DB2 for z/OS:

What will be important tomorrow

- and why

Peter Vanroose

abis TRAINING & CONSULTING

GSE NL Nat.Conf.

20 November 2014

Almere - Van Der Valk

“50 jaar en nu verder!”

DB2: Trends and Directions

Outline :

- **What important trends do we observe in DB2 over the past decade?**
- **What remarkable novelties appeared in recent DB2 versions?**
- **How does this relate to IT trends outside DB2?**
- **What are the trends and directions outlined by IBM?**

DB2 trends since version 8

DB2 v8 for z/OS -- released sept. 2004

Highlights from the IBM “What’s New” document:

- More *flexibility* with SQL
- Improved *security*
- Enhanced *compatibility* with the DB2 family
- *Scalability* and *performance*
- Reliability, *availability*, and serviceability

Versions 9, 10, and 11 have more or less the same highlights:

- v9: Availability, Performance, Security, *Information on demand*
- v10: Performance, Scalability, Availability, Security, *Integration*
- v11: Performance, Availability, *Ease-of-use*, Security

What do these terms mean? Do they indicate a trend?

DB2: Trends and Directions

Availability

1.1

“Data should be available at all times”. But ...

- **RDBMS => ACID: Atomic - Consistent - *Isolated* - Durable**

  => While Application A is writing data, Application B cannot write it

- **What about read + write ?**
  => e.g. let Application B read old “version” while A writes new one

1. Utilities:

- **SHRLEVEL NONE / REFERENCE / CHANGE**

  trend: ----------> (e.g. online reorg)

- **combined actions: e.g. REORG with COPY & RUNSTATS**
- **partition independency**

  no need to (b)lock more data than necessary (e.g. DPSI’s)

2. Applications:

- **WITH UR**
- **SKIP LOCKED DATA (readers & writers: DML & utilities)**
- **USE CURRENTLY COMMITTED (readers: BIND option)**

Availability (..)

3. DDL
   - “online schema changes” (v8 terminology):
     add column , add/drop PK, FK, index, ... (pre v8)
     change datatype (v8)
     add partition (v8)
     add index column (v8), include columns (v10)
     rename column (v9), rename table (v10), rename index (v9)
     drop column (v11)
     modify tablespace type (v10), page size (v10)
   - pending DDL (v10)
4. subsystem
   - optimistic locking (v9); lock avoidance
   - online change of zParms
   - system backup (flash copy)
   - online add an active log dataset (v10)

Scalability

1.2

Table size limits growing:

- row lengths
- number of rows

- 4096 partitions (v8) --> 128 TB per table(space)
- extended addressability for datasets: 4GB --> 256GB
- varchar(32767)
- CLOB(2G), BLOB(2G)
- inline LOBs!

Number of objects: 65535 for each of the 65535 databases

- catalog & directory evolution: Unicode / UTS / row-level locking

Data sharing

Performance

Optimizer intelligence in SQL query interpretation/translation:

- choosing the “most efficient” access path:
  - I/O avoidance !
  - early filtering
  - stage-1 predicates
- accurate cost estimation of an access path:
  - filter factor of a predicate
    problem: parameter markers; static SQL
  - availability of data statistics
- possibility for “hints” <==> autonomous (complex) optimizer

Storage & retrieval efficiency:

- space map info => segmented TS, universal TS
- in-memory
- predicting the next request: prefetch, look-aside, ...

Compatibility and integration 1.4

"Across DB2 family" compatibility

- similar new functionality at +/- same release

Oracle compatibility

- implicit casting (v10)
- versioning
- aliases (Oracle names) for several scalar functions:
  TO_CHAR; TO_DATE; TO_NUMBER; NVL; ...

