Skew-Aware Automatic Database Partitioning in Shared-Nothing, Parallel OLTP Systems

SIGMOD 2012, Pavlo et al.

Hefu Chai

Credit

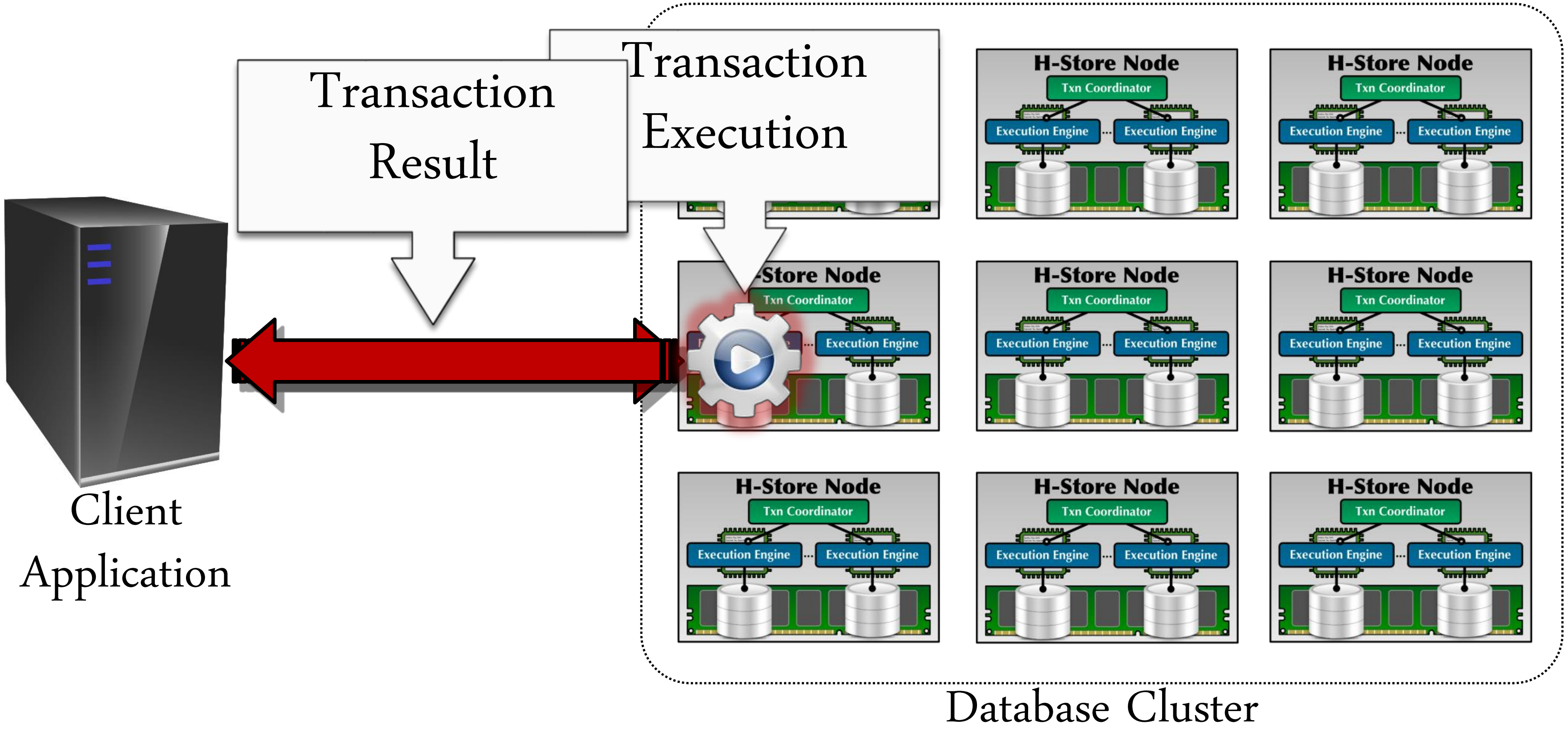
- Part of slides from Andy Pavlo

There is a saying…

- Girls are really only interested in two things. They want a guy that is good looking, or they want a guy that really knows a lot about databases.

Andy Pavlo

H-Store


Transaction Result
Transaction Execution
H-Store Node
Txn Coordinator
Execution Engine
Execution Engine
Client Application
Database Cluster

Existing database partitioning Techniques

- Notion of data declustering
  - Overhead of maintaining transaction consistency
  - Lock contention

Not applicable to OLTP systems !

H-Store

OLTP Transactions

Fast

Repetitive

Small

We need an approach that supports…

- Stored Procedure
- Load balancing in the presence of time-varying skew
- Complex schemas
- Deployments with larger number of partitions

Horticulture

Automatic Database Design Tool for Parallel Systems

Skew-Aware Automatic Database Partitioning
in Shared-Nothing, Parallel OLTP Systems

