RMAC: Resource Management Across Clouds

Vladimir Vlassov vladv@kth.se
Ahmad Al-Shishtawy ahmadas@kth.se
Seif Haridi seif@sics.se

KTH Royal Institute of Technology
SICS Swedish Institute of Computer Science
Stockholm, Sweden

EIT ICTlabs RCLD RMAC project kickoff meeting
Delft, February 20, 2012

KTH-SICS Joint Research Group

ROYAL INSTITUTE OF TECHNOLOGY

Cloud Research Areas

- Distributed storage: Key-value stores
  - Cloud-based, P2P
  - Elasticity, replication, consistency, optimization on queries
  - Partition tolerance (in dynamic environments)
- Computational storage
  - Programming model (storlets), execution environment
- Elasticity
  - Automation (elements of control theory, machine learning)
  - Fine-grained, fast resource allocation
  - Across clouds
  - Building on Mesos (Berkeley)
- Cloud management
  - Distributed and federated clouds
  - Building on Mesos (Berkeley), Grid4All, Selfman

The Carrier Project: E2E Cloud (2011-2016)

E2E Cloud: End-to-End information-centric Cloud

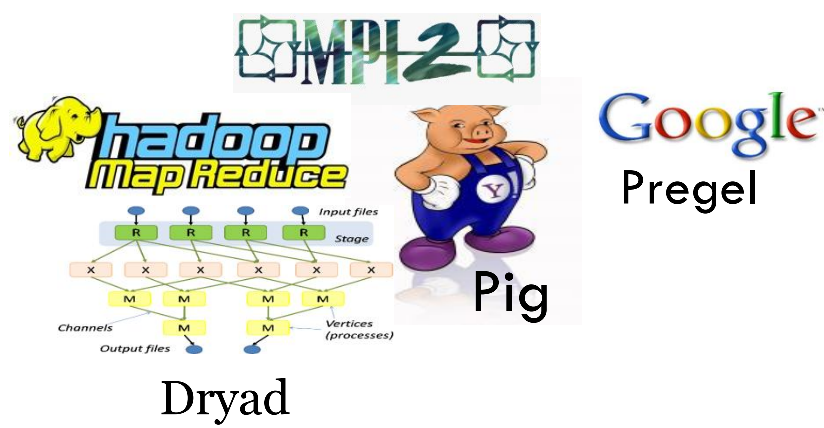
- Partners: KTH and SICS

- WP1 Distributed storage
  - Consistent membership, optimal replication and data placement
- WP2 Computational framework
  - Programming model and execution system for data-intensive applications
- WP3 Information centric networks
  - In-network storage: object caching and lookup; optimization
- WP4 Resource management
  - Distributed and federated clouds
- WP5 Resource allocation
  - Across clouds; extend the Mesos platform with multiple masters
- WP6 Trusted execution
- WP7 Platform Integration

Motivation

- Web 2.0 applications
  - WiKis, social networks, media sharing
- Data-intensive applications

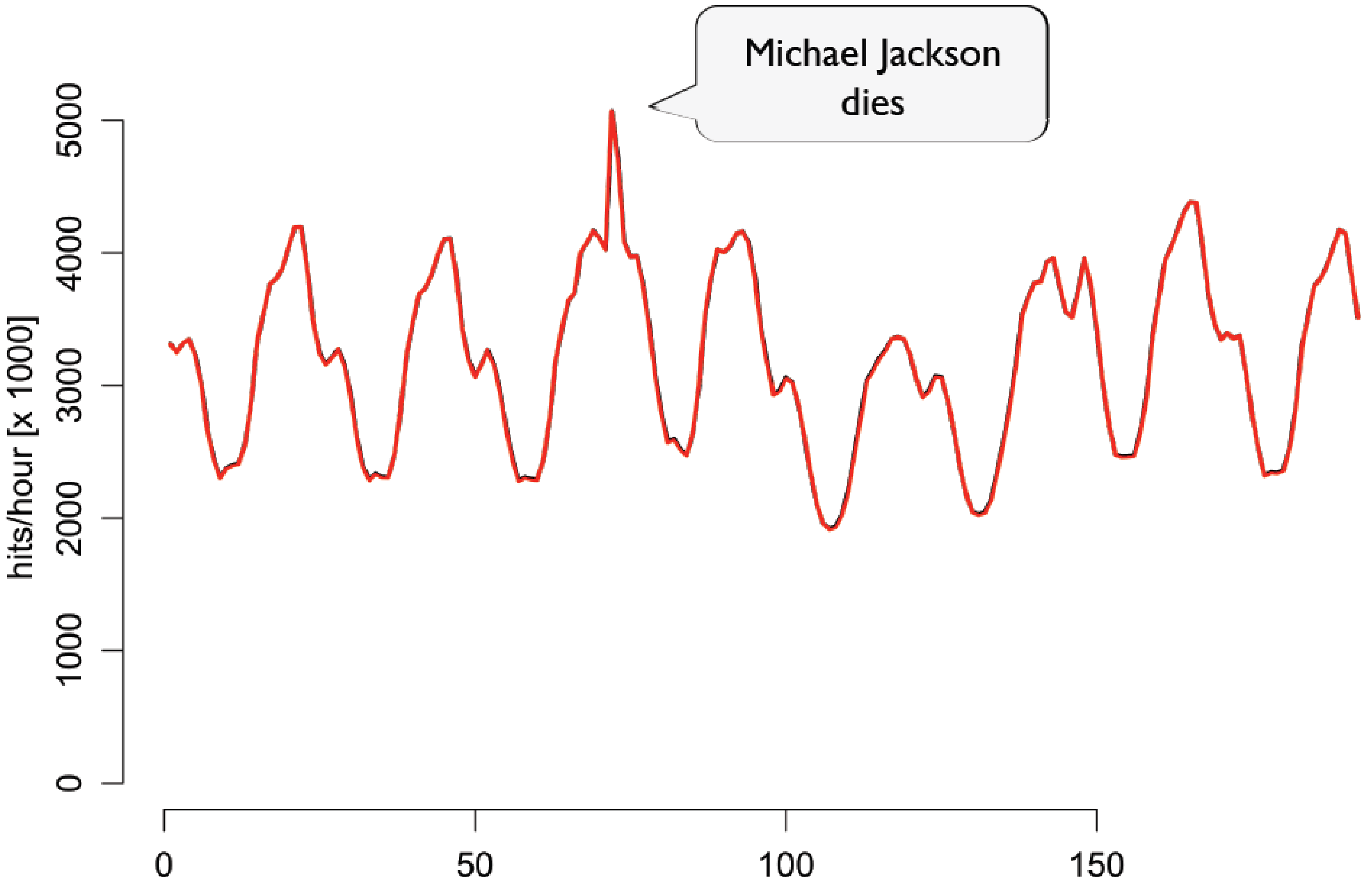
- Challenges
  - Rapidly growing number of users and amount of user-generated data, data-intensive applications (scalability, elasticity)
  - Uneven load, user geographically scattered (low request latency, load balancing)
  - Partial failures, very high load, load spikes (high availability)
  - Acceptable data consistency guarantees (e.g., eventual consistency)



