OSARE
Opportunistic Speculation in Actively REplicated Transactional Systems

SAPIENZA
UNIVERSITÀ DI ROMA

Roberto Palmieri, Francesco Quaglia, Paolo Romano

Roberto Palmieri – Workshop on Transactional Memory (WTM 2012)
Bern, Switzerland, April 10, 2012

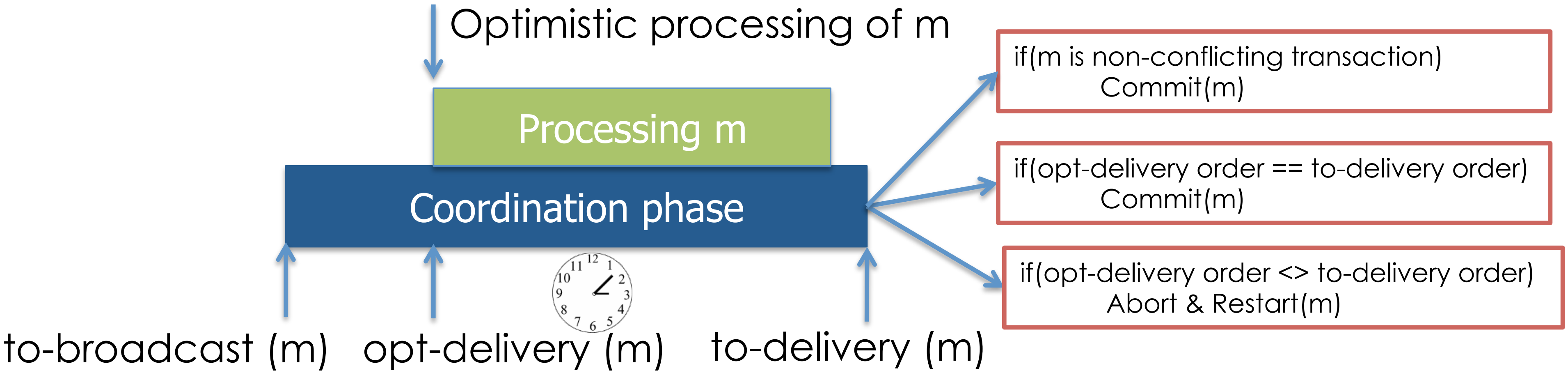
Why Active Replication

- Each replica keeps all data and executes the same transactions in the same order
- PRO (+)
  - Full failure masking
  - No coordination for processing read-only transactions
  - No network synchronization while processing
  - Prone to target performance issues
- CONS (-)
  - Agreement on common execution order

Literature solution for actively replicated transactional systems

Optimistic Approach (OPT)

- Based on Optimistic Atomic Broadcast as GCS
- It processes in optimistic manner:
  - At most one conflicting transaction
  - Any non-conflicting transactions



Critiques to OPT

- A-priori knowledge on transaction read/write sets
- Limited overlapping in case of fine-grain transactions

