On the Natural Degree of Parallelism in Transactional Memory: from Centralized to Distributed Architectures

D. Didona[1], P. Felber[2], D. Harmanci[2], P. Romano[1], **J. Schenker**[2]

Euro-TM Workshop on Transactional Memory (WTM 2012)

Bern, Switzerland, April 10, 2012

[1] Instituto Superior Técnico [2] Université de Neuchâtel

Table of Contents

1. Maximum Level of Parallelism for TM Applications
2. The case of (non distributed) TM
3. The case of Distributed TM
4. Conclusions and Current Research

Addressed Problem

Finding the maximum level of parallelism for a TM application, i.e., the number of threads

- below which adding threads improves throughput
- over which adding threads decreases throughput
  - even if sufficiently many cores are available

Challenges

Maximum Level of Parallelism is

- Workload specific
  - Data contention level
  - Transaction length
  - Read/Write operations ratio

Challenges

Maximum Level of Parallelism is

- Workload specific
  - Data contention level
  - Transaction length
  - Read/Write operations ratio
- Platform dependent
  - Concurrency Control algorithm
  - Scale of the TM (i.e., centralized or distributed)

Challenges

Maximum Level of Parallelism is

- Workload specific
  - Data contention level
  - Transaction length
  - Read/Write operations ratio
- Platform dependent
  - Concurrency Control algorithm
  - Scale of the TM (i.e., centralized or distributed)
- ...and varies over time
  - Any static value is sub-optimal
  - Offline trial-and-error is non viable

Table of Contents

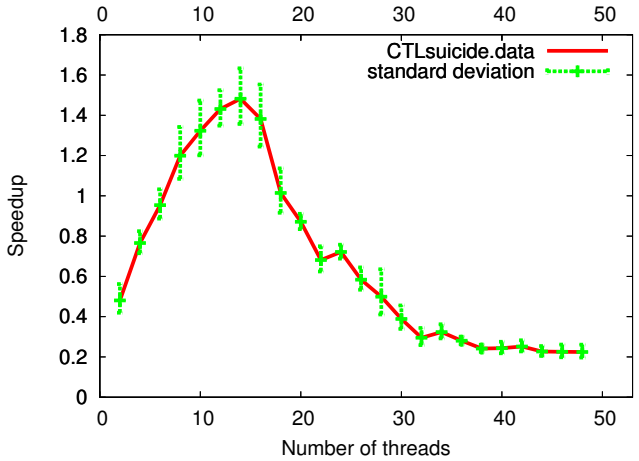
1 Maximum Level of Parallelism for TM Applications

