

Building a Dedicated Analytic Server with



John Sichi
Project Founder

A Bit About LucidDB

- GPL v2 (w/LGPL for JDBC driver)
- Developed by LucidEra
 - part of its SaaS business intelligence stack
 - used in production apps since 2006
 - 100+ trial/customer deployments
 - built on Eigenbase frameworks
 - design derived from an earlier commercial column store (Broadbase)

Why More Than One DBMS?



Because It's Worth The Complexity

- Row store

- Good for transactions
- Compression difficult
- Small scattered writes
- Row versioning
- I/O bound

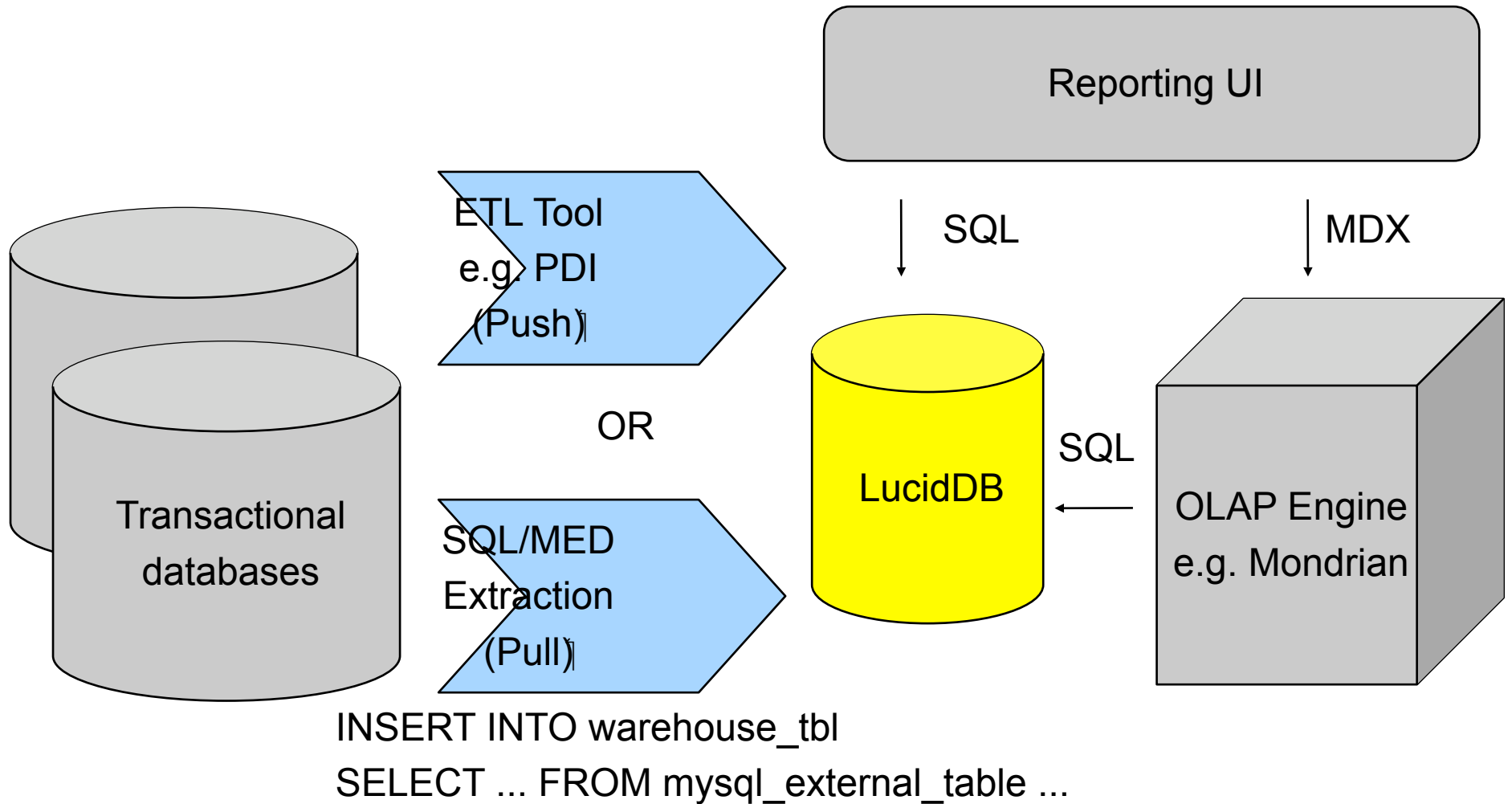
- Column store

- Good for queries
- Compression easy
- Bulk load
- Page versioning
- CPU bound

Data Transformations

- No transformation:
 - run analysis queries directly against OLTP system
- Physical:
 - replicate to read-only slaves; additional indexing
 - load into column store, materialize views
- Logical:
 - denormalize into stars (facts and dimensions)
 - fact archive/summary tables (sales by region for 2005)
 - dimension history tracking (customer single -> married)

System Dataflow



So, Why Use LucidDB?

