# DB2 for z/OS update seminar

## Early experiences from using DB2 10

Savings ... right out of the box
IBM DB2 10 for z/OS delivers faster queries and reduced cost with optimized technology
Proven
Secure
Simple
Cuts costs
Innovative

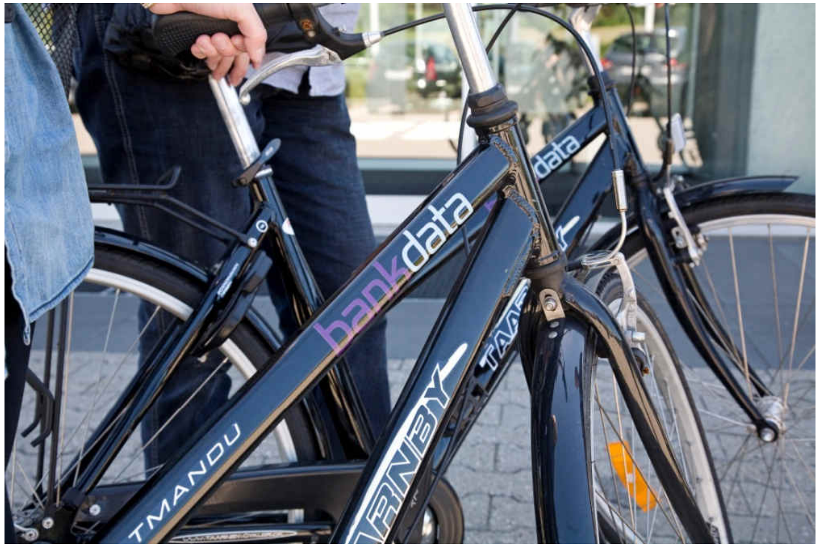
Oslo 16/3

Frank Petersen - fap@jndata.dk

Bankdata/JN Data,Denmark

Silkeborg 14/3

Stockholm 17/3



Copenhagen 15/3

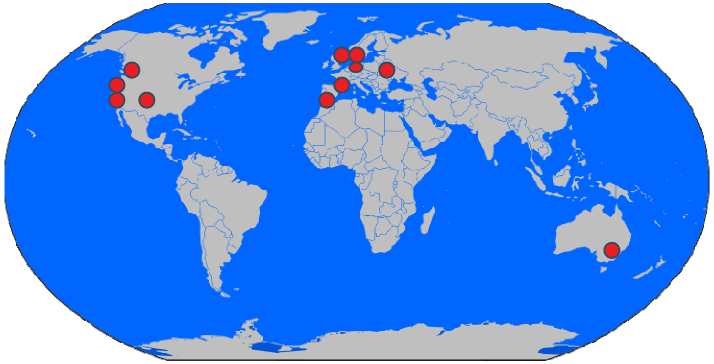
Helsinki 18/3

# DB2 for z/OS update seminar

IOD

GSE

IDUG



RUG

Regional User Group

DB2 has a very active user community
- DO volunteer to speak at the conferences
- Or help organizing/moderating
- Do not underestimate your skills or experiences
- Unlock the world – you have the key !!!

Who am I ?
- Working with DB2 since version 1.1
- Regular speaker since 2001
- Member of IDUG Speakers Hall of Fame since 2008

bankdata

# Un- Usual disclaimer ?

- ALL slides in this presentation can be trusted 100% !!
- Except this one !!!!<br>☺

bankdata

# Agenda :

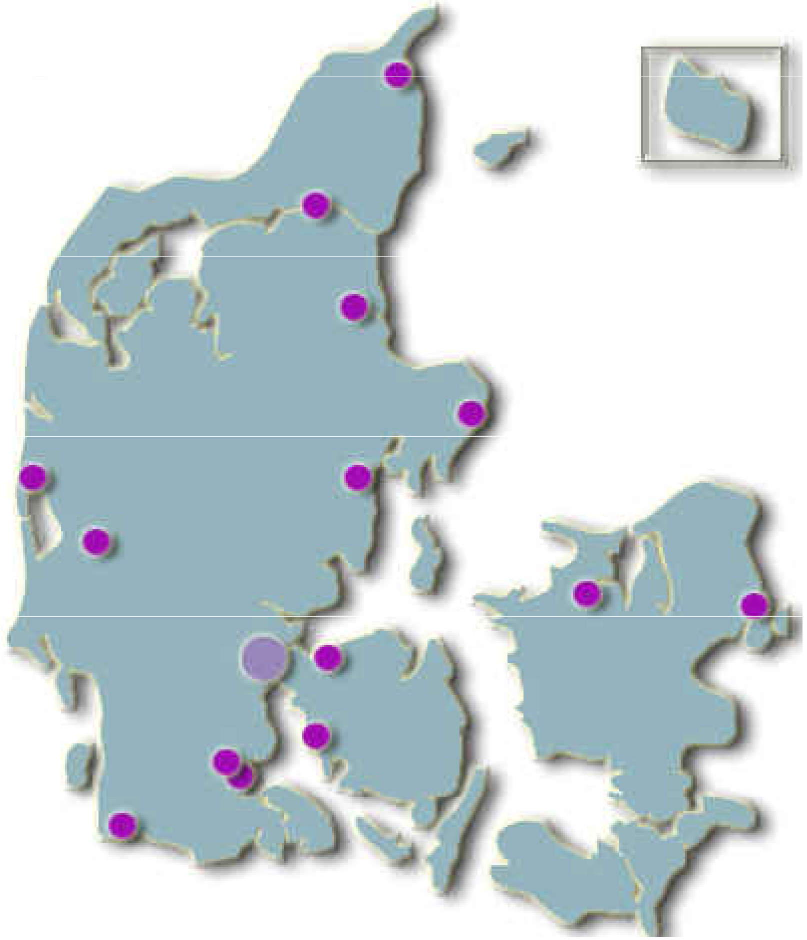
- Short info about Bankdata
- The Beta (QPP) process ?
  - "Why do the QPP-customers never find my problems ?"
- What did Bankdata test ?
- A few selected items in detail.
- As much of the rest as time permits
- Conclusions

# WHO ? - Location and history

- Bankdata is located in the city of Fredericia, Denmark
- Bankdata was founded on June 16th, 1966
- The first data processing action was made on October 18th, 1967
- Currently 500 employees
- Non-profit centre !!

bankdata

# WHO ? - Customers/members: 14 banks



And number 15 is on it’s way !!

●

bankdata

# The Beta process

- Driven by a special group at IBM
  - IBM Beta Works
  - Drives beta's for many products
  - Huge expertise in administration, controlling and documentation
  - Clever people – have technical knowledge indeed !!
- Who is selected/qualifying ?
  - Beats me !!
  - Installations with interesting product mix ?
  - Installations willing to invest time ?
  - Perhaps the "usual suspects" ?
- Timing
  - Candidates chosen autumn 2009
  - Startup meeting winter 2009/2010
  - First code delivery march 2010
  - Ended after GA

# Participating in a Beta process?

- Skills needed to volunteer to almost 1 year of work ?
  - Ignorance ?
  - Stupidity ?
  - Ambitions !
  - Will to contribute to making DB2 10 better than the best, faster than the fastest and with reduced TCO.
  - Persistency
  - Commitment by management
  - Ability to set a team
    - Technical skills
    - Administrative skills
  - Pay back time ?

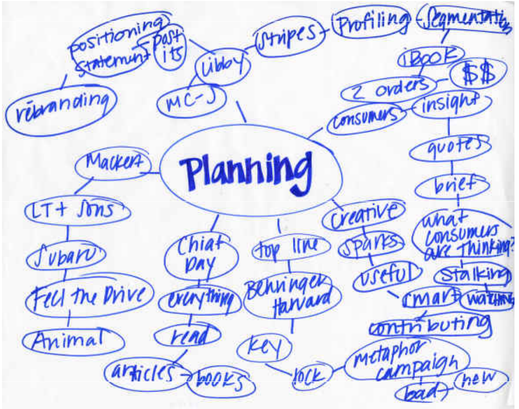
bankdata

# The beta - Preliminary Work

- Staff are a scarce resource
  - 2 Systems Programmers & 1 DBA part time assigned
  - Allowed to involve others on specific issues.
- No idea of the needed administrative tasks involved
  - Bi-weekly phone conferences
  - Status reports every week
  - Error reporting process
- Test plan was produced :
  - Candidates were things we could use or felt looked tempting
- Informed family that we would be MORE impossible to live with
  - Luckily they decided to stay anyway (good pensions ! ☺)

bankdata

# Test plans – theory and practice



- Considerations for building a test plan :
  - Only man power considerations
  - Code problems not considered
    - ("WHAT –are there bugs in the code ?")
  - Problems getting hardware resources not considered
    - CPU, Disk space
  - Existing monitors and tooling are not supporting DB2 10
  - Planned to upgrade 7 real DB2 subsystems
    - Later reduced to 1
  - The revised test plan was performed on :
    - 2 cloned systems
    - 1 purpose defined new system
    - 1 upgraded system

# Test plans – theory and practice

- Surprises :
  - The amount of time needed for administration and reporting
  - When test items came to a halt because of problems
    - We simply started on the next one
    - When one person start on test item >4 at the same time
      - Too many balls to juggle
      - Restarting halted ones for collecting more doc etc is expensive
      - One simply forgets things and become inefficient
      - We are only men ☺
  - The "Performance test" was extremely expensive
    - Finding scenarios difficult
    - Saving material for comparisons time consuming (job logs etc…)
    - Needed to upgrade real systems to even start (re-considered later)

# Items Tested

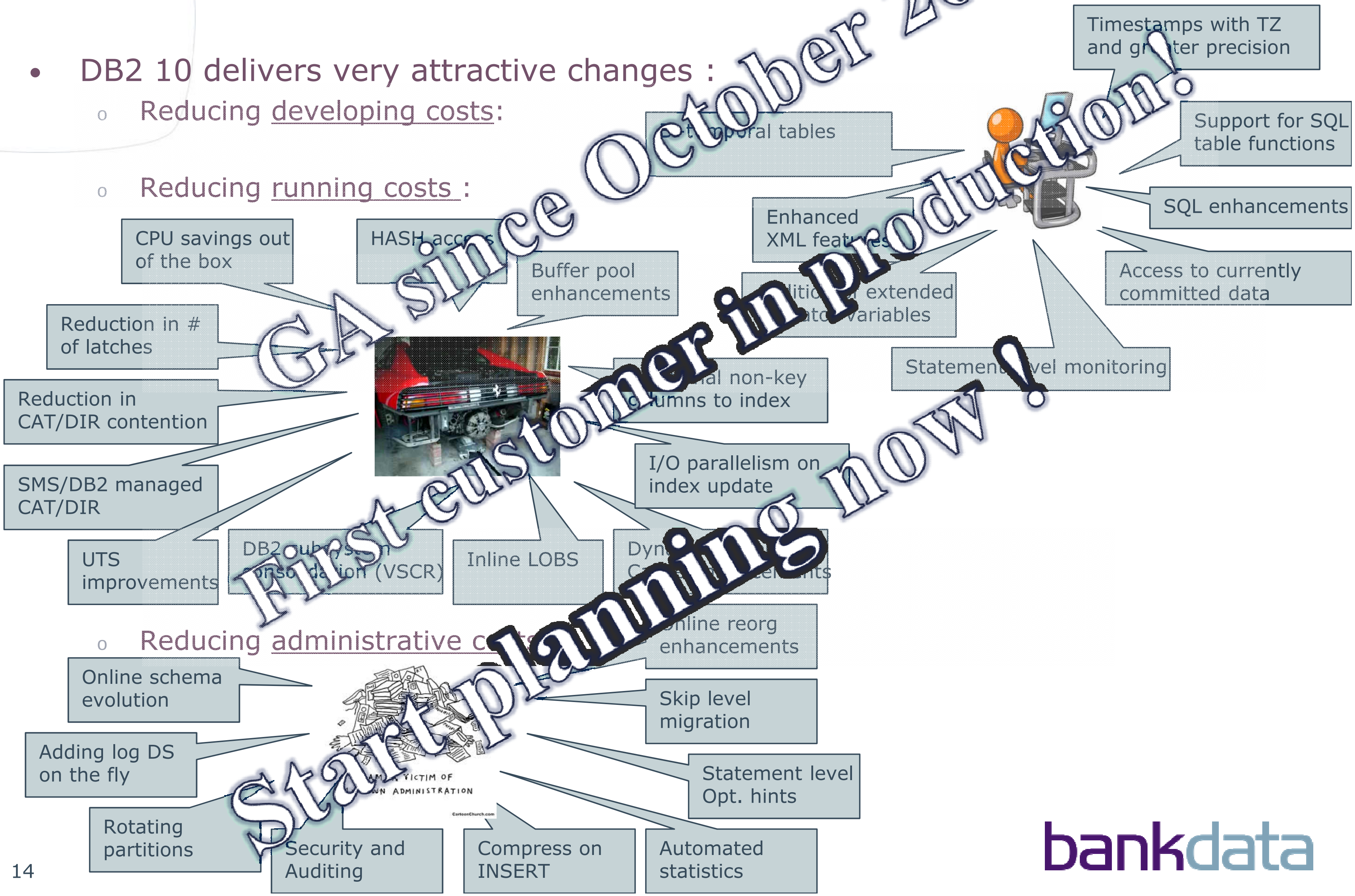
| | |
|---|---|
| Online Reorg Enhancements | Availability |
| REORG SHRLEVEL CHANGE for LOBs | Availability |
| Add Active Log | Availability |
| Auto-Compress on Insert | Productivity |
| Auto Stats | Productivity |
| V10 Installation and Samples | Productivity |
| Online Schema Evolution | Productivity |
| Automated DB2 Supplied SP Install | Productivity |
| Catalog Restructure | Scalability |
| Eliminate LOB Materialization | Scalability |
| UTSERIAL Elimination | Scalability |
| Eliminate DDF Private Protocol | SOA |
| New DBA Privileges | Security |

