Self-optimizing transactional data grids for elastic cloud environments

Paolo Romano

inesc id lisboa

About me

- Master and PhD from “Sapienza” University of Rome
- Researcher at Distributed Systems Group, INESC-ID, Lisbon (since 2008)
  - **Best INESC-ID Young Researcher 2011**
- Invited professor at Instituto Superior Técnico, Lisbon (since 2011)

Some international projects in which I am currently involved:

- Coordinator of the FP7 Cloud-TM Project (Jun 2010-Jun2012)
  - 4 international partners from industry and academy
- Coordinator of the Cost Action Euro-TM (fall 2010-fall 2013)
  - Pan-European Research network on Transactional Memories
  - 56 experts, 42 institutions, 12 countries

Talk overview

- Cloud-TM Overview:
  - key goals
  - background on Transactional Memories
  - progresses so far
- Self-optimizing transactional data grids:
  - methodologies explored so far
  - case studies
- Open research questions & future work

Cloud-TM at a glance

Partners:

INESC ID (PT)

Algorithmica (IT)

C.I.N.I. (IT)

Red Hat (IE)

Project coordinator:
Paolo Romano, INESC ID (PT)

Duration:
From June 2010 to May 2013

Programme:
FP7-ICT-2009-5 – Objective 1.2

Further information:
http://www.cloudtm.eu

Key Goals

Develop a transactional data platform for the Cloud:

1. Providing a simple and intuitive programming model:
   - hide complexity of distribution, persistence, fault-tolerance

2. Minimizing administration and monitoring costs:
   - automate elastic resource provisioning based on applications QoS requirements

3. Minimize operational costs via self-tuning
   - maximize efficiency adapting consistency mechanisms upon changes of workload and allocated resources

Background on the Cloud-TM Programming Paradigm....

TRANSACTIONAL MEMORIES

Cloudviews 2011, Porto, Portugal, Nov. 4 2011

Transactional Memories…

- Transactional Memories (TM):
  - replace locks with atomic transactions <u>in the programming language</u>
  - hide away synchronization issues from the programmer
    - avoid deadlocks, priority inversions, debugging nightmare
    - simpler to reason about, verify, compose
  - **simplify development of parallel applications**

…to Distributed Transactional Memories...

- Distributed Transactional Memories (DTM):
  - extends TM abstraction over the boundaries of a single machine:
    - enhance scalability
    - durability via in-memory replication
  - minimize communication overhead via:
    - speculation
    - batching consistency actions at commit-time

…to Cloud-TM

Open-source DTM middleware providing:

- Language level support for:
  - object-oriented domain model
  - highly scalable abstractions

- Elastic scale-up and scale-down of the DTM platform:
  - automatic scaling based on user defined QoS & cost constraints

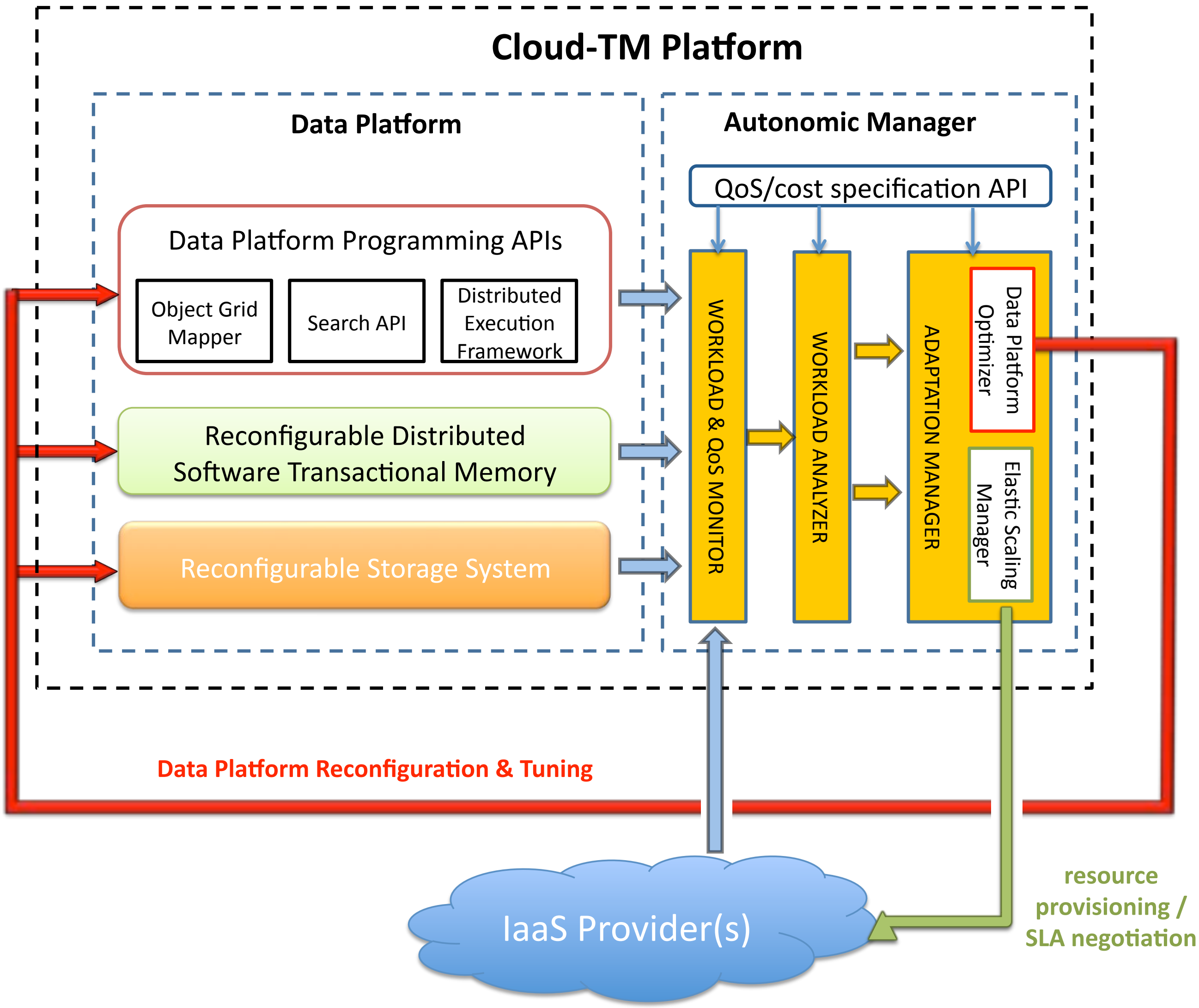
- Self-optimization as a pervasive feature:
  - pursue maximum efficiency via cross-layer self-tuning

PROGRESSES SO FAR

Main achievements

- Architecture specification
- Development of preliminary prototype
- Innovative transactional replication schemes
- Platform self-tuning

Architecture Specification



Cloudviews 2011, Porto, Portugal, Nov. 4 2011

Preliminary prototype

- 1st version of Data Platform already available: http://www.cloud-tm.eu
- Integration/extension of mainstream open source projects:
  - focus on innovation & avoid reinventing the wheel
  - maximize project's visibility & facilitate exploitation

Innovative transactional replication schemes

- Several approaches have been pursued:
  - overlap processing and communication via speculation
    [SPAA2010, ISPA2010, NCA2010, SYSTOR2011,SRDS2011]
  - asynchronous leases to reduce communication overhead
    [Middleware2010]
  - weaker consistency models
    [PRDC2011]
- Different approaches with a same common goal:

Data-grid self-optimization

- Self-tuning/performance forecasting of several platform layers
  - Software Transactional Memory layer
    [CMG10], [PEVA11]
  - Replication manager
    [Middleware2011]
  - Group communication system
    [SASO10], [Performance2011],[ICNC12]

Talk overview

- Cloud-TM Overview:
  - key goals
  - background on Transactional Memories
  - progresses so far
- Self-optimizing transactional data grids:
  - methodologies explored so far
  - case studies
- Open research questions & future work

Methodologies explored so far

- Analytical modeling:
  - queuing theory, markov processes
  - stochastic techniques
- Machine learning:
  - off-line techniques:
    - Decision Trees, Neural networks, Support Vector Machine
  - on-line techniques (reinforcement learning):
    - UCB algorithm

Analytical modeling

- white box approach:
  - requires detailed knowledge of internal dynamics
- good extrapolation power:
  - allow forecasting system behavior in unexplored regions of its parameters' space ☺
- minimal learning time:
  - basically parameters instantiation ☺
- complex and expensive to design/validate ☹
- subject to unavoidable approximation errors ☹

Machine learning

- black box approach:
  - observe inputs, context and outputs of a system
  - use statistical methods to identify patterns/rules
- good accuracy in already explored regions of the parameters' space ☺
- …but poor extrapolation power ☹
- learning time grows exponentially with number of features:
  - but eventually outperforms analytical models (typically!)

