#SHAREorg

SHARE
Technology · Connections · Results

# Deep Dive into IBM DB2 Analytics Accelerator Query Acceleration

Guogen Zhang and Ruiping Li
IBM

August 9, 2012
Session 11588

SHARE in Anaheim 2012

# Agenda

- IDAA design objectives
- Overall architecture and usage cycle
- Query acceleration
- Performance considerations
- Future directions

# IDAA: IBM DB2 Analytics Accelerator

*Accelerating decisions to the speed of business*

*Blending System z and Netezza technologies to deliver unparalleled, mixed workload performance for complex analytic business needs.*

**Get more insight from your data timely**

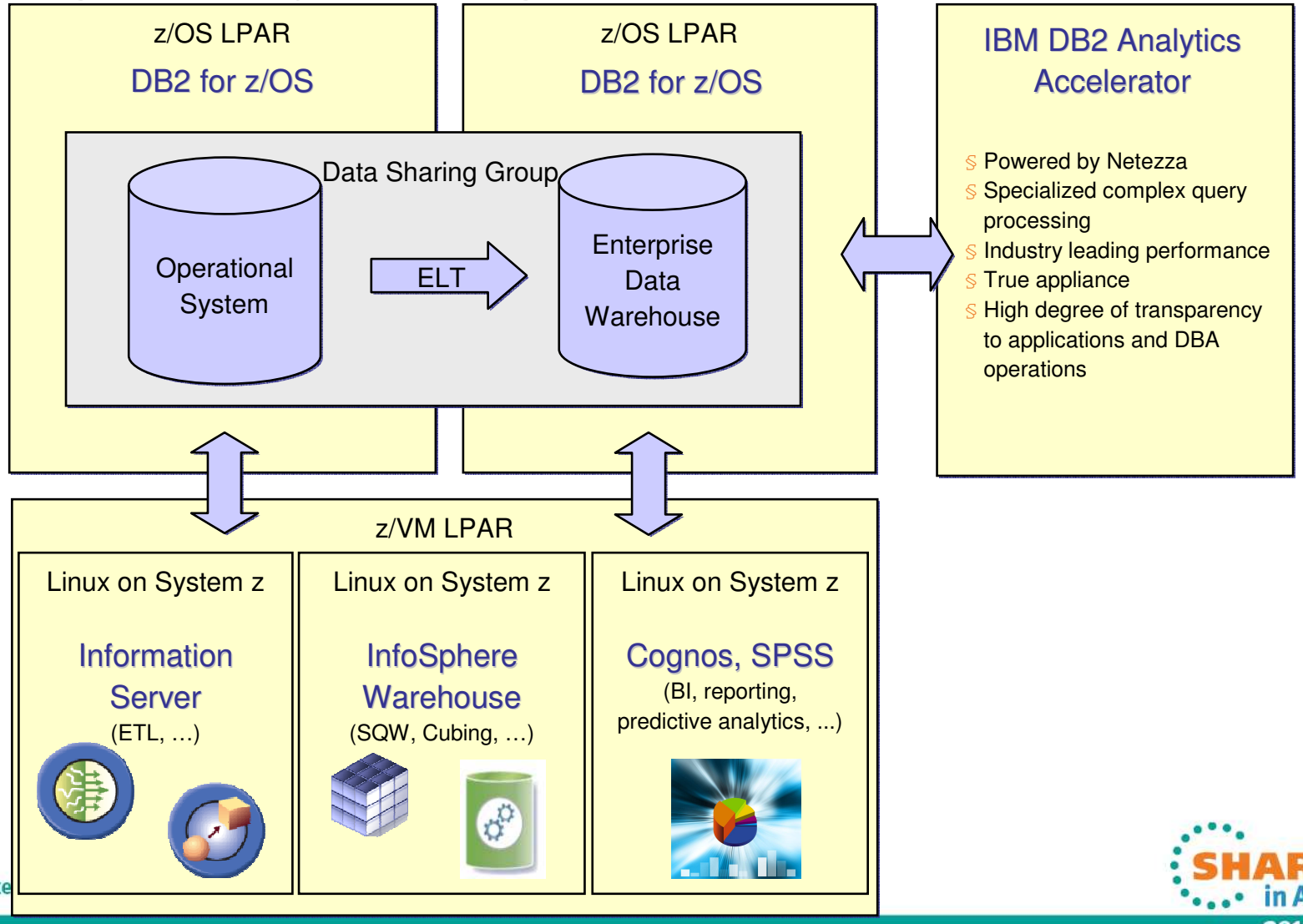
- Fast, predictable response times for “right-time” analysis
- Accelerate analytic query response times
- Improve price/performance for analytic workloads
- Minimize the need to create data marts for performance
- Highly secure environment for sensitive data analysis
- Transparent to the application

# IDAA Design Objectives
## *It's an accelerator*

- DB2 continues to own data (both OLTP and DW)
  - Access to data (authorization, privileges, …)
  - Data consistency and integrity (backup, recovery, …)
  - **Enables extending System z QoS characteristics to BI/DW data as well**
- Applications access data (both OLTP and DW) only through DB2
  - DB2 controls whether to execute query in DB2 mainline or route to IDAA
  - DB2 returns results directly to the calling application
  - Existing applications do not have to change.
  - **Enables mixed workloads and selection of optimal access path (within DB2 or IDAA) depending on access pattern**
- IDAA as a virtual DB2 component
  - DB2 provides key IDAA status and performance indicators as well as typical administration tasks by standard DB2 interfaces and means
  - No direct access (log-on) to IDAA accelerator
  - **Enables operational cost reduction through skills, tools and processes consolidation**

# IBM DB2 Analytics Accelerator

*Adding Industry Leading Performance*



# Feedback Customers: Fast Time to Value

- IBM DB2 Analytics Accelerator (Netezza 1000-12)
  - Production ready - 1 person, 2 days
- Table Acceleration Setup … **2 Hours**
  - DB2 “Add Accelerator”
  - Choose a Table for “Acceleration”
  - Load the Table (DB2 copy to Netezza)
  - Knowledge Transfer
  - Query Comparisons
- Initial Load Performance …
  - 5.1 GB in 1 Min 25 Seconds (24M rows)
    400 GB in 29 Min (570M rows)
- Actual Query Acceleration … **up to 1908x as fast**
  - 2 Hours 39 Minutes to 5 Seconds
- CPU Utilization Reduction … **99% less CPU**
  - 24M rows: 56.5 CPU seconds to 0.4 CPU seconds

# Accelerating SAP for more business value

- **Enhances the SAP Certified DB2 for z/OS**
- **Accelerates SAP NetWeaver BW**
- **Dramatic decrease in elapsed** time for **SAP BW ad-hoc reporting**

| No | Description | Records read | Records returned | DB2 [sec] | IDAA [sec] | Accel. factor |
|---|---|---|---|---|---|---|
| 1 | Simple mass aggregation | 17116647 | 21 | 117 | 0.78 | 150 |
| 2 | Query #1 + 70% filter | 11980812 | 21 | 94.2 | 0.86 | 110 |
| 3 | Query #1 + 30% filter | 5133708 | 21 | 54.8 | 0.82 | 67 |
| 4 | Query #1 + 10% filter | 1710293 | 21 | 17.6 | 0.87 | 20 |
| 5 | Screwed data, low filtering | 10790019 | 21 | 96.8 | 2.47 | 39 |
| 6 | Screwed data, high filtering | 24 | 14 | 7.28 | 0.83 | 9 |
| 7 | Many restrictions | 3805941 | 21 | 128 | 7.65 | 17 |
| 8 | Navigational attributes | 823646 | 21 | 17.1 | 1.27 | 13 |
| 9 | Navigational attributes + selective condition | 811 | 21 | 15.8 | 1.17 | 14 |
| 10 | Open value ranges | 2006 | 21 | 19.6 | 3.52 | 6 |
| 11 | Hierarchy | 1653981 | 21 | 17.6 | 0.97 | 18 |
| 12 | Hierarchy + selective condition | 55068 | 21 | 38.6 | 0.98 | 39 |
| 13 | Restricted key figures on 2 dimensions | 1314964 | 1948 | 207 | 7.22 | 29 |
| 14 | Query #14 + hierarchy | 132564 | 1499 | > 1000 | 1.27 | > 787 |
| 15 | Calculated key figures (OLAP) | 5321586 | 10 | 57.8 | 2.37 | 24 |
| 16 | OR linked values | 6212609 | 13 | 40.5 | 0.92 | 44 |
| 17 | Non uniform data distribution | 11016253 | 13 | 31.2 | 0.99 | 32 |
| 18 | Selective line item | 1724 | 1706 | 0.71 | 1.17 | 0.6 |
| 19 | Non-selective line item | 115481 | 68619 | 33.8 | 1.36 | 25 |
| 20 | All together | 3087692 | 468 | 87.7 | 4.42 | 20 |

Table 2: Dedicated Query Test on a 18-million-records InfoCube

# Netezza Strengths

- True appliance
  - Single integrated unit
  - Purpose fit
  - Pre-tuned
  - Deeply leverages hardware components (e.g., FPGA)
  - Extremely rapid deployment
- Large set-based queries
- High speed load
  - Up to 2.5TB of content per hour
- Huge data scales up for data marts and enterprise data warehouse
- Model agnostic, but well tuned for Snowflake, Star, Basically, any that is generally designed for reporting
- No indices (it uses zonemaps)
- World-class workload management
- Performance tuning dramatically simplified: **simple** partitioning strategies: HASH or RANDOM

