**Roberto Palmieri**
Francesco Quaglia
*(La Sapienza, University of Rome, Italy)*

Paolo Romano
Nuno Carvalho
*(INESC-ID, Lisbon, Portugal)*

# Evaluating Database-Oriented Replication Schemes in Software Transacional Memory Systems

# STM
# Software Transactional Memory

- A Software Transactional Memory is a mechanism for controlling concurrent accesses to a shared memory
- STM systems represent an emerging attractive programming model for multi-core systems
- They mask concurrency management to the overlying applications
- They spare the programmers from the pitfalls of conventional manual lock-based synchronization, significantly simplifying the development of parallel and concurrent applications
- They are typically distributed via libraries to be included in the development of application programs

# Dependability of STM

- Replication is a typical mean for achieving transactional systems dependability
- Active Replication (AR) is a common replication scheme
- In AR each replica keeps the entire shared data-set and executes the same transactions in the same order
- AR is based on two phases:
  - agreement on common execution order -> consensus/atomic broadcast
  - deterministic execution on each replica -> filtering out sources of non-determinism

# Atomic Broadcast

- Atomic Broadcast (AB) is a service that ensures an agreement on a common Global Serialization Order of transactions among replicas
- AB is a paradigm used as a building block to run transactions in the same order on all replicas
- It is a particular instance of a Consensus protocol
- The coordination phase run by the AB protocol determines a latency that can heavily influence performance

# Optimistic Atomic Broadcast

- Optimistic Atomic Broadcast (OAB) is an improvement of the Atomic Broadcast service
- OAB enriches AB with a new event called optimistic delivery
- Participants acquire information about the existence of a message before any common order for message treatment gets defined
- The transactional system may early start processing activities, provided that these activities are effectively carried out on the basis of the guessed order
- OAB is particularly useful where the guess about message order is reliable (e.g. in local area network)

# Work’s targets

- Evaluate replication solutions, originally tailored for database systems, in the context of STM systems
- The system model is based on Active Replication implemented using an Optimistic Atomic Broadcast service
- It is also based on realistic STM settings
- In case of possible efficiency lacks, provide (qualitatively and quantitatively) ideas for filing these lacks
- Propose innovative approaches for addressing performance improvements in replicated STM systems

# DBMS & STMS
## Workload characterization

- The workload of applications in STM environments is different from DBMS applications
  - Transaction execution time in STM is typically several orders of magnitude smaller than in conventional database environments
  - Transactions in STMs entail main memory read/write operations, without interactions with stable storage
  - In replicated STM systems, the small transaction execution time leads to an amplification of the impact of the cost of the coordination phase among replicas
  - Longer stall periods in processing activities lead to a possible underutilization of hardware resources

# DBMS & STMS
## Data Layout

- The layout of data-sets in STM Systems is completely different from its counterpart in DBMS applications
  - The number of shared objects in STM Systems is typically very low compared to DBMSs
  - In STM systems it’s hard to group shared objects, unlike in DBMSs (e.g. tables or views)
  - Due to different data layouts, partial locking over the whole data set in STM systems is complex, unlike in DBMSs where locks can operate at the granularity of single tables
  - Implementation of concurrency control schemes based on knowledge (or safe estimation) of data sets accessed by transactions can lead in STM systems to locking all the objects within the shared memory

# Simulation study

- We compared two common active replication protocols:
  - State Machine Approach:
    - Based on Atomic Broadcast
    - The idea of this protocol is that each transaction waits until the atomic broadcast service notifies the transaction order, after the transaction is activated.
  - Optimistic Approach:
    - Based on Optimistic Atomic Broadcast
    - It requires a priori knowledge of read/write set of transactions
    - This protocol, exploiting optimistic delivery, early starts processing transactions that don’t conflict with each other. Conflicting transactions wait until the OAB notifies the final correct serialization order.