$$\frac{\text{Coordination delay}}{\text{Local Transaction Execution Time}}$$

Unbalancing Ratio

Traditional Scenarios

$$\frac{\approx 2 \text{ msec}}{\approx 1/10 \text{ msec}}$$

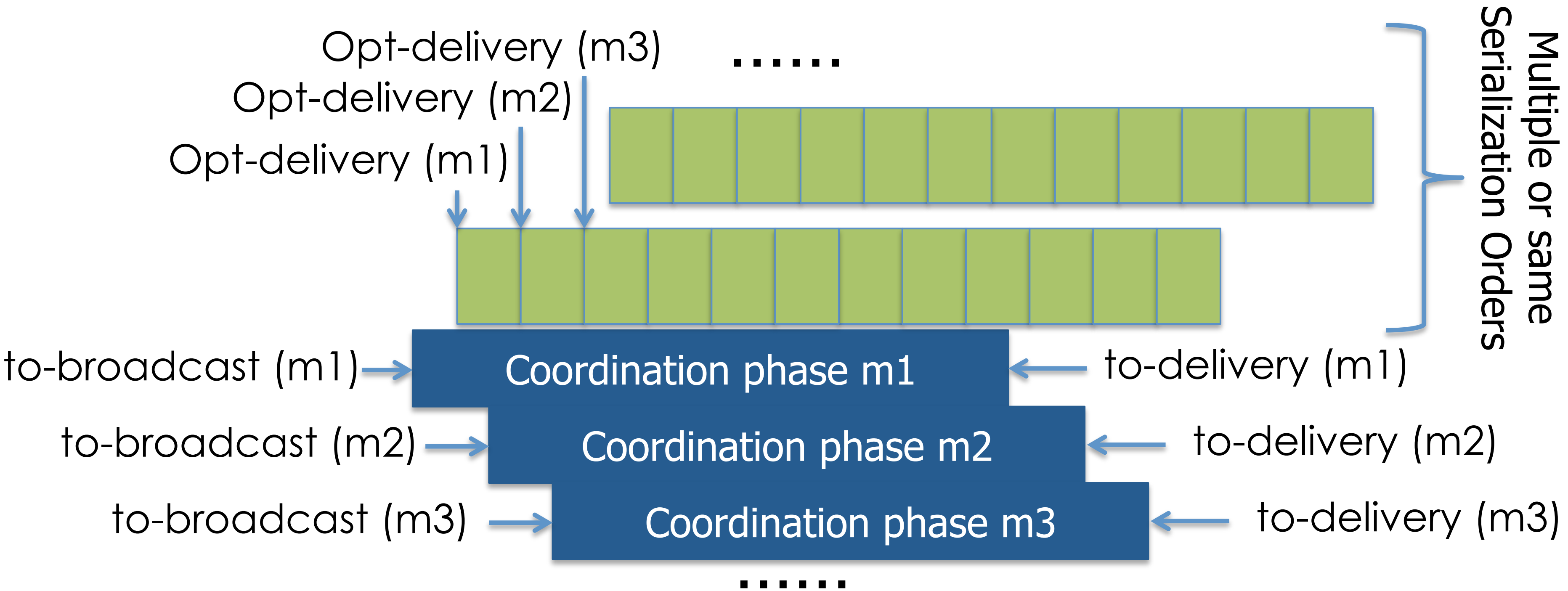
Modern (STM) Scenarios

$$\frac{\approx 2 \text{ msec}}{\approx 10/100 \ \mu\text{sec}}$$

Resources underutilization

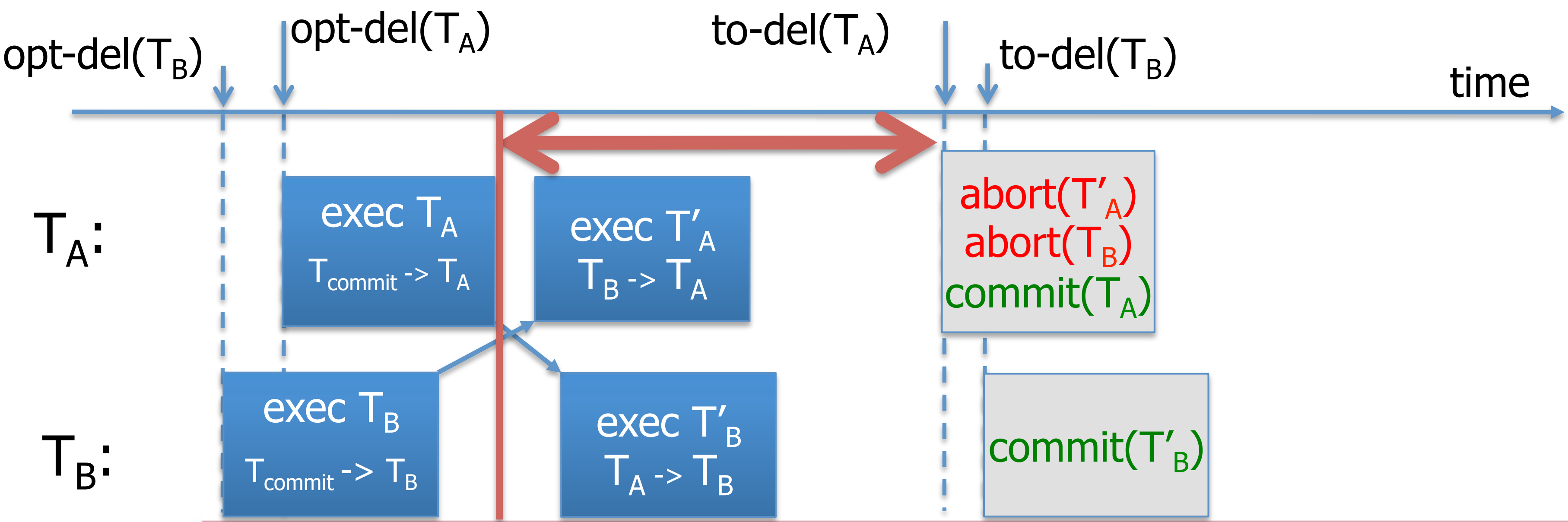
Target: Maximize the overlap

- Coordination delay Vs Local transaction processing
- Local transaction processing Vs Local transaction processing



How: Speculative Processing

- Basic ideas:
  - Activate all transactions as soon as they are optimistic delivered
  - Explore (in depth and/or in breadth) multiple serialization orders



Changing the Perspective of Speculation

OSARE