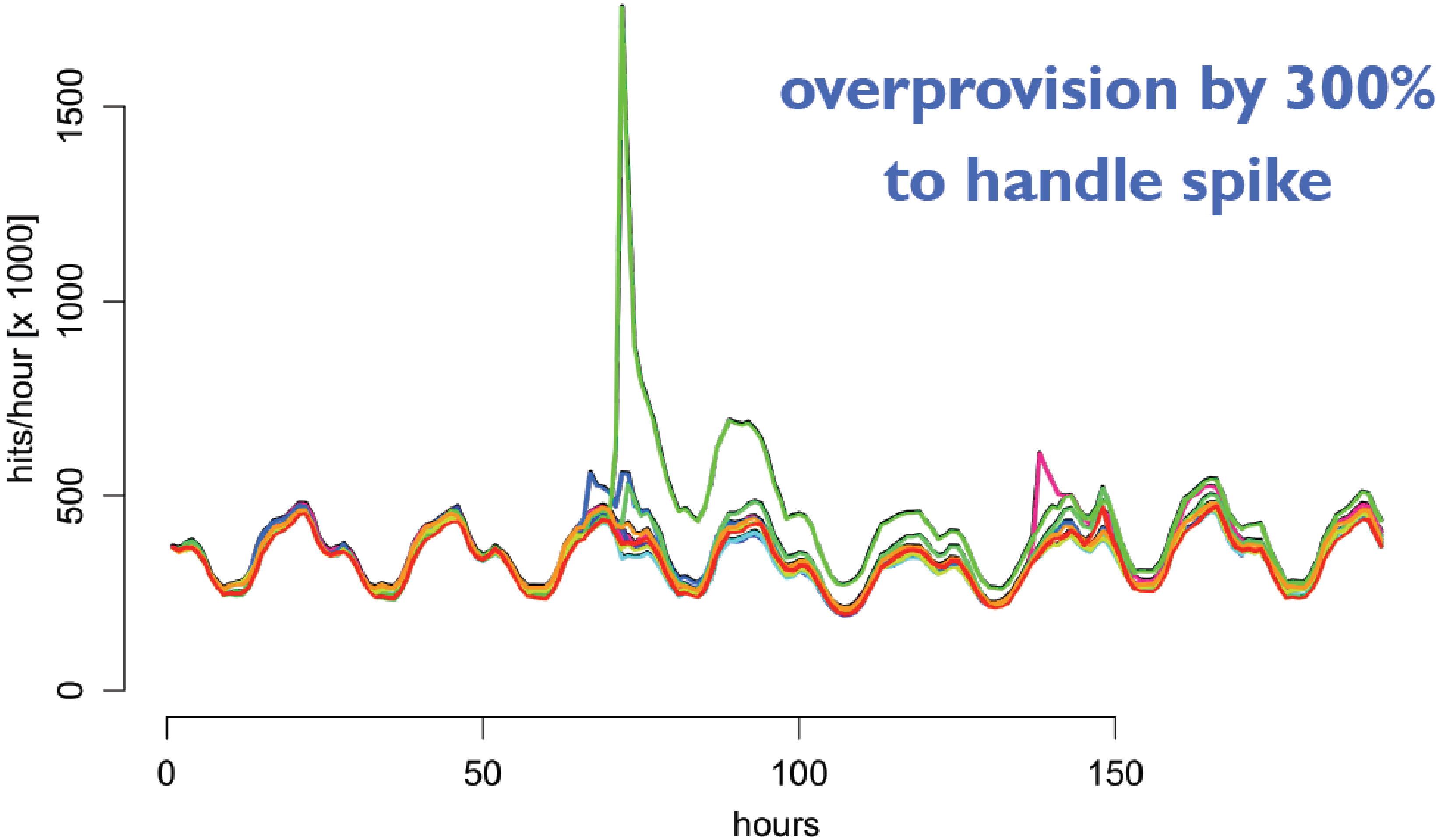
Elasticity

- Web services, e.g. storage, frequently experience high workloads
  - A service can become popular in just an hour
  - A compute job may require a large amount of resources for a very short time
- The high-level load does not last for long and keeping resources in the Cloud costs money
- This has led to **Elastic Computing**
  - Ability of a system to grow and shrink at run-time in response to changes in workload
- **Cloud computing** allows on-the-fly requesting and releasing VM instances to scale the service in order to meet SLOs at a minimal cost
  - Provides an illusion of an infinite amount of resources

Wikipedia Workload Trace - June 2009 [Tru2011, presentation]

