DB2® 10.5
with BLU Acceleration

George Baklarz, PhD
Program Director, DB2 Technical Sales
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.
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
“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 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
HADR in DB2 pureScale: Example

Primary site

Transactions

Member

Member

Member

Logs 1
Logs 2
Logs 3

TCP/IP

Failed member's logs

Standby site

Replay member

Member

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 online update, do a pre-commit check, and to subsequently commit 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.
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

```bash
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 SQL Enhancements
Tables with Extended Row Size

- Choosing a particular page size (4, 8, 16, or 32K) limits the maximum amount of data that a row can hold

<table>
<thead>
<tr>
<th>Page Size</th>
<th>Maximum Row Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>4K</td>
<td>4005</td>
</tr>
<tr>
<td>8K</td>
<td>8101</td>
</tr>
<tr>
<td>16K</td>
<td>16293</td>
</tr>
<tr>
<td>32K</td>
<td>32677</td>
</tr>
</tbody>
</table>

- Any row that exceeds the maximum size permitted cannot be inserted or updated on a page
- DB2 Kepler introduces the concept of a "Wide Row" that can extend beyond the size of a physical page (overflow)
Tables with Extended Row Size

- **Problem**
  - Biggest page size in DB2 is 32KB
    - Row has to fit into page
  - Tables may have many character based columns
    - VARCHAR(256) may take up to 1KB space – but rarely will!
  - Lazy database schema design
    - Max out Oracle (4000) or DB2 (32672) string columns
    - Actual content typically much smaller
  - Tables with many columns, but sparse content
    - Most columns are NULL, but each can be big when used
  - Sorting or grouping of wide result sets

- **Solution**
  - Spill large strings to LOB space if actual row exceeds page size limit
  - Optimized towards small percentage of rows spilling page size
Tables with extended row size

- Set database level permission to exceed page size for user tables
  - `UPDATE DB CFG USING EXTENDED_ROW_SZ ENABLE`
    - Default is `ENABLE` for new DB, `DISABLE` for existing DB
- Total defined row size must not exceed 1MB
  - `CREATE TABLE emp(name VARCHAR(4000),
                        address VARCHAR(4000),
                        cv VARCHAR(32000))`
- Variable length strings (`VARCHAR(n OCTETS), VARGRAPHIC(n/2 CODUNITS16)`) count `MIN(25,n)` towards page size limit.
- If no table space specified, pick smallest page size that fits defined row size if exists.
- Pick biggest available page size otherwise
- System temporary tables up to 1MB are allowed
  - `SELECT name, address, cv FROM emp ORDER BY name;`
    - 32KB system temp table space must exist
    - Prevents “silent” lobification of rows due to lack of appropriate system temp space.
Indexing Expressions

- **Problem**
  - Searching for computed values in table requires using Generated Columns
  - E.g. “Find employees without worrying about the case of their names”
  - Generated columns increase the row size
  - Makes table management more complex (e.g. LOAD)

- **Solution - Index the Expression directly**
  - `CREATE INDEX emp_name ON emp(UPPER(name));`
  - `SELECT salary FROM emp WHERE UPPER(name) = 'MCKNIGHT';`
  - Can use multi column index
  - Supports UNIQUE index and INCLUDE columns
  - No UDF or LOB column support at this point
  - Can collect statistics on index via associated statistical view

<table>
<thead>
<tr>
<th>Name</th>
<th>Salary</th>
<th>Manages</th>
</tr>
</thead>
<tbody>
<tr>
<td>McKnight</td>
<td>50000</td>
<td>Sales</td>
</tr>
<tr>
<td>Miller</td>
<td>25000</td>
<td>-</td>
</tr>
<tr>
<td>Van Gogh</td>
<td>45000</td>
<td>Finance</td>
</tr>
<tr>
<td>Chan</td>
<td>37000</td>
<td>-</td>
</tr>
</tbody>
</table>
Indexing Excluding NULL Keys

- **Problem**
  - Current solutions for uniqueness, require expression on primary key to be added to the key
  - Unique indexes cannot have more than one NULL value

- **Solution**
  - 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`
    - For multi column index all key parts must be NULL
Weakly typed distinct types

Problem
- DB2 introduced Weak Typing in V9.7
  - Comparison and arithmetic operators of unlike types was permitted as long as a valid conversion was available
    - "123" = 123
  - Strong typing of existing distinct types was not permitted

Solution
- Provide a weak form for distinct types without type safety
  - Use a unique alias for the data type of like variables, parameters and columns to provide a single point of control over these objects' type.
  - Derive data types with more restrictive domains than the base types
    - E.g. MONEY, NAME, GUID
- Provide check-constraints for distinct types
- Also available in DB2 10.1 starting with FP2
Weakly Typed Distinct Types in PL/SQL

- Declare SUBTYPE in PL/SQL blocks
  - Supported in
    - PL/SQL Package
    - Procedures
    - Triggers
  - Optional RANGE constraint for numeric types
  - Optional NOT NULL constraint
  - Usable by variables and parameters