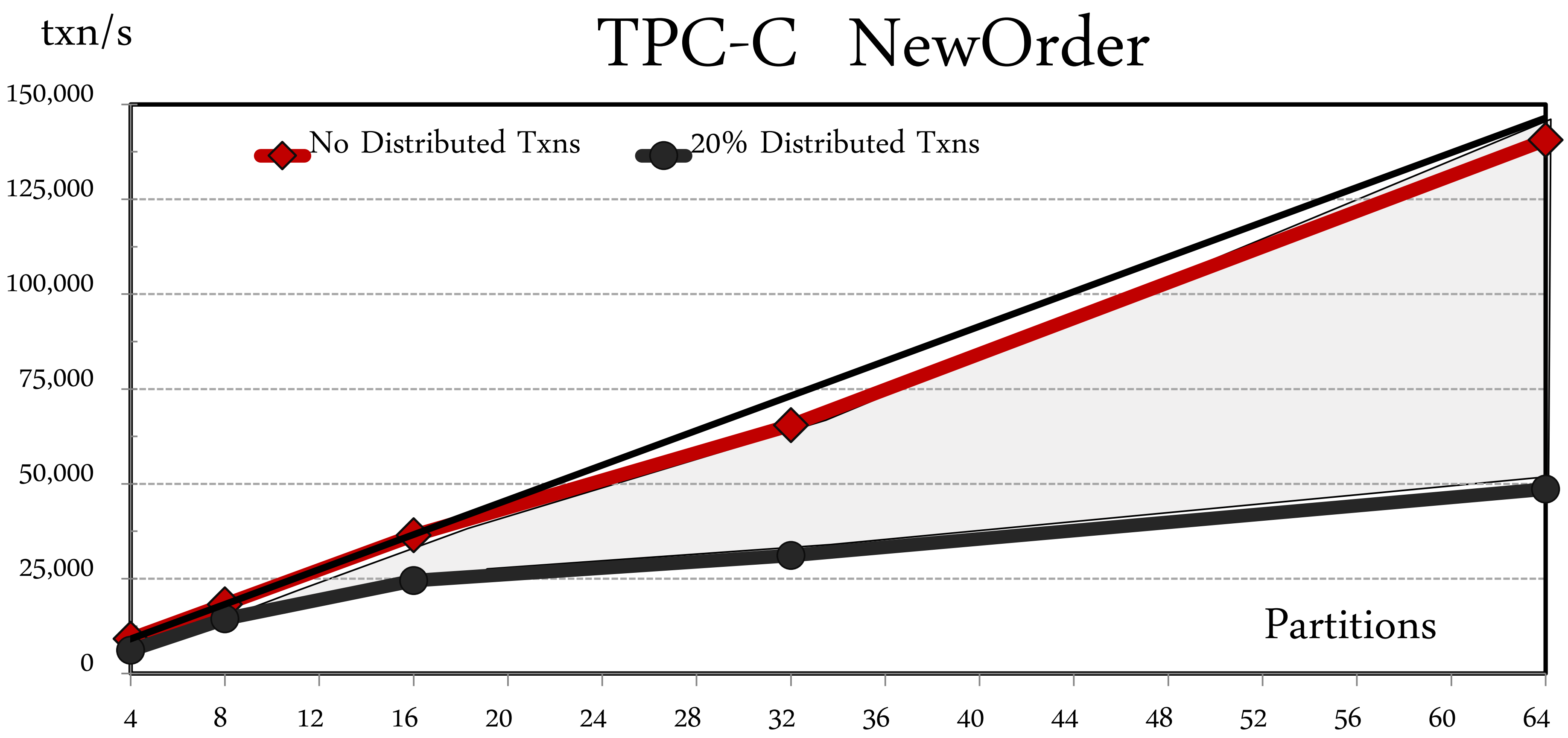
SIGMOD 2012

Horticulture

What are the key issues

- Distributed transactions
- Temporal workload skew

Distributed transactions



Horticulture

What are the key issues

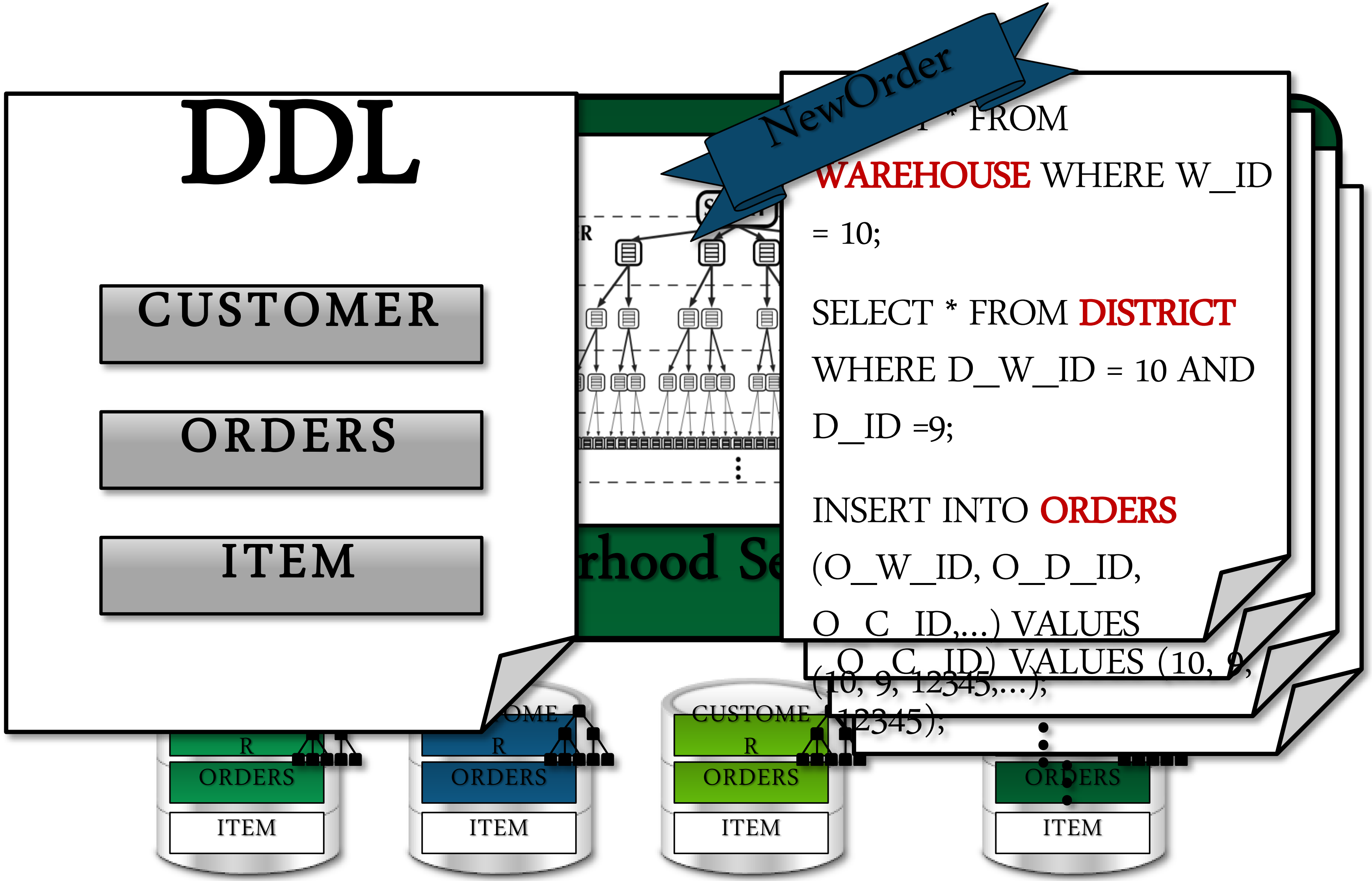
- Distributed transactions
- Temporal workload skew