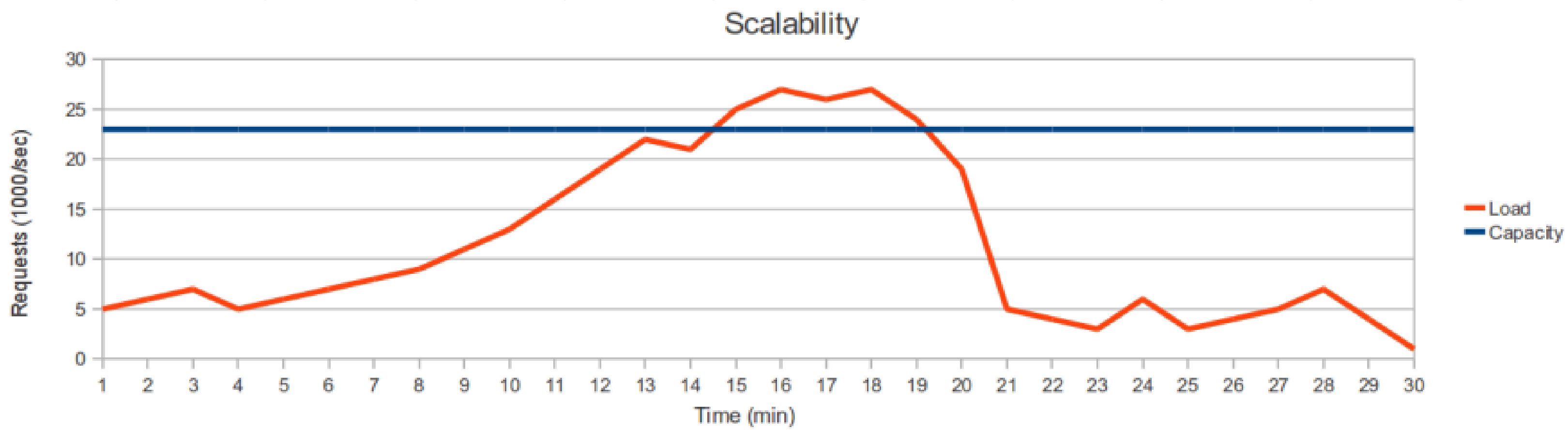


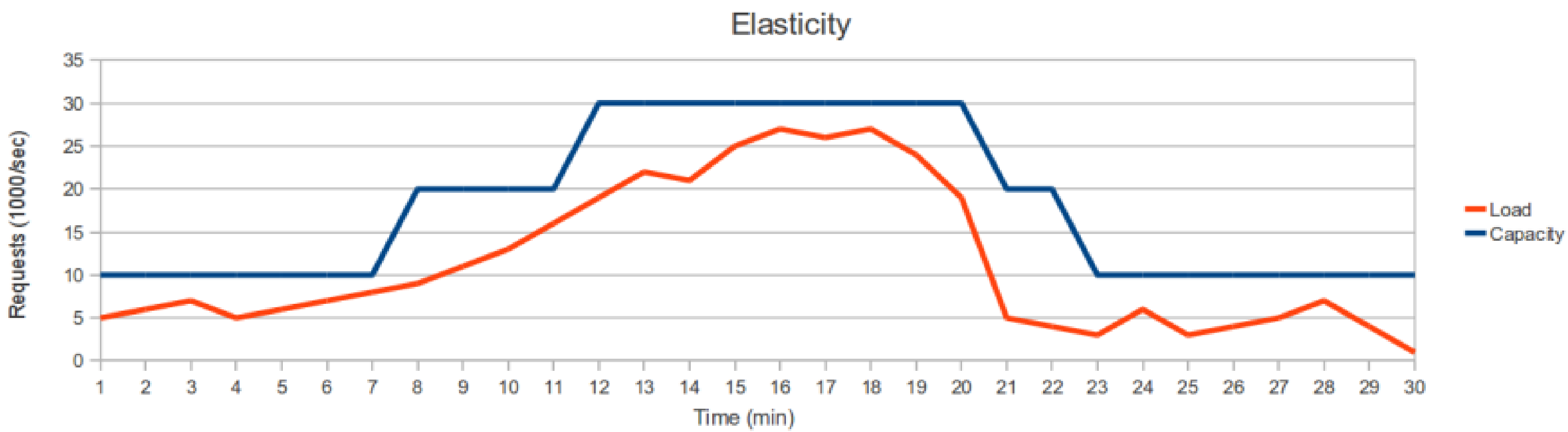
Over-Provisioning Storage System [Tru2011, presentation]



(assuming data stored on ten servers)

Static versus Dynamic Provisioning (Elasticity)





Resource Allocation Across Multiple Cloud Data-Centers [Mal2010]

