Schism: a Workload-Driven Approach to Database Replication and Partitioning

Carlo Curino et al.

Presented by: Rui Zhang

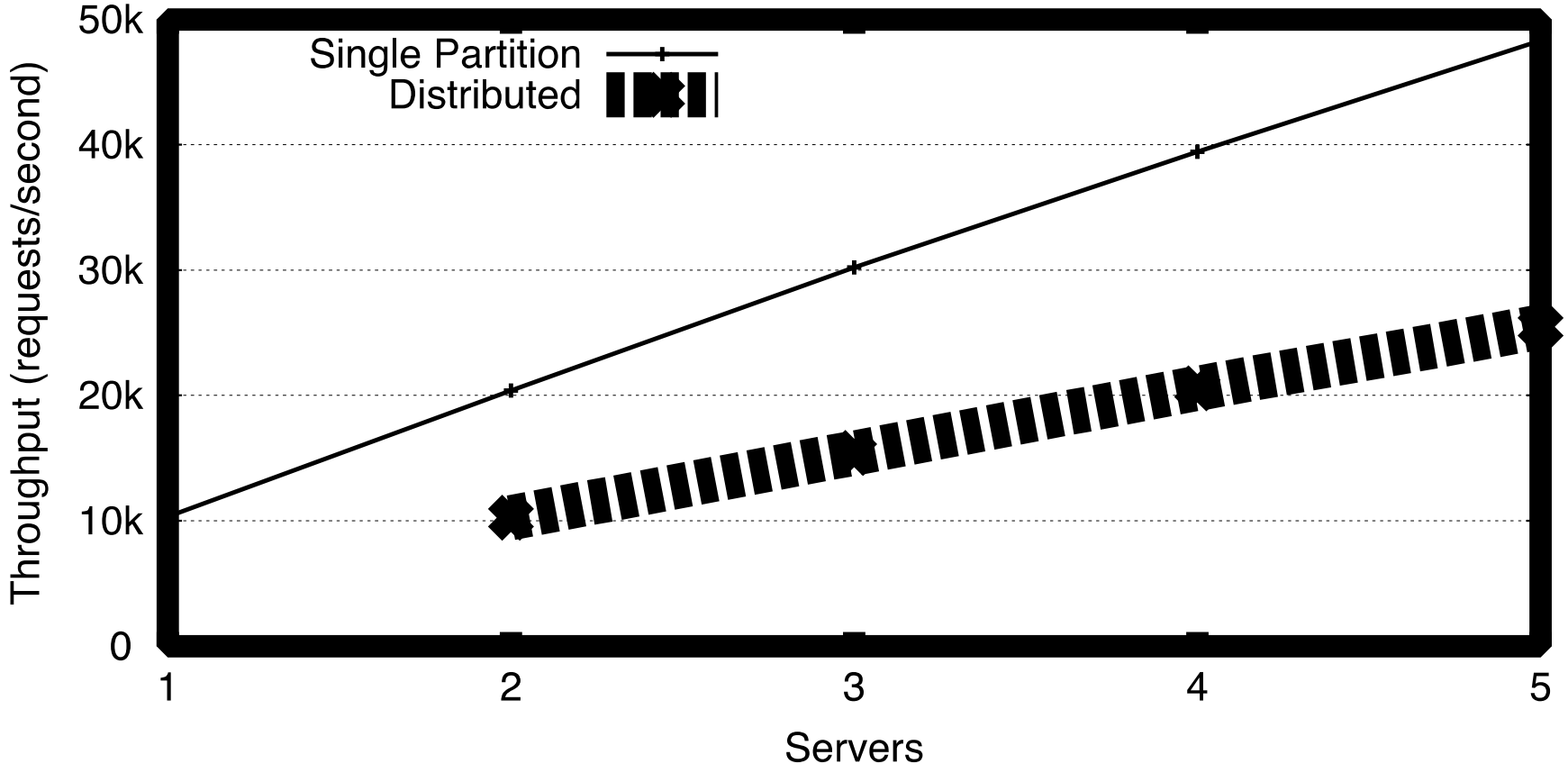
Motivation

- Why partitioning?
  - Scalability
  - Availability
  - Manageability
- Traditional Partitioning
  - Total replication
  - Round-Robin
  - Range
  - Hash

Motivation(Cont.)

- Distributed transactions are expensive
  - Consensus protocol
  - Distributed locks
  - More communication

- Many to Many relationship
  - Hard to partition


Single Partition
Distributed
Throughput (requests/second)
50k
40k
30k
20k
10k
0
1
2
3
4
5
Servers

Solution

- Minimize distributed transactions
- Balance the workload of each partition

Schism

- Part of RelationalCloud System
  - An MIT-based effort to investigate technologies and challenges related to Database-as-a-Service (DAAS) within cloud-computing[1]
  - Researchers including Carlo, Evan and Sam participate this project

- DAAS
  - Cloud computing based service is a novel market for data management
  - More and more dedicated data management engines which complicates the process of choosing and deploying
  - DAAS is to hide the complexity with a simple interface

[1] http://relationalcloud.com


H-Store

Schism

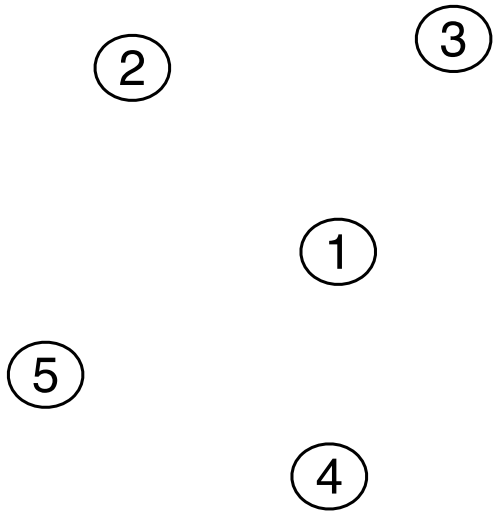
- Input
  - Workload trace
- Output
  - Partitioning and replication plan
- Five steps
  - Data pre-processing
  - Creating the graph
  - Partitioning the graph
  - Explaining the partition
  - Final Validation