# Items Tested

| | |
|---|---|
| • Buffer Pool Enhancements | Performance |
| • Inline LOBS | Performance |
| • Hash Access To Data | Performance |
| • Statement level Opt. hints | Performance |
| • Dynamic Statement Cache Enhancements | Performance |
| • General performance comparison V9->V10 | Performance |
| • Support For DataStudio Monitoring | Performance |
| • Reduce Log-Latch Contention | Performance |
| • I/O Parallelism for Index Updates | Performance |
| • Additional non-key Columns in Index | Performance |
| • SQL Pagination | Performance |
| • High Performance DBAT | Performance |
| • Greater Timestamp Precision | SQL |
| • Moving Sum and Average | SQL |
| • Timestamp With Timezone | SQL |
| • XML Engine Schema Validation | Application |
| • XML Multi Versioning | Application |
| • XML UPDATE | Application |
| • Binary XML Support | Application |
| • XML Date and Time Support | Application |
| • Check XML | Application |
| • Bi-Temporal Tables | Application |

# Going to DB2 10 ? When and why ?

- DB2 10 delivers very attractive changes :
  - Reducing developing costs:
  - Reducing running costs :
  - Reducing administrative costs



Timestamps with TZ and greater precision
Support for SQL table functions
SQL enhancements
Access to currently committed data
Statement level monitoring
Bi-temporal tables
Enhanced XML features
Additional extended indicator variables
CPU savings out of the box
HASH access
Buffer pool enhancements
Reduction in # of latches
Reduction in CAT/DIR contention
SMS/DB2 managed CAT/DIR
Additional non-key columns to index
I/O parallelism on index update
UTS improvements
DB2 subsystem consolidation (VSCR)
Inline LOBS
Dyn[illegible] C[illegible]cements
Online reorg enhancements
Online schema evolution
Adding log DS on the fly
Rotating partitions
Security and Auditing
Compress on INSERT
Automated statistics
Skip level migration
Statement level Opt. hints
AM A VICTIM OF
UN ADMINISTRATION
CartoonChurch.com

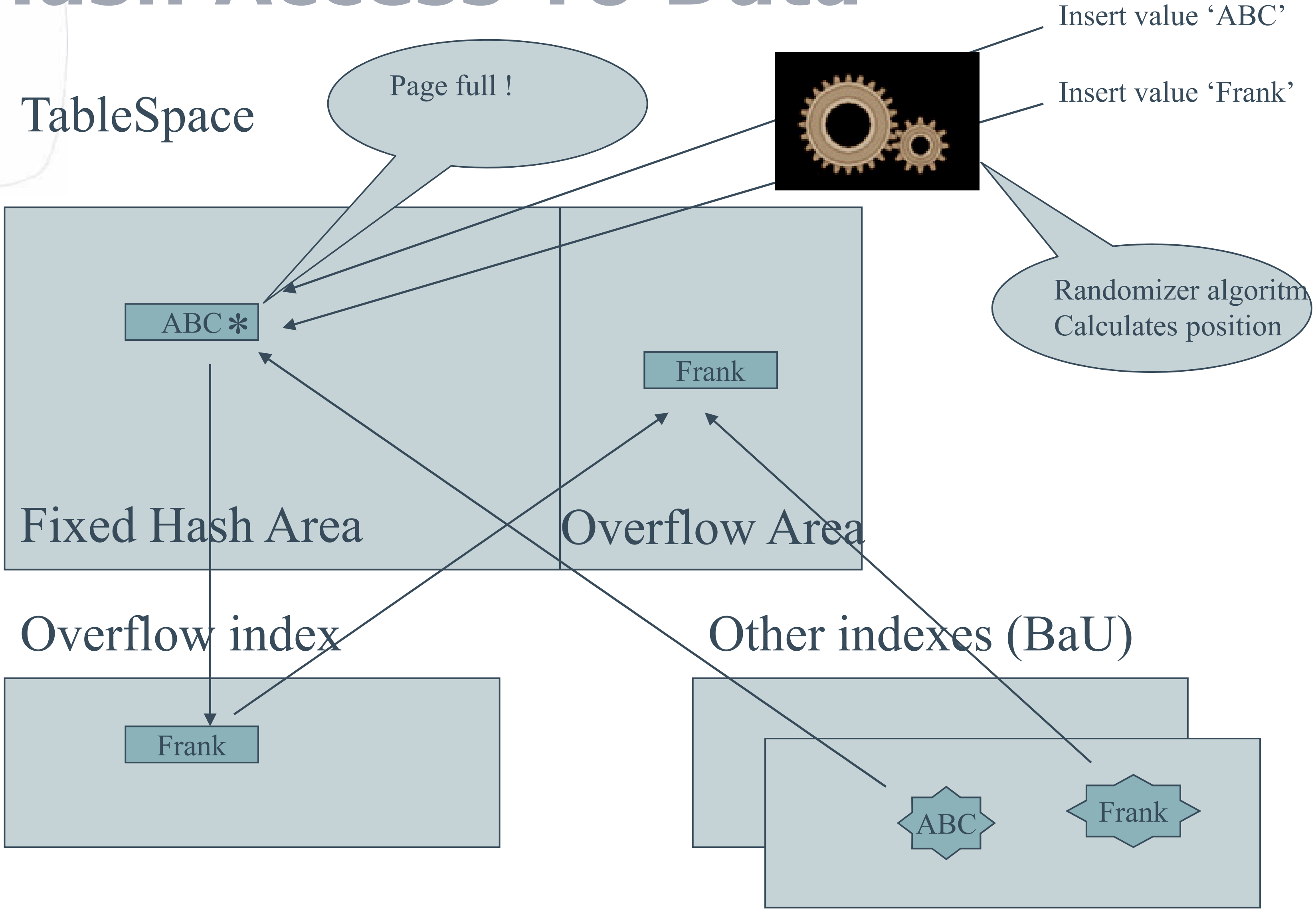
GA since October 2010

First customer in production!

Start planning now !

bankdata

# Hash Access To Data



# Hash Access To Data

- Facts
  - New syntax for the Create/Alter TABLE
    - ORGANIZE BY HASH UNIQUE (BANKNR. NGL_TYP) HASH SPACE 1 G
  - Max 64 columns, not NULLS and max 255 char for HASH key
  - Columns type for key-columns can not be changed
  - No clustering, No Append, No MEMBER CLUSTER
  - Define the HASHSPACE as x K/M/G
  - HASH can be turned on/off by ALTER
  - Altering to HASH enforces the UNIQUE immediately
    - The physical HASH'ing happens at next reorg. No inserts until then
    - DB2 will automatically add HASH overflow index (in Rebuild Pending)
  - The catalog will contain information when HASH was last used
    - SYSIBM.SYSTABLESPACESTATS-HASHLASTUSED

# Hash Access To Data

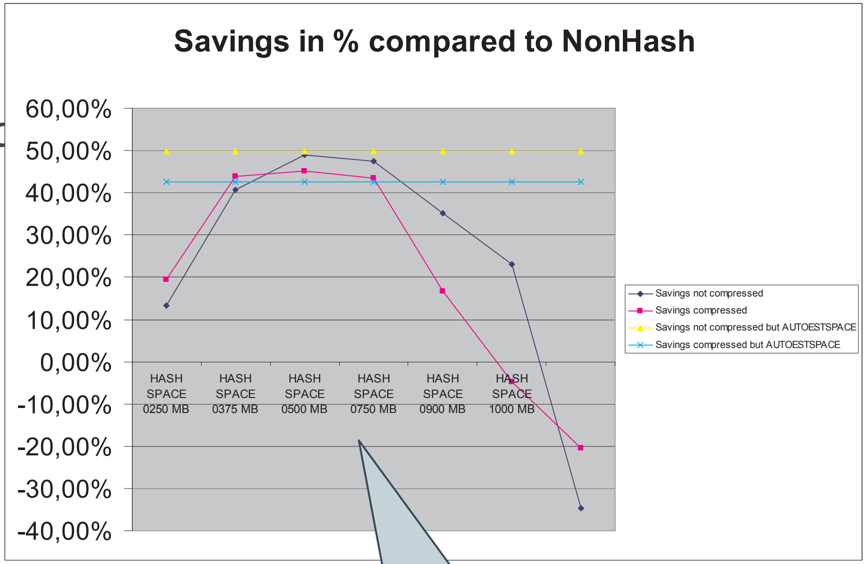
- Reorg
  - RTS will help determine when to reorganize :
    - SYSINDEXSPACESTATS.TOTALENTRIES not more than 10% of SYSTABLESPACESTATS.TOTALROWS.
  - Reorg produces information about the size of the HASH SPACE
    - DSNU2802I -DB2B 063 13:39:44.95 DSNUGHSH - THE HASH SPACE HAS BEEN PREALLOCATED AT 75967 FOR TABLESPACE
  - The Hash Space Size can be set before the reorg
  - Or you can use AUTOESTSPACE(YES)
  - We have seen AUTOESTSPACE is a good choice
    - Seem to do a better pick than you can
    - Use all the autonomics that you can ☺

# Hash Access To Data

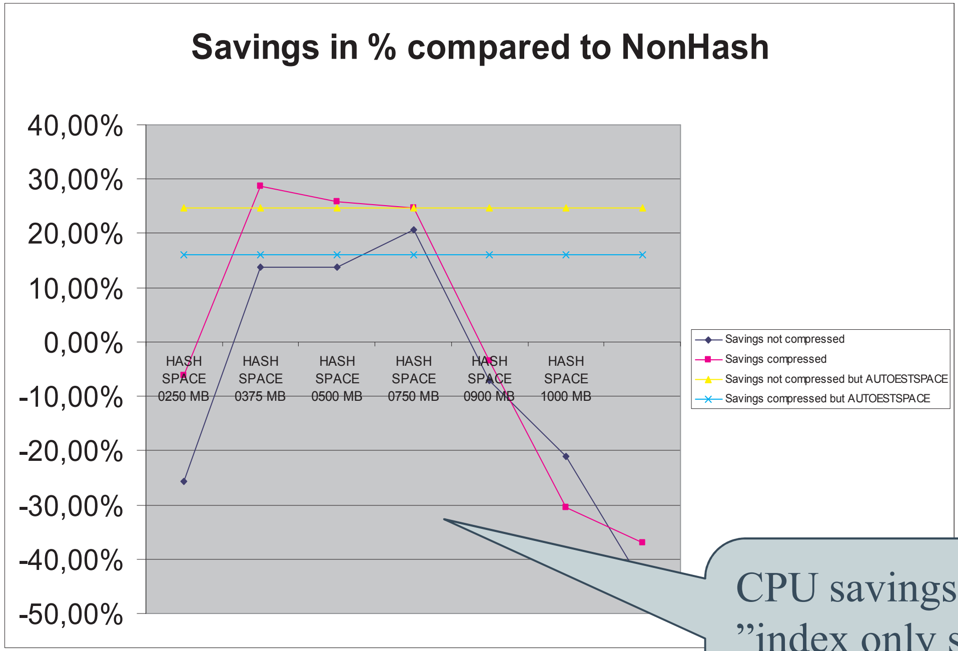
- Description and outcome
  - This is NOT IMS !!!
  - Don’t expect 0 rows in overflow after reorg. Performance will be best with a small amount of rows in overflow
  - Recoded business case prototype.
    - Picked business case according to specs
      - High NLEVELS in index
      - Purely Random access. Select with = predicate
      - SELECT access (1M, 370K re-read)
      - Same random keys, same random sequence in all runs.
      - Tried with variations of PCTFREE and COMPRESS
  - There were late improvements in the feature
    - The measuring was done before these changes

