Data Virtualization for Providing Data and its Application on z/OS

Tim Willging
Distinguished Engineer
Rocket Software
What is Data Virtualization?

Mainframe resident data virtualization solution for real time, universal access to data, regardless of location or format.
Data Virtualization Use Cases

- Makes data more accessible
  - Lowers barriers for customers to access for many purposes
- Provides modern access to legacy data
  - SQL access to data…in place
- Allows data to be integrated easily into other tools
  - JDBC access to virtualized data
  - Integrate into data movement tools
  - BI Tools
  - SQL Editors
- Useful for:
  - Dashboards
  - Analytics (BI Tools)
  - Websites
  - Data Consolidation
  - Mobile applications
Data Virtualization Components

- **Data Access Layer**
  - Provides access to supported data sources
  - Specialized access to mainframe data sources
    - VSAM, Flat Files, IMS, ADABAS, SMF, etc
  - Link to relational data sources

- **Data Catalog**
  - Repository of data sources
  - Information required to access each data source
  - Virtual Tables/Virtual Views

- **SQL Subsystem / Engine**
  - Processes standard SQL
  - Redirects SQL to underlying data sources
  - Calls Data Catalog and Data Access Layer
  - Ability to join data from different data sources
  - Returns result set in standard SQLDA format
How We Optimize Mainframe TCO

- Mainframes have multiple processors
  - DV has patented technology that allows it to run 99% of its own processing in the zIIP engine

Eligible Workloads Can Run Outside of GPP within zIIP
Architecture built to leverage zIIP
Extract, Virtualize, and Load
Optimizations for Data Virtualization

- zEDC
- SMT2
- SIMD
- Metal C optimized Server specific to z System version
- Pageable Large Frames
- SMC-R
- zIIP
- Parallelism – Intra-SQL and Intra-Partition
- Shared Memory Object based JNI interface
How does it work?
Virtualization / Mapping Process

- Specify location and name of “copy book”
- Identify data source attributes
  - Data Set Name (VSAM)
  - IMS DBD, PSB
  - ADABAS Attributes
  - Relation Data Source (DRDA attributes, Table name, View name)
- If not relational – You need to create a virtual table:
  - Enter name for “Virtual Table”
  - Fields of copy book end up being column names
  - Virtualization provides SQL access to data source where it resides
- Graphical interface provided for mapping process
- Saved in metadata repository
Embedded in Three Different Tools

- IBM DB2 Analytics Accelerator Loader V2.1
- IBM QMF 11.2.1 Data Service
- IBM Spark on z/OS
DB2 Analytics Accelerator Loader for z/OS

Extend analytic capabilities by bringing non-DB2 data to IBM Analytics Accelerator and z Systems

• In-memory transformation (ETL) of non-DB2 data. Significant cost and time reduction by reducing manual processes.

• Insight into more data types such as IMS, VSAM, sequential files, Oracle, Adabas, SMF, etc.

• Assists in data availability by loading to multiple accelerators in parallel

• Direct load of Accelerator-only tables (AOTs) for greater savings

• Load Resume process adds data – avoid reloading entire table

• Support for replicated tables
IBM DB2 Analytics Accelerator Loader V2.1

- Extends capability of Accelerator beyond DB2 data
- Addresses challenges identified in loading non DB2 data
  - Manual – Labor intensive Extract and Transform
  - Slow – due to staging data to disk

**Additional Features**

- Embedded capability to “virtualize” and load many types of data to accelerator
- Views to load 100s of SMF
- Ability to load Syslog data
- Load Resume
- Historical data load

![Data Load Types](image)
DB2 Analytics Accelerator Loader

Automates loading of non-DB2 data

Entire process is automated in one job:

- Creates DB2 table
- Table added to Accelerator
- Source data extracted
- In-memory conversion to DB2 format
- Data loaded to Accelerator
- Table is enabled for acceleration
In Memory Data Transformation

IBM DB2 Analytics Accelerator

Batch DSNUTILB

Accelerator Loader

Source SQL Statement

IBM z

SQL Result Set

Shared memory objects

Data I/O Hybrid SRB/TCB

Virtual tables Mappings

Relational and Non-relational data

DRDA Sources (Oracle) IMS DB Sequential VSAM Remote DB2
DB2 Analytics Accelerator Loader Infrastructure