# Why Both?

*Marrying the best of each*

**IBM Netezza**

**Focused Appliance**

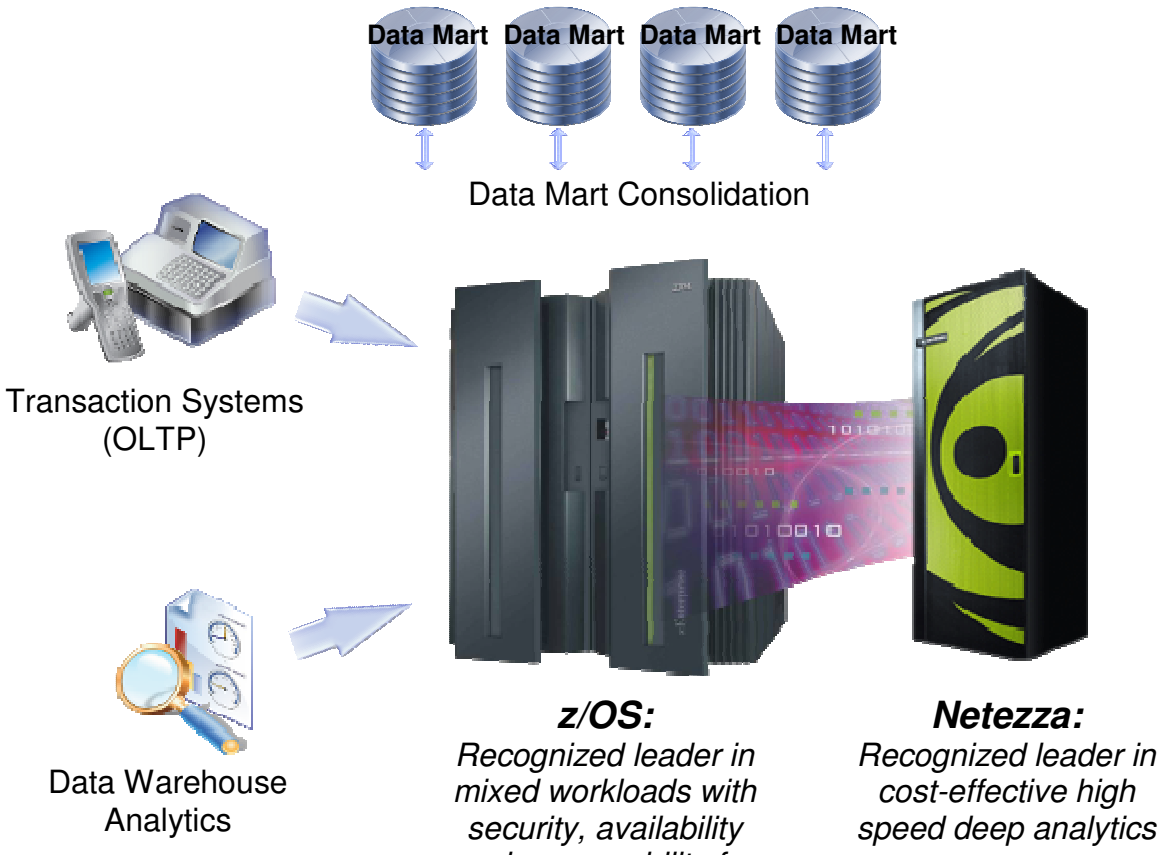
**IBM System z**

**Mixed Workload System**

*Capitalizing on the strengths of both platforms while driving to the most cost effective, centralized solution - destroying the myth that transaction and decision systems had to be on separate platforms*

**Very focused workload** ⟷ **Very diverse workload**

# The Best of All Worlds




**Best in OLTP**

*Industry recognized leader in mission critical transaction systems*

**Best in Data Warehouse**

*Proven appliance leader in high speed analytic systems*

**Best in Consolidation**

*Unprecedented mixed workload flexibility and virtualization providing the most options for cost effective consolidation*

# New value when building a decision system

Unmatched capabilities when combining the Smart Analytics System 9700 and the DB2 Analytics Accelerator

**Smart Analytics System 9700 / 9710**

+

**DB2 Analytics Accelerator**

=

***Higher Performance***

***Faster ROI***

***Immediate Value***

# IBM Blue Insight

*IBM's internal project leveraging Business Analytics and data warehousing on System z technologies to drive multi-million dollar benefits*

**Challenge:** Empower hundreds of thousands of employees via a single cost-effective BI platform

## Solution:

- IBM® Cognos® Business Intelligence
- IBM SPSS® Statistics and SPSS Modeler
- IBM InfoSphere® QualityStage and DateStage
- IBM DB2 for z/OS and IBM DB2 Analytics Accelerator
- IBM zEnterprise™ 196

## Key Benefits:

- Generates new insights that drive real business value – e.g. increasing software revenues by eight percent by enhancing small deals management.
- Delivers $25 million savings over five years through consolidation.
- Avoids approximately $250,000 in set-up costs for each new analytics project
- Scales seamlessly to meet increasing user demand

***"The business gets excellent performance and near-total availability, and can regard analytics as an always-on, real-time service."***

*— Larry Yarter, Chief Architect, Blue Insight Business Analytics Competency Center, IBM*

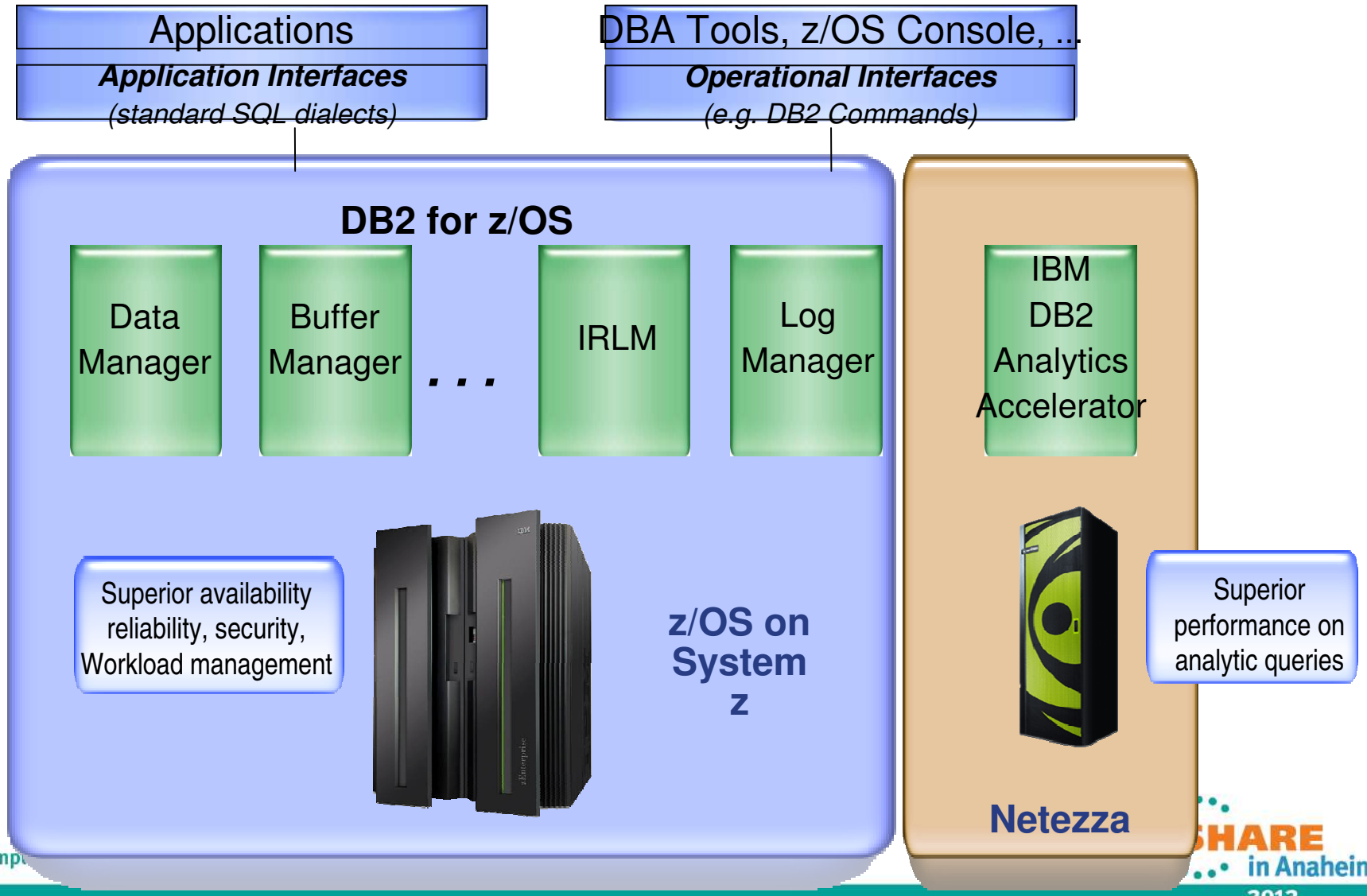
# Business Value of IBM DB2 Analytics Accelerator – Speed up your queries with low cost

- Bring industry-leading price/performance analytics to the z platform
  - No more excuse on price/performance
- Combine the best of both worlds: high volume short-running queries and heavy analytics queries in a single integrated system
  - More versatile than two separate systems for mixed workloads (1+1 > 2!)
- Reduce complexity of separate systems for data warehousing/ analytics, no separate security and data governance needed.
  - Take back control with less work
- Simple to use, reduce cost by consolidating systems
  - High utilization for both machine and human resources
- Can speed up any query that fits its characteristics no matter it's analytic or reporting
  - Any dynamic heavy queries, not necessarily analytic workloads
- New features save z storage and address archiving challenges
  - Query result consistency also (snapshot DB)

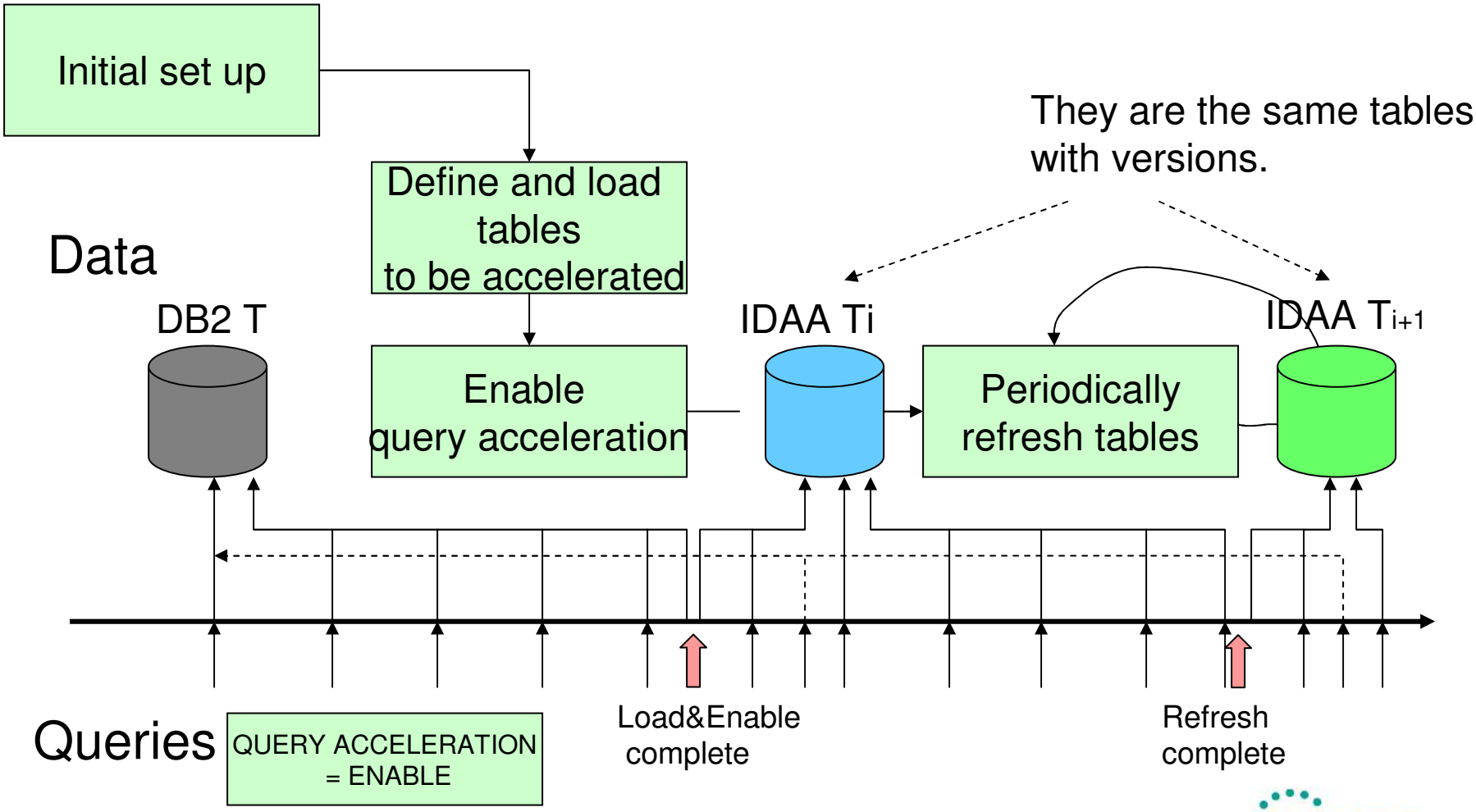
# Agenda

- IDAA design objectives
- Overall architecture and usage cycle
- Query acceleration
- Performance considerations
- Future directions