Horticulture

Temporal workload skew

- Think about the example of Wikipedia
  - Even though the average load of the cluster for the **entire day** is uniform, the load across the cluster for **any point** is unbalanced
- Static Skew Vs. Temporal Skew

Horticulture



Horticulture

- Maintain the **tradeoff** between distributed transactions and temporal skew

- Extend design space to include replicated secondary indexes

- Organically handling stored procedure routing

Large Neighborhood Search

Skew-Aware Cost Model

Horticulture

What are the design options

For each table:

- Horizontally partition
- Replicate on all partitions
- Replicate a secondary index for a subset of its column
- Effectively route incoming transaction requests

Horticulture

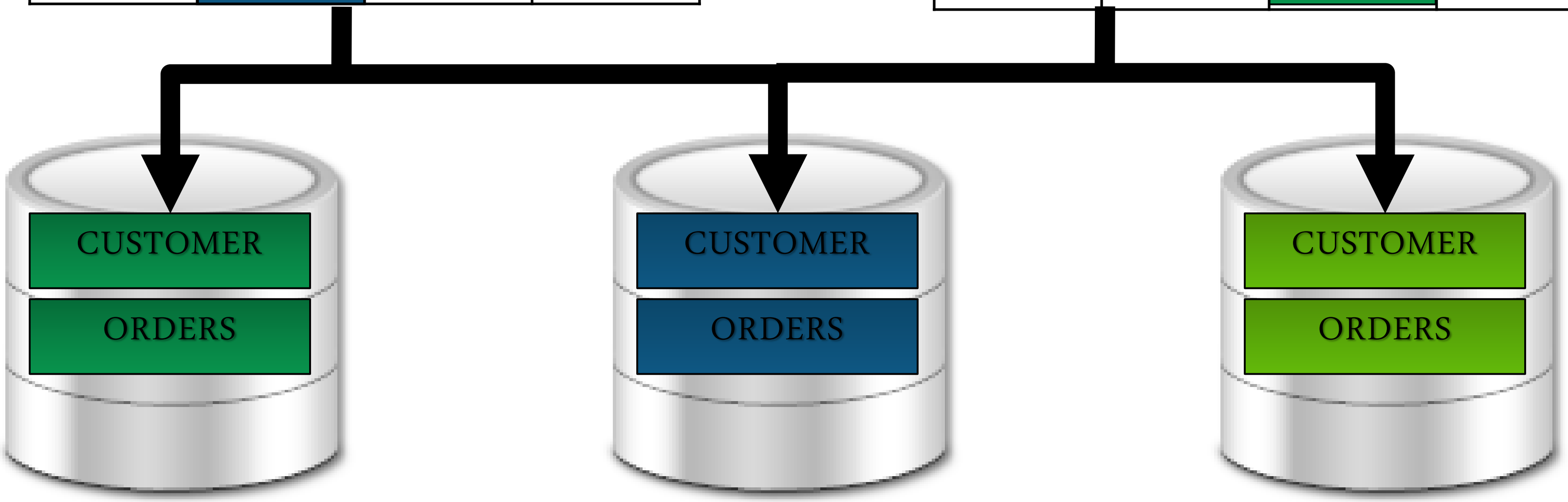
Horizontal Partitioning

CUSTOMER

c_id	c_w_id	c_last	…
1001	5	RZA	-
1002	3	GZA	-
1003	12	Raekwon	-
1004	5	Deck	-
1005	6	Killah	-
1006	7	ODB	-

ORDERS

o_id	o_c_id	o_w_id	…
78703	1004	5	-
78704	1002	3	-
78705	1006	7	-
78706	1005	6	-
78707	1005	6	-
78708	1003	12	-

