
# Gray box performance modeling of Distributed TM

Diego Didona

Instituto Superior Técnico, Universidade de Lisboa

# My STSM at a glance

- **Identifying the optimal level of parallelism for TM**
- Jan-Feb 2012, end of my 1st PhD year
- Home Institution
  - Instituto Superior Técnico, Universidade de Lisboa
  - Supervised by Prof. Paolo Romano
- Host Institution
  - Université de Neuchâtel
  - Supervised by Prof. Pascal Felber

# Home and Host expertise



- Home:
  - Design of distributed TM ($D^2STM$)
  - Analytical modeling of DTM (Aristos, Cloud-TM)
- Host:
  - Design of shared-memory TM (Velox, TMWare)
  - Online tuning of shared-memory TM (TinySTM)


# STSM Motivation

TM simplifies parallel programming

Performance depends on (varying) workload

- Read vs write dominated
- Low or high conflict
- Long or short xatcs

Self-tuning to adapt TM internals at runtime

# STSM purpose

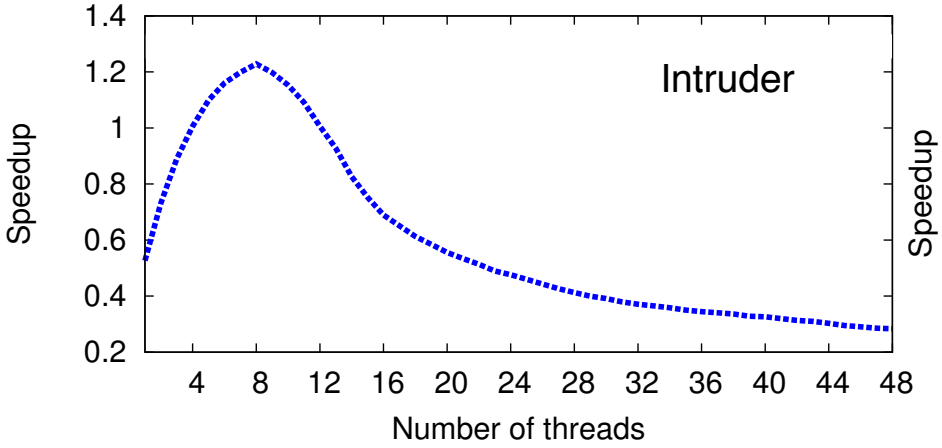
Exploit complementary home/host backgrounds

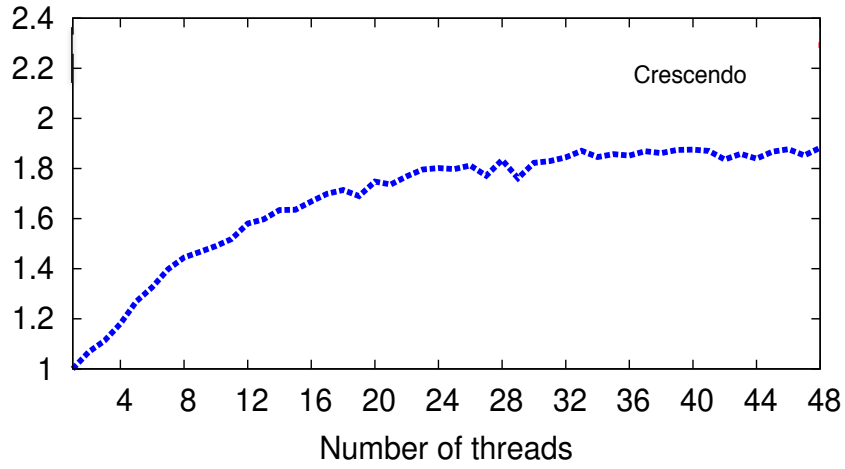
- investigate fundamental issues/differences self tuning in shared-memory vs distributed TM
- combine model-based & feedback-driven techniques

