<thead>
<tr>
<th>LAST_NAME</th>
<th>Encoding</th>
<th>Packed into register length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Johnson</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gilligan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampson</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7 Big Ideas: 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'
```

<table>
<thead>
<tr>
<th>LAST_NAME</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Johnson</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
<tr>
<td>Gilligan</td>
<td>□</td>
</tr>
<tr>
<td>Sampson</td>
<td>□</td>
</tr>
<tr>
<td>Smith</td>
<td>□</td>
</tr>
</tbody>
</table>
7 Big Ideas: 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: **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: 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
## Optimize the Entire Hardware Stack

### In-Memory Optimized
- **Memory latency optimized for**
  - Scans
  - Joins
  - Aggregation
- **More useful data in memory**
  - Data stays compressed
  - Scan friendly caching
- **Less to put in memory**
  - Columnar access
  - Late materialization
  - Data skipping

### CPU Optimized
- **CPU acceleration**
  - SIMD processing for
    - Scans
    - Joins
    - Grouping
    - Arithmetic
- **Keeping the CPUs busy**
  - Core friendly parallelism
- **Less CPU processing**
  - Operate on compressed data
  - Late materialization
  - Data skipping

### I/O Optimized
- **Less to read**
  - Columnar I/O
  - Data skipping
  - Late materialization
- **Read less often**
  - Scan friendly caching
- **Efficient I/O**
  - Specialized columnar prefetching algorithm
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

*Diagram:*

- 10TB data
- 1TB after storage savings
- 10GB column access
- 1GB after data skipping
- 32MB linear scan on each core
- Scans as fast as 8MB encoded and SIMD
- Sub second 10TB query
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
Informational Uniqueness

- **DB2 10.5 introduces informational uniqueness constraints**
  - Enforced uniqueness remains the default
  - Informational (i.e., NOT ENFORCED) constraints do not enforce uniqueness
  - Valuable when data is coming from a trusted source

- **Benefits**
  - Less storage required! No index is created to enforce the constraint.
  - No runtime overhead to maintain unique indexes during LOAD, INSERT, UPDATE, or DELETE
  - The uniqueness definition informs the query compiler of unique data, enabling opportunities for superior query execution plans

```
CREATE TABLE t1 (c1 INTEGER NOT NULL, c2 INTEGER,
    PRIMARY KEY (c1) NOT ENFORCED);

ALTER TABLE t1 ADD CONSTRAINT unique1 UNIQUE (c2) NOT ENFORCED;
```
Mixing Row and Columnar Tables

- DB2 10.5 supports mixing row and columnar tables seamlessly
  - In the same tablespace and bufferpools
  - In the same query

- Best query performance for analytic queries usually occurs with all tables columnar

- Mixing row and columnar can be necessary
  - Point queries (highly selective access) favor row-organized tables with index access
  - Small, frequent, write operations favor row-organized tables
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

```
DELETE * FROM MyTable
WHERE Year = 2012

These extents hold only deleted data
```
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 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
  - Online table reorganization
HADR in DB2 pureScale

- **Integrated disaster recovery solution**
  - Simple to setup, configure, and manage

- **Support includes**
  - `ASYNC` and `SUPERASYNC` modes
    - `SYNC/NEARASYNC` under development
  - Time delayed apply
  - Log spooling
  - Both non-forced (role switch) and forced (failover) takeovers
HADR in DB2 pureScale: Example

Primary site

Transactions

Member

CF

Logs 1
Logs 2
Logs 3

TCP/IP

Failed member's logs

Standby site

Replay member

Member

CF
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

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.

1. `db2stop member1` 
2. `db2iupdt -online_update member1 inst1`
3. `db2start member1`
4. `db2stop member2` 
5. `db2iupdt -online_update member2 inst1`
6. `db2start member2`
7. `db2iupdt -commit_new_level inst1`
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: Explicit Hierarchical Locking (EHL)

- Designed to remove data sharing costs for tables/partitions that are only accessed by a single member
  - Avoids CF communication if object sharing not occurring

- Target scenarios
  - Workload affinitization
  - Workload consolidation and application affinitization

- Enabled via new `OPT_DIRECT_WRKLD` database configuration parameter
  - Detection of data access patterns happens automatically and EHL will kick in when data is not being shared after configuration parameter set
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
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.
