NewSQL

Andy Pavlo

February 6, 2012

BROWN

Outline

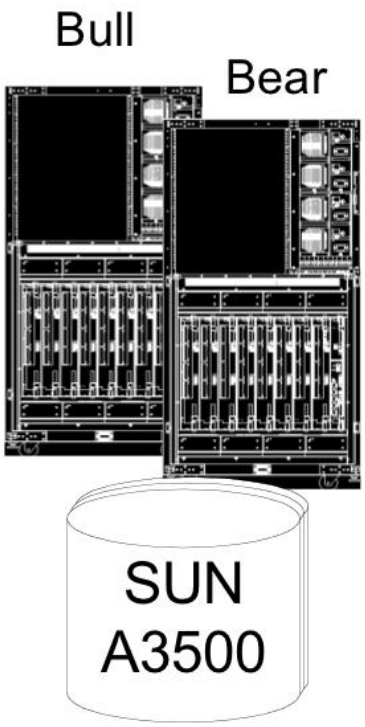
- The Last Decade of Databases
- NewSQL Introduction
- H-Store

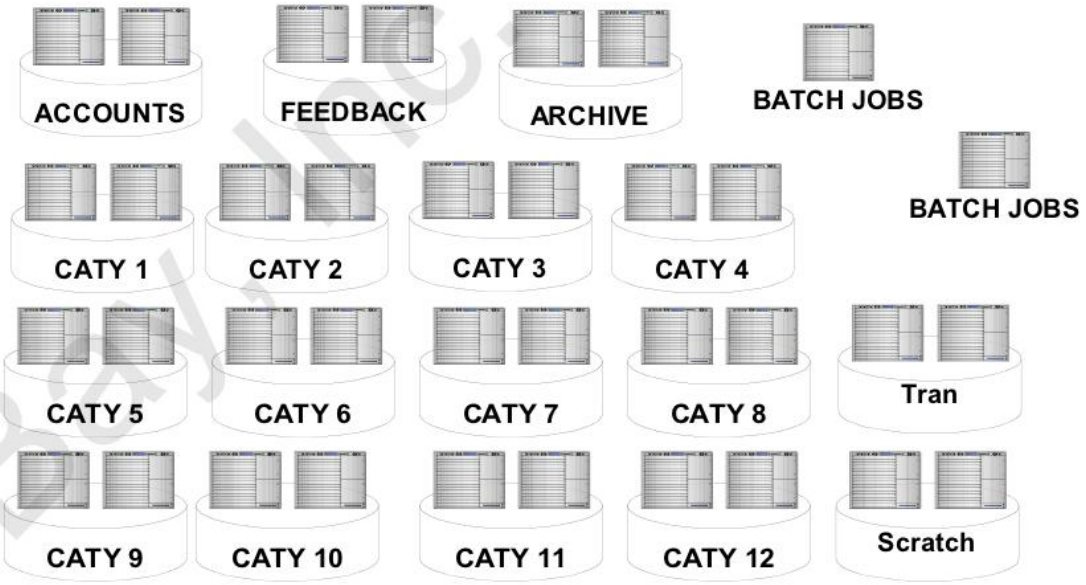
Early-2000s

- **All the big players were heavyweight and expensive.**
  - ***Oracle, DB2, Sybase, SQL Server, etc.***
- **Open-source databases were missing important features.**
  - ***Postgres, mSQL, and MySQL.***

V2.5 *April 2001 – December 2002*

- Horizontal scalability through database splits
- Items split by category
- SPOF elimination





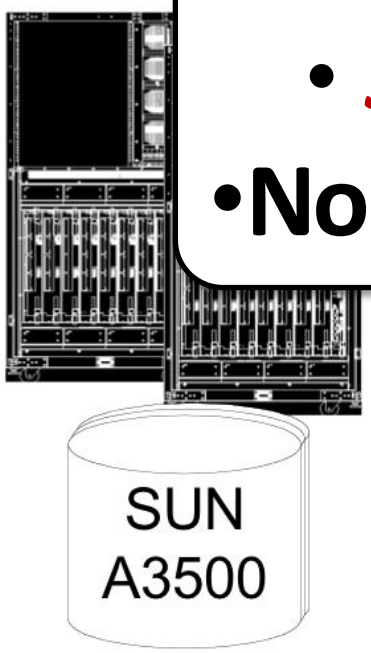
December, 2002

Randy Shoup - *"The eBay Architecture"*
http://highscalability.com/ebay-architecture

V2.5 *April 2001 – December 2002*

- Horizontal scalability through database splits
- Items split by category
- SPOF elimination

- **Push functionality to application:**
  - ***Joins***
  - ***Referential integrity***
  - ***Sorting done***
- **No distributed transactions.**