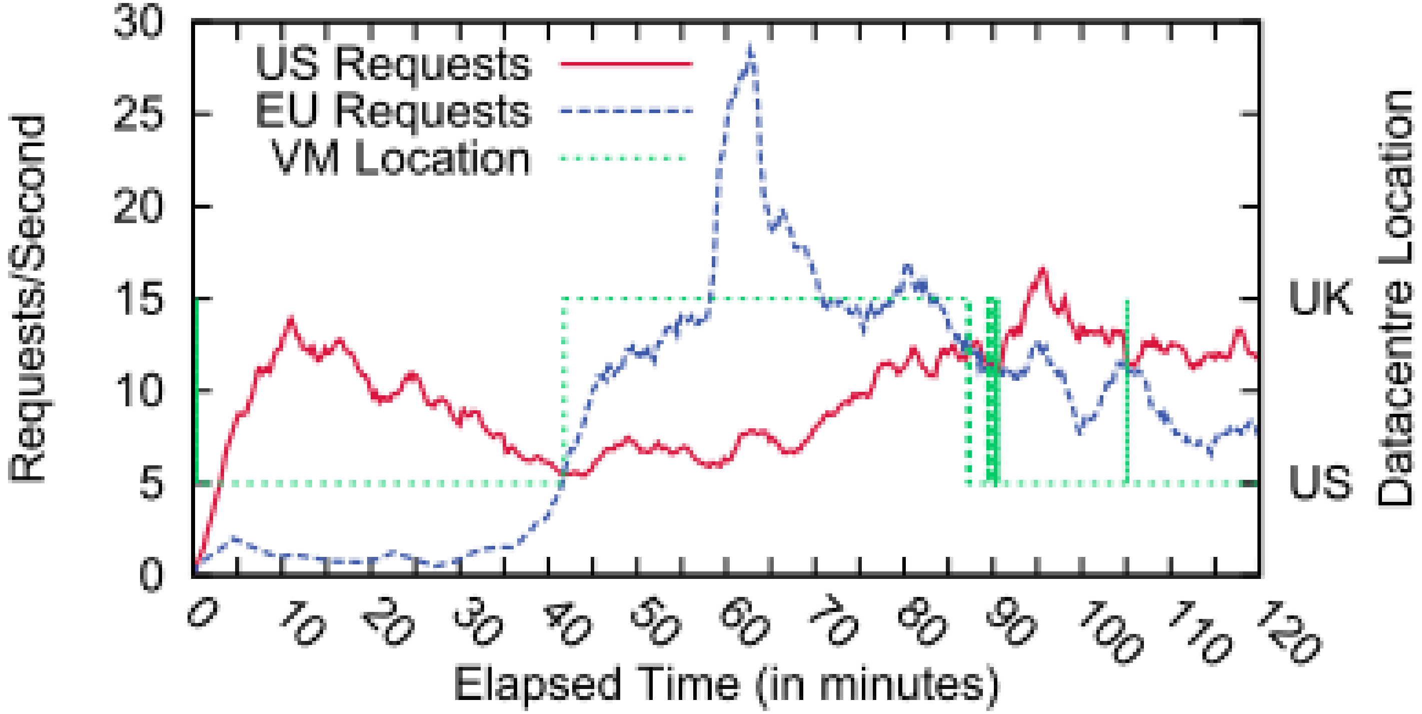


Figure 2: Observed client request rate, split into US and EU requests.

ROYAL INSTITUTE OF TECHNOLOGY

Automation of Elasticity

- Elasticity can be done either manually (by the syst-admin) or automatically (by an autonomic manager)

- **Elasticity Controller**
  - Helps to avoid SLO violations while keeping the cost low
  - Adds/removes VMs (servers, service instances) in response to changes in SLO metrics (e.g., request latency) caused by changes in workload
  - Can be built using elements of Control Theory
    - Feedback-loop (a.k.a. closed-loop) control
    - Model Predictive Control (MPC)

Elastic Storage

- Storage systems specially designed for horizontal scalability
  - Key-value stores
  - minimum functionality: `get(key)` and `put(key, value)`
- Examples
  - Yahoo! PNUTS
  - Google BigTable
  - LinkedIT Voldemort
  - Cassandra
  - SCADS (UPC)
  - File systems, HDFS

Challenges for Storage Elasticity Control [Lim 2010]

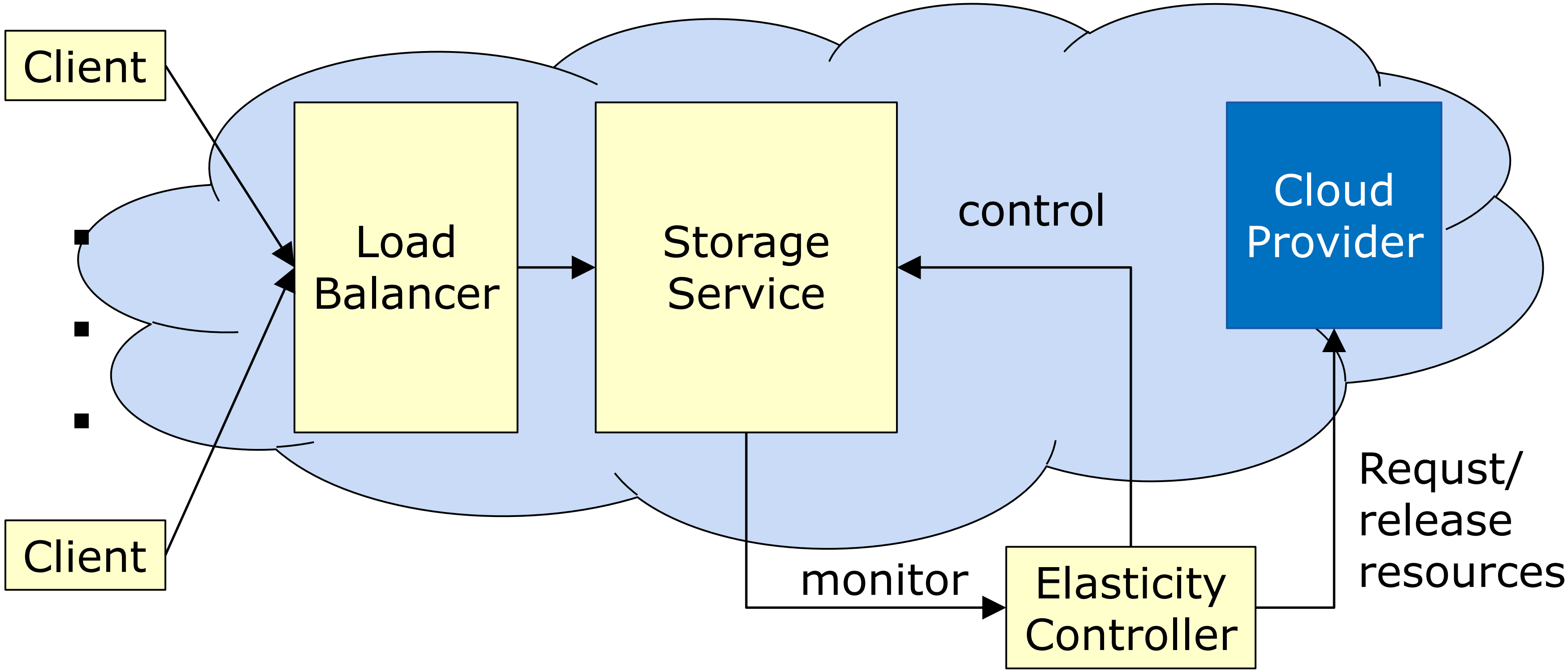
- Clouds present a problem of discrete actuators
  - Resources (VMs) are allocated in discrete units
- Actuator delays (actuation lag) due to rebalancing
  - Storage should redistribute (rebalance) data in response to join and leave events
- Interference with applications and sensor measurements
- The need to synchronize the multiple control elements, including rebalancing
- **The demand may exceed the supply**
  - **This calls for resource allocaton across clouds**

Necessary Conditions (Assumptions) for *Elastic* Storage [Lim2010]

- Horizontal scalability
  - The storage capacity and I/O capacity of the system scales (roughly linearly) with the size of the active server set.
- Load balancing
  - Storage distributes stored data across its servers in a way that balances load effectively;
  - Redistributes (rebalances) data in response to join and leave events
- Replication
  - Storage replicates data for robust availability; to resolve hotspots; the replication is sufficient to avoid service interruptions across a sequence of leave events