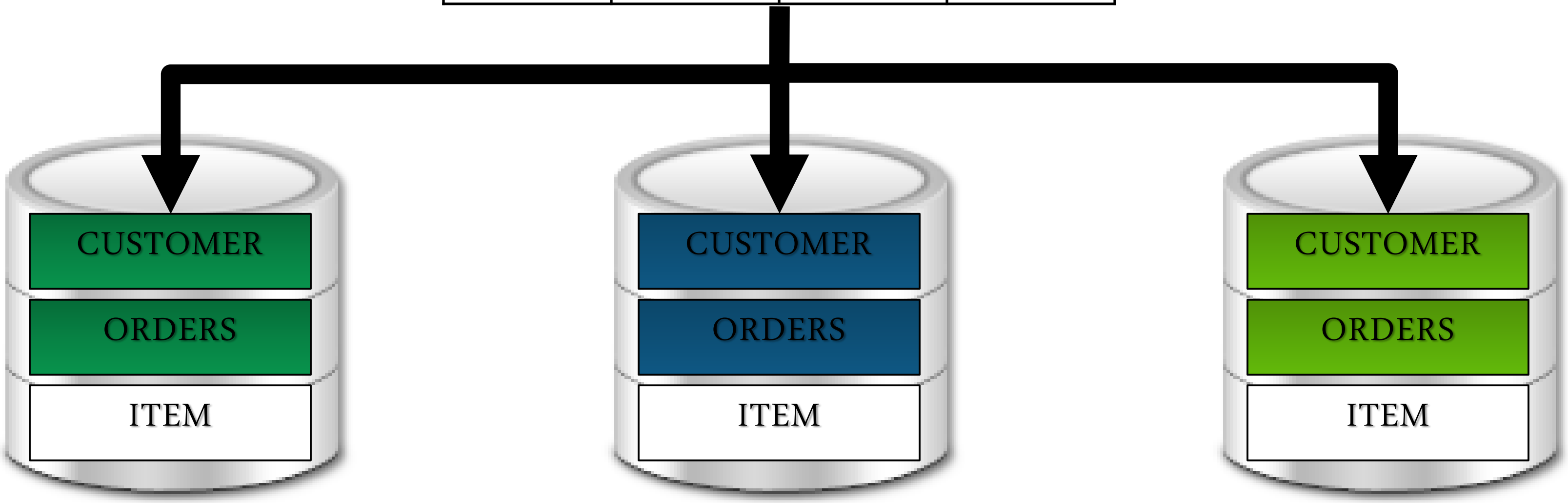


Horticulture

Table Replication

ITEM

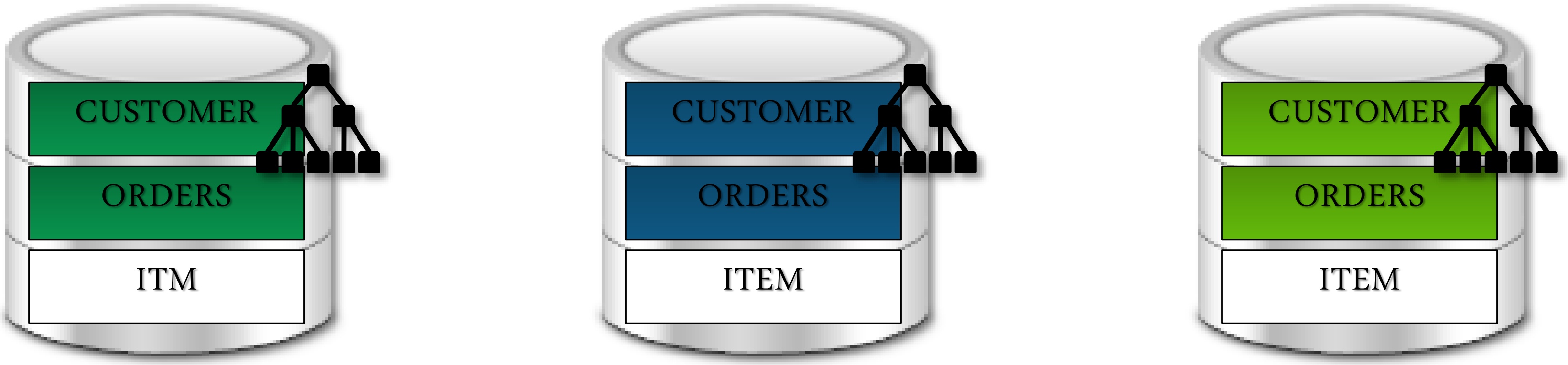
i_id	i_name	i_price	...
603514	XXX	23.99	-
267923	XXX	19.99	-
475386	XXX	14.99	-
578945	XXX	9.98	-
476348	XXX	103.49	-
784285	XXX	69.99	-