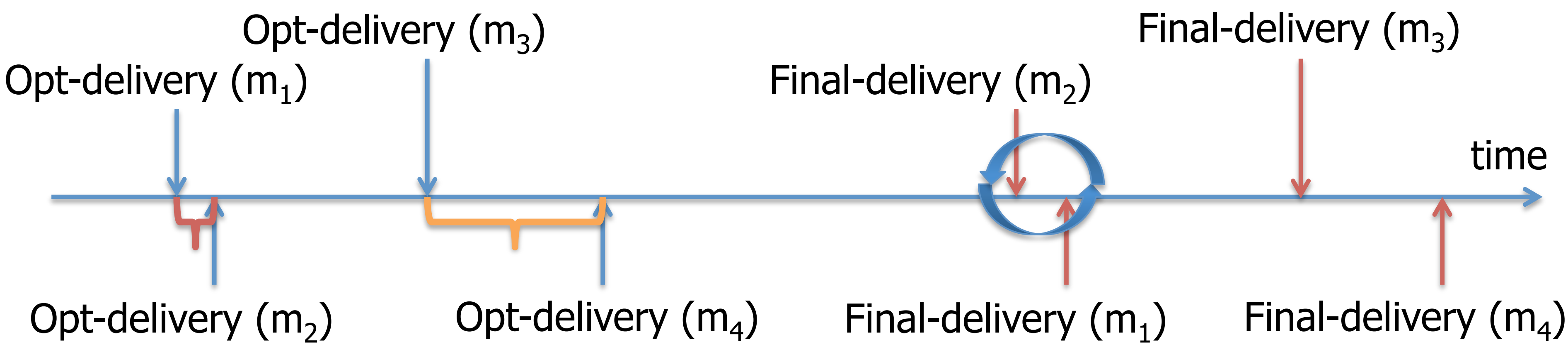
Speculating According to an Opportunistic Paradigm

Opportunistic Speculation in Active Replication (OSARE) [SRDS2011]

- Simple data structures -> Polynomial Overhead
- Targets:
  - Follow the optimistic delivery order
  - Explore multiple serialization orders only when concurrency increases
- How:
  - Exploit the so called *snapshot miss* event to expand the coverage of alternative SOs

OSARE: Key Ideas

- Speculation degree reflects network concurrency
- The probability to invert the optimistic delivery order wrt the final one increases with the arrival rate of network messages



Snapshot Miss Event

- A snapshot miss event occurs when:
  - T_i^s writes data item X for which T_j^t has already issued a read operation