Creating the graph

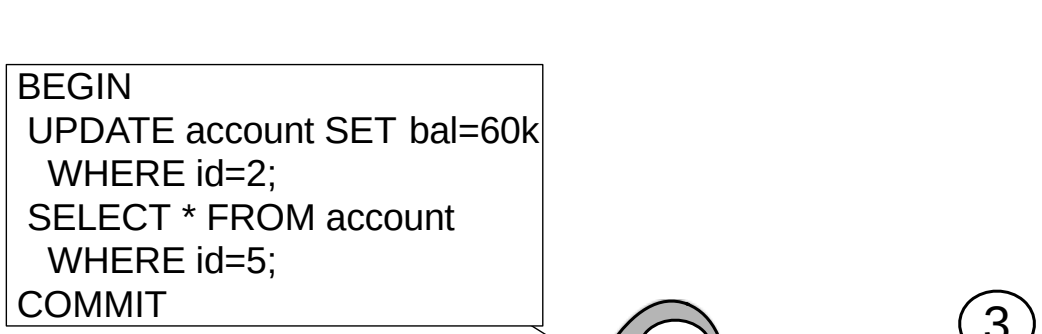
- Node: Tuple
- Edge: Transaction
- Weight: Number of transactions

Creating the graph

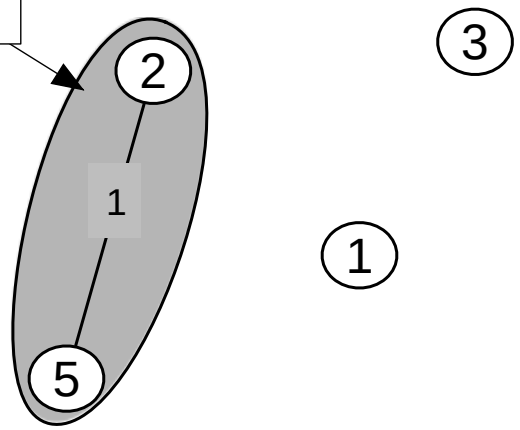
account		
id	**name**	**bal**
1	carlo	80k
2	evan	60k
3	sam	129k
4	eugene	29k
5	yang	12k
...	....	....