Hybrid techniques

IDEA: get the best of the two worlds

Two alternative approaches so far:

1. Divide-and-conquer:
   - AM for well-specified sub-components
   - ML for sub-components that are:
     - too complex to model explicitly, or
     - whose internal dynamics are only partially specified
2. Use AM to initialize ML knowledge:
   - reduce learning time of ML techniques
   - correct AM using feedback from operational system

Case studies

- Dynamic selection and switching of data replication protocols:
  - total order based replication protocols (Case study 1):
    - purely based on Machine Learning techniques
  - single-master vs multi-master (Case study 2):
    - hybrid ML-AM solution – divide-et-impera

- Group Communication System self-optimization:
  - batching in total order protocols (Case study 3)
    - hybrid ML-AM – ML bootstrapped with AM knowledge

SELF-OPTIMIZING DATA CONSISTENCY PROTOCOLS

Cloudviews 2011, Porto, Portugal, Nov. 4 2011

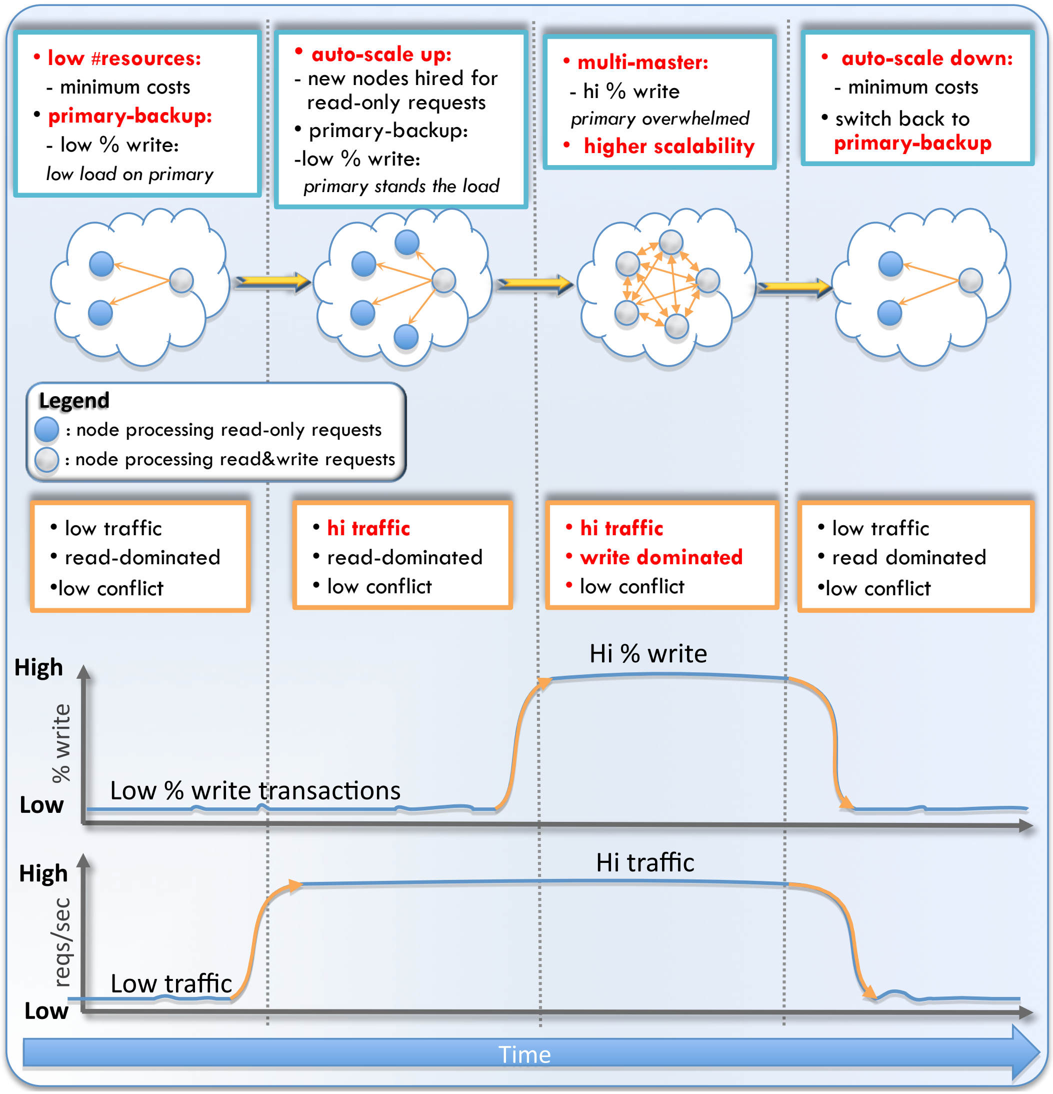
The search for the holy grail

transactional data
consistency protocols

- Single master (primary-backup)
- Multi master
  - Total order-based
    - Certification
      - Non-voting
      - BFC
      - Voting
    - State machine replication
  - 2PC-based

NO ONE SIZE FITS ALL SOLUTION

The Cloud-TM vision



Self-optimizing data replication: key challenges

1. allow efficient switch among multiple replication protocols:
   – avoid blocking transaction processing during transitions

2. determine the optimal replication strategy given the current workload characteristics:
   – machine learning methods (black box)
   – analytical models (white box)
   – hybrid analytical/statistical approaches (gray box)

Case studies

- Dynamic selection and switching between replication protocols:
  - total order based replication protocols (Case study 1):
    - purely based on Machine Learning techniques
  - Two phase commit vs primary backup (Case study 2):
    - hybrid ML-AM solution – divide-et-impera
- Group Communication System self-optimization:
  - batching in total order protocols (Case study 3)
    - hybrid ML-AM – ML bootstrapped with AM knowledge

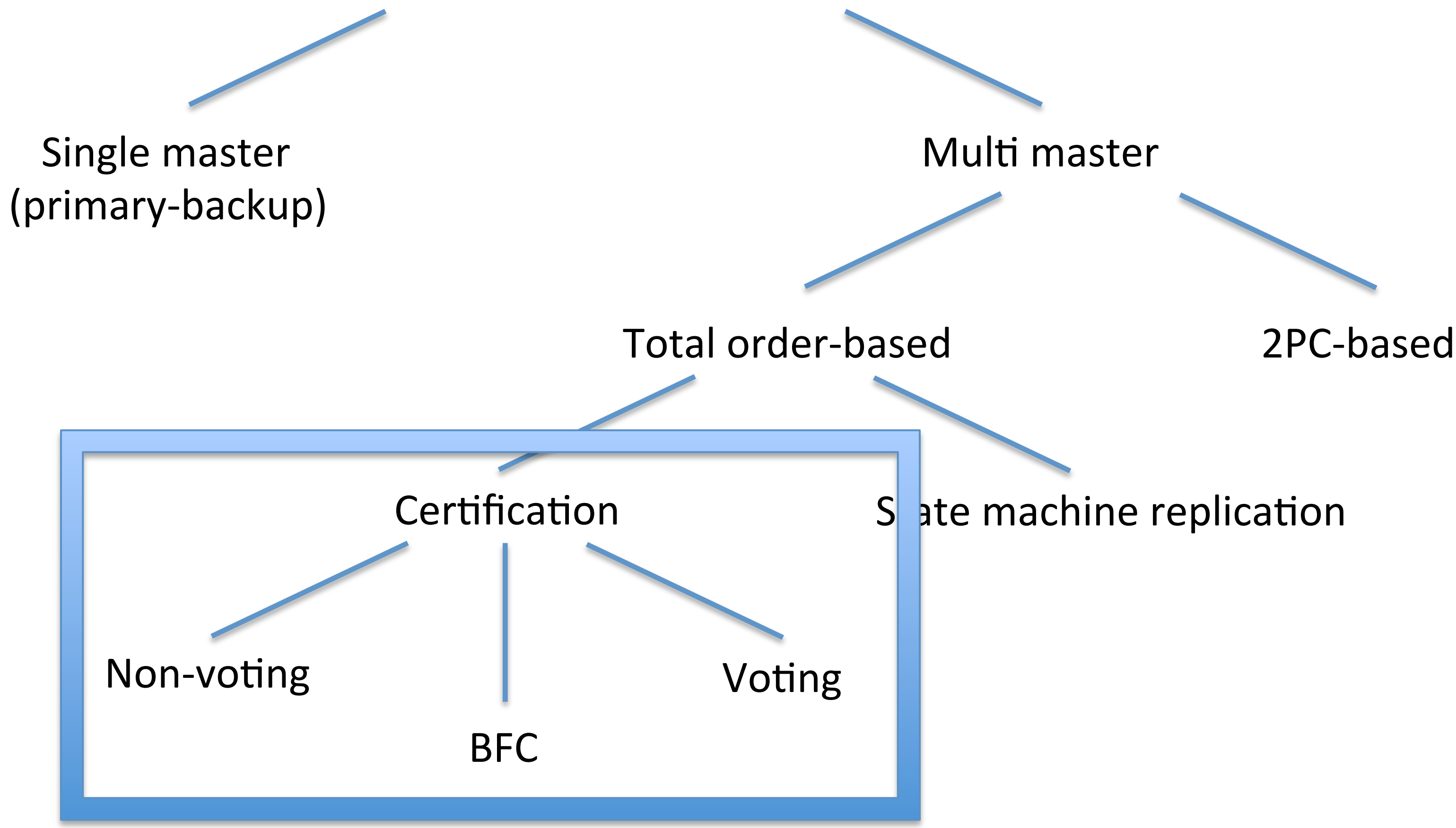
Maria Couceiro, Paolo Romano, Luis Rodrigues

ACM/IFIP/USENIX 12th International Middleware Conference (Middleware 2011)

POLYCERT: POLYMORPHIC SELF-OPTIMIZING REPLICATION FOR IN-MEMORY TRANSACTIONAL GRIDS

Cloudviews 2011, Porto, Portugal, Nov. 4 2011

Where they fit in the picture



Certification
(a.k.a. deferred update)

- A transaction is executed independently at a single replica until its commit phase:
  - minimize network traffic
- Distributed certification is run to detect conflicts with transactions executed concurrently at different replicas
- Certification is typically much more lightweight than full transaction execution
  - good scalability also in write intensive workloads

Certification

- Three different approaches in literature:
  - Non-voting algorithm
  - Voting algorithm
  - BFC
- One key commonality:
  - reliance on Total Order Broadcast to avoid distributed deadlocks
  - TOB ensures agreement on delivery order of broadcast messages in a non-blocking fashion

Classic Replication Protocols

- Focus on full replication protocols

Single master (primary-backup)