T_j^t missed the snapshot of T_i^s

A new instance of T_j^t -> $T_j^{(t+1)}$ is spawned

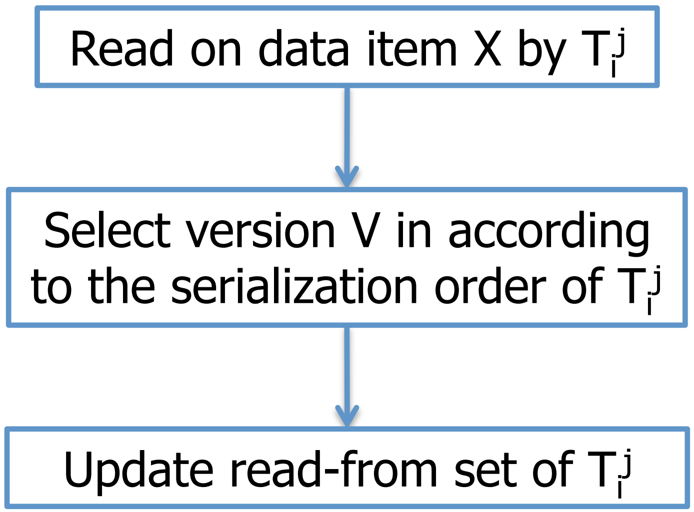
The snapshot miss events

+

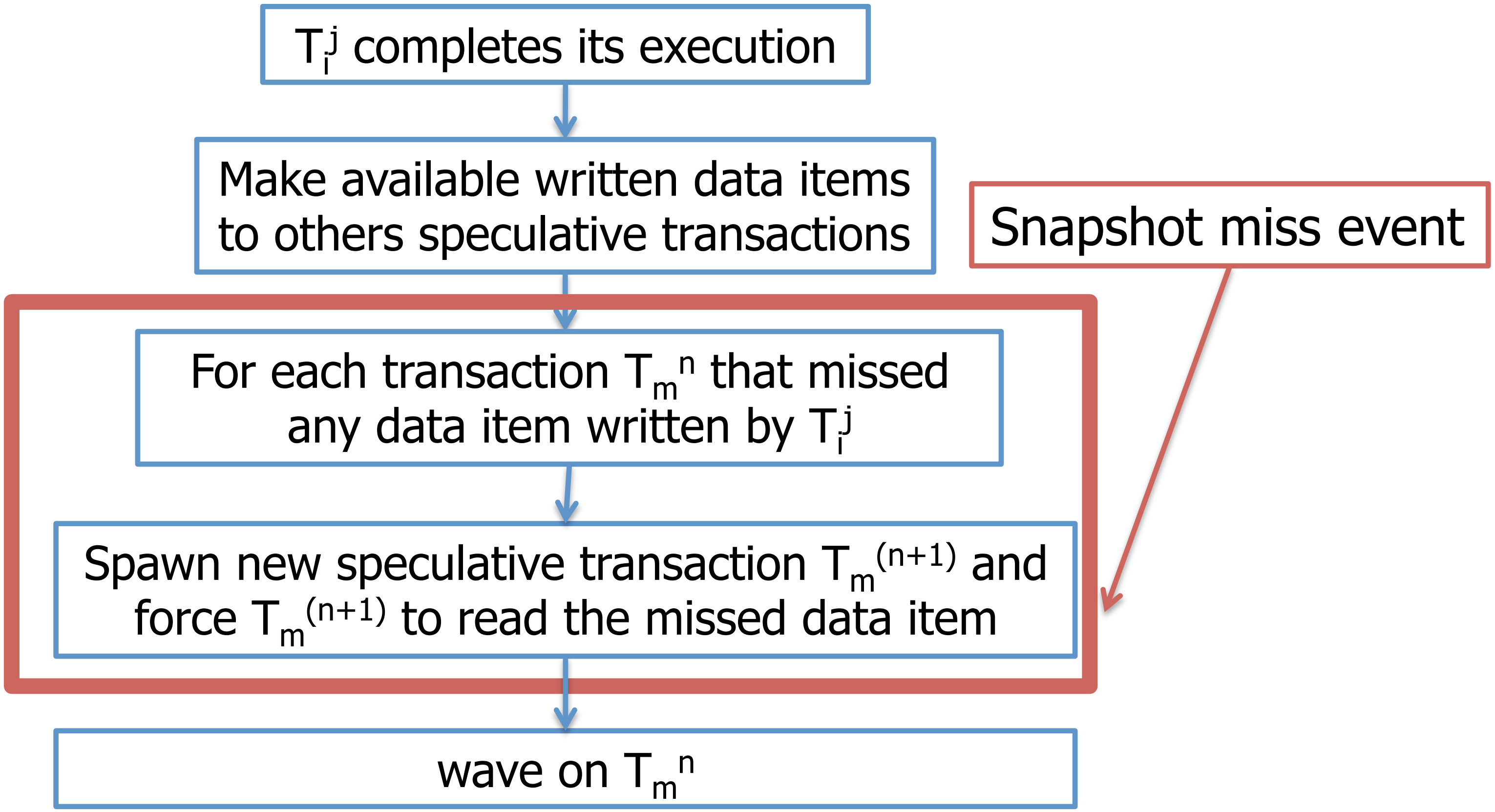
the optimistic delivery order

define the actual transaction serialization order that is stored within each transaction

Algorithm sketch: Read



