Pastramy

Persistent and highly Available Software TRansactional MemorY

Paolo Romano, Nuno Carvalho, João Cachopo, Luís Rodrigues

Roadmap

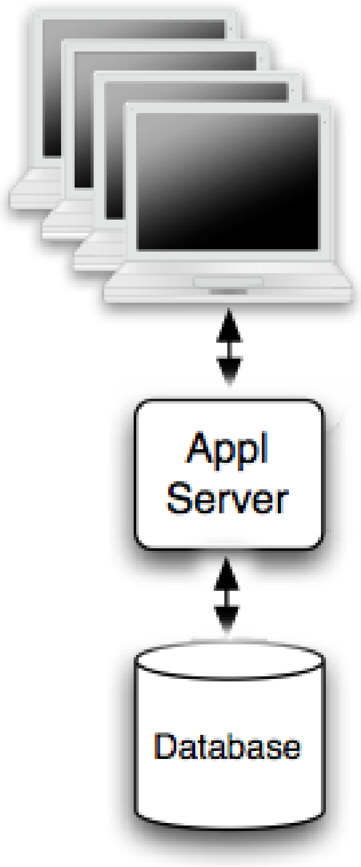
- The FénixEDU System
- The PASTRAMY Project
- Software Transactional Memory Replication
  - Critical issues
  - Current research directions

The FénixEDU System

- Open source project to support all the academic processes of an university e-campus:
  - already used in ~10 universities
  - three-tier J2EEWeb application
    - **first production system to rely on Software Transactional Memories (STMs)**

- Real-life system raising challenging research issues!

