ROYAL INSTITUTE OF TECHNOLOGY

Autonomic Control for Cloud-Based Services and Applications

Vladimir Vlassov vladv@kth.se

KTH Royal Institute of Technology
Stockholm, Sweden

ICAS 2012 Panel
Sint Maarten, Netherlands Antilles
28 March 2012

ROYAL INSTITUTE OF TECHNOLOGY

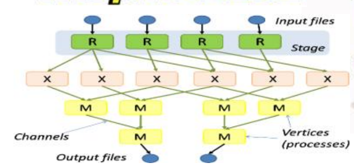
Motivation

- Web 2.0 applications
  - WiKis, social networks, media distribution and sharing
- Data-intensive applications; big data

- 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)

Pregel

Pig



Dryad

Cloud-Based Services and Applications

- **Clouds** provide the illusion of the infinite amount of resources
  - "Pay-as-you-go"
  - End-users are not involve in configuration & maintenance
- Enables **Cloud-based *Elastic* Services and Applications**
  - Storage and Compute services
  - Cloud-enhanced CDNs and P2P
  - Mobile apps; enterprise and scientific applications
  - Data-intensive apps, e.g., real-time audio language translation, expert guidance
  - On-line games
  - Etc.
- Clouds make possible **Resource *Elastic* Applications**

Elasticity

ROYAL INSTITUTE OF TECHNOLOGY

- 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*" [The Free Dictionary]

- In Cloud computing, **Elasticity** is the ability to scale resource usage up and down according to demand
  - The ability of a system to scale up and down (grow and shrink by requesting and releasing resources) in response to changes in its environment, workload, and QoS requirements

ROYAL INSTITUTE OF TECHNOLOGY

Automation of Elasticity (1/2)

Goal: Make effective and efficient use of elasticity in order to improve user-experience with cloud-based applications and services, and to build new applications and services

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
- Built using **Control Theory** [Hel2004], stochastic modeling (queuing theory), machine learning and distributed optimization techniques
  - Classical closed-loop (a.k.a. feedback) control
  - Model Predictive Control (MPC)

Automation of Elasticity (2/2)

- **Research issues**
  - Design methodology: design space, steps, algorithms, guidelines, APIs and tools for all stages of the MAPE loop
  - Touch-points: sensors (monitoring) and actuators (scale up/down, rebalance, etc.), SLO/QoS metrics, APIs
  - Management logic: partitioning, distribution, coordination, algorithms
  - System models

References

- **[Hel2004]** J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback Control of Computing Systems. Wiley-IEEE Press, 2004.*
- **[Lim 2010]** Harold C. Lim, Shivnath Babu, and Jeffrey S. Chase, *Automated control for elastic storage*. ICAC '10, doi=10.1145/1809049.1809051
- **[Tru2011]** Beth Trushkowsky, et al., *The SCADS director: scaling a distributed storage system under stringent performance requirements*. In the 9th USENIX Conf on File and Storage Technologies (FAST'11), 2011
- **[Mou2012]** M. A. Moulavi, A. Al-Shishtawy, and V. Vlassov, *State-Space Feedback Control for Elastic Distributed Storage in a Cloud Environment*, ICAS 2012
- **[Mon2011]** A. Montresor, and L. Abeni, *Cloudy Weather for P2P, with a Chance of Gossip*, IEEE P2P 2011 (best paper award)
- **[Hin2011]** B. Hindman, , et al., *Mesos: a platform for fine-grained resource sharing in the data center*. NSDI'11

Backup slides

ROYAL INSTITUTE
OF TECHNOLOGY

Storage Services

- Storage systems designed for horizontal scalability, such as key-value stores
  - minimum functionality: `get(key)` and `put(key, value)`
  - horizontal scalability, load balancing and replication

- Examples
  - Yahoo! PNUTS
  - Google BigTable
  - LinkedIn Voldemort
  - Apache Cassandra
  - UCB's SCADS
  - File systems, e.g., Hadoop Distributed File System