2 The case of (non distributed) TM

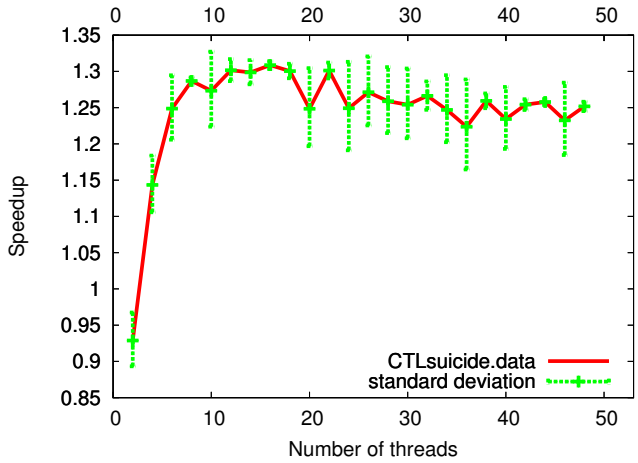
3 The case of Distributed TM

4 Conclusions and Current Research

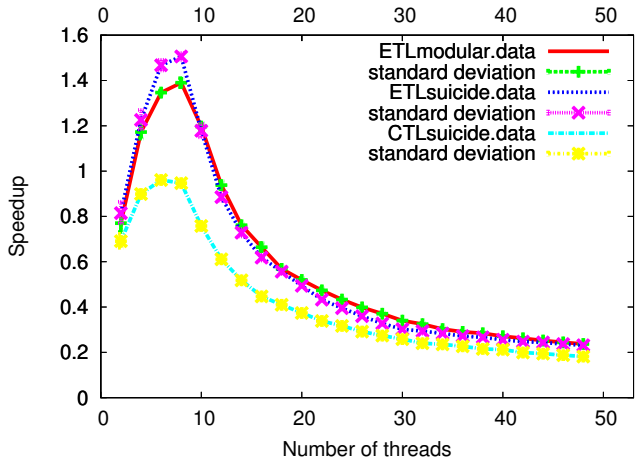
Natural/Optimal Degree of Parallelism



Kmeans-high

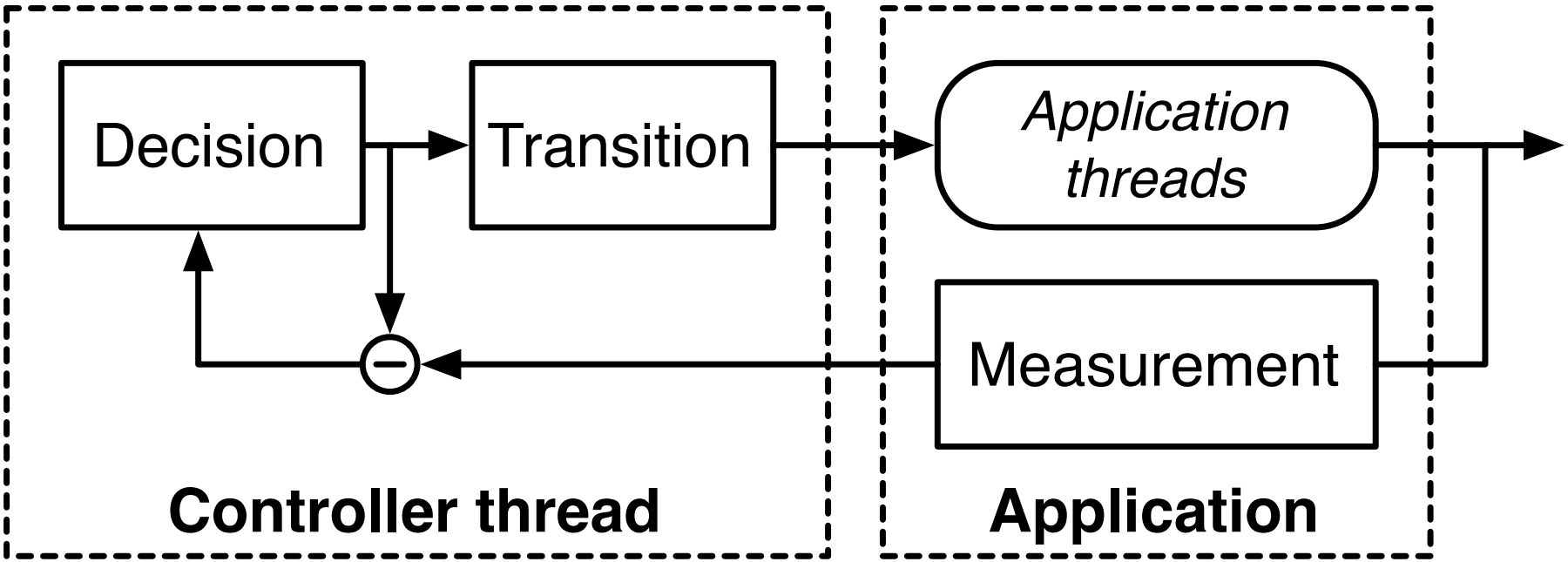


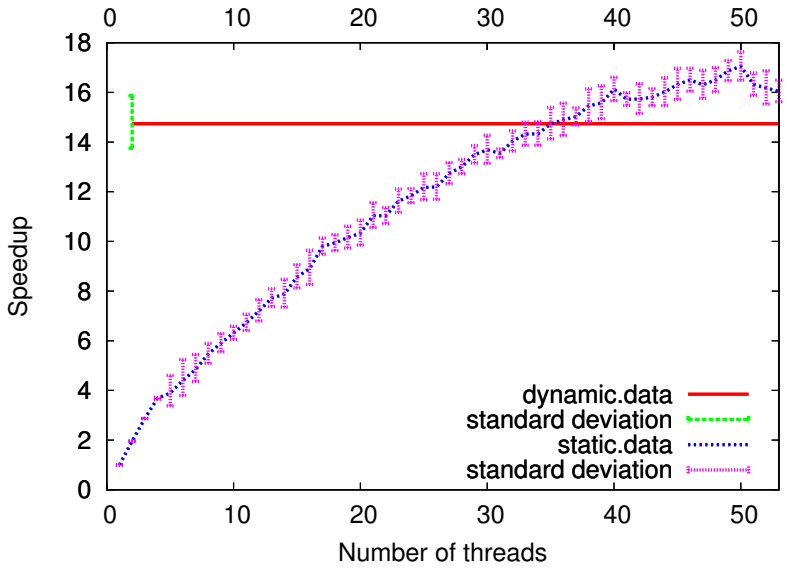
ssca2



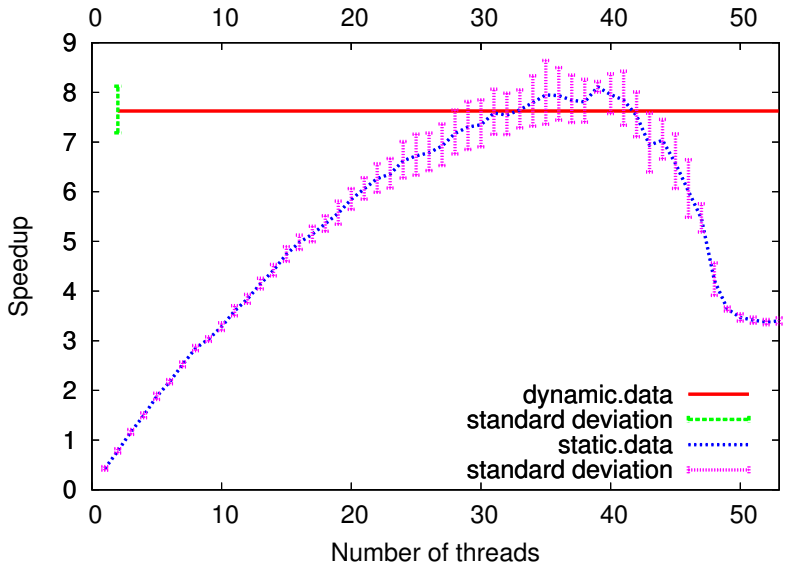
Intruder

Dynamic Thread Management

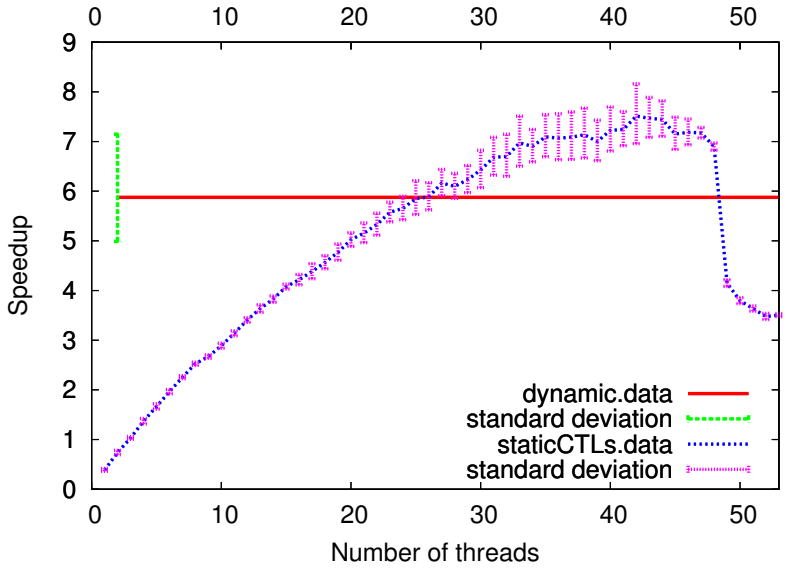




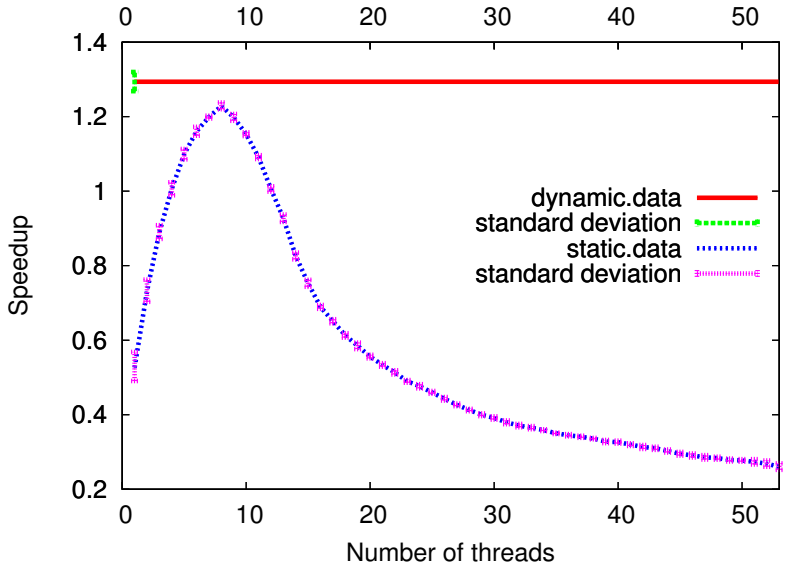
Labyrinth



Vacation-low



Vacation-high



Intruder

During an Intruder Execution

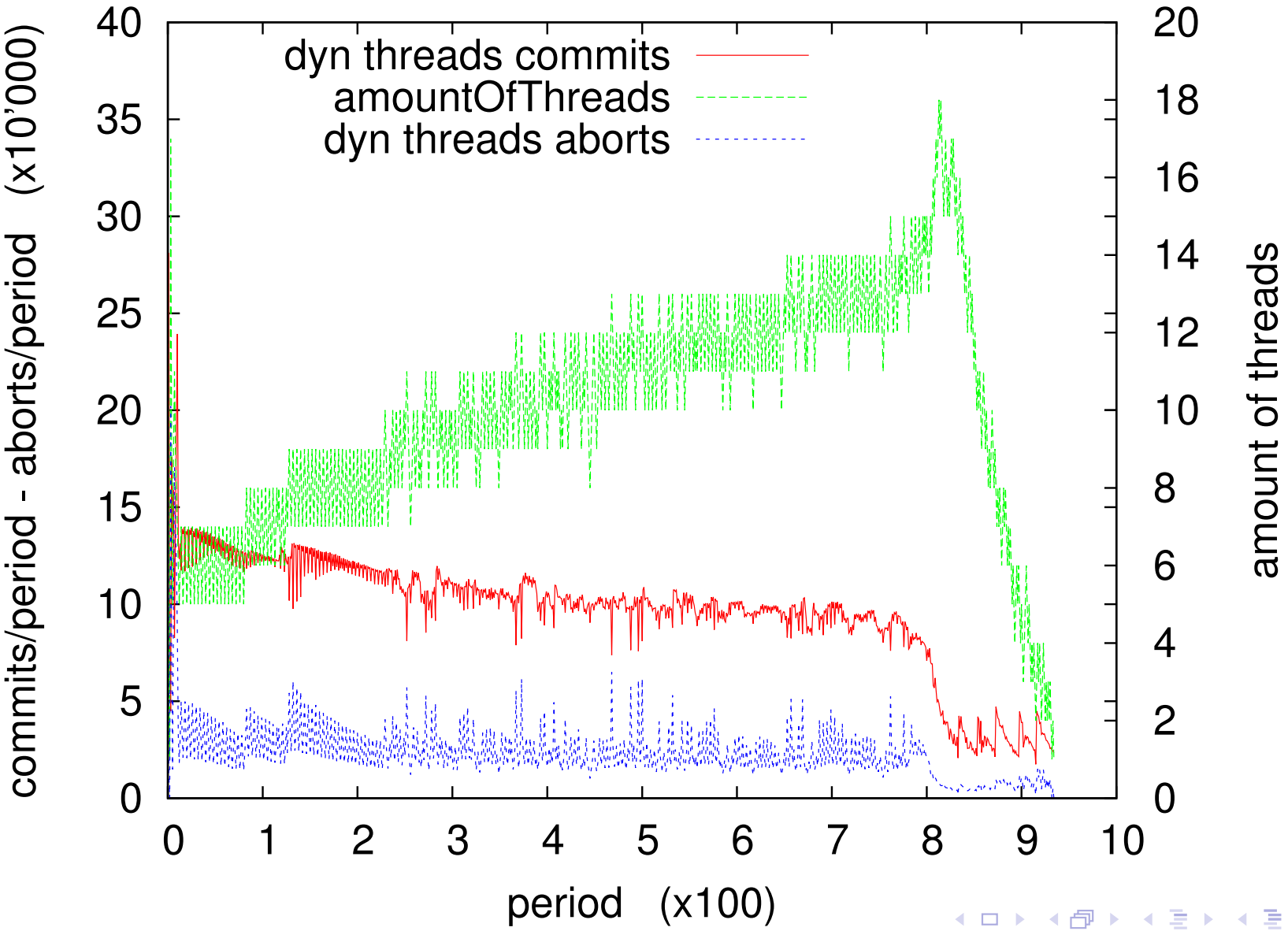