Inter Server Communication

- DB2 Analytics Accelerator Loader Infrastructure
- Inter Server Communication
- zOS Sysplex
- Loader V2.1 Batch Load Job
  - EXEC PGM=DSNUTILB
- z/OS
  - Started Task (Services)
- Accelerator Loader Server
  - DRDA
  - JDBC
- Accelerator Loader Server
  - DRDA
  - JDBC
- zOS Sysplex
  - APAR - PI65565
  - DB2 LUW Oracle Federation Server
Loading non-DB2 Data Sources into Accelerator

Solution Comparison

- Native Solution without Tooling
  - Multi-step manual solution

- Accelerator Loader V1.1
  - Partially automated solution

- Accelerator Loader V2.1—Automates entire process:
  - User builds a select statement from data source(s) (IMS, VSAM, Oracle, …)
  - **Automatically** creates the DB2 table
  - **Automatically** adds table to Accelerator
  - **Automatically** extracts specified source data
  - **Automatically** converts data to necessary DB2 format (in memory)
  - **Automatically** loads data to Accelerator
  - **Automatically** enables table for acceleration

Single batch job!
Lab Test Results Loading VSAM Data

VSAM KSDS 200 GB Data

VSAM Load Performance

- A: COBOL EXTRACT OF VSAM + STANDARD DB2 LOAD + STANDARD ACCELERATOR LOAD
- B: COBOL EXTRACT OF VSAM + LOADER V1.1 USED TO LOAD ACCELERATOR
- C: LOADER VERSION 2.1 DIRECT LOAD FROM VSAM

Minutes

Environment

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>z13 Model 739: z/OS LPAR with:</td>
</tr>
<tr>
<td></td>
<td>4 Dedicated CPs, 14 Dedicated zIIPs</td>
</tr>
<tr>
<td></td>
<td>MT=1</td>
</tr>
<tr>
<td>Memory</td>
<td>65 GB</td>
</tr>
<tr>
<td>z/OS</td>
<td>V2R2</td>
</tr>
<tr>
<td>DB2</td>
<td>V11</td>
</tr>
<tr>
<td>Accelerator DB2 SPs</td>
<td>V51</td>
</tr>
<tr>
<td>Accelerator H/W</td>
<td>N3001-010 (Mako full-rack)</td>
</tr>
<tr>
<td>Accelerator S/W</td>
<td>V51 Beta w/NPS 7205</td>
</tr>
<tr>
<td>Loader V21</td>
<td>Pre-GA: Code Drop 12</td>
</tr>
<tr>
<td>Key parameters</td>
<td>ACCEL_LOAD_TASKS = 16</td>
</tr>
<tr>
<td></td>
<td>MR Threads = ENABLED</td>
</tr>
</tbody>
</table>
Accelerator Loader V2.1 Direct Data Load Details

- Only Accelerator version of table will be loaded
  - If data desired in both DB2 and accelerator, extract to file and use external Dual Load
- Load RESUME is supported
  - User in control of what data is appended
- Source data is not locked during load
  - User can control (at the source) if necessary for data consistency
  - Cloning Tool an option for IMS data
- Performance can be tuned
  - Read Parallelism: VSAM and Sequential now
  - Write Parallelism: Number of pipes to accelerator
- Data Validity Checking – Discard Processing
  - Data type checking performed during load
  - Only messages issues on invalid data (no discard file support)
  - No duplicate key checking
    - Can be done via SQL post loa
Loading Data via Data Studio Interface
High Level Description

- Create Virtual Table (if non-relational)
  - Virtual table represents data source (mapping process)
  - Virtual view can be created to further refine data
    - where clause, eliminate columns, SQL data conversions (cast)
  - If single relational data source – No need to create virtual table
- Generate select statement from relation source or virtual table/view that represents data desired in accelerator
- Test select on small result set to verify it returns desired data
  - SQL result set represents data that will be loaded
  - Test and adjust until happy with result set
- Choose the DB2 subsystem and the Accelerator to load
- Build JCL to perform process
  - Creates DB2 table that matches SQL result set
  - Adds table to accelerator
  - Performs extract of source data and direct load to accelerator
  - Enables acceleration
- Submit JCL and view results
Loading Relational Data Sources

