



# Big Data Infrastructure

CS 489/698 Big Data Infrastructure (Winter 2016)

Week 6: Analyzing Relational Data (1/3)

February 11, 2016

Jimmy Lin

David R. Cheriton School of Computer Science  
University of Waterloo

These slides are available at <http://lintool.github.io/bigdata-2016w/>

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States  
See <http://creativecommons.org/licenses/by-nc-sa/3.0/us/> for details

# Structure of the Course

Analyzing Text

Analyzing Graphs

Analyzing  
Relational Data

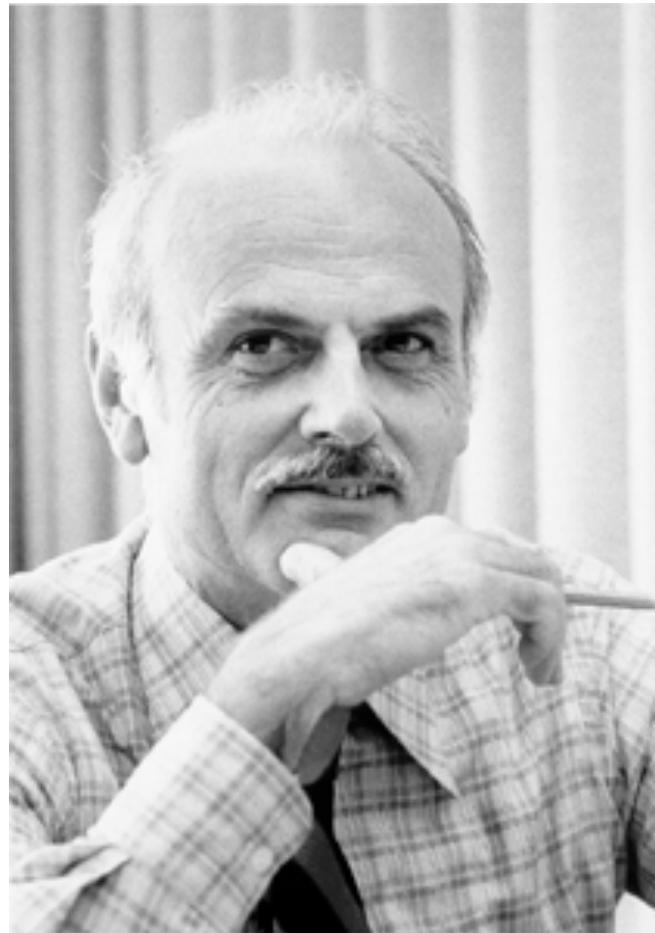
Data Mining

“Core” framework features  
and algorithm design

# **Business Intelligence**

An organization should retain data that result from carrying out its mission and exploit those data to generate insights that benefit the organization, for example, market analysis, strategic planning, decision making, etc.

**Duh!?**



# Database Workloads

- OLTP (online transaction processing)
  - Typical applications: e-commerce, banking, airline reservations
  - User facing: real-time, low latency, highly-concurrent
  - Tasks: relatively small set of “standard” transactional queries
  - Data access pattern: random reads, updates, writes (involving relatively small amounts of data)
- OLAP (online analytical processing)
  - Typical applications: business intelligence, data mining
  - Back-end processing: batch workloads, less concurrency
  - Tasks: complex analytical queries, often ad hoc
  - Data access pattern: table scans, large amounts of data per query

An organization should retain data that result from carrying out its mission and exploit those data to generate insights that benefit the organization, for example, market analysis, strategic planning, decision making, etc.

# One Database or Two?

- Downsides of co-existing OLTP and OLAP workloads
  - Poor memory management
  - Conflicting data access patterns
  - Variable latency
- Solution: separate databases
  - User-facing OLTP database for high-volume transactions
  - Data warehouse for OLAP workloads
  - How do we connect the two?

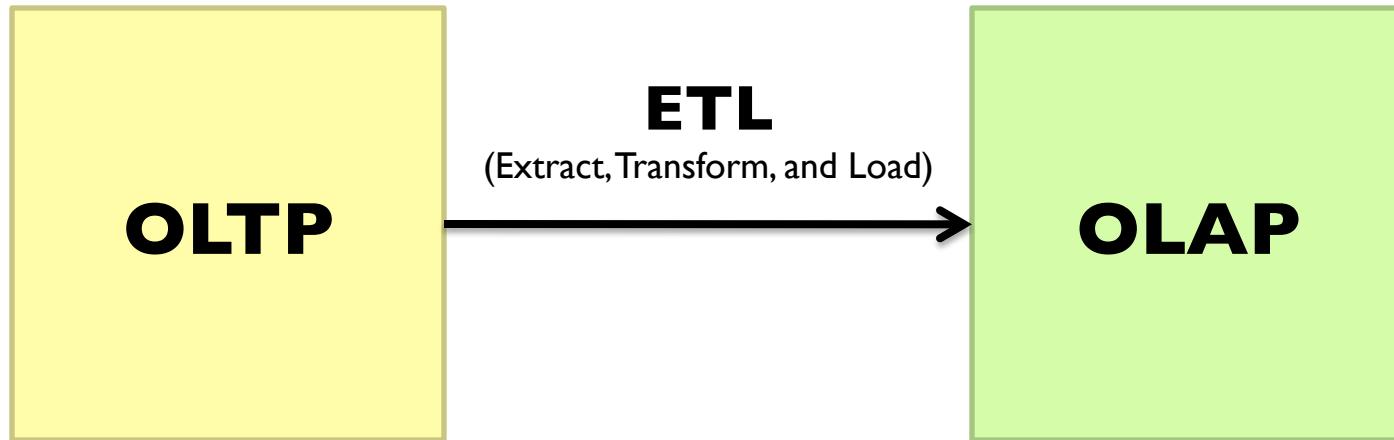


# Data Warehousing

# **OLTP/OLAP Integration**

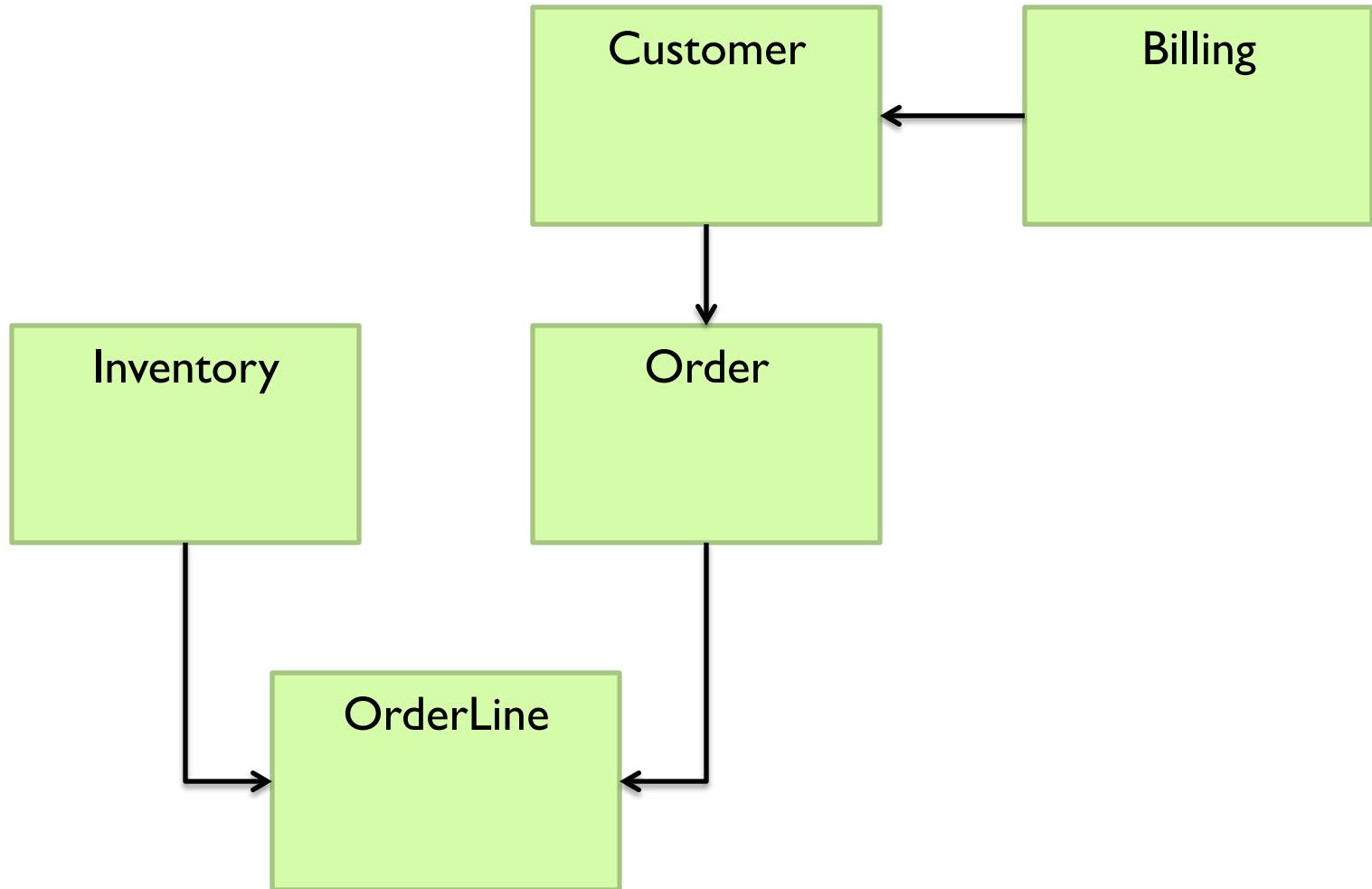
- OLTP database for user-facing transactions
- Extract-Transform-Load (ETL)
- OLAP database for data warehousing