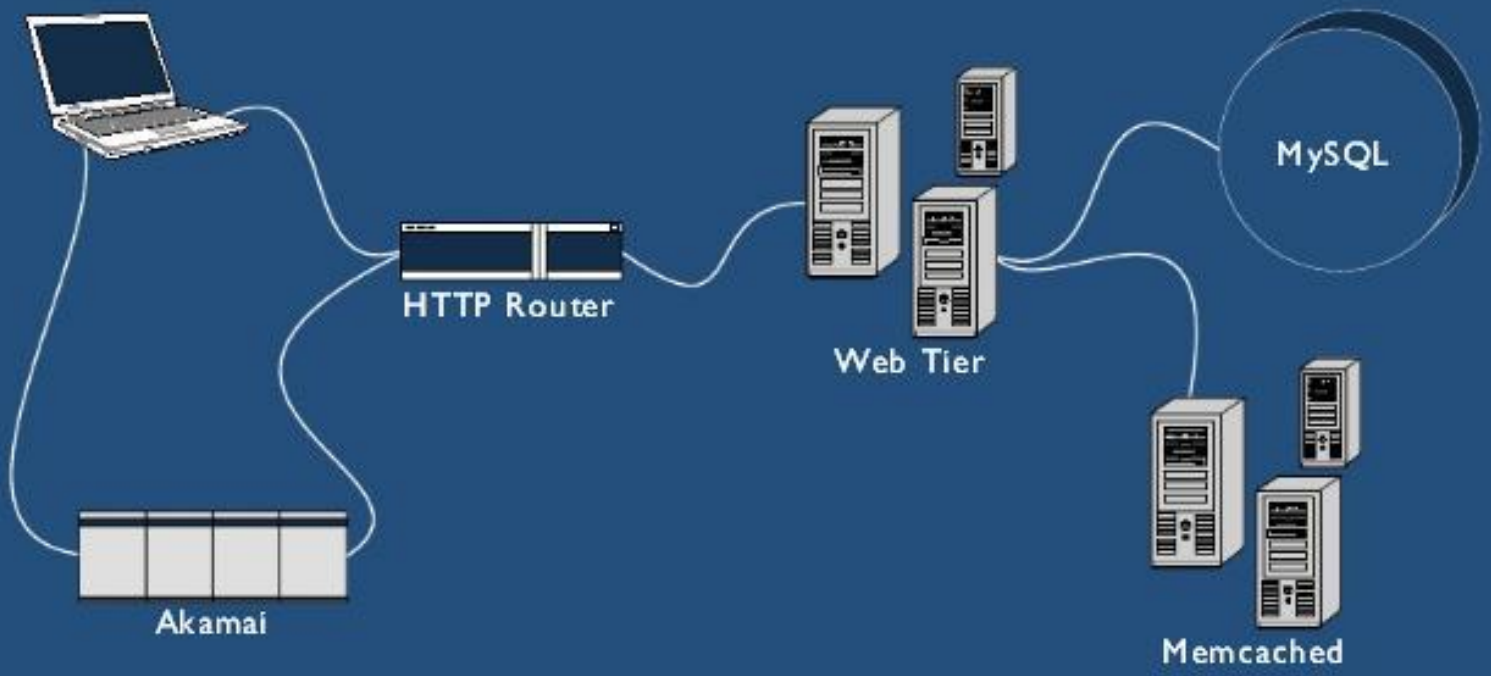
December, 2002

Randy Shoup - *"The eBay Architecture"*
http://highscalability.com/ebay-architecture

Mid-2000s

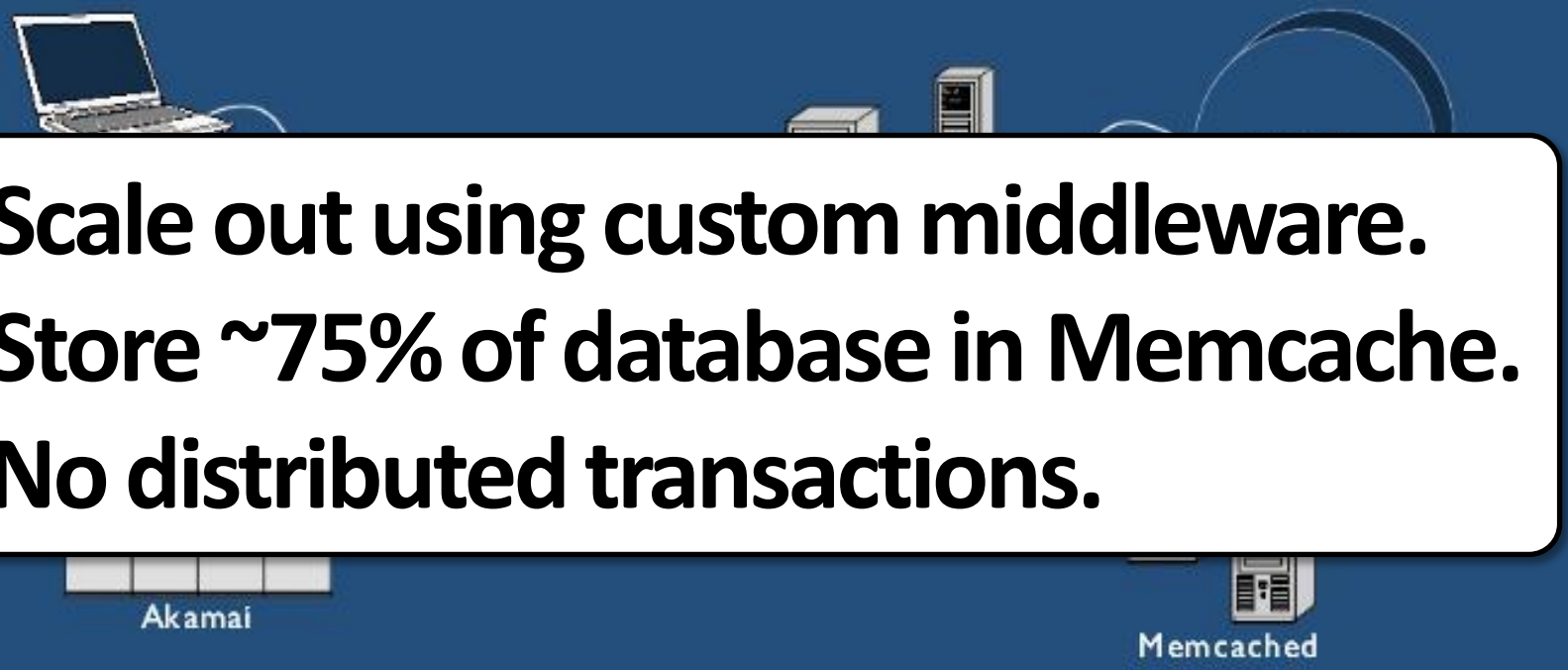
- **MySQL + InnoDB is widely adopted by new web companies:**
  - ***Supported transactions, replication, recovery.***
  - ***Still must use custom middleware to scale out across multiple machines.***
  - ***Memcache for caching queries.***

Facebook Architecture



Jay Thadeshwar - *"Technology Used by Facebook"*
http://www.techthebest.com/2011/11/29/technology-used-in-facebook/