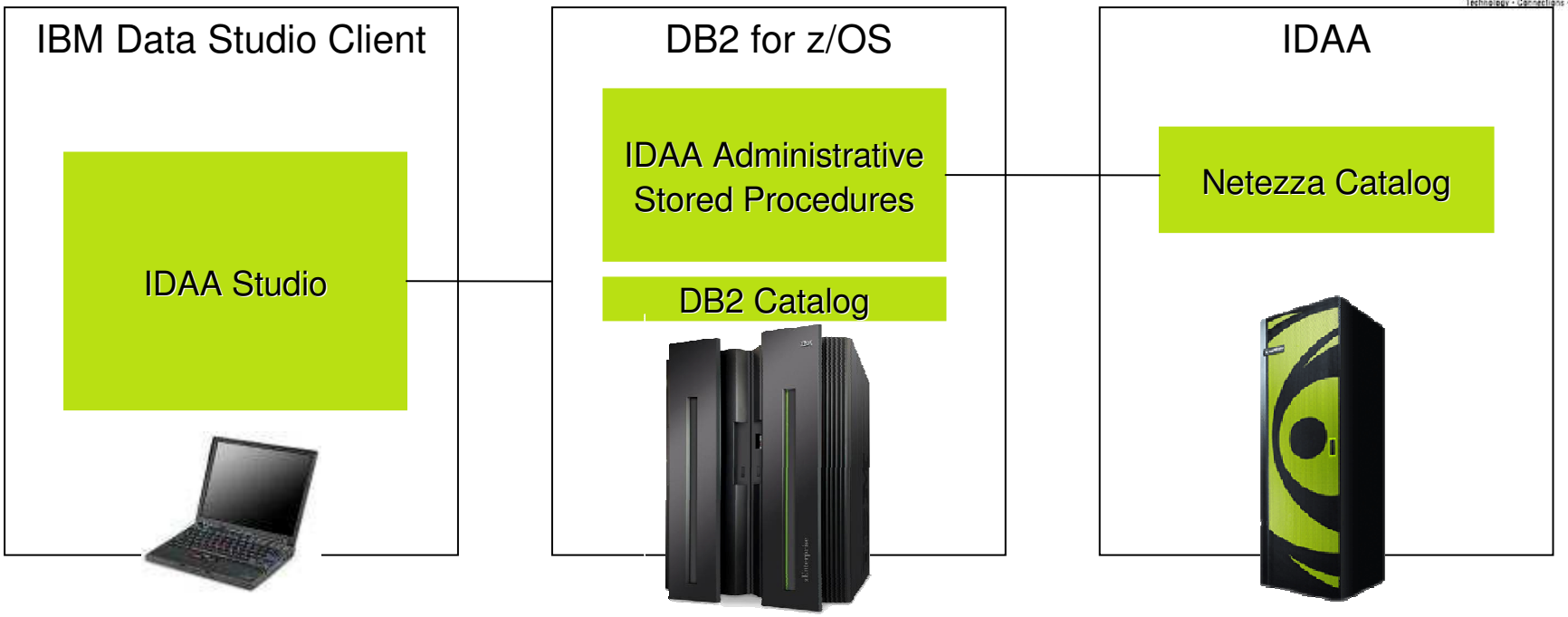
# Deep integration within DB2 for z/OS



# Usage Cycles: Deploy, Enable and Refresh

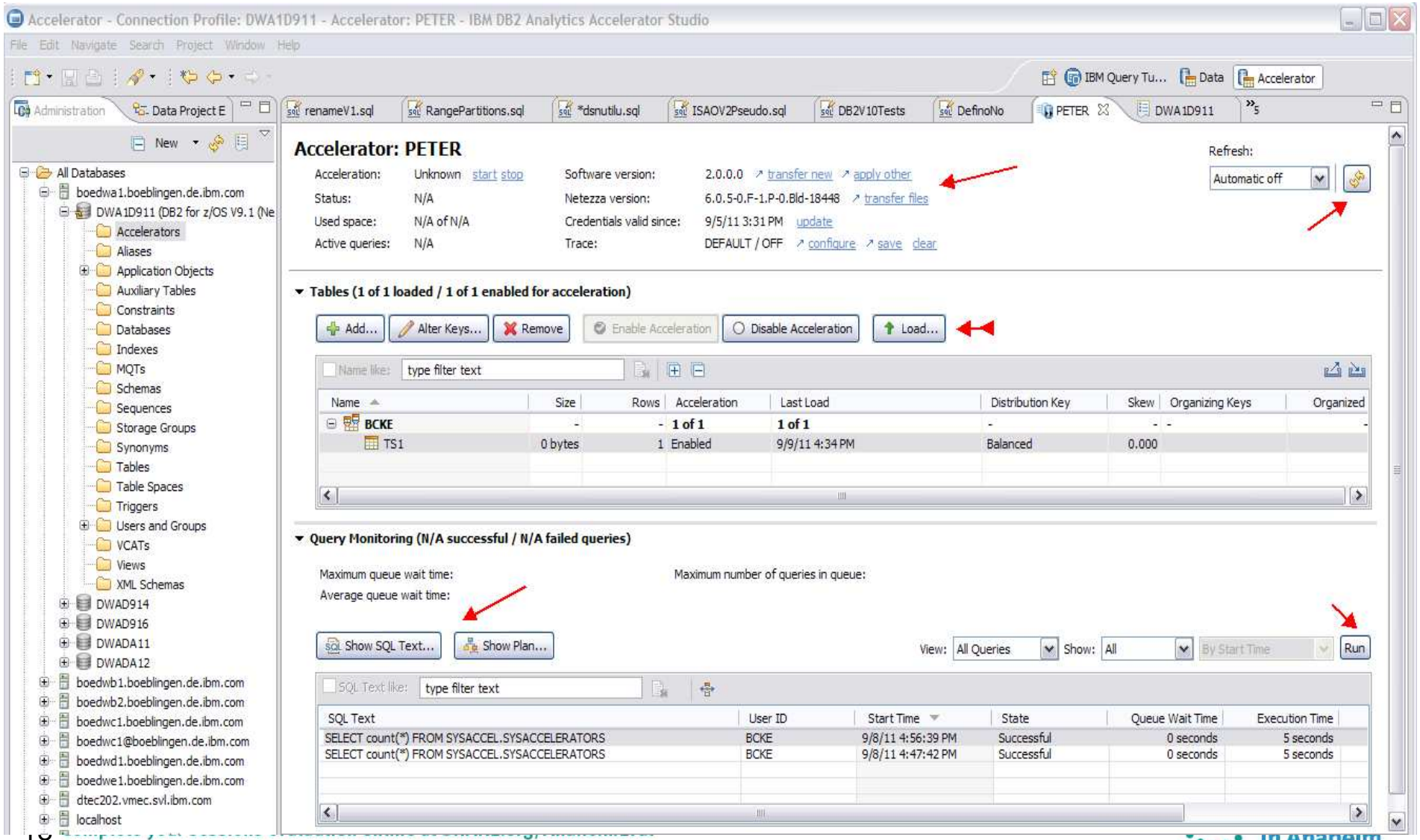


# IDAA Table Definition and Deployment




- The tables need to be defined and deployed to IDAA before data is loaded and queries sent to it for processing.
  - Definition: identifying tables for which queries need to be accelerated
  - Deployment: making tables known to DB2, i.e. storing table meta data in the DB2 and Netezza catalog.
- IDAA Studio guides you through the process of defining and deploying tables, as well as invoking other administrative tasks.
- IDAA Stored Procedures implement and execute various administrative operations such as table deployment, load and update, and serve as the primary administrative interface to IDAA from the outside world including IDAA Studio.

# IDAA Studio GUI Overview



# IDAA Admin Stored Procedures

| | |
|---|---|
| ACCEL_ADD_ACCELERATOR | Pairing an accelerator to a DB2 subsystem |
| ACCEL_TEST_CONNECTION | Check of the connectivity from DB2 procedures to the accelerator |
| ACCEL_REMOVE_ACCELERATOR | Removing an accelerator from a DB2 subsystem and cleanup resources on accelerator |
| ACCEL_UPDATE_CREDENTIALS | Renewing the credentials (authentication token) in the accelerator |
| **ACCEL_ADD_TABLES** | Add a set of tables to the accelerator |
| ACCEL_ALTER_TABLES | Alter table definitions for a set of tables on the accelerator (only distribution and organizing keys) |
| ACCEL_REMOVE_TABLES | Remove a set of tables from the accelerator |
| ACCEL_GET_TABLES_INFO | List set of tables on the accelerator together with detail information |
| **ACCEL_LOAD_TABLES** | Load data from DB2 into a set of tables on the accelerator |
| **ACCEL_SET_TABLES_ACCELERATION** | Enable or disable a set of tables for query off-loading |
| ACCEL_CONTROL_ACCELERATOR | Controlling the accelerator tracing, collecting trace and detail of the accelerator (software level etc.) |
| ACCEL_UPDATE_SOFTWARE | Update software on the accelerator (transfer versioned software packages or apply an already transferred package, new: also list software both on z/OS and accelerator side) |
| **ACCEL_GET_QUERY_DETAILS** | Retrieve statement text and query plan for a running or completed Netezza query |
| ACCEL_GET_QUERY_EXPLAIN | Generate and retrieve Netezza explain output for a query explained by DB2 |
| **ACCEL_GET_QUERIES** | Retrieve active and/or history query information from accelerator |

# Identifying workloads to accelerate

- The first question: whether the application can accept non up-to-date query results.
- The next question: whether the queries are transaction-like? That is touching a small number of rows.
  - They may not benefit from query acceleration.
- Do you have forbidden and forgotten queries?
  - These may fit well for query acceleration.
- New applications: bring it on. The potential is endless.

