Euro-TM Workshop on Transactional Memory

Bern, Switzerland, April 10, 2012

Differentiated Access to Virtual Resources in Cloud Environments

M. Fazio and A. Puliafito

mfazio{apuliafito}@unime.it

www.mdslab.unime.it

Outline

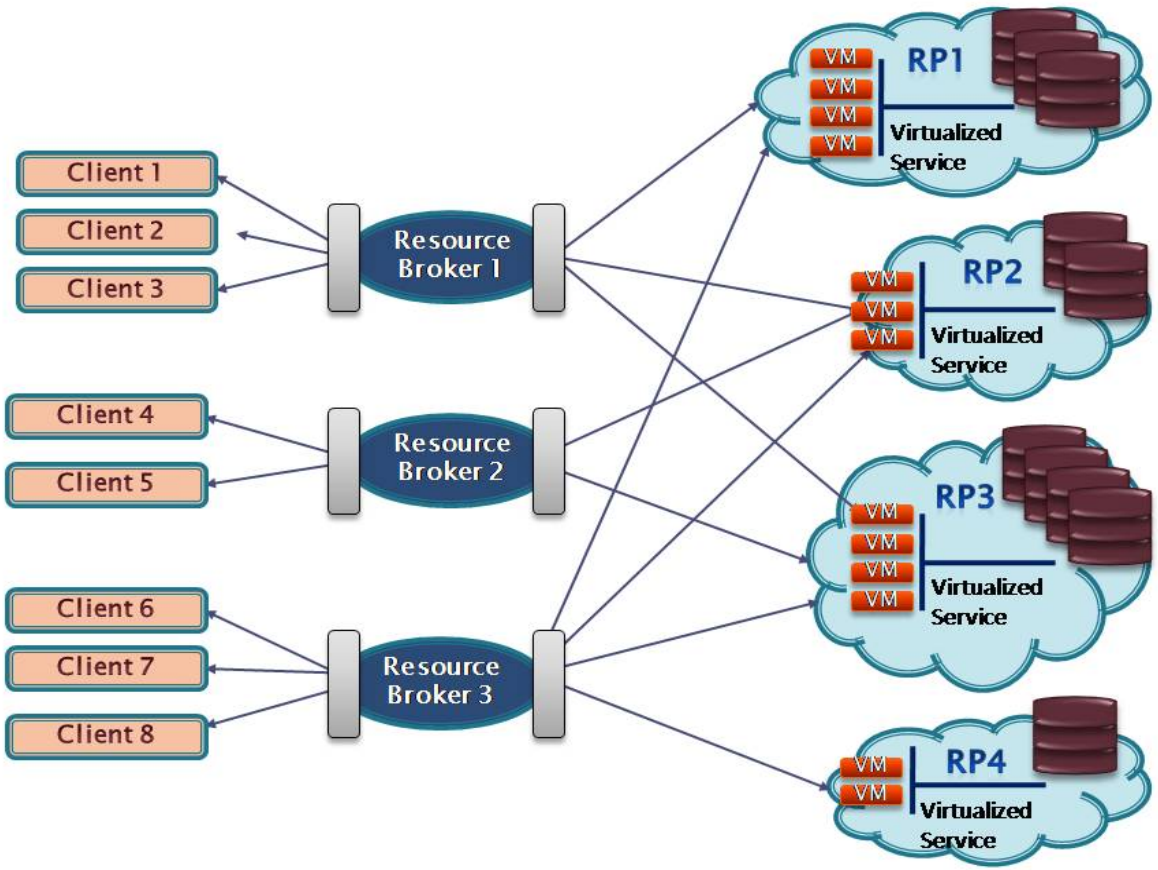
- Resource management in Cloud environments
- How STM may be useful for resource management in the Cloud
- Our proposed solution
  - Resource & QoS Manager (RQM)
  - Cloud@Home project
- Conclusions

Resource management in Cloud environments

- One of the main challenges in cloud computing is the efficient management of virtual resources offered from cloud providers to their clients.
- Resource management means:
  - locating available resources
  - allocating resources to user requests
  - satisfy the Service Level Agreement (SLA) with the user
  - offering advanced services to provide efficient usage of resources (monitoring, migration, redundancy,...)

Realistic resource management scenario

- Resource Providers (RPs)
  - set of hosts and clusters which provide the physical resources
- Clients
  - users or applications making the requests of resources.
- Resource Brokers (RBs)
  - take care of resource allocation and use services and/or applications on multiple virtual machine
  - Kaavo, RightScale, Cloudkick...