An Elastic Cloud-Based Storage with Feedback Elasticity Controller [Mo2012]

- An Elastic storage, e.g. a key-value store, in the Cloud
  - Shrinks/grows in size to meet SLOs at the minimal cost



ROYAL INSTITUTE OF TECHNOLOGY

A Cloud Compute Service

- For Internet services and data-intensive applications
  - Many IaaS offerings, e.g., industrial Amazon EC2, MS Windows Azure, ..., open-source openStack, etc.
- Cluster computing frameworks, e.g.,
  - Hadoop MapReduce
  - Dryad
  - Pregel
  - MPICH2
  - Torque
- Cluster computing frameworks on the Cloud, e.g.,
  - E.g. Hadoop on Amazon EC2 and S3
  - MPI clusters on EC2
  - Mesos on EC2

Challenges for Compute Services

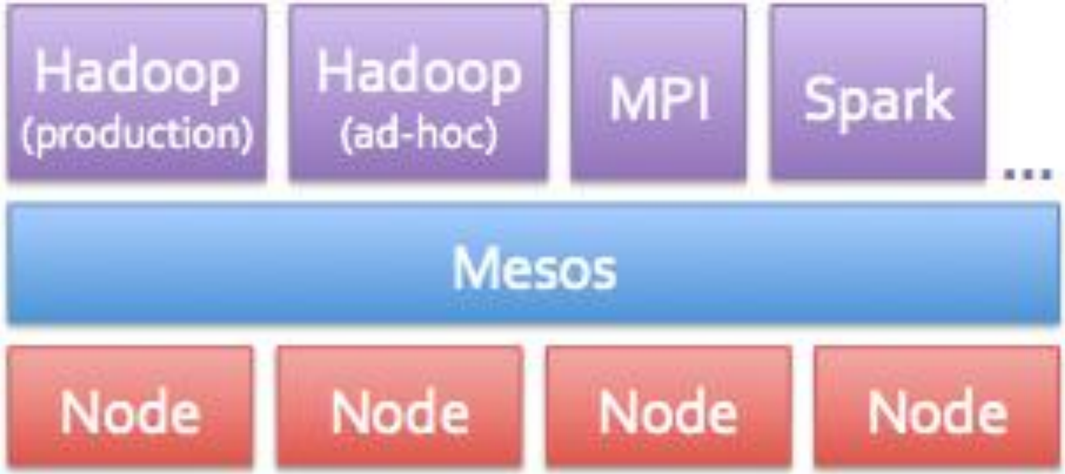
- The mismatch between the allocation granularity of Clouds and the granularity of compute jobs
  - The mismatch between allocation granularities of Clouds and of cluster computing frameworks
  - Course-grained resources for fine-grained jobs/tasks
  - Inefficient resource utilization
  - Inefficient data sharing across jobs, applications, and frameworks
- **The demand may exceed the supply**
  - A job may request a large amount of resources for a very short time

Solutions

  - Multiple frameworks in a single cluster, e.g. Mesos
  - Resource allocation across clouds

Mesos (UC Berkeley) http://www.mesosproject.org/

- A platform for fine-grained resource sharing in the data center
- Resource isolation and sharing across multiple distributed frameworks (applications)
  - *Inter-framework* scheduling (e.g., fair sharing)