Multi master

AB-based

2PC-based

Certification

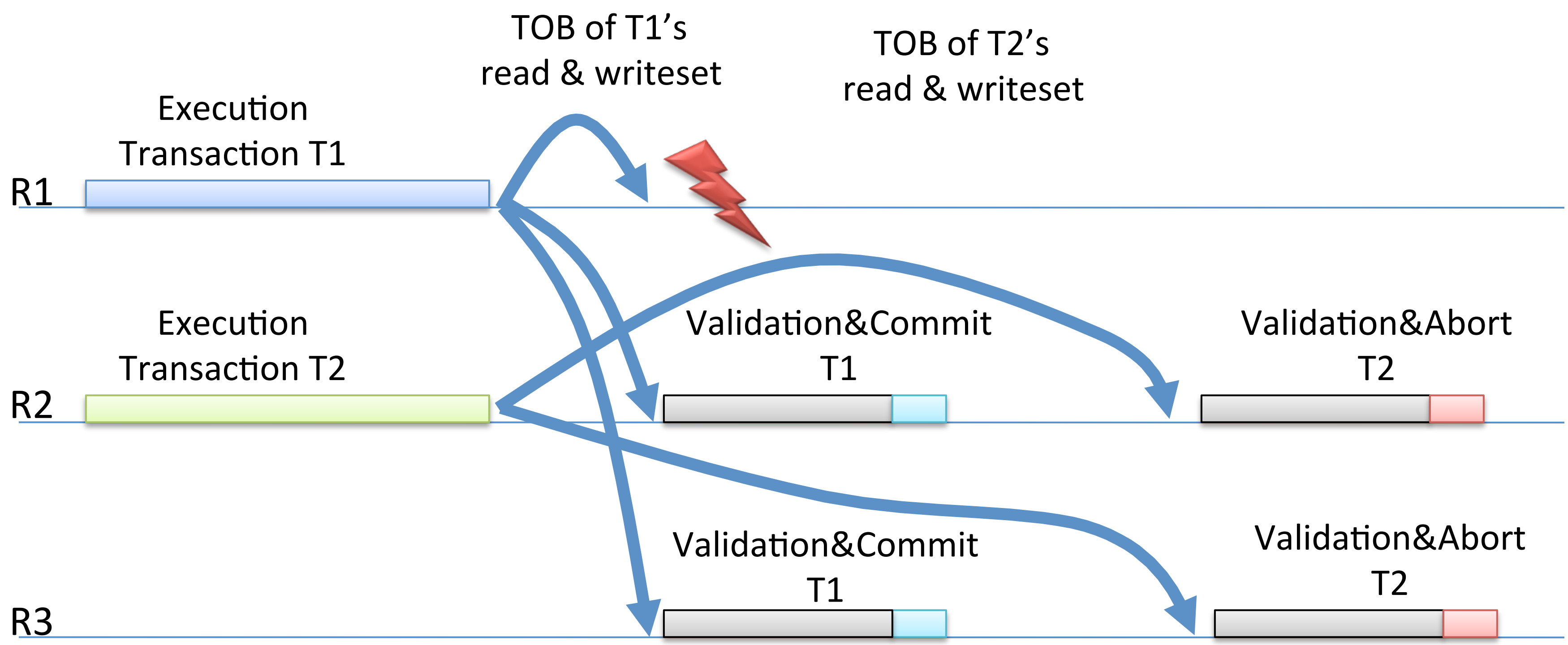
State machine replication

Non-voting

BFC

Voting

Non-voting



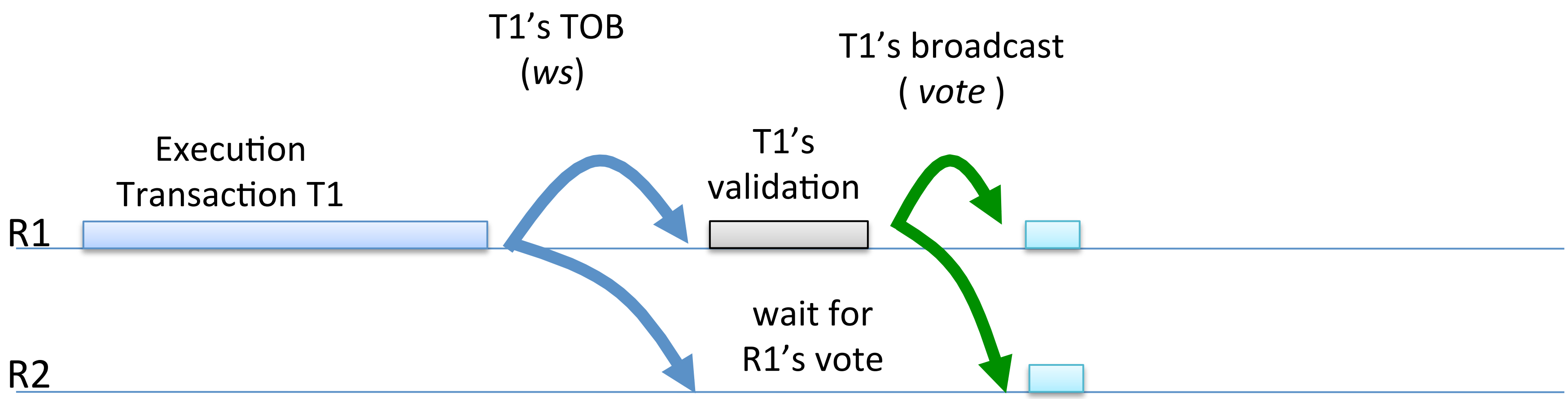
+ *only validation executed at all replicas:*
high scalability with write intensive workloads

- *need to send also readset: often very large!*

Classic Replication Protocols

- Focus on full replication protocols
  - Single master (primary-backup)
  - Multi master
    - AB-based
      - Certification
        - Non-voting
        - BFC
        - Voting
      - State machine replication
    - 2PC-based