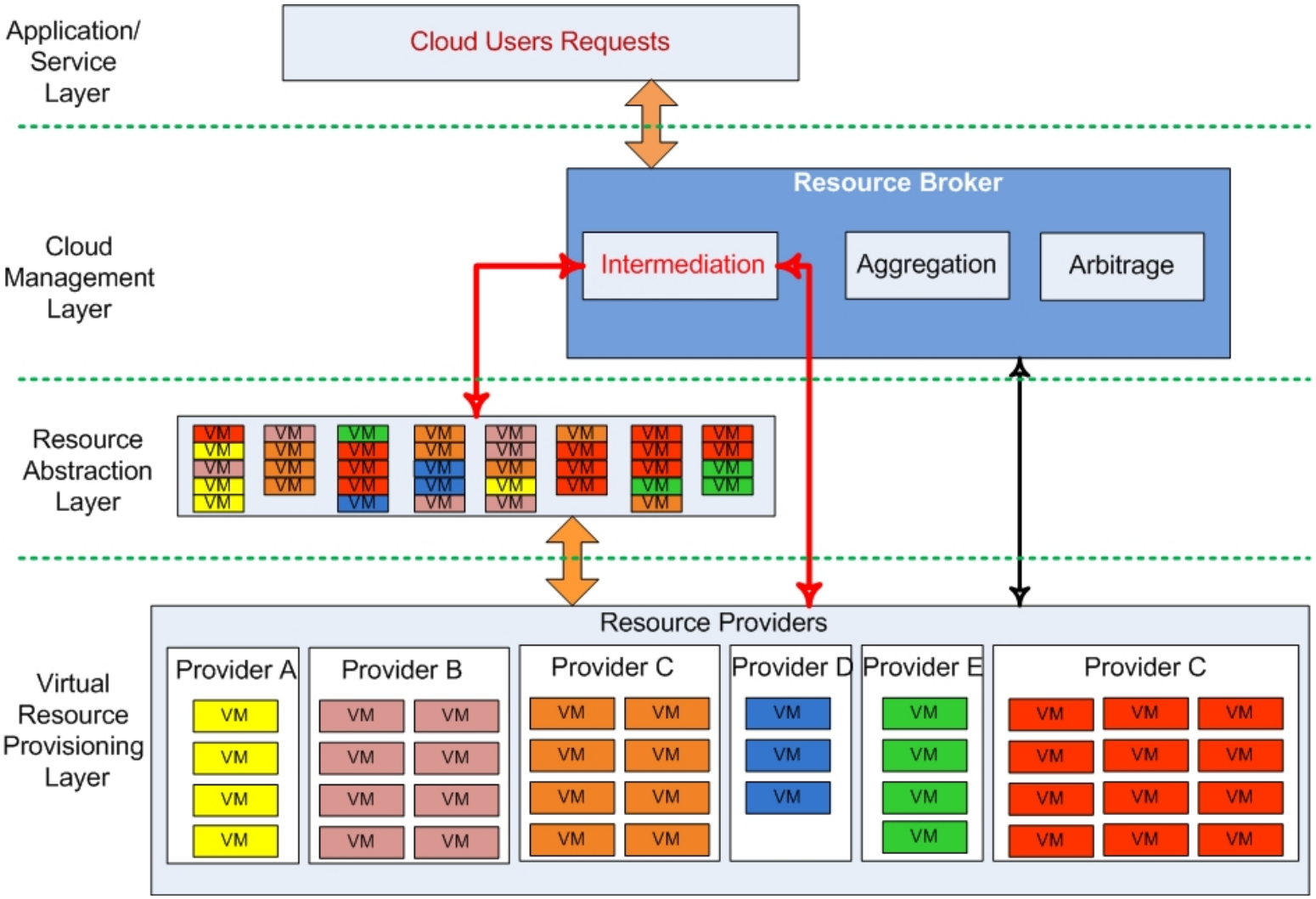


Aim of our work

We propose a new architecture for resource management in the Cloud, which allows RBs to perform the following tasks:

- Intermediation
  - building basic services to perform resource provisioning for applications/services
  - differentiated SLA according to different charges
- Aggregation
  - deploying customer services over multiple cloud platforms
- Arbitrage
  - Brokers supply flexibility and opportunistic choices and foster competition between RPs

Resource Management Architecture

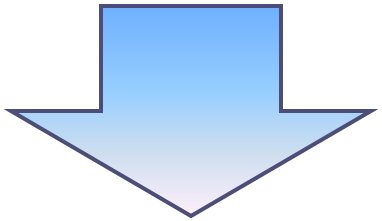


Resource Abstraction Layer

The Resource Abstraction Layer (RAL) provides an abstraction of all the virtual resources provided by many RPs.

It has to:

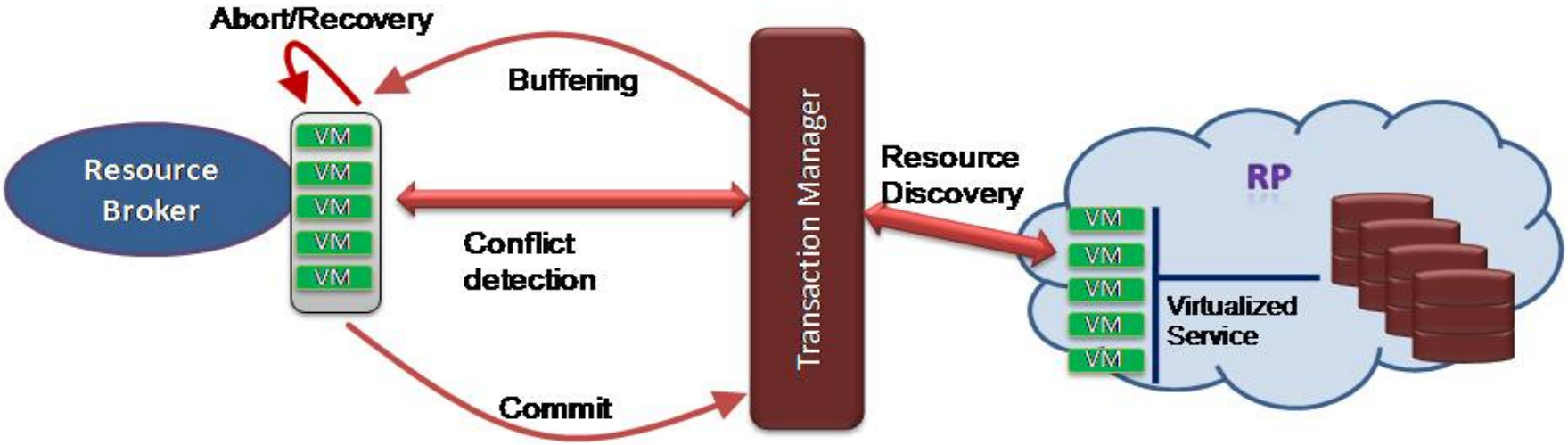
- handle concurrent client requests from a RB
- manage multiple accesses from several RBs to resources in a transparent and effective way
- react to changes in the availability of resources due to RPs