# Loading and Refreshing IDAA Data Contents

- Implemented as a stored procedure ACCEL_LOAD_TABLES
  - Can be invoked directly or through IDAA Studio

- A single and uniform interface for:
  - Initial load of a DB2 table into IDAA
  - Refresh of a table or any subset of table's partitions in IDAA
    - The previous table's or partition's content in IDAA is removed and replaced by the current content of the corresponding DB2 table or partition
  - Loading a subset of partitions is supported for partitioning-by-range only

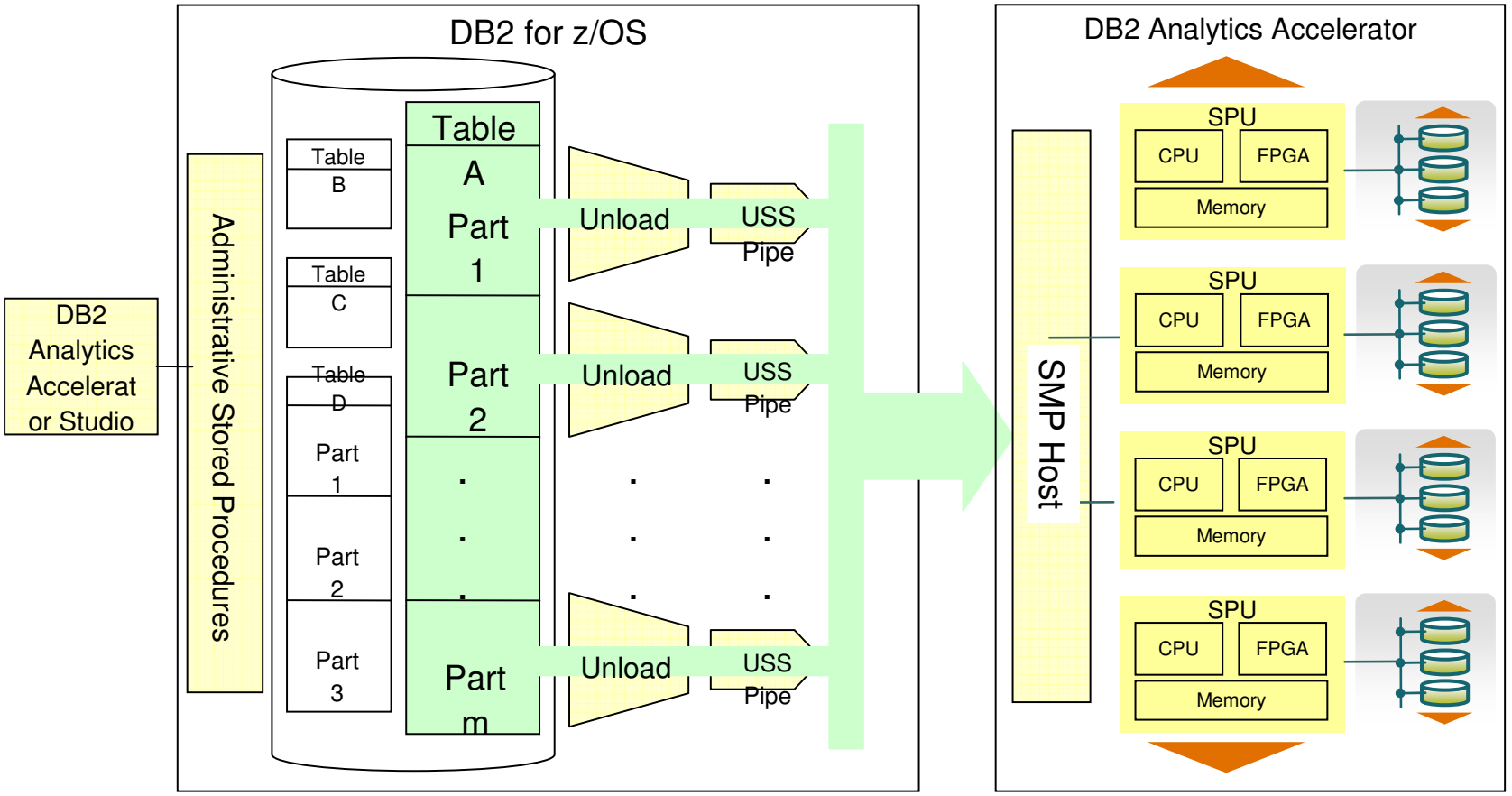
- There is no automatic checking if a table or partition needs to be refreshed due to its contents having changed, but any request to load a subset of partitions results in checking if some additional partitions need to be loaded as a result of the following DB2 operations:
  - ADD PARTITION: results in loading the added partitions
  - ROTATE PARTITION: results in refreshing the *FROM* partition
  - ALTER PARTITION RANGE: results in exception

- DB2 can route queries to IDAA while the table's or partition's content is being refreshed and they will be executed by IDAA in parallel to the refresh operation
  - LOCKMODE option for ACCEL_LOAD_TABLES
  - Provides means of either allowing or preventing data modifying operations (insert, update, delete, some utilities) during the execution of ACCEL_LOAD_TABLES, i.e. during importing the DB2 data into IDAA
  - Read-only DB2 operations are fully compatible with the IDAA importing process.
  - Possible values: TABLESET, TABLE, PARTITION, NONE

# Table Qualification and Column Projection

- A table can be accelerated if it does not have:
  - Security label column
  - A row permission (for V10)
- Some columns are left behind when other columns are copied to IDAA. A column is not copied, if it:
  - Has FIELDPROC
  - Has MBCS or DBCS ASCII or EBCDIC encoding
  - Is of type ROWID, LOB or XML, TIMESTAMP(n!=6), BINARY, CHAR FOR BIT DATA
  - Has a column mask (V10)

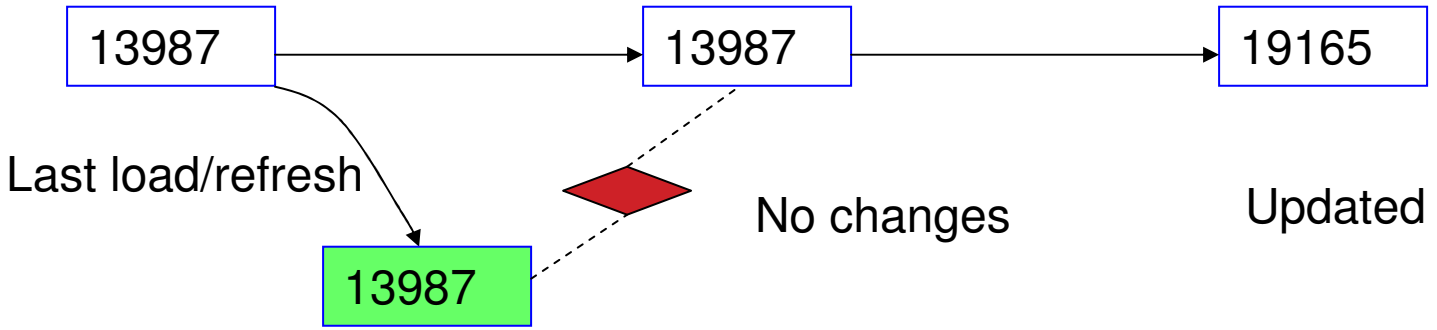
# DB2 Analytics Accelerator Content Maintenance



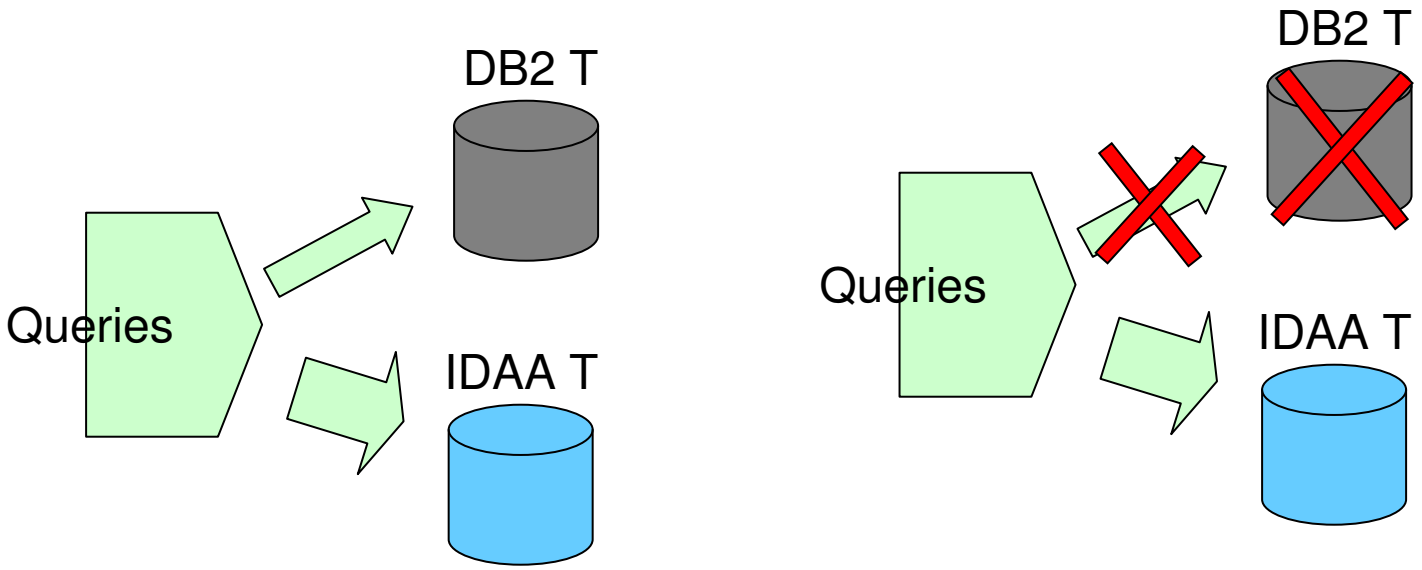

- Partitions belonging to the same table can be loaded in parallel
  - User-defined degree of parallelism
- Updates are done on a per-table or per-partition level