Facebook Architecture



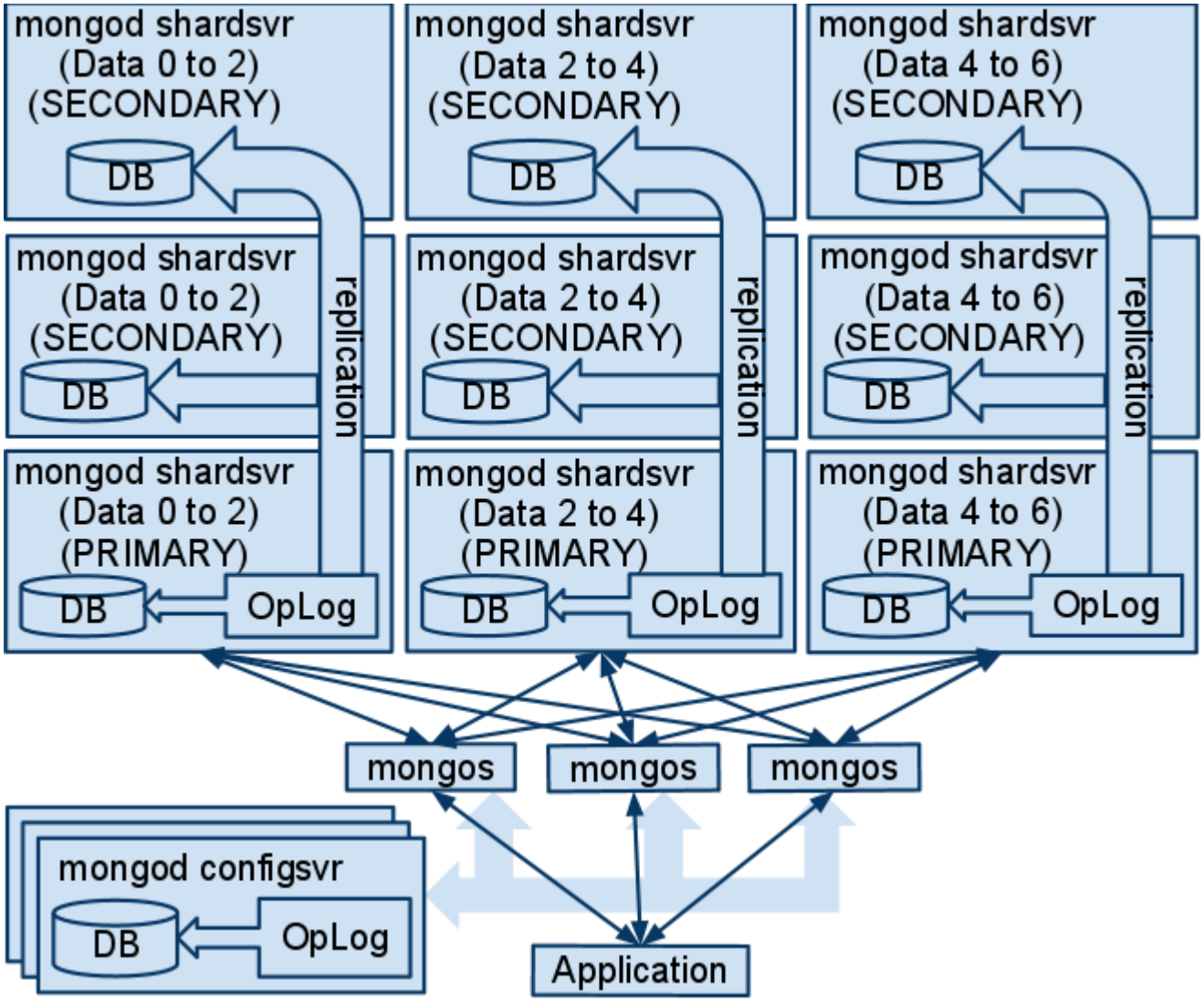
- Scale out using custom middleware.
- Store ~75% of database in Memcache.
- No distributed transactions.

Jay Thadeshwar - *"Technology Used by Facebook"*
http://www.techthebest.com/2011/11/29/technology-used-in-facebook/

Late-2000s

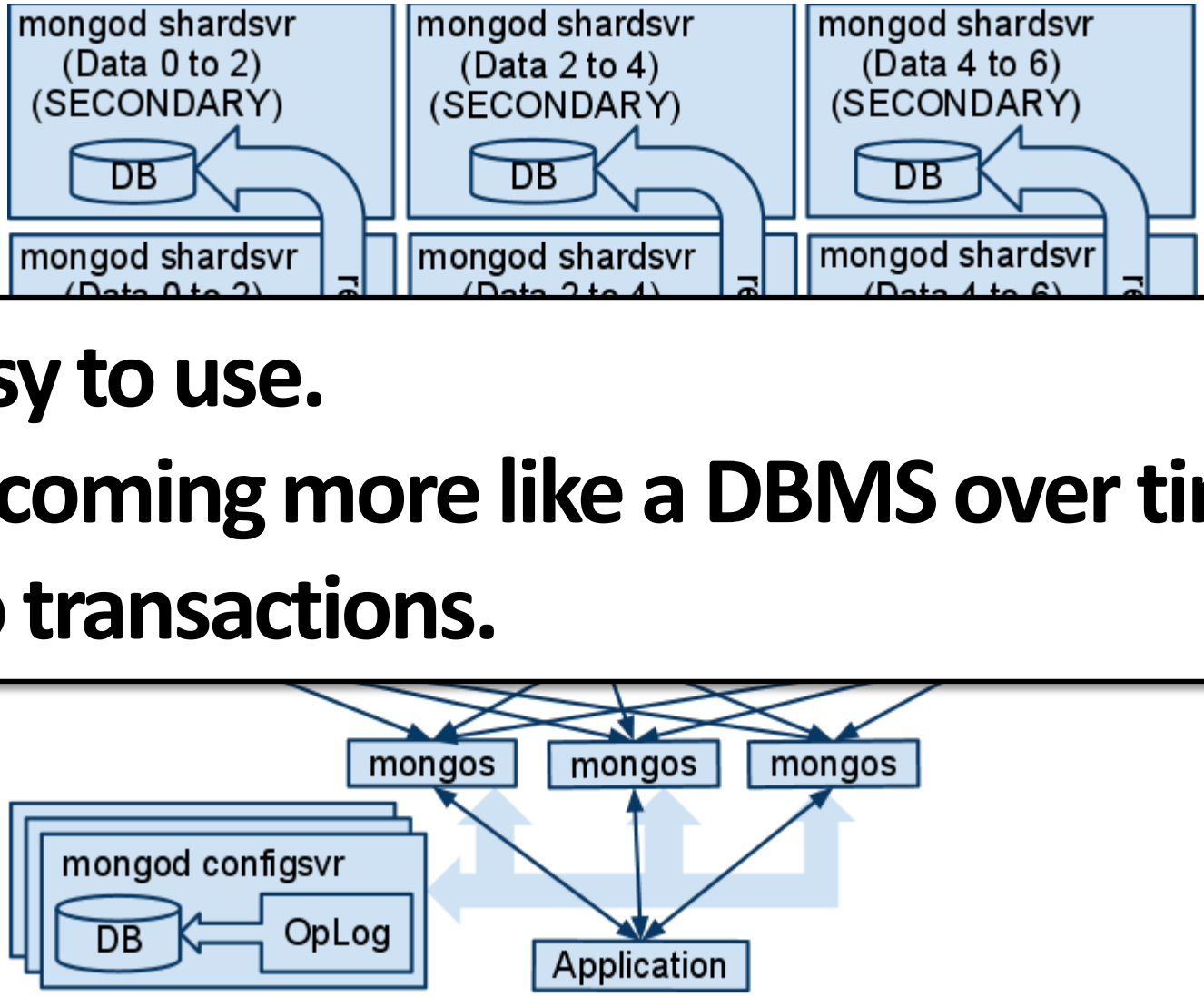
- **NoSQL systems are able to scale horizontally right out of the box:**
  - ***Schemaless.***
  - ***Using custom APIs instead of SQL.***
  - ***Not ACID (i.e., eventual consistency)***
  - ***Many are based on Google's BigTable or Amazon's Dynamo systems.***

MongoDB Architecture



Nathan Tippy- *"MongoDB"*
http://sett.ociweb.com/sett/settAug2011.html

MongoDB Architecture



- Easy to use.
- Becoming more like a DBMS over time.
- No transactions.