The design of the RAL is based on the Software Transactional Memory technology

STM in our Resource Management architecture

- the RAL is hosted by the Transaction Manager (TM), an agent running on the STM platform
- The TM interact with RPs during the Resource Discovery process, necessary to detect and periodically refresh all the available virtual resources.
- On the other side, the interaction with the RBs is aimed at the management of the resources.

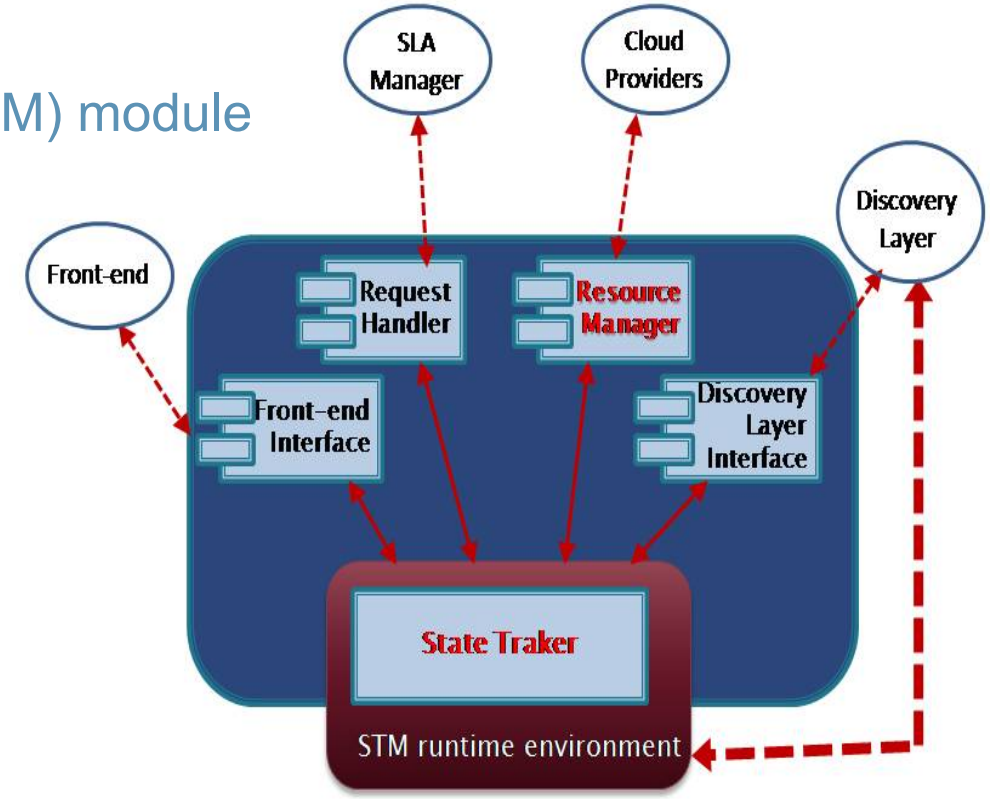


Cloud@Home project

The core idea behind the Cloud@Home project is to use voluntary contributors as cloud resource providers

Resource & QoS Manager (RQM) module

The central part of the RQM is the State Tracker, which keeps the state of the client requests, given by the Request Handler, their associated resources, found by the Discovery Layer, and managed by the Resources Manager.



State Tracker (ST) and Resource Manager (RM)

- When a request arrives to the RM, it has to decide if the user request can be satisfied and, eventually, it allocates a resource to it
  - the RM could spend a lot of time before deciding how to settle a specific request, due to several factors, such as discovery of resources able to fulfill the request, priority in request queues, renegotiation of SLA…
- there is no need to lock the virtual resources in the State Tracker during the whole decision tasks