# Simulation study

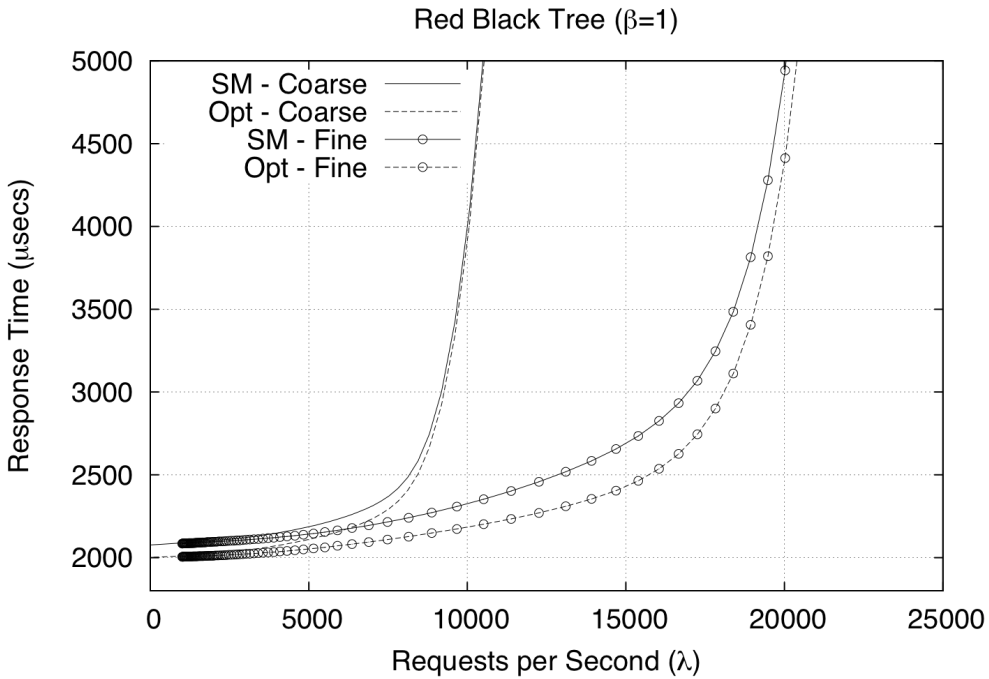
- Three benchmarks for STM Systems:
  - RB-Tree (Mean Transaction Execution Time: *77 microsec*)
  - SkipList (Mean Transaction Execution Time: *281 microsec*)
  - List (Mean Transaction Execution Time: *324 microsec*)
- Benchmarks are applications that perform repeated insertion, removal, and search of a randomly chosen integer in a set of pre-configurable integer values

# Simulation study

- Trace-based simulation with realistic data access patterns
- (Optimistic) Atomic Broadcast service entails the possibility of batching messages to improve performance
- Atomic Broadcast delays:
  - Optimistic delivery: *500microsec*
  - Final (total ordered) delivery: *2millisec*
- LAN environment with no mismatch between optimistic and final deliveries