Algorithm sketch: Complete



Algorithm sketch: Wave

When a transaction T_m^n misses a snapshot

↓

For each transaction T_l^f belonging to the read-form set of T_m^n

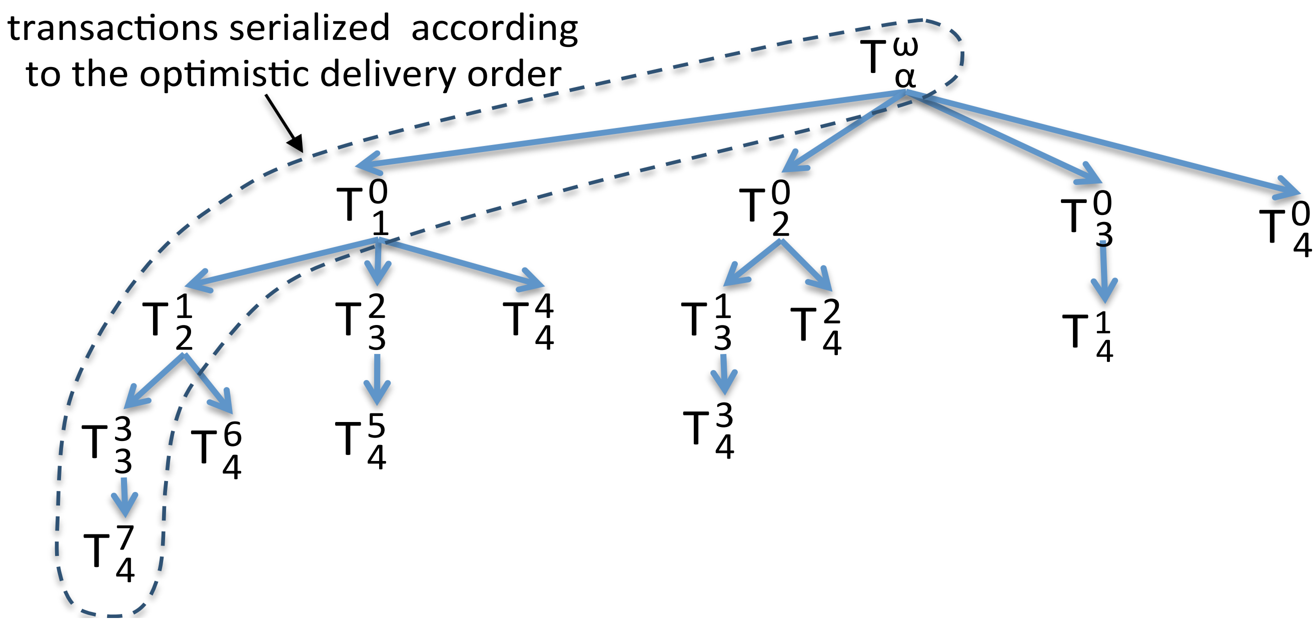
↓

Spawn new speculative transaction $T_l^{(f+1)}$

Wave (recursively) propagates the snapshot miss to all the transactions that read from T_m^n