- OLAP query acceleration
 - Column store, bitmap indexes, hash join/agg
 - Cost-based star join optimization
 - *Mondrian aggregate table builder*
- Extract/transform/load expressed as SQL
 - SQL/MED data extraction
 - Upsert, user-defined transformations
 - Page multiversioning, hot/incremental backup
 - *Pentaho Data Integration bulk load via fifo pipe*

Extract/Transform/Load

- With an external ETL tool
 - tool pushes data into fifo pipe; LucidDB pulls from other end via its flatfile reader
- Or cut out the middleman
 - LucidDB can query source system directly via JDBC
 - You can also write your own SQL/MED foreign data wrappers and plug them in

Pentaho Data Integration

The screenshot displays the Pentaho Data Integration (PDI) interface. The main window shows a job design with a 'CSV file input' step connected to a 'LucidDB Bulk Loader Plugin' step. A configuration dialog for the 'LucidDB Bulk Loader' is open, showing the following settings:

- Step name: LucidDB Bulk Loader Plugin
- Connection: LucidDB on localhost
- Target schema: PDI
- Target table: TERRITORIES
- Maximum errors: 0
- FIFO file path (folder): /tmp/ffifo/
- FIFO foreign server name: ffsrv

The 'Fields to load' section shows a table with the following data:

#	Table field	Stream field	Field format OK
1	Territory	Territory	Y

The 'Execution Results' window at the bottom shows the following table:

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active
1	CSV file input	0	0	5	6	0	0	0	0	Finished
2	LucidDB Bulk Loader Plugin	0	5	5	0	5	0	0	0	Finished

Applying Transformations

- Example transformations
 - Surrogate key assignment/lookup
 - Cleansing dirty data “Calif -> CA”
 - Deduplication “C.J. Date -> Chris Date”
- How?
 - express directly as SQL (e.g. CASE ... WHEN)
 - plug custom Java transformations into SQL
 - or use ETL tool's transformation library

LucidDB Agg Table Benefits

- Column store and bitmap compression work well with repeated values in aggregate compound keys
 - save disk space, I/O
- LucidDB hash aggregation is quite fast
 - reduce load time
- Page versioning efficiently guarantees a consistent query view across all tables

Mondrian Aggregate Designer

- LucidDB system procedure for automation

The screenshot displays the Pentaho Aggregation Designer interface. The window title is "Pentaho Aggregation Designer". The menu bar includes "File" and "Help". The toolbar contains icons for "Connection", "Advisor", and "Export".

Edit Aggregate Definition

Name: FoodMart_Sales_3

Description: [Year], [time_by_day].[Quarter], [customer].[Education Level]

Aggregation Levels

Dimension	Level
Store	Store State
Store Size in SQFT	(All)
Store Type	Store Type
Time	Time : Quarter
Product	(All)
Promotion Media	(All)
Promotions	(All)
Customers	(All)
Education Level	Education Level
Gender	(All)
Marital Status	(All)
Yearly Income	(All)

Selected Aggregates: 5
Number of Rows: 14232
Approximate Disk Space: 1339004 bytes

Chart: A bar chart showing Benefit (Y-axis) versus Cost (X-axis). The chart displays five bars of increasing height, with a blue line connecting the tops of the bars.