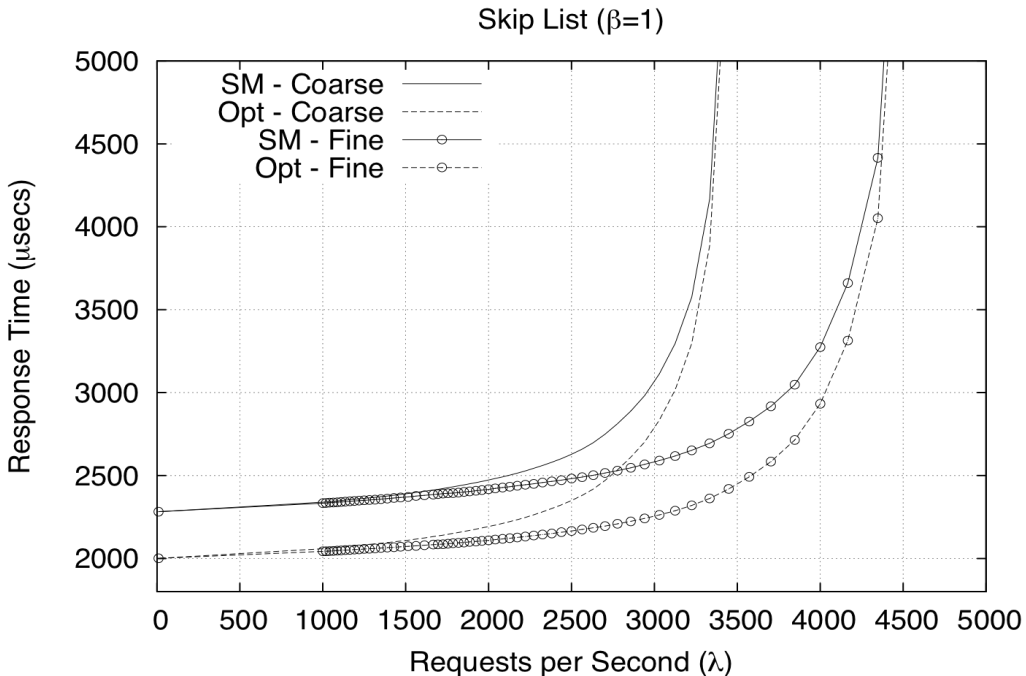
# RB-Tree results

- RB-Tree benchmark with medium data contention
- State Machine Vs Optimistic
  - realistic data access pattern of transactions (fine)
  - overestimation of lock granularity (coarse)



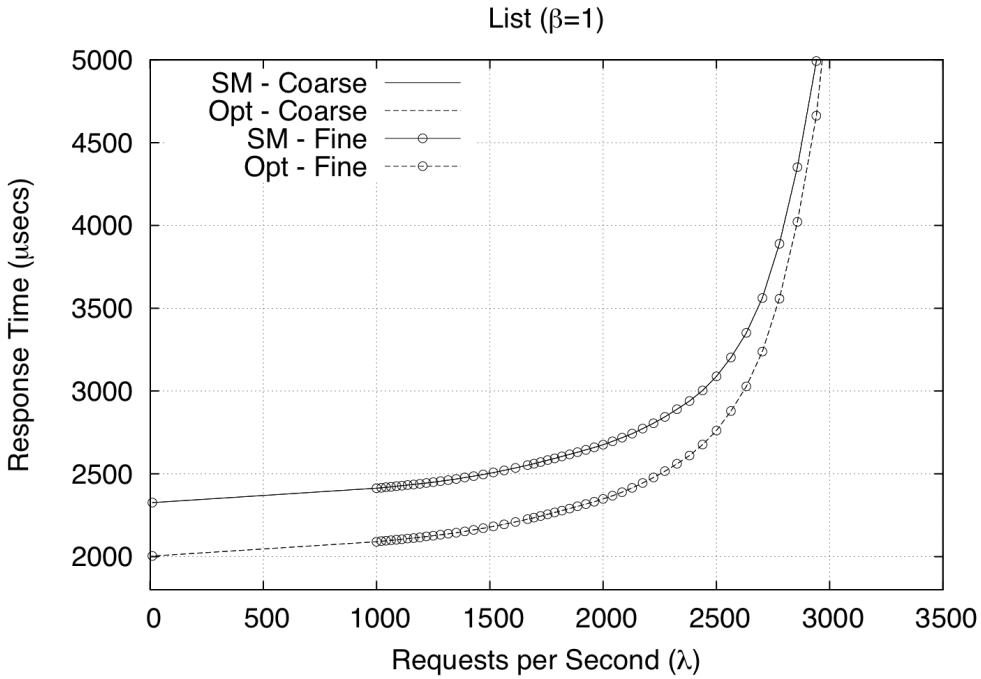
# SkipList Results

- SkipList benchmark with high data contention
- State Machine Vs Optimistic
  - realistic data access pattern of transactions (fine)
  - overestimation of lock granularity (coarse)