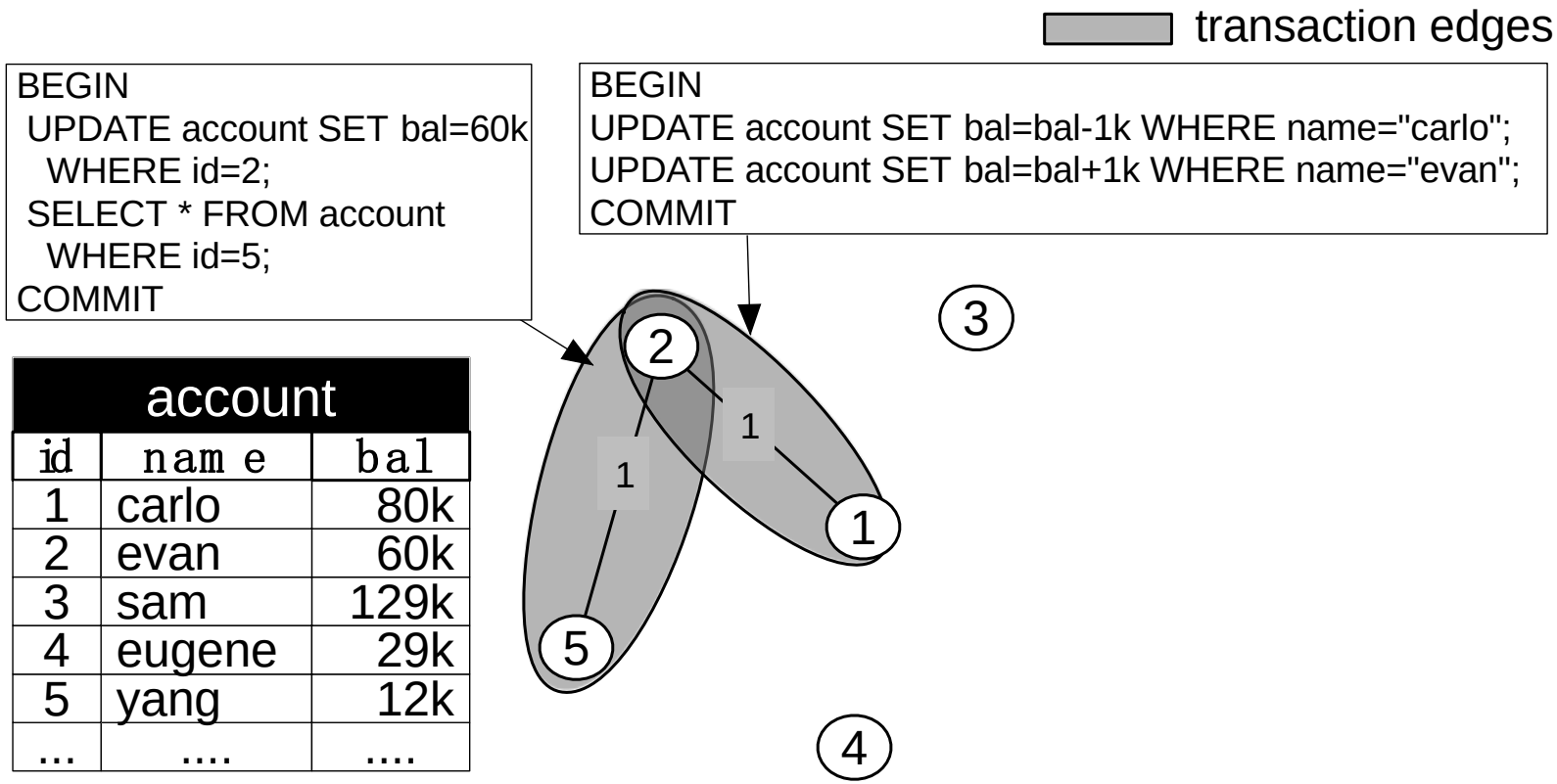
Creating the graph



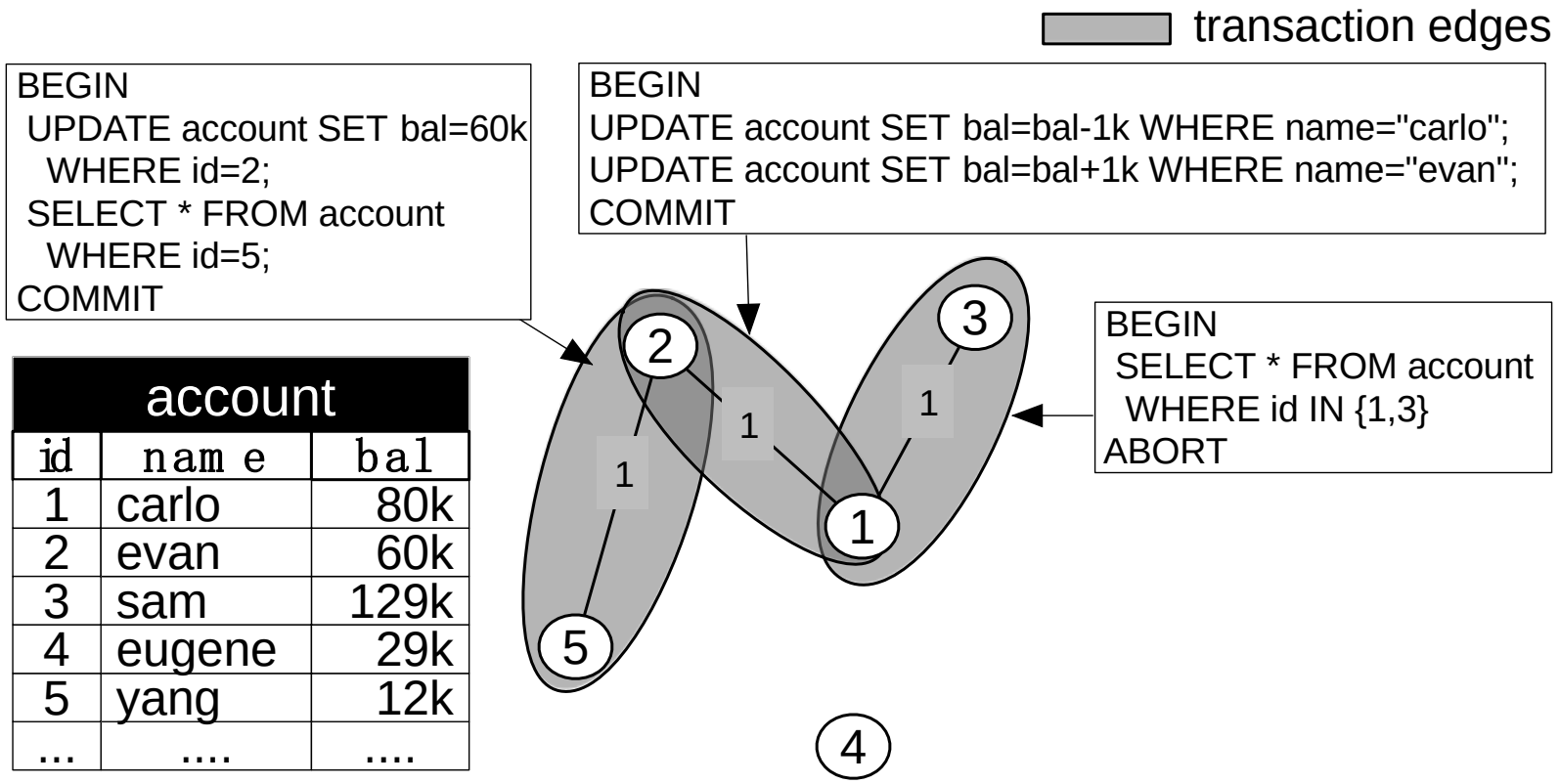
account		
id	name	bal
1	carlo	80k
2	evan	60k
3	sam	129k
4	eugene	29k
5	yang	12k
...	....	....