System Prototype

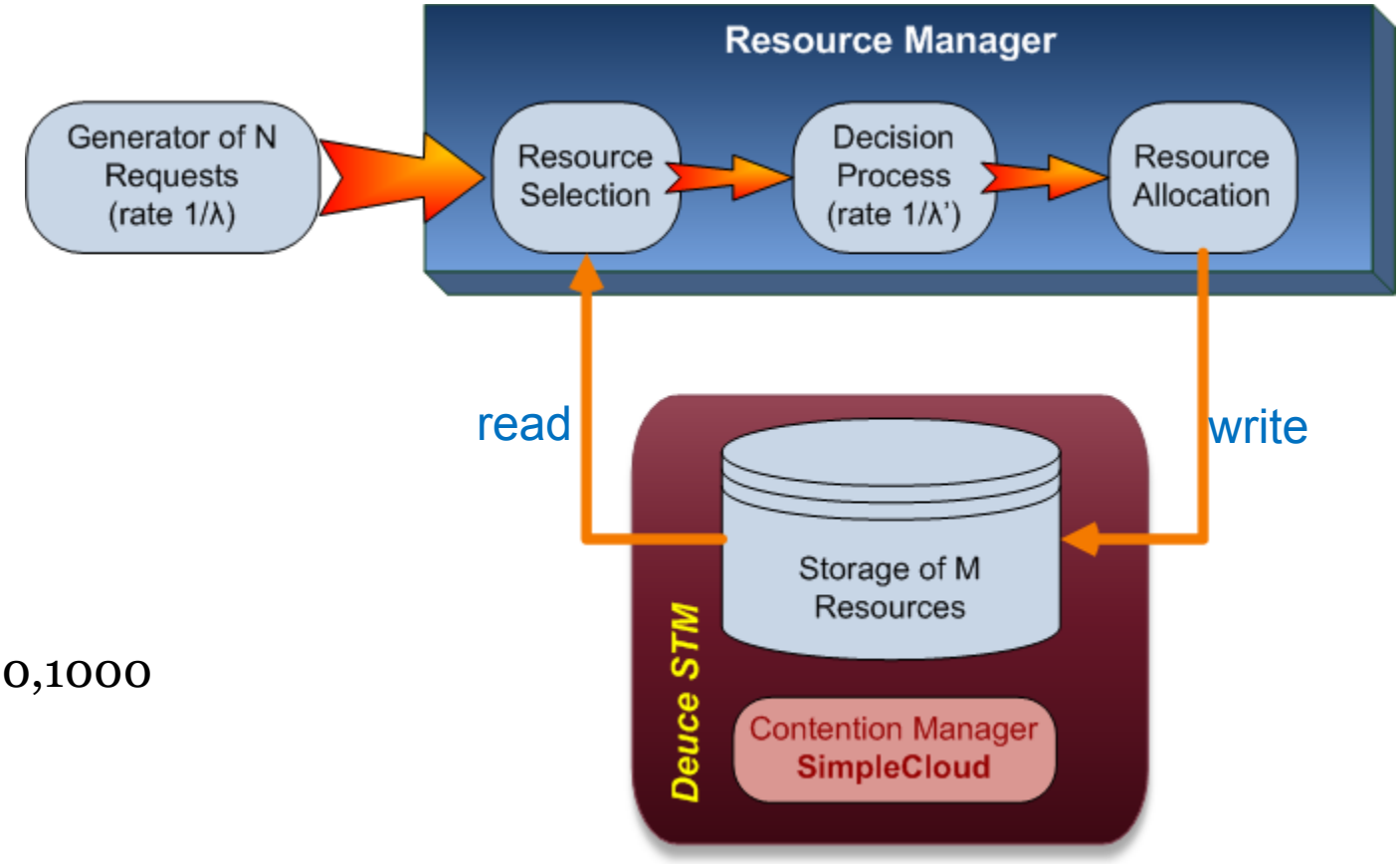
State Traker

- stores all the available resources in the system
  - virtual resources for clients
  - requests of clients
  - state of processes in the system
  - …
- Virtual resource are organized in a table and each entry specifies:
  - `int id ;`
  - `String Description ;`
  - `boolean Availableflag ;`
  - `boolean Reservedflag ;`
- a transaction involves the writing of the `AvailableFlag` of a table record