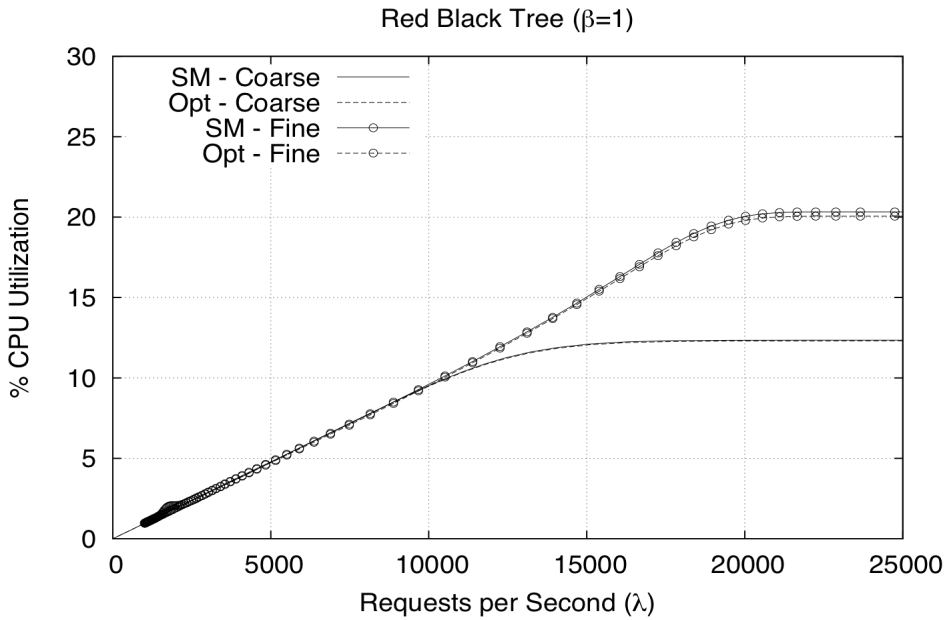
# List Results

- List benchmark with very high data contention
- State Machine Vs Optimistic
  - realistic data access pattern of transactions (fine)
  - overestimation of lock granularity (coarse)



# CPU

- Each replica is  composed by:
  - 8 cores per replica
  - Processor: 2.53Ghz
  - RAM:4Gb
- At the saturation point, CPU utilization is < 20%



# Considerations

- STM workloads manifest, on average, high data contention respect to DBMS workloads
- In such a case, the optimistic protocol blocks most of the transactions and this is reflected in :
  - minimization of the overlap between coordination and processing
  - underutilization of resources
  - scalability problems

# Possible approaches

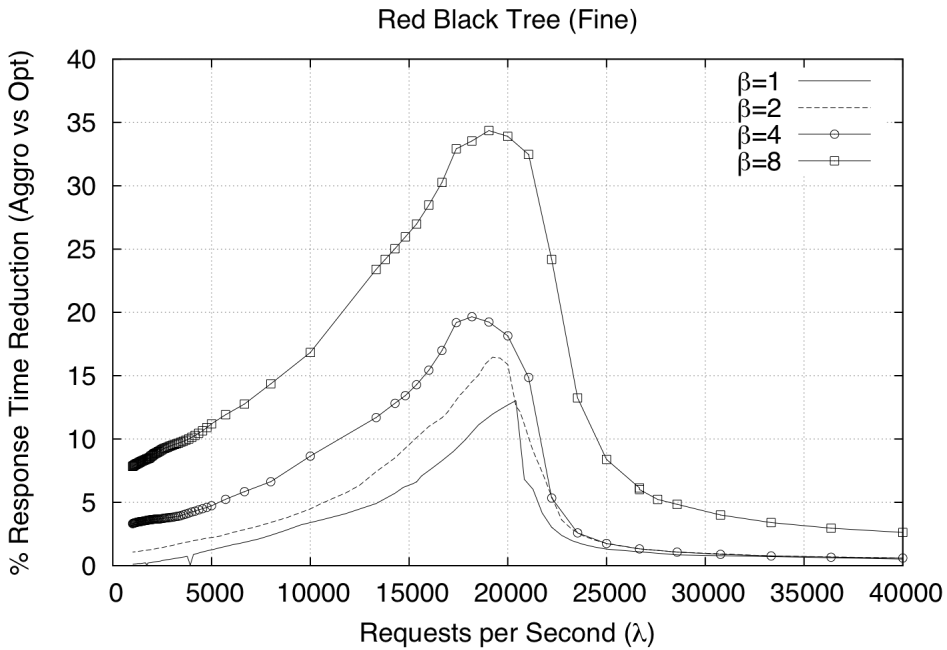
- Increase the optimism while processing transactions that are not yet final delivered
- Maximize the overlap between replicas coordination and local process of transactions
- Avoid protocols that need knowledge about read/write set of transactions
- Avoid coarse approaches that can boil down to locking all the shared objects

# Boosting
# the optimistic approach

- Capture the event of complete transaction
- When a transaction performs all its operations and waits for the notification of the final order (completes), releases all its locks
- With the early released of locks, a conflict transaction can be started according to the optimistic delivery order
- Especially useful when the guess on the optimistic delivery order is reliable

