SLA Enforcing in Transactional Cloud Data Stores: the Cloud-TM Approach

Paolo Romano

Project goals

Develop a ***data-centric PaaS*** aimed to minimize:

1. **developments costs**:

➪ introducing abstractions aimed to hide complexity

2. **administration costs**:

➪ via QoS based automated resource provisioning

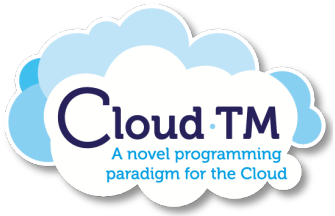
3. **operational costs**:

➪ maximizing efficiency via self-tuning

The Cloud-TM approach

- Innovation in three main areas:

Programming Model

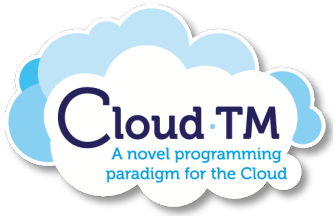


- **Transaction-centric** programming paradigm
  - abstractions aimed to hide complexity of concurrency, distribution and failure management
- Full support for **object-oriented data model**
  - manipulation of objects & associations graphs
- Support for **querying** object-oriented domain:
  - subset of industry standard JP-QL interface

The Cloud-TM approach

- Innovation in three main areas:

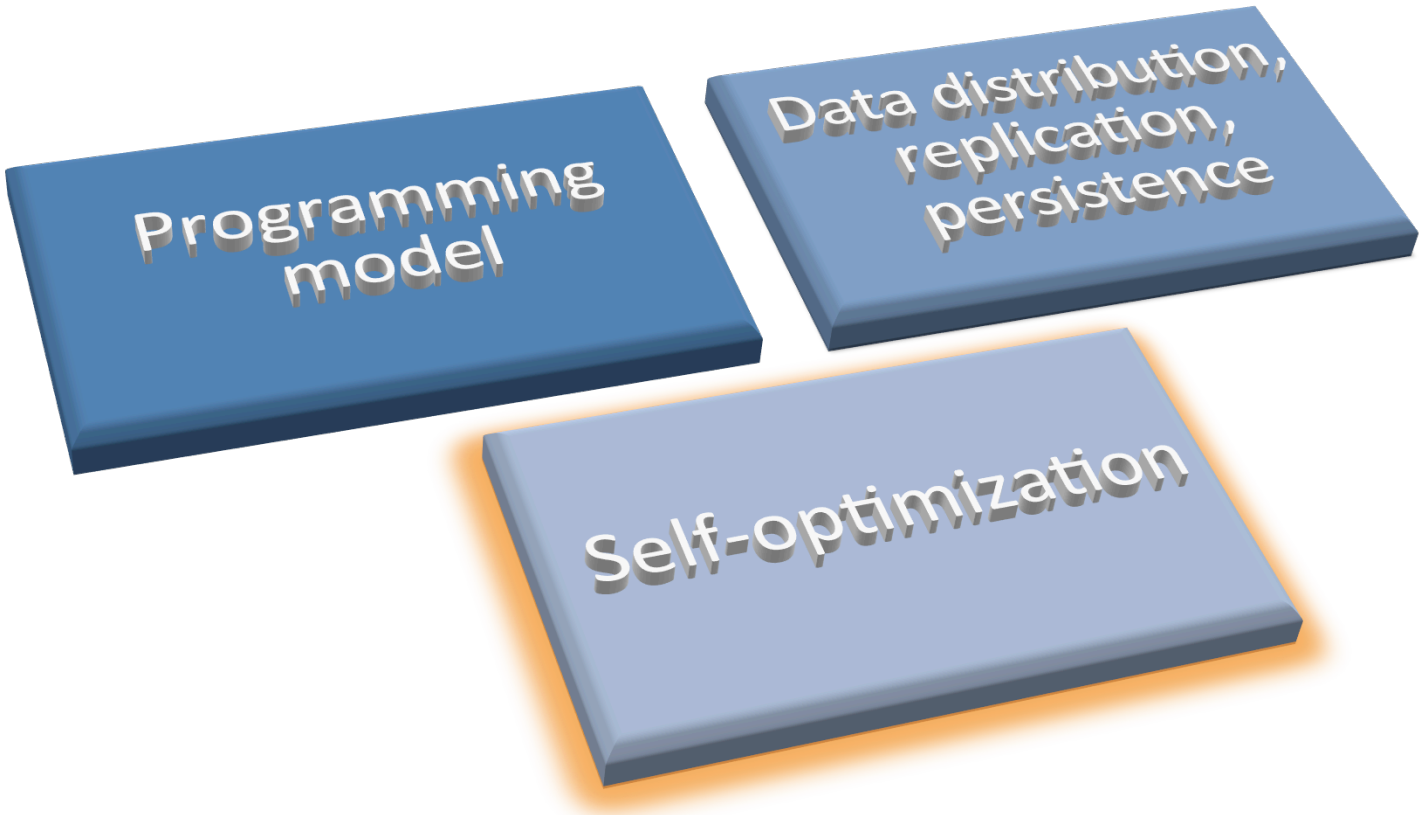
Data distribution, replication, persistence