# **OLTP/OLAP Architecture**

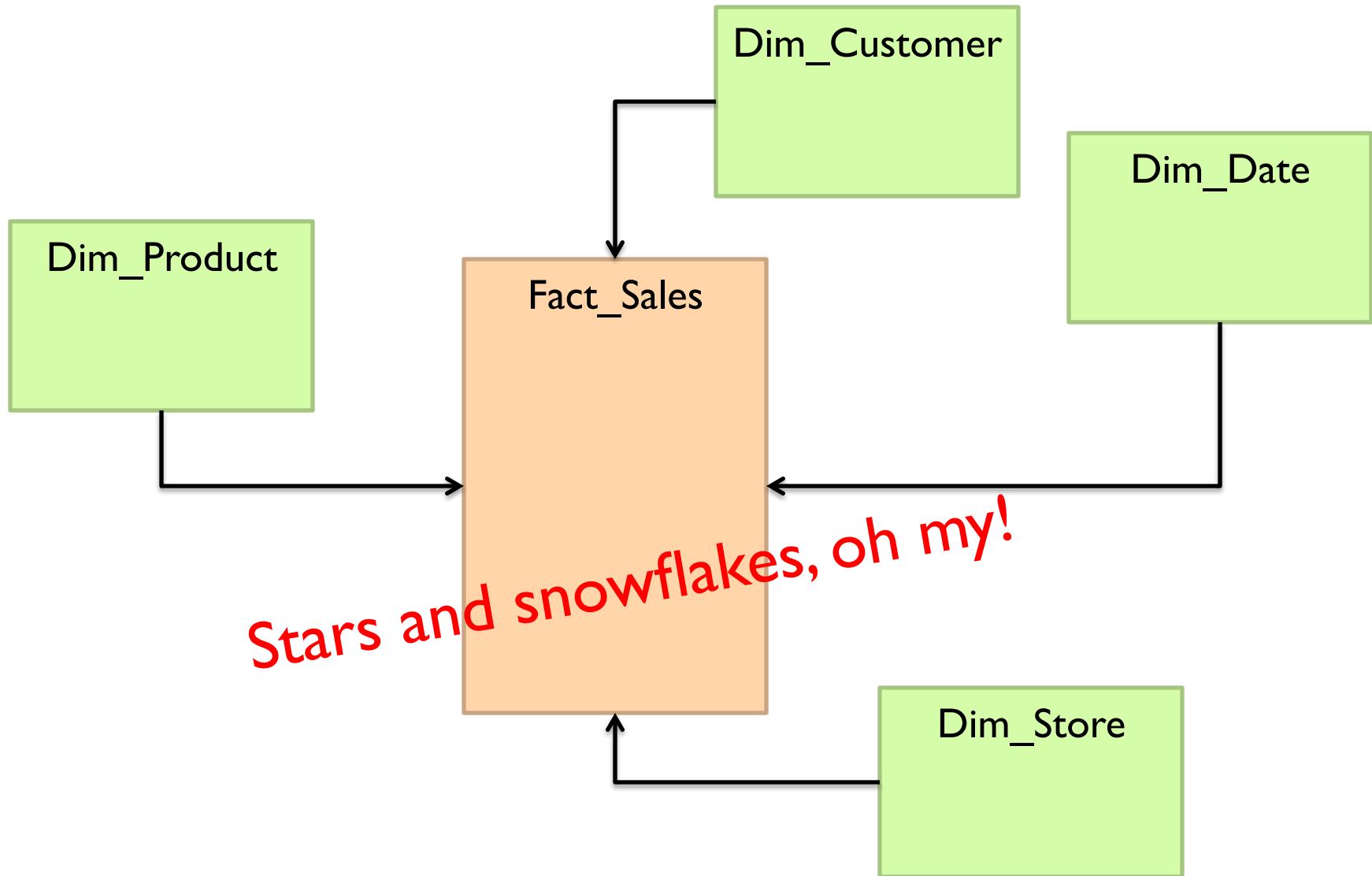


A simple example to illustrate...

# A Simple OLTP Schema



# A Simple OLAP Schema



# ELT

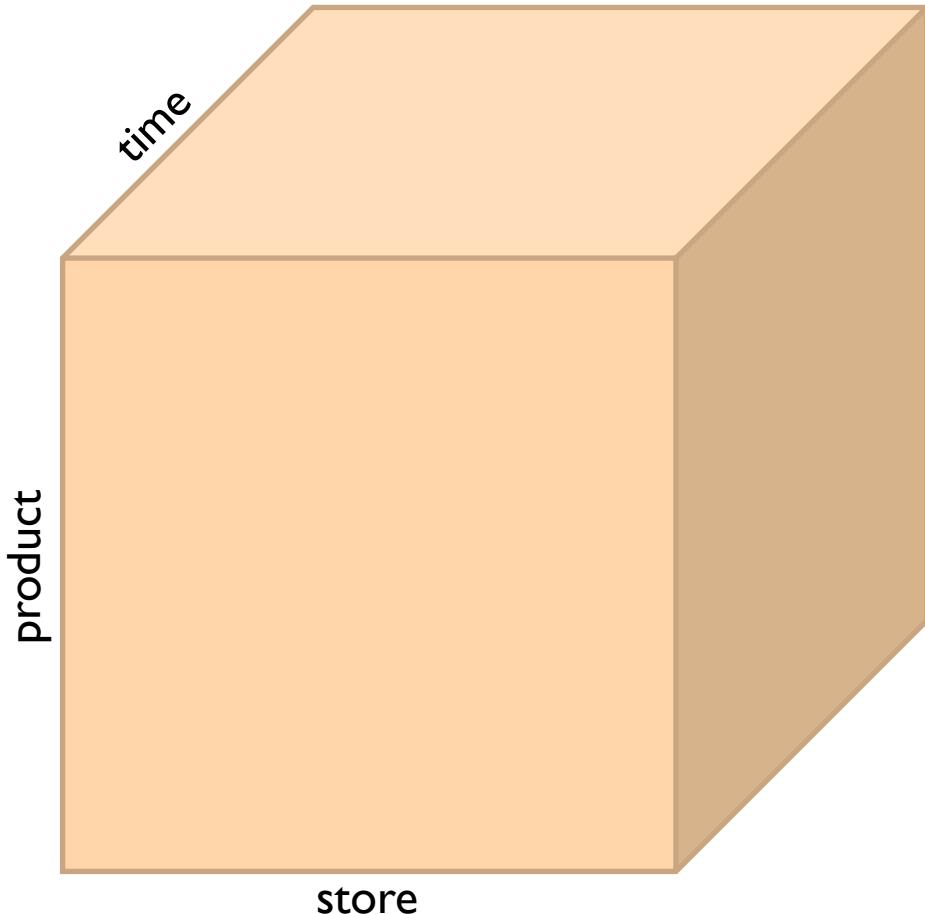
- Extract
- Transform
  - Data cleaning and integrity checking
  - Schema conversion
  - Field transformations
- Load

When does ELT happen?

# **What do you actually do?**

- Report generation
- Dashboards
- *Ad hoc* analyses

# OLAP Cubes



## Common operations

- slice and dice
- roll up/drill down
- pivot

# OLAP Cubes: Challenges

- Fundamentally, lots of joins, group-bys and aggregations
  - How to take advantage of schema structure to avoid repeated work?
- Cube materialization
  - Realistic to materialize the entire cube?
  - If not, how/when/what to materialize?

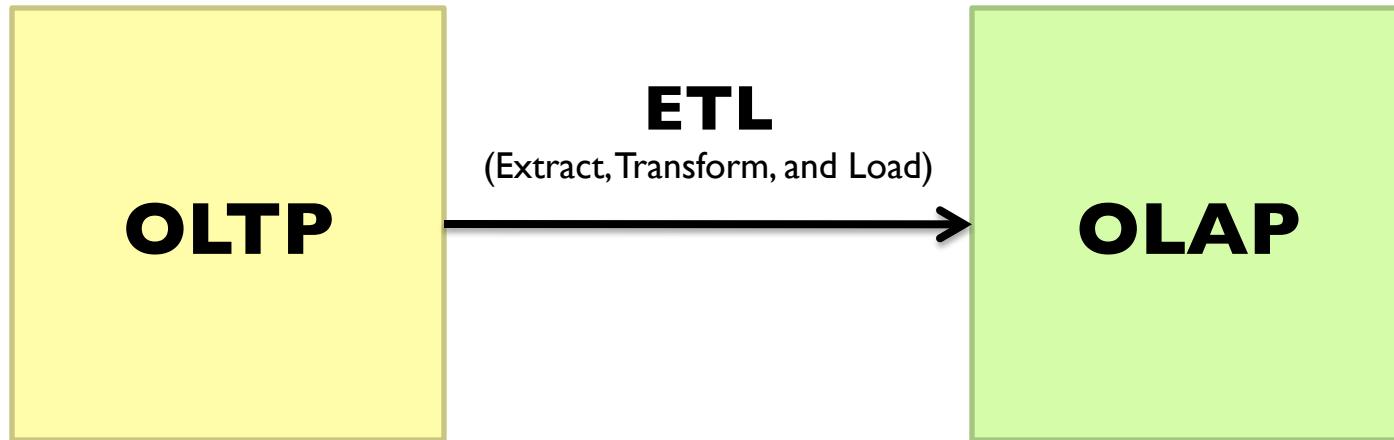
**Fast forward...**



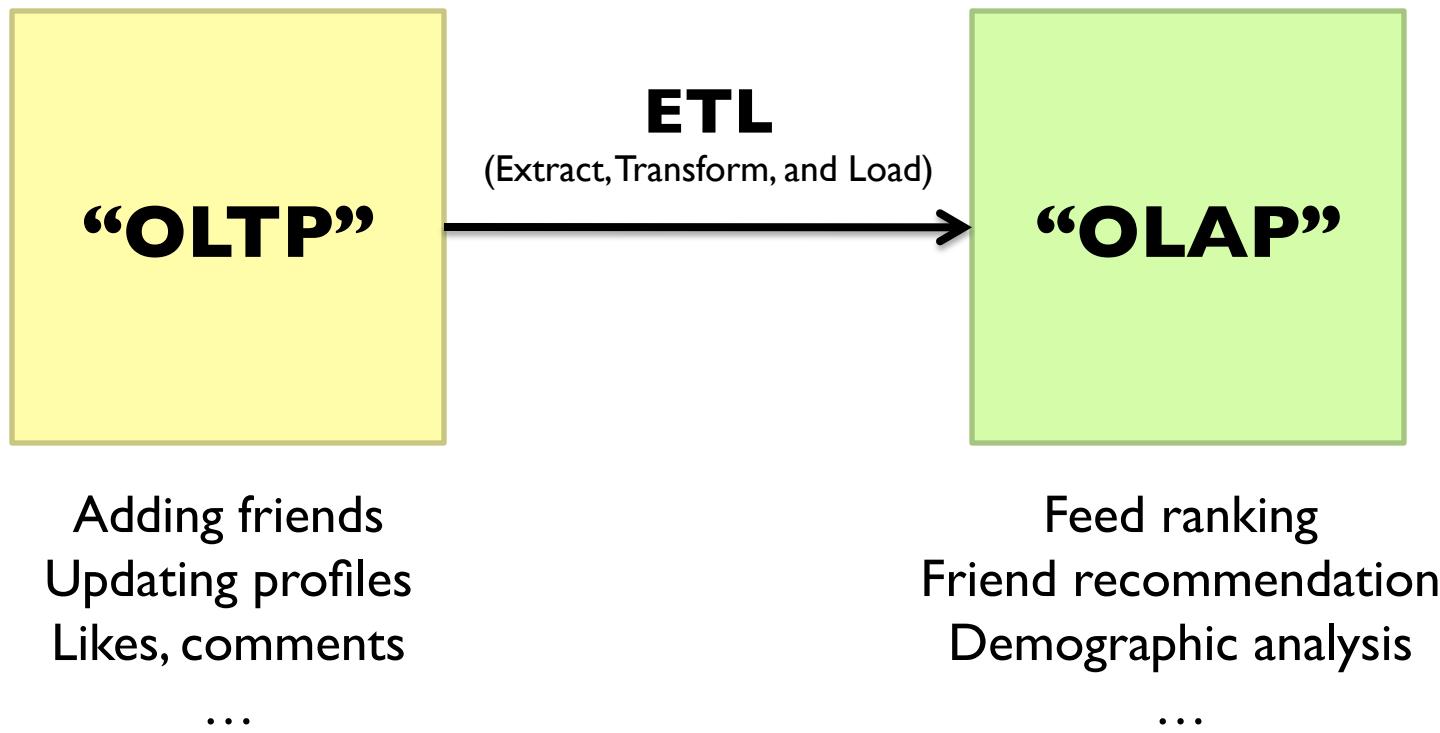
Jeff Hammerbacher, Information Platforms and the Rise of the Data Scientist.  
In, *Beautiful Data*, O'Reilly, 2009.