# Detecting updates using RTS (by users)

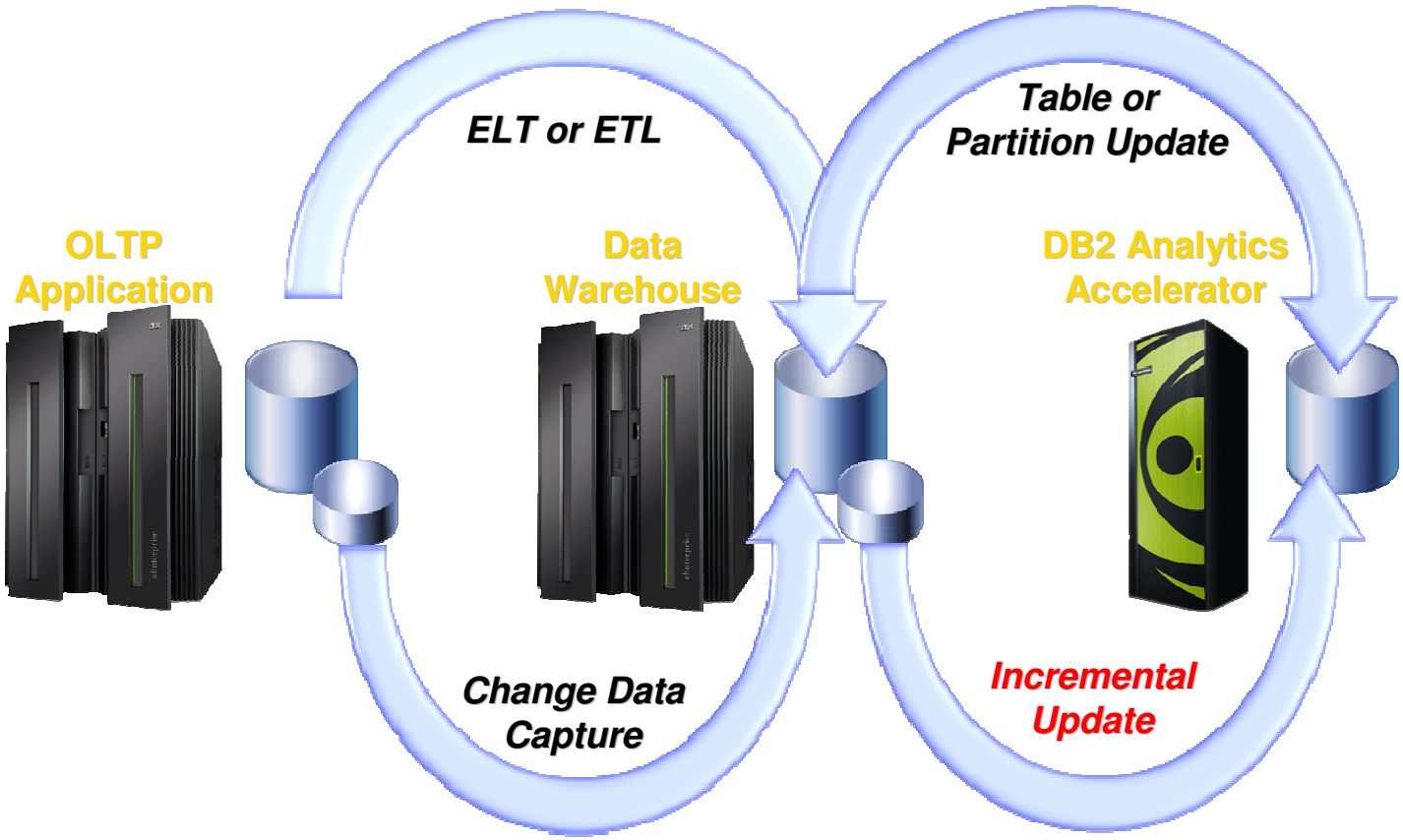
- SYSIBM.SYSTABLESPACESTATS contains the real-time stats. Your applications/tools can use some columns to keep track if there is any update since the last refresh.
- For example, COPYCHANGES keeps the sum of insert, update, delete operations, and records of LOAD since COPY was last run.



# IDAA Tables are available even if DB2 tables are not



# Introducing Incremental Update (future)




*Synchronizing data to lower data latency from days to minutes/seconds*

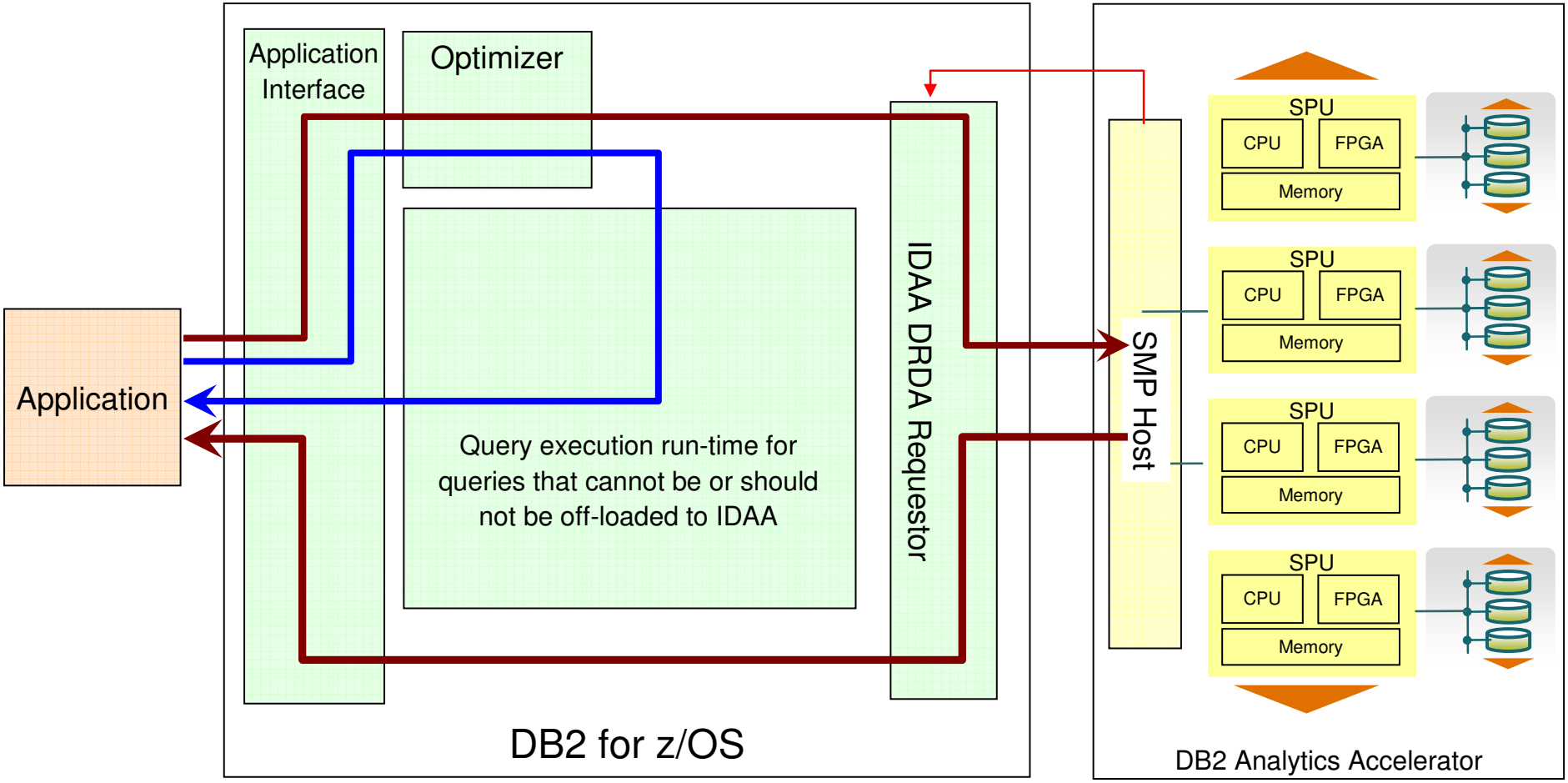
# Instrumentation Enhanced for Statistics and Accounting

```
1   LOCATION: PMOV91A                  OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R1)                    PAGE: 1-23
      GROUP: N/P                                 STATISTICS REPORT - LONG                  REQUESTED FROM: NOT SPECIFIED
     MEMBER: N/P                                                                                      TO: NOT SPECIFIED
  SUBSYSTEM: V91A                                                                          INTERVAL FROM: 09/06/11 21:49:41.35
DB2 VERSION: V9                                    SCOPE: MEMBER                                       TO: 09/06/11 23:41:50.70

---- HIGHLIGHTS ------------------------------------------------------------------------------------------------------
INTERVAL START  : 09/06/11 21:49:41.35    SAMPLING START: 09/06/11 21:49:41.35    TOTAL THREADS      :    76.00
INTERVAL END    : 09/06/11 23:41:50.70    SAMPLING END  : 09/06/11 23:41:50.70    TOTAL COMMITS      :   109.00
INTERVAL ELAPSED:       1:50:52.248034    OUTAGE ELAPSED:          1:17.097273    DATA SHARING MEMBER:      N/A

ZGRYPHON         ACCELERATOR            QUANTITY    ZGRYPHON         CONTINUED            QUANTITY
-------------------------------  ------------------  -------------------------  ------------------
CONNECTS                                    2.00    AVG QRY QUEUE LEN (3 HRS)                0.00
REQUESTS                                    9.00    AVG QRY QUEUE LEN (24 HRS)               0.00
REQUESTS TIMED OUT                          0.00    HWM QRY QUEUE LENGTH                     0.00
REQUESTS FAILED                             0.00    DATA SKEW                                0.00
BYTES SENT                               4630.00    AVG QUEUE WAIT ELAPSED TIME          0.000000
BYTES RECEIVED                         224887.00    MAX QUEUE WAIT ELAPSED TIME          0.000000
MESSAGES SENT                              27.00
MESSAGES RECEIVED                          27.00    PROCESSING CAPACITY                      0.00
BLOCKS SENT                                 0.00    PROCESSORS                               1.62
BLOCKS RECEIVED                             5.00
ROWS SENT                                   0.00    QUERY REQUESTS SUCCESSFUL                1.00
ROWS RECEIVED                               0.00    QUERY REQUESTS FAILED                    1.00
                                                      QUERY REQUESTS INVALID                   0.00
SVCS TCP/IP ELAPSED TIME                7.036035
ACCELERATOR CPU TIME                    0.000000    SHR MEM WORKER NODES    (MB)             0.00
ACCELERATOR ELAPSED TIME                0.000001      AVG IN USE            (MB)             0.00
ACCELERATOR WAIT TIME                   0.000000      MAX IN USE            (MB)             0.00

CUR ACTIVE REQUESTS                         0.00    DISK STORAGE AVAILABLE (MB)            98842.63
MAX ACTIVE REQUESTS                         0.00      IN USE                (MB)             0.81
                                                      IN USE FOR DB         (MB)             0.81
                                                    DATA SLICES                              3.25

                                                    MEM COORD AVG IN USE    (MB)             0.00
                                                    MEM WORKER AVG IN USE   (MB)             0.00
```

# Agenda

- IDAA design objectives
- Overall architecture and usage cycle
- Query acceleration
- Performance considerations
- Future directions