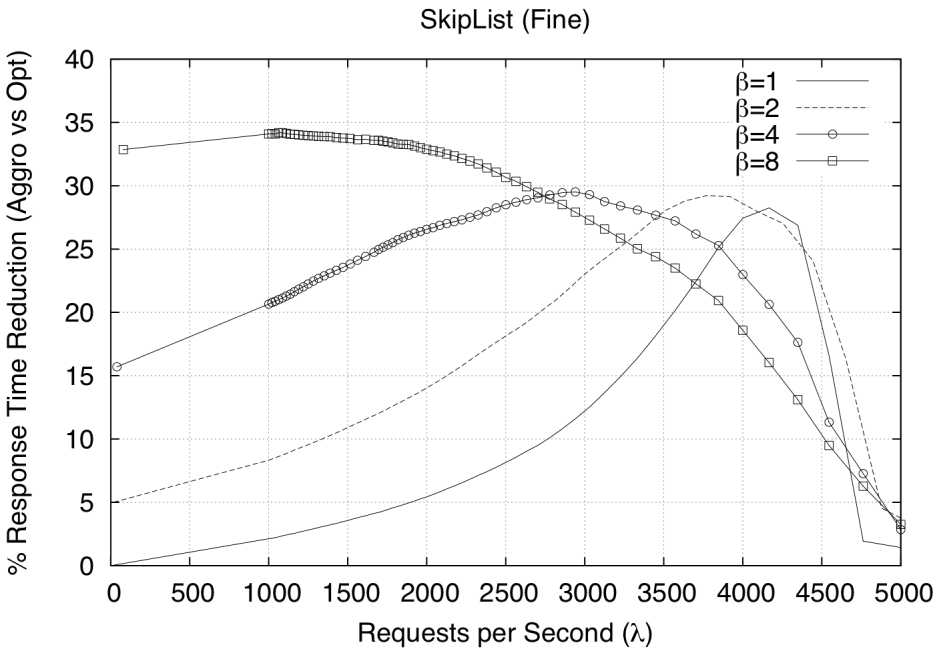
# RB-Tree results

- RB-Tree benchmark
- Atomic Broadcast service batched (1,2,4,8)
- We show the gain from the Boosting protocol (called AGGRO) Vs standard Optimistic approach



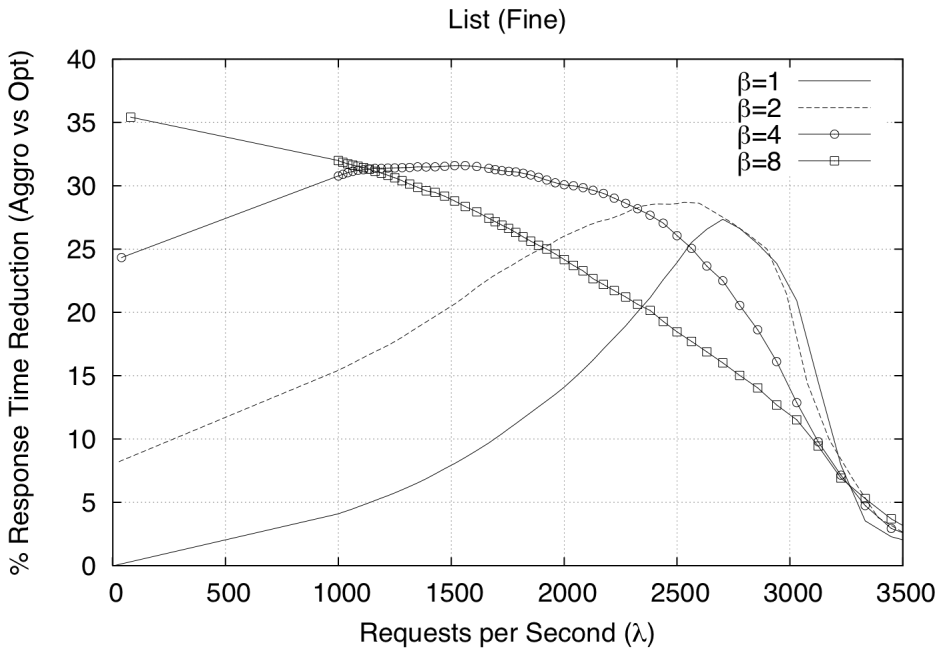
# SkipList Results

- SkipList benchmark
- Atomic Broadcast service batched (1,2,4,8)
- We show the gain from the Boosting protocol (called AGGRO) Vs standard Optimistic approach



# List Results

- List benchmark
- Atomic Broadcast service batched (1,2,4,8)
- We show the gain from the Boosting protocol (called AGGRO) Vs standard Optimistic approach



Thank you for your attention