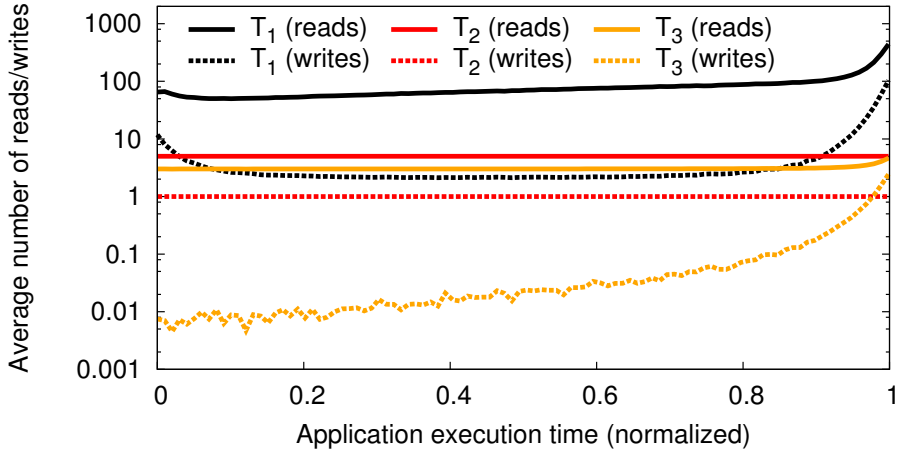
TARGET: **Identifying optimal level of parallelism for TM**