OSARE Trajectory

- Largest set of explored speculative serialization orders: $O(2^n)$



Trace-based Simulator

- A discrete event simulator
- Realistic data access patterns
- Realistic optimistic atomic broadcast traces (coming from Appia Toolkit)
- Realistic probability to mismatch between optimistic and final deliveries

Simulation results

- Stamp Benchmarks
- OSARE Vs:
  - AGGRO

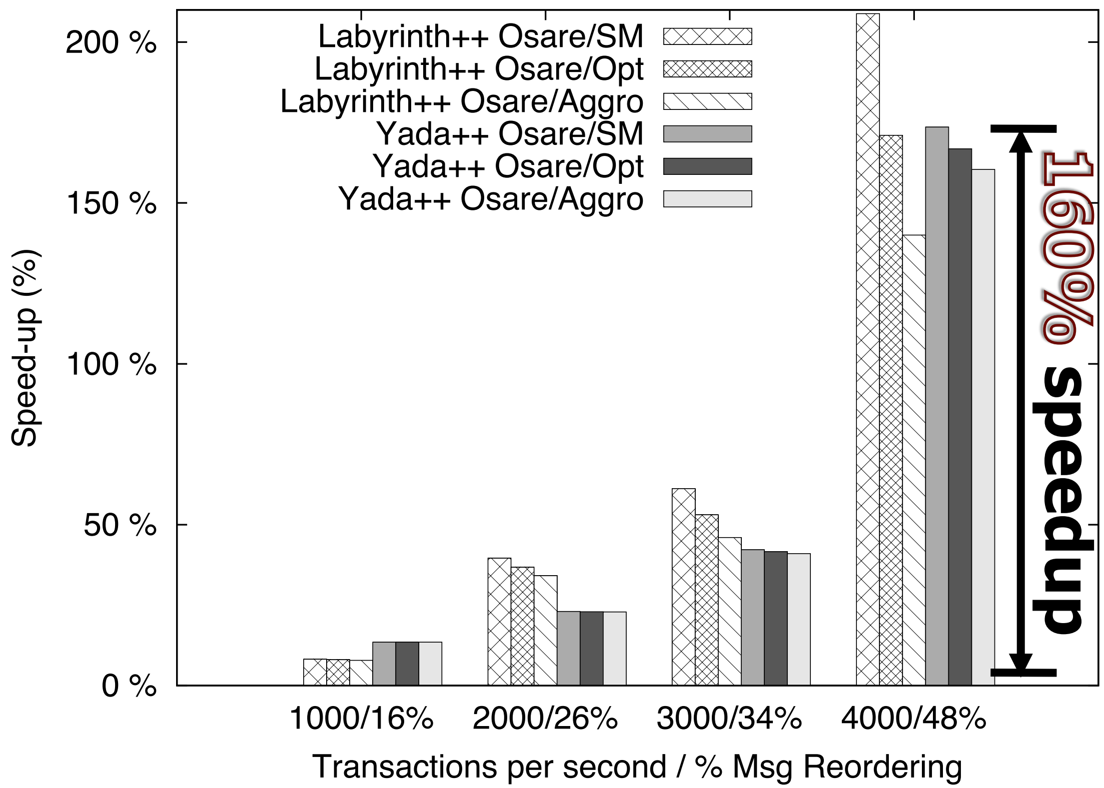
(aggressive at-most-one serialization order)

  - OPT

(at-most-one conflicting transaction)

  - SM

(No optimistic processing)



Thank you for the attention