System Prototype

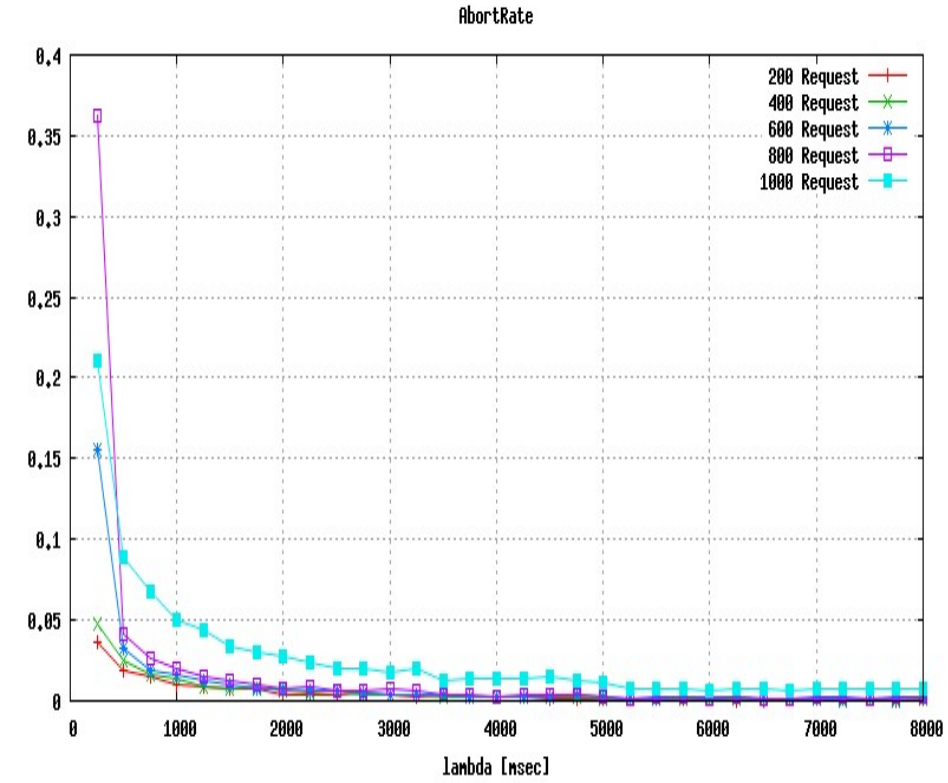
SimpleCloud CM

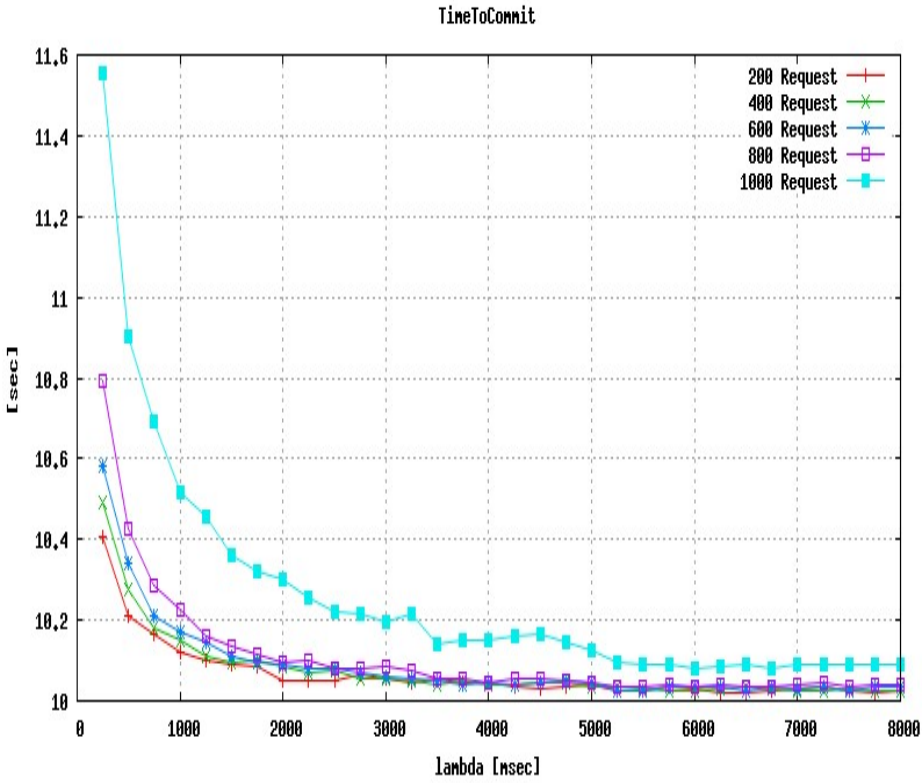
- it aborts the other transaction in write conflicts (Aggressive Policy)
- if aborted, a new transaction starts on a different resource