# Query Execution Process Flow



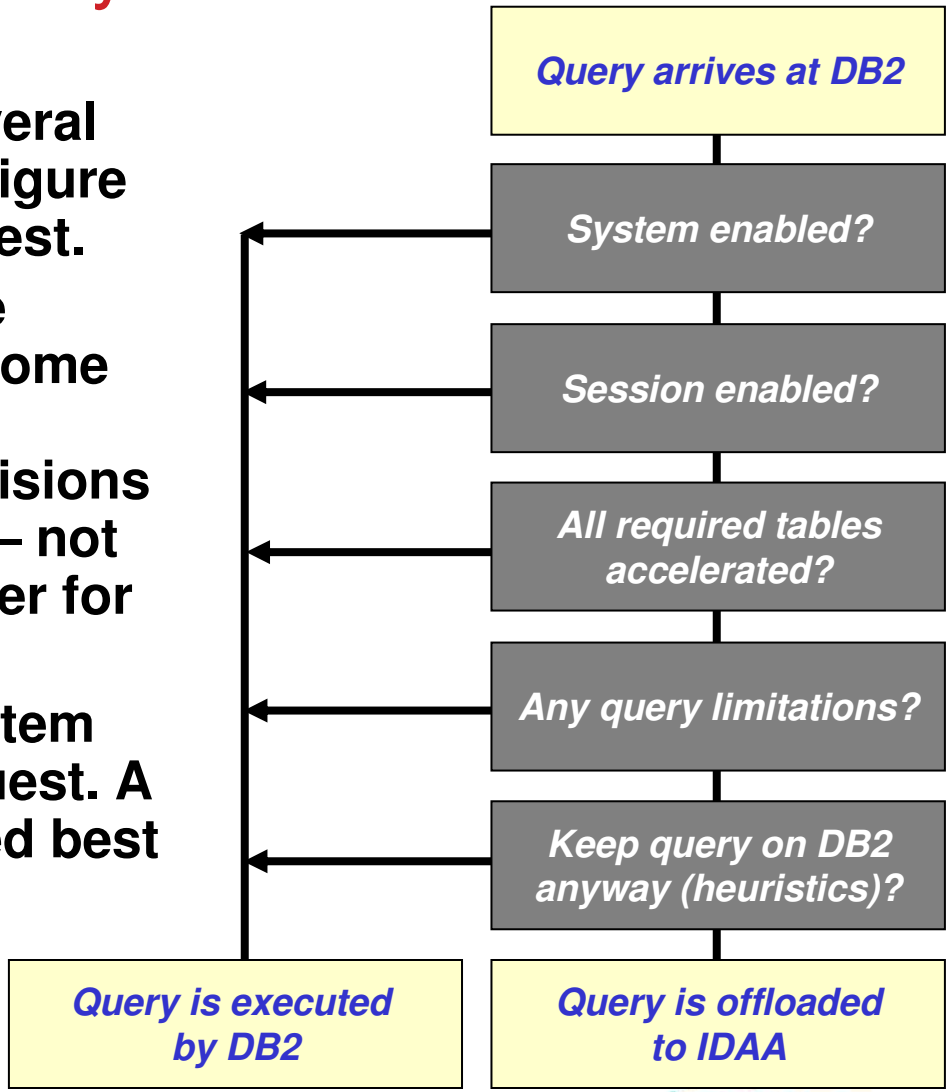

Heartbeat (DB2 Analytics Accelerator availability and performance indicators)

Queries executed without DB2 Analytics Accelerator

Queries executed with DB2 Analytics Accelerator

# DB2 decision to offload query

- When a query arrives at DB2, several checks have to be performed to figure out where to execute the query best.
- Some decisions are based on the capabilities of the accelerator – some query constructs might prevent accelerated execution. Other decisions are based on customer decision – not all data is enabled by the customer for acceleration.
- The query is executed on the system that is better suitable for the request. A transaction-like query is answered best by DB2, data intensive analytical queries make best use of the accelerator.



# Query Acceleration Criteria

A query can be routed to IDAA if:

- DB2 in V9 NFM, V10 CM9 or V10 NMF.
- The entire query can be accelerated, i.e. the unit of acceleration is a whole query
  - The whole query will either run in DB2 or in the accelerator
  - All tables referenced to in the query must be enabled for acceleration
- The query is dynamic
- The query is defined as read-only
- The query is a SELECT statement
- The associated cursor is not defined as a scrollable or a rowset cursor
- The DRDA protocol access is in effect for a remote workload (no private protocol)
- The query is from a package (not plan DBRM)
- Routing to IDAA is considered more efficient for performance than to execute the query in DB2 mainline

# Routing Control Knobs

- System parameters
  - ACCEL: Possible values: NO, AUTO, COMMAND
  - ACCEL_LEVEL: Valid in DB2 9 only
    - *Possible values: V1 (default) and V2*
    - *Set it to V2*
  - QUERY_ACCELERATION
    - *Sets the initial value for the CURRENT QUERY ACCELERATION special register*
    - *Possible values: NONE (default), ENABLE and ENABLE WITH FAILBACK*
- Special register CURRENT QUERY ACCELERATION
  - Can be set implicitly by inheriting the value of the system parameter, or
  - Explicitly by SET CURRENT QUERY ACCELERATION

| Value | Description |
|---|---|
| NONE | No query is routed to the accelerator |
| ENABLE | A query is routed to the accelerator if it satisfies the acceleration criteria.<br>If there is an accelerator failure while running the query, or the accelerator returns an error, DB2 will return a negative SQL code to the application. |
| ENABLE WITH FAILBACK | A query is routed to the accelerator if it satisfies the acceleration criteria.<br>Under certain conditions the query will run on DB2 after it fails in the accelerator. In particular, any negative SQLCODE will cause a failback to DB2 during PREPARE or first OPEN. No failback is possible after a successful OPEN of a query. |

# Query Routing Heuristics

- You get best of both worlds for mixed workloads (OLTP/OLAP)
- DB2 Optimizer uses a set of rules to determine whether a given query is better off being executed in DB2 core engine or routed to the accelerator, such as:
  - typical OLTP access path patterns run in DB2, e,g.
    - *Equal unique or near equal unique access*
    - *One fetch access*
  - typical OLAP/warehousing access patterns run in accelerator
    - *If none of these: WHERE, GROUP BY, ORDER BY, aggregate functions is specified (i.e. all rows are to be returned), the query is not routed*
  - If all the tables referred in the query are “small”, the query is not routed.
    - *ACCEL_TABLE_THRESHOLD is specified by the DB2 Profile_table mechanism. The default value is 1 million rows total*
  - If the estimated cost is lower than the ACCEL_TOTALCOST_THRESHOLD, the query will not be offloaded. The default value is 5,000 .
  - If a huge result set is expected, the query is not routed
    - *ACCEL_RESULTSIZE_THRESHOLD is specified by the DB2 Profile_table mechanism. By default this check is skipped (-1)*
- Recommendation: use the default values. If you have to, change them only after rigorous testing!

# PROFILE_TABLE Adjusting Parameters for Heuristics

| KEYWORD in DSN_PROFILE_ATTRIBUTES table | Possible values | Default setting | Remark |
|---|---|---|---|
| ACCEL_TABLE_THRESHOLD | Positive integer OR -1 | 1,000,000 | Threshold for the total table cardinality of the entire query. -1 means small table checking is disabled. |
| ACCEL_TOTALCOST_THRESHOLD | Positive integer or -1 | 5,000 | Threshold for the total estimated cost of the entire query to run in DB2. -1 means this checking is disabled. |
| ACCEL_RESULTSIZE_THRESHOLD | Positive integer or -1 | -1 | Threshold for the maximum number of thousand rows for the estimated result size for the query to be offloaded. -1 means this checking is disabled. |

## PROFILE_TABLE Adjusting Parameters for Heuristics (continued)

- Create the profile tables. See sample DSNTIJOS (V9)/DSNTIJSG(V10) of the SDSNSAMP library
  - SYSIBM.DSN_PROFILE_TABLE
  - SYSIBM.DSN_PROFILE_HISTORY
  - SYSIBM.DSN_PROFILE_ATTRIBUTES
  - SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY

- Insert rows into SYSIBM.DSN_PROFILE_TABLE to create a profile. The PROFILEID column identifies the profile and matches rows in the SYSIBM.DSN_PROFILE and DSN_PROFILE_ATTRIBUTES tables.
- Insert rows into SYSIBM.DSN_PROFILE_ATTRIBUTES table to define the attributes, using PROFILEID, KEYWORD, and ATTRIBUTEn columns.
- Issue command: START PROFILE
  to start or reload the PROFILE tables
- To disable a particular profile, delete the row from DSN_PROFILE_TABLE, or change the PROFILE_ENABLED column to value N. And START PROFILE to refresh.
- To disable all profiles, issue command STOP PROFILE.

# PROFILE_TABLE Adjusting Parameters Example

- DSN_PROFILE_TABLE

| AUTHID | PLAN NAME | COLLID | PKG NAME | IPADDR / LOCATION | PROFILEID | PROFILE_ENABLED |
|---|---|---|---|---|---|---|
| null | null | null | null | null | 1 | Y |
| TOM | null | null | null | 9.10.128.3 | 2 | Y |

| PROFILEID | KEYWORDS | ATTRIBUTE1 | ATTRIBUTE2 | ATTRIBUTE3 | ATTRIBUTE TIMESTAMP |
|---|---|---|---|---|---|
| 1 | ACCEL_TABLE_THRESHOLD | | 500000 | | 2011-06-24… |
| 2 | ACCEL_TOTALCOST_THRESHOLD | | 10000 | | 2011-06-25… |

# EXPLAIN

- DB2 EXPLAIN function is enhanced to provide basic information about accelerator usage
  - Whether query qualifies for acceleration and, if not, why
  - The access path details associated with the query execution by Netezza are provided independently of DB2 EXPLAIN by the IDAA Studio.
- For each query (irrespective of the number of query blocks) a row is inserted in the following tables:
  - in both PLAN_TABLE and DSN_QUERYINFO_TABLE, if the query is accelerated
    - PLAN_TABLE's ACCESSTYPE column is set to a value of 'A'
    - DSN_QUERYINFO_TABLE's QI_DATA column shows the converted query text
  - in DSN_QUERYINFO_TABLE only, if the query is not qualified
    - REASON_CODE and QI_DATA columns provide details
- Note that the EXPLAIN tables can be populated with above described information even if there is no accelerator connected to DB2
  - Specifying EXPLAINONLY on START ACCEL command does not establish any communications with an actual accelerator, but enables DB2 to consider its presence in EXPLAIN.