“On the first day of logging the Facebook clickstream, more than 400 gigabytes of data was collected. The load, index, and aggregation processes for this data set really taxed the Oracle data warehouse. Even after significant tuning, we were unable to aggregate a day of clickstream data in less than 24 hours.”

# **OLTP/OLAP Architecture**



# Facebook Context

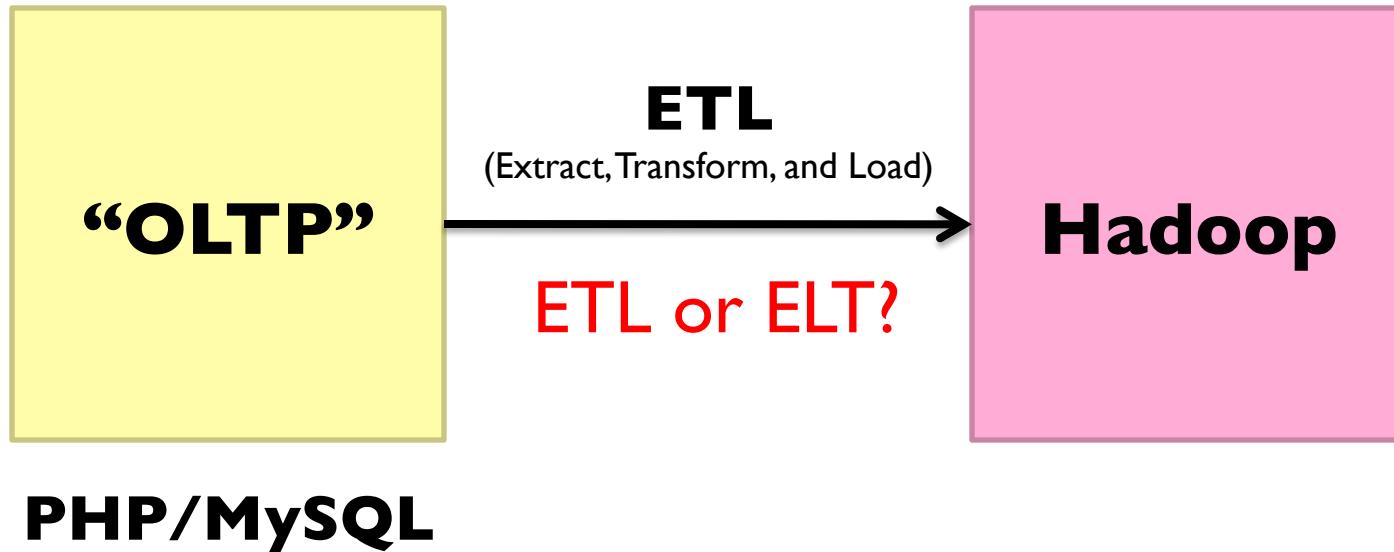


# **Facebook Technology**

**“OLTP”**

**PHP/MySQL**

# Facebook's Datawarehouse



# What's changed?

- Dropping cost of disks
  - Cheaper to store everything than to figure out what to throw away

# What's changed?

- Dropping cost of disks
  - Cheaper to store everything

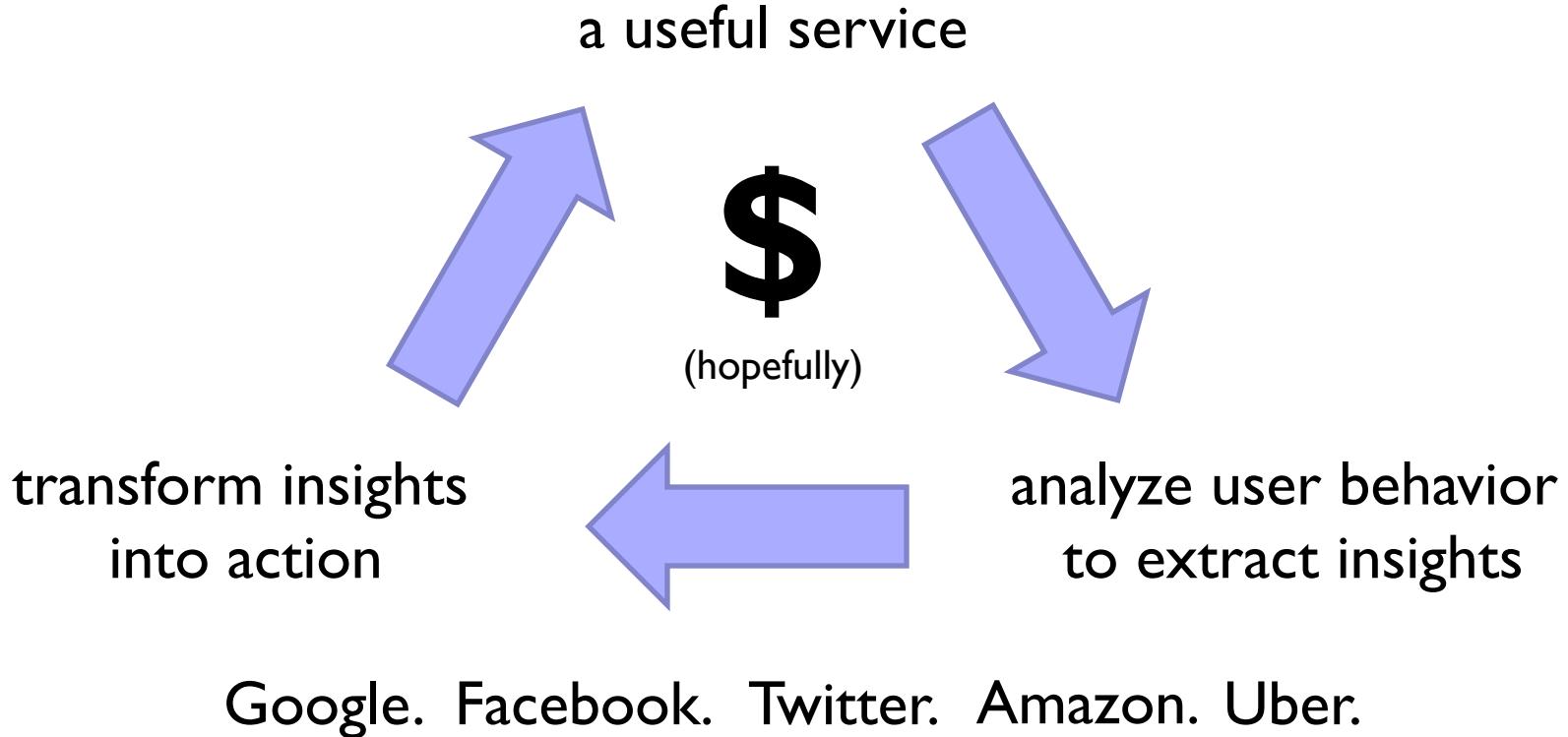


5 MB hard drive in 1956

# What's changed?

- Dropping cost of disks
  - Cheaper to store everything than to figure out what to throw away
- Types of data collected
  - From data that's *obviously* valuable to data whose value is less apparent
- Rise of social media and user-generated content
  - Large increase in data volume
- Growing maturity of data mining techniques
  - Demonstrates value of data analytics