# Parallelism in Shared-memory TM

- # active transactional threads

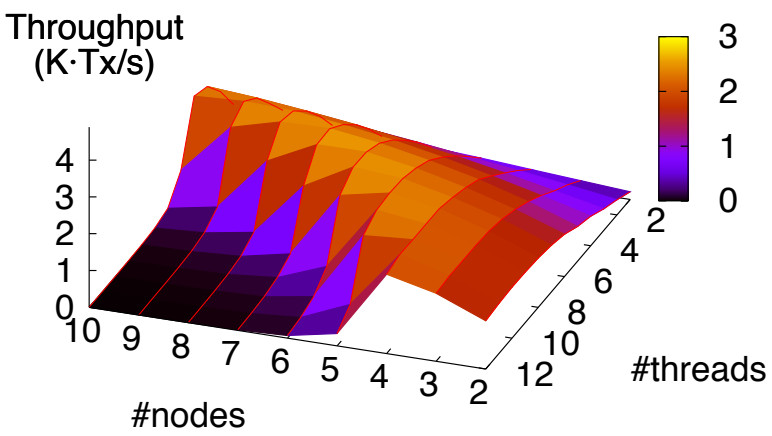


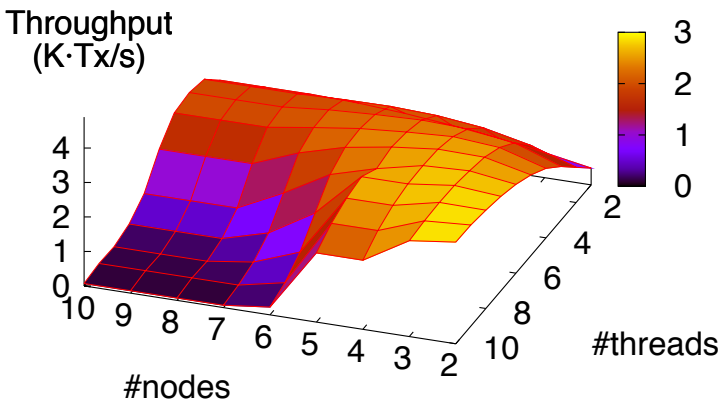



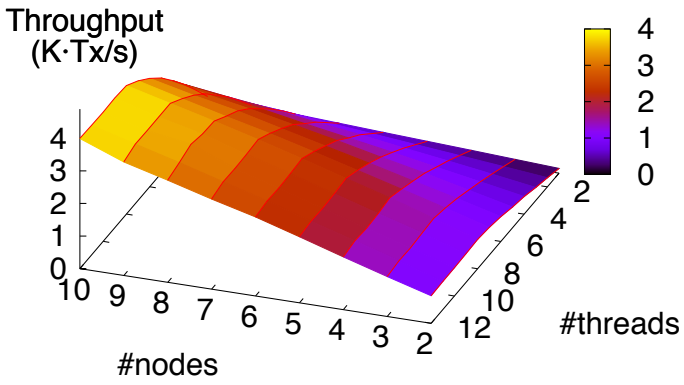



# Parallelism in Distributed TM

- # Nodes and active threads per node