Nathan Tippy- *"MongoDB"*
http://sett.ociweb.com/sett/settAug2011.html

Early-2010s

- **New DBMSs that can scale across multiple machines natively and provide ACID guarantees.**
  - ***MySQL Middleware***
  - ***Brand New Architectures***


SCALEBASE
Tokutek
dbShards
ScaleArc
AKIBAN
ScaleDB
SCHOONER
NewSQL
JustOneDB
TRANSLATTICE
VoltDB
nuoDB
SQLFire
H-Store
GENIEDB
xeround
Clustrix

451 Group's Definition

- **A DBMS that delivers the scalability and flexibility promised by NoSQL while retaining the support for SQL queries and/or ACID, or to improve performance for appropriate workloads.**

Matt Aslett – *"How Will The Database Incumbents Respond To NoSQL And NewSQL?"*
https://www.451research.com/report-short?entityId=66963

Stonebraker's Definition

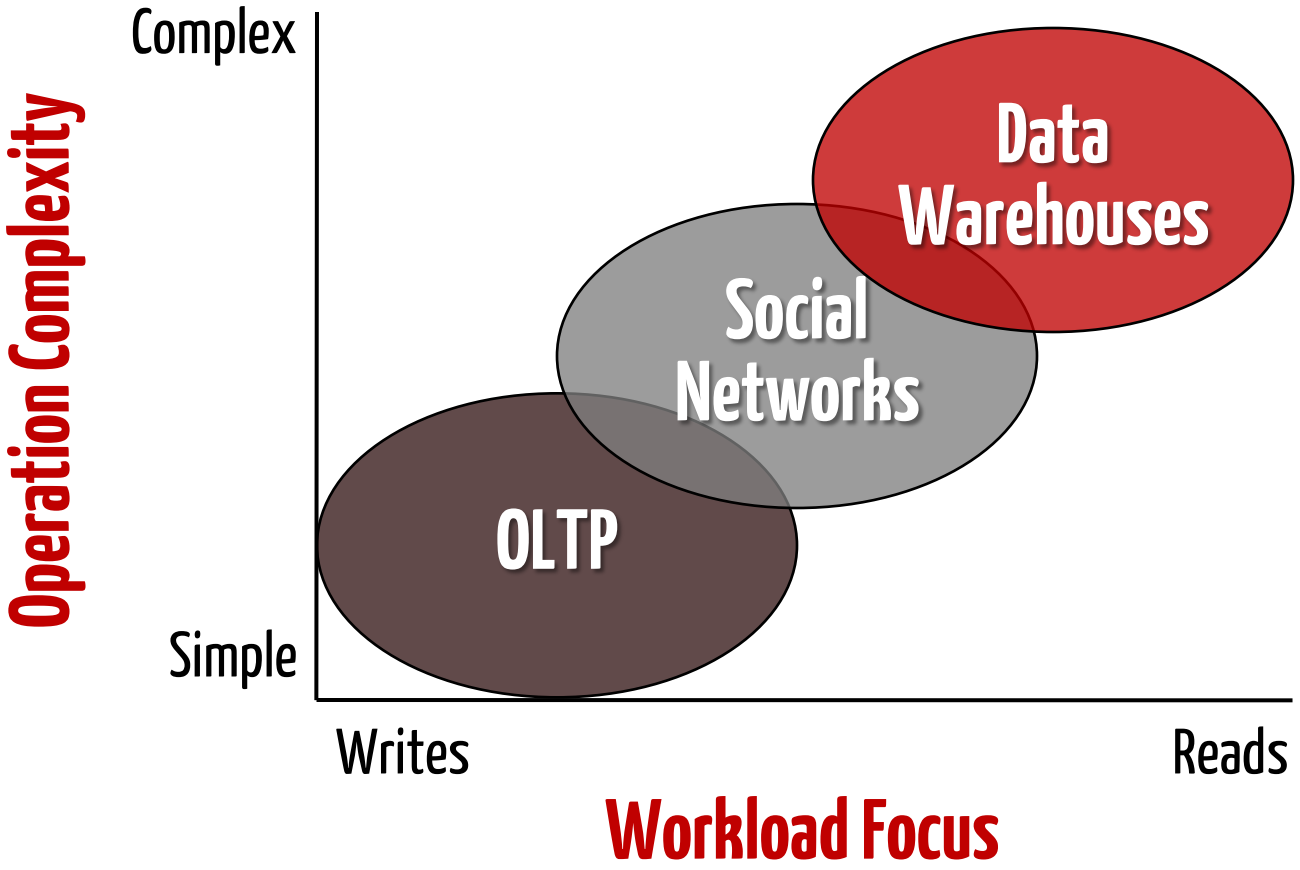
- SQL as the primary interface.
- ACID support for transactions
- Non-locking concurrency control.
- High per-node performance.
- Parallel, shared-nothing architecture.