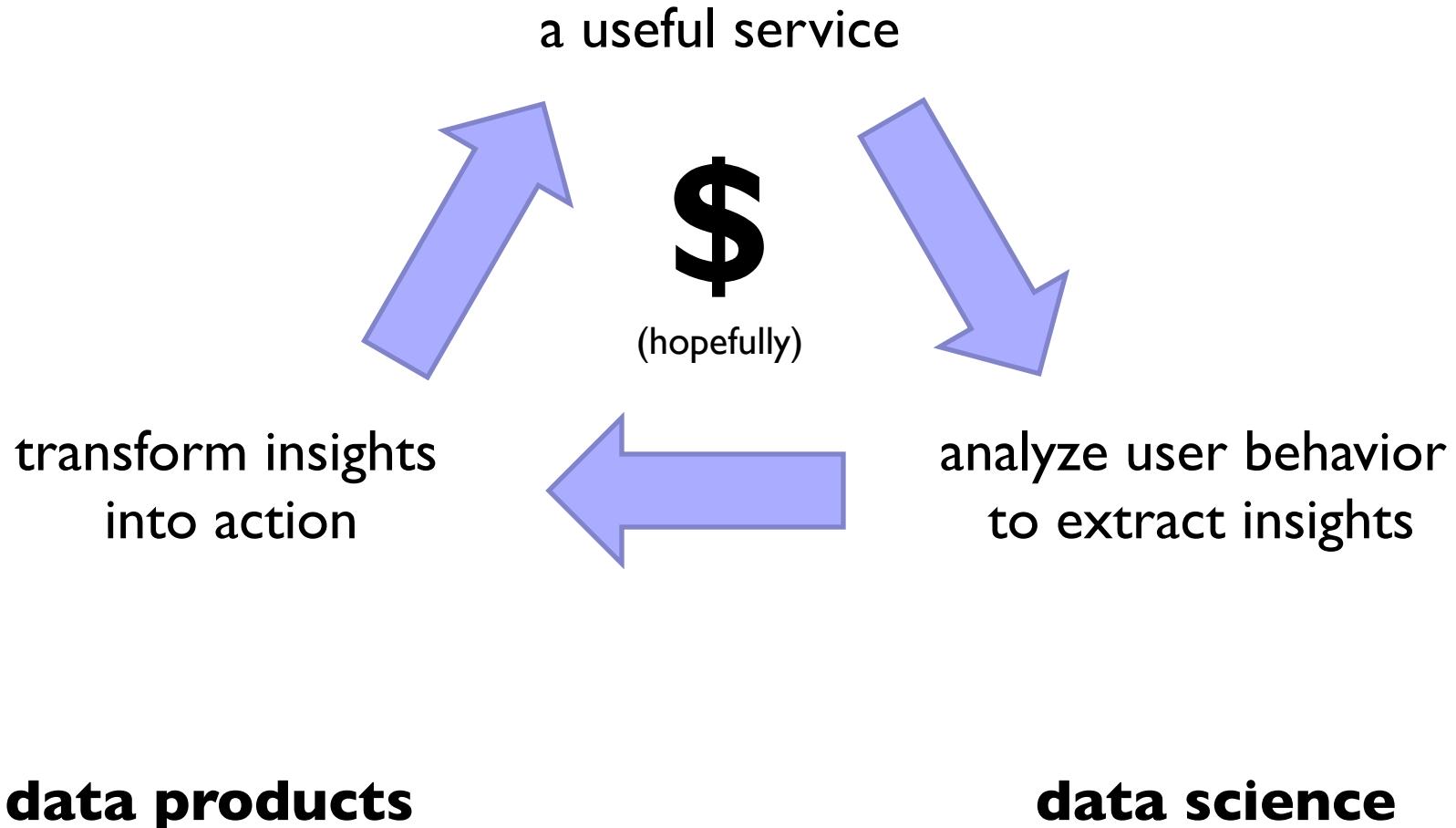
# Virtuous Product Cycle



# What do you actually do?

- Dashboards
- Report generation
- Ad hoc analyses
  - “Descriptive”
  - “Predictive”
- Data products

# Virtuous Product Cycle

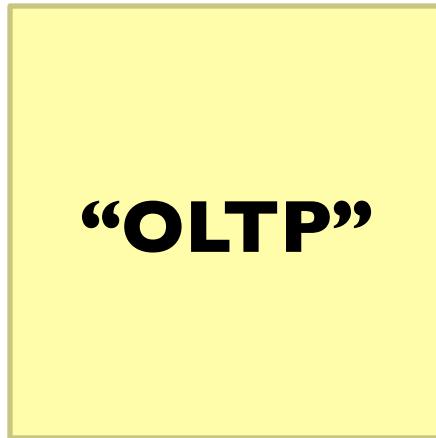




Jeff Hammerbacher, Information Platforms and the Rise of the Data Scientist.  
In, *Beautiful Data*, O'Reilly, 2009.

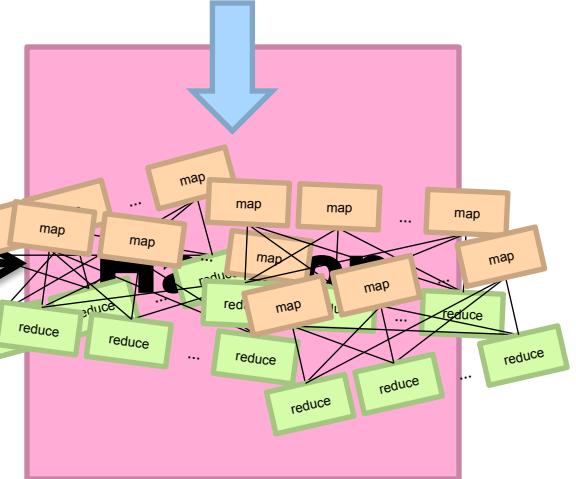
“On the first day of logging the Facebook clickstream, more than 400 gigabytes of data was collected. The load, index, and aggregation processes for this data set really taxed the Oracle data warehouse. Even after significant tuning, we were unable to aggregate a day of clickstream data in less than 24 hours.”

# The Irony...



**PHP/MySQL**

**ELT**



**Wait, so why not use a database to begin with?**

Why not just use a database?  
SQL is awesome

Scalability. Cost.

# Databases are great...

If your data has structure (and you know what the structure is)

If your data is reasonably clean

If you know what queries you're going to run ahead of time

# Databases are not so great...

If your data has little structure (or you don't know the structure)

If your data is messy and noisy

If you don't know what you're looking for



“there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are unknown unknowns – the ones we don't know we don't know...” – Donald Rumsfeld

# Databases are great...

If your data has structure (and you know what the structure is)

If your data is reasonably clean

If you know what queries you're going to run ahead of time

**Known unknowns!**

# Databases are not so great...

If your data has little structure (or you don't know the structure)

If your data is messy and noisy

If you don't know what you're looking for

**Unknown unknowns!**

# Advantages of Hadoop dataflow languages

Don't need to know the schema ahead of time

Raw scans are the most common operations

Many analyses are better formulated imperatively

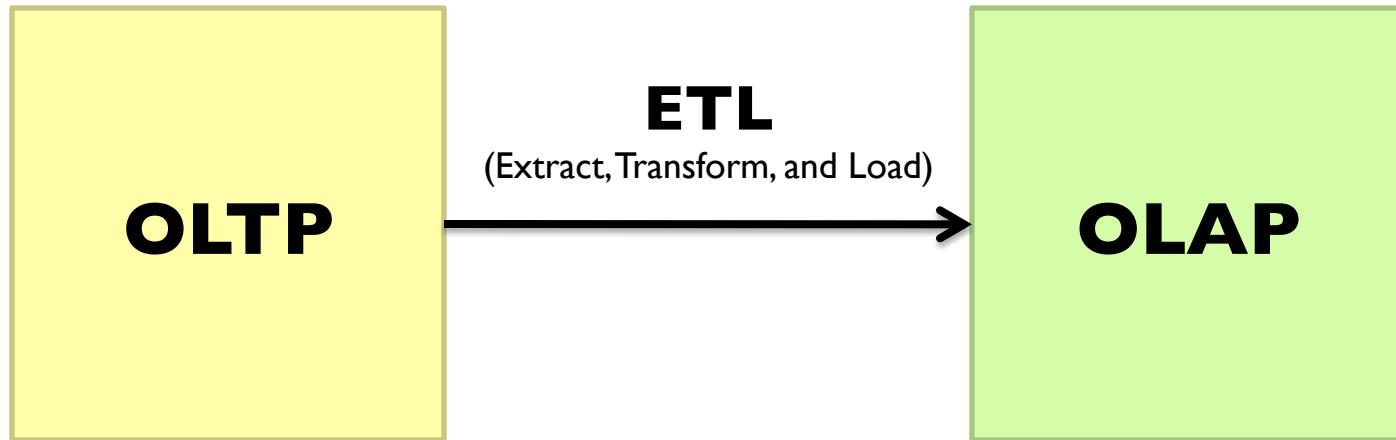
Also compare: data ingestion rate

# What do you actually do?

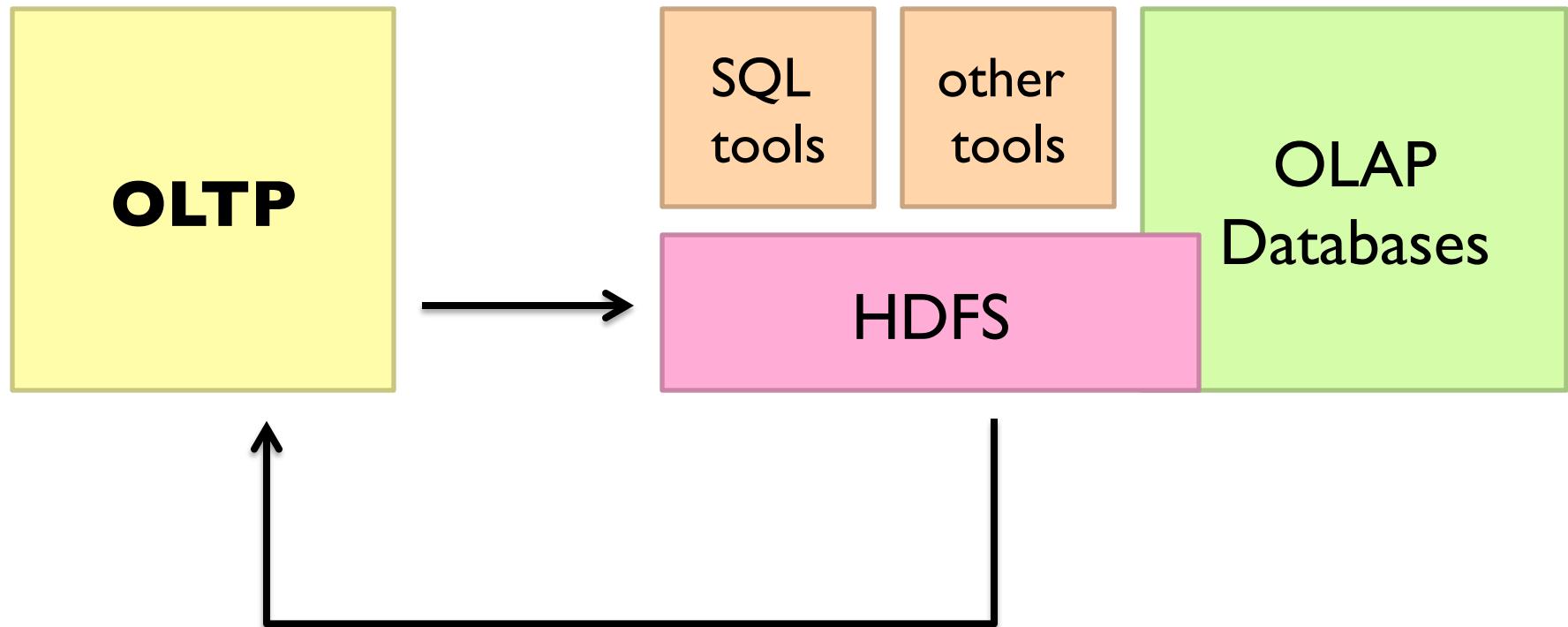
- Dashboards
- Report generation
- Ad hoc analyses
  - “Descriptive”
  - “Predictive”
- Data products

Which are known unknowns  
and unknown unknowns?