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
    ```sql
    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

- DB2 10.1
  - Max 32K

- DB2 10.5
  - Max 1M
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';`

- **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
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](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
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.
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 its 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**

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

### Limited capacity

- DB2 Developer Edition
- DB2 Express and DB2 Express-C

### Full capacity

- DB2 CEO
- DB2 Advanced CEO
# DB2 10.5 Edition Details

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Limit</td>
<td>2 cores</td>
<td>8 cores</td>
<td>16 cores</td>
<td>16 cores</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>N/A</td>
</tr>
<tr>
<td>Socket Limit</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>4 Sockets</td>
<td>Unlimited</td>
<td>4 Sockets</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>N/A</td>
</tr>
<tr>
<td>DB2 Instance Memory Limitation</td>
<td>16 GB</td>
<td>64 GB</td>
<td>128 GB</td>
<td>128 GB</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>N/A</td>
</tr>
<tr>
<td>TB Limit</td>
<td>Unlimited</td>
<td>15 TB</td>
<td>15 TB</td>
<td>15 TB</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>N/A</td>
</tr>
<tr>
<td>Other Limits</td>
<td>AUSI - 5 user min</td>
<td>AUSI - 5 user min; Socket limit for Socket pricing</td>
<td>25 user per 100 VU</td>
<td>Must use DPF or BLU</td>
<td>25 user per 100 VU</td>
<td>25 user per 100 VU</td>
<td>Must use DPF or BLU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ready to go VMWARE Installer</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
<td>Linux</td>
</tr>
</tbody>
</table>
# DB2 10.5 Edition Details – Deployment and Options

|---------|-----|-------------------|-----------------|--------------------------|-----------------------------------|---------------------------|-----------------------------------|-------------------|

## Pricing Metric
- **free**
- Auth User Single Install, Limited Use Virtual Server, PVU, LUVS FTL
- Auth User Single Install, Socket, PVU
- Auth User Single Install, PVU
- TB
- Auth User Single Install, PVU

## Deployment Models Available
- BLU
- pureScale
- Data Partitioning Feature

## Tools, DB2 Feature Options (Opt)
- Advanced Recovery Feature
- Optim Performance Manager Extended Edition
- Optim Query Workload Tuner
- Optim Configuration Manager
- InfoSphere Optim pureQuery Runtime for LUW
- Warehouse Model Packs
- WHS Design Studio
- WHS SQW
- WHS Mining & text Analytics
- Cognos 10.2.5 Users
- WHS Cubing Services
- Data Studio
- InfoSphere Data Architect Capabilities

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Recovery Feature</td>
<td>✗</td>
<td>OPT</td>
<td>OPT</td>
<td>OPT</td>
<td>OPT</td>
<td>OPT</td>
<td>OPT</td>
</tr>
<tr>
<td>Optim Performance Manager</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Optim Query Workload Tuner</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Optim Configuration Manager</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>InfoSphere Optim pureQuery Runtime for LUW</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Warehouse Model Packs</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>WHS Design Studio</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>WHS SQW</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>WHS Mining &amp; text Analytics</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Cognos 10.2.5 Users</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>WHS Cubing Services</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Data Studio</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>InfoSphere Data Architect Capabilities</td>
<td>✗</td>
<td>✗</td>
<td>✓ (3)</td>
<td>✓ (3)</td>
<td>✗</td>
<td>✓ (3)</td>
<td>✓ (3)</td>
</tr>
</tbody>
</table>
# DB2 10.5 Edition Details – Core Function

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2 Capabilities and Engine Functionality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>solidDB &amp; solidDB UC</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>IBM Mobile Database Sync</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>IBM Mobile Database Support</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>pureXML Storage</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Time Travel Query</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Compression</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Compression Backup</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Row &amp; Column Access Control</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Label Based Access Control</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>SQL Replication</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Websphere MQ</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>High Availability Disaster Recovery</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