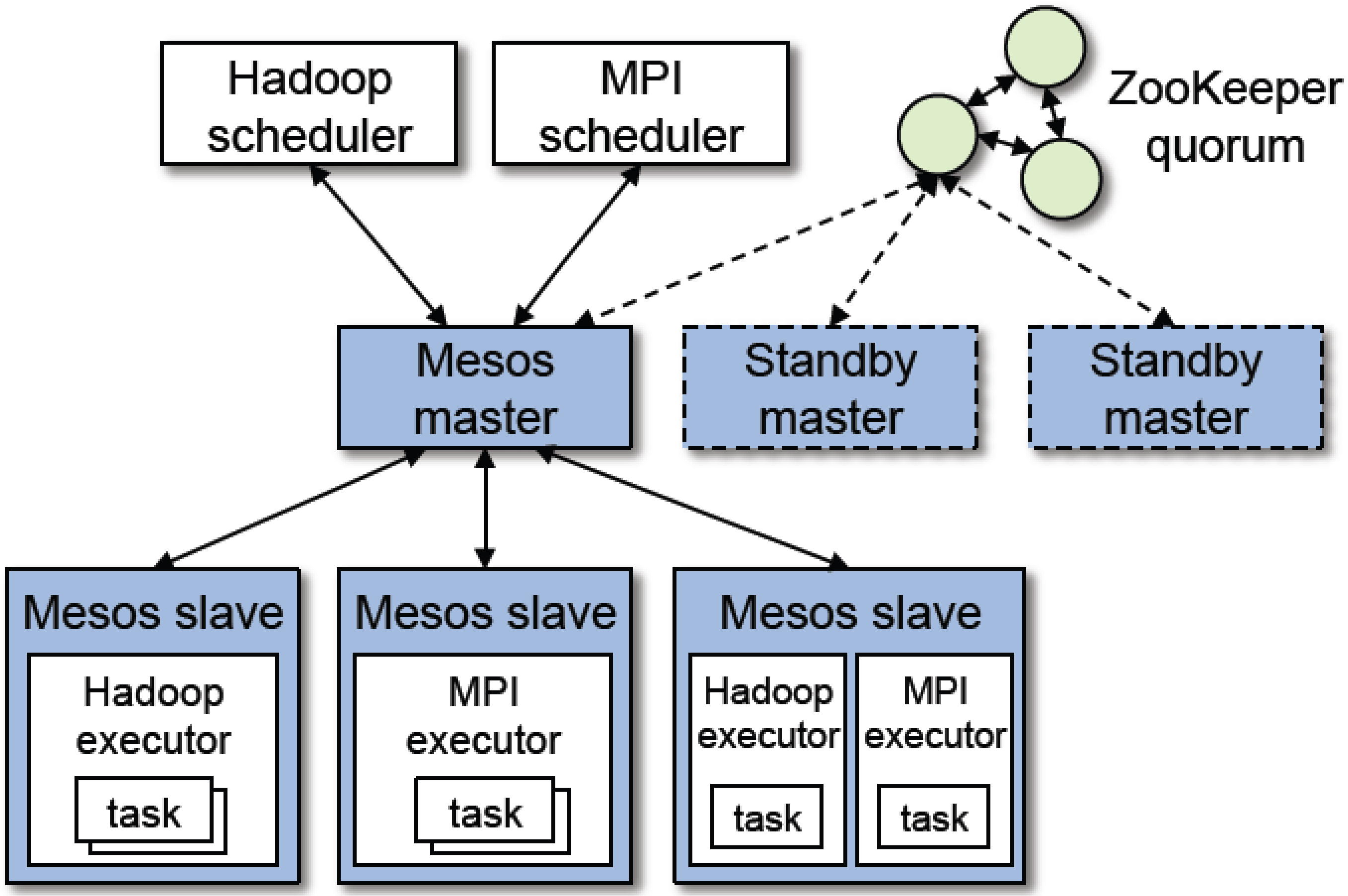


Mesos Design Features [Hin2011]

- **Fine-grained sharing:**
  - Allocation at the level of tasks within a job
  - Improves utilization, latency, and data locality

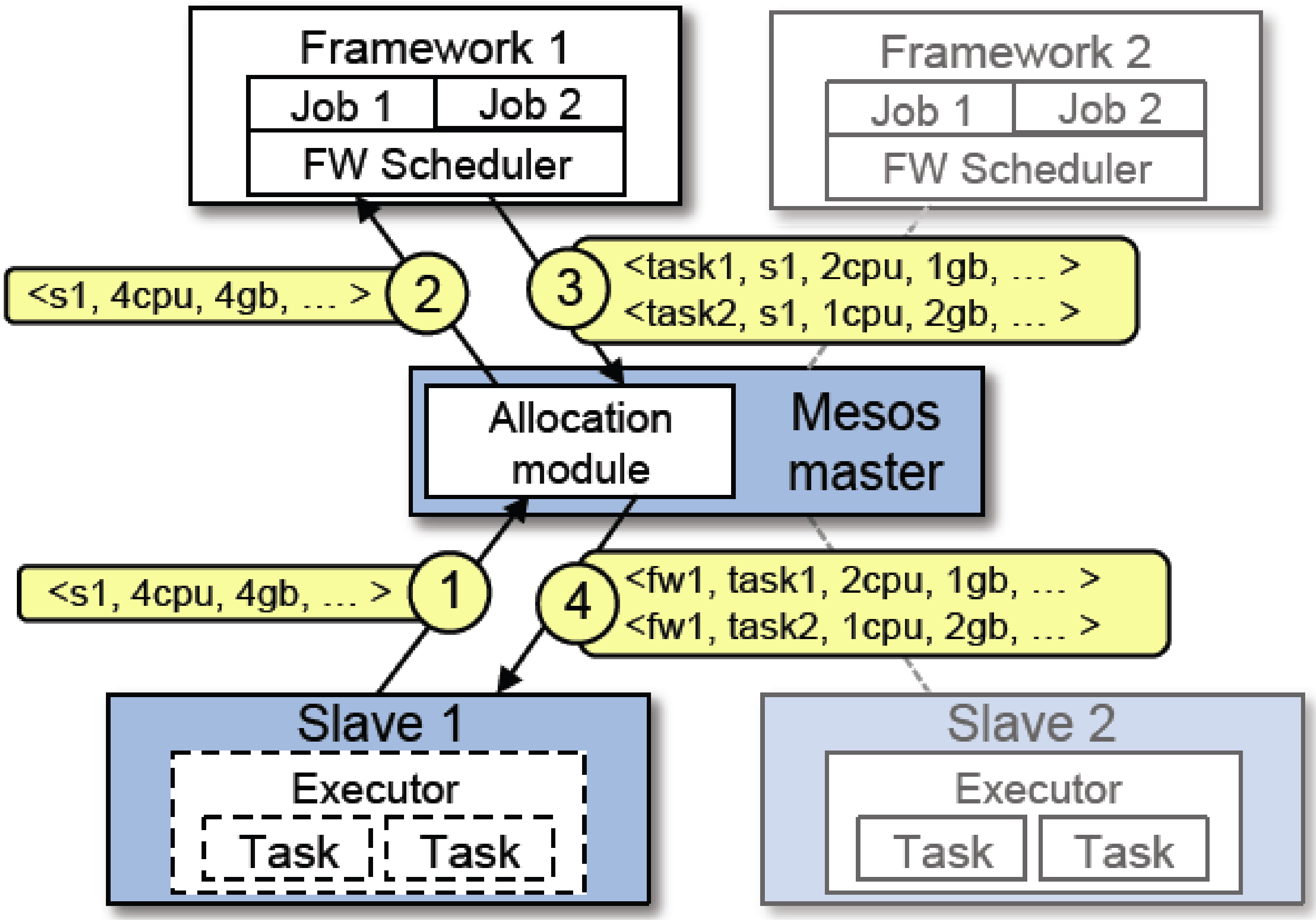
- **Resource offers:**
  - Simple, scalable application-controlled scheduling mechanism

Mesos Architecture [Hin2011]



HDFS is (can be) used as a file system

Resource Offer Example [Hin2011]



KTH VETENSKAP OCH KONST
ROYAL INSTITUTE OF TECHNOLOGY

Mesos Resource Allocation and Isolation

- Resource allocation using **pluggable allocation modules**
  - Fair sharing based on a generalization of max-min fairness for multiple resources [Gho2011]
  - Based on strict priorities.