λ' = 10 sec
λ =[0.25 ; 8] sec
M=1000
N=200,400,600,800,1000

Performance Evaluation





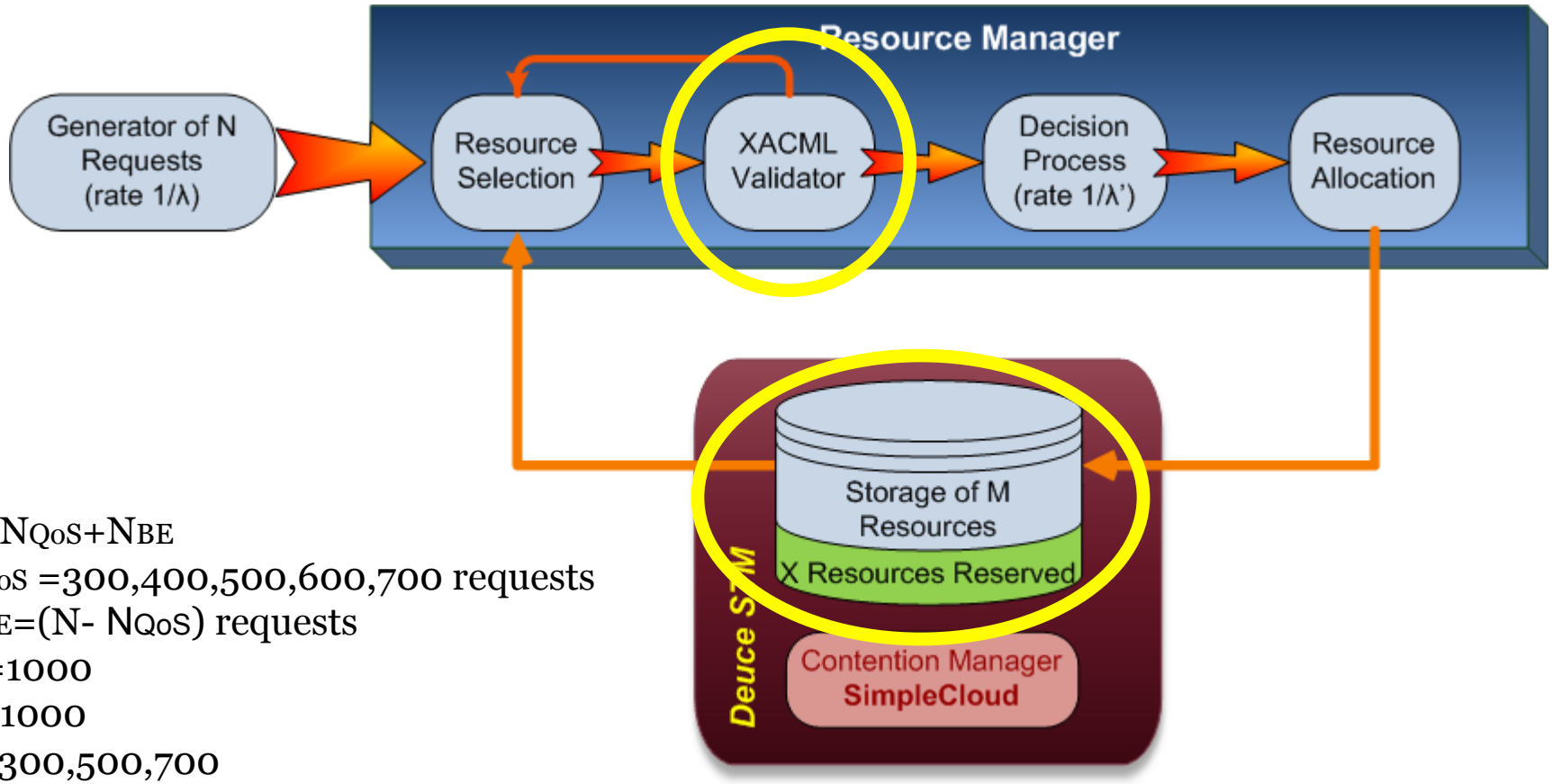
Differentiated services

- Different types of client requests mean differentiated services
  - virtual resources availability
  - response time
- Policies for managing SLAs depends on the system requirements
- Coupling XACML and STM technologies to implement differentiated accesses to available resources

XACML (eXtensible Access Control Markup Language)

- a Rule is the most elementary unit of policy
- rules are not exchanged amongst system entities. Therefore,
- they need to be included in a Policy
- a Policy Set allows to combine several Policies.