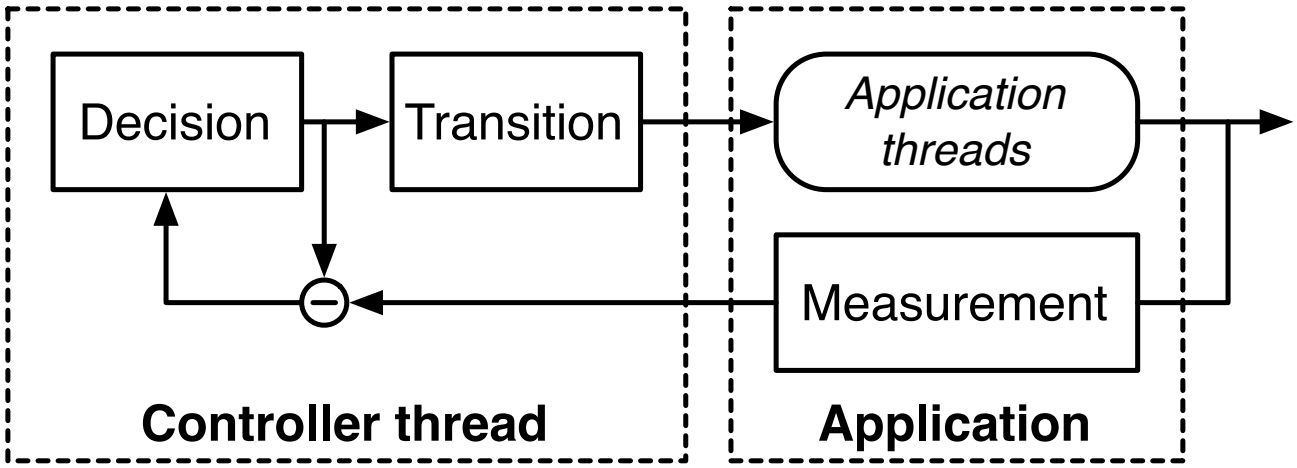


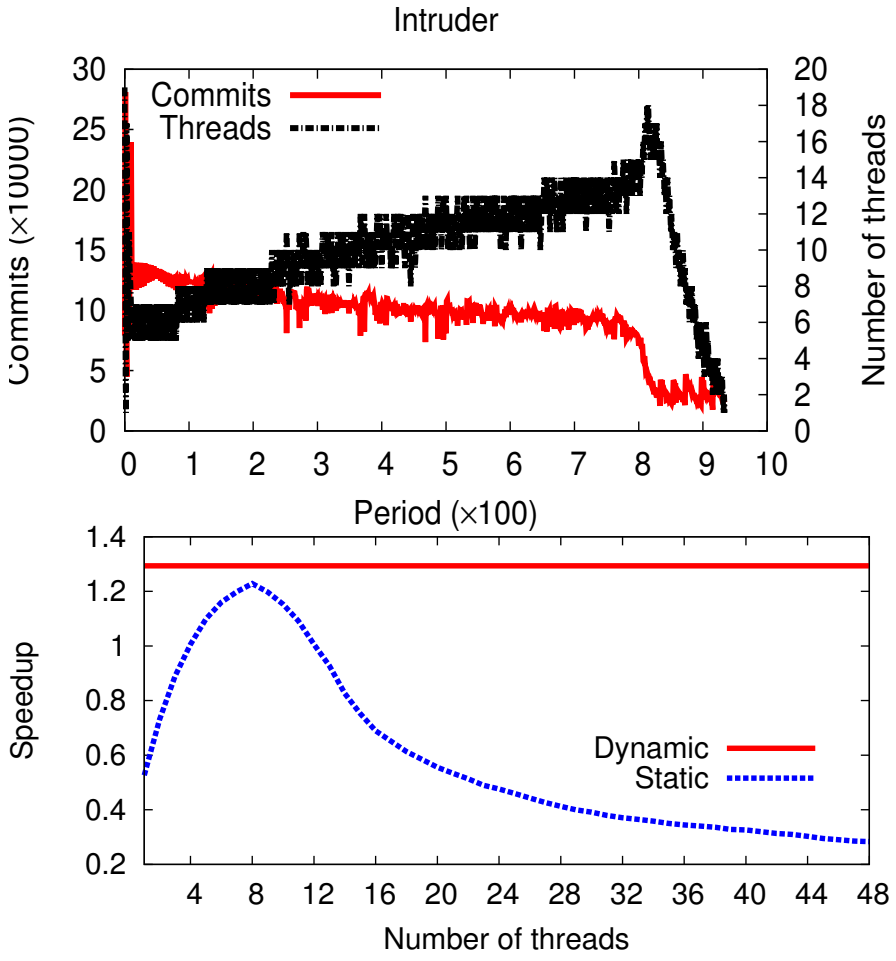



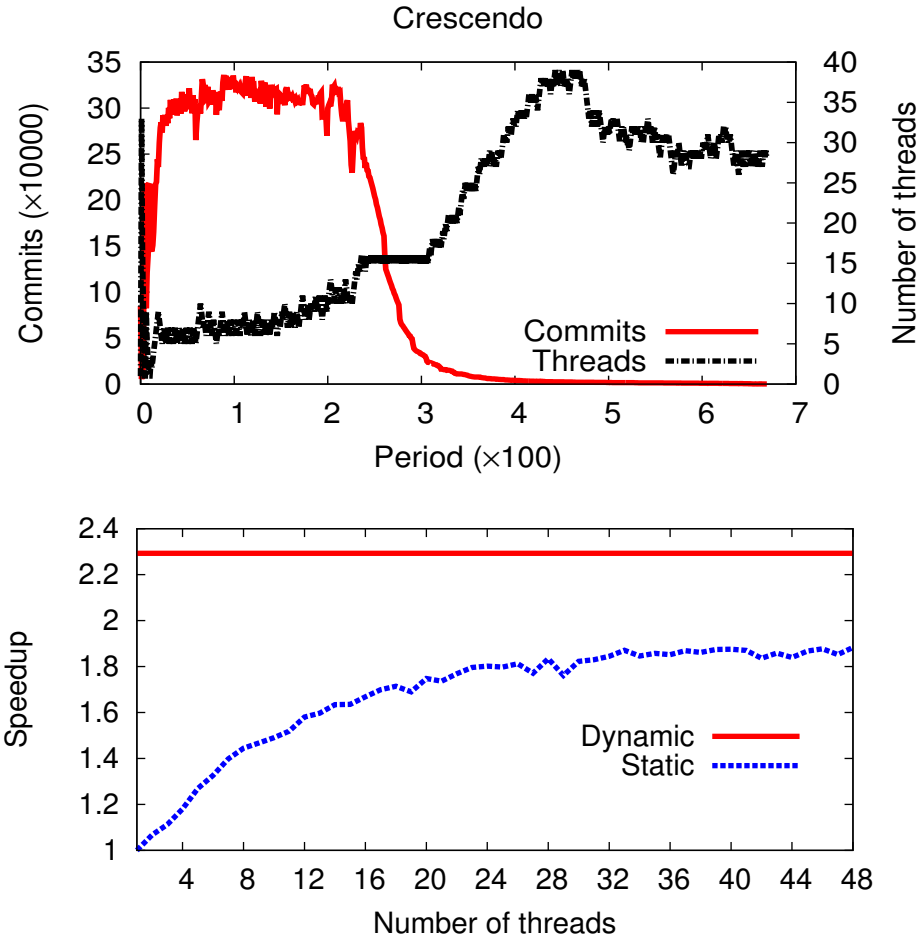
# Solution for shared-memory TM

- Hill climbing-based controller
  - Trial and error
  - Increase/decrease #threads according to feedback