Table of Contents

1 Maximum Level of Parallelism for TM Applications

2 The case of (non distributed) TM

3 The case of Distributed TM

4 Conclusions and Current Research

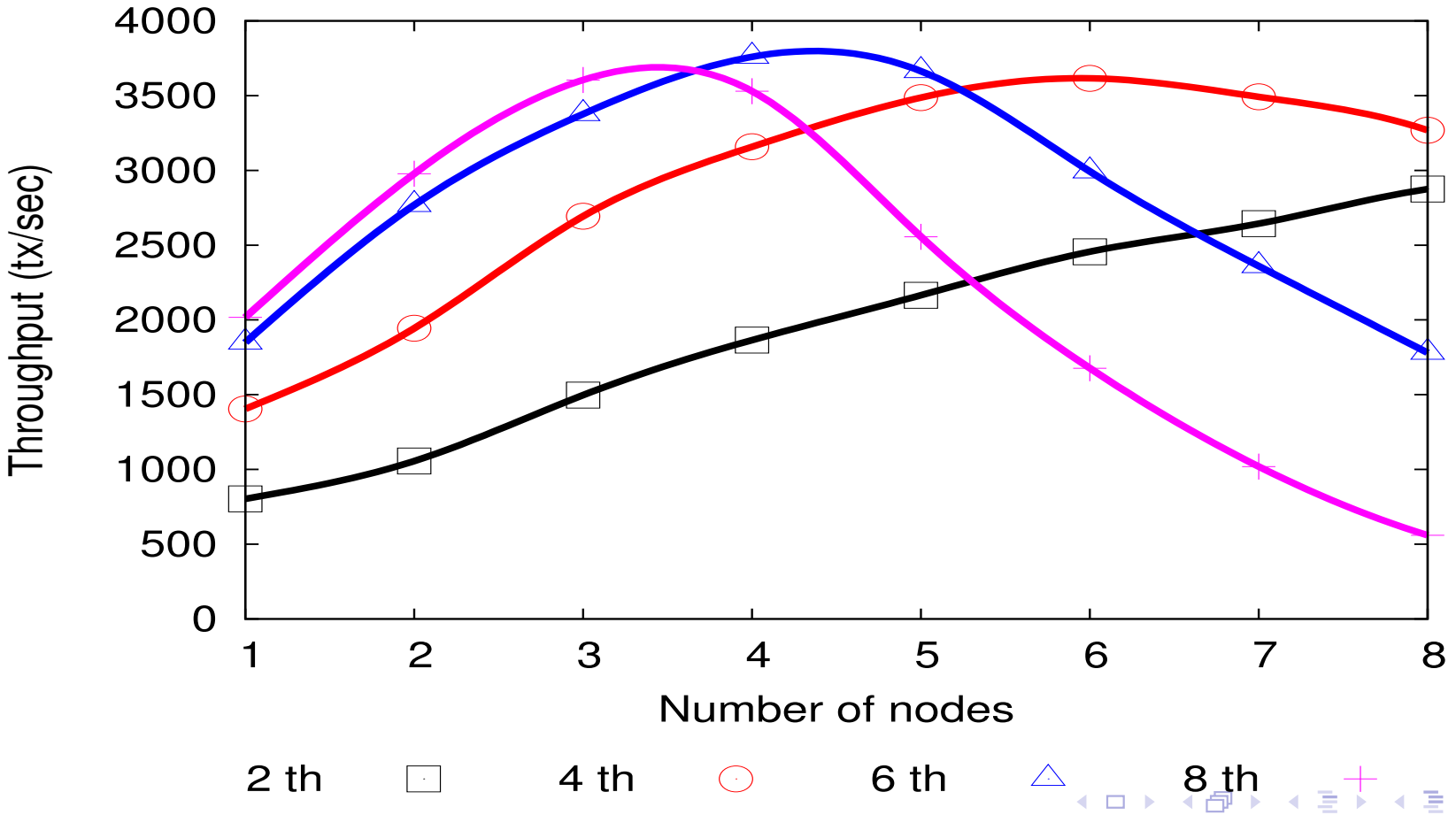
Challenges

- Increase in the dimensionality of the problem
  - At single node level
    - # threads per node
    - Local execution time
  - At system level
    - #nodes in the DTM
    - Dominant distributed synchronization costs

Case study: TPC-C Benchmark

Infinispan as DTM

- Exclusive write lock taken locally at encounter time
- Two-phase commit for global consistency



Maximum level of parallelism in DTM