- Remote DB2 System (not attached to accelerator) or relational data through DB2 LUW AESE or DB2 Federation Server (Oracle)
- No need to create a virtual table if single relational data source
- Can join two or more tables to load a single accelerator table
- Data Studio plug-in provides method to generate query
  - Where clause and SQL functions can be used
- If joining data from more than one relational data source or joining data from relational and non-relational sources
  - Create virtual table that represents data from each source
  - Create virtual view that joins data from each virtual table
  - Load data from virtual view
Loading Non-Relational Data Sources

- VSAM, IMS, Sequential, Adabas
  - Define copy book source library
  - Import copy book
  - Each field of copy book will be represented as table column
  - Choose parallelism to read data source
  - Designate read exit (required for compression)
- Virtual table not required for relational data sources unless there is need to join data with non-relational or other relational data sources
- Create virtual view to:
  - Refine data via SQL functions or casts
  - Subset data via where clause
  - Join two or more virtual tables
Accelerator Loader High Availability Load

1. Reduced Elapsed Time
2. Reduced CPU
3. Simplified Operations
4. Batch ALT syntax
5. Additional options

Loads to multiple Accelerators are performed in parallel!
Accelerator Loader Benefits

- LOAD RESUME support
  - Avoid the need for full LOAD replace
- Accelerator-only table (AOT) Support
- Support for tables in replication mode when loading from a file
- Usability and control features
  - Optionally add table to accelerator, Remove/Add, Enable Acceleration
  - Parallelism options
- Performance enhancements
  - Improved:
    - Elapsed Time
    - CPU Time
    - zIIP Offload
- Improved ISPF interface
Operational Analytics

- SMF Records
- RMF Data
- Syslog
- DB2 Logs
- Audit Data
- Websphere Logs
- Custom Metrics
- CICS Logs
- SQL Performance
- Transactional Performance
Large Financial Data Virtualization Use Case

- IDAA placed via a Solution
  - Never have found a viable analytics workload to run on this appliance

- Global Clearing Management System
  - 7 clearing cycles per day for clearing funds between Issuers & Acquirers
  - Interchange Adjustment (IA) process
  - Attended an IDAA PoC which started the discussion of using IDAA for IA

- IA is not an analytics workload, but does fit a Big Data use case
  - Mostly manual process today
  - Each case can bring unique requirement
  - Can take weeks

- The solution
  - Load the payment transactions from each clearing cycle into IDAA (back loading 2 years)
  - When IAs come up SQL will be used to identify and stage transactions for adjustment
    - COBOL is written today to read from VTS
  - This will then flow through the IA process
IDAA PoC Challenge / Response

- The foundation of this process are ISO-8583 records
  - This is a very loose standard with a bit map identifying groups / subgroups of data
- This data was considered too voluminous to maintain online in transactional systems
- The IDAA platform (along with it’s compression capabilities)
  - Offered the storage at an acceptable price point
  - Provided a platform for potentially intense queries that would not interfere with the transactional environment
  - GCMS is a traditional CICS / COBOL DB2 application today, so this is natural
- The IDAA Loader (Data Virtualization) provides for:
  - Loading the appliance without having to load the data into DB2
    - Can provide parallelism for this load (even from tape)
  - Has plan to load multiple accelerators in parallel
  - Has virtualization capabilities to federate many non-DB2 sources
  - Mapping exits / Rocket consulting to develop mapping routines for complex formats
  - Provides LOAD RESUME capabilities (possible future use)
Accelerator Data Consolidation

One Consolidated Access Point

- Distributed Databases
- TCPIP or DRDA
- IBM DB2 Accelerator Loader

Expanded Use Cases
- Relational
- IMS
- VSAM
- SMF
- Sequential

Simplicity
Additional Accelerator Customer Use Cases

- Customizable Virtual Views
  - Loads the result set of the Virtual View into the Accelerator

- Data Validation
  - Data validation tends to be costly
  - Efficiently exploiting Accelerator to validate different versions of the data
QMF for z/OS – an enterprise business analytics suite

- Access all forms of enterprise data on any device
- Add more data sources or users without increasing price
- Addresses all user types and BI needs – No need for separate BI tools for specific data sources or user profiles

