Maximize Your Return on Big Data

May 15-18, 2012
Informatica / Teradata Best Practices

Chris Ward, Sr. Consultant, DI CoE, Teradata

John Hennessey, Sr. Consultant, DI CoE, Teradata
Teradata DI Center of Expertise

• Establish and Promote Teradata ETL Best Practices
  • Incorporate 3rd Party ETL Tool Methodology into TSM (Teradata Solution Methodology)

• Centralized Involvement on all Engagements
  • Learn from other accounts - “We ran into this problem”
  • Leverage innovation – “We solved it like this”
  • Serve the client – “What are your ETL needs”

• Resource Management
  • Centralized Talent Base
  • Training (Formal, OJT, Mentoring, Cross Training)
  • Right person, Right place, Right time
Informatica-Teradata Overview

• **Strategic Teaming**
  • Strategic alliance since 2000 (INFA Reseller since 2002)
  • Partnership delivers solutions for end-to-end Enterprise Data Warehousing

• **3rd Party Tool Relationship**
  • Direct line to Support
  • Direct line to Engineering
  • Quick Escalation when needed

• **Push Down Optimization (PDO)**
  • A direct result of Collaboration of the CTO from both Teradata and Informatica resulted in the creation of PDO
Teradata / Informatica Best Practice

- Load into stage table in Teradata as fast as possible (via TPT API)
- During load, convert and cast fields to the correct data type. Minimize cast/convert functions later specifically for columns which will be joined on.
- Use SET In-Database Processing taking advantage of the Massively Parallel Processing (MPP) Nature of Teradata to move the data from stage to Core ie (via PDO)
- Maintain Field Level Lineage for maximum traceability (via IMM)
Basics - Informatica / Teradata Communication

- Informatica ‘talks’ to Teradata Tools and Utilities (TTU) installed on Informatica Server.
- TTU communicates with Teradata database.

Installation / Test Order:
1. Install Teradata Database
2. Install TTU On Informatica Server
3. Install ALL TTU eFixes (patches).
4. Test TTU at the command line (ie TPT/fastload/mload/tpump/odbc)
5. Install Informatica PowerCenter
6. Install Informatica TPT API module
7. Test Informatica to Teradata Load
What is Teradata Parallel Transporter (TPT)

• Teradata’s premier load product
• Combines many load protocols (FastLoad, MultiLoad, FastExport, TPump) into one tool
• Database load protocols are same, client code rewritten
• No learning curve for protocols – learn new features
• Most everything about load tools still applies
  • Similar basic features, parameters, limitations (e.g., number of concurrent load jobs), when to use, etc.
TPT - Continued

• TPT API exposed to 3rd Party Tools

• Parallel input streams for performance on load server
  • Data and functional parallelism on the client box

• Benefits:
  • Easier to use
  • Greater Performance (scalable)
  • API Integration with 3rd Party Tools.
Teradata Load / Unload Operators

• **TPT Operators**
  • Load - Bulk loading of empty tables
  • Update – Bulk Insert, Update, Upsert, & Delete
  • Export – Export data out of Teradata DB
  • Stream – SQL application for continuous loading

• **Replace old external Utilities**
  • Load replaces Fastload
  • Update replaces Mload
  • Export replaces Fast Export
  • Stream replaces TPump
Before API – Integration with Scripting

1. INFA PowerCenter/PowerExchange creates Teradata utility script & writes to file.
2. INFA PowerCenter/PowerExchange reads source data & writes to intermediate file (lowers performance to land data in intermediate file).
3. INFA invokes Teradata utility (Teradata doesn’t know INFA called it)
4. Teradata tool reads script, reads file, loads data, writes messages to file
5. INFA PowerCenter/PowerExchange reads messages file and searches for errors, etc.
With API - Integration Example

1. INFA PowerCenter/PowerExchange passes script parameters to API
2. INFA PowerCenter/PowerExchange reads source data & passes data buffer to API
3. Teradata tool loads data and passes return codes and messages back to caller
Parallel Input Streams Using API

PowerCenter Launches Multiple Instances that Read/Transform Data in Parallel

Teradata Parallel Transporter Reads Parallel Streams with Multiple Instances Launched Through API

Teradata Database
Why TPT- API

- **Scalable increased performance**
  - Use the power of PowerCenter and TD parallel processing
  - Multiple load stream instances from source through TD target
  - No landing of data, just pass buffers in memory
  - 64 bit instead of TTU 32 bit

- **Provides Tight Integration with PowerCenter**
  - ETL Integration is faster and easier
  - PowerCenter has control over the entire load process
    - Errors and statistics returned programmatically
How to use TPT with Informatica
Session Source / Target Configuration
Teradata TPT Recommendations

• Stage mappings should:
  • Bring over all of the fields exactly as they are in the source
  • Insert these raw fields into VARCHAR fields to insure they are accepted
  • Explicitly convert the format of all dates and timestamps into Teradata format (keeping the type VARCHAR within Informatica)
  • If the conversions are unsuccessful, fill the fields with NULL
  • Record that an Error record exists in the Database

• KISS Principle (Keep It Simple)
What is Pushdown Optimization (PDO)?