- Scalability ~~*or*~~ ***and*** consistency?
  - novel data management protocols guarantee strong consistency w/o hampering scalability
- Elastic scaling:
  - high throughput even when rescaling the platform thanks to non-blocking state transfer protocols
- Interoperability with diverse cloud storages:
  - S3, vs local file-system or DBMS

The Cloud-TM approach

- Innovation in three main areas:



Self-optimization

- Seen as a pervasive feature of the platform:
  - monitoring across entire architectural stack
  - workload characterization and forecasting
  - QoS/cost driven self-tuning:
    - automatic elastic scaling
    - adaptive data placement and consistency schemes
    - self-tuning of critical configuration parameters

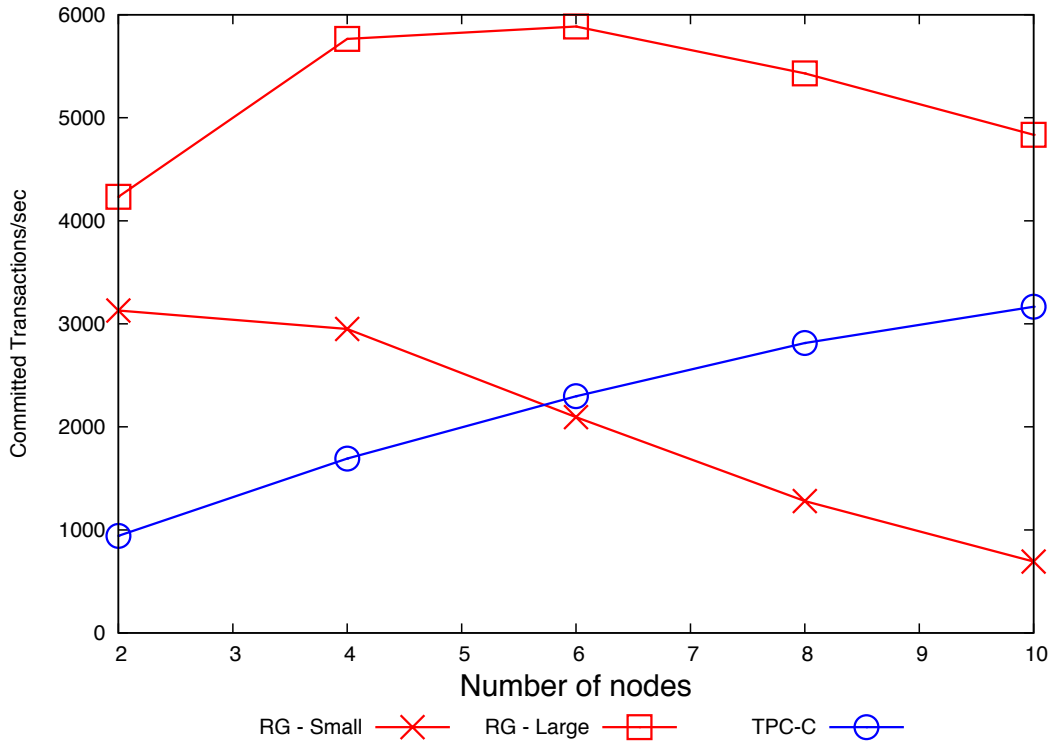
Elastic scaling

- …is one of the most appealing characteristics of cloud computing

- many commercial cloud providers already support for simple automatic elastic scaling policies, e.g.:
  - if CPU/Mem are above threshold X, add a node

What your cloud provider won't tell you…

- Adding one more node will **NOT** always improve your application's performance!