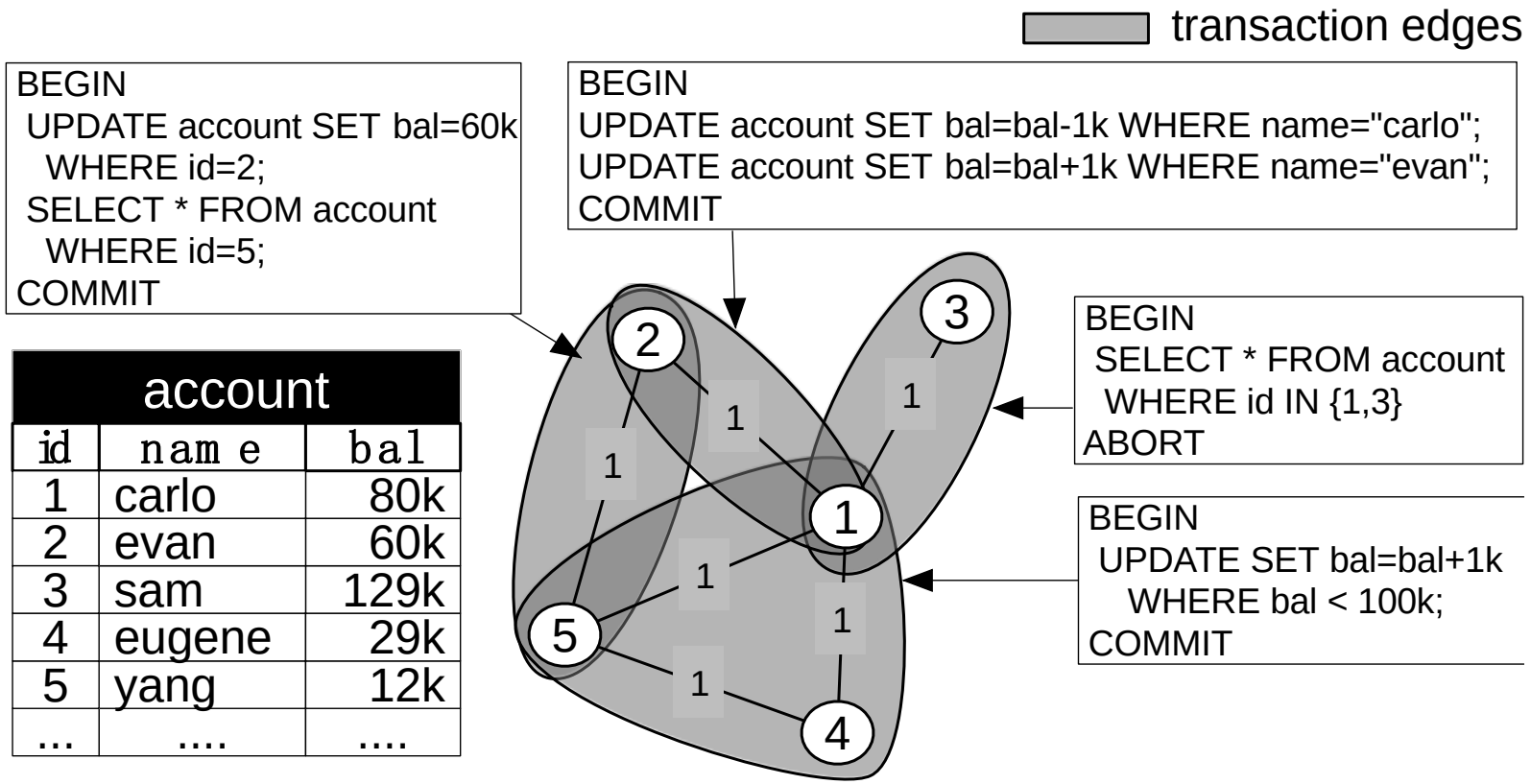
Creating the graph



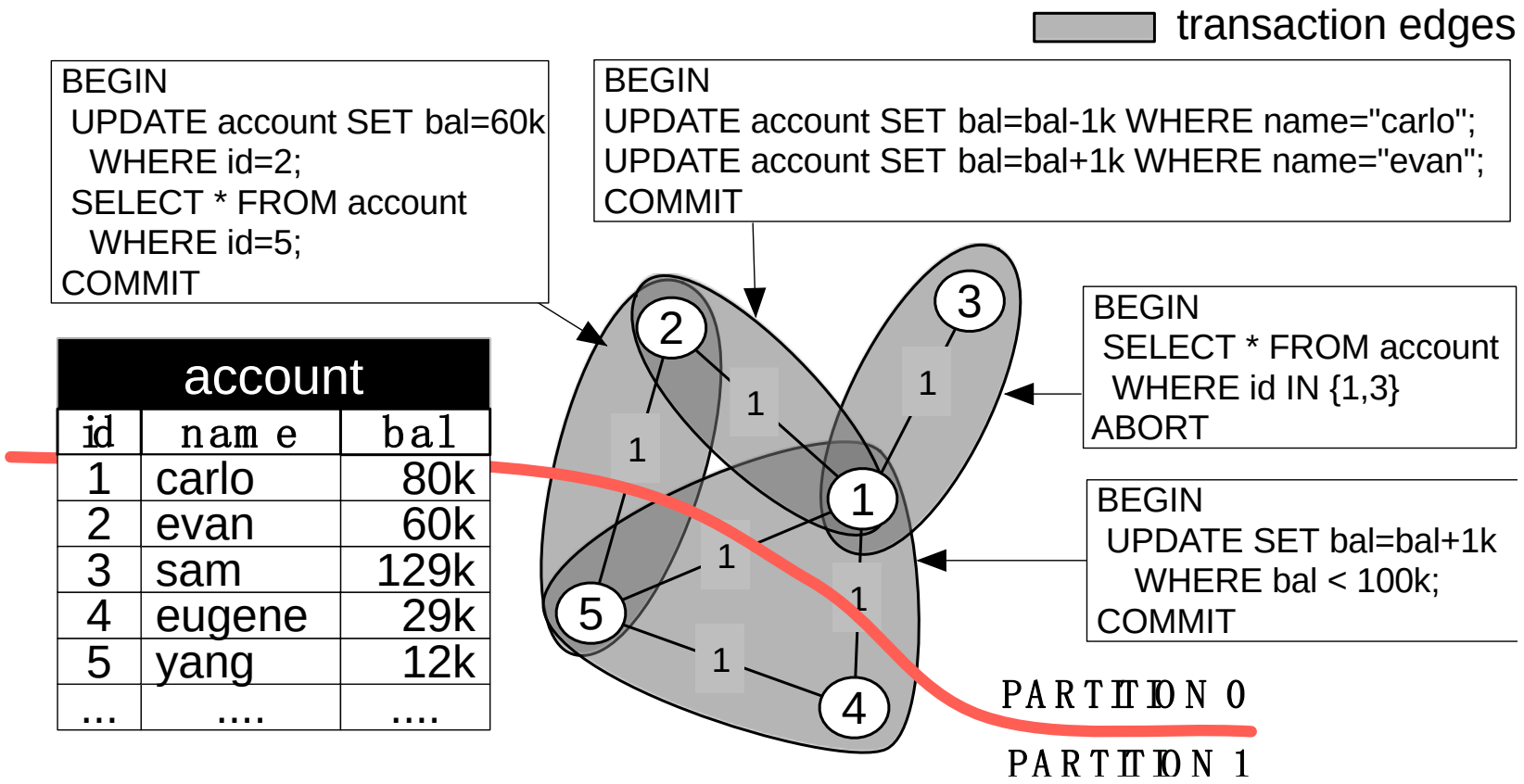
Creating the graph



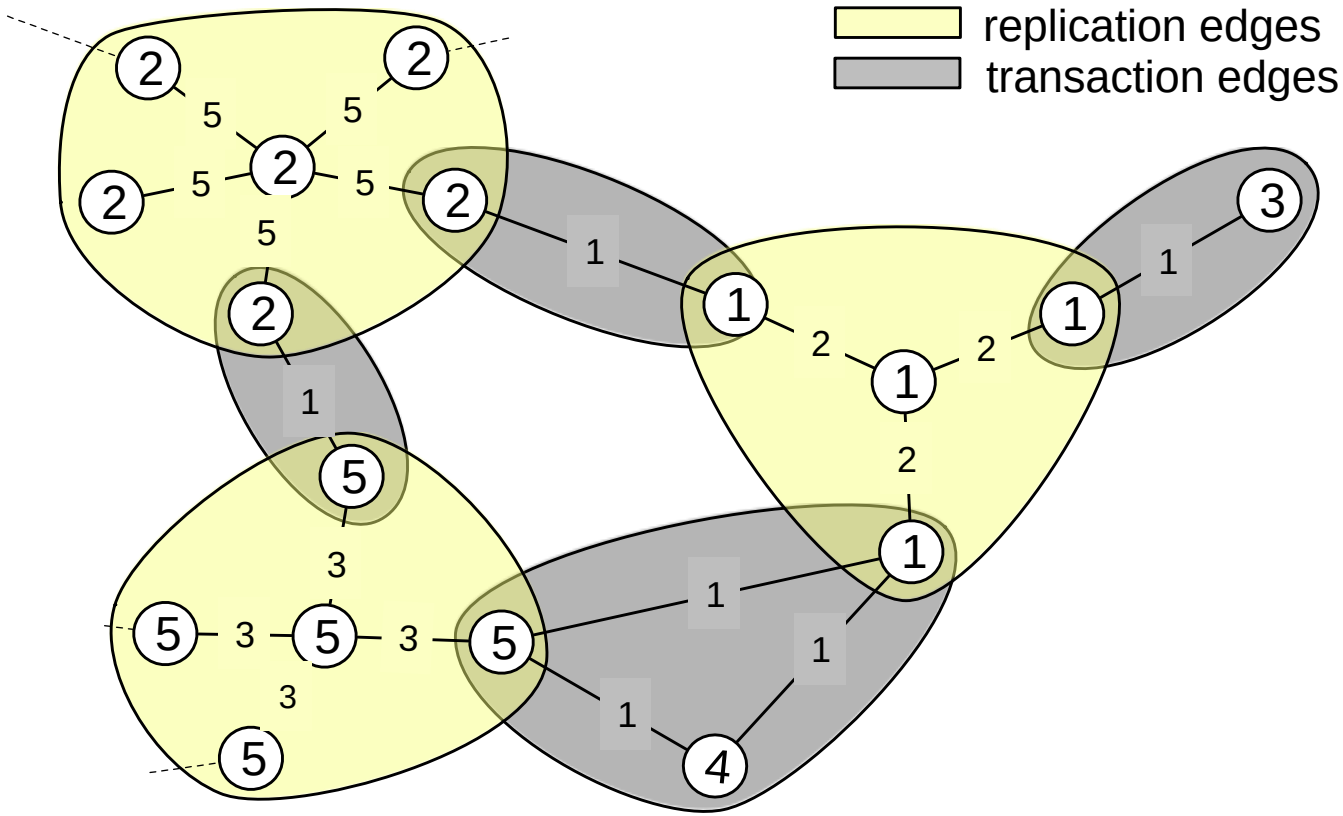
Creating the graph



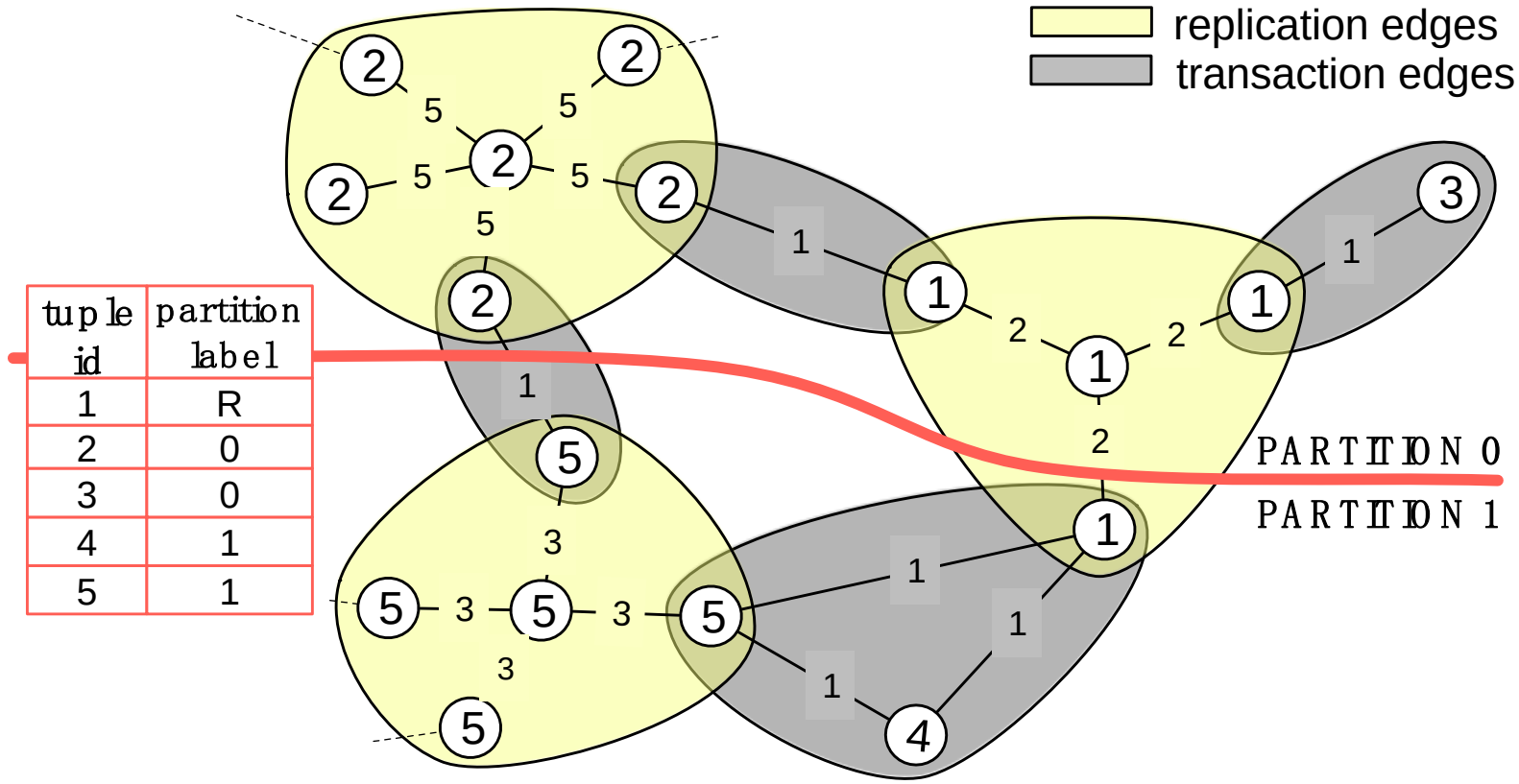
Creating the graph



Tuple Replication