# Adaptation in shared-memory TM



# Parallelism tuning in DTM

Two dimensions (# nodes and # threads/node)

Pure exploration is cumbersome

Changing # nodes is costly!

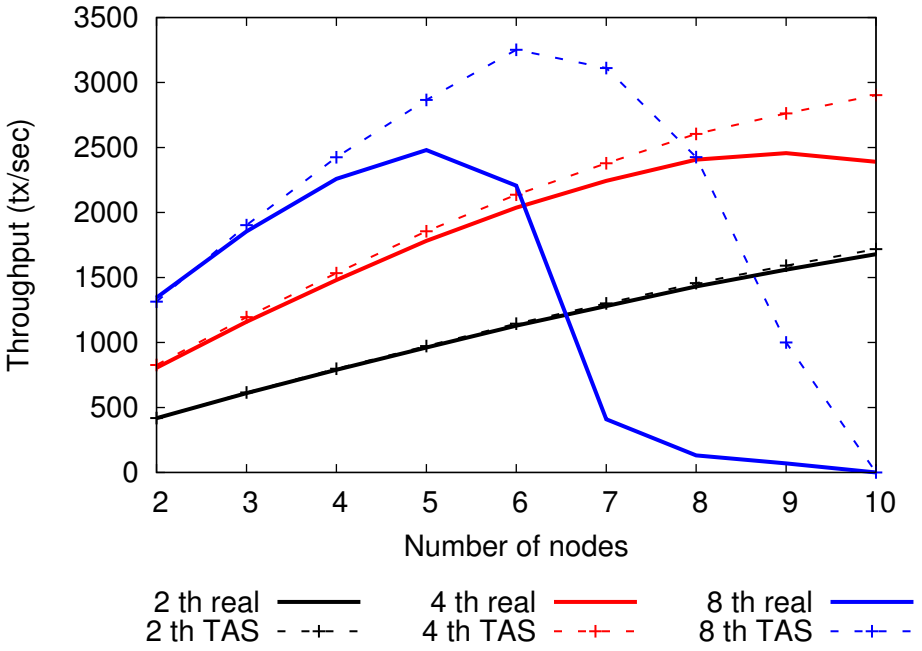
Model-based solution

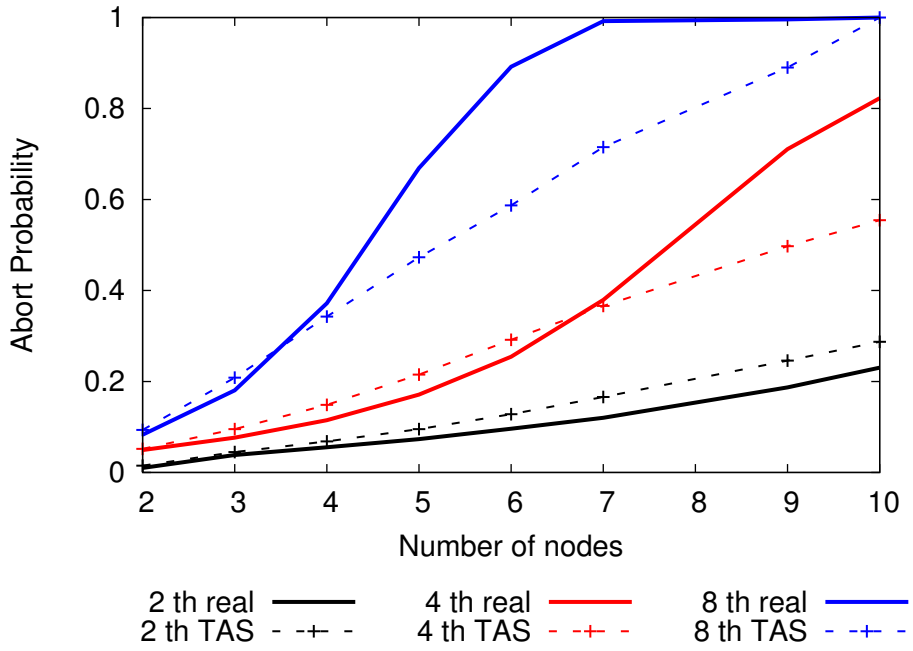
- Input: workload, # nodes, # threads/node
- Output: throughput