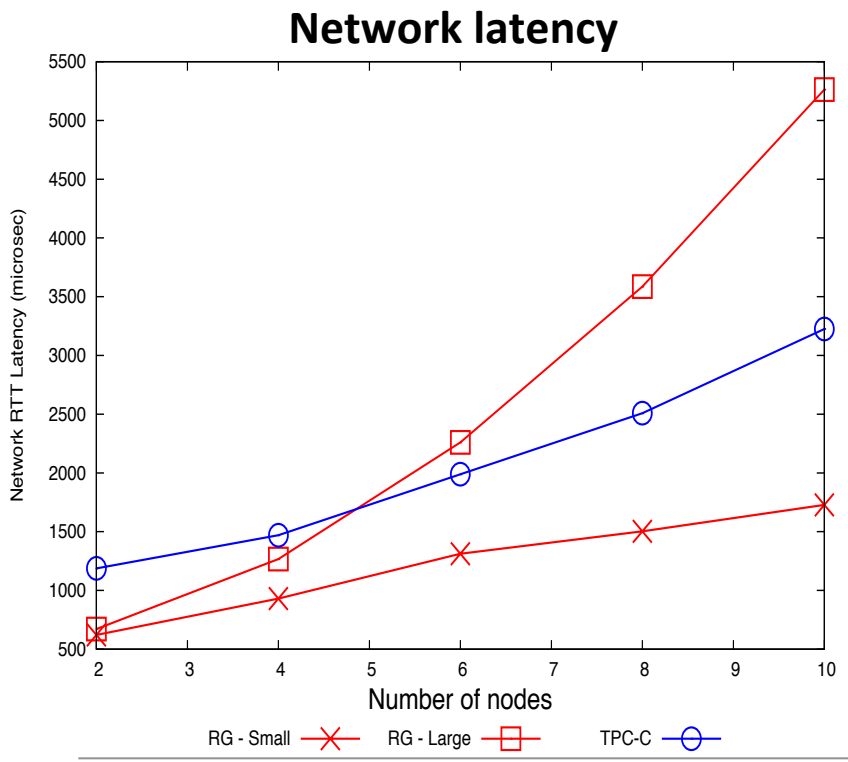


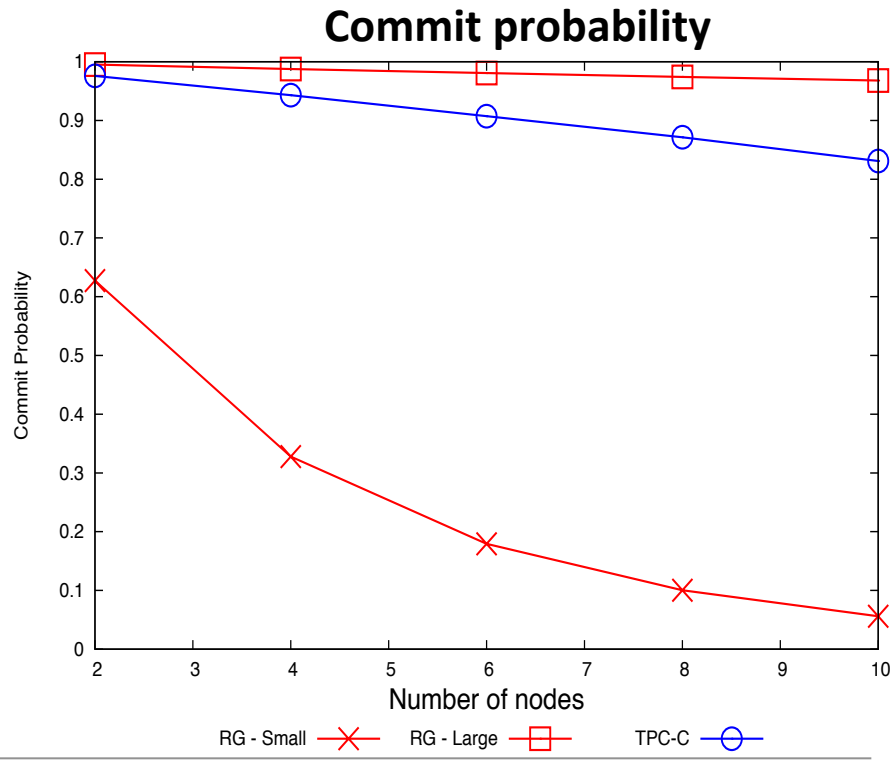
Data collected running heterogeneous workloads on top of a transactional key-value store:

(Red Hat's Infinispan)

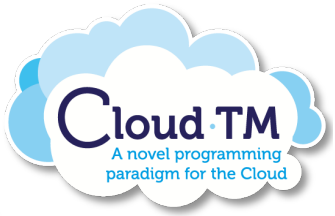
What your cloud provider won't tell you…

- Scaling out can drastically amplify contention at the logical and physical level



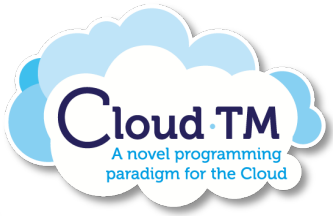


Morale?