Secondary Index

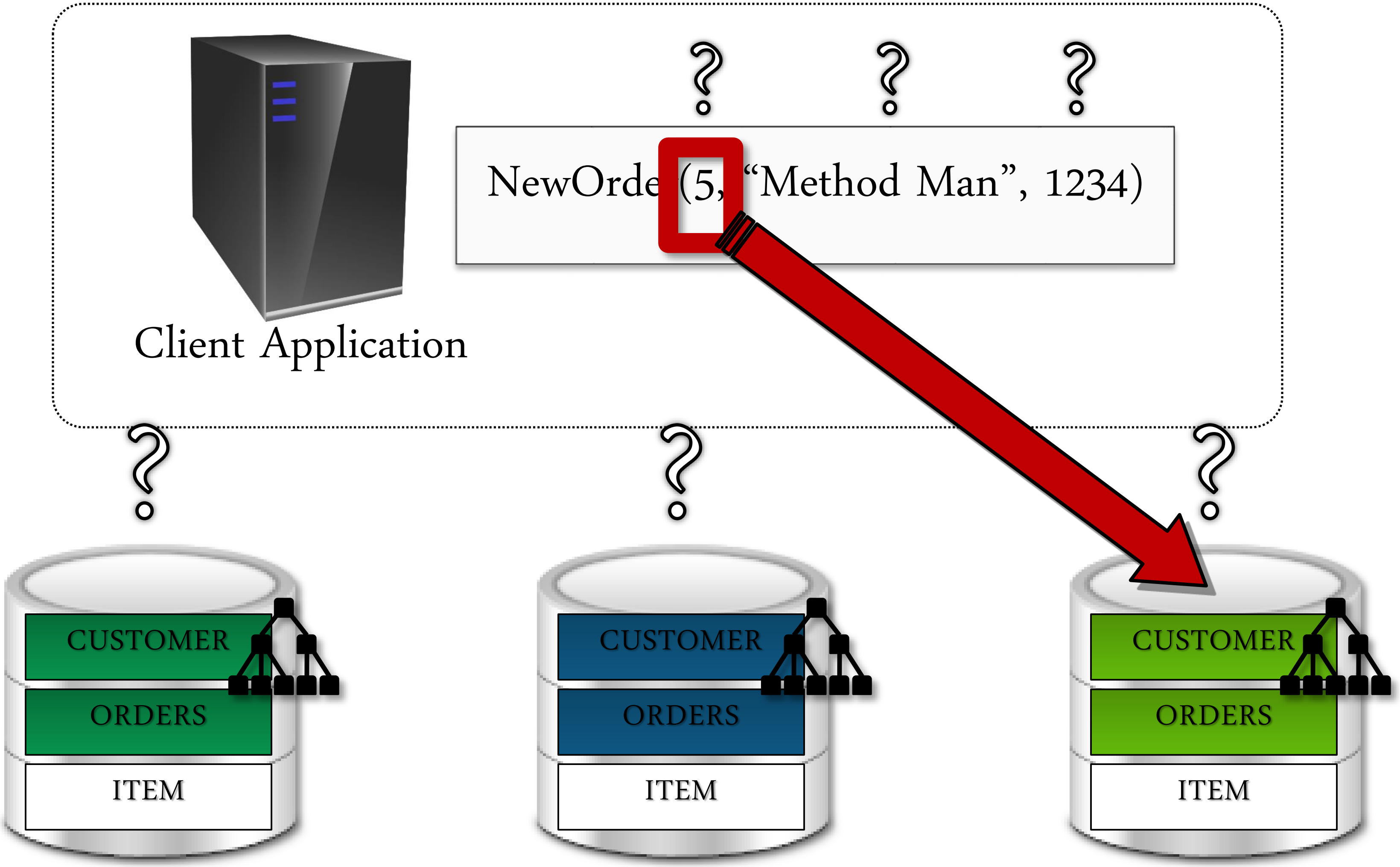
CUSTOMER

c_id	c_w_id	c_last	...
1001		RZA	-
1002		GZA	-
1003		Raekwon	-
1004		[illegible]	-
1005		[illegible]	-
1006		ODB	-



Horticulture

Stored Procedure Routing

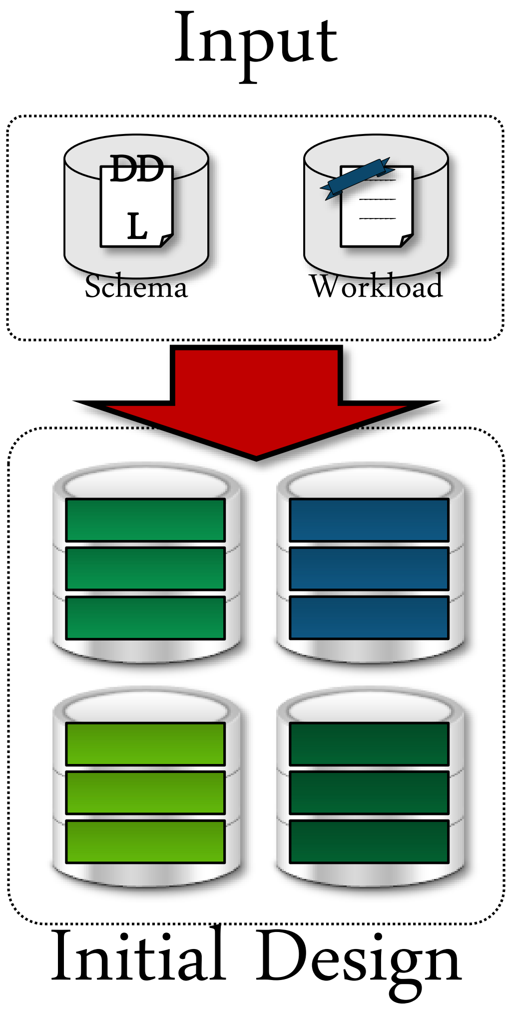


Horticulture

What are the key technique contributions

- Large-Neighborhood Search
- Skew-Aware Cost Model

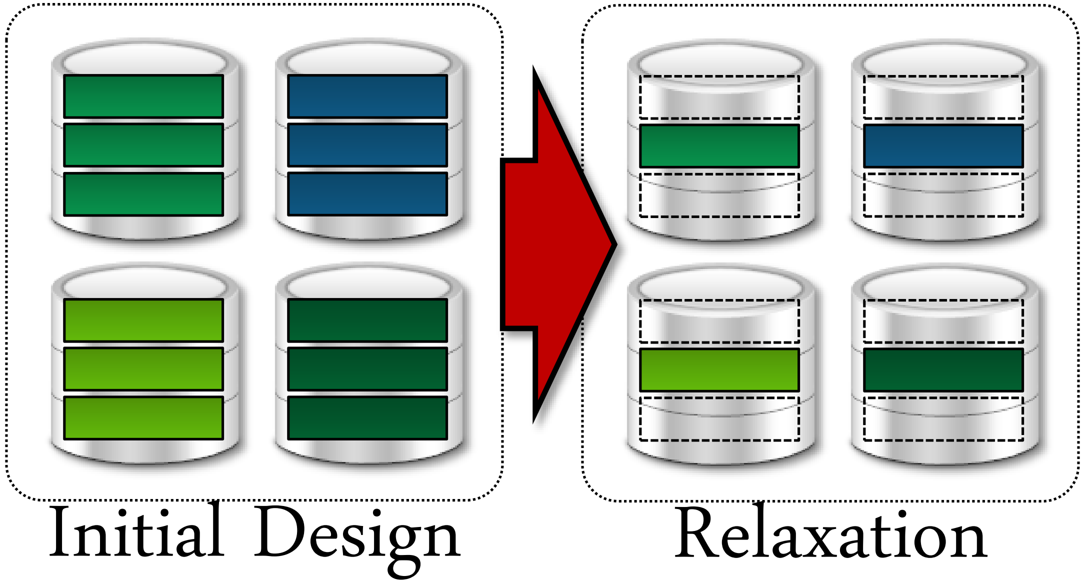
Large-Neighborhood Search



Initial Design

- Select the **most frequently** accessed column for horizontal partitioning
- Greedily replicate **read-only** tables until no space left
- Select next most frequently accessed, **read-only** column as secondary
- Index attribute
- Select the routing parameter for stored procedures