# EXPLAIN output for offloaded queries

PLAN_TABLE

| Column name | Data type | Brief Description |
|---|---|---|
| … | | |
| ACCESSTYPE | CHAR(2) NOT NULL | The method of accessing the new table:<br>A Query is accelerated |
| … | | |

DSN_QUERYINFO_TABLE: one row with offloading info

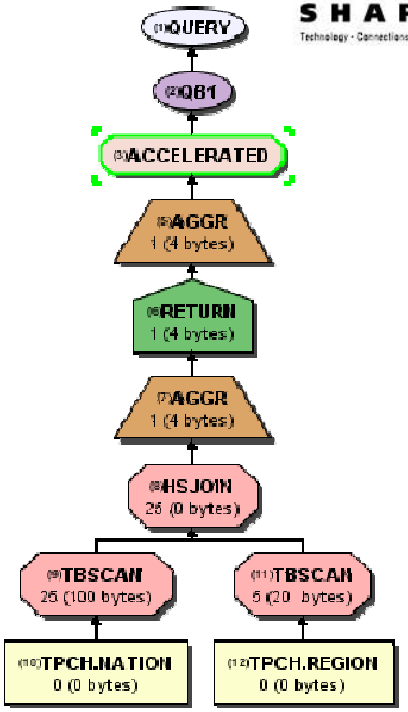
| Column name | Data type | Brief Description |
|---|---|---|
| … | | |
| QINAME1 | VARCHAR(128) NOT NULL WITH DEFAULT | When TYPE='A'<br>ISAOV1: The creator of the accelerated query table.<br>**ISAO V2: When REASON_CODE = 0, this row contains the accelerator name which the query will be offloaded to.** |
| QINAME2 | VARCHAR(128) NOT NULL WITH DEFAULT | When TYPE='A'<br>ISAOV1: The name of the accelerated query table.<br>**ISAO V2: When REASON_CODE = 0, this row contains the location name which the query will be offloaded to.** |
| … | | |
| TYPE | CHAR(8) NOT NULL WITH DEFAULT | The type of the output for this row:<br>'A' This row is for a query that DB2 attempts to run on an accelerator server |
| REASON_ CODE | SMALLINT NOT NULL WITH DEFAULT | The reason code for this row: When TYPE='A'<br>0 The query successfully qualifies for offloading<br>**with QINAME1 is accelerator name and QINAME2 is location name** |
| … | | |
| QI_DATA | CLOB(2M) NOT NULL WITH DEFAULT | **When TYPE='A' and REASON_CODE = 0, this row contains converted offloading query text.** |
| … | | |

# EXPLAIN reason code for an unoffloaded query

| REASON_CODE | Description |
|---|---|
| 0 | Query qualifies for acceleration. |
| 1 | No active accelerator was found when EXPLAIN was executed. |
| 2 | The special register CURRENT QUERY ACCELERATION is set to NONE. |
| 3 | The query is a DB2 short running query or re-routing to the accelerator is not considered advantageous. |
| 4 | The query is not read-only. |
| 5 | The query is running under the private protocol. |
| 6 | The cursor is defined as scrollable or rowset cursor. |
| 7 | The query refers to multiple encoding schemes. |
| 8 | The query FROM clause specifies a data-change-table-reference. |
| 9 | The query contains a correlated table expression. |
| 10 | The query contains a recursive common table expression reference. |
| 11 | The query contains an unsupported expression. QI_DATA contains the expression text. |
| 12 | The query references table *table-name* that is either not defined in accelerator, or the table is defined, but is not enabled for query re-routing. |
| 13 | The accelerator *accelerator-name* containing the tables of the query is not started. |
| 14 | The column *column-name* referenced in the query is altered in DB2 after the data is loaded in the accelerator. |
| 15 | The query references a DB2 10 new SQL feature. |
| 16 | The query is not from a package. |
| 900 through 999 | IBM Internal use. |

# Netezza EXPLAIN integration

- The Accelerator GUI is able to receive the Netezza Plan files for query executions that happened on the accelerator side. These files are parsed and embedded into DB2 Visual Explain.
- Distribution and Organizing keys can be altered “on the fly” based on the Explain output. The accelerator is redistributing table data in the background.




| Name | Size | Rows | Acceleration | Last Load | Distribution Key |
|---|---|---|---|---|---|
| **TPCH** | - | - | **7 of 8** | **7 of 8** | - |
| CUSTOMER | - | - | Enabled | 7/5/11 9:37 AM | Balanced |
| LINEITEM | - | - | Enabled | 7/5/11 9:37 AM | L_ORDERKEY |
| NATION | - | - | Enabled | 7/8/11 3:14 PM | N_REGIONKEY |
| ORDERS | - | - | Enabled | 7/5/11 9:37 AM | O_ORDERKEY |
| PART | - | - | Enabled | 7/5/11 9:37 AM | Balanced |
| PARTSUPP | - | - | Enabled | 7/5/11 9:37 AM | Balanced |
| REGION | - | - | Enabled | 7/5/11 9:37 AM | R_REGIONKEY |

# Ways to increase query acceleration chances

- De-correlate a correlated subquery/scalar fullselect into non-correlated.
- Substr(mixed_col, ..): use CAST( mixed-col as VARCHAR(n) FOR SBCS DATA) if it contains single byte data.
- Avoid V10 implicit cast feature: numeric_col = ‘123’ => numeric_col = 123
- Avoid multi-row fetch (ROWSET cursor)
- Re-load after DB2 ALTER TABLE on column so the accelerator contains consistent data and types

# Example: DSNTIAUL query acceleration

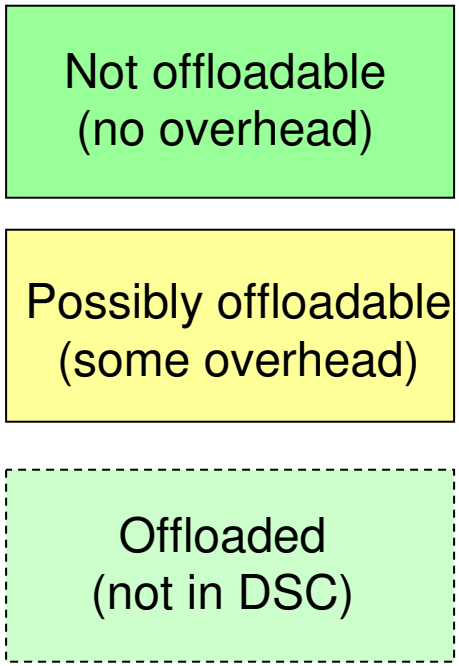
- Rebind the shipped DSNTIAUL plan into a package (V9 only)
  - Take care of GRANT also
- Turn off the default multi-row fetch by using PARMS('SQL,1')
- Use dynamic SQL access local tables or
- CONNECT TO remote server with IDAA, no 3-part names (private protocol)
- SET CURRENT QUERY ACCELERATION = ENABLE or ENABLE WITH FAILBACK

# LOAD from CURSOR with Query Acceleration

- INSERT w/ SELECT does not get accelerated yet
- DB2 10 workaround: use LOAD from CURSOR
- SET CURRENT QUERY ACCELERATION = ENABLE
- Define the query, which is to be accelerated.
- LOAD into a DB2 table

# Dynamic Statement Cache (DSC) Impact

- Dynamic statements that are not qualified for offloading will not be impacted by the offloading logic.
  - Including queries referencing a table not enabled for acceleration.
  - Once a table is added for acceleration, use RUNSTATS … REPORT NO UPDATE NONE to invalidate the dynamic statement cache entries on that table.
- Possibly offloadable queries are checked every time before searching DSC
- Dynamic statements that are accelerated are not cached in the dynamic statement cache.
  - EXPLAIN STMTCACHE ALL will not contain accelerated queries.



# Workload Self-assessment without an accelerator (1/2)

- **Step1: Setup**
  - Set system parameter ACCEL as AUTO or COMMAND
  - DB2 9 only: set system parm ACCEL_LEVEL set to a value of *V2*
  - Create pseudo catalog table `"SYSACCEL"."SYSACCELERATORS` and `"SYSACCEL"."SYSACCELERATEDTABLES"`
  - Create explain tables

- **Step2: Generate a virtual accelerator**
  - `INSERT INTO "SYSACCEL"."SYSACCELERATORS" VALUES('SYSACCEL',NULL);`

- **Step3: Populate SYSACCELERATEDTABLES with intended tables**
  - `INSERT INTO "SYSACCEL"."SYSACCELERATEDTABLES" (NAME,CREATOR,ACCELERATORNAME,REMOTENAME,REMOTECREATOR, ENABLE,CREATEDBY,SUPPORTLEVEL) SELECT NAME,CREATOR,'SYSACCEL',NAME,CREATOR,'Y',CREATEDBY, 1 FROM SYSIBM.SYSTABLES WHERE TYPE='T' AND DBNAME= <database name>;` -- or your own criteria

## Workload Self-assessment without an accelerator (2/2)

- **Step4: Start the virtual accelerator with EXPLAINONLY**
  - **Issue command:**
    `-START ACCEL(*) ACCESS(EXPLAINONLY)`

- **Step5: Explain queries**
  - `SET CURRENT QUERY ACCELERATION = ENABLE` or Set system parameter QUERY_ACCELERATION = ENABLE
  - Execute EXPLAIN statement.
  - Check whether a query is eligible for offloading – you can COUNT it
    ```
    SELECT QUERYNO
    FROM SYSADM.PLAN_TABLE WHERE  ACCESSTYPE = 'A';
    ```
  - To see why a query is not eligible for offloading:
    ```
    SELECT QUERYNO, REASON_CODE, QI_DATA
    FROM SYSADM.DSN_QUERYINFO_TABLE;
    ```

  - Or explore V10 new functionality `CURRENT EXPLAIN MODE = 'EXPLAIN'` and execute a normal SQL statement.

# Latest Enhancements

- Cancel thread (DB2 enhancement for distributed)
- EDITPROC support (DB2 encryption)
- EBCDIC MBCS, DBCS support (converted to UFT-8 in Netezza Accelerator)
- Large result spooling in IDAA accelerator
- UNLOAD light exploitation phase 1
- Multi-rack accelerator support
- Allow tables enabled on multiple accelerators

# Agenda

- IDAA design objectives
- Overall architecture and usage cycle
- Query acceleration
- Performance considerations
- Future directions

# Performance Test System Configurations

- z/OS LPAR set up
  - z/OS Level:  01.12.00
  - CPU: 6 z196 processors by default.
  - Storage: 94GB
  - DASD: DS8800
- Netezza TF-12
  - CPU Cores: 96
  - Cache (8G per SPU): 96GB
  - User Data: 32 TB
  - S-Blades: 12