Windows integration

- Data Studio
- support for JDBC & ODBC

Unix integration

- command-line interface (v11)
- zLinux
- XML functionality (e.g. XSLT) through libraries under USS

Security & auditing

1.5

- **Audit tracing: increasing set of possibilities**
- **SSL (secure socket layer protocol) on TCP/IP for z/OS**
  - **encryption**

Access control additions since v8 (on top of GRANT/REVOKE):

- **Multi-level security (v8) => a dead end?**
- ***Roles* and *trusted contexts* (v9)**
  - **abandon the tight coupling of “login ID” and authorizations**
  - **make authorizations dependent on connection profile**
- ***Row permissions* & *Column masks* (v10)**
- ***Separate security* (v10):**
  - **SECADM: can (only) grant & revoke**
  - **SYSADM: can (only) access data & metadata**
  - **this “separation of duty” is optional**

Flexibility, ease-of-use

1.6

- **SQL syntax flexibility:**
  - **more SQL building block recombinations:**
    - common table expressions (CTEs, “WITH” clause)
    - SELECT FROM UPDATE/INSERT/DELETE
    - ORDER BY, FETCH FIRST in subqueries
  - **triggers on views (“instead of” triggers)**
  - **new statements: MERGE; TRUNCATE; XQUERY**
  - **datatypes: BIGINT; VARBINARY; TIMESTAMP WITH TIMEZONE ... DECFLOAT! (v9)**
  - **implicit casting (v10)**
- **utilities: automatically RESTART(current) after failure**
- **not enforced FK (v8)**
- **LOAD/UNLOAD to/from CSV (“delimited”)**
- **point-in-time recovery with BACKOUT (v10)**
- ***automatic* space allocation: SECQTY -1; utility aux datasets**

DB2: Trends and Directions

Application integration

1.7

native SQL ("SQL PL")

- stored procedures (v9)
- user-defined scalar functions (v10)
- user-defined table functions (v11)

"ARRAY" datatype in SQL PL (v11)

global variables (user-defined registers) (v11)

autonomous stored procedures (v11)

Use of DataStudio (GUI; on Unix, Linux or Windows)

- is Eclipse-based
- freely available
- includes Visual Explain

Gradually better support for ODBC / JDBC

DRDA; 64-bit; Unicode; integrated type-4 JDBC driver

Enterprise information on demand

1.8

Expensive term for "there is more than just relational data"...

- **XML (v9) ==> *very important long-term commitment!***
- **data analytics**
  - **OLAP: online analytic processing**
  - **data warehouses, dimensional modeling, star schema's**
- **"Big Data" and "NoSQL":**
  - **sometimes, ACID is too strong**

    ==> BASE: Basically Available, Soft-state, Eventually consistent
  - **data replication, "sharding"**
  - **columnar store (instead of row-based)**
  - **key-value store**
  - **write-efficient store: only inserts (with timestamp), "versioned"**

==> currently, only very limited support for all this in DB2 !

(e.g.: APPEND YES; optimistic locking; index on expression;
not enforced FKs; not-logged tablespaces; ...)

Important novelties in recent DB2 releases

Each DB2 version came with 2 or 3 “important novelties”:

- **v8: Unicode (=> varchar)**
  64-bit virtual storage
  Better partitioning support
- **v9: XML (native)**
  UTS => more autonomy
  RRF => varchar behaviour
- **v10: “versioning”: table history; last committed; logs; backout**
  separate security
  pending DDL
- **v11: Autonomy & automation (utilities & optimizer)**
  NoSQL & “Big Data” analytics: OLAP, JSON, IDAA

All versions: evolution of optimizer, utilities, memory mgmt, locking, ...

Unicode

Universal representation for all text characters of *any* language!

See www.unicode.org

One, unique “codepoint” (integer) per character

Several byte representations:

- **UTF-8 (used by DB2)**
  - **1 byte for first 128 entries: (“ASCII”)**
    non-accented letters, numbers, some punctuation
  - **2 bytes for ~ 20000 entries: most European languages**
    accented letters, Greek, Cyrillic, IPA, Hebrew, Arabic
  - **3 bytes for ~ 200 000 entries:**
    special symbols (e.g. currency, arrows); Thai, Ethiopic, Japanese,...
  - **4 bytes for Chinese, hieroglyphs, Minoan (linear B), emoticons, ...**
- **UTF-16 (used by Microsoft), UCS-2 => big-endian & little-endian**
- **UTF-32**