Tuple Replication



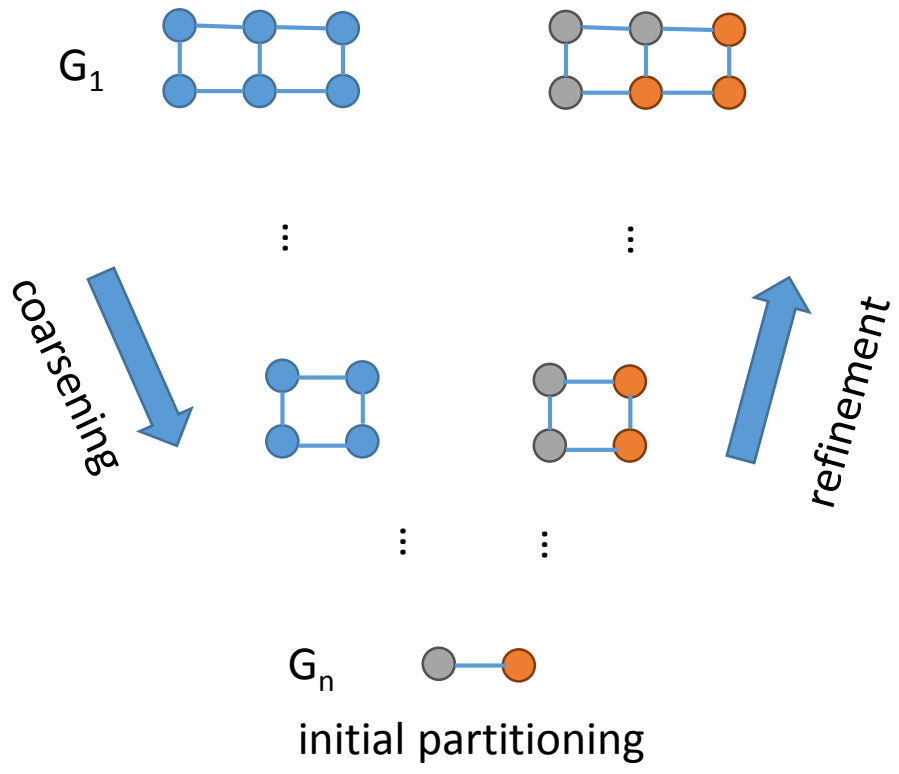
tuple id	partition label
1	R
2	0
3	0
4	1
5	1

Graph Partitioning

- Metis algorithm
  - Coarsening
  - Initial partitioning
  - Refinement

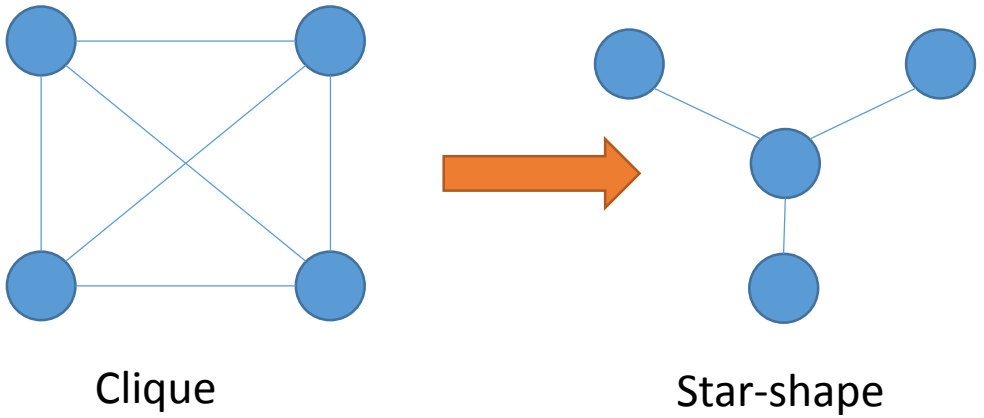
  See www.cs.umn.edu/~metis for more detail

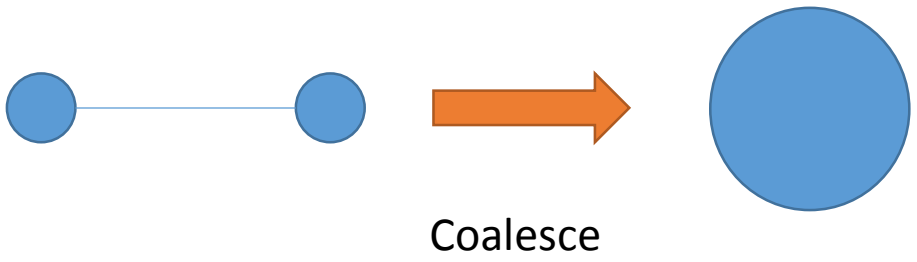
- Goal
  - Minimum cut and balanced partitions
- Output
  - Lookup table



Graph Size Reduction

- Sampling
  - Transaction level
  - Tuple level
- Blanket-statement filtering
- Relevance filtering
- Star-shaped replication
- Tuple coalescing