- Efficiency of exploration-based techniques depends on the efficiency of exploration steps
  - Adding/removing threads in a centralized TM is cheap
  - Adding/remove nodes on a DTM is costly
    - involves state transfers
    - in Cloud, new replica = \$
    - introduces instability
    - requires global reconfiguration protocol

Maximum level of parallelism in DTM

- Efficiency of exploration-based techniques depends on the efficiency of exploration steps
  - Adding/removing threads in a centralized TM is cheap
  - Adding/remove nodes on a DTM is costly
    - involves state transfers
    - in Cloud, new replica = $
    - introduces instability
    - requires global reconfiguration protocol

Pure exploration-based techniques are cumbersome and inefficient in DTM

- What about predictive models?

Model-based techniques

Predictive models allow to forecast the impact of adding/removing nodes on a DTM

- Target configuration reached in a single step
- Avoid useless state transfers

Transactional Auto Scaler

System that given the current workload allows to predict the optimal scale for the DTM

- Joint usage of Analytical Models and Machine Learning
  - Analytical model captures logical contention (e.g., abort prob.)
  - Machine learning predicts physical layer's behavior (e.g., Rtt)
- General and modular methodology
  - We tailored it for Infinispan

TAS' models

Analytical Model

- White-box approach: knowledge of DTM consistency alg.
- Good extrapolation power
- Queueing theory

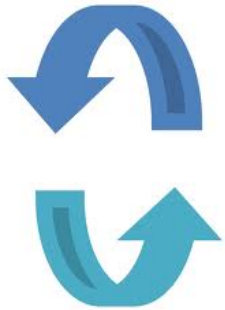
Machine Learning

- Black-box approach: physical layer is hard to model
- Suitable for Virtualized environments
- Offline trained decision-tree regressors

TAS' models

Analytical Model

- White-box approach: knowledge of DTM consistency alg.
- Good extrapolation power
- Queueing theory