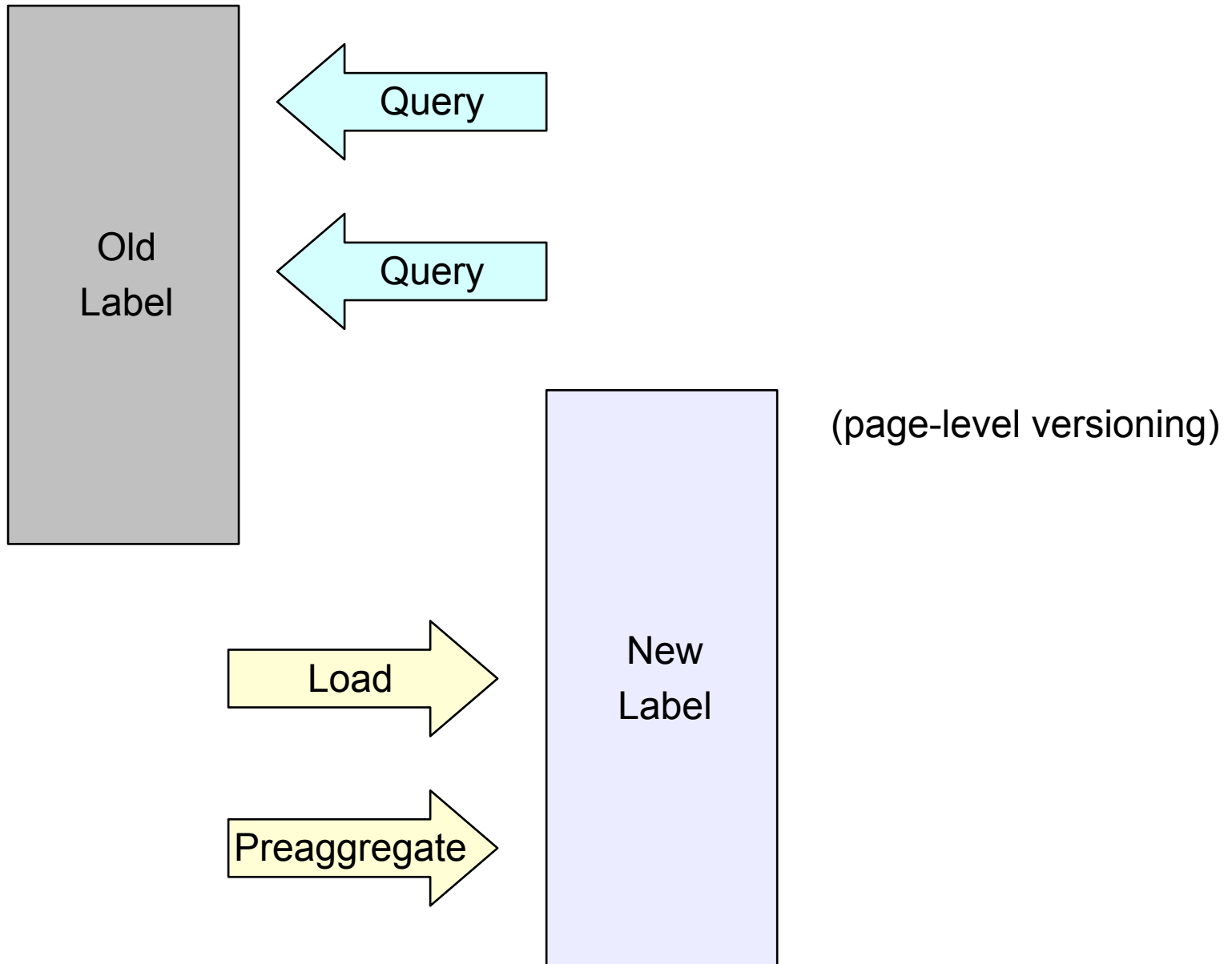
Table:

Type	Name	Row Count	Space (bytes)
<input checked="" type="checkbox"/>	Advisor FoodMart_Sales_1	96	2688
<input checked="" type="checkbox"/>	Advisor FoodMart_Sales_2	1560	74880
<input checked="" type="checkbox"/>	Advisor FoodMart_Sales_3	7200	748796
<input checked="" type="checkbox"/>	Advisor FoodMart_Sales_4	4896	489600
<input checked="" type="checkbox"/>	Advisor FoodMart_Sales_5	480	23040