Obtain: highest-throughput configuration

# Limitation of performance models

- Models rely on assumptions/approximations
  - Accuracy is hindered when they do not hold…
  - … or in challenging scenarios (e.g., >=50% abort)



# Implemented solution

Exploration + modeling + Machine Learning

- *Gray box solution*

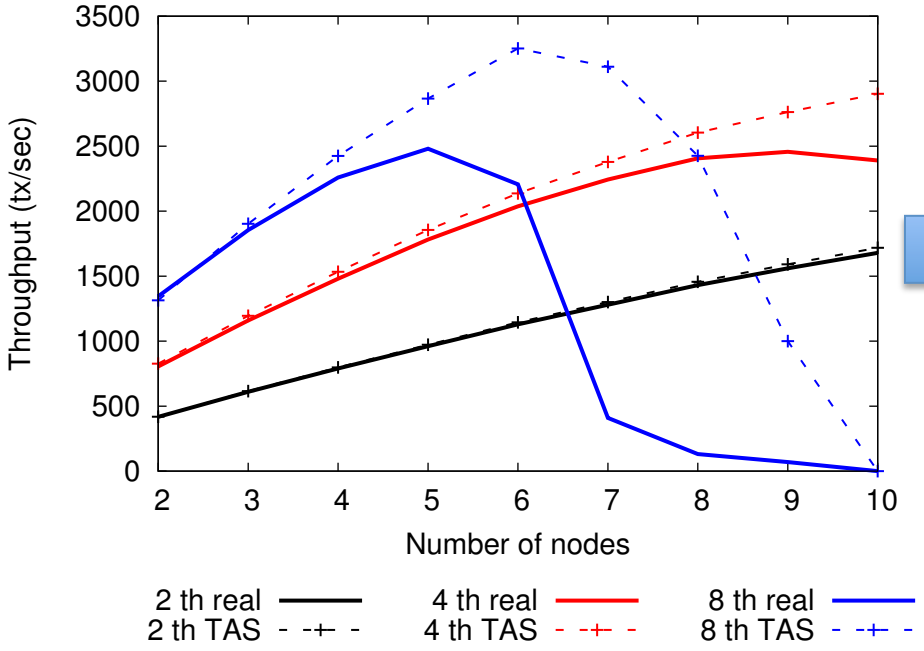
1. Explore to gather feedback on model's accuracy
2. **LEARN** corrective functions to "patch" model