```
DECLARE
  SUBTYPE tinyint IS INTEGER RANGE -256..255 NOT NULL;
  val tinyint := 255;
BEGIN
  val := val + 1;
END;
/
SQL20552N The cast or assignment failed because the value does not conform to the data type constraint of the user-defined type. User-defined type: "TINYINT". Value: "256".
```
Weakly typed distinct types in SQL

- **Create distinct type** **WITH WEAK TYPE RULES**
  - No to/from cast functions generated
  - Function resolutions treats type as base type
  - Optional check constraint
  - Optional NOT NULL constraint
  - Constraints enforced on assignment

- **Supported in**
  - Global types
  - Procedures
  - Triggers
  - Modules
  - Functions
  - Anonymous blocks

- **Used by variable, parameters, sequences, columns**
  - No constraints allowed for columns and sequences
    - `CREATE TYPE color AS CHAR(1) WITH WEAK TYPE RULES
    NOT NULL CHECK (VALUE IN ('R','G','B'));
    CREATE VARIABLE my_color color DEFAULT 'R';
    SET my_color = 'W';
    SQL20552N The cast or assignment failed because the value does not conform to the data type constraint of the user-defined type. User-defined type: "COLOR". Value: "W".`
Pipelined table functions

- **Problem**
  - Java, C and .NET table functions provide support for Pipelined functions, but limited development skills to build these routines
  - Inline SQL table functions must compose result set in single RETURN statement

- **Solution**
  - 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.
  - Also available in DB2 10.1 starting with FP2
Pipelined table function PL/SQL

- Create a type to hold the result set
  - CREATE TYPE datearray AS TABLE OF DATE;
- Create the function:
  - CREATE FUNCTION dates(dt IN DATE) RETURN datearray PIPELINED AS
    BEGIN
      LOOP
        PIPE ROW(dt);
        dt := dt + 1;
      END LOOP;
      RETURN;
    END;
  /
- Use the function in a FROM clause.
  - SELECT * FROM TABLE(dates('2013-01-01')) WHERE ROWNUM <= 2;

<table>
<thead>
<tr>
<th>COLUMN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-01-01 00:00:00</td>
</tr>
<tr>
<td>2013-01-02 00:00:00</td>
</tr>
</tbody>
</table>
Pipelined table function SQL PL

- Create the function
  - CREATE FUNCTION dates(IN dt DATE) RETURNS TABLE(dt DATE)
    BEGIN
      LOOP
        PIPE (dt);
        SET dt = dt + 1 DAY;
      END LOOP;
    RETURN;
  END;
/

- Use the function in a FROM clause
  - SELECT * FROM TABLE(dates('2013-01-01'))
    FETCH FIRST 2 ROWS ONLY;

    DT
    2013-01-01
    2013-01-02

- In SQL PL no need to create a type for the result set.
Ad-hoc federated table access

- **Problem**
  - Traditionally DB2 requires creation of NICKNAME to map to a remote table
    - Need to know accessed tables in advance
  - Syntax incompatible with Oracle

- **Solution**
  - Support ad-hoc reference to remote table using server in the identifier
    - Reach out to a table in a remote database
  - Support for static (SQL PL and PL/SQL) new in DB2 10.5
    - DB2 syntax
      ```sql
      SELECT * FROM server.schema.table;
      ```
    - Oracle syntax
      ```sql
      SELECT * FROM schema.table@server;
      ```
Function library extensions

- **RTRIM(<string> [, <character>]) and LTRIM (<string> [, <character>])**
  - Optional second parameter to specify trim-character
  - Example
    ```
    LTRIM(RTRIM('Hello', 'o'), 'H')
    → 'ell'
    ```

- **MOD(<expr1>, <expr2>)**
  - Built-in with support for DOUBLE and DECFLOAT
  - Example
    ```
    MOD(3.2, 0.3)
    → 0.2
    ```

- **SUBSTR4(<string>, <start> [, <length>])**
  - Mostly equivalent to CODEUNIT32 based SUBSTR but:
    - <start> can be negative to start relative to the end of the string
    - <length> can be greater than actual length of the string
  - Example
    ```
    SUBSTR4('Hello', -2);
    → 'lo'
    SUBSTR4('Hello', 2, 10);
    → 'ello'
    ```

- **RTRIM, LTRIM, and MOD available in 10.1 with FP2**
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
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 (thankst) 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](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
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>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Core function</th>
</tr>
</thead>
<tbody>
<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**
Thank You!