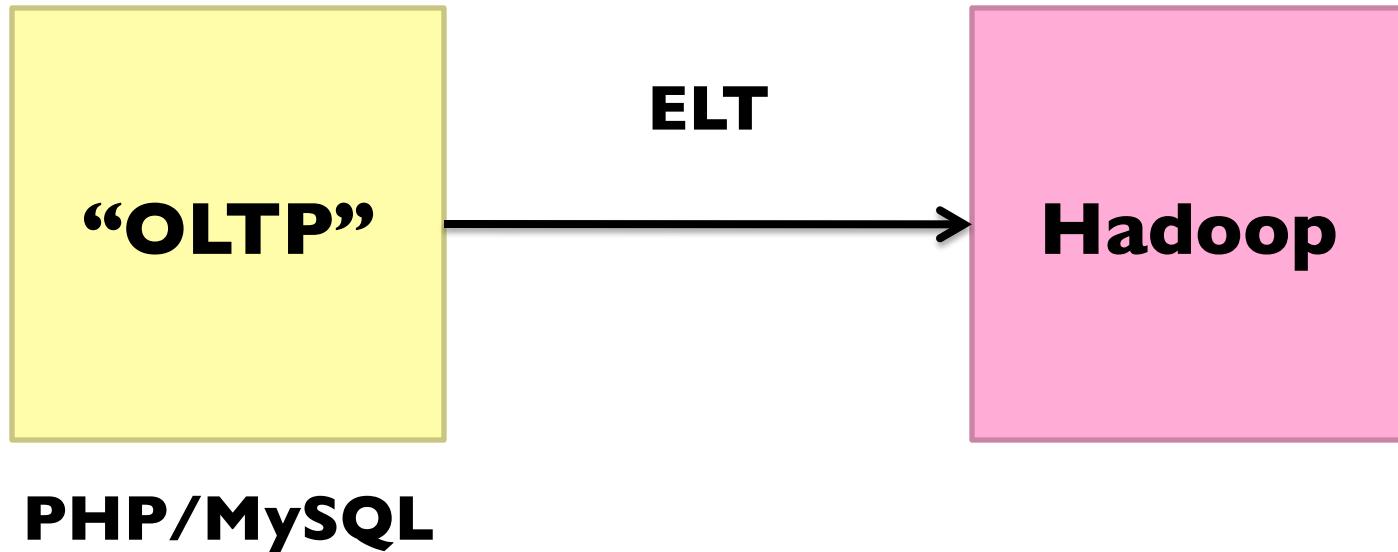
# **OLTP/OLAP Architecture**



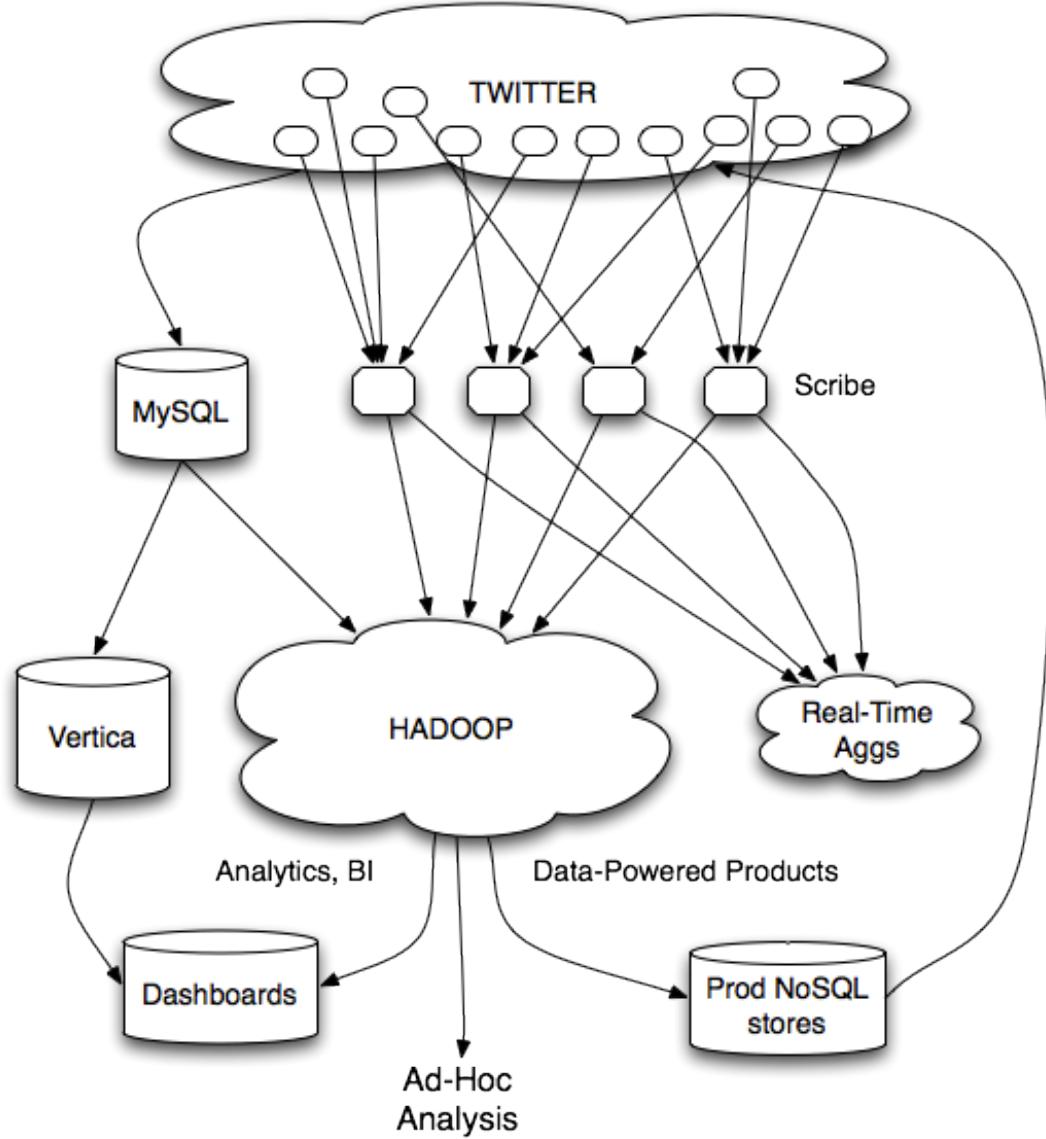
# Modern Datawarehouse Ecosystem



# Facebook's Datawarehouse



How does this actually happen?