On-Line Transaction Processing

OLTP Transactions

Fast **Repetitive** **Small**

Workload Characterization



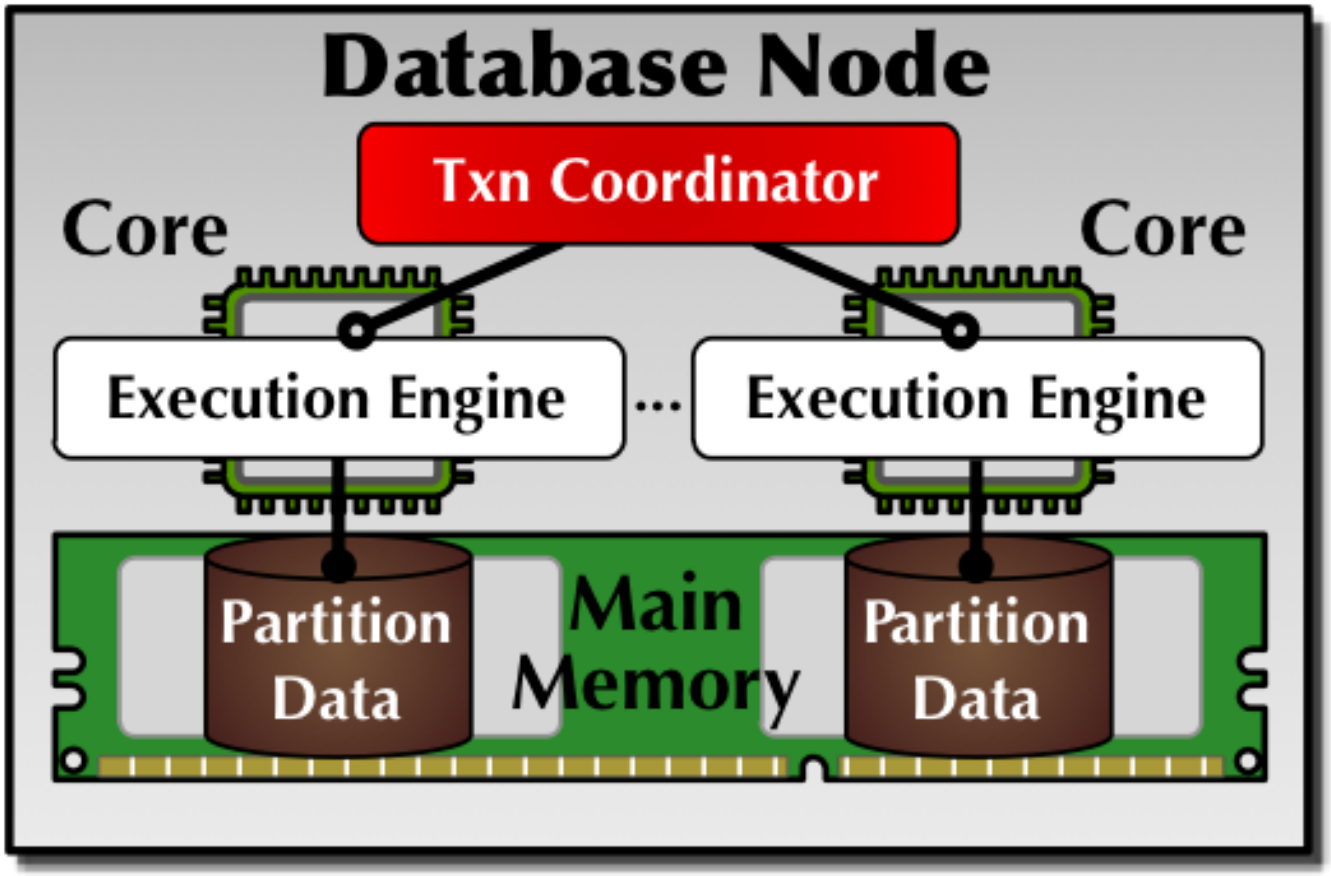
Transaction Bottlenecks

- Disk Reads/Writes
  - *Persistent Data, Undo/Redo Logs*
- Network Communication
  - *Intra-Node, Client-Server*
- Concurrency Control
  - *Locking, Latching*

An Ideal OLTP System

- Main Memory Only
- No Multi-processor Overhead
- High Scalability
- High Availability
- Autonomic Configuration