High level FénixEDU Architecture



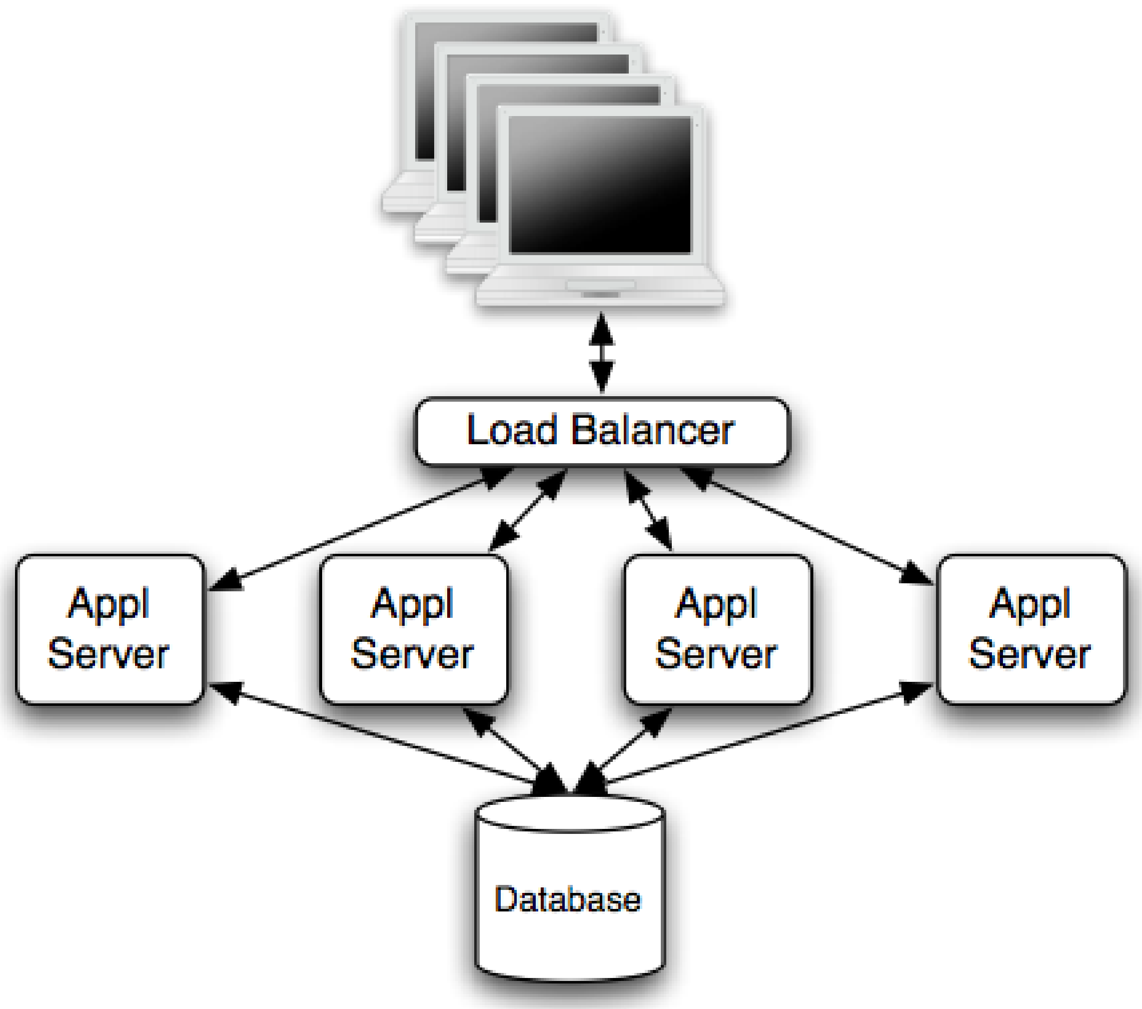
Application Server

- in-memory object oriented domain model
- concurrent transactions transparently synchronized by a *Software Transactional Memory*, **JVSTM**:
    - multi-versioned, lock-free
    - guarantees atomicity and isolation

Back-end Database

- ensures data durability
- overcomes application servers' memory capacity constraints

Current FénixEDU Architecure



- Replicated application servers

- Replica synchronization achieved through a centralized validation at the back-end database

Open Problems

- Interface with relational DBMS consumes an excessive amount of memory

- DBMS is the system's bottleneck and single point of failure

Pastramy Project

- Collaborative project among INESC-ID, U. of Minho and U. of Lisboa

- **Goals:** improve performance and reliability of the FènixEDU system by means of:
  - efficient **application server replication**
  - ad-hoc, lightweight **storage system**

Our Research Focus

- Our focus is on designing high performance replication schemes for STM systems.