- Framework isolation using **pluggable isolation modules**
  - Mesos uses OS isolation mechanisms, such as
  - **Linux containers lxc**
  - Solaris projects
  - Containers currently support CPU, memory, IO and network bandwidth isolation

Linux Containers (LXC) [http://lxc.sourceforge.net/]

- LXC is an OS-level virtualization method for running multiple isolated systems in user-space containers on a single host.
  - LXC does not provide a VM, but rather a lightweight virtual environment that has its own process and network space
  - "chroot on steroids"

- Container is a lightweight virtual system with full resource isolation and resource control for an application or a system
  - Linux Containers isolate process groups from each other in the kernel

Linux Containers and the Cloud

- Linux Containers as a hypervisor
  - openStack supports using Linux Containers (through libvirt) to run Linux-based virtual machines

- Linux Containers in the Cloud (on VMs)
  - E.g. One can create Linux containers on VMs in the Amazon EC2 cloud

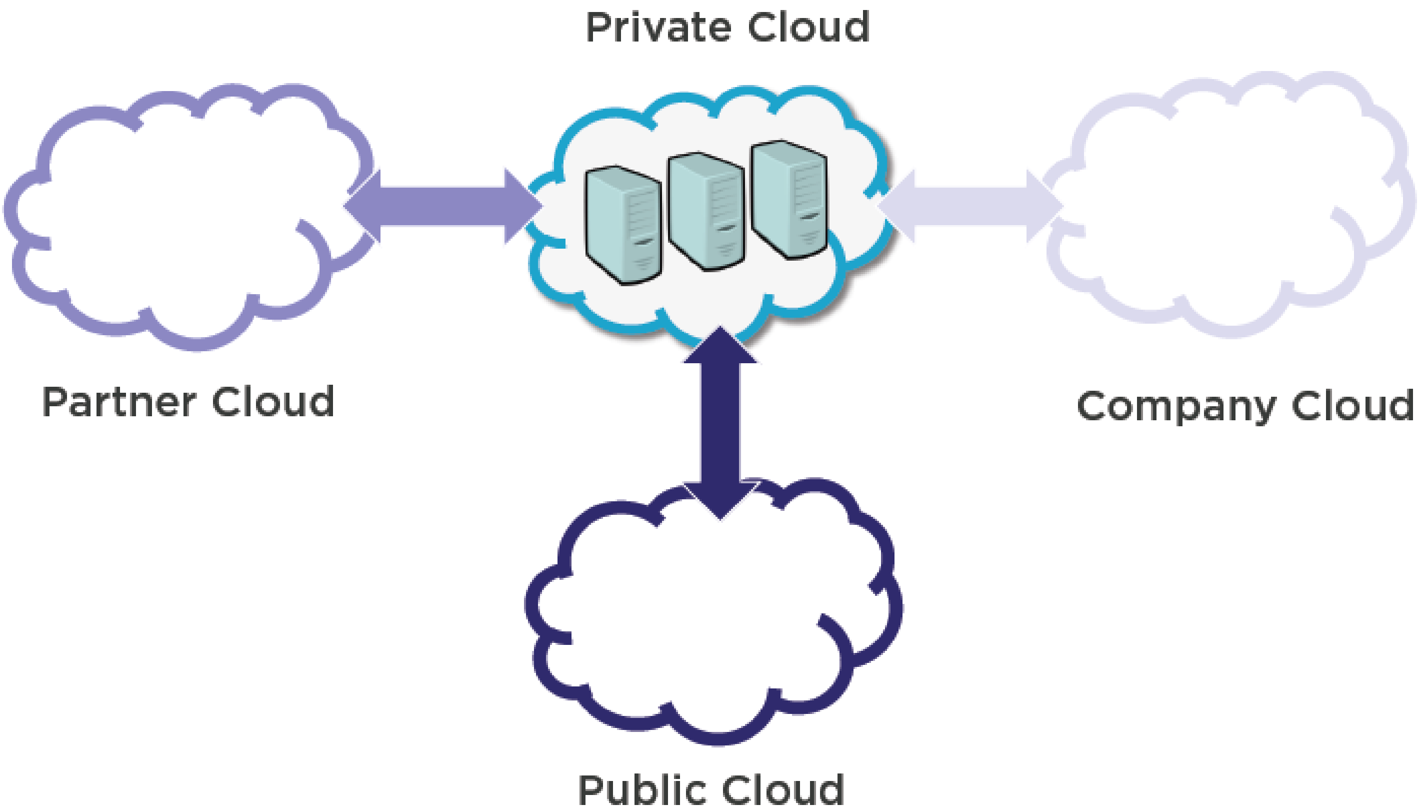
- **Mesos can run in the EC2 Cloud**

Benefits of Cloud Federations [Llo2011]

- Scalability
  - Cloudbursting to address peak demands
- Collaboration
  - Sharing of infrastructure between partners
- Multi-site Deployments
  - Infrastructure aggregation across distributed data centers
- Reliability
  - Fault tolerance architectures across sites
- Performance
  - Deployment of services closer to end users
- Cost
  - Dynamic placement to reduce the overall infrastructure cost
- Energy Consumption
  - Minimize energy consumption

Federation Levels [Llo2011]

- Different Levels of Control, Monitoring, Cross-site Functionality and Security



Our Cloud Environment

- SICS cluster (18 Dell PowerEdge servers)
  - Each with 2 x 6-core AMD Opteron 2435 processors (12 cores in total)
  - 32 GB of RAM & 1 TB Storage / server
  - Ubuntu 10.10/11.04
- KTH-ICT cluster (4 HP ProLiant DL380 servers)
  - Each with 2 x 6-core x 2-way hyper-threading Intel Xeon X5660 processors (24 cores in total)
  - 44 GB of RAM & 2 TB Storage / server
  - Red Hat Enterprise Linux 6
- KTH-EES cluster (9 Dell PowerEdge server)
- To be federated

Our Cloud Environment

- OpenStack (Diablo release)
  - IaaS cloud computing
  - by Rackspace Cloud and NASA
  - free open source software (Apache license)