ETL/ELT processing can be combined

Pushdown Processing

Push transformation processing from PowerCenter engine to relational database
There are three main approaches:

1. **ETL Approach**: (1) Extract from the source systems, (2) Transform inside the Informatica engine on integration engine servers, and (3) Load into target tables in the data warehouse.

2. **ELT Approach**: (1) Extract from the source systems, (2) Load into staging tables inside the data warehouse RDBMS servers, and (3) Transform inside the RDBMS engine using generated SQL with a final insert into the target tables in the data warehouse.

3. **Hybrid ETLT Approach**: (1) Extract from the source systems, (2) Transform inside the Informatica engine on integration engine servers, (3) Load into staging tables in the data warehouse, and (4) apply further Transformations inside the RDBMS engine using generated SQL with a final insert into the target tables in the data warehouse.
ETL versus ELT

When does **ETL** win?

- Ordered transformations not well suited to set processing.
- Integration of third party software tools best managed by Informatica outside of the RDBMS (e.g., name and address standardization utilities).
- Maximize in-memory execution for multiple step transformations that do not require access to large volumes of historical or lookup data (note: caching plays a role).
- Streaming data loads using message-based feeds with “real-time” data acquisition.
ETL versus ELT

When does **ELT** win?

- Leverage of high performance DW platform for execution reduces capacity requirements on ETL servers - this is especially useful when peak requirements for data integration are in a different window than peak requirements for data warehouse analytics.

- Significantly reduce data retrieval overhead for transformations that require access to historical data or large cardinality lookup data already in the data warehouse.

- Batch or mini-batch loads with reasonably large data sets, especially with pre-existing indices that may be leveraged for processing.

- Optimize performance for large scale operations that are well suited for set operations such as complex joins and large cardinality aggregations.
ETL versus ELT

The Answer: Hybrid ETLT

• Best of both worlds!

• Use the appropriate combination of ETL and ELT based on transformation characteristics.

• Informatica creates and defines metadata to drive transformations independent of the execution engine.

• ETL versus ELT execution is selectable within Informatica and can be switched back-and-forth without the redefinition of transformation rules.

Note that not all transformation rules are supported with pushdown optimization (ELT) in the current release of Informatica 8.x.
Why PDO (1)

• Cost-effectively scale by using a flexible, adaptable DI architecture
  • Do the work in the “appropriate” environment

• Increase developer and team productivity
  • All code in one development tool
  • Developers know one tool

• Eliminate the need to write custom-coded solutions
  • No longer need to develop Bteq Scripts
Why PDO (2)

• Easily adapt to changes in Teradata Database
  • Future versions will adapt to the new TD functionality

• Improved traceability through metadata management
  • Better understanding and estimation on Impact of change
  • From source systems to EDW to BI reporting
  • On the contrary, Bteq does not offer metadata

• Delivered superior performance
  • Data never leaves Teradata
  • Heavy transformation from stage to work layers
Why PDO (3)

• Meet SLA
  • Well-positioned to make data available on time
PDO Table to Table Comparison

- 10 Million rows copied from a table in Teradata to another table in Teradata

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC</td>
<td>6hrs +</td>
</tr>
<tr>
<td>PDO</td>
<td>3 sec</td>
</tr>
</tbody>
</table>
## Supported Transformations

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Full</th>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregator</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Expression</td>
<td>Yes$^1$</td>
<td>Yes$^1$</td>
<td>Yes$^1$</td>
</tr>
<tr>
<td>Filter</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Joiner</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Lookup</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes$^2$</td>
</tr>
<tr>
<td>Unconnected Lookup</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Router</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sequence Generator</td>
<td>Yes$^2$</td>
<td>Yes$^2$</td>
<td>Yes$^2$</td>
</tr>
<tr>
<td>Sorter</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Union</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Update Strategy</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

1 PowerCenter expressions can be pushed down only if there is an equivalent database function
2 Not all databases are supported, Refer to documentation for further details
Informatica Metadata Manager (IMM)

What problems does it solve?