Large-Neighborhood Search



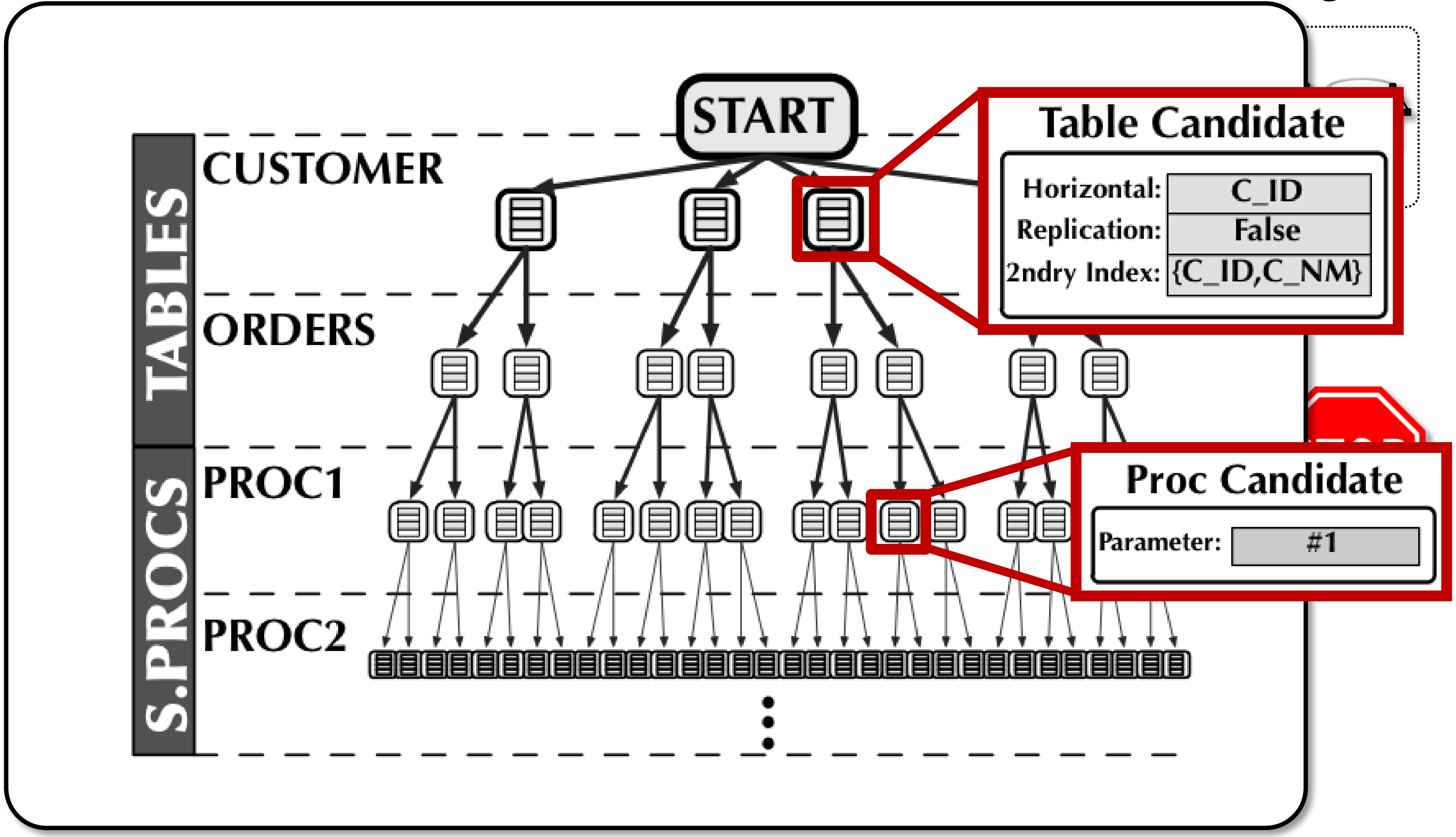
Relaxation

- Allow LNS to **escape a local minimum** and jump to a new neighborhood of potential solutions
- Horticulture must decide:
  - How many tables to relax
  - Which tables to relax
  - What design options will be examined for each relaxed table