# Twitter's data warehousing architecture

## **circa ~2010**

~150 people total

~60 Hadoop nodes

~6 people use analytics stack daily

## **circa ~2012**

~1400 people total

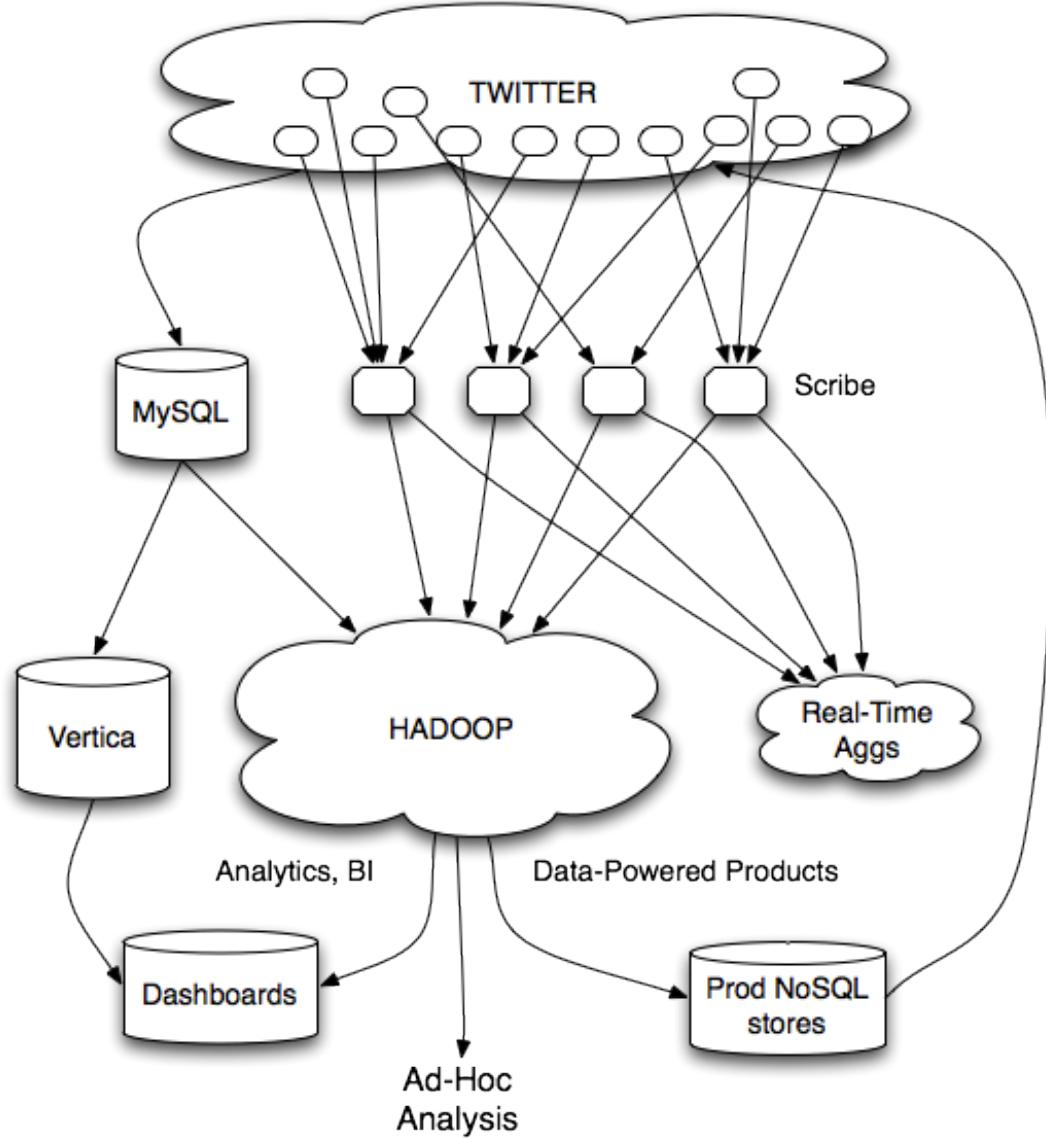
10s of Ks of Hadoop nodes, multiple DCs

10s of PBs total Hadoop DW capacity

~100 TB ingest daily

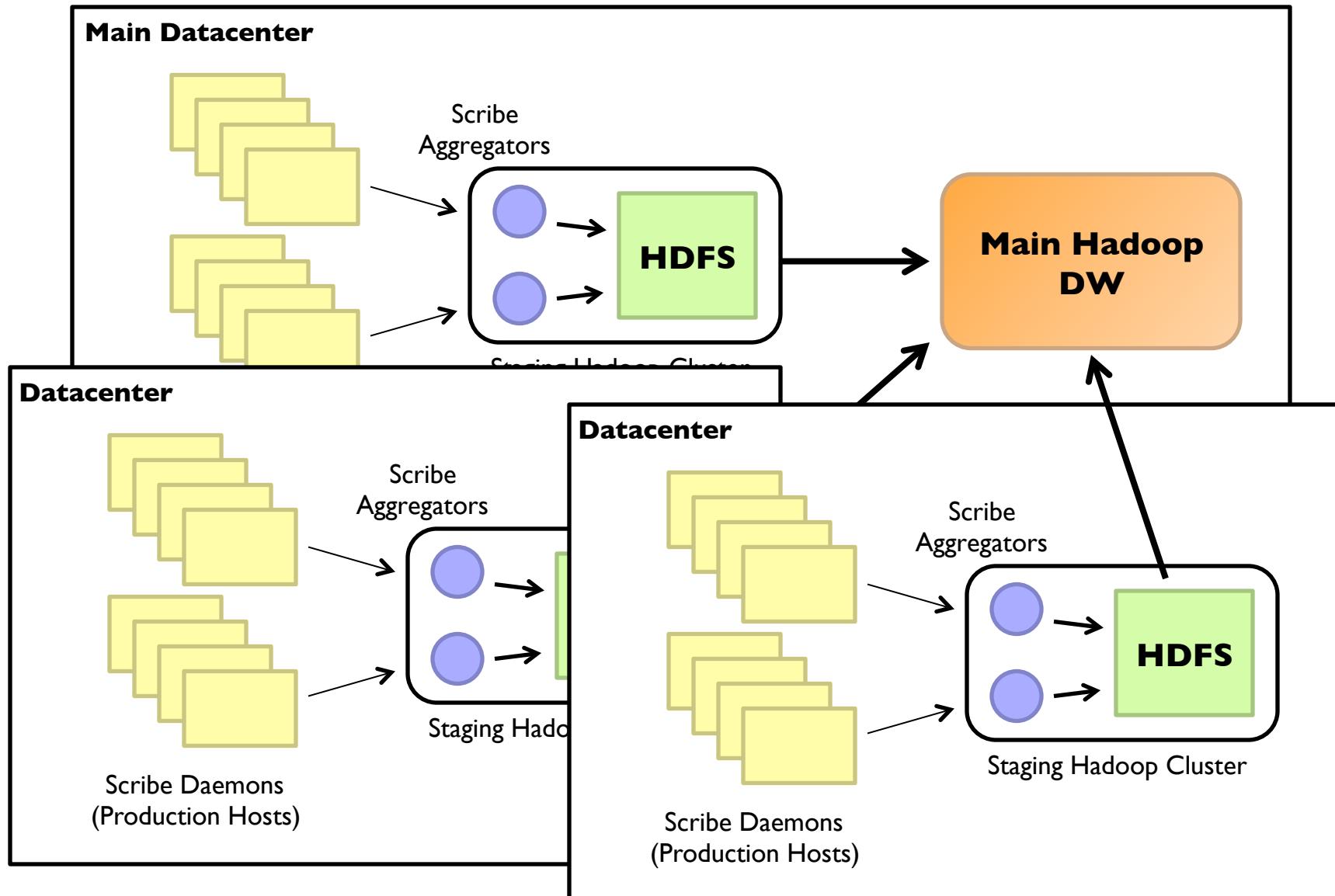
dozens of teams use Hadoop daily

10s of Ks of Hadoop jobs daily



# Twitter's data warehousing architecture

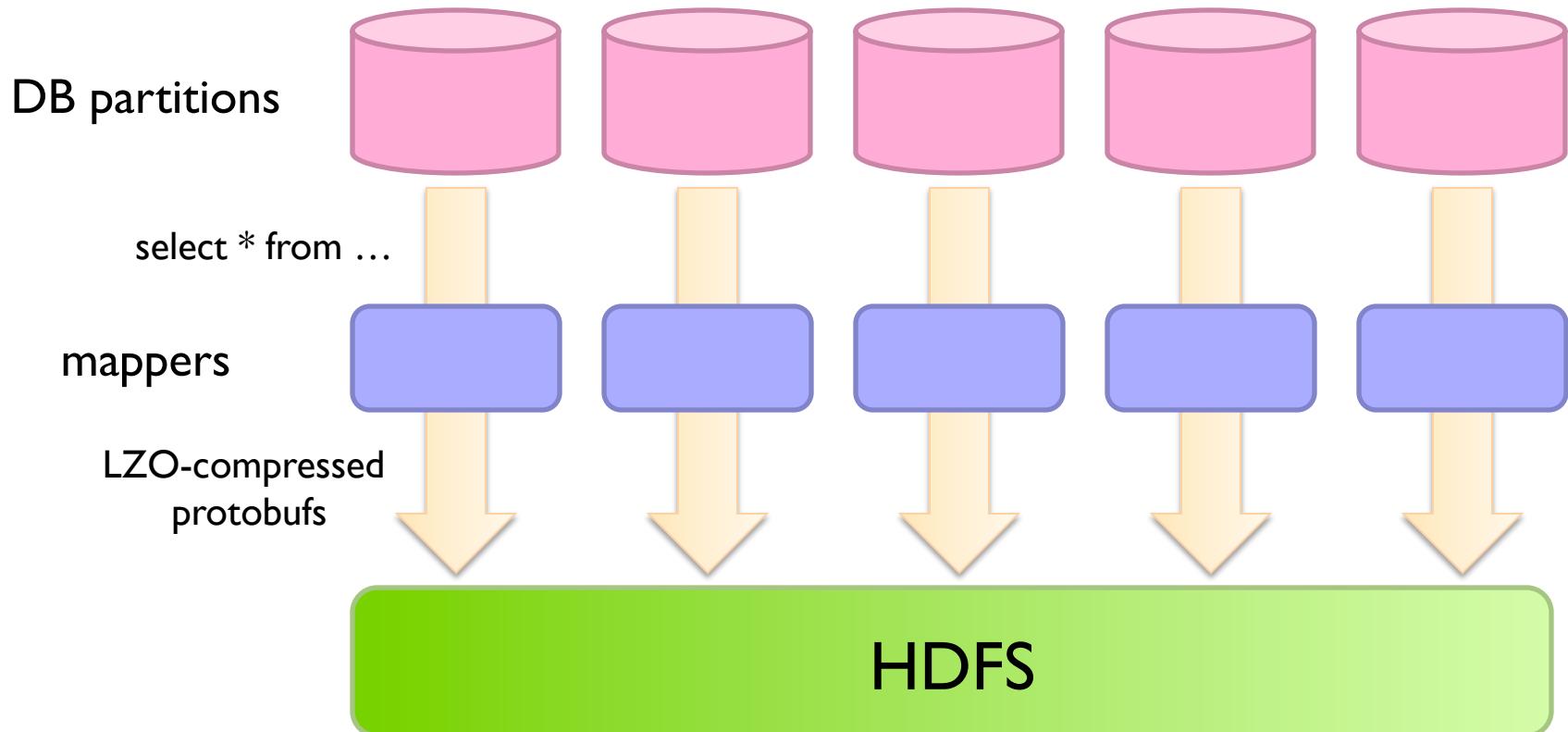
# Importing Log Data



# Importing Structured Data\*

Tweets, graph, users profiles

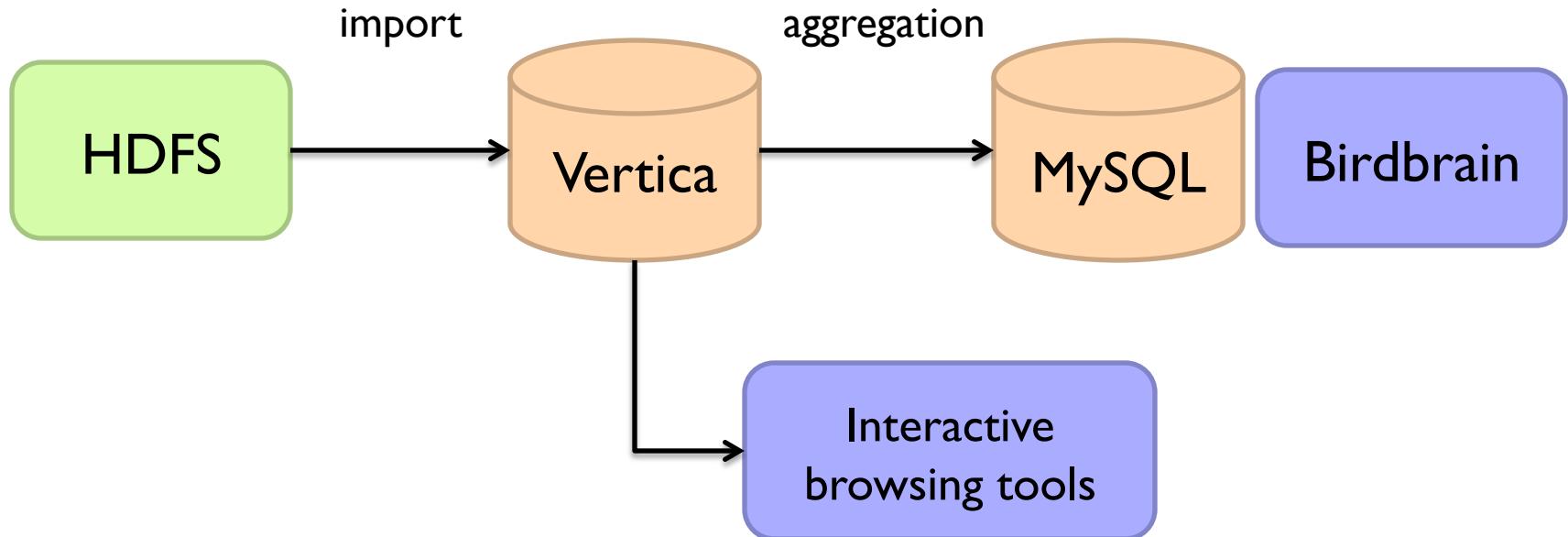
Different periodicity (e.g., hourly, daily snapshots, etc.)