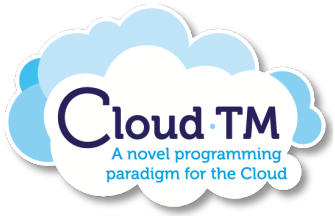
- Adaptation policies based on resource utilization:
  - no guarantees on QoS achieved after adaptation

- Key challenge faced in our project:
  - Predicting performance after elastic scaling requires forecasting the effects of logical/physical contention



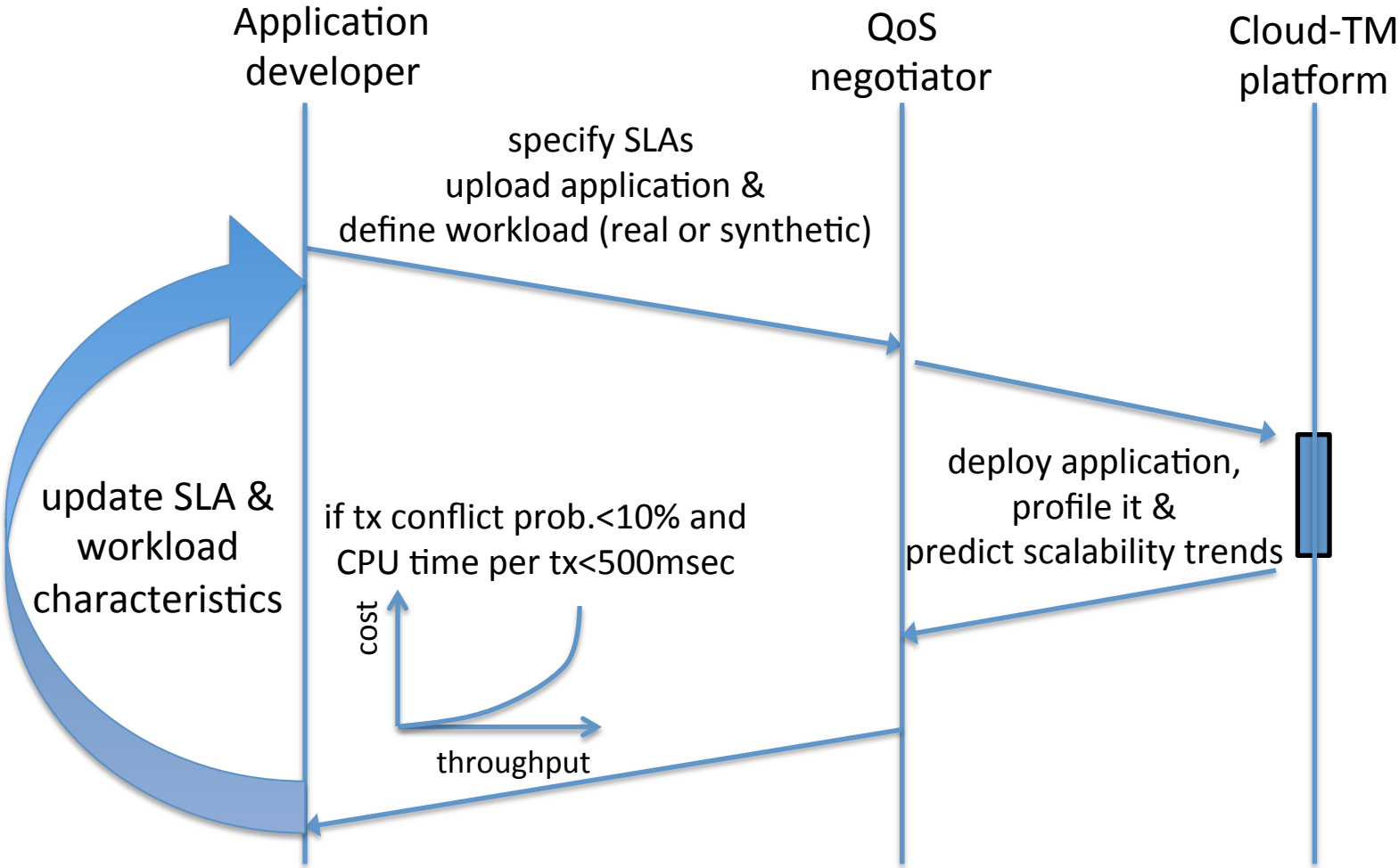
SLA NEGOTIATION IN THE CLOUD-TM PLATFORM

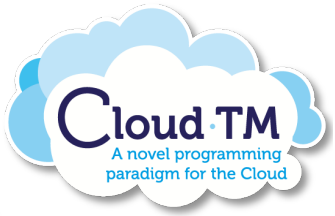
QoS in data-centric applications



- QOS enforcement in data-centric applications requires **mutual obligations between application developers and platform providers**, e.g.:
  - platform providers QoS obligations, e.g.:
    - avg. transaction execution time<1sec
  - applications' obligations, e.g.:
    - tx. execution time in absence of contention < 200msec
    - probability of tx. contention < 1%

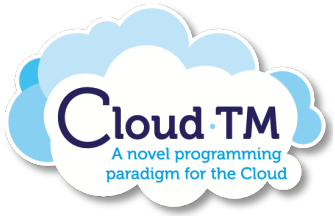
SLA negotiation





KEY RESULTS IN THE AREA OF SLA ENFORCEMENT

Key results in SLA area



- Methods and tools for:
  - performance forecasting
  - workload characterization

of **distributed transactional data stores**

Transactional Auto Scaler [2,4]