Trading I/O for CPU

***VARCHAR*(n) instead of CHAR(n) ; UTF-8 instead of EBCDIC**

- **almost identical performance compared to CHAR(n)**
- **slight CPU overhead for (possibly high) I/O & storage gain**

NOT PADDED indexes

RRF: *reordered row format* (v9)

- **fixed-width row data always accessible at fixed record offset**

compression **=> savings typically up to factor 5 à 10**

- **hardware acceleration**
- **tables: row-based; indexes: page-based (& fixed conversion)**

in-memory work tables

larger bufferpools, RID pools, sort pools

- **require increasingly more real memory ...**

prefetch, pre-sort

Partition independence

2.3

Partition = *physical* unit (data set), not a logical unit (table)

Horizontal subdivision of tables

- **based on logical subdivision: PBR - partition by range**
- **based on size constraints: PBG - partition by growth**

Cf. vertical subdivision of tables (into multiple tablespaces):

- **normalisation ==> really independent tables**
- **LOB, XML ==> auxiliary tables**

Partitions gradually becoming more loosely coupled

Utilities: reorg; backup; recover. Applications: locking; pruning

potential problem: non-partitioned indexes (NPIs)

- **de-coupled partitioning and clustering (v8)**
- **DPSIs: data-partitioned secondary indexes (v8)**
- **cloning (v9) & versioning (v10): yet another flavour of partitioning?**

Universal tablespaces

2.4

“The way to go” (since v9):

- **segmented internal structure (esp. space map pages)**
- **but also (potentially) partitioned**
- **automatic partitioning if necessary (PBG)**
- **will eventually (v12?) allow multiple tables**

==> to replace all other tablespace types (simple/segmented/partition)

-> XML tablespace IS UTS

-> most new features (since v9) only available for UTS

==> unification: reducing dataset infrastructure complexity:

- **UTS**
- **Index spaces**
- **LOB tablespaces**
- **(backup datasets)**

XML

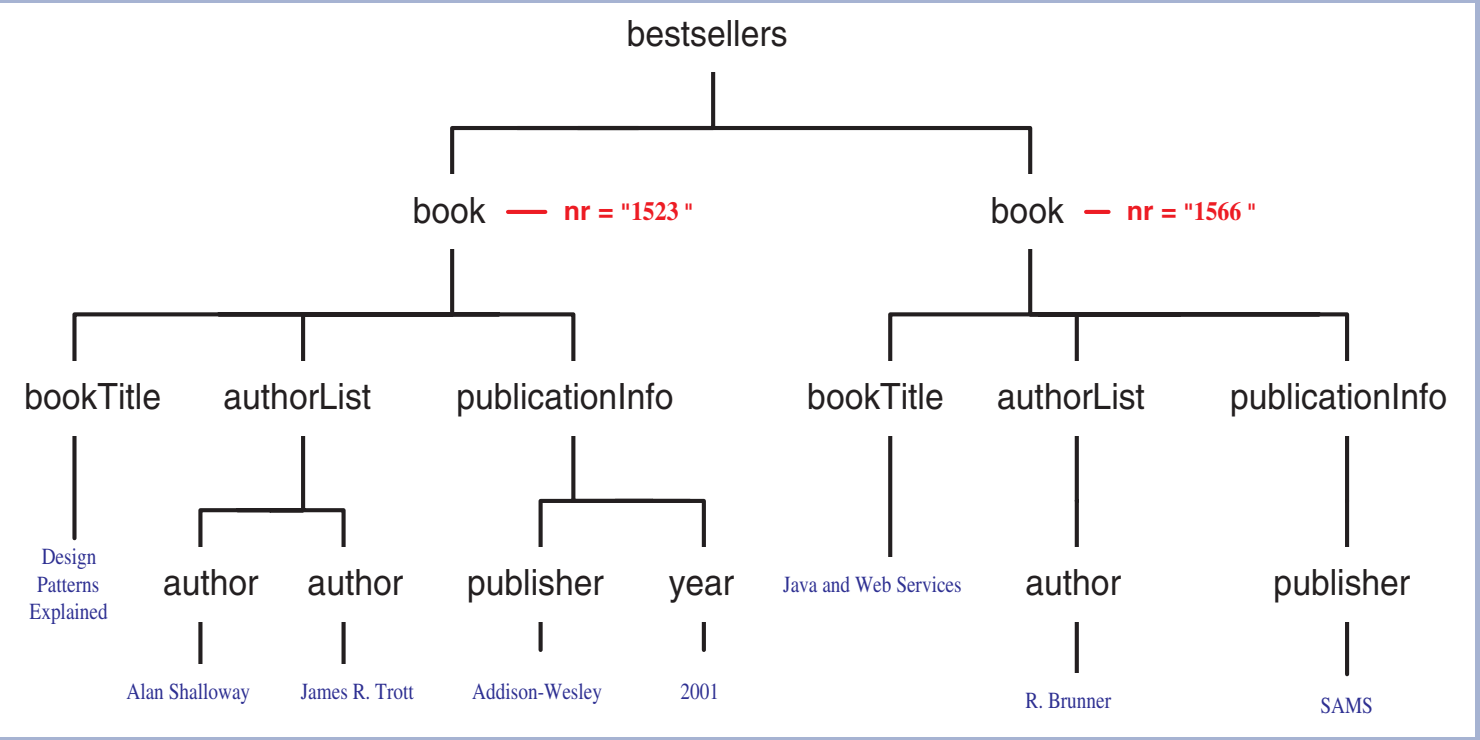
A strategic, log-term choice!