Voting

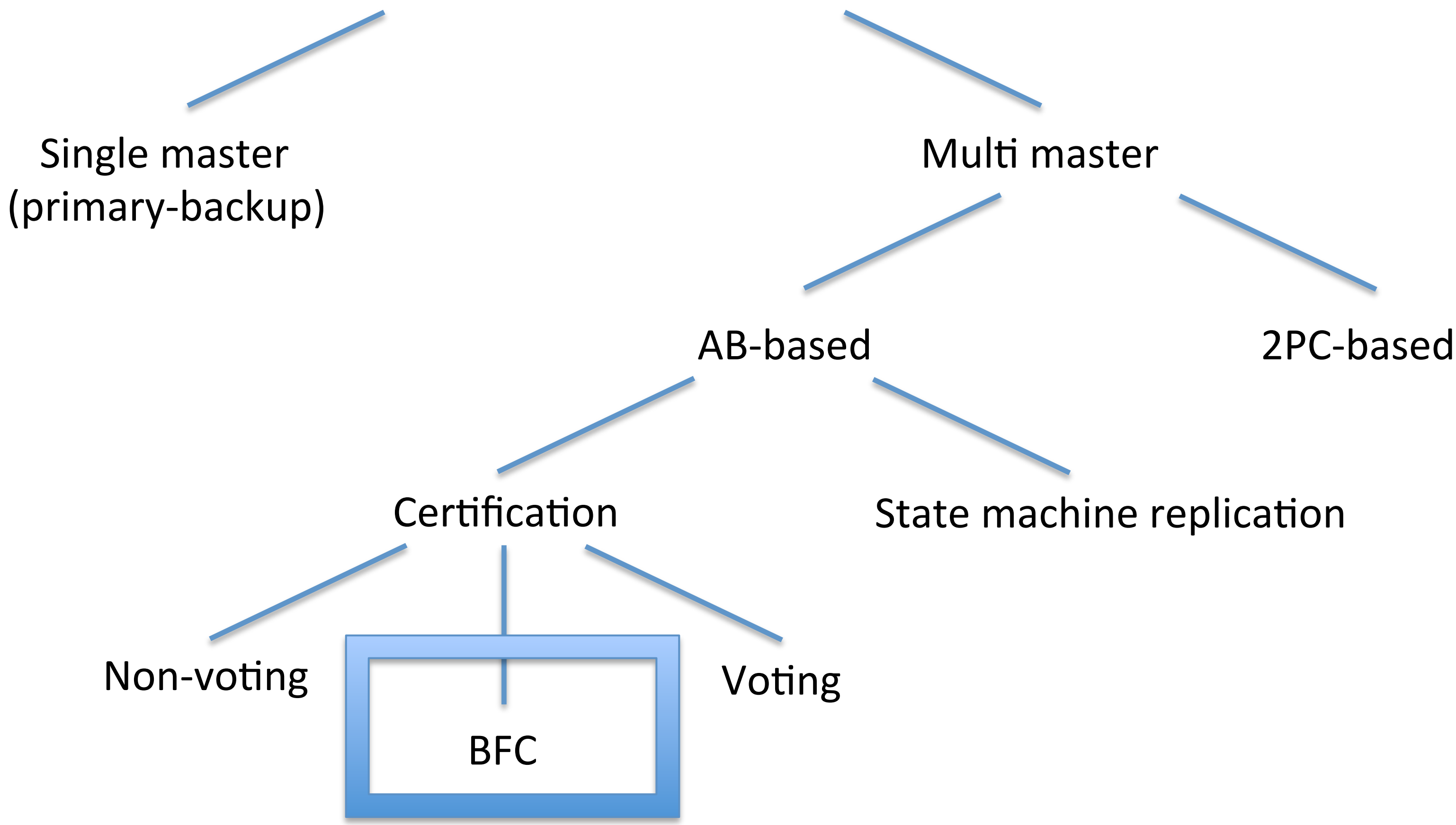


+ *sends only write-set (much smaller than read-sets normally)*

- *Additional communication phase to disseminate decision (vote)*

Classic Replication Protocols

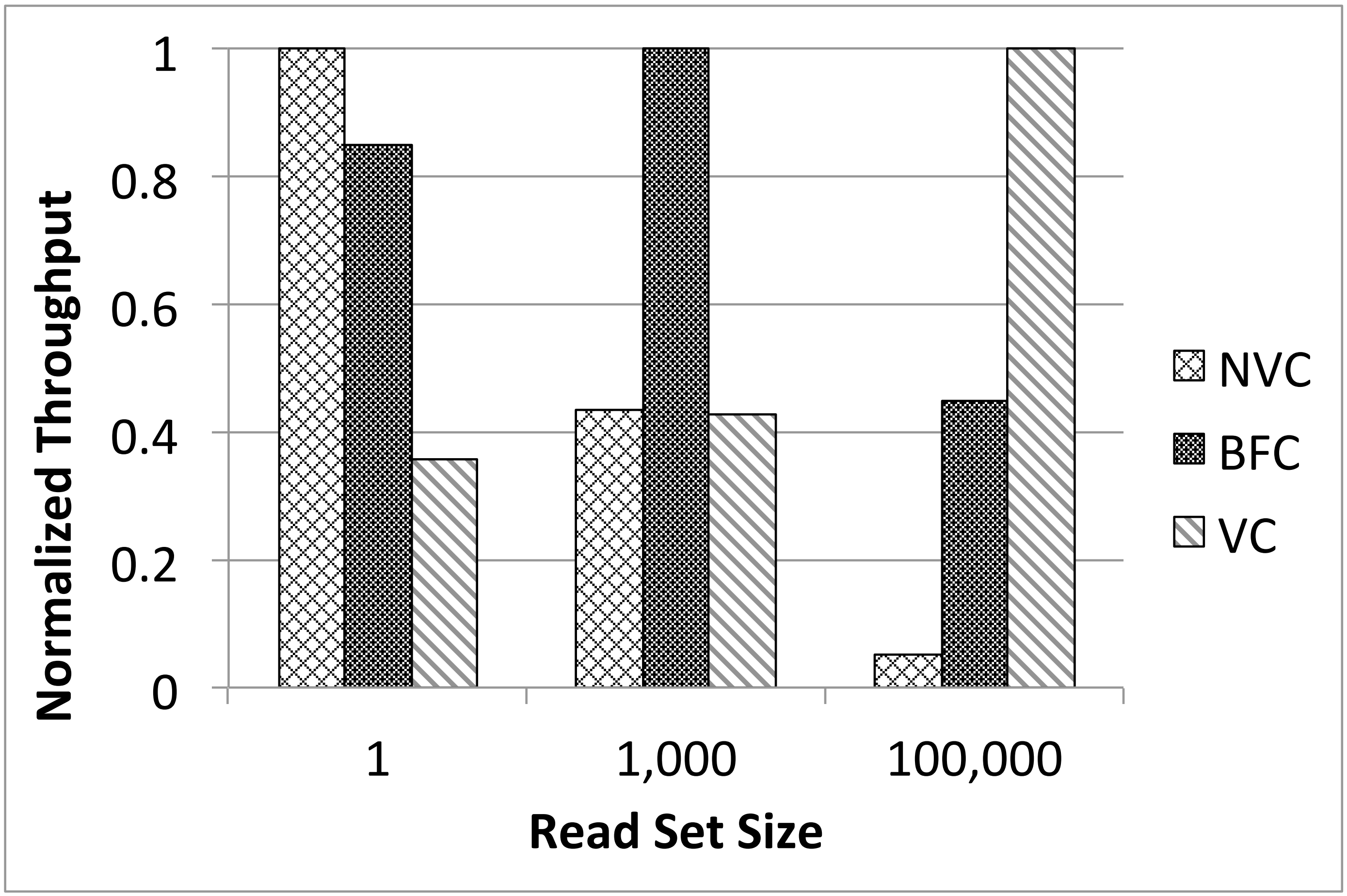
- Focus on full replication protocols



Bloom Filter Certification (BFC)

- Use a single TOB as in non-voting, but encode readset in a Bloom filter
  - Bloom filters:
    - space-efficient, probabilistic data structure for test membership
    - compression is a function of a (tunable) false positive rate
- A transaction T is certified successfully only if:
  - none of the items written by concurrent transactions is present in the BF used to encode T's readsed
  - strongly reduce network traffic at the cost of negligible abort increase
    - 1% false positive yields up to 30x compression

BFC vs Voting vs Non-Voting



Self-optimizing data replication: key challenges

1. allow efficient switch among multiple replication protocols:
   – **coexistence** of multiple certification schemes via the Polymorhic Certification (PolyCert) protocol

2. determine the optimal replication strategy given the current workload characteristics:
   – entirely based on **machine learning** techniques
      • off-line: decision-trees, neural network, SVM
      • on-line: reinforcement learning (UCB)

PolyCert

- Polymorphic Self-Optimizing Certification

- Co-existence of the 3 certification schemes:
  - exploit common reliance on total order broadcast

- Machine-learning techniques to determine the optimal certification strategy per transaction

Replication Protocol Selector Oracle

- Two implementations:
  - Off-line Machine Learning Techniques
  - On-line Reinforcement Learning

Off-line Machine Learning Techniques

- For each transaction:
  - Determine size of exchanged messages for each certification scheme
  - Forecast AB latency for each message size. We evaluated several ML approaches:
    - Regression decision trees (best results)
    - Neural networks
    - Support vector machine

Off-line Machine Learning Techniques

- Uses up to 53 monitored system attributes:
  - CPU
  - Memory
  - Network
  - Time-series

- Requires computational intensive feature selection and training phase

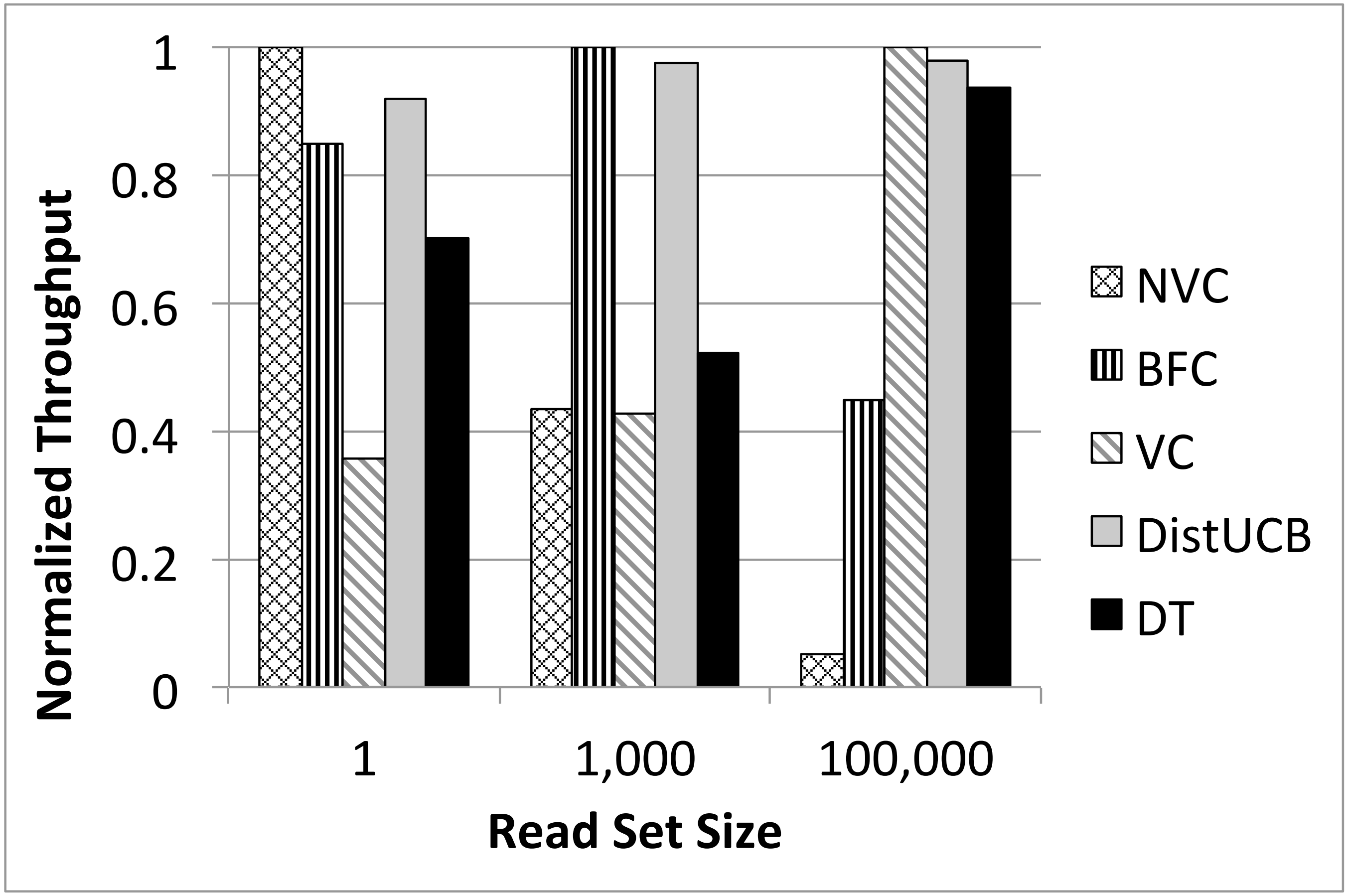
On-line Reinforcement Learning

- Each replica builds on-line expectations on the rewards of each protocol:
  - no assumption on rewards' distributions
- Upper Confidence Bound (UCB) algorithm:
  - lightweight and provably optimal solution to the exploration-exploitation dilemma:
    - did I test this option sufficiently in this scenario?