# Hash Access To Data

- Description and outcome (cont'ed)
  - Hash has a competitor in DB2 10 :
    - Non key columns could enable inc



CPU savings compared to ”index and data scenario”
Flat lines are ”autoestspace”
The rest is varying HashArea



CPU savings compared to ”index only scenario”
Flat lines are ”autoestspace”
The rest is varying HashArea

# Hash Access To Data

- Attention :
  - The results suggest that the data should be relatively static in nature
  - Use AUTOESTSPACE unless the table has not reached its normal state
  - The column HASHLASTUSED indicates when HASH was last used.
    - Track users not using hash access and understand why.
  - In the beta longer elapsed time was seen in LOAD and REORG. Work was in progress in this area, but the outcome unknown at the time of writing.
  - PLAN_TABLE ACCESSTYPE will tell if Hash Access is being used :
    - H, MH or N
  - SYSTABLESPACESTATS REORGHASHACCESS tells how often Hash Access has been used
  - SYSINDEXSPACESTAT REORGINDEXACCESS tells how often the overflow index has been used
  - No UPDATE of Hash Key. DELETE and INSERT required !

# XML – The new things

- DB2 10 XML support contains various new  ‘goodies’ :
  - Engine Schema Validation
  - Multi Versioning
  - Update of parts of the document
  - Binary support
  - Date and Time support
  - Check XML

- Delivery that makes the DB2 for z/OS XML support more complete

# XML Engine Schema Validation

- Description and outcome
  - In V9 validation was done from an UDF
    - Expensive in CPU and storage
    - Setup of WLM etc
    - No zIIP usage
    - Had to be done in XMLPARSE
  - ***Now uses engine via scalar function DSN_XMLVALIDATE***
  - ***Set up the XML schema repository*** environment, USS,JAVA etc..
  - 3 new stored procedures to register the schemas :
    - XSR_REGISTER – names the schema and the location
    - XSR_ADDSCHEMADOC – To be called once per schema document
    - XSR_COMPLETE – Closing of the registration process
  - Validation happen automatically at INSERT, UPDATE, LOAD
  - CHECK PENDING can happen if Schema is changed

# XML Engine Schema Validation

- Example :

Let’s create a table :

```
CREATE TABLE MotorCycles ( MotorCycleNumber BIGINT NOT NULL,
STATUS VARCHAR(10) NOT NULL WITH DEFAULT 'UnRestored',
MotorCycleData XML (XMLSCHEMA ID SYSXSR.MOTORCYCLES), …
```

Let’s behold a schema :

```
<?xml version=”1.0”?>
<xs:schema targetNamespace=”http://sample.com/motorcycles” xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns=http://www.ibm.com/db2/java/xml/ns elementFormDefault=”qualified”>
<xs:element name=”motorcyclenumber">
<xs:complexType><xs:sequence>
<xs:element name=”Mark" minOccurs="1" maxOccurs="1” type=”xs:string”/>
<xs:element name=”line" type="xs:string" minOccurs="1"/>
</xs:sequence></xs:complexType></element></schema>”
```

Let’s register this by using a Java program :

```
DB2Connection ds = (DB2Connection) con;
// Invoke registerDB2XmlSchema
ds.registerDB2XmlSchema( xmlSchemaNameQualifiers, xmlSchemaNames, xmlSchemaLocations,
xmlSchemaDocuments, xmlSchemaDocumentsLengths, xmlSchemaDocumentsProperties,
xmlSchemaDocumentsPropertiesLengths, xmlSchemaProperties, xmlSchemaPropertiesLength, false);
```

# XML Engine Schema Validation

- Example :

```
INSERT INTO MotorCycles (MotorCycleNumber, Status, MotorCycleData)
values(18,’Running’,
XMLPARSE(DOCUMENT SYSFUN.DSN_XMLVALIDATE(’
<Root><MotorCycleNumber  ID=18> <FirstReg Date=1966-06-01/>
<IDData>

<Location>
<line>Coventry</line>
<line>UK</line>
</Location></IDData>
<repairs>
<repair>
<partnumber>T14821</partnumber>
<name>Piston</name>
<quantity>2</quantity>
<price>112.25</price>
</repair>
</repairs>
</MotorCycleNumber></Root>','SYSXSR.MOTORCYCLES')));
```

```
DSNT408I SQLCODE = -20399, ERROR:  ERROR ENCOUNTERED DURING XML PARSING
       VALIDATION AT LOCATION 364 Unknown reason RC=0018,RSN=8604,  OR XML
       PARSING OR VALIDATION AGAINST XML SCHEMA XSRID 26.
```

# XML Engine Schema Validation

- Description and outcome XML_CHECK
  - Now we alter the schema
    - Change the physical schema and do a new registration
    - Point the table to another schema (or add a schema)
  - Existing XML are now doubtful..
  - TableSpace goes into CHECK PENDING
  - Do a CHECK UTILITY :
    - SCOPE PENDING/REFONLY/AUXONLY
      - REFONLY check base table
      - AUXONLY checks the LOB/XML objects
    - XMLERROR REPORT/INVALIDATE
    - AUXERROR REPORT/INVALIDATE
  - If AUXERROR INVALIDATE and invalid XML doc found :
    - Document MOVED to dynamic shadow XML space
    - XML column in invalid base row marked invalid
    - Space goes info AUX WARNING-You have to clean up !

```
ALTER TABLE MOTORCYCLES
   ALTER INFO SET DATA TYPE
   XML(XMLSCHEMA ID SYSXSR.MOTORCYCLES1);

---------+---------+---------+---------+---------+
DSNT404I SQLCODE = 162, WARNING:  TABLE SPACE
JSKDBTST.XMYC0001 HAS BEEN  PLACED IN CHECK
PENDING
```

# XML Engine Schema Validation

- Description and outcome XML_CHECK
  - How to do a CHECK UTILITY :

```
//SYSIN    DD   *
CHECK DATA TABLESPACE JSKDBTST.JSKUTS
INCLUDE XML TABLESPACES ALL
  SCOPE PENDING SHRLEVEL REFERENCE
  AUXERROR INVALIDATE  XMLERROR INVALIDATE
```

```
…DSNU849I    208 14:20:52.34 DSNUKERK - XML DOCUMENT WITH DOCID X'008000000000000002' IS NOT VALID AGAINST
 THE XML TYPE MODIFIER OF XML COLUMN INFO FOR TABLE UBIJSK.MYCUSTOMER_UNIVERSAL
DSNU739I    208 14:20:52.35 DSNUKDAT - CHECK TABLE UBIJSK.MYCUSTOMER_UNIVERSAL COMPLETE, ELAPSED TIME=00:00:00
DSNU1180I   208 14:20:53.60 DSNUGSQL -  SQLCODE = 000,  SUCCESSFUL EXECUTION
DSNU848I  -DTST 208 14:20:53.63 DSNUKNFK - XML EXCEPTION TABLE UBIJSK.XET_MYCUSTOM_INFO_0000 CREATED FOR XML COLUMN
INFO  IN BASE TABLE UBIJSK.MYCUSTOMER_UNIVERSAL
DSNU826I  -DTST 208 14:20:53.93 DSNUKRDN - XML COLUMN INFO IN TABLE UBIJSK.MYCUSTOMER_UNIVERSAL HAS BEEN
  MARKED INVALID BY THE CHECK UTILITY.
                     DOCID X'008000000000000002'
DSNU846I  -DTST 208 14:20:53.94 DSNUKRDY - 1 XML DOCUMENTS DELETED FROM TABLE UBIJSK.MYCUSTOMER_UNIVERSAL  FOR XML
COLUMN INFO
DSNU816I  -DTST 208 14:20:53.96 DSNUGSRX - TABLESPACE JSKDBTST.JSKUTS IS IN AUX WARNING STATE
```

- Attention :
  - DB2 10 CHECK utility recap:
    - Can check Indexes on parts of XML tree
    - Check XML document against schema
    - Exception table will be implicitly created with invalid row.
    - Manually delete error XML row and insert it again well-formed.

# XML Multi Versioning

- <u>Description and outcome</u>
  - DB2 will automatically assign a version number for a XML document (UTS)
  - When a document is updated the version number is increased
  - Readers for a document will see the prior version if the updater has not committed the changes
  - When the new version of the document is committed AND all readers of the prior version has finished DB2 will automatically remove the now outdated version (Garbage Collection)
  - New versioning format is build when table is changed
  - In the new versioning format no locks are obtained by readers
  - Locking and concurrency is now very different between “normal” relational data and XML data inside relational data.

# XML UPDATE

- Description and outcome
  - Before DB2 10 you could not update part of a XML document.
  - Now, DB2 comes with real SQL update syntax for XML.
  - Delivers subset of XQUERY update facility
    - Insert
    - Delete
    - Replace
  - Implemented using the scalar function XMLMODIFY
  - Easy to implement in SQL syntax.
- Attention
  - ***IBM has build "challenges" into messages related to XML***

- ***Let's see how XML Update works:***

# XML UPDATE - Replace node

```
<MotorCycleNumber><id>18</id> <FirstRegDate=“1966-06-01“/>
<IDData>
<Mark>Triumph</Mark>
<Location>
<line>Coventry</line>
<line>UK</line>
</Location>
</IDData>
<repairs>
<repair>
<partnumber>T14821</partnumber>
<name>Piston</name>
<quantity>2</quantity>
<price>112.25</price>
</repair>
</repairs>
</MotorCycleNumber>
```

```
UPDATE MotorCycles SET MotorcycleData
= xmlmodify(
'replace value of node
/motorcyclenumber/iddata/location/line[2]
with “United Kingdom"')
WHERE MotorCycleNumber=18;
```

```
<MotorCycleNumber><id>18</id><FirstRegDate=“1966-06-01“/>
<IDData>
<Mark>Triumph</Mark>
<Location>
<line>Coventry</line>
<line>United Kingdom</line>
</Location>
</IDData>
<repairs>
<repair>
<partnumber>T14821</partnumber>
<name>Piston</name>
<quantity>2</quantity>
<price>112.25</price>
</repair>
</repairs>
</MotorCycleNumber>
```

# XML UPDATE - Insert node

```
<MotorCycleNumber><id>18</id><FirstReg Date=“1966-06-01“/>
<IDData>
<Mark>Triumph</Mark>
<Location>
<line>Coventry</line>
<line>United Kingdom</line>
</Location>
</IDData>
<repairs>
<repair>
<partnumber>T14821</partnumber>
<name>Piston</name>
<quantity>2</quantity>
<price>112.25</price>
</repair>
</repairs>
</MotorCycleNumber>
```

```
UPDATE MotorCycles SET MotorcycleData
= xmlmodify(
‘insert node $in as last into
/motorcyclenumber/location’,
xmlparse(document
’<region>Britain</region>’) as ”in”)
WHERE MotorCycleNumber=18;
```

- The insert can happen at :
  - ”before”
  - ”after”
  - ”as first into”
  - ”as last into”
  - ”into”

```
<MotorCycleNumber><id>18</id><FirstReg Date=“1966-06-01“/>
<IDData>
<Mark>Triumph</Mark>
<Location>
<line>Coventry</line>
<line>United Kingdom</line>
<Region>Britain</Region>
</Location>
</IDData>
<repairs>
<repair>
<partnumber>T14821</partnumber>
<name>Piston</name>
<quantity>2</quantity>
<price>112.25</price>
</repair>
</repairs>
</MotorCycleNumber>
```

# XML UPDATE - delete node

```
<MotorCycleNumber><id>18</id><FirstReg Date=“1966-06-01“/>
<IDData>
<Mark>Triumph</Mark>
<Location>
<line>Coventry</line>
<line>United Kingdom</line>
<Region>Britain</Region>
</Location>
</IDData>
<repairs>
<repair>
<partnumber>T14821</partnumber>
<name>Piston</name>
<quantity>2</quantity>
<price>112.25</price>
</repair>
</repairs>
</MotorCycleNumber>
```

```
UPDATE MotorCycles SET MotorcycleData =
xmlmodify(
‘delete node
/motorcyclenumber/location/region’) WHERE
MotorCycleNumber=18;
```

```
<MotorCycleNumber><id>18</id><FirstReg Date=“1966-06-01“/>
<IDData>
<Mark>Triumph</Mark>
<Location>
<line>Coventry</line>
<line>United Kingdom</line>
</Location>
</IDData>
<repairs>
<repair>
<partnumber>T14821</partnumber>
<name>Piston</name>
<quantity>2</quantity>
<price>112.25</price>
</repair>
</repairs>
</MotorCycleNumber>
```

# Binary XML Support

- Description and outcome
  - A standard designed to DRDA (read : ODBC, SQLJ and JDBC)
  - More efficient handling in the DB2 engine.
  - FASTER ! SMALLER !
  - Data saved in same format in DB2
  - Testing has been done with small java programs inserting a few Mb XML from file via streaming, and selecting the inserted row again.
  - JDBC 4.0 and SQLXML used for getting binary support for insert/select.
    - In ODBC binary support is honored by SQL_C_BINARYXML data type
  - Test programs compared to code without binary support.
  - Using binary XML naturally will save some resources on inserts.
  - Comparing accounting reports, shows that binary ***inserts really gives a saving in CPU time.***
  - ***Selects not that big a difference***.