</tr>
<tr>
<td>Tivoli System Automation</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Online Reorg</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
</tr>
<tr>
<td>Advanced Copy Services</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Multi-temperature Storage</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Materialized Query Tables</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Multi-Dimensional Clustering</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Query Parallelism</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Connection Concentrator</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Table Partitioning</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Plan Lockdown</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Scan Sharing</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>DB2 Workload Management (WLM)</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>(2) (8)</td>
<td>√</td>
<td>(2) (8)</td>
<td>x</td>
</tr>
<tr>
<td>MQ Replication / CDC</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>(2) (8)</td>
<td>√</td>
<td>(2) (8)</td>
<td>x</td>
</tr>
<tr>
<td>Continuous Data Ingest</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>DB2 Connect</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
<td>(12)</td>
<td>√</td>
<td>(12)</td>
<td>x</td>
</tr>
</tbody>
</table>

© 2013 IBM Corporation
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
Tools Support for DB2 10.5
Optim Tools- June 2013

- **Day 1 Support** – easy to adopt, monitor, manage and maintain new capabilities within DB2
  - BLU made easier
    - BLU Workload Advisor in Query Workload Tuner
    - Helps identify which tables to convert to column organization to improve performance
    - How to convert a table from a row to column organization, BLU configuration, etc
    - BLU metrics support in Optim Performance Manager
  - Enhanced pureScale support
    - Centrally view pureScale cluster and instance configuration, track configuration changes
    - Dynamically isolate application transactions to run on a member subset without requiring app changes or outages
    - `Db2cluster` command incorporated into DataStudio
  - Ability to identify enterprise-wide "hot spots" by reviewing access patterns for indexes, tables and data partitions, tablespaces and database partitions in Optim Configuration Manager
  - Support for HADR multi-stand by in DataStudio
  - Completion of control center gap around Federation support
    - Support for federation wizard and sources
  - Support for Explicit Hierarchical Locking, New health alerts and enhanced notification capability for user defined alerts in Optim Performance Manager

- **Workload-optimized Expert Integrated Systems with Expertly Integrated Optim Tools** – continuous enhancements for PureSystem family
Optim Tools- June 2013

- Query Capture Replay - multiple new features/enhancements

- Enabling New Class of Applications
  - NoSQL – JSON support (Tech preview)
    - Flexible schemas allow rapid delivery of applications
    - Combine with enterprise strengths of DB2
  - DB2 RDF (Delivered first with DB2 10.1)
    - Further enhancements to the Triple store
    - Customers: Ripples Homeland Security (ISV), Japan Govt. Smarter Fish Project, Bank of America Merrill Lynch, Netage.nl (BP), IBM Smarter Cities Dublin

- Seamless shell sharing capability with latest Rational product stack

- Packaging simplification
  - Down from 13 to 7 Editions
  - Down from 12 to 1 Add-on Feature
    - Advanced Recovery Feature – Recovery Expert, Merge Backup and HPU
BLU Acceleration Configuration

- DB2 10.5 with BLU Acceleration introduces automatic workload management when DB2_WORKLOAD=ANALYTICS is set
  - Configure BLU Acceleration menu
BLU Acceleration Configuration

- Leverages the Database configuration assistant
  - DFT_TABLE_ORG will configure the default type for tables

DFT_TABLE_ORG will configure the default type for tables.
BLU Acceleration Display

➢ Administration Explorer
  - Organization icon
  - Organization column in the Object List Editor
**BLU Acceleration Display**

- **Property View**
  - Column/Row organization custom display behavior
  - “Default for Database” aligned with DFT_TABLE_ORG value
BLU Acceleration supported statements and related user interfaces

- **CREATE**
  - ORGANIZE BY ROW/COLUMN
BLU Acceleration supported statements and related user interfaces

➤ New option added on LOAD for column-organized table
Convert to BLU Acceleration

- db2convert vs. ADMIN_MOVE_TABLE?
  - System command vs. Stored Procedure – similar functionality
    - Both can be executed from Data Studio
  - Data Studio generates code for ADMIN_MOVE_TABLE
    - JDBC execution
    - Recommended flow:
      - Init, Copy, Replay
      - **Backup**
      - Swap
Thank You!