SQL (relational) and XML (hierarchic) data in a *hybrid database*

==> integrated access to both types of data

- **datatype conversion between XML & SQL:**
  - **text: VARCHAR => no problem**
  - **numeric: DECFLOAT**
  - **date/time (since v10) => incl. timezone support**
  - **SQL indexes on XML “xpath” expressions**
- **binary XML (v10)**
  - **is essentially the internally stored format**
  - **in communication with applications & utilities**
  - **2-step metadata: list of nodenames => only need integer IDs**
  - **data: length-prefixed varchars**
- **“FLWOR” expressions (XQuery) inside XML functions in SQL (v10)**

XML is ...



and not so much

<bestsellers><book nr="1523"><bookTitle>Design Patterns Explained</bookTitle><authorList><author>Alan Shalloway</author><author>James R. Trott</author></authorList><publicationInfo><publisher>Addison-Wesley</publisher><year>2001</year></publicationInfo></book><book nr="1566"><bookTitle>Java and Web Services</bookTitle><authorList><author>R. Brunner</author></authorList><publicationInfo><publisher>SAMS</publisher></publicationInfo></book></bestsellers>

DB2: Trends and Directions

1. DB2 trends since version 8
   1. Availability
   2. Scalability
   3. Performance
   4. Compatibility
   5. Security
   6. Flexibility
   7. Application integration
   8. “Information on demand”
2. Important novelties in recent DB2 releases
   1. Unicode
   2. Trading I/O for CPU
   3. Partition independence
   4. Universal tablespaces
   5. XML
   6. “Time travel”
   7. Pending DDL
   8. Autonomy
3. Optimizer evolution
4. More trends & directions (?)
   Analytics
   Mobile
   Cloud
   Social
   Hybrid

"Time travel"

"Temporal" tables: (v10)

- **system-period *versioned* data**
  - **automatically maintained on INSERT / UPDATE / DELETE**
    - old "version" of the data kept, but moved out of base table
    - each row is labeled with "valid from...to" time range (2 timestamps)
  - **new SELECT syntax:**

    FROM table_name AS OF timestamp
  - **implemented through coupled "history table"**
    - in different tablespace, but logically coupled (table DDL)
- **business-period versioning**
  - **to be manually maintained**
    - each row is labeled with "valid from...to" time range (2 timestamps)
    - old and new "versions" of same PK are kept in same table
    - DB2 only *guarantees non-overlapping validity time ranges*

==> auditing? versioning? insert-only evolution?