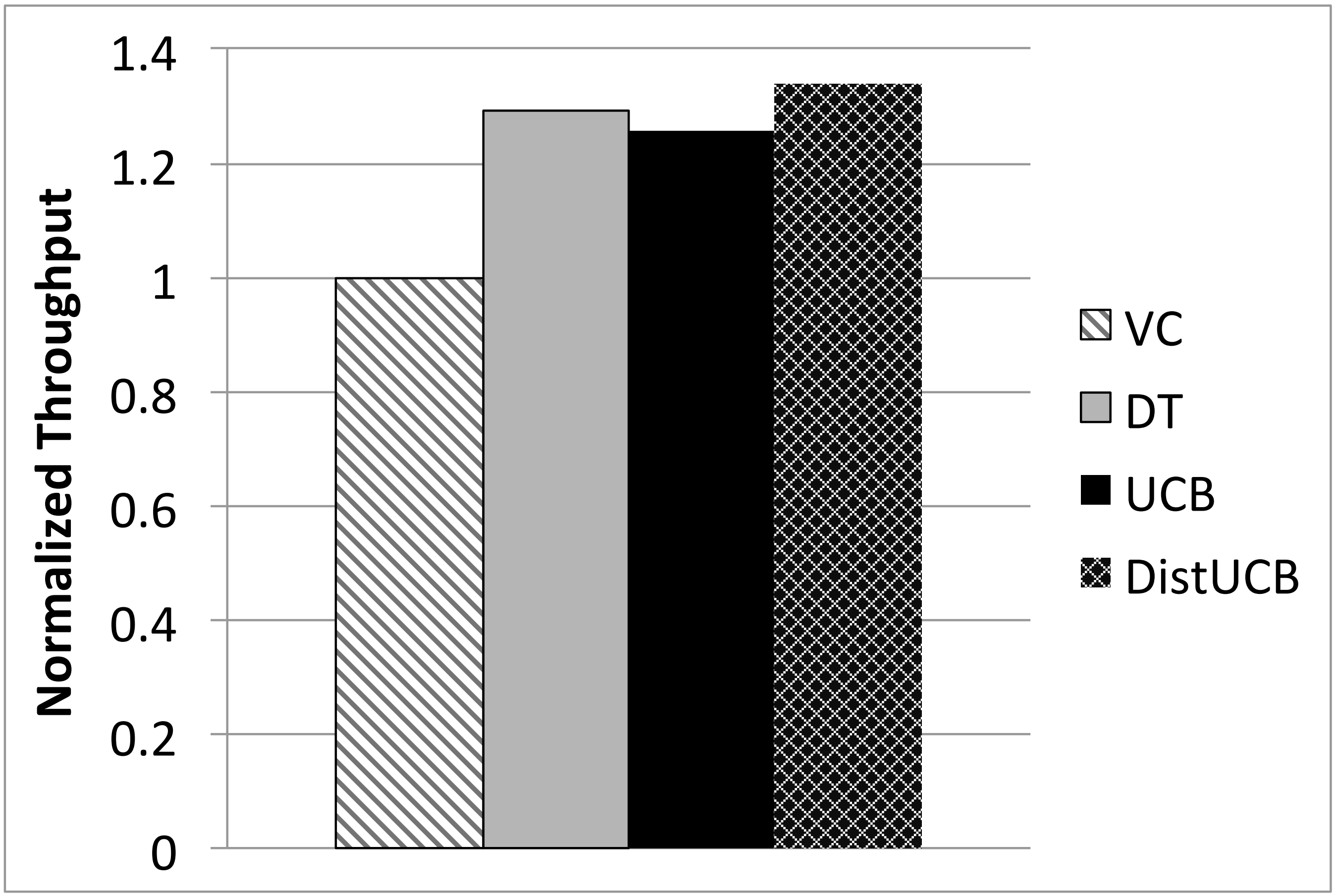
On-line Reinforcement Learning

- Distinguishes workload scenario solely based on read-set's size
  - exponential discretization intervals to minimize training time
- Replicas exchange statistical information periodically to boost learning

Chasing the optimum…



...and beating it!



Case studies

- Dynamic selection and switching between replication protocols:
  - total order based replication protocols (Case study 1):
    - purely based on Machine Learning techniques
  - single-master vs multi-master (Case study 2):
    - hybrid ML-AM solution – divide-and-conquer
- Group Communication System self-optimization:
  - batching in total order protocols (Case study 3)
    - hybrid ML-AM – ML bootstrapped with AM knowledge

Diego Didona, Sebastiano Peluso, Paolo Romano and Francesco Quaglia,
Self-tuning replication of elastic in-memory transactional data platforms,
INESC-ID Tec. Rep. 25/2011, May 2011.

SINGLE VS MULTI-MASTER

Classic Replication Protocols

- Focus on full replication protocols

Single master (primary-backup)

Multi master

Total order based

2PC-based

Certification

State machine replication

Non-voting

Voting

Single Master

- Write transactions are executed entirely in a single replica (the primary)

- If a write transaction is ready to commit, coordination is required to update all the other replicas (backups).

- Read transactions can be executed on backup replicas.

- **No distributed deadlocks**
- **No distributed coordination during commit**

- **Throughput of write txs doesn't scale up with number of nodes**

Classic Replication Protocols

- Focus on full replication protocols

Single master
(primary-backup)

Multi master

Total order based

2PC-based

Certification

State machine replication

Non-voting

Voting

2PC-based replication

- Transactions executed at all nodes w/o coordination till commit time
- Acquire atomically locks at all nodes using two phase commit (2PC):
  - 2PC materializes conflicts among concurrent remote transactions generating:

DISTRIBUTED DEADLOCKS

+ good scalability at low conflict

- thrashes at high conflict

Goals

- Autonomically select the best suited protocol that
  - Minimizes transactions' service time
  - Maximizes achievable throughput
- Automate elastic scaling
  - Scale up if the system needs more computational power
  - Scale down if the system is oversized

Self-optimizing data replication: key challenges

1. allow efficient switch among multiple replication protocols:
   - here coexistence of the 2 schemes is impossible:
     - design of an non-blocking protocol switching strategy

2. determine the optimal replication strategy given the current workload characteristics:
   - analytical models of effects of data contention
   - machine learning methods to predict hw-dependant latencies
     - CPU execution time, network RTT

Key Technical Problem

- How to forecast:
  - Performances of protocol B while running protocol A?
  - Performances with X nodes while running on Y nodes?

given that replication protocol/scale changes affect:

  - The transaction conflict probability
  - The transport layer latency

Methodology

Joint usage of analytical modeling and machine learning techniques:

- analytical model of replication algorithm dynamics:
  - lock contention, distributed deadlock probability
  - message exchange pattern
- machine learning to forecast performance of group communication layer:
  - RTT as a function of msg size, throughput, #nodes

Analytical Model

- Distributed lock contention dynamics captured via an analytical model:
  - the replication algorithms' behavior is fully specified
  - it is possible to mathematically model them
    (...although not easily, but that’s another story!)

- Key methodologies:
  - Mean value analysis & Queuing theory

- Rely on Machine Learning to forecast hardware dependent metrics, in particular network latencies...

Machine learning techniques

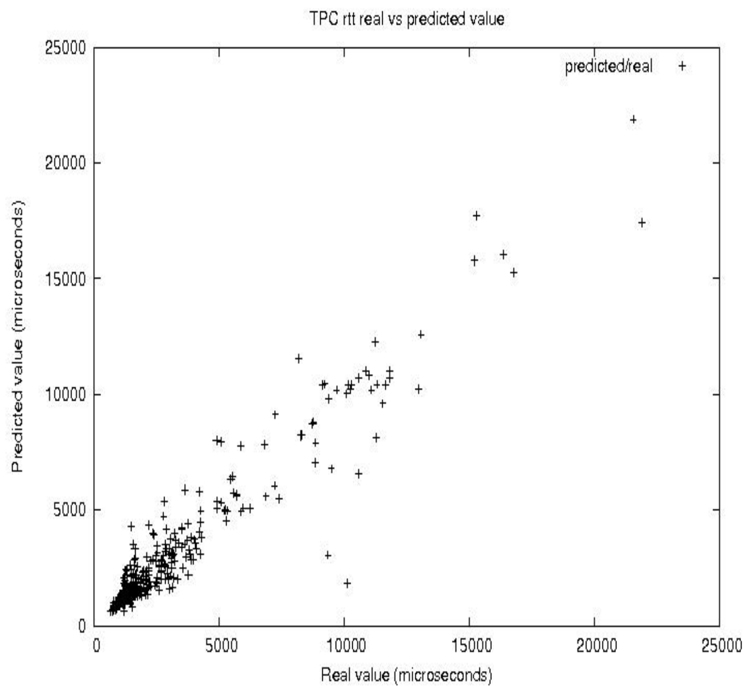
- Resource virtualization makes mathematical modeling unfeasible:
  - No knowledge of actual load
  - No knowledge of actual physical resources

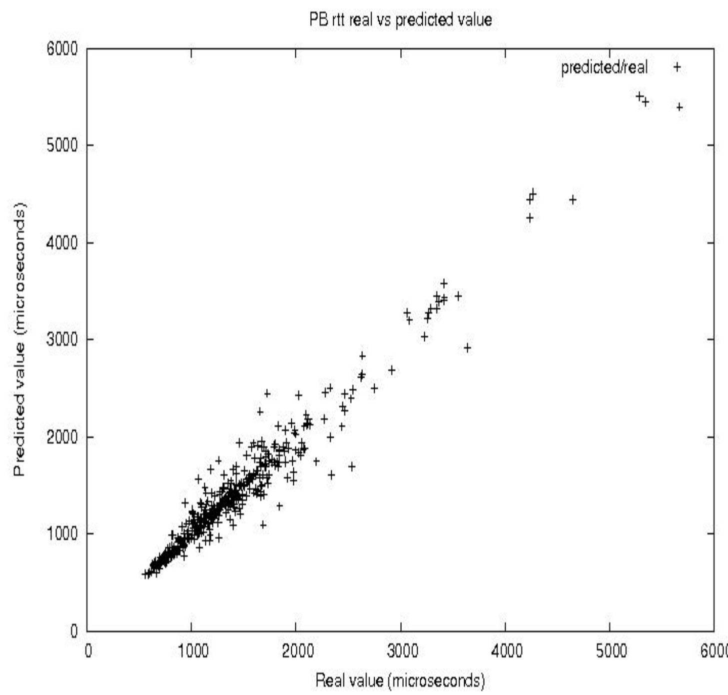
- Transport layer latency (RTT) of the two protocols predicted via decision tree regressors