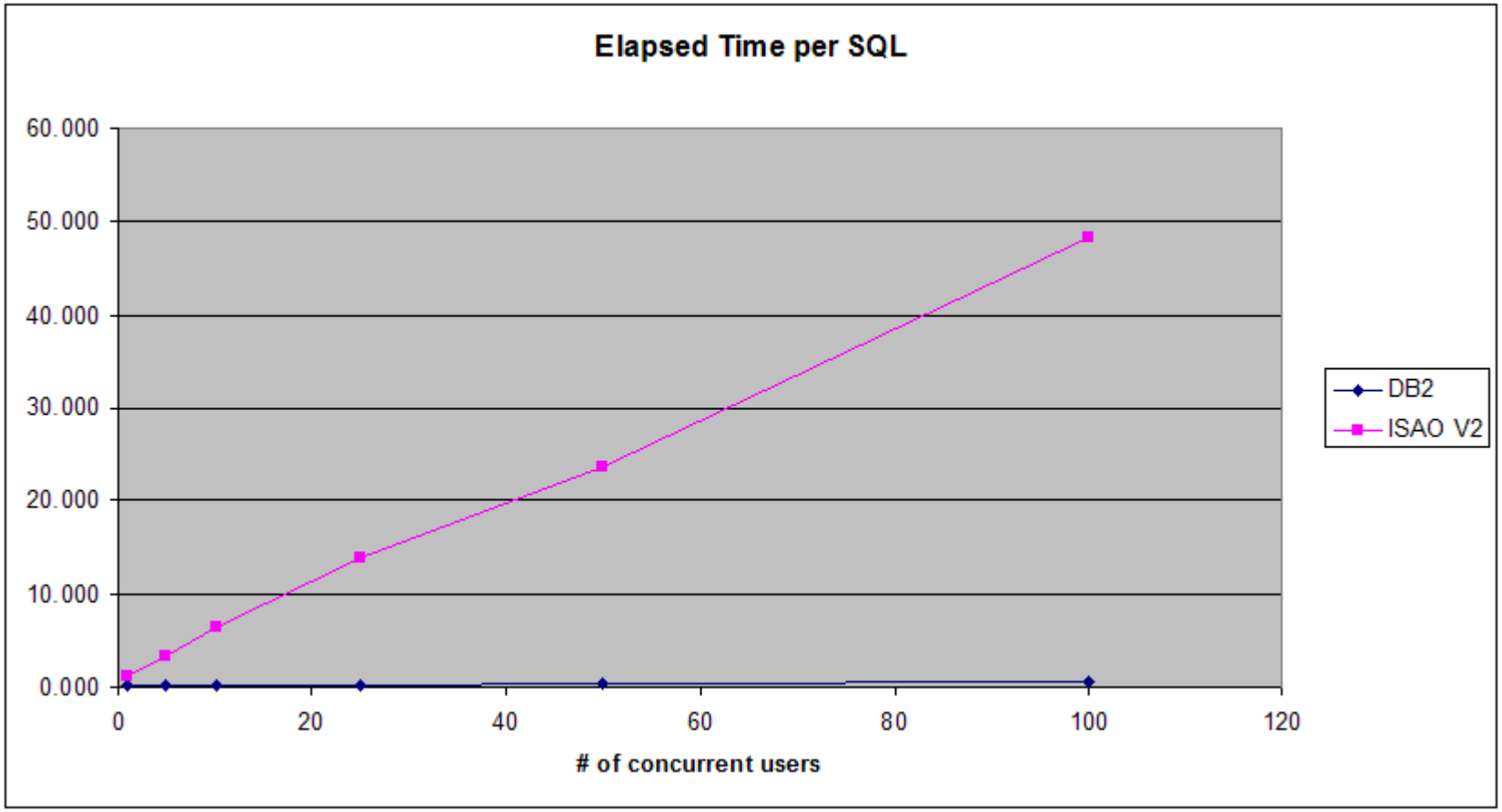
# Summary of Performance

| Workload | TPCH30GB (30parts) | TPCH 300GB (30parts) | TPCH 1TB (100parts) | TPCH 5TB (250parts) | TPCH 10TB (1000parts) |
|---|---|---|---|---|---|
| Date | 9/14/11 | 9/14/11 | 9/20/11 | 9/20/11 | 9/28/11 |
| # of SQL | 22 | 22 | 22 | 22 | 22 |
| # of offloaded | 21/22 | 21/22 | 21/22 | 21/22 | 21/22 |
| % offloaded | 95% | 95% | 95% | 95% | 95% |
| Min Speed up (DB2 ET / IDAA ET) | 0.75 | 1.9x | NA* | NA* | NA* |
| Max Speed up | 53x | 191x | NA* | NA* | NA* |
| Avg Speed up | 17x | 34x | NA* | NA* | NA* |
| | | | | | |
| Load rate | 808GB/hr | 1057GB/hr | 1085GB/hr | 1115GB/hr | 1103GB/hr |
| | | | | | |
| Comments | •DB2 in parallelism<br>•z196 6 CPs<br>•12 unload job | •DB2 in parallelism<br>•z196 6 CPs<br>•12 unload job | •DB2 in parallelism<br>•z196 6 CPs<br>•12 unload job | •DB2 in parellelism<br>•z196 6 CPs<br>•12 unload job | •DB2 in parallel ism<br>•z196 6 CPs<br>•12 unload job |

NA*: Due to hardware resource constraint, cannot perform DB2 reference measurements for TPCH 1TB, 5TB and 10TB queries. As a result, there is no speedup factor provided.

# Concurrent Short Running Query Offload

- 20 queries running via multiple virtual users. Zero think time between queries.
- With only one user, the average elapsed time per SQL is 200 milliseconds on DB2 and 1200 milliseconds on IDAA.




- Hand selected short running SQL against the 30GB TPC-H schema
- IDAA versus DB2 on a 6 way z196
- Verified linear scaling of elapsed time for concurrent SQL on IDAA

# IDAA Performance Considerations

- Query acceleration
  - Consider trade-offs when determining which workload/queries to offload. Speed up factor and CPU savings will need to be weighed against query volume for maximum throughput.
  - Keep DB2 table and index statistics up-to-date so that DB2 could make optimal IDAA offloading decisions.
  - Do not turn on QUERY ACCELERATION for applications not intended for offloading to avoid any overhead.
  - Watch for queries that return large result sets and push down data aggregation into SQL for acceleration if applicable

- Load data to Netezza
  - Tune AQT_MAX_UNLOAD_IN_PARALLEL WLM environment variable for the IDAA load stored procedure and weigh the available the system CPU resources and number of optimal concurrent active threads (recommended maximal 10 threads) on Netezza for optimal load performance.
  - Specify appropriate distribution and organizing keys before loading the tables into Netezza from IDAA client, considering both even distribution and co-located joins.

# Agenda

- IDAA design objectives
- Overall architecture and usage cycle
- Query acceleration
- Performance considerations
- Future directions

# IDAA Future Directions (1/2)

- Incremental update
- More QUERY ACCELERATION control: ELIGIBLE, ALL
- INSERT with SELECT support: SELECT can be accelerated.

- DB2 Accelerator Partitions: Online storage/archiving support - Accelerator-only data
- Enlarge query acceleration scope
  - SUBSTR etc. Use Netezza character-based for DB2 byte-based semantics, and CODEUNITS32 & CODEUNITS16 support
  - OLAP functions: moving average etc.
- Detect table update in DB2 based on RTS (real-time stats)
- WLM connection & query prioritization
  - Workload isolation: production v.s. testing
  - Mapping z/OS WLM to NZ WLM

# IDAA Future Directions (2/2)

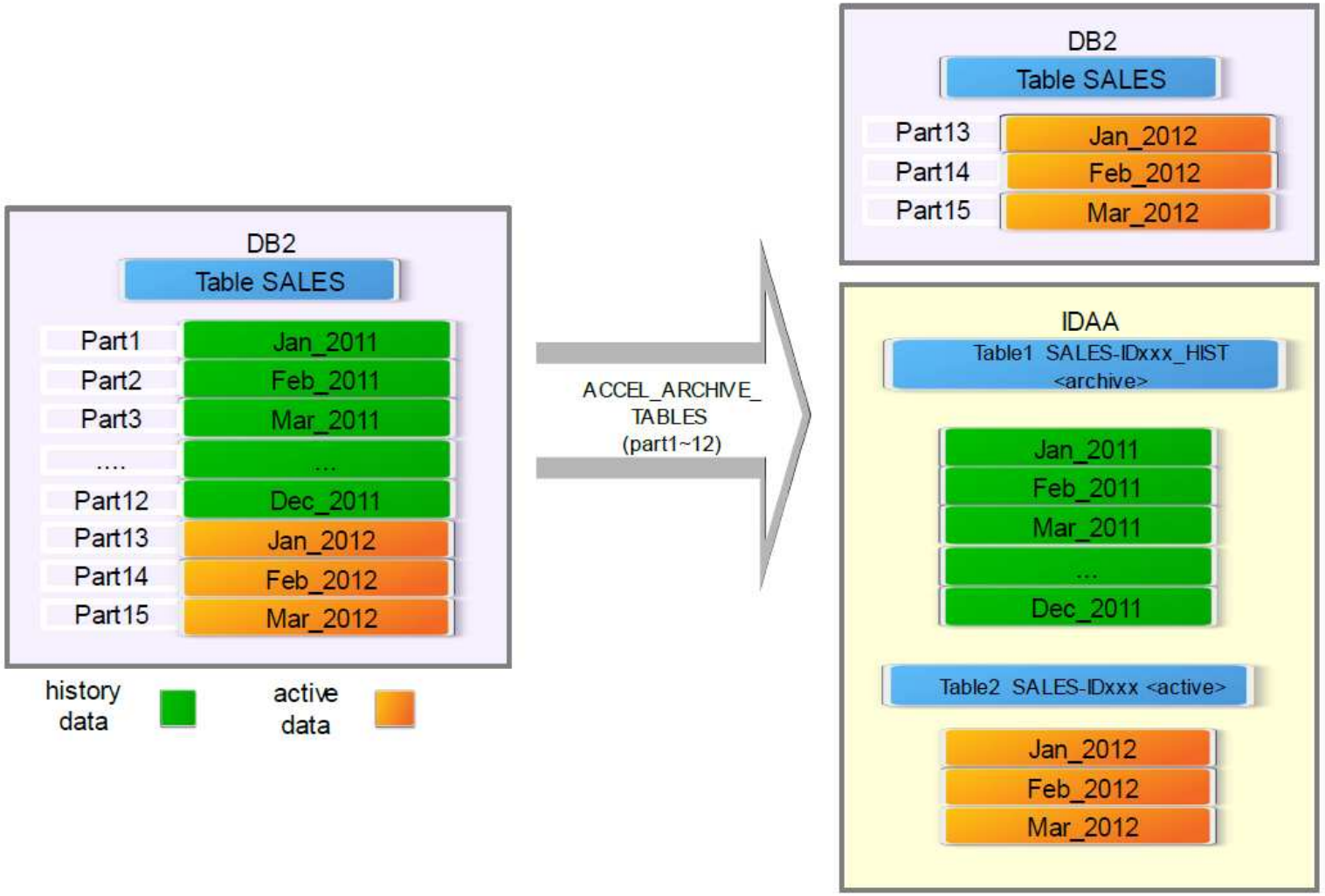
- Data currency: enhancements on asynchronous propagation
- Unload light exploitation phase 2
- Static SQL support
- Expose more analytics functions:
  - SPSS model building, scoring
  - SAS batch scoring
  - Hadoop/MapReduce
  - Open source R

# DB2 Accelerator Partitions/Online Archiving (1/2)

- Data in a partitioned table is divided into two parts:
  - Hot data: currently active data
  - Warm/cold data: read-only when needed
- Save z storage for non-active data by storing them on the accelerator only (purged from DB2 table spaces), queryable through accelerator
- Queries against active data by default, can be accelerated.
- By SET CURRENT GET_ACCEL_ARCHIVE = YES, queries will include the accelerator-archived data, and be executed in accelerators only

- For Sequoia, will recognize native archive tables to move to IDAA.

# DB2 Accelerator Partitions/Online Archiving (2/2)



# Summary

- IDAA opens unprecedented opportunities to analyze massive data in DB2 for z/OS for competitive advantages
- IDAA is simple to use and very effective
- IDAA brings the competitive price/performance to System z for analytics workloads
- We've covered main steps in exploiting IDAA:
  - Identify workloads
  - Define and load data, refresh data
  - Enable query acceleration
  - Monitor and tune the performance – like an appliance