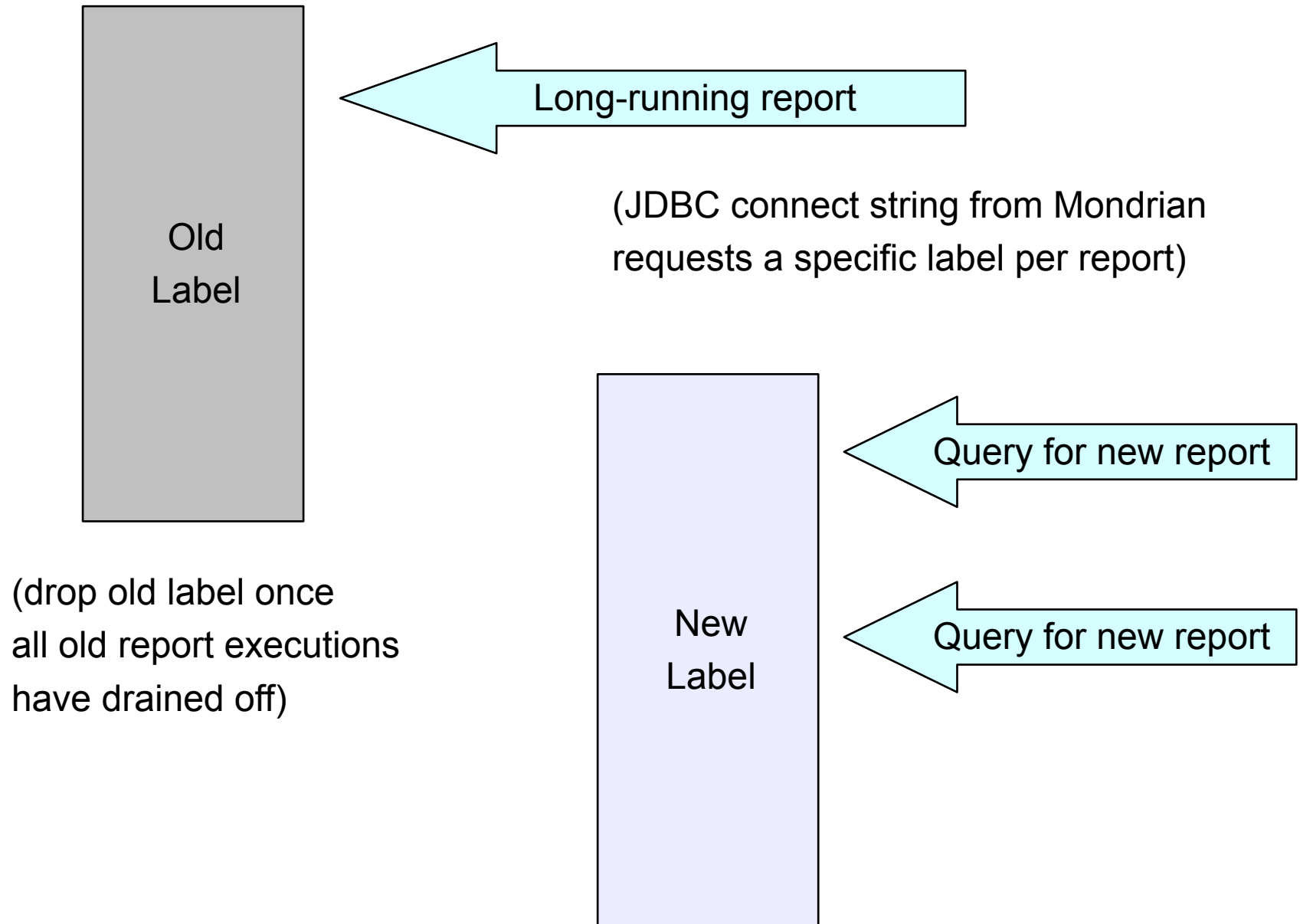
Buttons: Reset, Apply

Database: LucidDB FOODMART Schema: FoodMart Cube: Sales

Warehouse Labels



Rolling the Active Label



Performance (anecdotal)

- Biggest difference seen vs row store when data does not fit in memory (can be 10X and up)
 - <http://pub.eigenbase.org/wiki/LucidDbTp>
- When data fits in memory, more like 2X
 - <http://www.tholis.net/news/open-source-data-warehousing/>
 - <http://pentahomusings.blogspot.com/2009/02/lucid-warehouse-results-well-sorta.html>
- No rigorous comparisons performed yet; would like to collaborate on DBT-3 benchmarks etc
- SSD helps row store more than column store

Future Efforts

- Incremental view materialization
- SQL/OLAP support (e.g. fast TOP-N)
- Multicore parallel load/query
 - currently experimental
- Scale-out parallelism
 - multi-node load/query partitioning
- Tool support: generate SQL for ETL

References

- jsichi@gmail.com
- luciddb-users@lists.sf.net
- <http://www.luciddb.org>
- <http://pub.eigenbase.org/wiki>
 - [\[\[LucidDbPdiBulkLoad\]\]](#)
 - [\[\[LucidDbAggregateDesigner\]\]](#)

Attributions

- Images
 - <http://www.flickr.com/photos/jekemp/3424782/>
 - <http://www.flickr.com/photos/lenore-m/409731388/>
 - Mondrian aggregate table documentation
- Trademarks
 - LucidDB is a trademark of LucidEra, Inc.
 - Pentaho is a trademark of Pentaho, Inc.