Inputs for the ML

- Number of nodes
- RTT in the current configuration
- Size of exchanged messages
- Throughput of the current configuration
  - **Unknown!!!**
  - Guessed using the analytical model (more next)

Statistical Model Accuracy





- Correlation between 0.96 and 0.98
- Relative error between 0.19 and 0.22

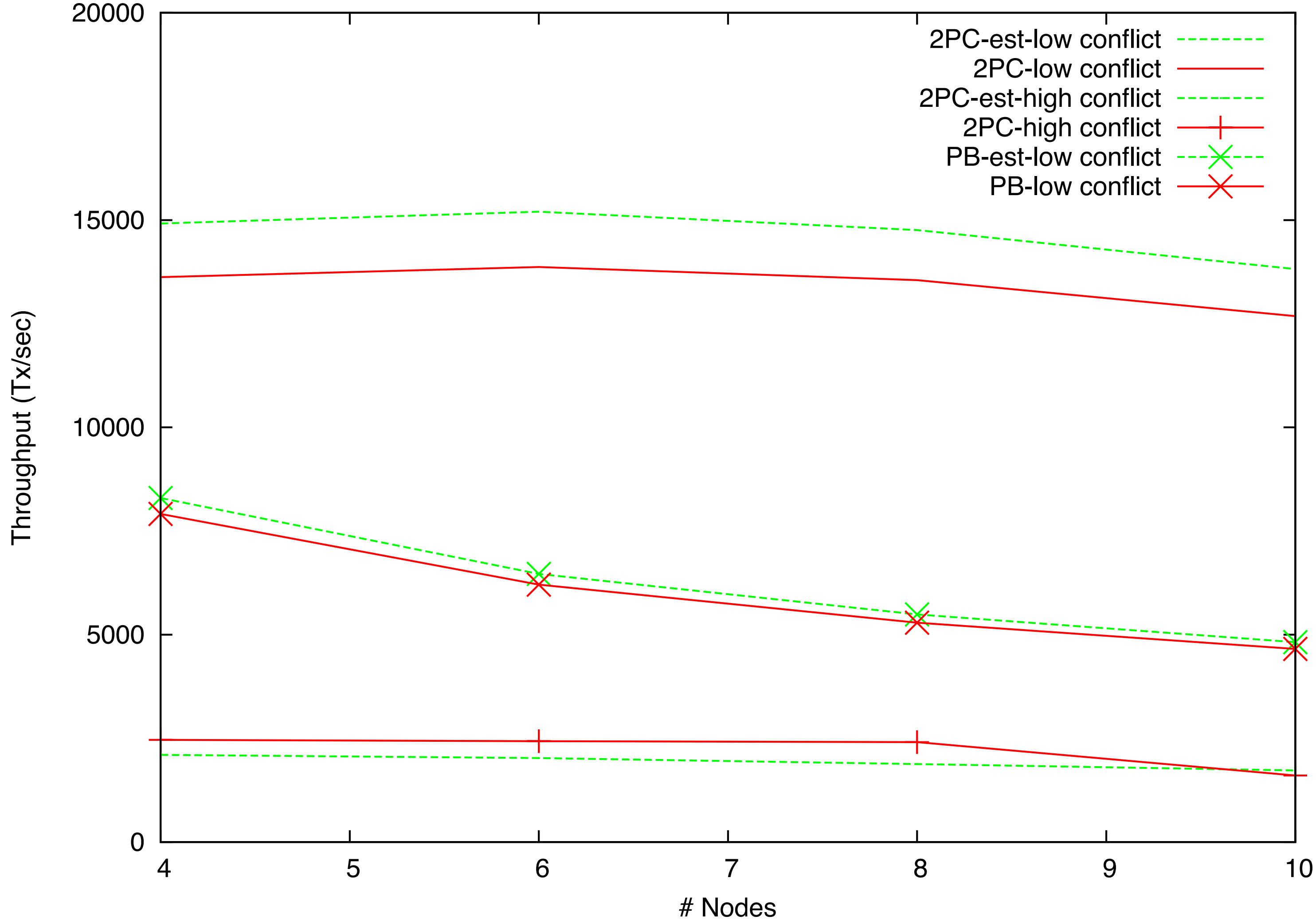
Models Coupling

Analytical model forecasts the **data grid throughput** taking as input the **RTT** in the target configuration.

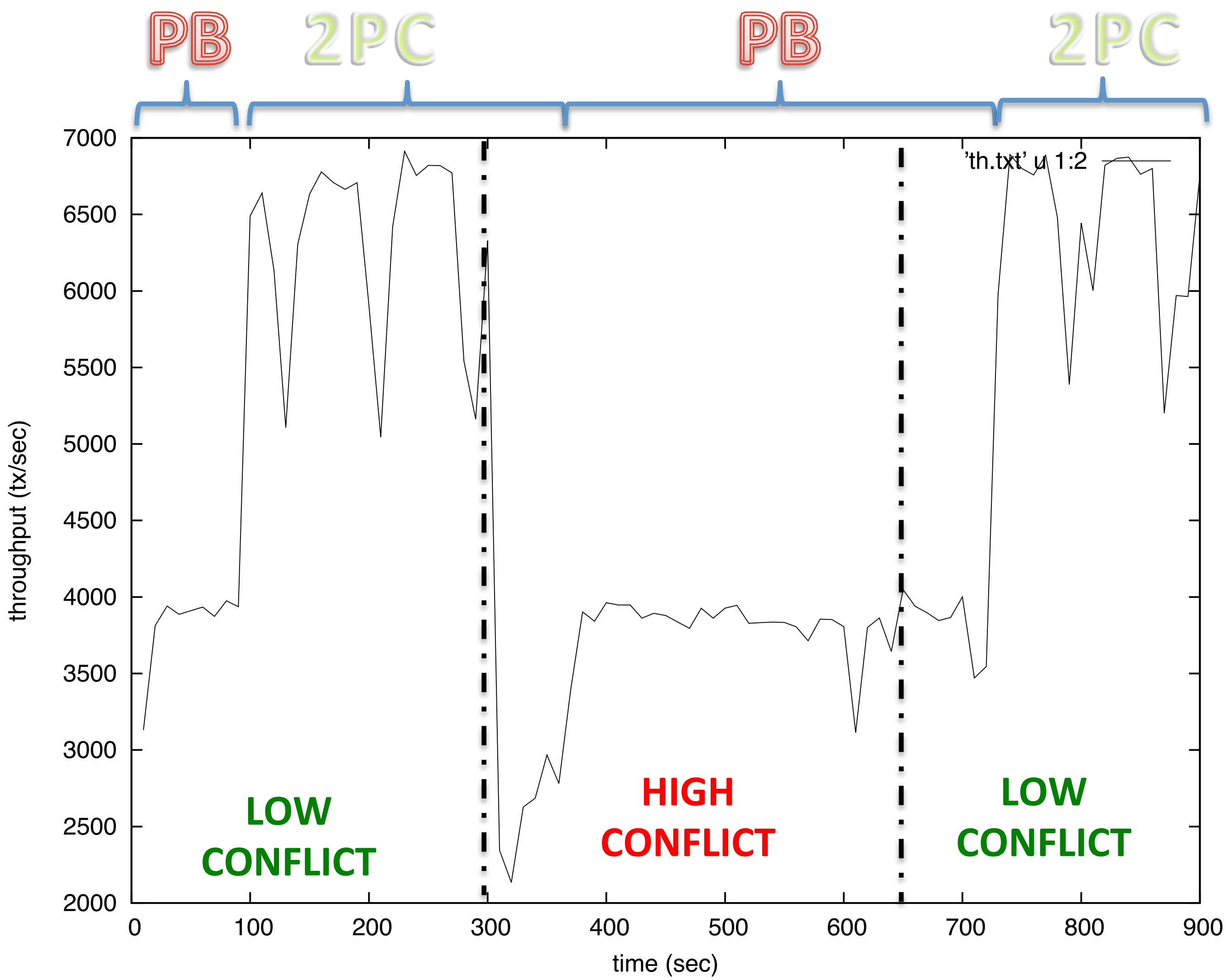
Machine learning forecasts the **RTT** taking as input the **data grid throughput** in the target configuration

Fixed point solution found using recursion

Global Model Accuracy



...and now in action!