Large-Neighborhood Search

Local Search

Best Design



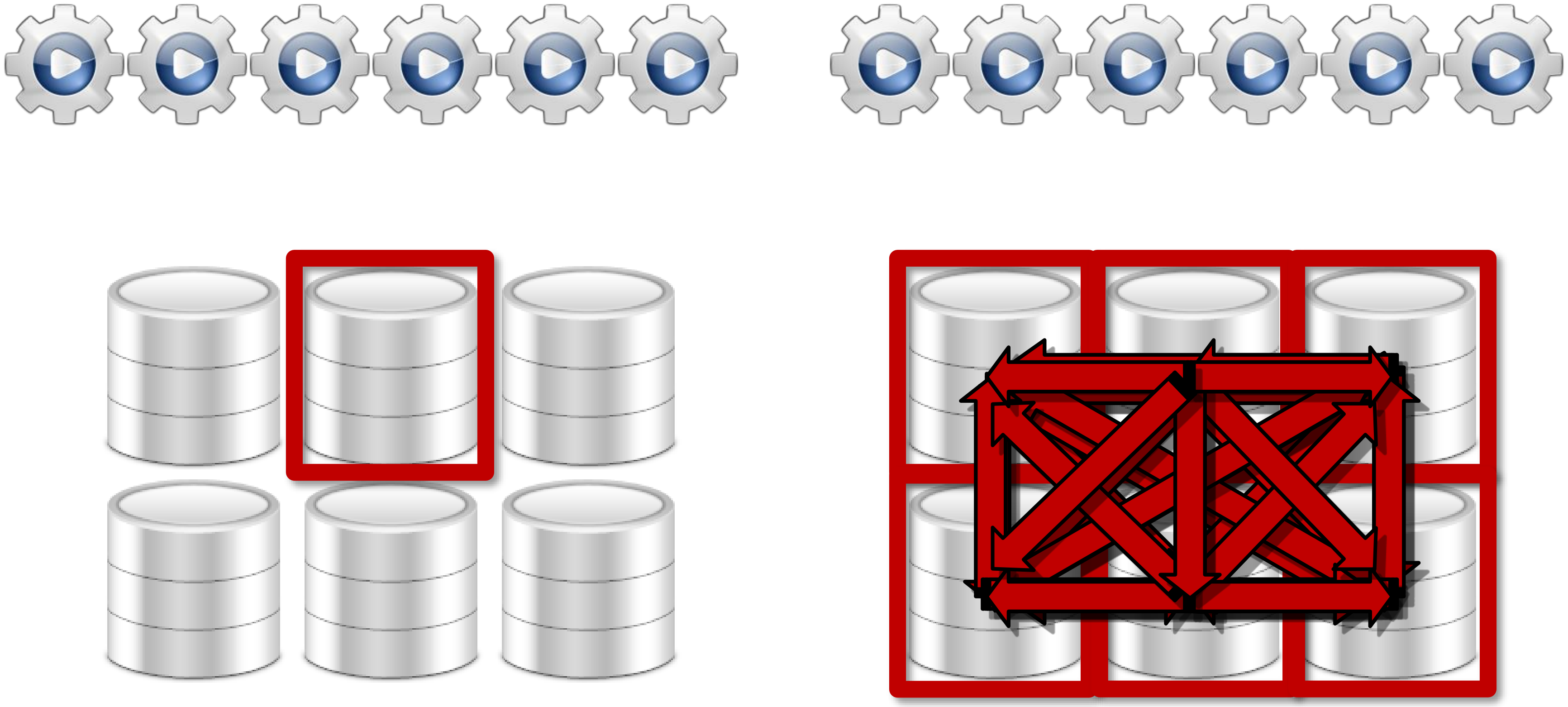
Horticulture

What are the key technique contributions

- Large-Neighborhood Search
- Skew-Aware Cost Model

Cost Model

Distributed Transactions + Workload Skew Factor



Skew-Aware Cost Model

- Accentuates the properties that are important in a DB
- Compute quickly
- Estimate the cost of an incomplete design
- The cost estimates must increase monotonically as more variables are set

Skew-Aware Cost Model

- Measure
  - How much workload executes as a single-partition transactions
  - How uniformly load is distributed across the cluster

$$cost(\mathcal{D}, \mathcal{W}) = \frac{(\alpha \times \boxed{CoordinationCost(\mathcal{D}, \mathcal{W})}) + (\beta \times \boxed{SkewFactor(\mathcal{D}, \mathcal{W})})}{(\alpha + \beta)}$$

Tradeoff!

Skew-Aware Cost Model

Coordinator Cost

$$\left(\frac{partitionCount}{(txnCount \times numPartitions)} \times \left(1.0 + \frac{dtxnCount}{txnCount} \right) \right)$$

Total number of partitions accessed divided by total number of partitions could have been accessed, and scale it up.

Skew-Aware Cost Model

Skew Factor

$$\left(\frac{\sum_{i=0}^{numIntervals} skew[i] \times txnCounts[i]}{\sum txnCounts} \right)$$

To avoid time varying skew, divide W into finite intervals

Incomplete Designs

- Query that references a table with an unset attribute in a design as being unknown
- For each unknown query:
  - Coordinator Cost: Assume that any unknown query is single-partitioned
  - Skew Factor: Assume that unknown queries execute on all partitions in the cluster
- ‘Unknown’ change to ‘known’
- ‘Known’ cannot change to ‘Unknown’

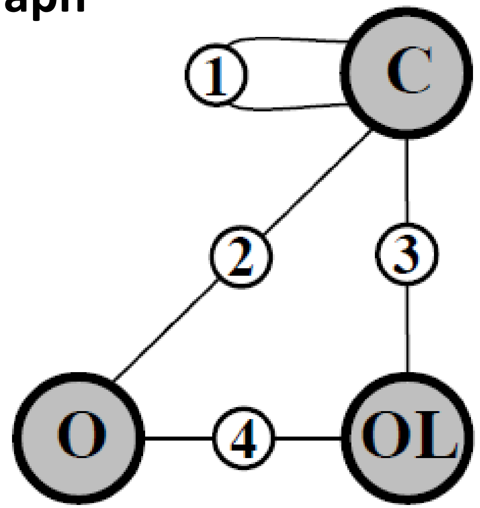
monotonically increase!

Horticulture

Optimizations

- Access Graphs
- Workload Compression

Access Graph



Edge#	Columns	Weight
(1)	C.C_ID ↔ C.C_ID	200
(2)	C.C_ID ↔ O.O_C_ID	100
(3)	O.O_ID ↔ OL.OL_O_ID	100
(4)	O.O_ID ↔ OL.OL_O_ID O.O_C_ID ↔ OL.OL_C_ID	100

Vertex: Table
Edge: tables are co-accessed
Weight of edges: the number of times the queries forming the relationship