- True “data federation” - combine data from multiple structured data sources with one SQL statement
- Combine data from structured and unstructured data
- QMF is zIIP eligible, lowering mainframe CPU consumption

Mainframe Data Sources
- DB2 for z/OS, IMS, VSAM, IDAA, SMF, Tape, SysLog, Operlog, Log Stream, Sequential, ADABAS

Big Data
- Hadoop, Spark, Netezza, Teradata, Cloudant

Non-Mainframe Data Sources
- DB2 LUW, DB2 iSeries, Derby, MySQL, SQL Server, Informix, Oracle, Sybase, JDBC/ODBC, web services

Self-service Analytics

✓ QMF users gain direct access to additional data sources: VSAM, sequential files, ADABAS, and more
✓ QMF users can create their own visualizations and collaborate with other users real-time – truly self service BI!
QMF 11.2.1 with Data Virtualization

- Allows QMF to access non-relational data as if it were relational
- Same ‘mapping’ process as in Accelerator Loader
  - Maps can be shared across tools, but each has its own address space
- VSAM, Flat Files, Analysis of SMF data, IMS, ADABAS, etc.
- Exploits a z/OS server for query processing
  - Can be more performant than using JDBC driver to IMS and joining with DB2
  - With the virtualization technology, the data joining happens in z/OS server address space, not client side
- May not be appropriate where query is executed often and fast result set is needed when accessing large amounts of data
- Complimentary to Accelerator and Loader solution
  - If repeated access and/or high speed results required, load data to IDAA first
What Is Spark?

- An Apache Foundation open source project. Not a product.
- An in-memory compute engine that works with data. Not a data store.
- Enables highly iterative analysis on large volumes of data at scale.
- Unified environment for data scientists, developers and data engineers.
- Radically simplifies process of developing intelligent apps fueled by data.

Why Spark?

- Spark is open so accelerates community innovation
- Spark is fast 100x faster than Hadoop MapReduce
- Spark is about all data for large scale data processing
- Spark supports agile data science to iterate rapidly
- Spark can be integrated with IBM solutions
Apache Spark is…

Fast
- Leverages aggressively cached in-memory distributed computing and JVM threads
- Faster than MapReduce for most workloads

Ease of use (for programmers)
- Written in Scala, an object-oriented, functional programming language
- Scala, Python and Java APIs
- Scala and Python interactive shells
- Runs standalone, on Hadoop, Mesos, or cloud

General purpose
- Covers a wide range of workloads
- Provides SQL, streaming, machine learning, graphing

Logistic regression in Hadoop and Spark

from http://spark.apache.org
Apache Spark Use Cases

- **Customer Data**
  - Fraud Detection
  - Customer retention
  - Suggestive selling

- **DB2 Tooling Possibilities**
  - Operational Insights
    - SMF, RMF, Audit data, DB2 Performance Tools, etc
  - Anomaly Detection
    - DB2 Log processing
    - DB2 Audit logs
  - Prediction
    - DB2 Performance tuning
    - Utility run time execution
IBM z/OS Platform for Apache Spark, V1.1

z Systems & Apache Spark – A unified analytics platform

Key Business Transaction Systems

Spark Applications: IBM and Other Commercial Vendors

Spark SQL  Spark Stream  MLib  GraphX

Apache Spark Core

RDD  RDD  RDD  RDD

Rocket Mainframe Data Service for Spark

Type 2 / Type 4

DB2 z/OS  IMS  VSAM  SMF

Type 4

Oracle  HDFS

and more…
Spark z/OS & IDAA -- Complementary

- IDAA and Spark are complementary solutions. What makes sense to use depends on what needs to be done.
  - Spark SQL that is initiated on DB2 which qualifies for acceleration will be accelerated on the IDAA transparently to the Spark application
  - Spark does not do query acceleration; IDAA is primarily a query accelerator
  - Spark is not a data store; IDAA can be a data store
  - IDAA interaction is via SQL interfaces to DB2 z/OS; Spark interaction is via Spark jobs written in scala, python, java ---and there are capabilities to interact via machine learning and graph analytics, as well as SQL
  - Spark can access data-in-place, real-time across a variety of environments without moving the data; IDAA data is largely from DB2, other data can be loaded into IDAA.