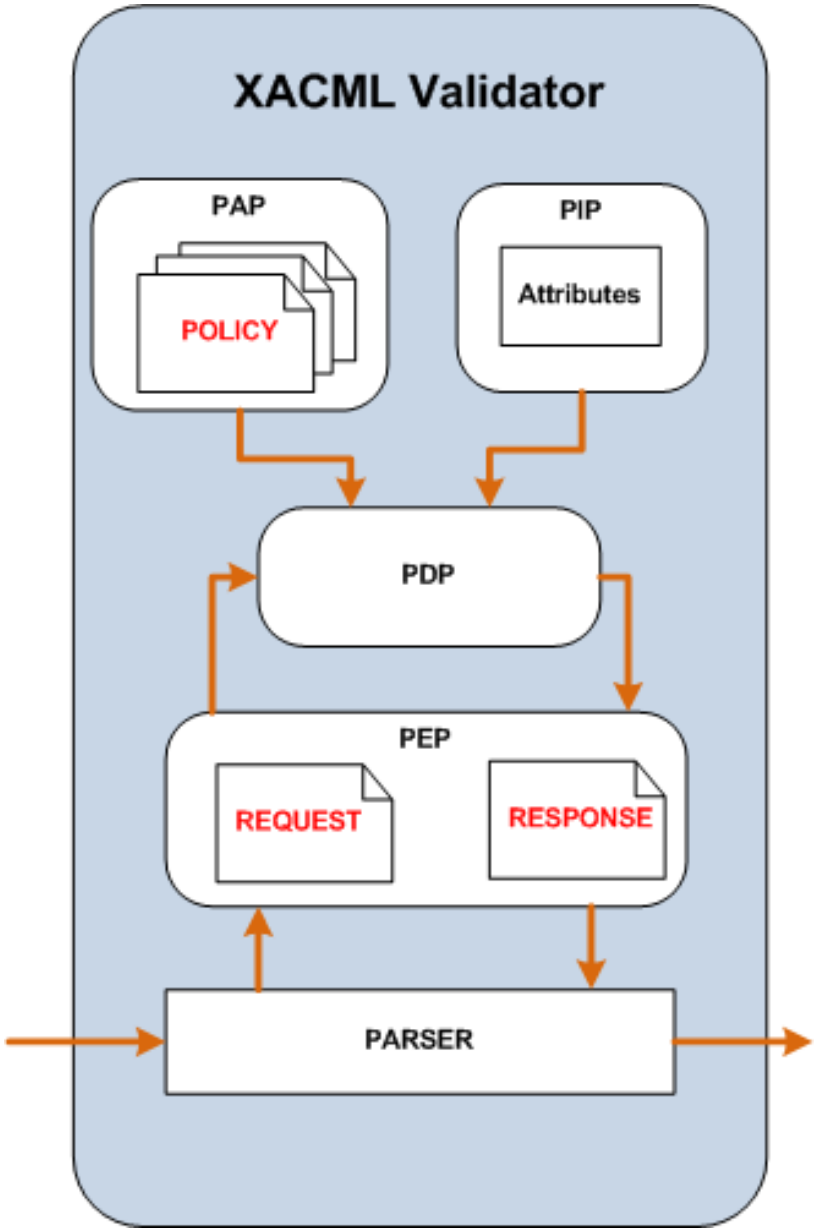
XACML Validation



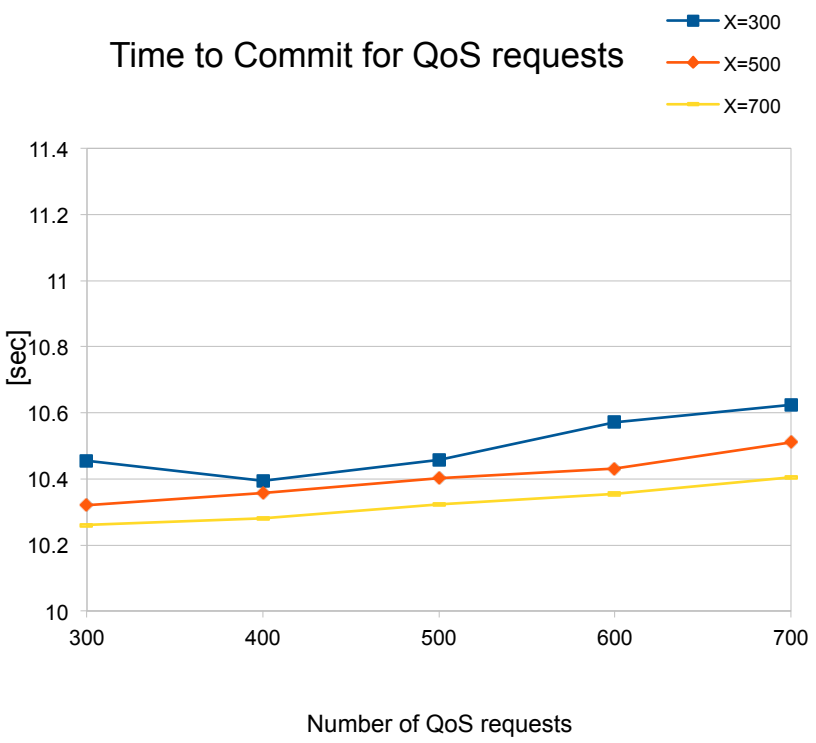
$N=N_{QoS}+N_{BE}$
N_{QoS} =300,400,500,600,700 requests
$N_{BE}=(N- N_{QoS})$ requests
M=1000
N=1000
X=300,500,700

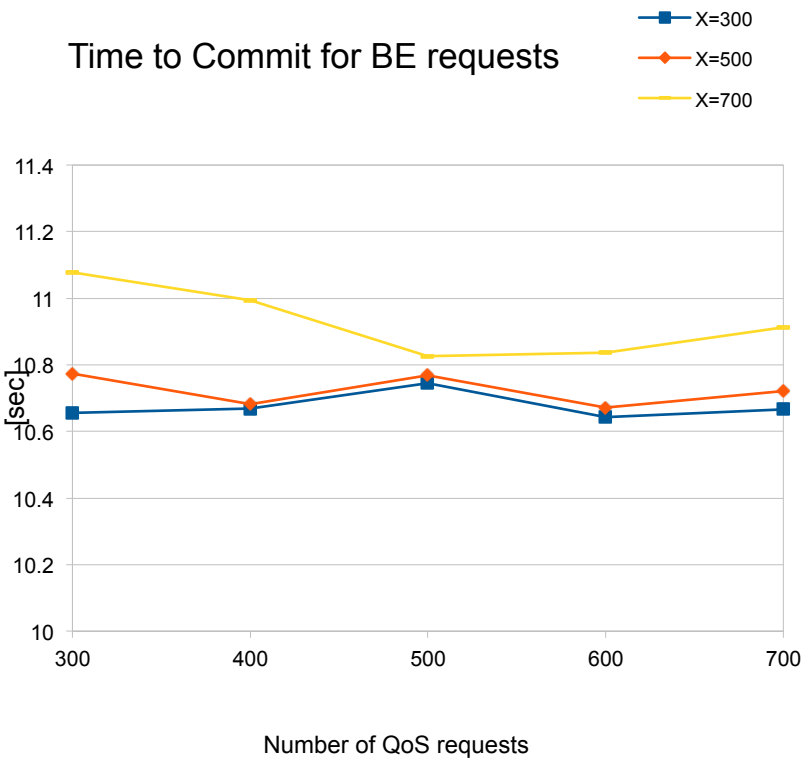
XACML Validation

- **PEP** (Policy Enforcement Point): gets users request and enforce decisions of the PDP;
- **PDP** (Policy Decision Point): evaluates the pair (request, policy) in order to provide a decision;
- **PAP** (Policy Administration Point): policy maker;
- **PIP** (Policy Information Point): stores all the information necessary for an XACML validation