**Important:** Must carefully throttle resource usage...

\* Out of date – for illustration only

# Vertica Pipeline



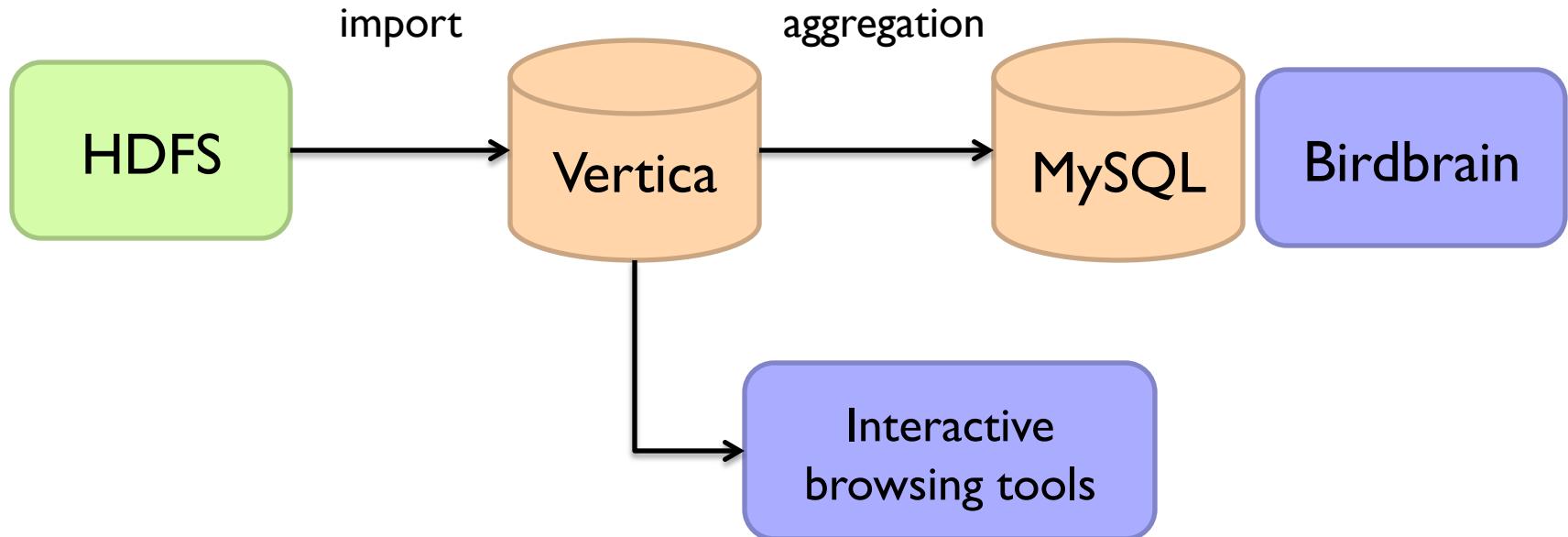
## Why?

Vertica provides *orders of magnitude* faster aggregations!

“Basically, we use Vertica as a cache for HDFS data.”