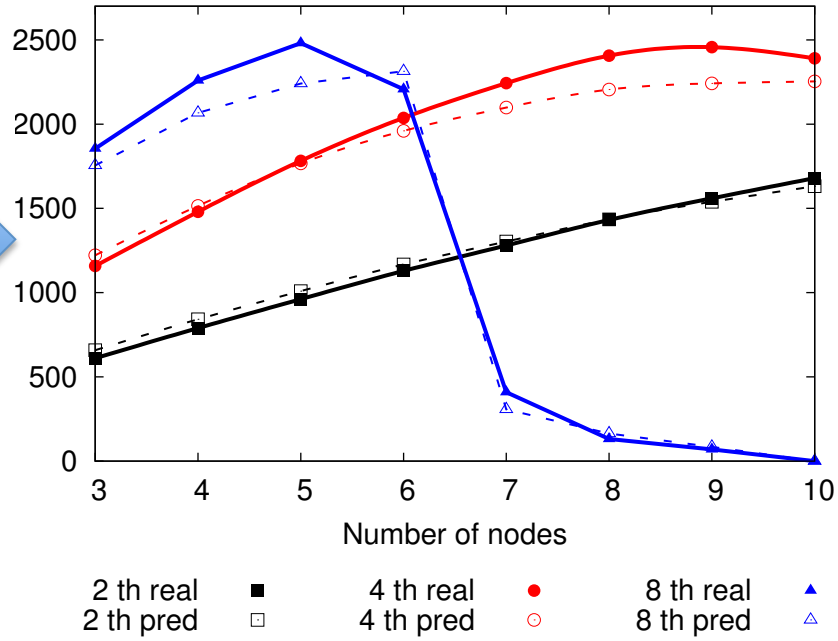
Try to avoid global reconfiguration (# nodes)

- Rely on local # threads exploration (cheap)

Increase accuracy

# Before and After



# Resulting publications

- D.Didona, D. Harmanci, P. Felber, P. Romano, J. Schenker. *Identifying the Optimal Level of Parallelism in Transactional Memory Applications*

- Workshop on Hot Topic in Parallelism (HotPar) 2012
  - Poster

- Int'l Conf. on Networked Systems (NETYS), 2013

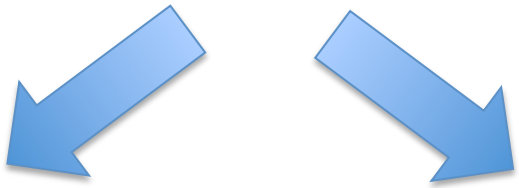
  **Best paper award**

- Computing Journal, Springer, 2013

# Follow ups

- Sparked a broader investigation on gray box
  - turned into the focus of my doctoral studies

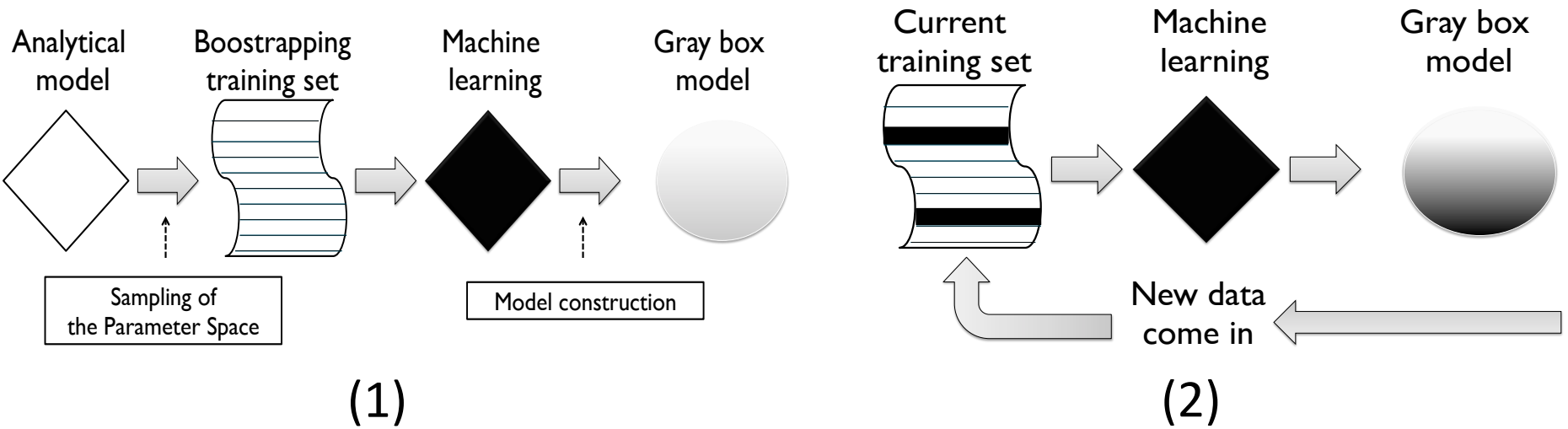
- Build hybrid robust performance models
  - No (or low) training time of analytical modeling
  - Incremental learning of Machine Learning