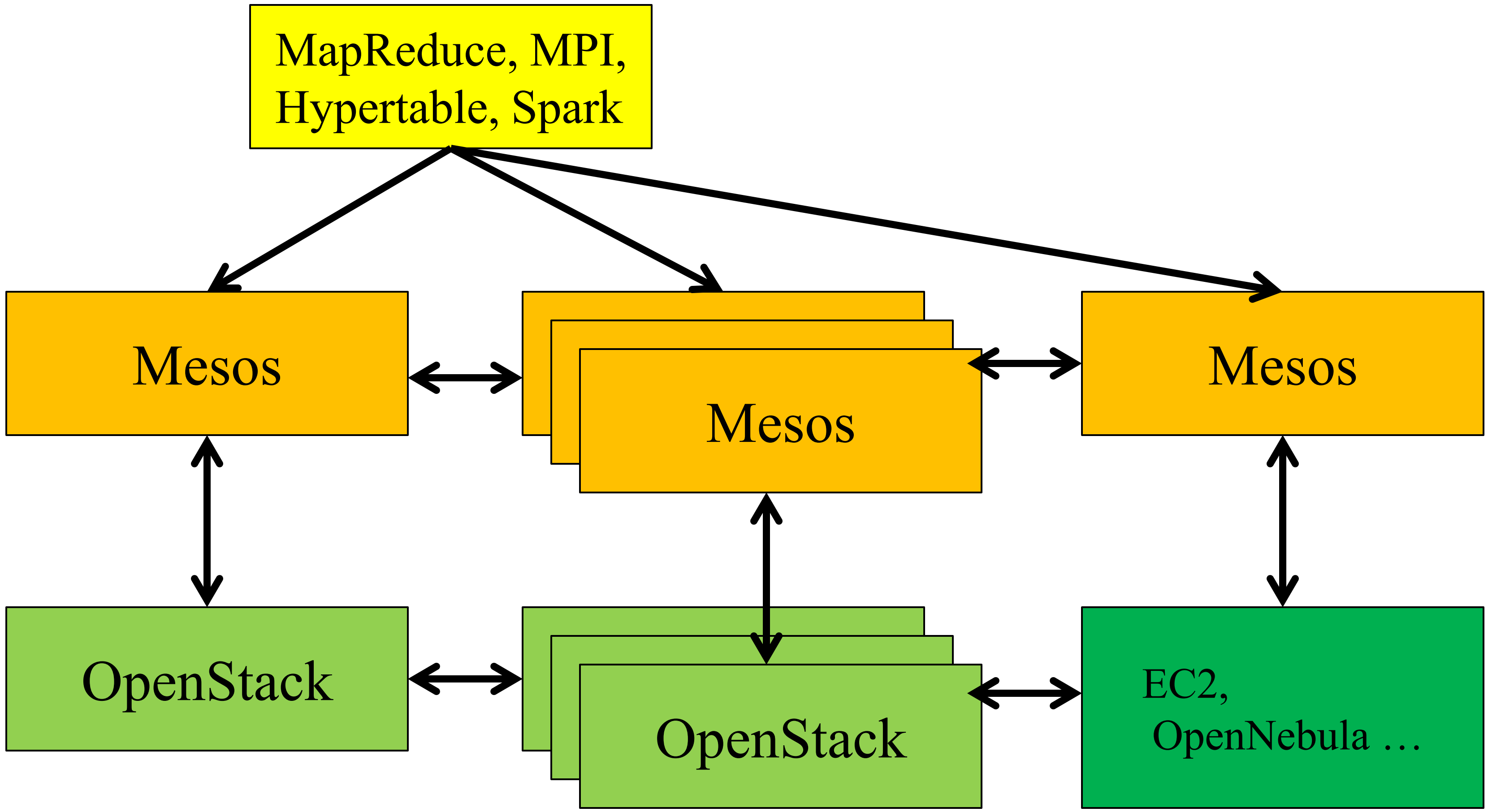
- Why OpenStack
  - Industry Standard
    - Possible cooperation with industry
  - Open source and flexible
  - EC2 support

ROYAL INSTITUTE OF TECHNOLOGY

RMAC Task 4

- **Goal**: Enable fine-grain elastic resource allocation across clouds by providing Mesos in an openStack Cloud and Cloud federation
  - Build a Cloud federation
  - Run Mesos in the openStack Cloud
  - Provide a federation of Mesos masters in the Cloud federation
    - Extend Mesos with multiple masters
    - each controlling and accounting for the resources of one cluster in the Cloud, and holding snapshots of each other’s state;
    - Masters should cooperate and coordinate their work
  - Should allow compute frameworks to schedule jobs according to their preferences with respect to different clouds, as the geographical location of the clouds
    - This will impact jobs in terms of bandwidth and latency.

Research Workflow



References

- **[Mal2010]** B. Malet and P. Pietzuch. *Resource allocation across multiple cloud data centres.* MGC '10
- **[Lim2010]** Harold C. Lim, Shivnath Babu, and Jeffrey S. Chase, *Automated control for elastic storage*. ICAC '10
- **[Mou2012]** M. A. Moulavi, A. Al-Shishtawy, and V. Vlassov, *State-Space Feedback Control for Elastic Distributed Storage in a Cloud Environment*, ICAS’12, to appear
- **[Tru2011]** Beth Trushkowsky, et al., *The SCADS director: scaling a distributed storage system under stringent performance requirements*. FAST'11
- **[Hin2011]** Benjamin Hindman, , et al., *Mesos: a platform for fine-grained resource sharing in the data center*. NSDI'11
- **[Gho2011]** A. Ghodsi, et al. *Dominant resource fairness: fair allocation of multiple resource types*. NSDI'11
- **[Llo2011]** I. M. Llorente, *Challenges in Hybrid and Federated Cloud Computing*, presentation at the Cloud Day 2011, Stockholm, Sweden, http://opennebula.org/_media/community:challenges_in_hybrid_and_federated_cloud_computing_-_cloudday2011.pdf

Backup Slides

Cloud-Based Services

- ***Cloud computing*** offers an efficient and effective solution to the challenges of scale and the (highly) dynamic load
- Provides the illusion of infinite amount of resources
- "Pay-as-you-go": pay for a service only when/if you use it
- End-user does not need to be involve in the configuration and maintenance of the cloud-based system
- Enables development of ***Cloud-based Elastic Services and Applications***

Elasticity

- According to The Free Dictionary
  In Physics, **Elasticity** is “*the property of a body or substance that enables it to resume its original shape or size when a distorting force is removed*”

- According to Reuven Cohen, Enomaly Inc.
  Elasticity is “*The quantifiable ability to manage, measure, predict and adapt responsiveness of an application based on real time demands placed on an infrastructure using a combination of local and remote computing resources.*”

- **Elasticity** in Cloud computing is an ability of a system to scale up and down (grow and shrink by requesting and releasing resources) in response to changes in its environment and workload

Example: An Elasticlity Controller for the Voldemort k-v Store