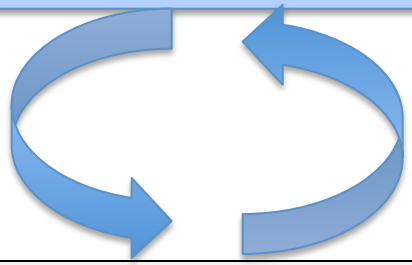
- Innovative methodology for **predicting performance** of transactional applications when deployed over platforms of different scale:
  - instantiated for both 2PC and primary-backup replication
  - methodology is generic enough to be used with different transactional storage systems

TAS: Performance Predictor

Cloud·TM
A novel programming paradigm for the Cloud

Analytical Modelling

- White-box approach:
  - knowledge of consistency algorithms internals
- Good extrapolation power
- Queueing theory

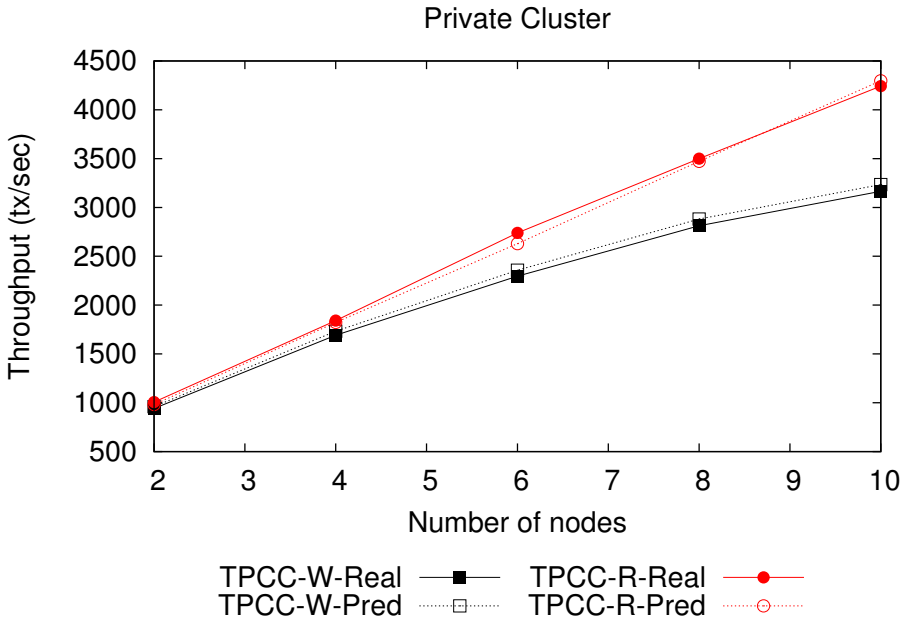


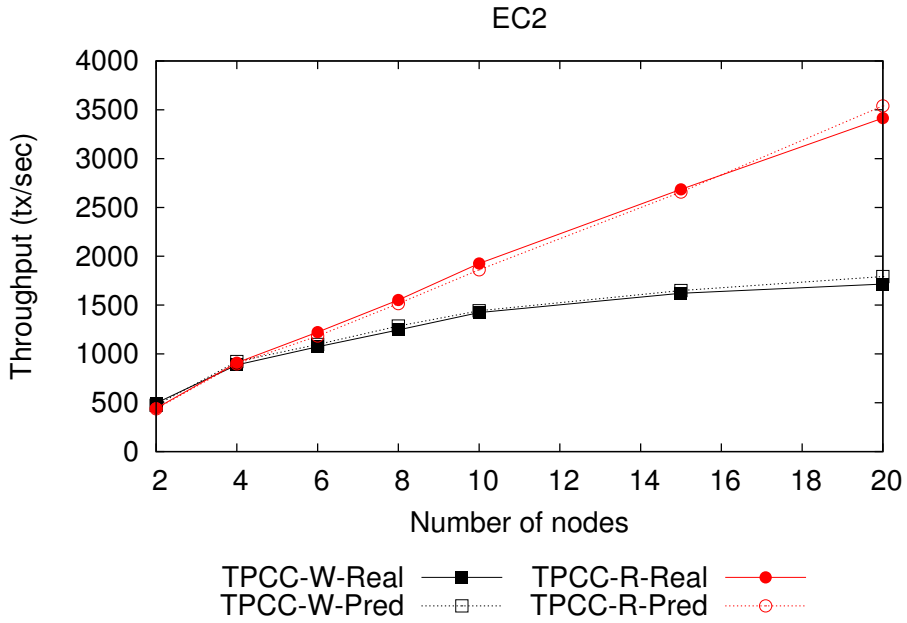
coupled via an iterative fixed-point solution technique