# XML Date and Time Support

- Description and outcome
  - DB2 10 now supports :
    - Data types : dateTime, date, time, duration, yearMonthDuration,dayTimeDuration
    - Functions and operators : comparisons, arithmetic , Extractions
    - XML index on these types
  - dateTime : 2011-02-28T12:00:00.000000000000+02:00
  - date : 2011-02-28+02:00
  - Time : 12:00:00.000000000000+02:00
  - Duration : P52Y10M13DT5H30M   52 years, 10 months, 13 days, 5 hours, 30 minutes
  - yearMonthDuration : P52Y10M      52 years, 10 months
  - dayTimeDuration : P364TD10H30M  364 days, 10 hours, 30 minutes
- XMLQUERY :
  - XPATH can be used to boolean evaluation of “times” :
    - SELECT XMLQUERY(‘xs:duration(“PT72H”) = xs:duration(“P3D”)’)   → TRUE
  - To do your arithmetics :
    - SELECT XMLQUERY(’xs:yearMonthDuration(”P2Y11M”) + xs:yearMonthDration(”P0Y1M”)’)  -> P3Y  (3 years !)

# XML Date and Time Support

- Description and outcome (cont’d) :
  - fn:current-time() , current-dateTime, current-date
  - Extraction :
    - SELECT XMLQUERY(‘fn:month-from-dateTime(xs:current-dateTime())’) -> 3
  - Adjustment :
    - SELECT XMLQUERY(‘fn:adjust-datetime-to-timezone(xs:datetime(“2011-03-14T12:00:00+02:00”),xs:dayTimeDuration(“PT3H”))’)

      -> returns 2011-03-14T13:00:00+01:00
  - ***Created a table with xml data containing Date and Time data: <shipDate>2010-05-31</shipDate><shipTime>14:30:00</shipTime>***
  - Tested SQL with various example, ex:
    - Select …. where XMLEXISTS(
      'declare default element namespace "http://www.sample.com/customer";
      $I/customer*[shipDate=$date]*'
      PASSING INFO AS "I**", CURRENT DATE as "date**");
    - Select …. where XMLEXISTS(
      'declare default element namespace "http://www.sample.com/customer";
      $I/customer[**shipDate>xs:date**("2010-05-31")]');

# Bi Temporal Tables

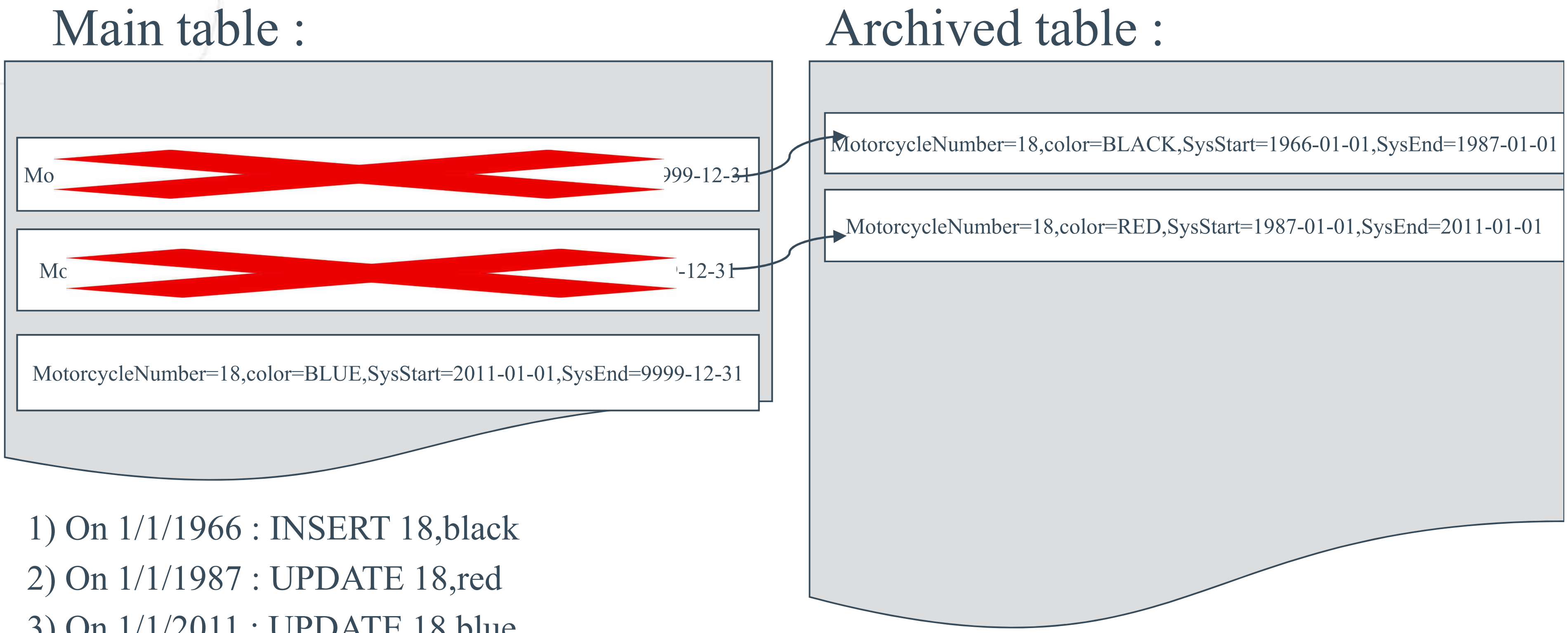
- Description System Time
  - Timing is everything !
  - DB2 can now automatically archive a row when data is changed

```
CREATE TABLE MotorCycles (
MotorCycleNumber INTEGER,Color CHAR(10),
SYS_START TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN,
SYS_END TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END,
CREATE_ID TIMESTAMP(12) GENERATED ALWAYS AS TRANSACTION START ID,
PERIOD SYSTEM_TIME(SYS_START, SYS_END));

CREATE TABLE MotorCycles_HIST (
MotorCycleNumber INTEGER,Color CHAR(10),
SYS_START TIMESTAMP(12) NOT NULL,
SYS_END TIMESTAMP(12) NOT NULL,
CREATE_ID TIMESTAMP(12));

ALTER TABLE MotorCycles ADD VERSIONING USE HISTORY TABLE MotorCycles_HIST;
```

# Bi Temporal Tables



Main table :

Mo 999-12-31

Mc -12-31

MotorcycleNumber=18,color=BLUE,SysStart=2011-01-01,SysEnd=9999-12-31

Archived table :

MotorcycleNumber=18,color=BLACK,SysStart=1966-01-01,SysEnd=1987-01-01

MotorcycleNumber=18,color=RED,SysStart=1987-01-01,SysEnd=2011-01-01

1) On 1/1/1966 : INSERT 18,black
2) On 1/1/1987 : UPDATE 18,red
3) On 1/1/2011 : UPDATE 18,blue

How to retrieve a historical row :

**SELECT * FROM MotorCycles FOR SYSTEM_TIME AS OF '2010-01-01' WHERE MotorCycleNumber=18-Result =”RED”**
**SELECT * FROM MotorCycles FOR SYSTEM_TIME AS OF ’1965-01-01' WHERE MotorCycleNumber=18-Result =SQLCode +100**
**SELECT * FROM MotorCycles FOR SYSTEM_TIME AS OF '2020-01-01' WHERE MotorCycleNumber=18-Result =”BLUE”**

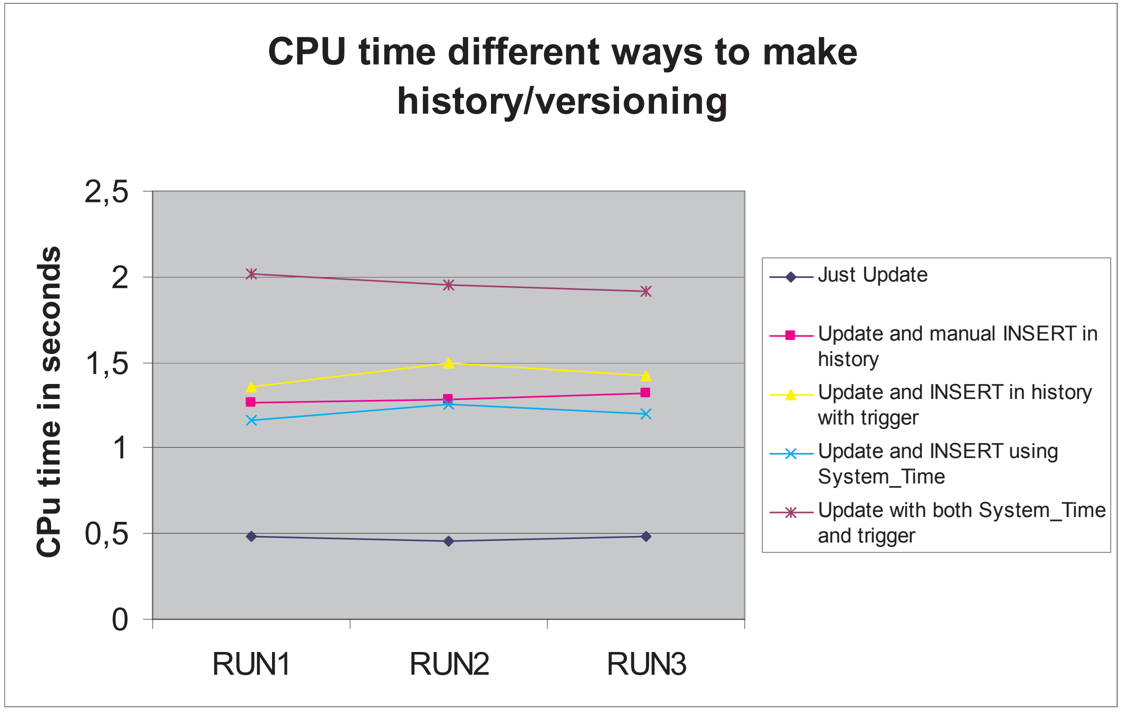
# Bi Temporal Tables

## • Description System Time Cont'd

- A few things to observe :
  - BeginTime inclusive, EndTime exclusive
    - NOT customizable !!
  - Can be DROP'ped
  - Security at the history table is separate
  - SELECT from history table directly is indeed possible, though not advised !
  - INSERT into the history table also possible, even more strongly not advised
  - Will naturally require that you calculate on the space demand for history table.
  - Will only carve 1 history row per row per UOW. Can be a problem if a batch program is updating rows repeatedly per UOW.
  - Restriction to have the same columns in the tables
  - NO row retention time and auto purge… ☹

# Bi Temporal Tables

- Test description and outcome
    - In many industries a common problem
    - Identified business cases-separated test for SYSTEM TIME/BUSINESS TIME
    - Today ***SYSTEM TIME can be achieved by :***
        - ***The application logic***
        - ***Triggers***
    - So ***we compared these approaches from a performance*** perspective