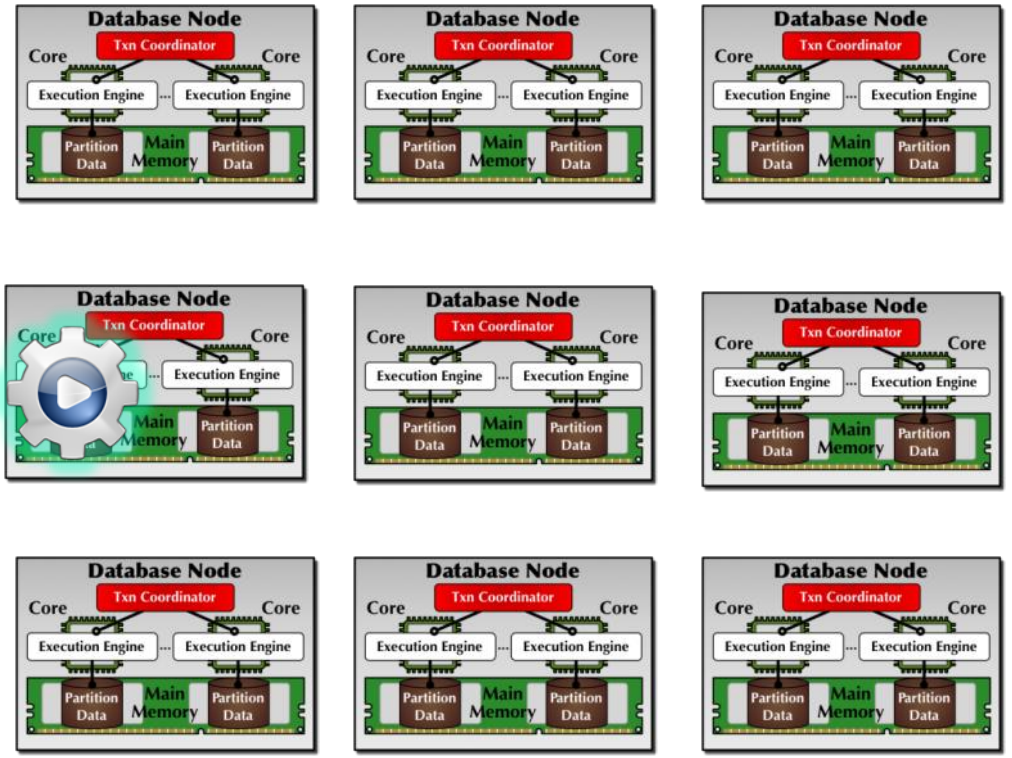
H-Store

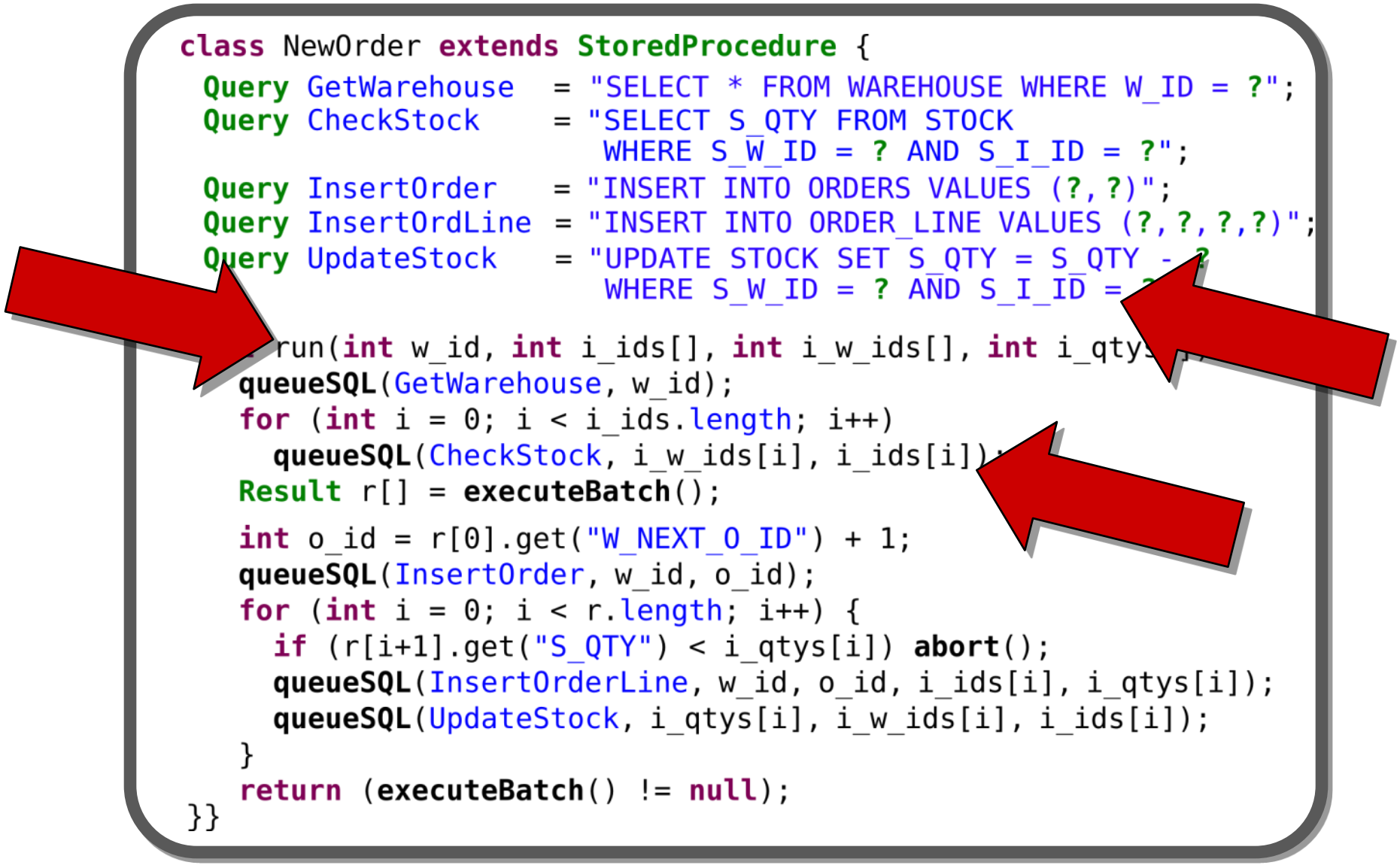

Database Node
Txn Coordinator
Core
Core
Execution Engine
...
Execution Engine
Partition Data
Main Memory
Partition Data