**Analyst /Subject Matter Expert**
- Lack confidence on the data presented to business users
- Find it hard to relate business concepts to technical artifacts
- Need to collaborate with developers for greater productivity

**Developer**
- Too much time spent on impact analysis vs. actual work
- Too much time spent trying to locate technical artifacts and recreating what already exists

**Architect**
- Costs associated with managing disparate systems generating metadata
IMM – How it works.

Informatica PowerCenter Repository

Informatica Metadata Manager Server

BI Tool Metadata

Database Schema(s) (tables/views/procedures)

Data Modeling Tool

Custom

Other

Imports
Personalized lineage increases insight into complex data relationships

- Intuitive, dynamic data lineage enables users to understand what the data means, where it came from, where it’s going, and how it’s changed
- Integrated profiling results for additional insight into data
- Highlight, filtering and personalization capabilities improve usability
PDO Tips - Parallel lookups:

- A mapping containing parallel lookups cannot be pushed to the database
- Design the mapping and change the lookups into JOINERS
PDO TIPS - Unconnected lookups

- A mapping containing unconnected lookups might perform slower than the integration service’s DTM when utilizing PDO.
- The Cause is SQL generated by the integration service can be complex and slow as unconnected lookups are converted into outer joins.
- Compensate by changing the lookups to joiners whenever possible.
PDO TIPS: INSERT ONLY & INSERT/UPDATE:

- A Mapping that is INSERT only, that needs to do UPDATES, or both will need to be converted to the following as Informatica has difficulty creating SQL when JOINERS or LOOKUPS are used.
- We are working with Informatica to correct this problem.
PDO Tips: Sorted Aggregation

- Mappings containing an Aggregator downstream from a Sorter transformation can not utilize pushdown
- Handle this by redesigning the mapping to achieve full or source-side pushdown optimization, configure the Aggregator transformation so that it does not use sorted input, and remove the Sorter transformation
PDO TIPS: Variable Ports

Pushdown Optimization is not possible for this mapping because variable ports are not supported. Consider replacing the following

\[(\text{NET\_AMOUNT} = \text{AMOUNT} - \text{FEE}, \text{DOLLAR\_AMT} = \text{NET\_AMOUNT} \times \text{RATE})\]

with

\[(\text{DOLLAR\_AMT} = (\text{AMOUNT} - \text{FEE}) \times \text{RATE})\]
Teradata PDO Best Practices (1)

• Avoid unconnected lookups whenever possible

• Avoid parallel lookups for sessions that would benefit from PDO – create sequential lookups

• Eliminate sorted aggregation and pass-through ports in aggregators

• Eliminate variable ports

• Use Full Pushdown Optimization – because of large Teradata data volumes, best performance can be obtained by doing all processing inside the database
Teradata PDO Best Practices (2)

• When using Pushdown overrides with view – override should contain tuned Teradata SQL

• Filter data using WHERE clause before doing outer joins

• Avoid full table scans for large tables

• Use staging processing if necessary

• Validate use of primary and secondary indexes
# Informatica-Teradata Compatibility

<table>
<thead>
<tr>
<th>INFA</th>
<th>Stand Alone Loaders</th>
<th>TPT-API</th>
<th>PDO</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC 9.0.1 GA – June ‘10</td>
<td>TTU 13.10, TTU13.0, TTU12</td>
<td>TTU13.10, TTU 13.0, TTU12</td>
<td>TTU 13.10, TTU13.0, TTU12</td>
</tr>
<tr>
<td>PC 9.0.0 (no upgrade from 8.6.1) GA – Nov ’09</td>
<td>TTU13.0, TTU12</td>
<td>TTU13.0, TTU12</td>
<td>TTU12</td>
</tr>
<tr>
<td>PC 8.6.1 GA – June ’08</td>
<td>TTU13.0, TTU12, TTU8.2</td>
<td>TTU13.0¹, TTU12, TTU8.2</td>
<td>TTU13.0, TTU12, TTU8.2</td>
</tr>
<tr>
<td>PC 8.6.0</td>
<td>TTU12, TTU8.X, TTU7.1</td>
<td>TTU12 , TTU8.2</td>
<td>TTU12, TTU8.2</td>
</tr>
<tr>
<td>PC 8.5.1</td>
<td>TTU12, TTU8.X,TTU7.1</td>
<td>Not Available</td>
<td>Yes</td>
</tr>
</tbody>
</table>

8.6.1.0.2 PowerExchange for Teradata PT API Plug-in (at a minimum) is required for TPTAPI13.0 Support.
Thank you

Chris Ward
Teradata Sr Consultant
ChristopherPaul.Ward@Teradata.com
312.543.9284

John Hennessey
Teradata Sr Consultant
xxxxx@Teradata.com