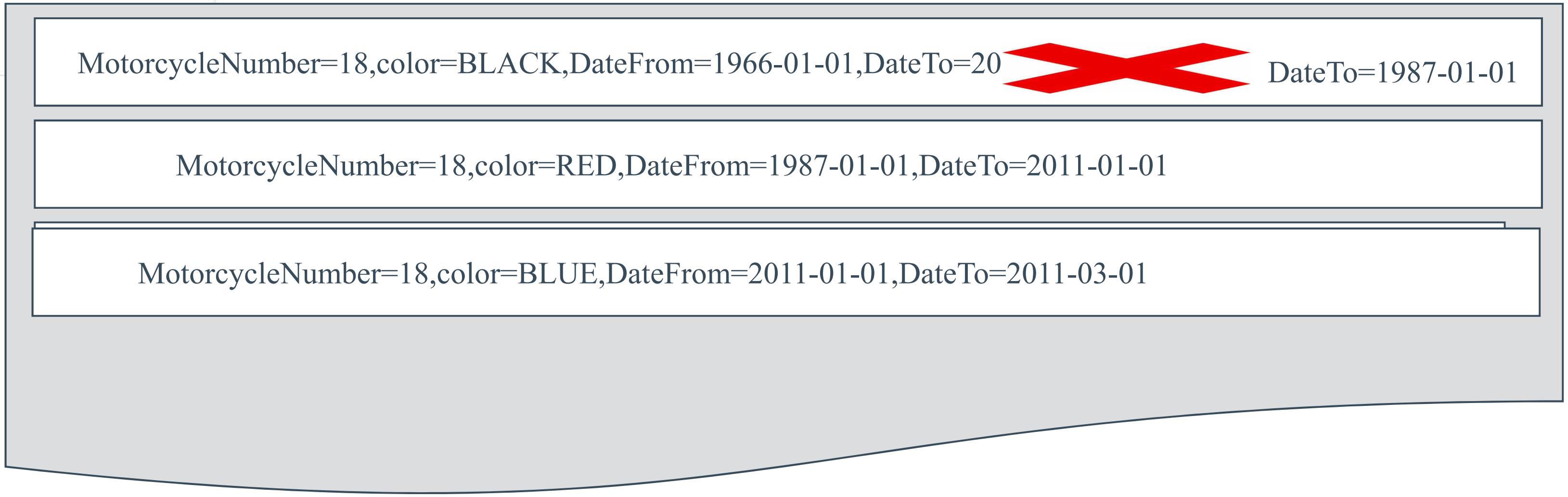


# Bi Temporal Tables

- Description business Time

  - DB2 can now maintain multiple copies of ”same” row on a date/timestamp level

## Main table :



MotorcycleNumber=18,color=BLACK,DateFrom=1966-01-01,DateTo=20 DateTo=1987-01-01

MotorcycleNumber=18,color=RED,DateFrom=1987-01-01,DateTo=2011-01-01

MotorcycleNumber=18,color=BLUE,DateFrom=2011-01-01,DateTo=2011-03-01

1) INSERT MotorCycleNumber=18,Color=black,DateFrom=1966-01-01,DateTo=2011-01-01
2) INSERT MotorCycleNumber=18,Color=red,DateFrom=2011-01-01,DateTo=2011-03-01

3) UPDATE FOR PORTION OF BUSINESS_TIME FROM '2011-01-01' TO '2011-03-01‘Color=BLUE WHERE MotorCycleNumber=18

4) UPDATE FOR PORTION OF BUSINESS_TIME FROM ‘1987-01-01' TO '2011-01-01‘Color=RED WHERE MotorCycleNumber=18

## How to retrieve a historical row :

**SELECT * FROM MotorCycles FOR BUSINESS_TIME AS OF ‘1988-01-03' WHERE MotorCycleNumber=18**
**Result =”RED”**

# Bi Temporal Tables

- Description System Time Cont’d

```
CREATE TABLE MotorCycles (

MotorCycleNumber INTEGER,Color CHAR(10),

Date_From DATE NOT NULL,

Date_To DATE NOT NULL WITH DEFAULT,

PERIOD BUSINESS_TIME(Date_From, DateTo));


CREATE UNIQUE INDEX MotorCycleIx ON MotorCycles (MotorCycleNumber, BUSINESS_TIME
WITHOUT OVERLAPS)………;
```

# Bi Temporal Tables

- BUSINESS TIME - Test description and outcome
  - Identified business cases-separated test for SYSTEM TIME/BUSINESS TIME
  - Build prototype of application
  - Verified the ”splitting” of rows
    - We are not very keen on this. We accept that it is needed function but
      - Would have likes option to force applications to cover complete time span
      - And give SQL code if not.
      - Problem is that an update becomes an INSERT etc.
    - Identified a need for a third dimension (sigh !)
- Attention
  - BeginTime inclusive, EndTime exclusive (NOT customizable !!)
  - Time can be dates or timestamp BUT
    - but not with timezones
    - not with greater precision
  - NO row retention time and auto purge… ☹

# Greater Timestamp Precision

- Description and outcome
  - In DB2 10 timestamp precision can go from micro to pico (6->12)
  - In fact YOU decide how many fractions you want (0->12)
  - Even at the biggest precision no guaranty for uniqueness (in theory)
  - From static or dynamic SQL.

# Timestamp With Timezone

- Description and outcome
  - Designed for data entered from different timezones to enable an easy calculation of duration and conversions.
  - In our part of the world we might be more worried about DST
  - The problem in DST is that z/OS moves the timezone. We analyzed the new TIMESTAMP WITH TIMEZONE to judge if this can make applications tolerant for DST-changes in the future compared to ordinary TIMESTAMP.

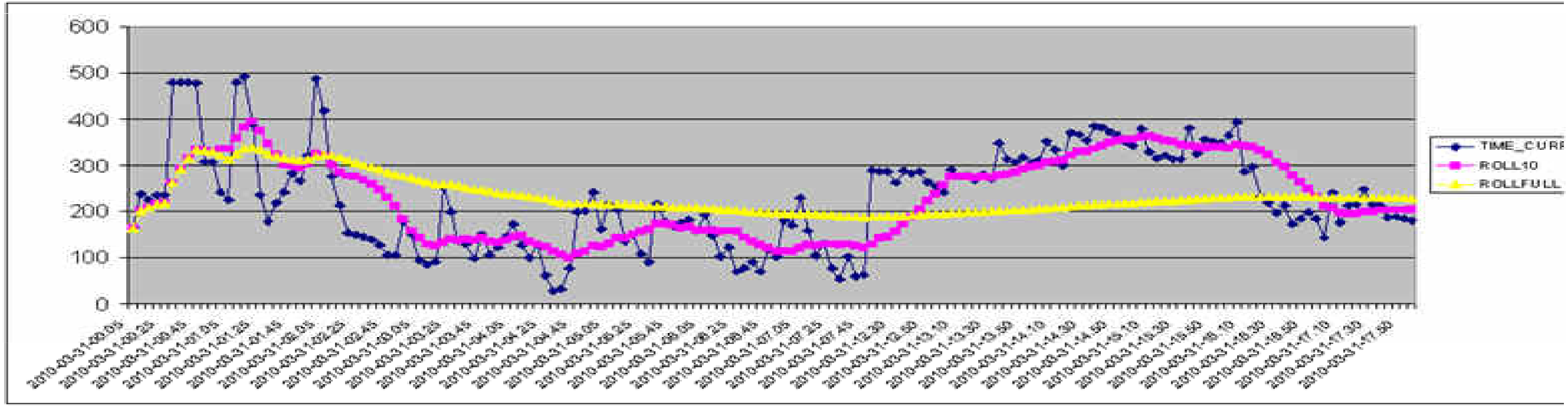
- Attention
  - ***The feature seems "difficult usable" from a DST point of view***. Changes to this feature to make it cope with DST are difficult to spot and would require a discussion.....

# Timestamp With Timezone

- DST discussion :
  - Let’s look at things with “DST-glasses” :
    - Wall-time 08:00 AM gives 2010-03-01-08:00:00:000000+01:00
    - DST Wall-time 08:00 AM gives 2010-04-01-08:00:00:000000+02:00
  - SELECT’ing the data above is not usable for printing customers balances etc so we will do a SELECT TIMESTAMP(Field_with_TZ) :
    - 2010-03-01-08:00:00:000000 and 2010-04-01-08:00:00:000000
  - Now we will try to extract the HOUR(timestamp) :
    - If we take HOUR with no timezone definitions we get the expected 08
    - But if I take HOUR(TIMESTAMP_WITH_TZ) I get this for the 2 rows:
      - 07 and 06 which is the value of the GMT representation.
  - To get the '''real''' hour-value : HOUR(TIMESTAMP(TIMESTAMP_WITH_TZ))
  - It will require a lot of testing to implement this feature with DST-glasses in applications because it will be very easy for programmers to make mistakes….
  - It seem that this new TIMESTAMP_WITH_TZ is not build for handling DST only to handle data from different timezones. ***One could argue that the real problem is the implementation of DST in z/OS.***

# Moving Sum and Average

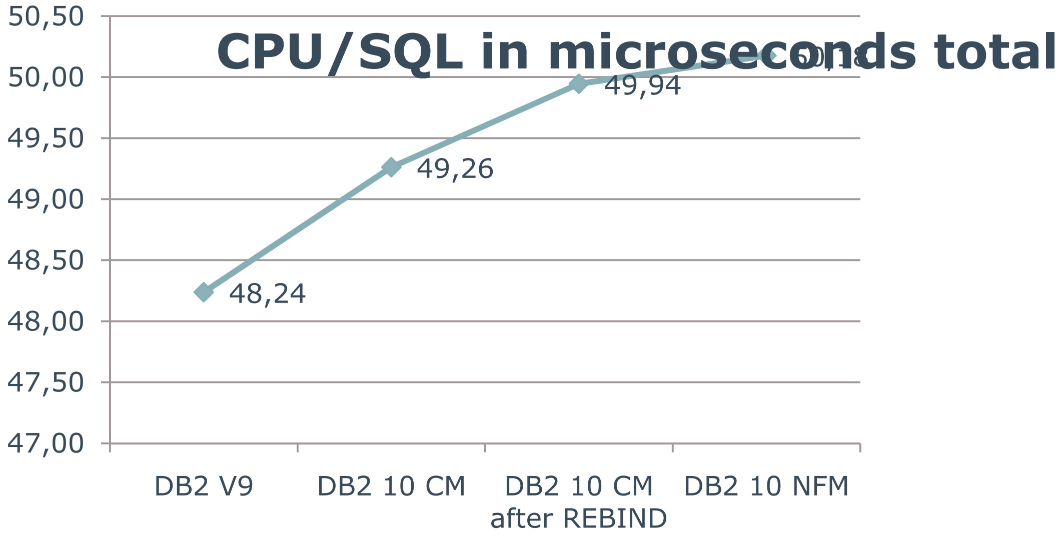
- Description and outcome
  - New OLAP functions - full controllable
  - The 'moving' can be by the "last X" or ?
  - INT(AVG(TIME_CURRENT) OVER (PARTITION BY MY_KEY ORDER BY COLX ROWS BETWEEN 10 PRECEDING AND CURRENT ROW))AS AVG_10
  - We then did some tests on a business case using different moving windows to judge the impact.
  - All works very well indeed and the graphs look as hoped.
  - Attached is a hardcopy of a graph containing the single plots (blue curve), moving average last 10 occurrences (pink curve) and moving average total (yellow curve) and the Excel-sheet behind this graph....