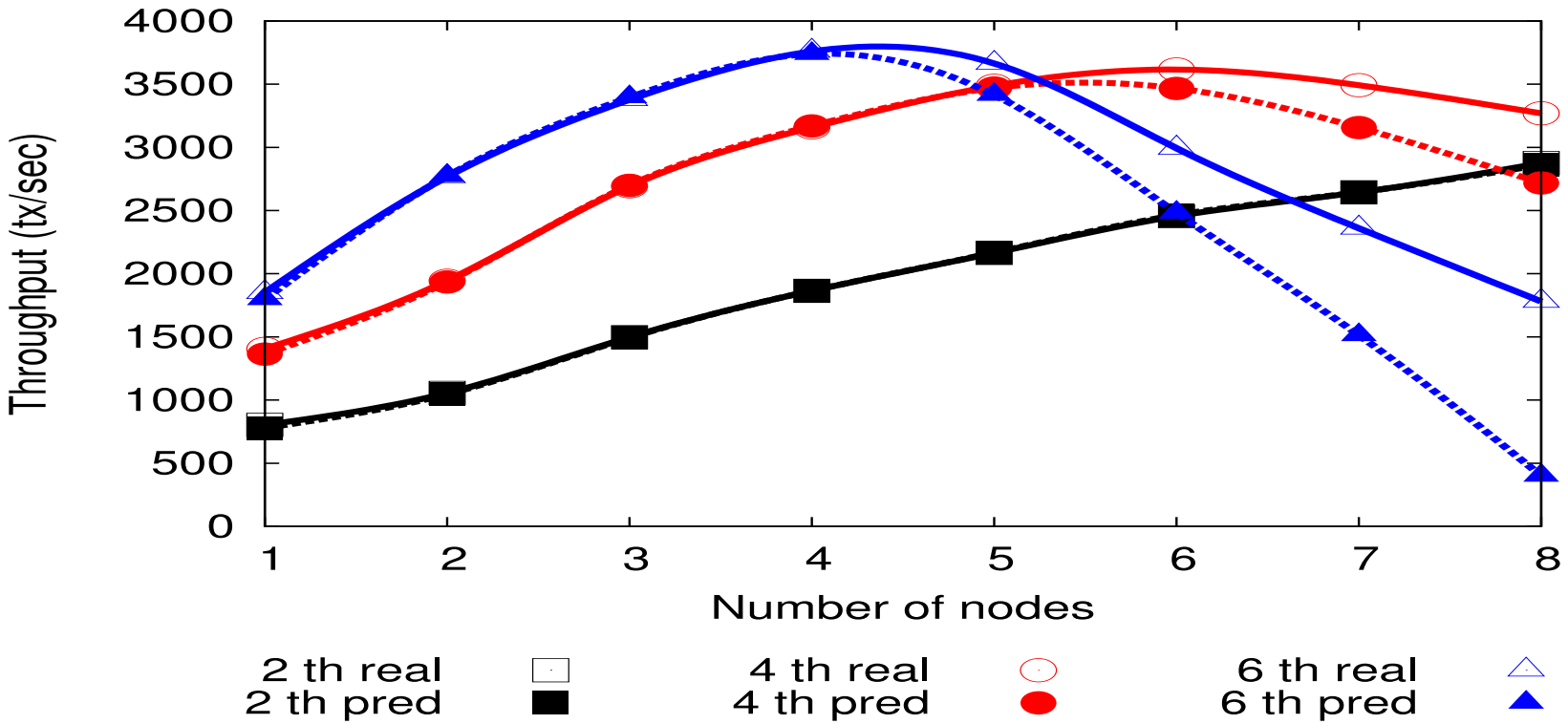


Machine Learning

- Black-box approach: physical layer is hard to model
- Suitable for Virtualized environments
- Offline trained decision-tree regressors