Pending DDL

Since v10, “some” (new) DDL ALTERs are *not immediate:*

- **ALTER TABLESPACE:**
  [change pctfree, freepage, priqty, secqty: already pre-v8!]
  change type (viz. into UTS), page size, dataset size, ...
- **ALTER TABLE:**
  drop column (v11); change partition limit key (v11); (but not: hash)
- **ALTER INDEX:**
  page size (but not: add column, or padded)

DDL request is registered in catalog (SYSIBM.SYSPENDINGDDL)

Executed on next REORG

- **but meanwhile non-blocking**
- **not visible in the object meta-data (catalog)**
- **can even be un-done (before reorg) without side effects**

Autonomy

Since v9, but increasingly in v11:

DBA-style actions happen “automatically”, without explicit action

E.g.:

- **create PK index**
- **create XML aux. table and XML tablespace**
- **create required XML indexes**
- **clean up pseudo-empty index pages (without reorg) (v11)**
- **keep old access path on rebind (v11)**
- **...**

Is a longer-term, strategic choice! (To be continued...)

Optimizer evolution

The optimizer is of course steadily improving

Remarkable tendencies:

- **from *static* SQL only to more & better support for *dynamic* SQL**
  - **dynamic statement cache (DSC); e.g. literals replacement**
  - **REOPT(once), REOPT(always) , REOPT(auto)**
- **new optimizer techniques & query rewrites:**
  - **non-matching datatypes (v8 ++)**
  - **transitive closure**
  - **partition pruning**
  - **across query block optimization (v9)**
  - **externalisation: more explain tables**
- **more detailed *statistics***
  - **column groups (correlation); distribution stats; histograms**
  - **optimizer feedback about missing statistics (v11)**

Optimizer evolution (..)

- **In-memory auxiliary structures**
  (thanks to 64-bit addressability)
  - **sparse indexes (pre-v8) now sometimes in memory (v9)**
  - **work files now sometimes in memory (v9)**
  - **new auxiliary indexing structures (v10, v11)**
- **"Access path stability":**
  - ***hinting* mechanism evolution**
  - **introduction of an "*uncertainty measure*" for a filter factor**
    => optimizer will prefer access path with more stable cost estimate
  - **PLANMGMT(BASIC/EXTENDED/OFF) & SWITCH bind options**
  - **APREUSE & APCOMPARE bind options**
- **Improved implementation for**
  - **OR, IN (v9 ++)**
  - **stage-2 predicates (e.g.: index on expression; earlier eval; ...)**
  - **parallel updates of indexes (v10 ++)**

More trends & directions (?)

Analytics coming to the data

Mobile

Cloud

Social

Hybrid

ftp://ftp.software.ibm.com/software/os/systemz/briefing0605/Mike_Perera_-_DB2_Trends_and_Directions.pdf

ftp://ftp.software.ibm.com/software/os/systemz/summit/handouts/04-Jeff_Josten_-IBM_DB2_for_zOS_Trends_and_Directions.pdf

http://pages.ca.inter.net/~ccdb2/Downloads/JohnCampbell-DB2zOS-TD.pdf

http://www.slideshare.net/Surekha06/idaa-trends-directions

https://share.confex.com/share/122/webprogram/Handout/Session15144/SHARE%202014%20Aneheim%20-%20System%20z%20Platform%20Strategy.pdf

http://www.gsebelux.com/system/files/Trends%20in%20DBMS%20-%20DB2%20Perspective.pdf

Questions, remarks, feedback, ... ?

Thank you!

Peter Vanroose

ABIS Training & Consulting

pvanroose@abis.be

DB2: Trends and Directions

1. DB2 trends since version 8
   1. Availability
   2. Scalability
   3. Performance
   4. Compatibility
   5. Security
   6. Flexibility
   7. Application integration
   8. “Information on demand”
2. Important novelties in recent DB2 releases
   1. Unicode
   2. Trading I/O for CPU
   3. Partition independence
   4. Universal tablespaces
   5. XML
   6. “Time travel”
   7. Pending DDL
   8. Autonomy
3. Optimizer evolution
4. More trends & directions (?)
   Analytics
   Mobile
   Cloud
   Social
   Hybrid