@squarecog

# Vertica Pipeline

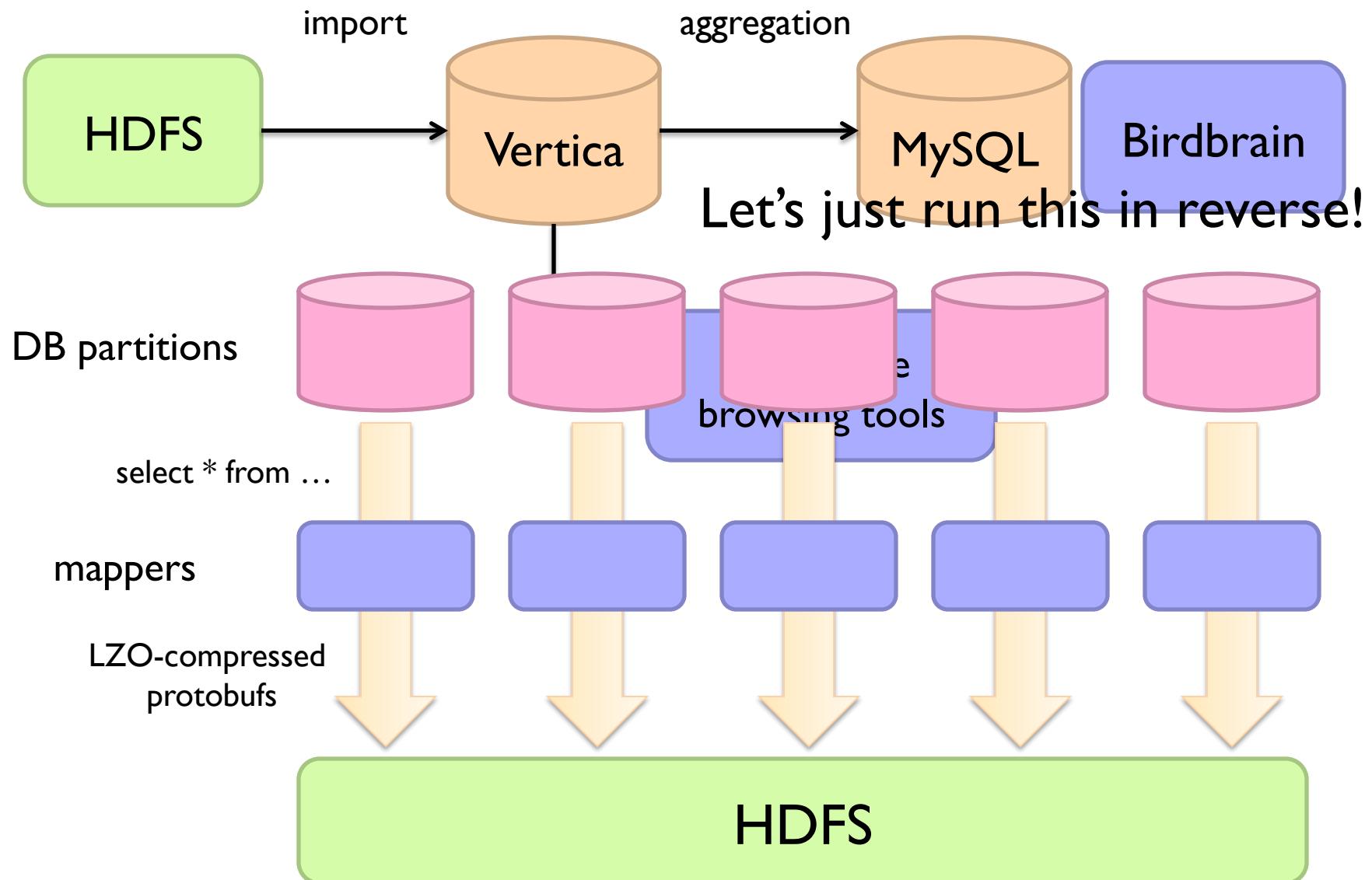


**The catch...**

Performance must be balanced against integration costs

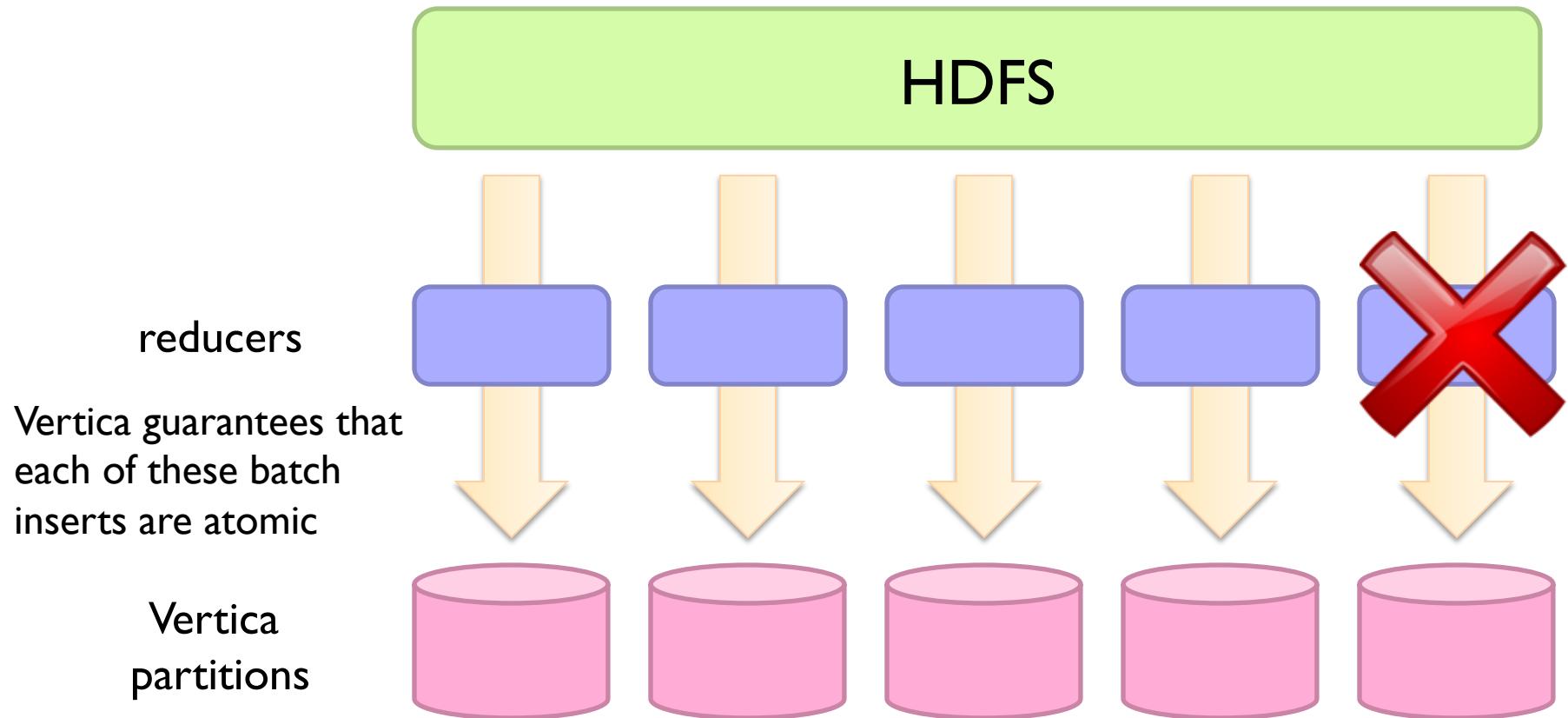
Vertica integration is non-trivial

# Vertica Pipeline



\* Out of date – for illustration only

# Vertica Pig Storage

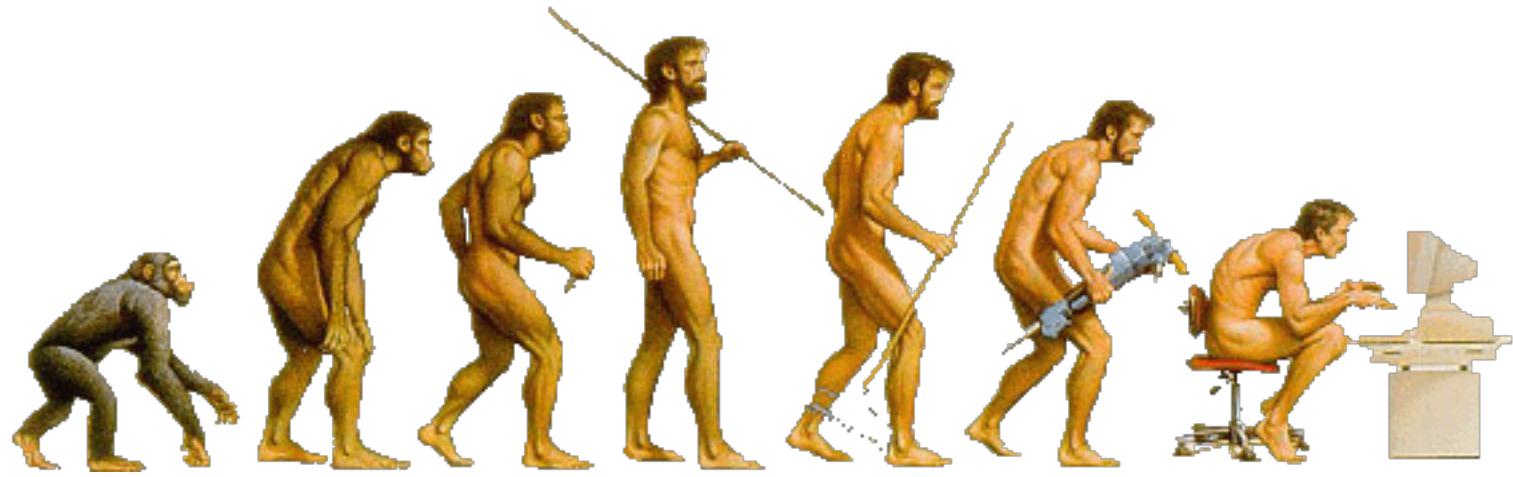


## So what's the challenge?

Did you remember to turn off speculative execution?

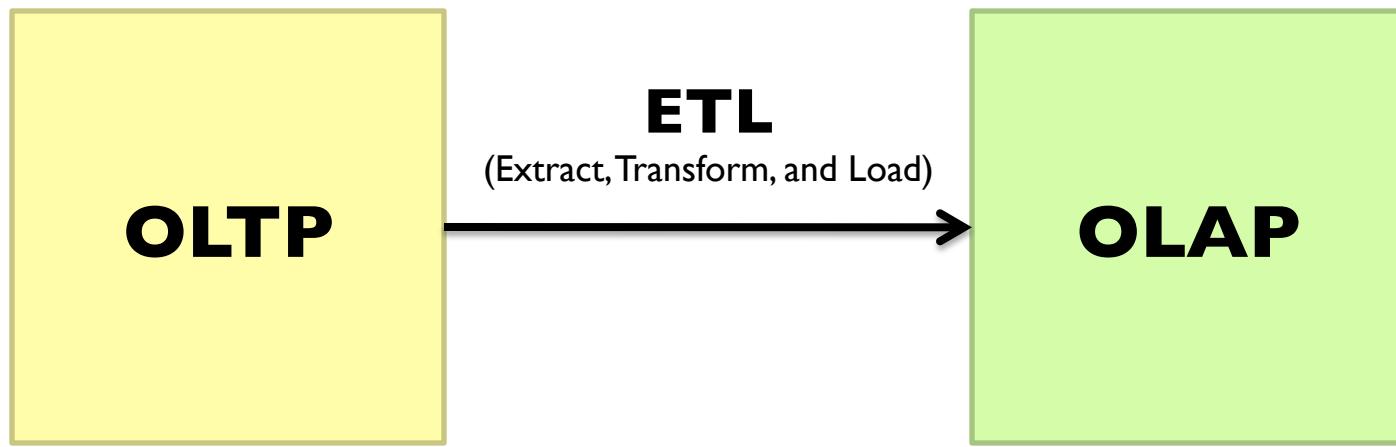
What happens when a task dies?

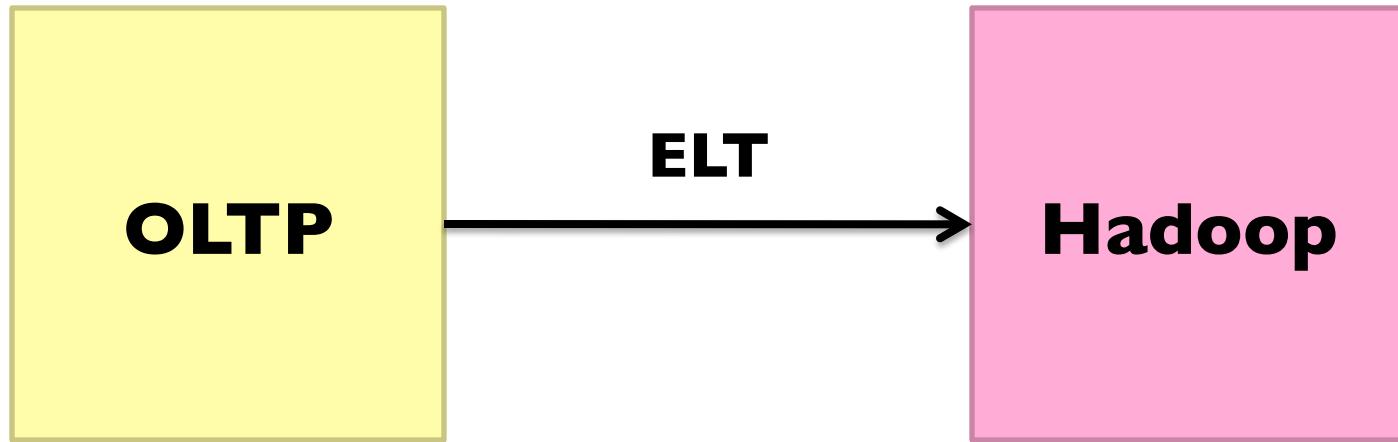
\* Out of date – for illustration only

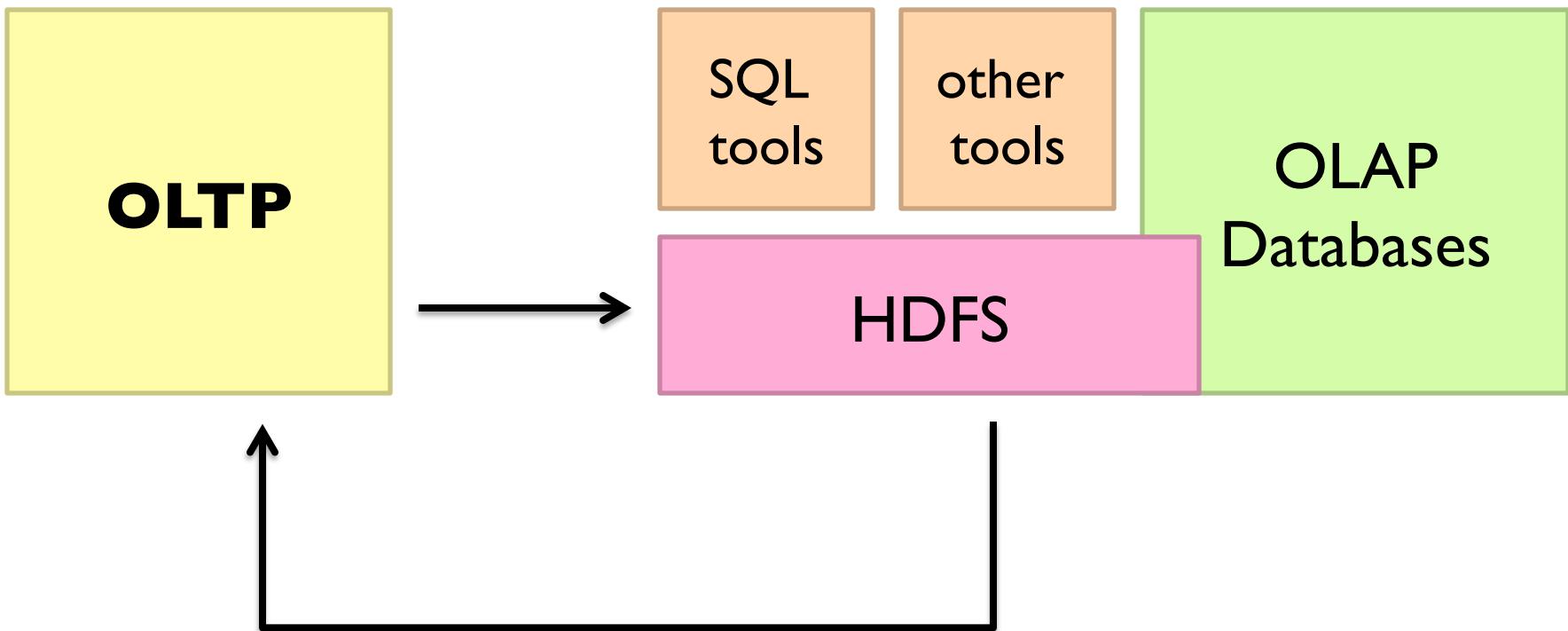


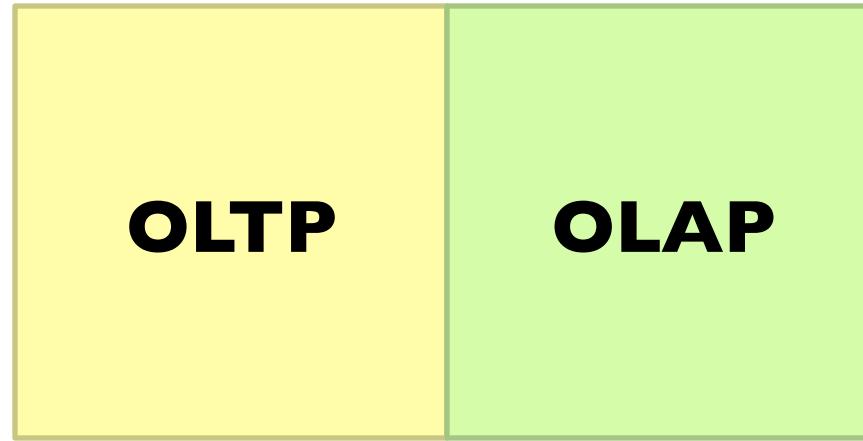
What's Next?

**RDBMS**









Hybrid Transactional/Analytical Processing (HTAP)

Coming back full circle?

A photograph of a traditional Japanese rock garden. In the foreground, a gravel path is raked into fine, parallel lines. Several large, dark, irregular stones are scattered across the garden. A small, shallow pond is nestled among rocks in the middle ground. The background features a variety of trees and shrubs, some with autumn-colored leaves, and traditional wooden buildings with tiled roofs.

# Questions?