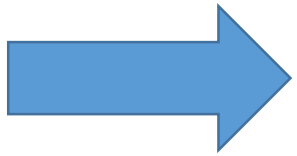




Explanation Phase

- Method
  - Decision Tree
- Input
  - Tuple->partition mappings
- Output
  - Compact Model

tuple id	partition label
1	R
2	0
3	0
4	1
5	1



(id = 1) -> partitions = {0, 1}
(2 <= id < 4) -> partition = 0
(id >= 4) -> partition = 1

Explanation Phase

- Candidate attributes
  - Frequent used attributes in WHERE clauses
    - Item ids and warehouse ids for “stock”
  - Join predicates
    - Co-location of the joined tuples
    - Limited use (perform the join) but supported

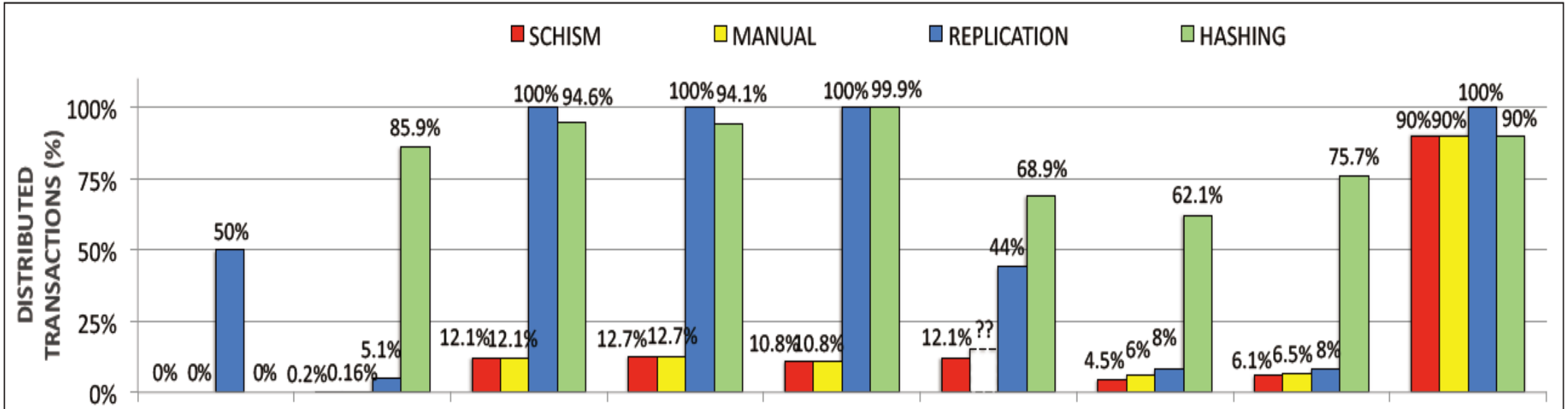
Query/Tuple Routing

- Centralized Router
  - A decision tree
  - Determine which partition the query or tuple belongs to
  - Spare partition that store tuples don’t hit
    - If full, rerun the partitioning algorithm and retrain the decision tree
    - After retraining, some tuples should be migrated from one partition to another via data migration SQL scripts
  - Broadcast to all partitions for attributes that are not partitioning attributes

Evaluation

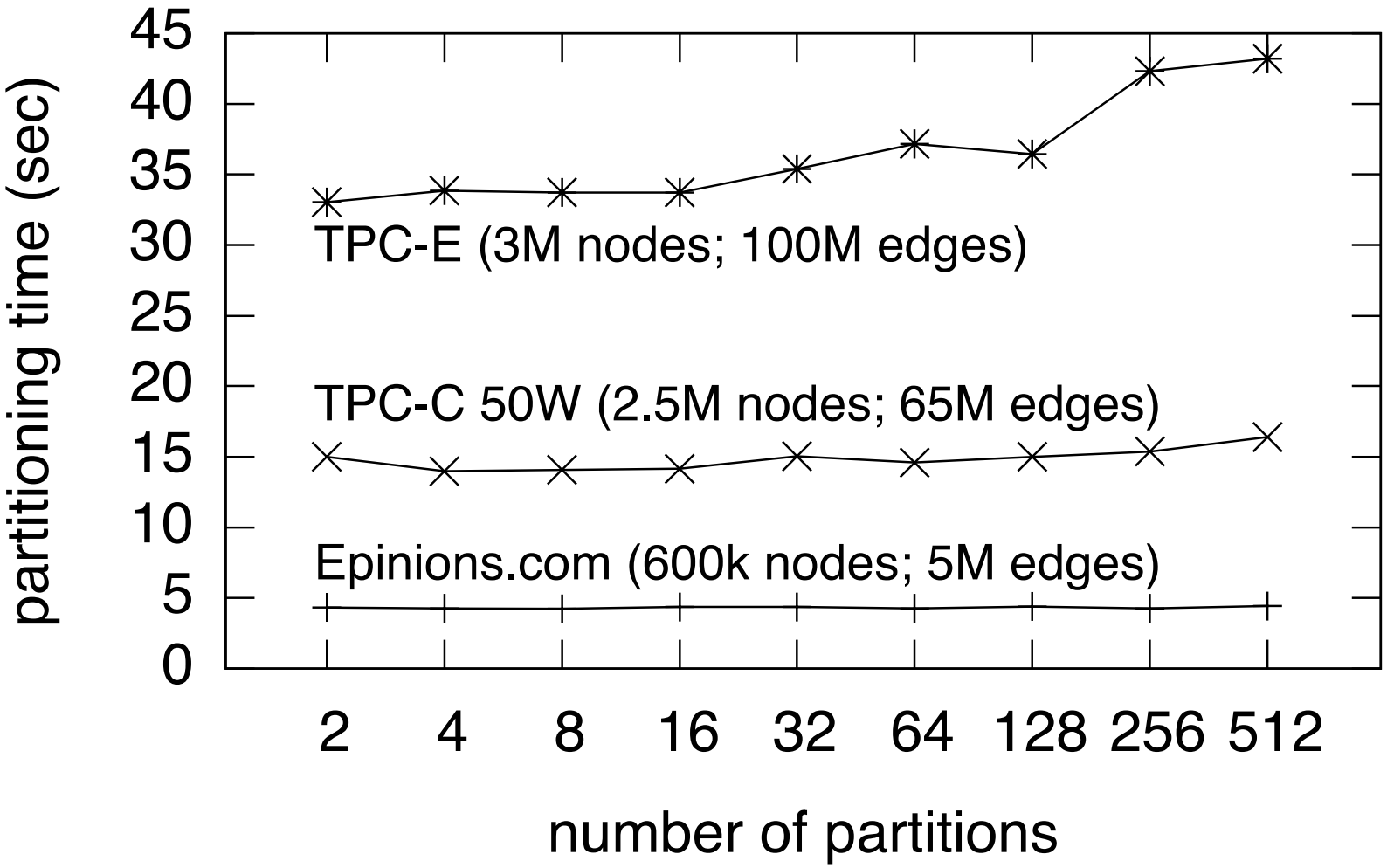
- Schism
- Manual
- Total Replication
- Hash partitioning