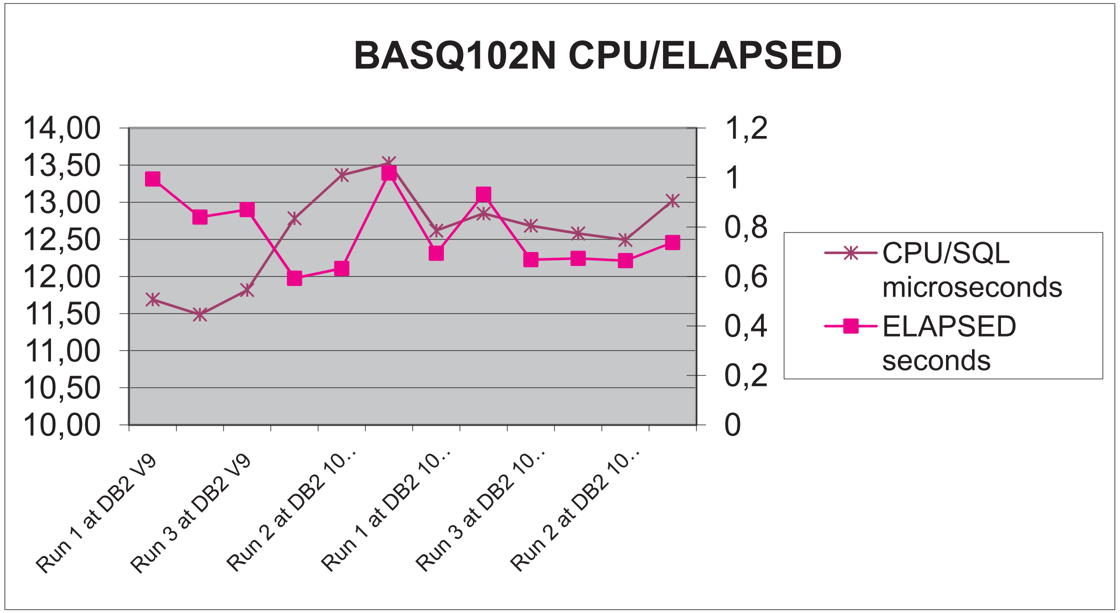


# Performance testing

- Planned was a “real upgrade” of the QA environment.
  - Identification of “workload” with DDL, Binds, CICS, IMS Batch and standalone Batch (CAF)
  - Collecting performance figures before migration.
  - Re-doing at DB2 CM
  - Re-doing at DB2 CM after REBIND
  - Re-doing after going to NFM.
- Decided to clone the QA environment back to another LPAR
  - Was not planned – resources not allocated (man-power, disk, JCL)
  - Gave a boost in progress as we were back in the driver seat !
  - Good for fall-back and redoing scenarios
  - Bad as we had to create a reduced/phony workload
- Difficult to compare performance issues in an understandable way.
  - Huge report with Excel and SMF shipped to IBM
  - Several areas identified by IBM for reason for degradation
  - Tests not redone after changes
  - Bankdata has highly efficient static SQL-did not hit DB2 10 “sweet spots”

# Performance testing

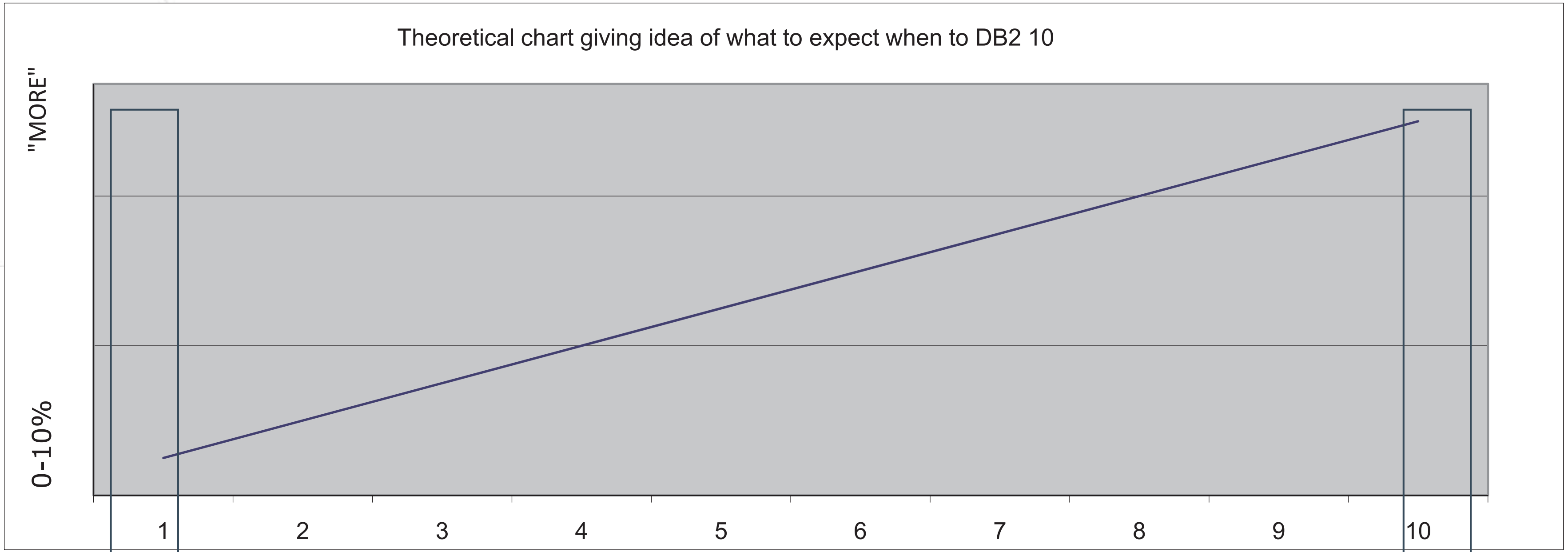




# Performance testing

- The promised CPU reduction is expected to vary greatly
  - Between installation and workloads
- “Out of the box” – yes
  - But small adjustments may be needed to gain the maximum
  - Go through the list of “sweet spots” and level set your expectations
    - Across your applications and peaks
  - Verify your expectations ASAP in a suitable environment
    - Before production
  - Remember the expectations that your boss has !
    - Your boss has been reading the release info of DB2 10
  - Do your home work early :
    - Identify things that will bring you savings with minimal investment
    - Verify !!
- Good news :
  - IBM’s performance experts have identified why we got degradation
  - APAR’s on its way
  - We hope to be able to redo scenario when fixes available

# Frank’s unofficial Performance chart



- Concurrent inserts in sequential (or skip sequential) order (CM)
- Concurrent inserts into non-partitioned by taking an advantage of PBG Member Cluster
- Queries with multi range predicates and/or IN predicates (stage1) & evaluating many rows (CM)
- Queries with stage 2 predicates and evaluating many rows (CM)
- Native SQL procedures after regenerated in DB2 10 (CM)
- Workloads prev. tuned to save virtual storage and can take advantage of DB2 10 VSCR - using protected threads, RELEASE DEALLOCATE bind option (CM)
- Distributed workload by turning on high performance DBATs in DB2 10 (CM)
- Dyn.SQL with KEEPDYNAMIC YES with limited MAXKEEPD values by increasing this (CM)
- Concurrent insert suffering from log latch contentions - LC19 (CM/NFM)
- Online transactions requiring a large number of package allocation - LC24 (CM)
- Queries with CPU parallelism - LC24 (CM)
- Dyn.SQL using literals by enabling CONCENTRATE STATEMENTS WITH LITERALS (NFM)
- JCC type 2 (e.g., WAS on z/OS) or ODBC on z/OS applications executing queries returning multiple rows by using limited block fetch (CM)
- Workload accessing small sized LOB by taking an advantage of INLINE LOB (NFM)
- Workload updating small nodes on large XML documents by using XML MODIFY (NFM)
- Applications accessing date in random orders by taking an advantage of HASH ACCESS (NFM)
- Applications accessing the table with multiple indexes by using INCLUDE index (NFM)
- By replacing application logic (triggers, stored procedures) with Bi-Temporal features (NFM)

<< >>

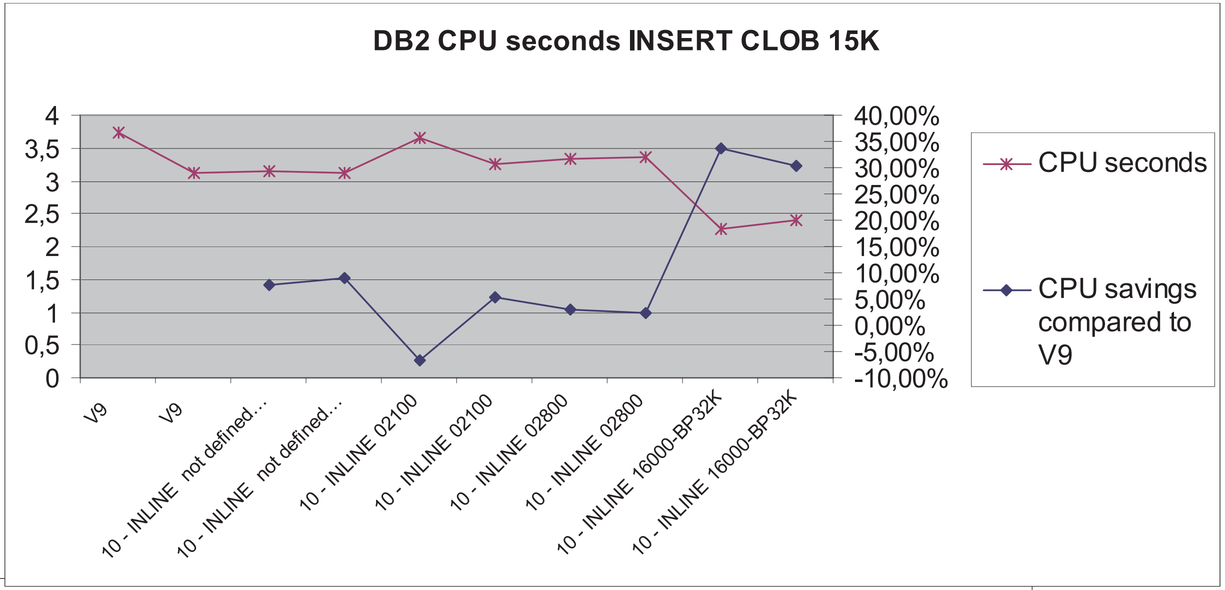
# Inline LOBS

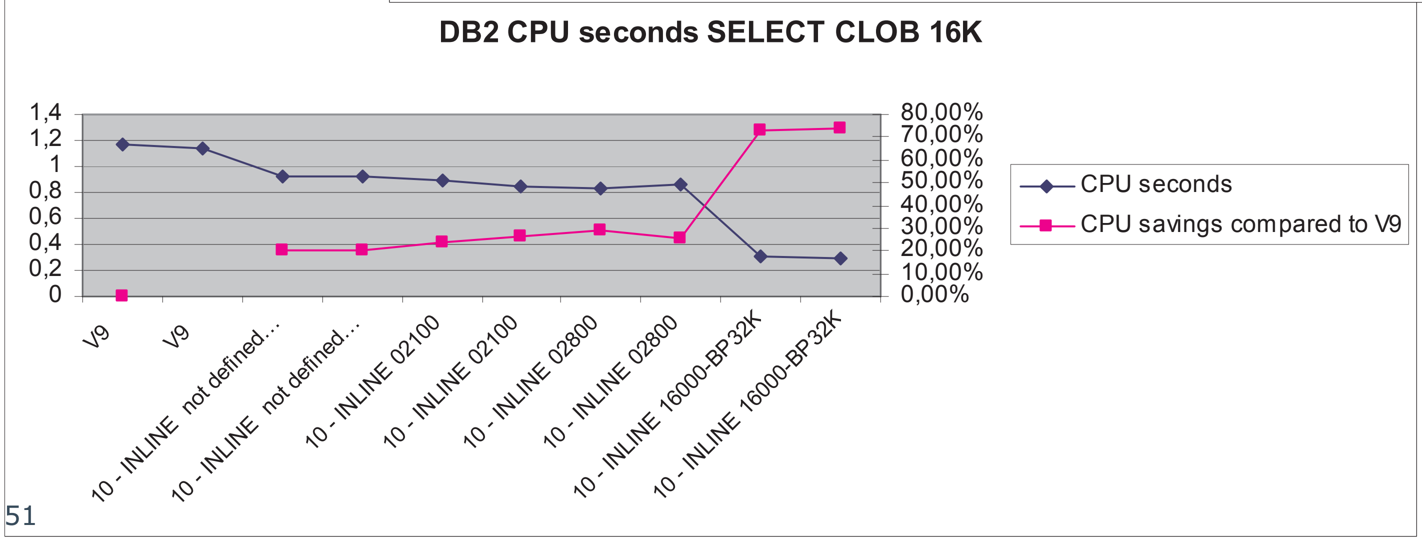
- Description and outcome
  - New Zparm LOB_INLINE_LENGTH sets the installation standard
  - Can also be defined in the DDL
  - If a given LOB is smaller than this LOB goes to base table.
  - If a given LOB is greater it partly goes to base and partly to aux.
  - Expect a massive performance benefit
    - On both SELECT and INSERT
    - Index on expression can be defined for inline portion
  - Testing was done by a purpose build copy of an existing Bankdata program.
    - INSERT XML-documents of varying length and SELECT those again.
    - The SELECT can be controlled to limit the SELECT to LOBs with a given length.
    - We have then performed executions with following combinations :
      - At DB2 V9
      - At DB2 10 with no INLINE length
      - At DB2 10 with INLINE length of 2100 bytes, 2800 bytes, 16000 bytes
  - The idea is that as we move the INLINE LENGTH up more and more of the documents will be kept in the base table.