Machine Learning

- Black-box approach:
  - hard to model physical layer's dynamics (or even knowing them)
- Suitable for virtualized environments
- Offline trained decision-tree regressors

TAS Accuracy



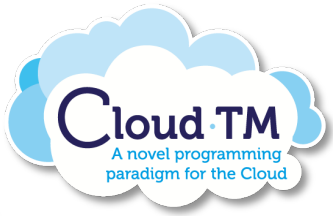


Key results

- Methods and tools for:
  - performance forecasting
  - workload characterization

of **distributed transactional data stores**

Measuring application scalability

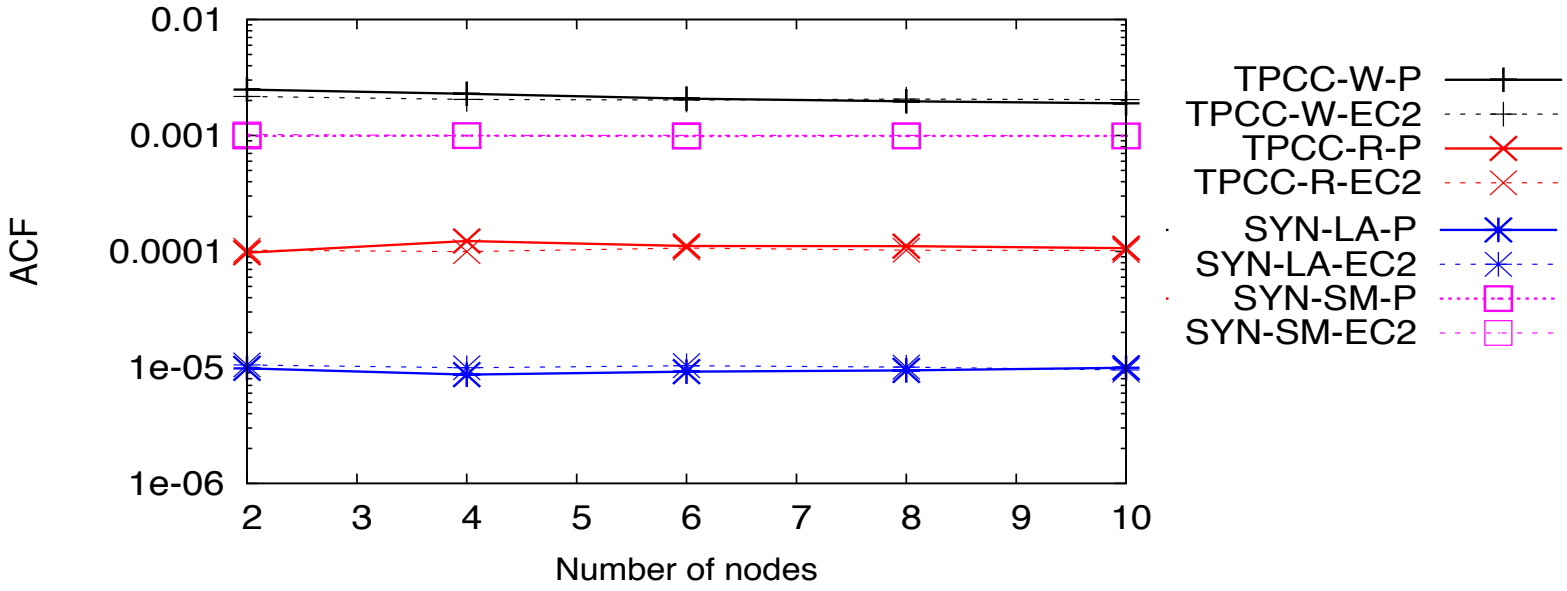


Can we measure the scalability of a **transactional application** independently from:

- the algorithm used by the transactional data platform on which it is deployed
- the capacity of the current platform (e.g. CPU speed)
- the data management scheme we are using (e.g. 2PC vs Primary backup protocols)

Application Contention Factor [2]

- Scalar metric based on queuing theory arguments
- Automatically computable by monitoring:
  - lock duration, arrival rate and conflict probability
- Independent from scale and deployment platform:



Conclusions

- Key results (in the QoS area)
  - Methods and tools for **workload characterization and performance forecasting** of transactional applications
- Key standardization opportunities:
  - **QoS & cost specification** based on custom SLA@SOI templates
  - SLA negotiation process and tools

References

[1] Joao Paiva, Pedro Ruivo, Paolo Romano and Luis Rodrigues, AutoPlacer: scalable self-tuning data placement in distributed key-value stores, The 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, USA, 26-28 June 2013