Bootstrapping Ensemble

# Bootstrapping

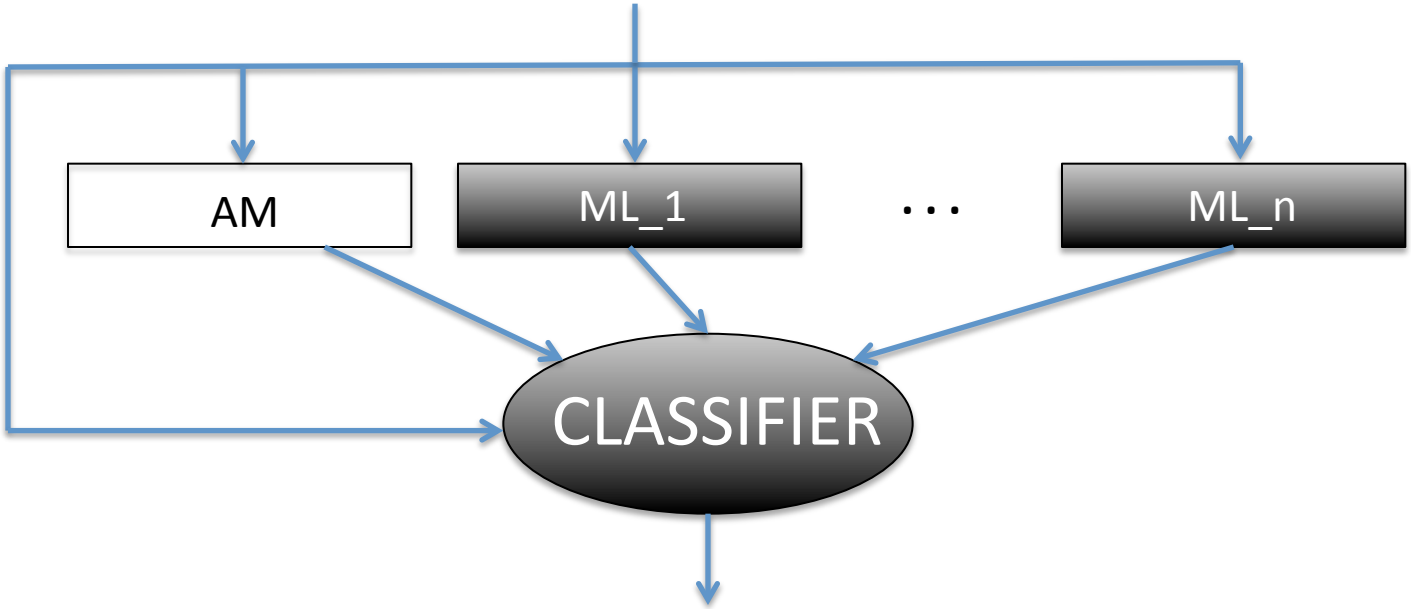
- Train ML on the output of an AM
  1. Initially train the ML at zero (or low) training time
  2. Retrain periodically the ML to increase accuracy



# Ensemble

Use AM and ML in parallel

- Classifier chooses between AM or ML **per query**
- Classifier and the MLs are retrained periodically



# STSM & my PhD thesis

- *Gray Box Performance Modeling of Distributed Applications* (to be defended)

- Taxonomy & description of gray box approaches

- Applications
  - Perf. forecasting (capacity planning, anomaly detection)
  - Resource provisioning (Cloud scenarios)
  - Self-tuning (online optimization)

# Questions?



didona@gsd.inesc-id.pt