# Inline LOBS

•Description and outcome (cont'd)

- A few measurements

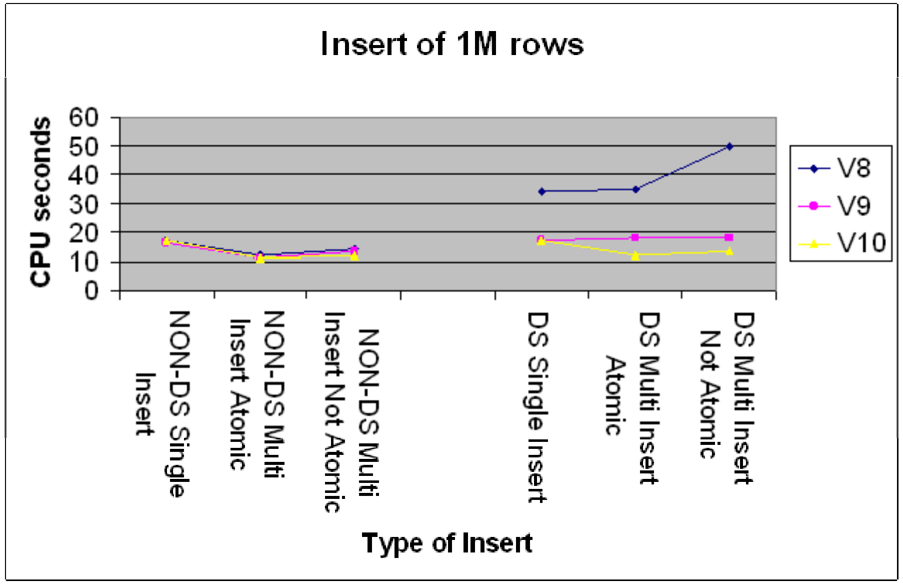




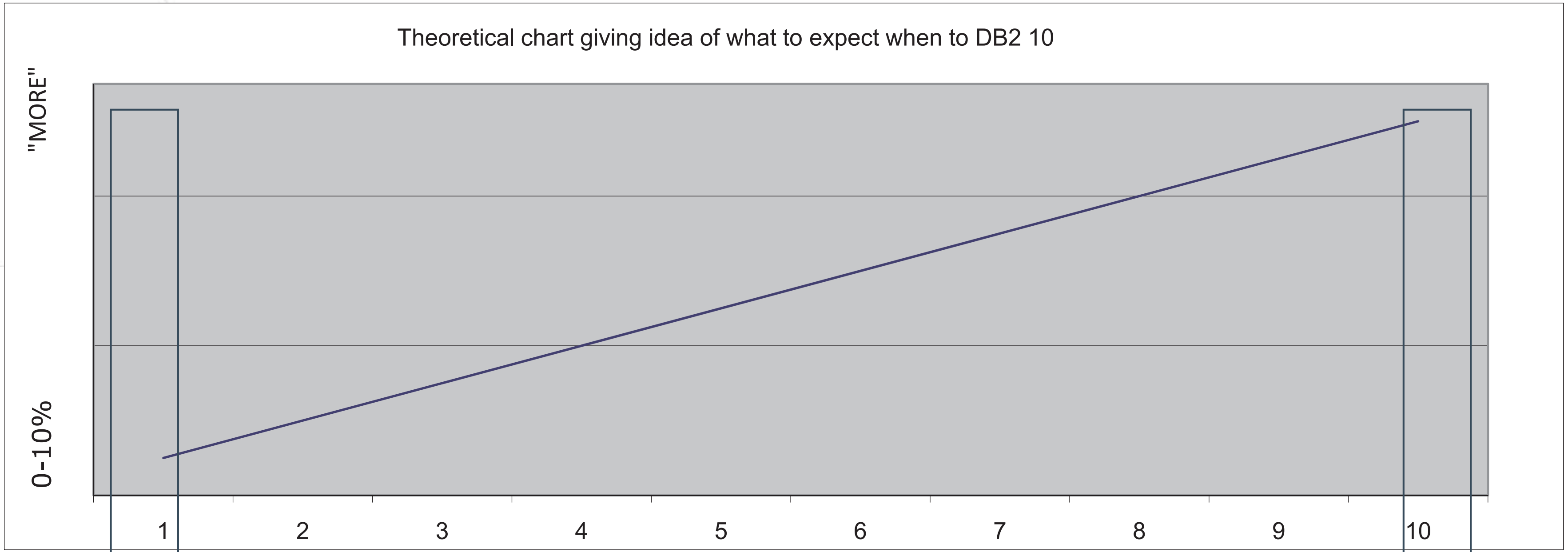
bankdata

# Reduce Log-Latch Contention

- Description and outcome
  - A problem with unknown impact for heavy inserts in Data Sharing.
  - In version 8 spin loop in log record carving
    - In version 9 partly solved
  - A benchmark comparing V8->V9->V10 both datasharing/non-datasharing in all thinkable combinations of multirow and singlerow.
  - ***The conclusion is that DB2 10 DS can now run as fast as non-DS and finally take full advantage of MultiRow Insert.***
  - ***We are extremely pleased with this result !***



# Frank’s unofficial Performance chart



Highly optimized Static SQL

”User defined” Dyn. SQL with literals

- Concurrent inserts in sequential (or skip sequential) order (CM)
- Concurrent inserts into non-partitioned by taking an advantage of PBG Member Cluster
- Queries with multi range predicates and/or IN predicates (stage1) & evaluating many rows (CM)
- Queries with stage 2 predicates and evaluating many rows (CM)
- Native SQL procedures after regenerated in DB2 10 (CM)
- Workloads prev. tuned to save virtual storage and can take advantage of DB2 10 VSCR - using protected threads, RELEASE DEALLOCATE bind option (CM)
- Distributed workload by turning on high performance DBATs in DB2 10 (CM)
- Dyn.SQL with KEEPDYNAMIC YES with limited MAXKEEPD values by increasing this (CM)
- Concurrent insert suffering from log latch contentions - LC19 (CM/NFM)
- Online transactions requiring a large number of package allocation - LC24 (CM)
- Queries with CPU parallelism - LC24 (CM)
- Dyn.SQL using literals by enabling CONCENTRATE STATEMENTS WITH LITERALS (NFM)
- JCC type 2 (e.g., WAS on z/OS) or ODBC on z/OS applications executing queries returning multiple rows by using limited block fetch (CM)
- Workload accessing small sized LOB by taking an advantage of INLINE LOB (NFM)
- Workload updating small nodes on large XML documents by using XML MODIFY (NFM)
- Applications accessing date in random orders by taking an advantage of HASH ACCESS (NFM)
- Applications accessing the table with multiple indexes by using INCLUDE index (NFM)
- By replacing application logic (triggers, stored procedures) with Bi-Temporal features (NFM)

<< >>

# Conclusions Beta process

- Total # of hours accounted on beta :
  - Approx 2000 !!!
  - Equals one fulltime man year
  - Estimated 5 hours administration per week

- Opened 35 pmr's
  - Assumed to be average per beta customer
  - So some hundred problems removed before GA
- Requested many information (doc) changes
  - Doc been read as neutral as possible
  - Not just requested errors to be corrected
  - But also things difficult to understand

# •Conclusion DB2 10

- DB2 10 is an important release
- Will give significant CPU saving in many applications
- Delivers extremely useful application features :
  - Temporal tables
  - XML
  - Hashing
- Sets the cornerstone to new important features :
  - Read last committed version of data
  - Autonomic maintenance (hopefully)
- Finally solves some scalability and administrative issues :
  - VSCR – Reduce # of sub-systems
  - Online schema evolution – next step taken
  - New DBA privilege
- But don’t under-estimate the smaller issues :
  - Catalog restructure
  - Statement level monitoring

# There is more to life than IT…..

# DB2 update seminar – bonus material

# SQL Pagination

- Description and outcome
  - V9 can not use OR predicate as matching predicate with single index access.
  - In DB2 10 when using complex ”OR” as matching predicate the optimizer now can pick single index access
  - Example: WHERE (LASTNAME='JONES' AND FIRSTNAME>‘HANS‘) OR (LASTNAME>’HANS’) ORDER BY LASTNAME, FIRSTNAME;
  - Mostly used when reposition cursors in batch programs and when repositioning for next page in CICS ?
  - Plan_table reflects new accesstype “NR”
    - Means that DB2 V10 will by more effective index exploitation improve the query performance
  - “Perhaps” no code changes ?
    - (There is “something” with sequence of predicates)
- Effective duration
  - 2 Days effectively

# Online Reorg Enhancements

- Description and outcome
  - Option to cancel blocking threads
  - Allow disjoint partition ranges
  - Message to estimate length of reorg phases and time to complete
  - The cancel was found to be of limited use as it did not do what we hoped
  - AUX YES/NO to control reorg of LOB table spaces

- Attention :
  - Blockers have to do work for cancelation.
  - If not the risk is that BOTH blocker and reorg goes away
  - Read the manual carefully
    - Validate your environment accordingly !
- Effective duration
  - 5 days effectively

# REORG SHRLEVEL CHANGE for LOBs

- Description and outcome
  - DB2 V10 promised online reorg for LOB's.
  - A reorg SHRLEVEL REFERENCE is needed on LOB's to rebalance partition.
  - The keyword DISCARD on base table can discard associated LOB values.
  - An online reorg can shrink a tablespaces with LOB's.

- Attention :
  - The DISCARD option does not work when LOB is larger than 32K.

- Effective duration
  - 5 days effectively

# Add Active Log

- Description and outcome
  - Enables one to add new active datasets “on the fly”
  - Is this for emergency (tape system failure) ?
    - Perhaps more for planned changes to log datasets
    - Only Add – No delete nor mark as NotReusable

- Attention :
  - It is possible that a LOG SWITCH takes place between add of COPY1 and add of COPY2. So suddenly we were running on COPY1 of dataset number X and COPY2 of dataset number X-1.
    Normally we will always run on same pair of datasets and this might confuse automation procedures and staff.

- Effective duration
  - Few hours

# Auto Compress on Insert

- Description and outcome
    - Problem has been when table is starting empty and massively inserted
    - Needed a (online) reorg to build a compression dictionary.
    - Has given outage and administrative task in installations.
    - In DB2 10 COMPRESS YES is honored by inserts.
    - The compression ratio is to be expected a bit lower as to build by reorg
        - Dictionaries will be build on a smaller or different subset of the data.
- Effective duration
    - 3 days effectively

# Auto Stats

- Description and outcome
    - **SET PROFILE** used to allow RUNSTATS to generate a profile from the options specified in the RUNSTATS invocation and store this in the system catalog tables.
    - Scheduling autonomic statistics monitoring..
        - ***No Data Studio support at the time tested so we wrote java code :***
        - Call the ADMIN_TASK_ADD stored procedure to schedule calls to the ADMIN_UTL_MONITOR stored procedures in the admin. task scheduler.
        - The system supplied Stored Procedures have to be correctly configured and running.SYSPROC.DSNUTILU is used to execute RUNSTATS for objects.
        - Administrative Scheduler Address Space has to be up and running.
- Attention :
    - Look for the implementation in your favorite tooling package
        - For IBM this is DataStudio and Automation Tool ?
    - Coding your own frontend is COMPLICATED !
- Effective duration
    - 7 days

# V10 Installation and Samples

- Description and outcome
  - Migration is PARTLY business as usual
  - New installation/migration checklists….
    - ✗ Sysprogs are going directly to migration step 1… ;-)

mature

  - Skip level migration – go from V8 to 10.
    - New phase names in play
  - DSNTINST can now save work in progress (“SAVE”)
  - New job to merge DSNTIDXA member with new parms “offline”
  - Remember the elimination of Private Protocol
  - As always examine new Zparm values and synchronize these
  - Clone cat/dir/logs/BSDS to other LPAR to try out CatMaint ?
- Effective duration
  - 3 days

# Online Schema Evolution

- Description and outcome
  - All thinkable (??) conversions between TS types
  - Change to page size
    - Index pagesize requires UTS
  - Change to segsize
  - Change member cluster on/off
  - Drop pending changes
  - Alter MAXPARTITIONS will convert segmented table tablespaces to UTS PBG

- Attention :
  - Conversion from classic partitioning or UTS PBR to UTS PBG did not make GA.
  - This can delay the usage of PBG in the installations.