[2] Diego Didona, Pascal Felber, Derin Harmanci, Paolo Romano and Joerg Schenker, Identifying the Optimal Level of Parallelism in Transactional Memory Systems, The International Conference on Networked Systems 2013 , BEST PAPER AWARD

[3] M. Couceiro, P. Ruivo, Paolo Romano, L. Rodrigues, Chasing the Optimum in Replicated In-memory Transactional Platforms via Protocol Adaptation, The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2013)

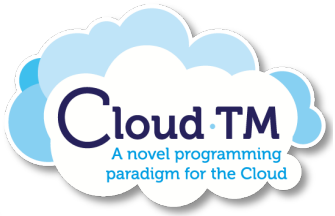
[4] D. Didona, Paolo Romano, S. Peluso, F. Quaglia, Transactional Auto Scaler: Elastic Scaling of In-Memory Transactional Data Grids, The 9th International Conference on Autonomic Computing (ICAC 2012), San Jose, CA, USA, 17-21 Sept. 2012

[5] 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

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

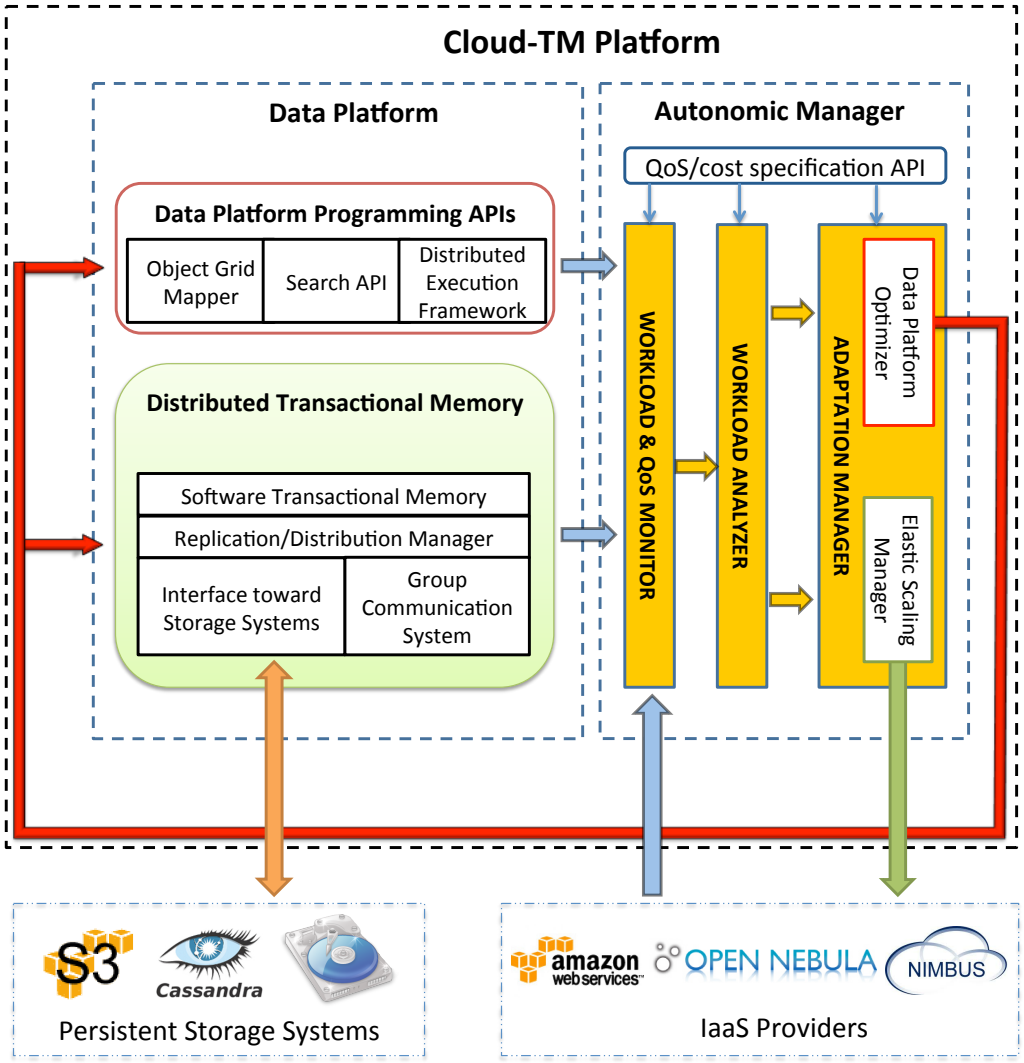
[7] P. Di Sanzo, F. Antonacci, B. Ciciani, R. Palmieri, A. Pellegrini, S. Peluso, F. Quaglia, D. Rughetti and R. Vitali, A Framework for High Performance Simulation of Transactional Data Grid Platforms, Proc. 6th International ICST Conference on Simulation Tools and Techniques (SIMUTools), Cannes, French Riviera, 2013.