Case studies

- Dynamic selection and switching between replication protocols:
  - total order based replication protocols (Case study 1):
    - purely based on Machine Learning techniques
  - single-master vs multi-master (Case study 2):
    - hybrid ML-AM solution – divide-and-conquer

- Group Communication System self-optimization:
  - batching in total order protocols (Case study 3)
    - hybrid ML-AM – ML bootstrapped with AM knowledge

Paolo Romano and Matteo Leonetti

Self-tuning Batching in Total Order Broadcast Protocols via Analytical Modelling and Reinforcement Learning

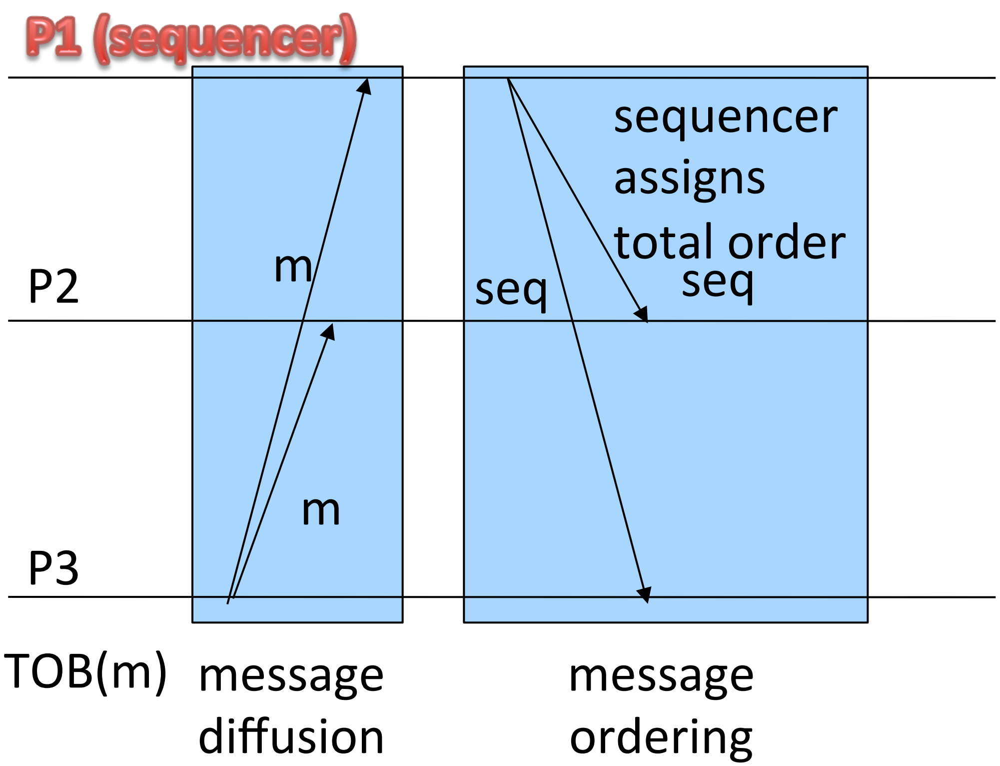
IEEE International Conference on Computing, Networking and Communications (ICNC'12), Jan. 2012

BATCHING IN TOTAL ORDER BROADCAST PROTOCOLS

Cloudviews 2011, Porto, Portugal, Nov. 4 2011

Sequencer based TOB (STOB)

- Total order broadcast (TOB) algorithms rely on a special process to ensure total order:



Batching in STOB protocols

- STOB have theoretically optimal latency:
  - 2 comm. steps, independently of the number of processes
- …but sequencer becomes the bottleneck at high throughput

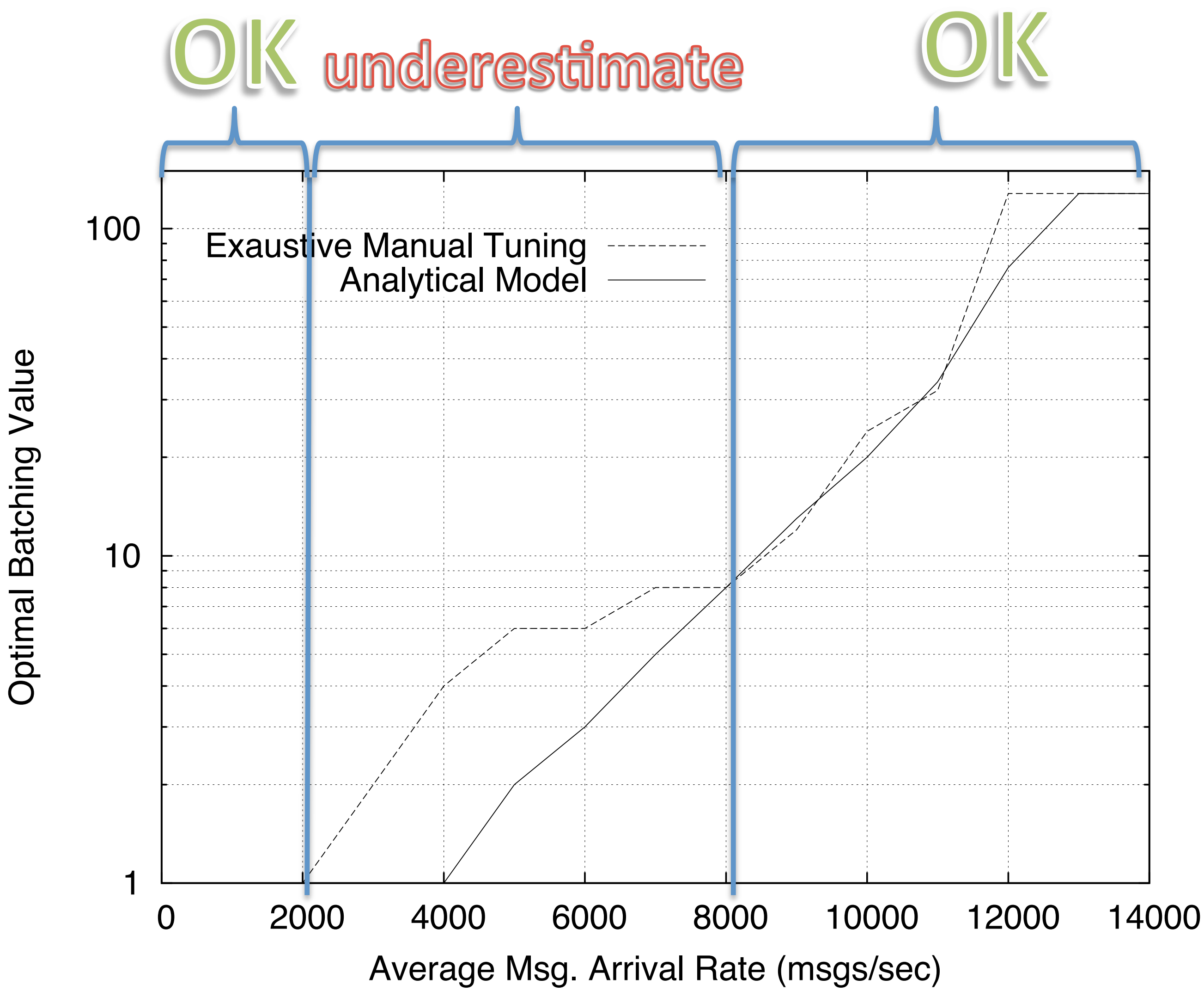
- Batching at the sequencer process:
  - wait for several msgs and order them altogether:
    - amortize sequencing cost across multiple messages
  - optimal waiting time depends on message arrival rate:
    - very effective at high throughput…
    - very bad at low throughput!

Analytical model

- Using queuing theory arguments we can determine the optimal batching time, b^*, as a function of the current load, m:

$$b^*(m) = \begin{cases} 1, \text{if } m < \frac{T_{add}\sigma^2}{2} + \frac{1}{2}\sqrt{\frac{4\sigma^2+2T_{add}^2\sigma^4}{2}} \\ \\ \frac{2m-\sigma-2mT_{add}\sigma}{\sigma-2mT_{add}\sigma+\sqrt{\frac{2(\sigma+2m(T_{add}\sigma-1))^2}{(2\sigma T_{add}-1)^2(1+2\sigma T_{add})\sigma^2}}}, \\ \text{if } \frac{T_{add}\sigma^2}{2} + \frac{1}{2}\sqrt{\frac{4\sigma^2+2T_{add}^2\sigma^4}{2}} < m < m^* \end{cases}$$

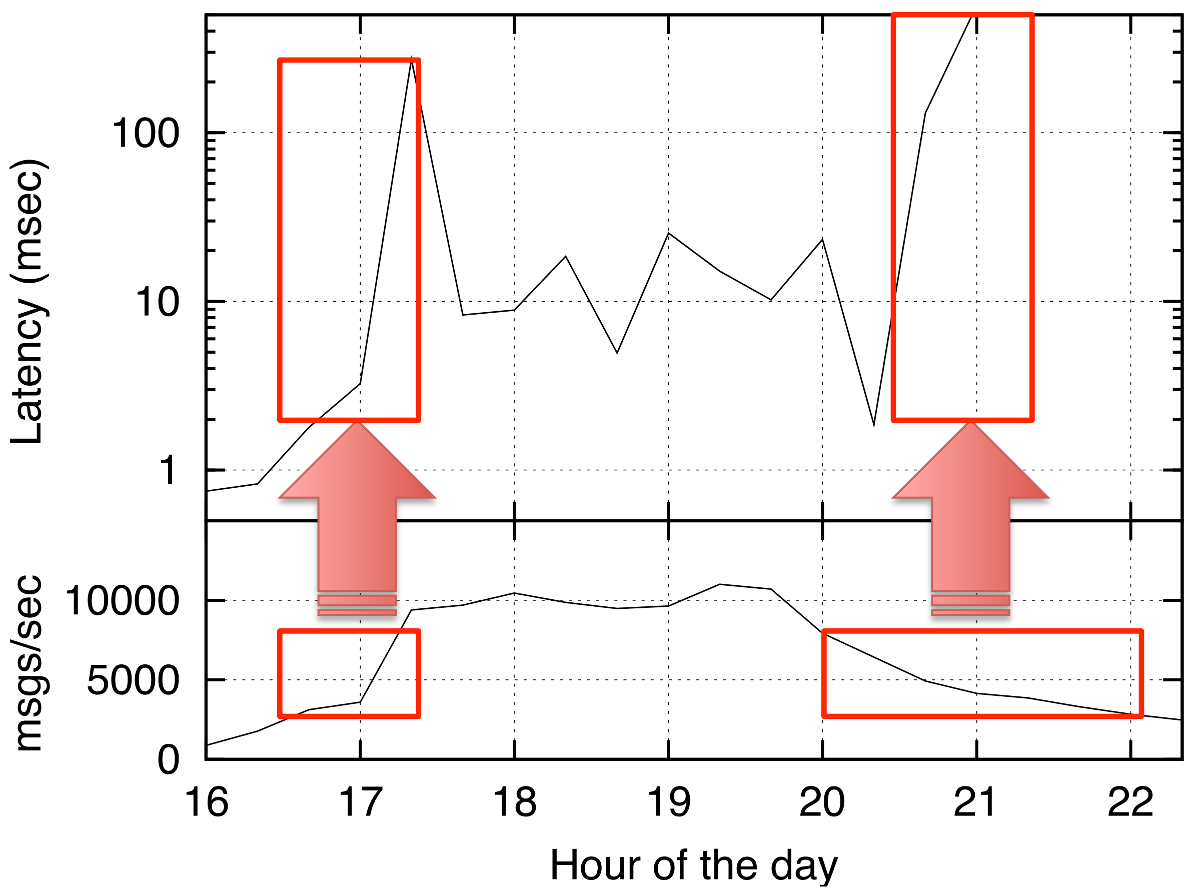
Model Accuracy



Model underestimates optimal batching value at "medium" load...