- Effective duration
  - 8 days effectively

# Automated DB2 Supplied SP Install

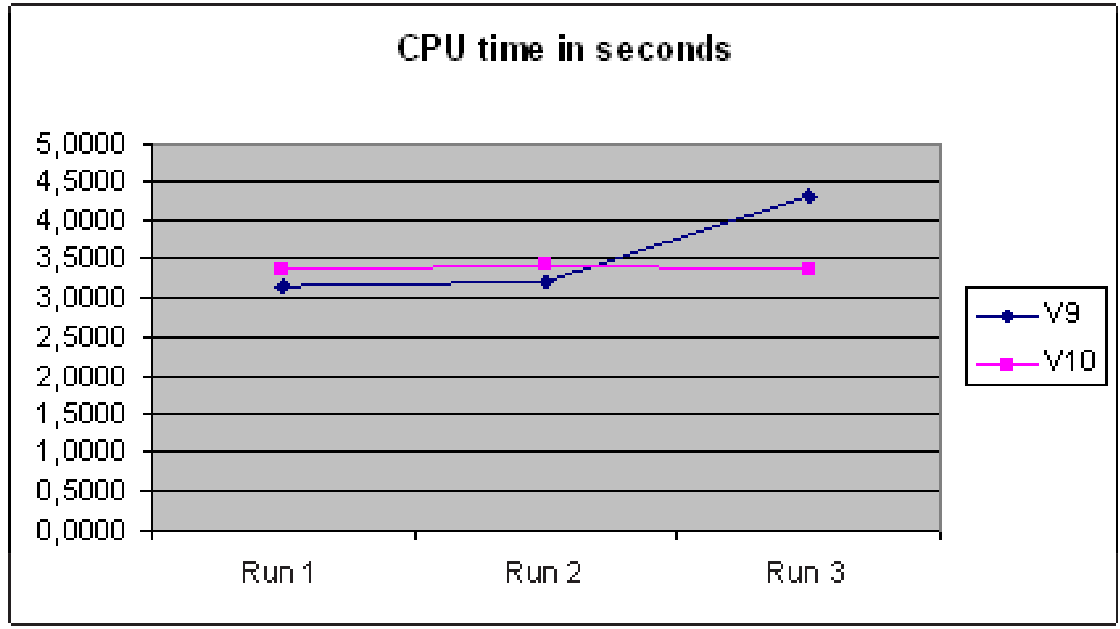
- Description and outcome
  - Elimination of definition errors from these system SP
  - The # of system SP and UDF is ever increasing and DDL was in many jobs
  - Job DSNTIJRW.
    - Defines things in WLM and activate the policy
    - Because of divided responsibilities it shops between z/OS and DB2 sys.progs this will require an agreement between the 2 parties.
  - Job DSNTIJRT.
    - Will validate and redo all settings for the DB2 objects including binds
    - There is PREVIEW option to validate the impact
    - The job still requires manual process to verify, run and check afterwards but will provide valuable benefit in many shops.
  - Job DSNTIJRV
    - Is supposed to check if routines are operative. But DSNTIJRV does NOT validate the state of the packages so -805 can still be received after service !
- Effective duration
  - 15 days effectively

# Catalog Restructure

- Description and outcome
  - No more Links and Hashing in catalog.
  - PBG and Row Level Locking used for many spaces
  - Online reorg SHRLEVEL CHANGE on all spaces
  - SMS and DB2 controlled VSAM. No more manual define of datasets !!
  - We build a scenario with 10 parallel reorgs where only 2-4 survive at V9
    - At DB2 10 all 10 survives
  - Several scenarios using BINDs , Utilities and DDL that gave deadlock in V9.
    - The deadlock disappeared in all cases EXCEPT at the heavy DDL jobs
- Attention :
  - We ***still see some deadlocking if running parallel heavy DDL*** on different databases. This is a ***know limitation*** in the code as it is now. There is no doubt that the majority of the locking problems seen in ordinary administration has disappeared. Some degree of additional elapsed time must be expected but will be a very welcome trade off.
- Effective duration
  - 10 days effectively

# Eliminate LOB Materialization

- Description and outcome
  - V8 introduced LOB/XML streaming in DRDA but not in DB2 engine.
  - DB2 pre 10 takes chunks of 1 Mb storage to materialize object
  - DB2 10 introduces streaming in the DB2 engine
    - Only when 1 LOB/XML per row
    - Other cases still
  - Small java program t
  - Ran at both V9 and 1
  - Accounting shows dif
    - MAX STO LOB V
    - In V10 it doesn't



- Attention :
  - Storage reduction
  - ***CPU time looks more stable in V10, but not always better.***
- Effective duration
  - 2 days effectively

# UTSERIAL Elimination

- Description and outcome
  - DB2 has been recoded removing entirely the UTSERIAL latch
    - Introducing a new Utility locking protocol
  - Build a test scenario in V9 that hit this latch
  - Redid in DB2 10 and verified (and that no other obstruction replaced it)
  - Great !! One of those small things that removes some frustrating episodes.
- Effective duration
  - 1 day effectively

# Eliminate DDF Private Protocol

- Description and outcome
  - This is a DB2 V9 ACTIVITY. Get rid of PP before DB2 10
  - IBM delivers several tools:DSNTIJPM/DSNTIJPA, DSNTP2DP,DSN1PPTP
  - Install the V9 ALIAS resolution "fix" (PK64045) and understand it.
    - Set Zparm DRDA_RESOLVE_ALIAS and make all this work at DB2 V9.
  - Find and analyze all packages bound with DBPROT(P). These have to be rebound to DBPROT(D). Validate this can happen on V9 without any negative side effects (All DRDA applications continue to work).
  - We tried a large amount administrative Rexx code using 3-part names after setting this.
  - We had some difficulties in getting this to work, but it eventually turned out to be definition problems both in the CDB and in RACF.
  - After these problems were cleared ***3 part names from dynamic SQL worked in all combinations of DB2 V9 and DB2 10 and using SNA as well as IP.***
- Attention :
  - ***None – but do your V9 homework carefully !!***
- Effective duration
  - 5 days effectively

# New DBA Privileges

- Description and outcome
  - Desire to separate admin. rights from data rights
  - New EXPLAIN privilege gives explain, prepare and describe
  - New SQLADM gives all privileges to monitor and tune SQL
  - New DATAACCESS gives access to all data
  - New ACCESSCTRL privilege lets users perform GRANT/REVOKE
  - We tested:
    - ACCESSCTRL, DATAACCESS, DBADM with and without ACCESSCTRL and DATAACCESS, SQLADM and the EXPLAIN privileges
  - The basic procedure for testing was:
    - Verified that the initial privilege where as expected.
    - Granted or Revoked privileges
    - Verified any changes in authorizations, due changes in privileges
- Effective duration
  - 6 days effectively

# Buffer Pool Enhancements

- Description and outcome
  - Need reduction of Latch 14+24 when very large BP's
  - BP now allocated in chucks on demand !!!!
  - Exploitation of z/OS 1 MB page sizes
    - LFAREA set in SYS1.PARMLIB and PGFIX(YES)
    - Can be seen in RMF3 - DBM1 is using Large memory object
    - Please understand the impact for other subsystems as well.
    - Expect reduction in latches (LC24)
  - Stay resident in bufferpools - PGSTEAL(NONE) and PGFIX(YES)
    - Optimizer assumes IO cost=0
    - If BP to small pages will be stolen using FIFO
    - Data will be reloaded at first application which also will pay the cost.
    - Alternatively use  ACCESS DATABASE MODE=OPEN

- Effective duration
  - 5 days

# Statement-level opt. hints

- Description and outcome
  - Many installation have packages where optimizer might pick ”bad” access path.
    - With this feature many of these problems can be eliminated.
  - How does it work ?
    - Define DSN_USERQUERY_TABLE
    - Insert the statement text into the DSN_USERQUERY_TABLE
    - Issue a BIND QUERY command.
      - DB2 creates hints for **every** row in DSN_USERQUERY_TABLE
      - Result stored in various other system tables
    - Now you can bind the application with OPTHINT
      - DB2 will search these system tables and favor the accesspath found.
- Attention :
  - At the present there is a problem with cursor names using hyphens (“-”)
    - PMR on its way
- Effective duration

  7 days effectively

# Dyn. Statement Cache Enhancements

- Description and outcome
  - One of the smaller things in DB2 that might have a huge impact
  - ***DB2's now has a way of replacing literals with a new special marker so that the constants will not be cached as the part of the SQL text.***
  - Doing this we can ***avoid the "full prepare"*** and instead benefit from the "short" Cache prepare like the behavior of parameter markers.
  - Accounting information can be used to judge impact
  - New syntax on the prepare:
    - **CONCENTRATE STATEMENTS WITH LITERALS**
    - Can be set manually on the PREPARE ATTRIBUTES
    - Is build into newest drivers for DRDA
      - Use the JCC Driver setting to include the "enableliteralReplacement='YES'" in the data source or connection property.
      - Set LITERALREPLACEMENT in the ODBC initialization file in z/OS.
  - LIKE predicate not supported
- Effective duration
  - 3 days

# Support For DataStudio Monitoring

- Description and outcome
  - Consist of several enhancements :
    - Additional info for DSNT375I & DSNT376I
      - Mostly for distributed threads
    - Statement level monitoring for static and dynamic SQL
      - Produces IFCID records with timing info per statement.
      - Can remove much of the need for doing tracing
      - YOU can do IFI programming
      - Or expect support for this in your tooling
    - PROFILING to catch or limit distributed threads
      - Can filter on IP-adresses, users, driver-level etc etc
      - Can give info-message or actually limit access
      - Hopefully you can see who/what is being queued ?
      - Gives extreme control !!!
- Effective duration
  - 20 days effectively

# I/O Parallelism for Index Updates

- Description and outcome
  - New Zparm INDEX_IO_PARALLELISM=YES/NO
  - To check if you are using I/O parallelism for index update you can investigate ***the new IFCID's*** 357,358 to judge the degree of parallelism.
  - ***Is not kicking in for sequentially inserts but only for random.***
  - ***In tailored test sync I/O's against indexes have been reduced up to 40%*** which means reduction of I/O wait time as well as class 2 CPU cost for performing sync I/Os.
    - Tailored means way to small BP's
  - Understand this is a feature that will kick in and out at will. It can be debated if it is controllable or autonomic.
  - Understand that it is NOT parallel index update but parallel I/O's for index page updates
  - The parallel I/O's will be done by DBM1 SRB's, thus zIIP eligible. !
    - Also true for Sequential Prefetch
- Effective duration
  - 5 days effectively

# Additional non-key Columns in Index

- Description and outcome
  - Can reduce the # of indexes on a given table
    - Thus reduces INSERT time
    - Can give index-only access
    - Simplifies administration
    - DB2 LUW has this feature
    - But index will become more "fat"
  - At our installation ***we are now able to reduce the number of indexes on more than 100 tables***.
- Attention :
  - Be careful as ***altering an index to add an Include Column throws the index into Rebuild Pending status.***
  - One solution is to create a new index with defer yes, rebuild this index with shrlevel change and update statistics. Then drop the old index and do a rebind.
- Effective duration
  - 6 days effectively

# High Performance DBAT Support

- Description and outcome
  - ***Zparm: CMTSTAT has to be INACTIVE***
    - CMTSTAT=INACTIVE uses less storage, and threads can be pooled and reused. This is required for high performance DBAT.
  - Activate with :CMTSTAT=INACTIVE, packages bound with RELEASE(DEALLOCATE) and MODIFY DDF PKGREL(BNDOPT).
  - ***Several tests with combinations of the settings has been done.***
  - ***Saved about 1% CPU per transaction, compared to CMTSTAT=ACTIVE***
- Effective duration
  - 3 days