[8] P. Di Sanzo, D. Rughetti, B. Ciciani and F.Quaglia, Auto-tuning of Cloud-based In-memory Transactional Data Grids via Machine Learning, Proc. 2nd IEEE International Symposium on Network Cloud Computing and Applications (NCCA), London, UK, IEEE Computer Society Press, December 2012.

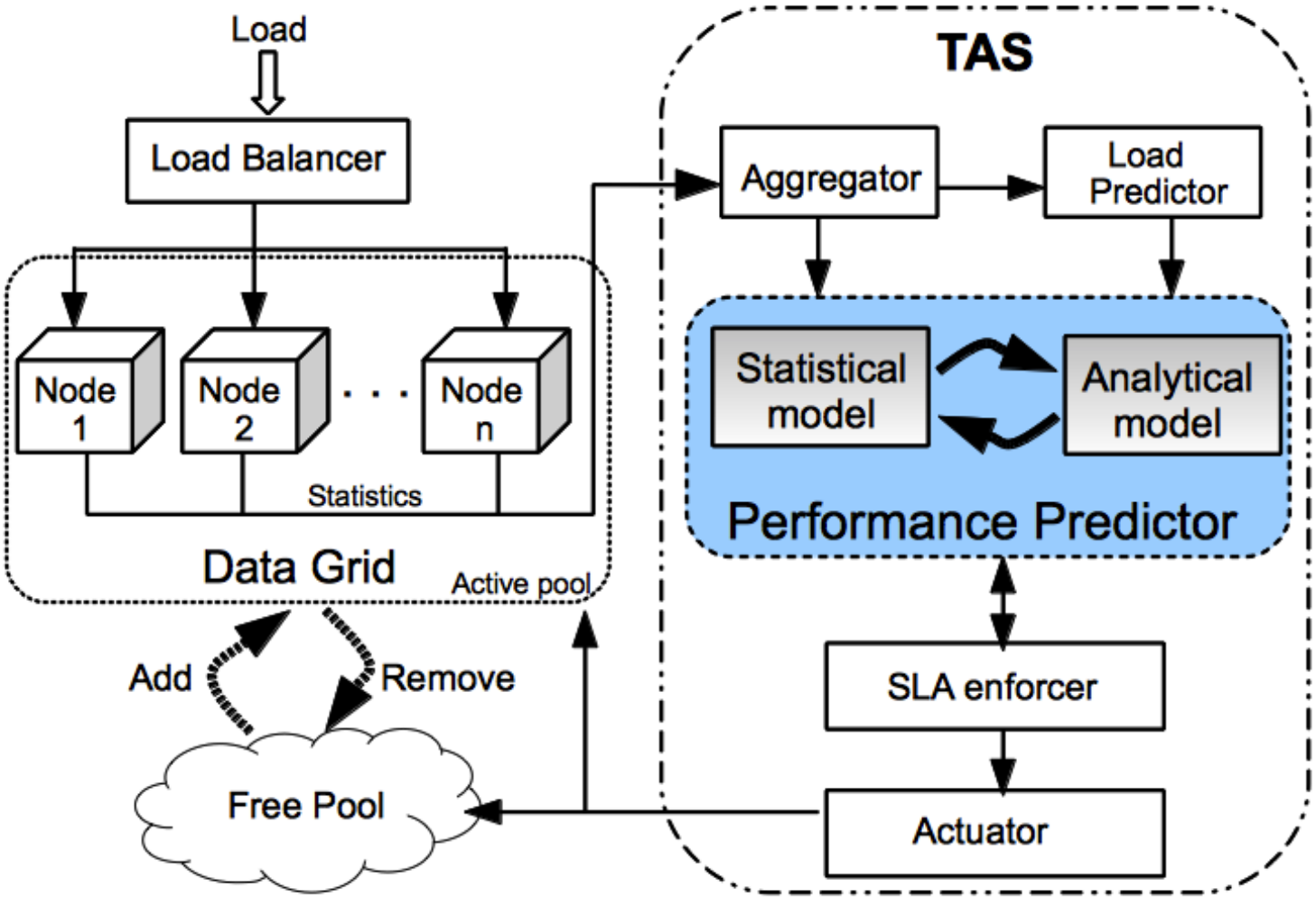


BACKUP SLIDES

Architecture Overview



TAS' Architecture



Cloud computing: issues addressed in Cloud-TM

- Lack of programming models effectively hiding the issues of:
  - concurrency
  - distribution
  - fault-tolerance
  - ela

Complexity

- Lack of effective tools to automate elastic scaling:
  - manual monitoring expensive and error-prone
  - existing techniques provide no guarantee on perceived QoS

- Performance tuning is essential in an elastic cloud infrastructure, but costly and error prone