Evaluation

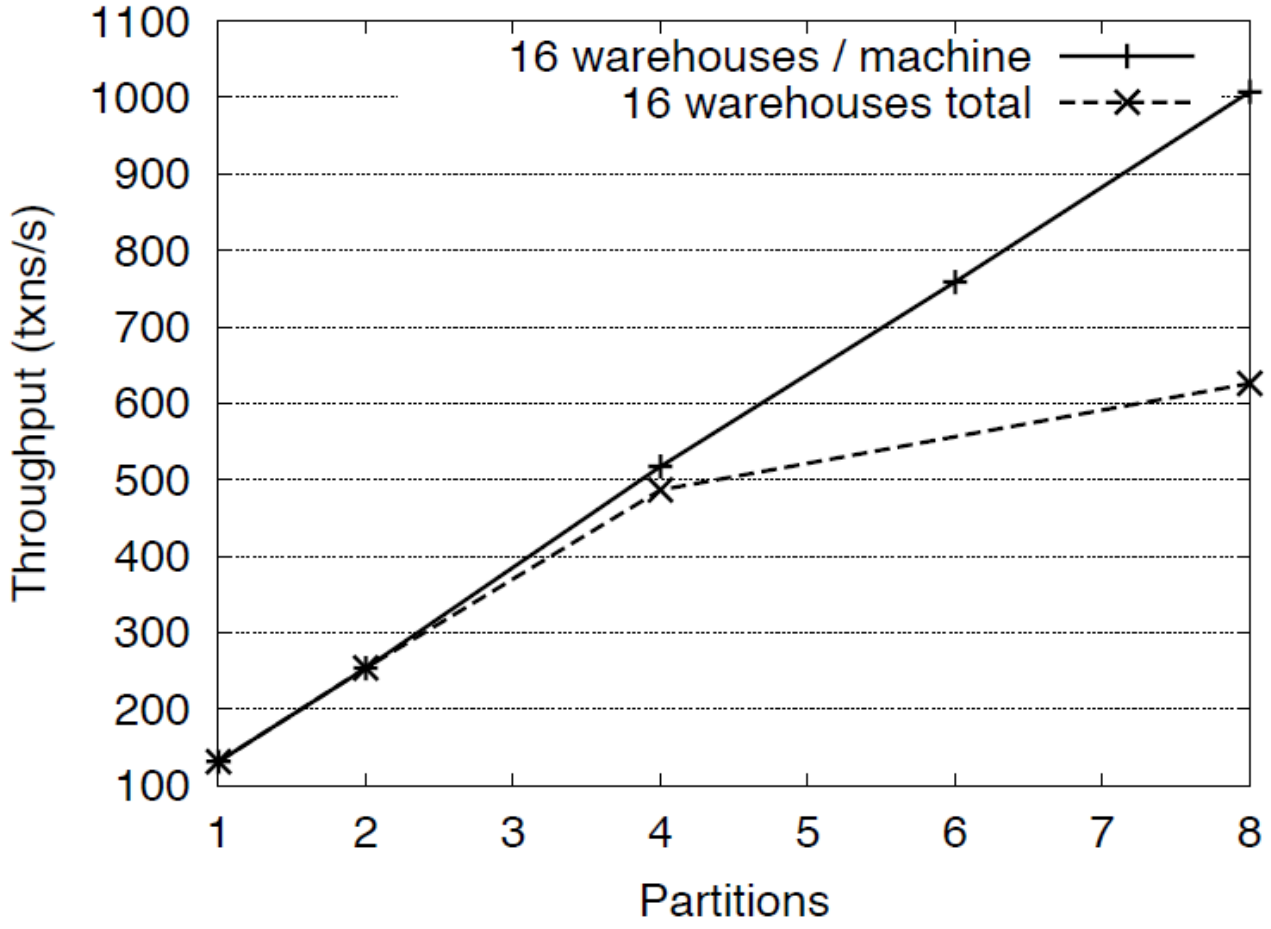


dataset:	YCSB-A	YCSB-E	TPCC-2W	TPCC-2W	TPCC-50W	TPC-E	EPINIONS	EPINIONS	RANDOM
partitions:	any	100	2	2	10	2	2	10	10
coverage:	any	1%	50%	0.5%	1%	5%	15%	15%	any
SCHISM:	hashing	range-predicates	range-predicates	range-predicates	range-predicates	range-predicates	look-up table	look-up table	hashing

Evaluation


partitioning time (sec)
45
40
35
30
25
20
15
10
5
0
TPC-E (3M nodes; 100M edges)
TPC-C 50W (2.5M nodes; 65M edges)
Epinions.com (600k nodes; 5M edges)
2
4
8
16
32
64
128
256
512
number of partitions

Evaluation


16 warehouses / machine
16 warehouses total
Throughput (txns/s)
1100
1000
900
800
700
600
500
400
300
200
100
1
2
3
4
5
6
7
8
Partitions

Limitation

- Not suitable for insert-heavy workloads
  - Involves many data migrations
- Not suitable for databases with large size
  - Partitioning time will be long
- There will be false positive in decision tree
  - Overhead to correct this error
- Compared with Horticulture, it does not consider temporal skew