Key Observation

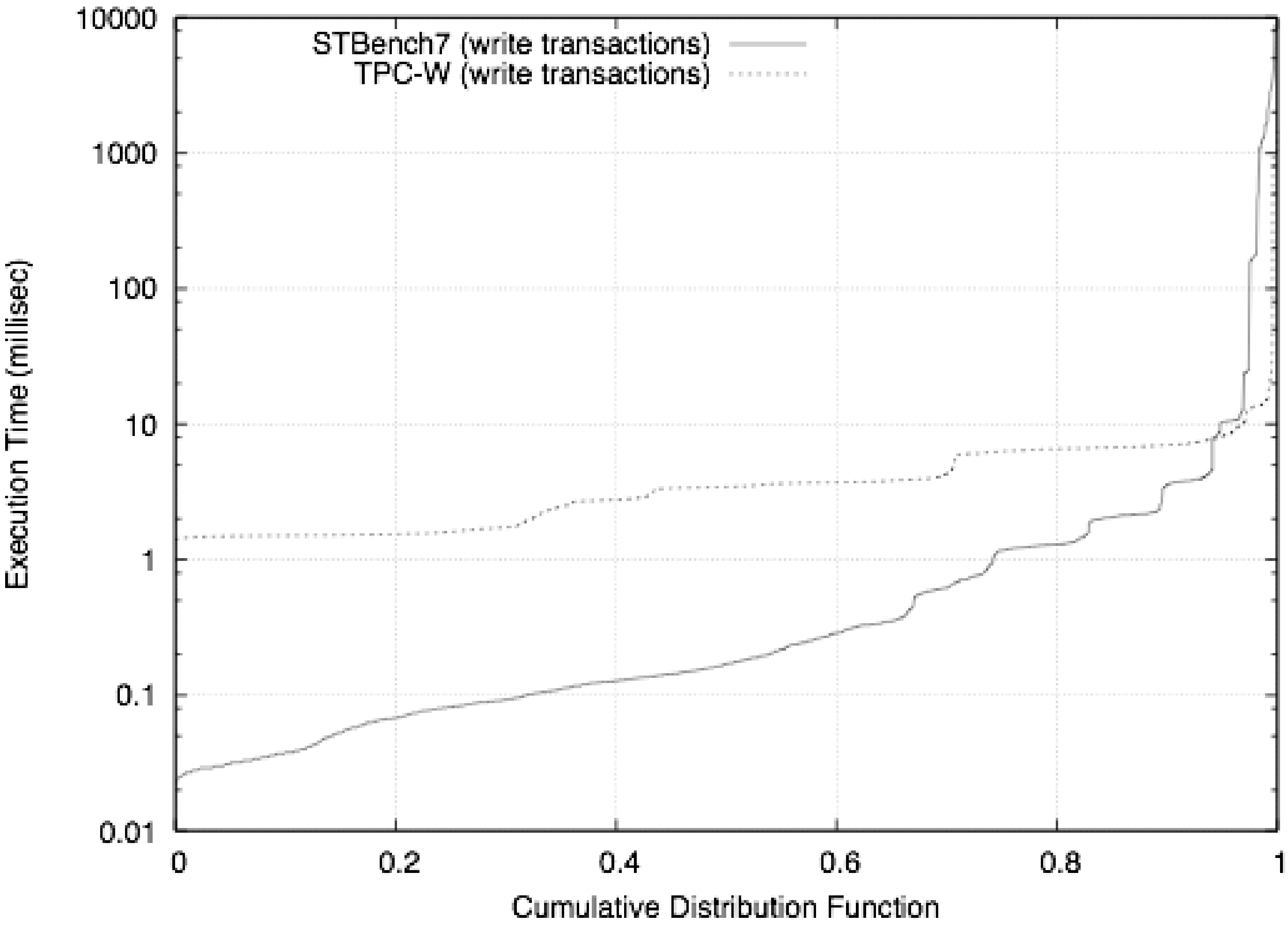
- Databases and STMs share the same fundamental notion of atomic transaction...
- ...database replication schemes represent a natural starting point for STM replication as well!

A (very brief) recall of recent database replication schemes

- Recent database replication solutions rely on Atomic Broadcast (AB) to establish global transaction serialization order
  - certification based, state-machine, ...
- Relatively high AB latency amortized by considerable transaction execution costs
  - SQL parsing, optimization of the execution plan, ...

Critical Issues for STM Replication

- >70% of transactions are 10-100 times shorter in STMs :
    - correspondingly larger impact of AB overhead!
- Transactions' lifetime span a much wider range in STMs:
    - no “one size fits all” solutions!



Current Research Directions

Speculative Transaction Execution

Problem: high AB latency causing CPU underutilization

Idea: employ idle CPUs to explore alternative transaction serialization orders

Challenges:

- not trivial integration with existing AB-based replication schemes and STM's concurrency control mechanisms
- identify effective heuristics to efficiently explore the O(n!) possible serialization orders

Space efficiency via Bloom Filters

Problem:

- most efficient AB-based replication schemes require exchanging transactions readsets...
- ...which can be huge, drastically affecting the AB latency!

Idea: exploit Bloom Filters space efficient encoding to deterministically limit the message size

Challenge: (efficiently) accommodating unavoidable false positives

Self-adapting Replication Strategies

Problem: No single replication protocol is able to optimally cope with the high heterogeneity of STM based systems

Idea: Develop STM replication protocols able to self-adapt depending on, e.g.:

- transaction's object-set size
- estimated transaction conflict probability
- ...