H-Store

Client Application

Procedure Name
Input Parameters

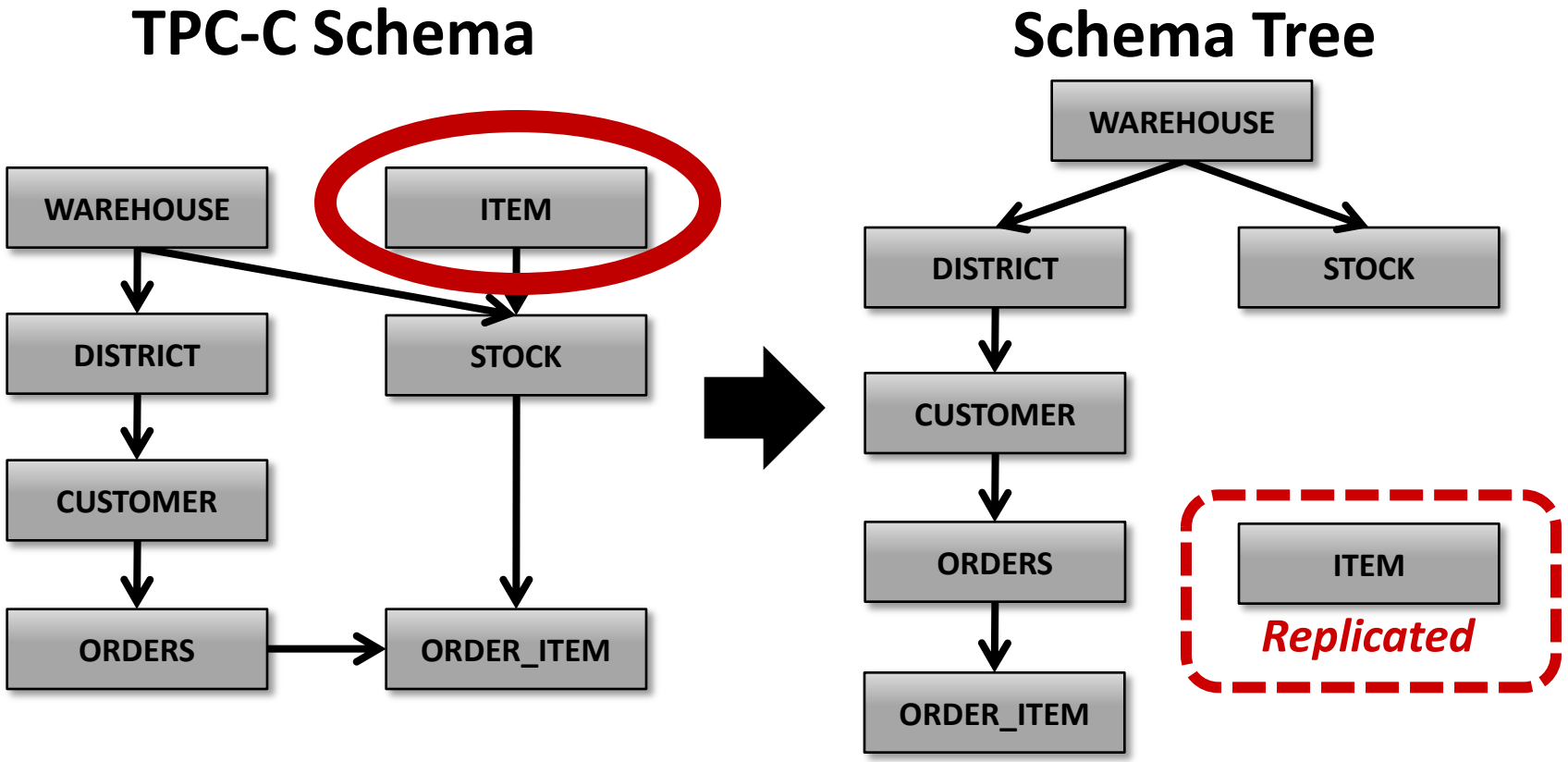




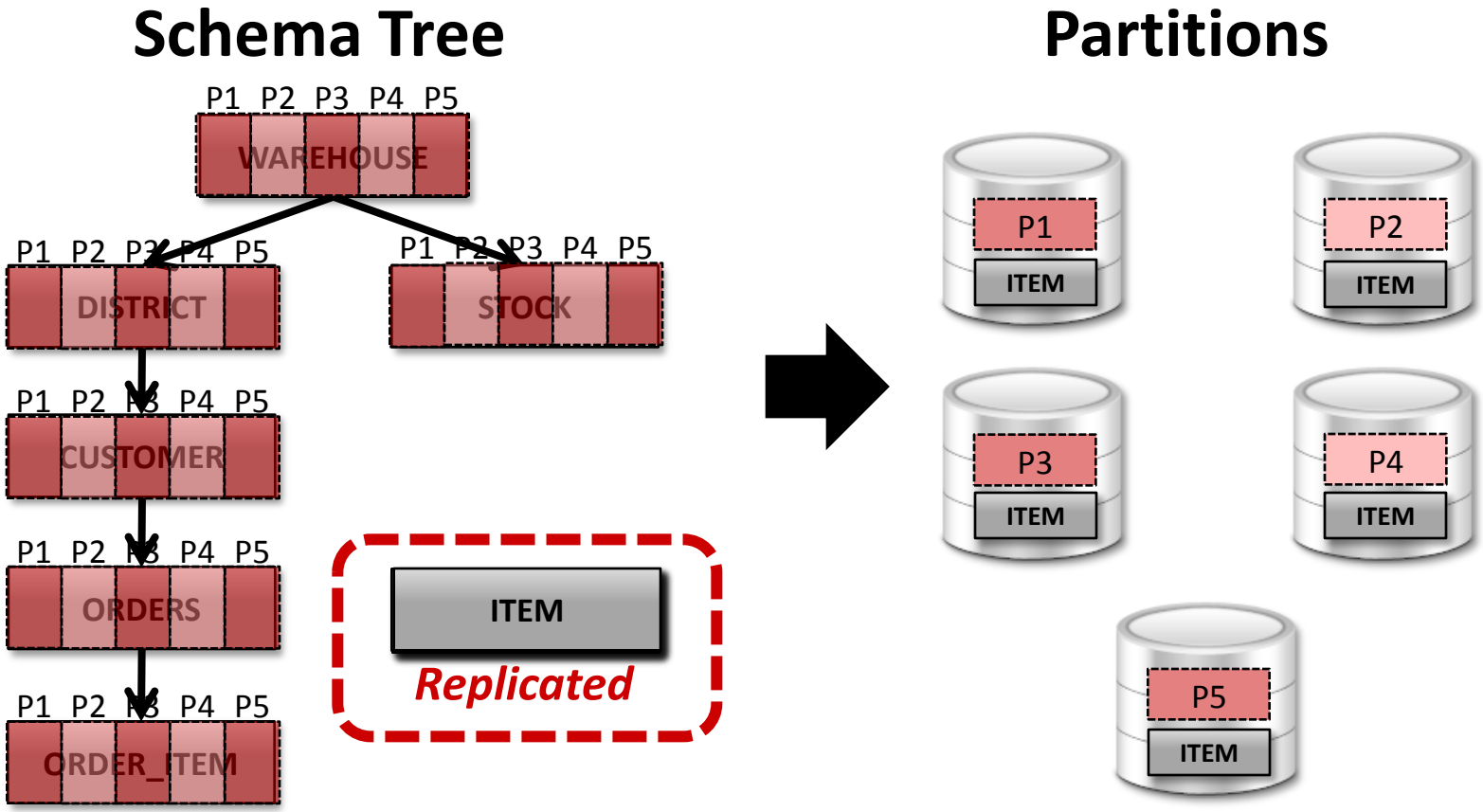
```
class NewOrder extends StoredProcedure {
  Query GetWarehouse  = "SELECT * FROM WAREHOUSE WHERE W_ID = ?";
  Query CheckStock    = "SELECT S_QTY FROM STOCK
                         WHERE S_W_ID = ? AND S_I_ID = ?";
  Query InsertOrder   = "INSERT INTO ORDERS VALUES (?,?)";
  Query InsertOrdLine = "INSERT INTO ORDER_LINE VALUES (?,?,?,?)";
  Query UpdateStock   = "UPDATE STOCK SET S_QTY = S_QTY - ?
                         WHERE S_W_ID = ? AND S_I_ID = ?";

  run(int w_id, int i_ids[], int i_w_ids[], int i_qtys[]) {
    queueSQL(GetWarehouse, w_id);
    for (int i = 0; i < i_ids.length; i++)
      queueSQL(CheckStock, i_w_ids[i], i_ids[i]);
    Result r[] = executeBatch();
    int o_id = r[0].get("W_NEXT_O_ID") + 1;
    queueSQL(InsertOrder, w_id, o_id);
    for (int i = 0; i < r.length; i++) {
      if (r[i+1].get("S_QTY") < i_qtys[i]) abort();
      queueSQL(InsertOrderLine, w_id, o_id, i_ids[i], i_qtys[i]);
      queueSQL(UpdateStock, i_qtys[i], i_w_ids[i], i_ids[i]);
    }
    return (executeBatch() != null);
}}
```