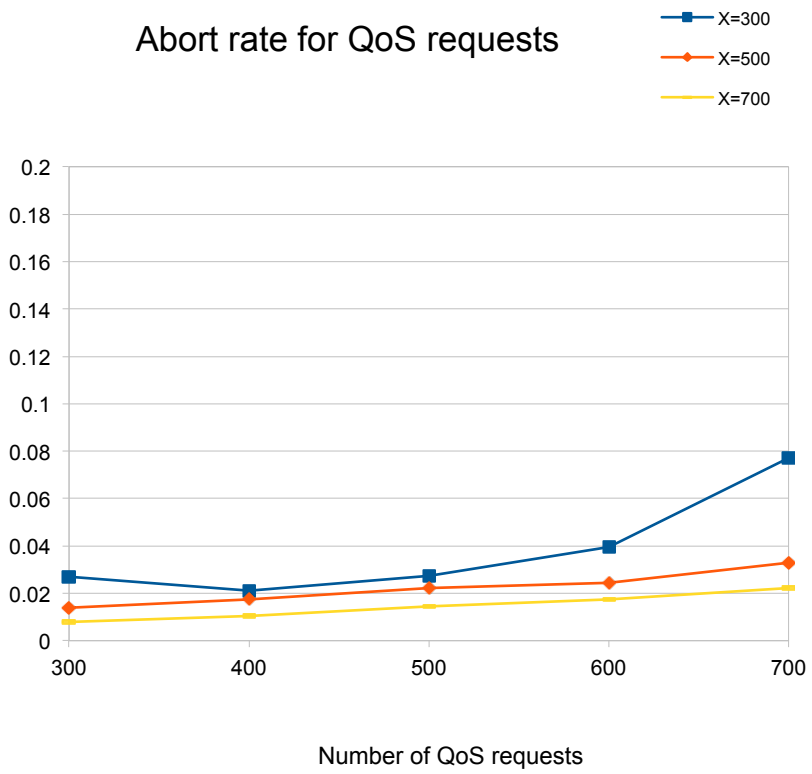


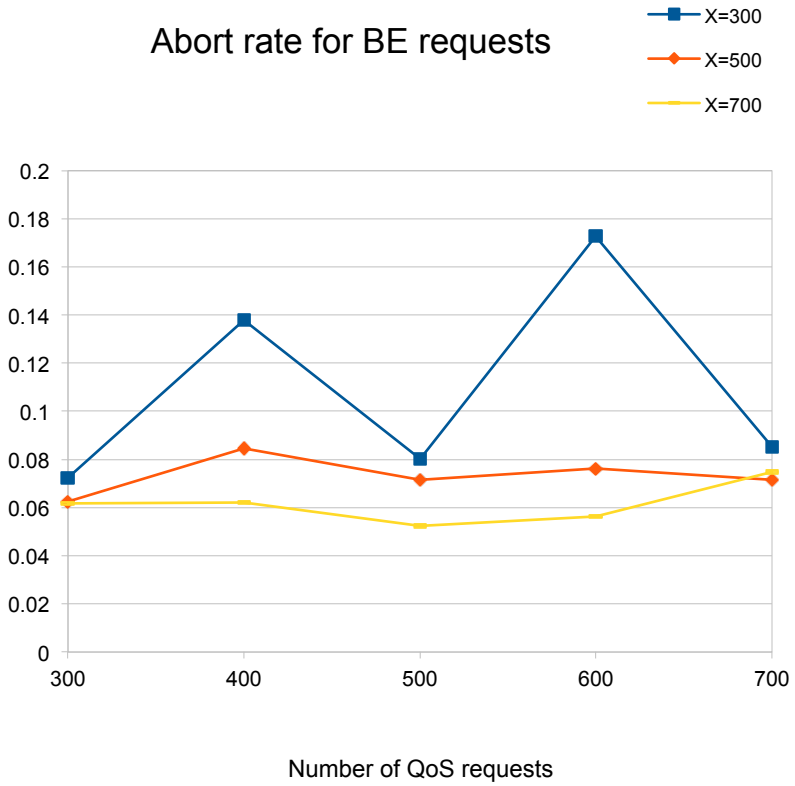
Performance Evaluation



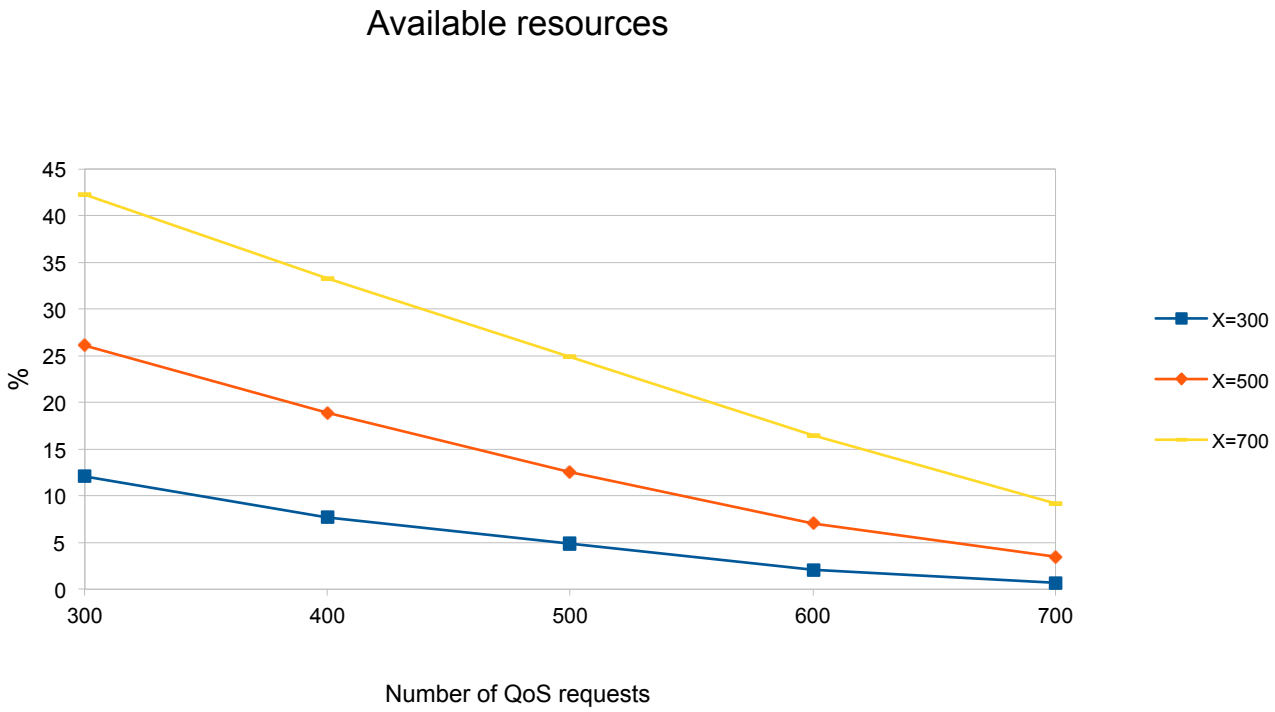


Performance Evaluation





Performance Evaluation



Conclusions

A new architecture for the management of virtual resources in cloud environments

- based on the STM and XACML technologies
- support the provisioning of heterogeneous resources
- address resources from multiple cloud providers
- manage concurrent requests from users
- provide Quality of Service (QoS)

Next step:

- making use of Distributed STM to improve the scalability of our system

Thank You

For Your Attention