Challenge: allow consistent coexistence of multiple replication schemes

Lease Based Replication Schemes

Problem: AB is extremely costly in STM environments

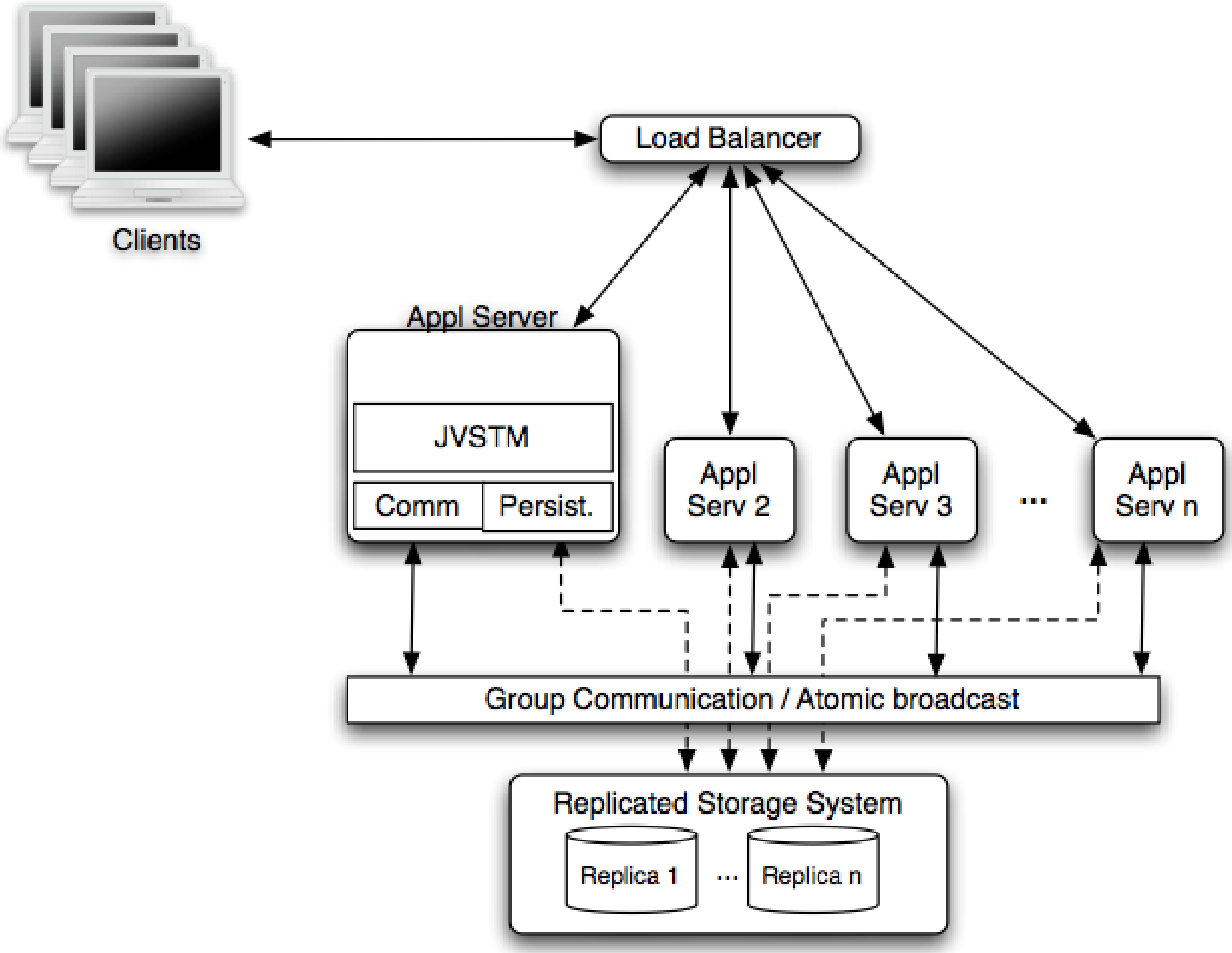
Idea: Reduce frequency of AB-based synchronization through leases

- At transaction's commit time, AB can be used:
  - not only to establish the serialization order....
  - but also to obtain a **lease** on the accessed data items
- Replicas already owning a lease avoid AB and simply propagate updates in FIFO order

Challenge: maximize lease re-usage VS load balancing fairness

Thanks for
the attention

Expected Future Architecture


Clients
Load Balancer
Appl Server
JVSTM
Comm
Persist.
Appl Serv 2
Appl Serv 3
...
Appl Serv n
Group Communication / Atomic broadcast
Replicated Storage System
Replica 1
...
Replica n