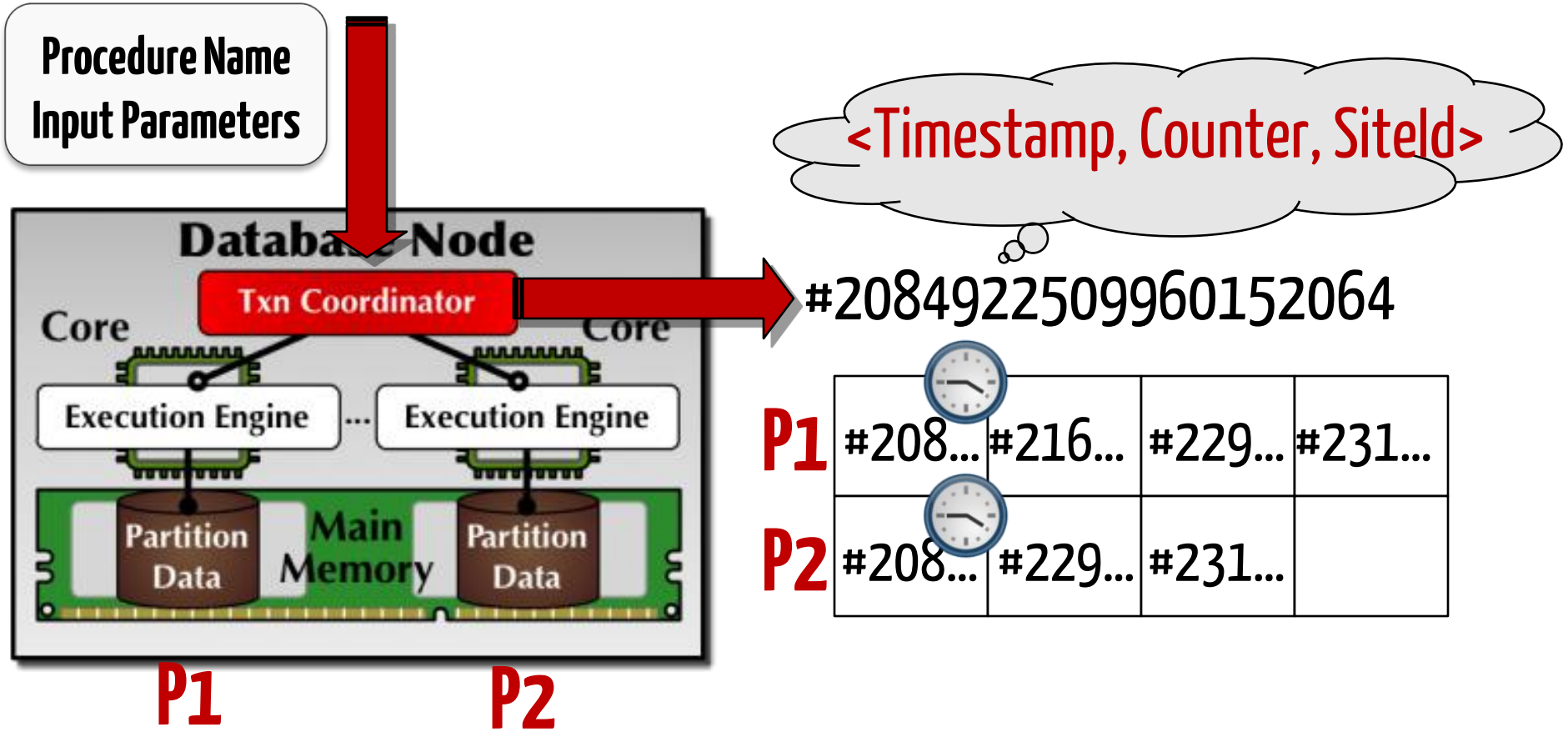
Database Partitioning



Database Partitioning



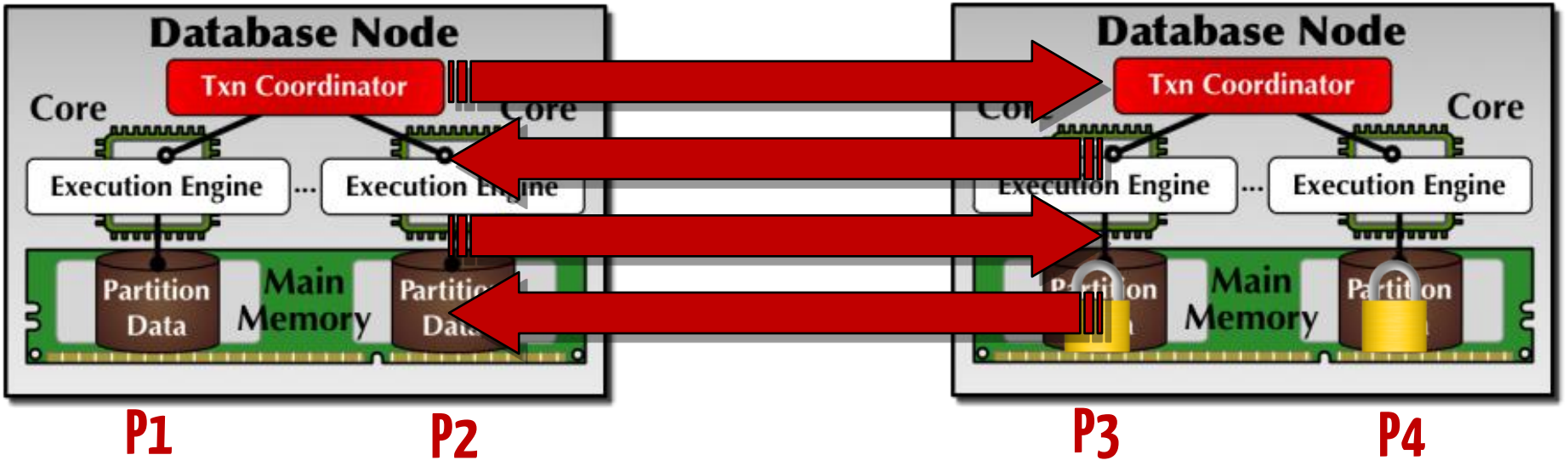
Distributed Transaction Protocol



Distributed Transaction Protocol

Two-Phase Commit

- TransactionPrepare Request
- TransactionPrepare Response
- TransactionFinish Request
- TransactionFinish Response



H-Store vs. VoltDB

- **An incestuous past**
  - *H-Store merged with Horizontica (Spring 2008)*
  - *VoltDB forked from H-Store (Fall 2008)*
  - *H-Store forked back from VoltDB (Winter 2009)*
- **Major differences:**
  - *Support for arbitrary transactions.*
  - *Google Protocol Buffer Network Communication*