TAS accuracy

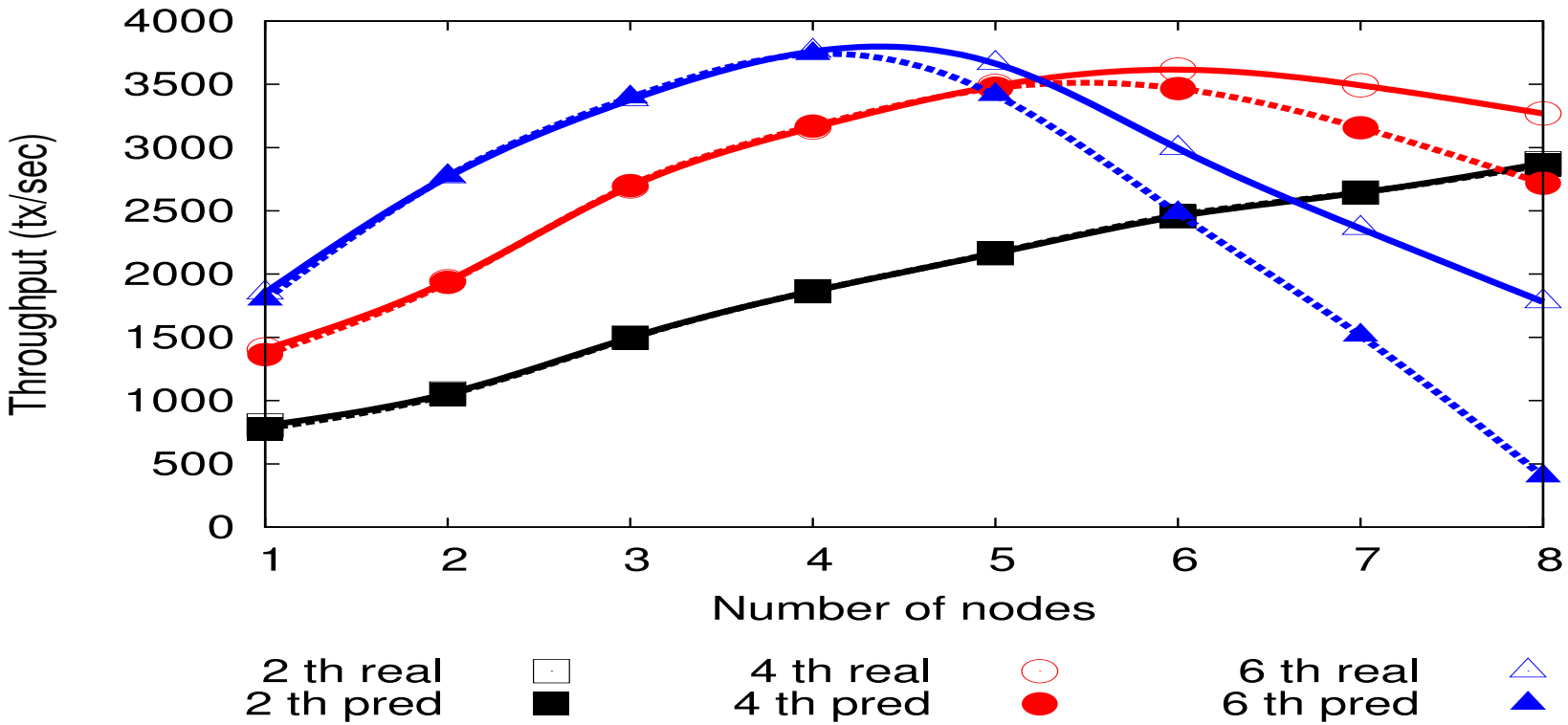
System performance predicted from scenario 2 nodes, 2 threads



- Very good accuracy in "similar" scenarios (e.g., same # threads)
- Ability to determine the optimal level of parallelism

TAS accuracy

System performance predicted from scenario 2 nodes, 2 threads



- Loss of accuracy for configurations where
  - model's hypotheses are challenged (i.e., abort probability $> 80\%$)
  - machine learner has not been extensively trained

Table of Contents

1 Maximum Level of Parallelism for TM Applications

2 The case of (non distributed) TM

3 The case of Distributed TM

4 Conclusions and Current Research

Towards a combined approach

- Exploration-based techniques are robust but not scalable to distributed settings
- Model-based techniques are more attractive for DTMs but can suffer from inaccuracy
- The two techniques can be jointly exploited
  - On-line exploration can improve model's accuracy
  - Analytical model can improve scalability of on-line technique

Current research

Taking the best of the two worlds:

- Using predictive models to guide on-line exploration
  - goal: maximize convergence speed of on-line exploration
- Incorporating in the model the knowledge acquired with on-line exploration
  - goal: progressively enhance predictive model's accuracy

Questions?


cost
ACTION IC1001
EUR TM


Cloud·TM
A novel programming
paradigm for the Cloud