Optimizations

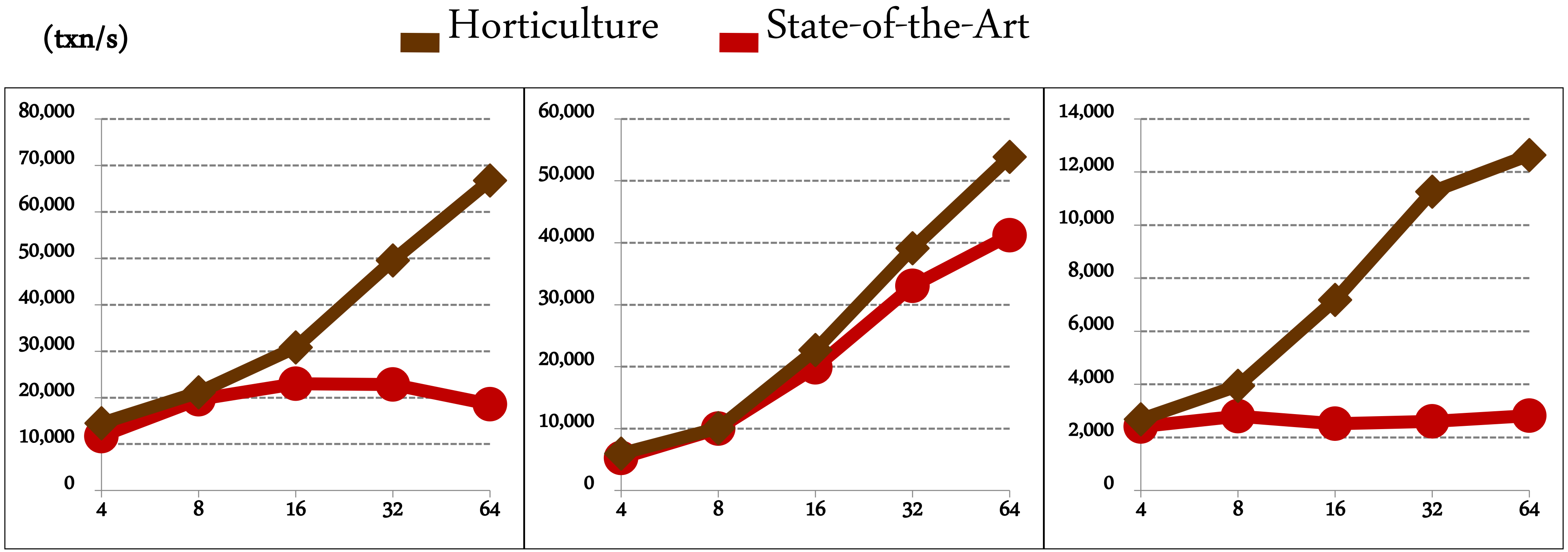
- Access Graphs
- Workload Compression

Horticulture

Workload Compression

- combine sets of similar queries in individual transactions into fewer weighted records
- combine similar transactions into a smaller number of weighted records in the same manner

Throughput

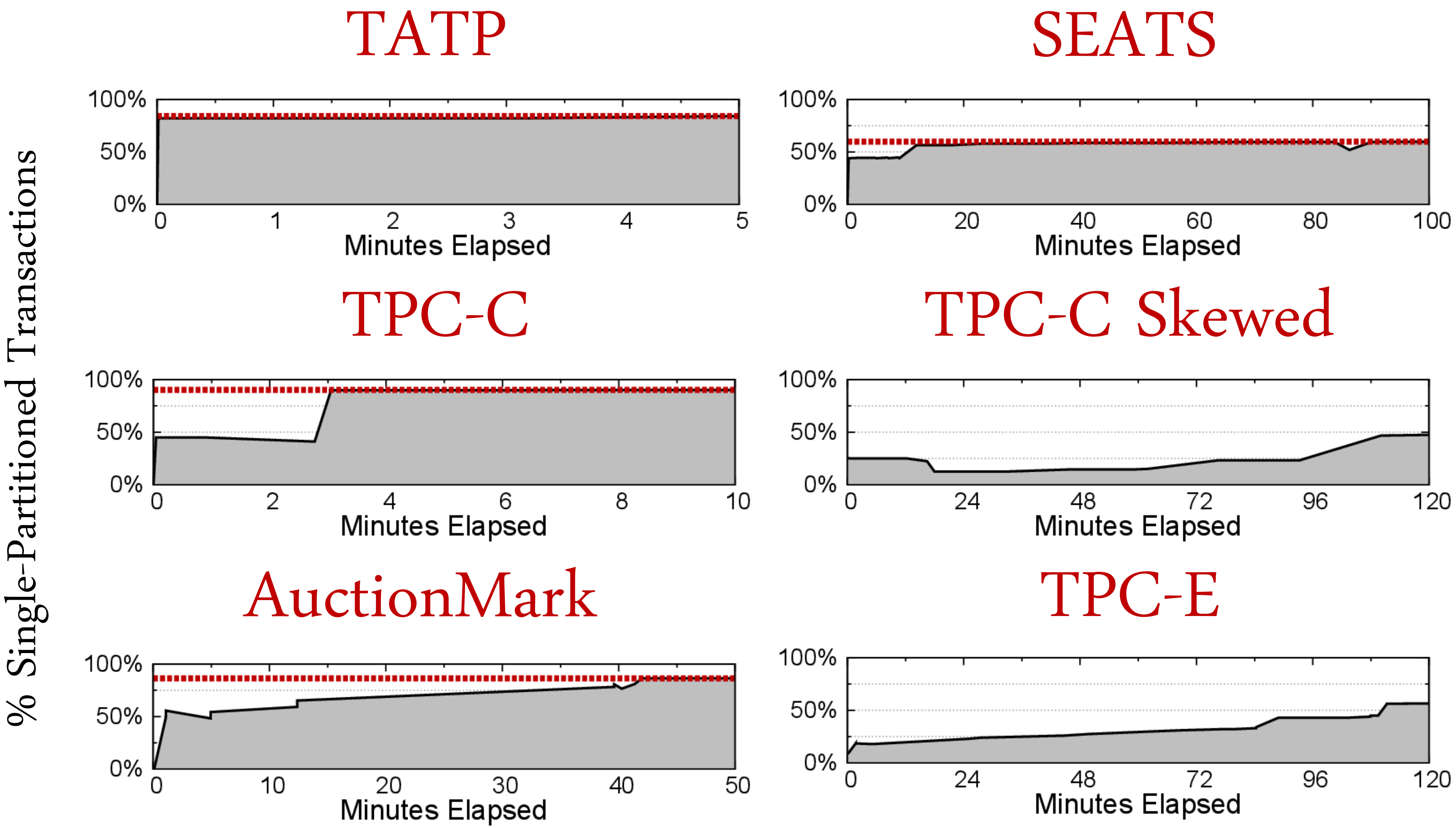


TATP +88%

TPC-C +16%

TPC-C Skewed +183%

Search Times



Andy: it works !