Problem:
batching underestimation causes system instability!

Peak period analysis



Ramp-up & ramp-down transition through the "problematic" areas:

- ramp-up is sufficiently short:

⇒ system "struggles", but recovers

- ramp-down is longer:

⇒

What about a pure ML approach?

- Problem:
  - ML techniques need to explore different solutions (batching values) to identify optimal one:
    - low load: useless additional latency

INITIALLY INEFFICIENT

    - medium-high load: insufficient batching values lead very rapidly to instability and thrashing

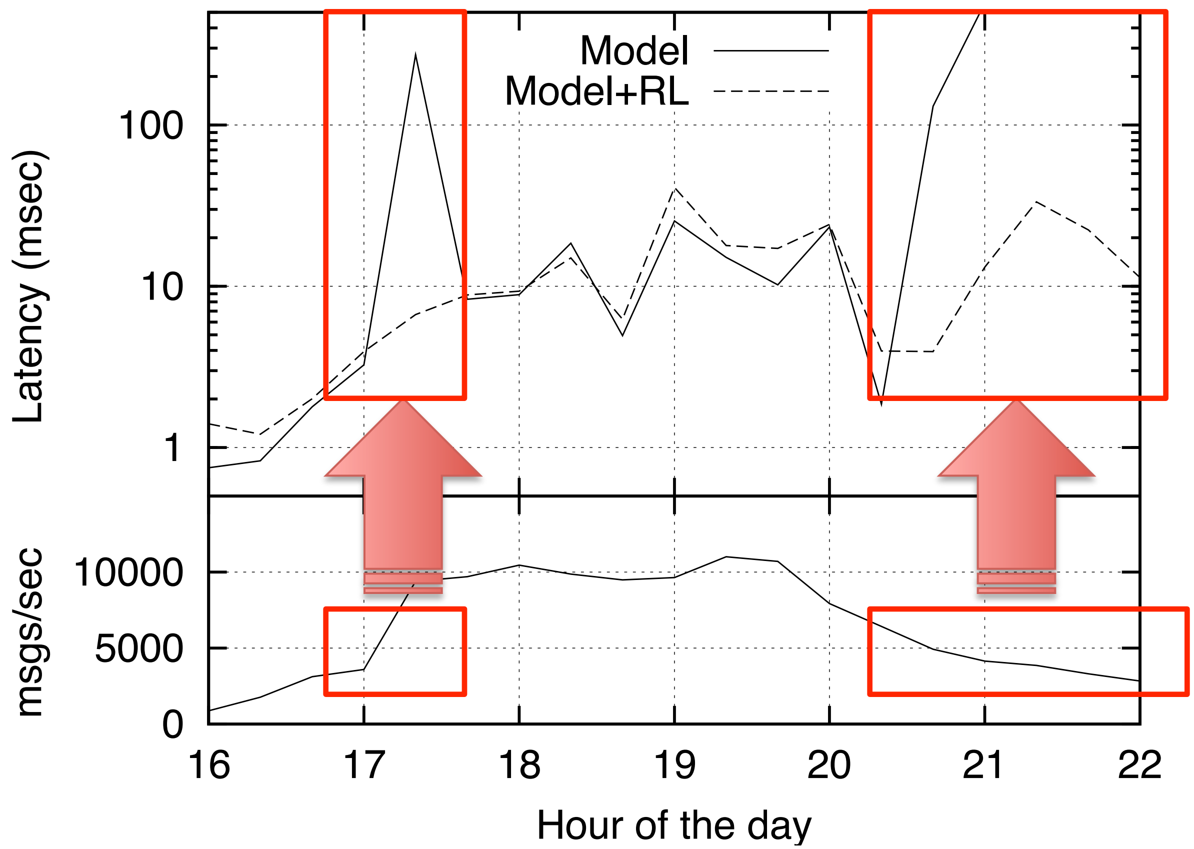
UNFEASIBLE

Combining the two approaches

1. Initialize ML's knowledge with the predictions of the analytical model:
   – reduce frequency of obviously wrong explorations

2. Let ML update the initial reward values:
   – correct model's prediction errors exploiting feedback from the system

Combining the two approaches



Talk structure

- Cloud-TM Overview:
  - key goals
  - background on Transactional Memories
  - progresses so far
- Self-optimizing transactional data grids:
  - self-optimizing methodologies explored so far
  - case studies
- Open research questions & future work

Open research issues

- Holistic approaches to self-optimization:
  - understand effects of self-optimizing multiple, mutually dependent layers
- QoS-aware programming paradigms:
  - methodologies and tools to allow providers to assess feasibility of fulfilling SLAs
- Highly scalable data consistency models:
  - beyond eventual consistency

Future work

- Focus on elastic-scaling, keeping into account data grids dynamics:
  - consistency costs, transaction conflicts
- Study effects of self-tuning multiple, mutually dependent layers of the data grid
- Highly scalable quasi-serializable protocols

THANKS FOR THE ATTENTION

References (I)

[CMG10] P. Di Sanzo, B. Ciciani, F. Quaglia, R. Palmieri and Paolo Romano, Analytical Modelling of Commit-Time-Locking Algorithms for Software Transactional Memories, Proc. 35th International Computer Measurement Group Conference (CMG), Orlando, Florida, Computer Measurement Group, December 2010

[PEVA11] P. Di Sanzo, B. Ciciani, F. Quaglia, R. Palmieri and Paolo Romano On the Analytical Modeling of Concurrency Control Algorithms for Software Transactional Memories: the Case of Commit-Time-Locking, Elsevier Performance Evaluation Journal, to appear

[Middleware2010] N. Carvalho, Paolo Romano and L. Rodrigues, Asynchronous Lease-based Replication of Software Transactional Memory, Proceedings of the ACM/IFIP/USENIX 11th Middleware Conference (Middleware), Bangalore, India, ACM Press, November 2010

[Middleware2011] PolyCert: Polymorphic Self-Optimizing Replication for In-Memory Transactional Grids, M. Couceiro, P. Romano and L. Rodrigues, ACM/IFIP/USENIX 12th International Middleware Conference (Middleware 2011)

[SPAA2010] Paolo Romano, R. Palmieri, F. Quaglia, N. Carvalho and L. Rodrigues, On Speculative Replication of Transactional Systems (Brief Announcement), Proc. 22nd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), Santorini, Greece, ACM Press, June 2010

[ISPA2010] Paolo Romano, R. Palmieri, F. Quaglia, N. Carvalho and L. Rodrigues, An Optimal Speculative Transactional Replication Protocol, Proc. 8th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), Taiwan, Taipei, IEEE Computer Society Press, September 2010

[NCA2010] R. Palmieri, Paolo Romano and F. Quaglia, AGGRO: Boosting STM Replication via Aggressively Optimistic Transaction Processing, Proc. 9th IEEE International Symposium on Network Computing and Applications (NCA), Cambridge, Massachussets, USA, IEEE Computer Society Press, July 2010

References (II)

[SYSTOR2011] N. Carvalho, Paolo Romano and L. Rodrigues, SCert: Speculative Certification in Replicated Software Transactional Memories, Proceedings of the 4th Annual International Systems and Storage Conference (SYSTOR 2011), Haifa, Israel, June 2011.

[SRDS2011] R. Palmieri, F. Quaglia and Paolo Romano, OSARE: Opportunistic Speculation in Actively REplicated Transactional Systems (Short Paper), The 30th IEEE Symposium on Reliable Distributed Systems (SRDS 2011), Madrid, Spain, Oct. 2011

[SASO10] M. Couceiro, Paolo Romano and L. Rodrigues, A Machine Learning Approach to Performance Prediction of Total Order Broadcast Protocols , Proc. 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Budapest, Hungary, IEEE Computer Society Press, September 2010

[Performance2011] Paolo Romano and M. Leonetti, Poster: Self-tuning Batching in Total Order Broadcast Protocols via Analytical Modelling and Reinforcement Learning, ACM Performance Evaluation Review, to appear (also presented at IFIP Performance 2011 Symposium)

[ICNC12] Paolo Romano and M. Leonetti, Self-tuning Batching in Total Order Broadcast Protocols via Analytical Modelling and Reinforcement Learning, IEEE International Conference on Computing, Networking and Communications, Network Algorithm & Performance Evaluation Symposium (ICNC'12), Jan. 2012

[PRDC2011] Exploiting Total Order Multicast in Weakly Consistent Transactional Caches, P. Ruivo, M. Couceiro, P. Romano and L. Rodrigues, Proc. IEEE 17th Pacific Rim International Symposium on Dependable Computing (PRDC’11)