Challenges of Elasticity Control for a Cloud-Based Storage

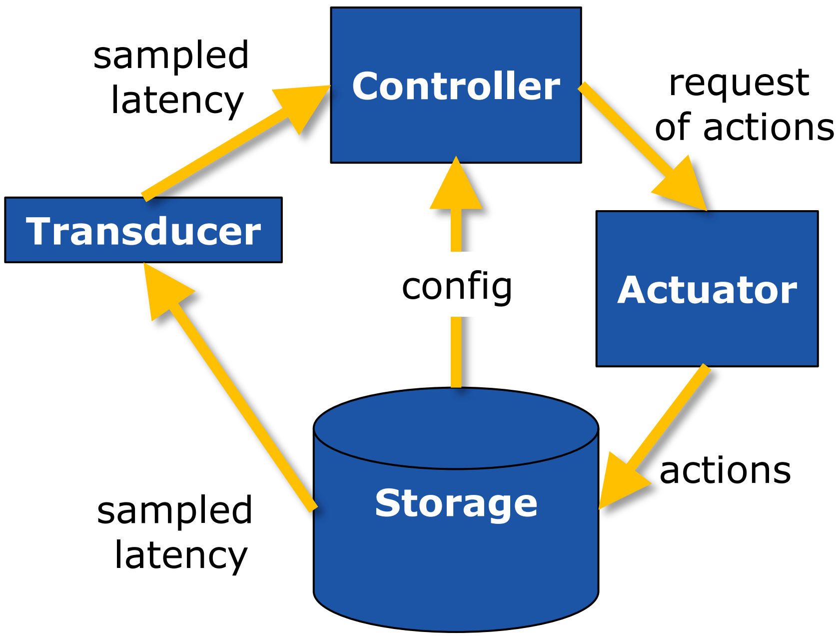
[Lim 2010]

- Clouds present a problem of discrete actuators
- Actuator delays (actuation lag) due to redistribution/rebalancing of data in response to joins and leaves events
- Interference with applications and sensor measurements
- The need to coordinate the multiple control elements

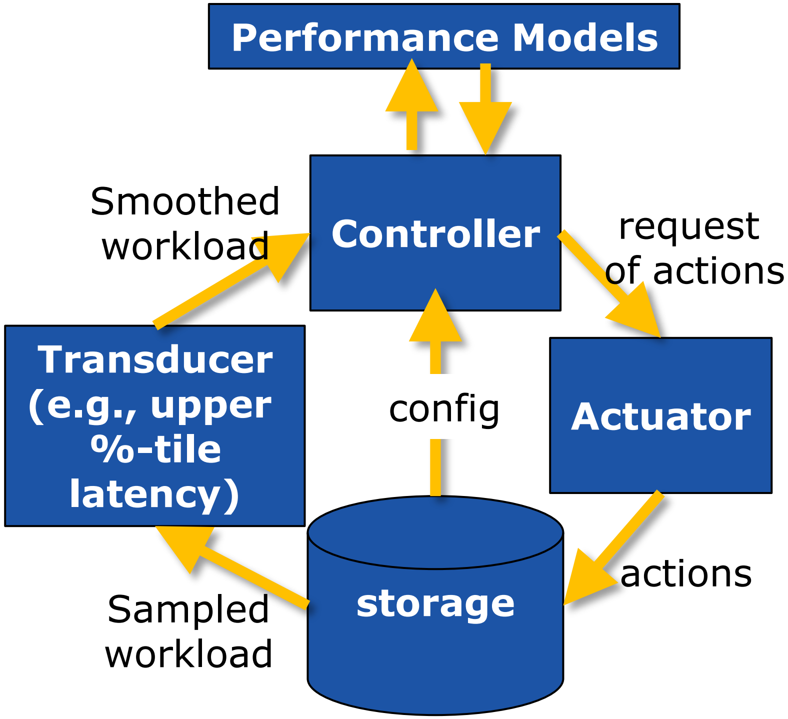
- In addition to above
- The demand may exceed the supply (capacity)
  - Lost revenue and lost users
  - Solution: resource allocation across clouds, cloud federations
- Large scale of a storage
  - Centralized vs decentralized control
- Geographically scattered users and data

Approaches to Automated Elasticity Control for a Cloud-Based Storage

- Closed-loop control
  e.g., [Lim2010][Mol2012]

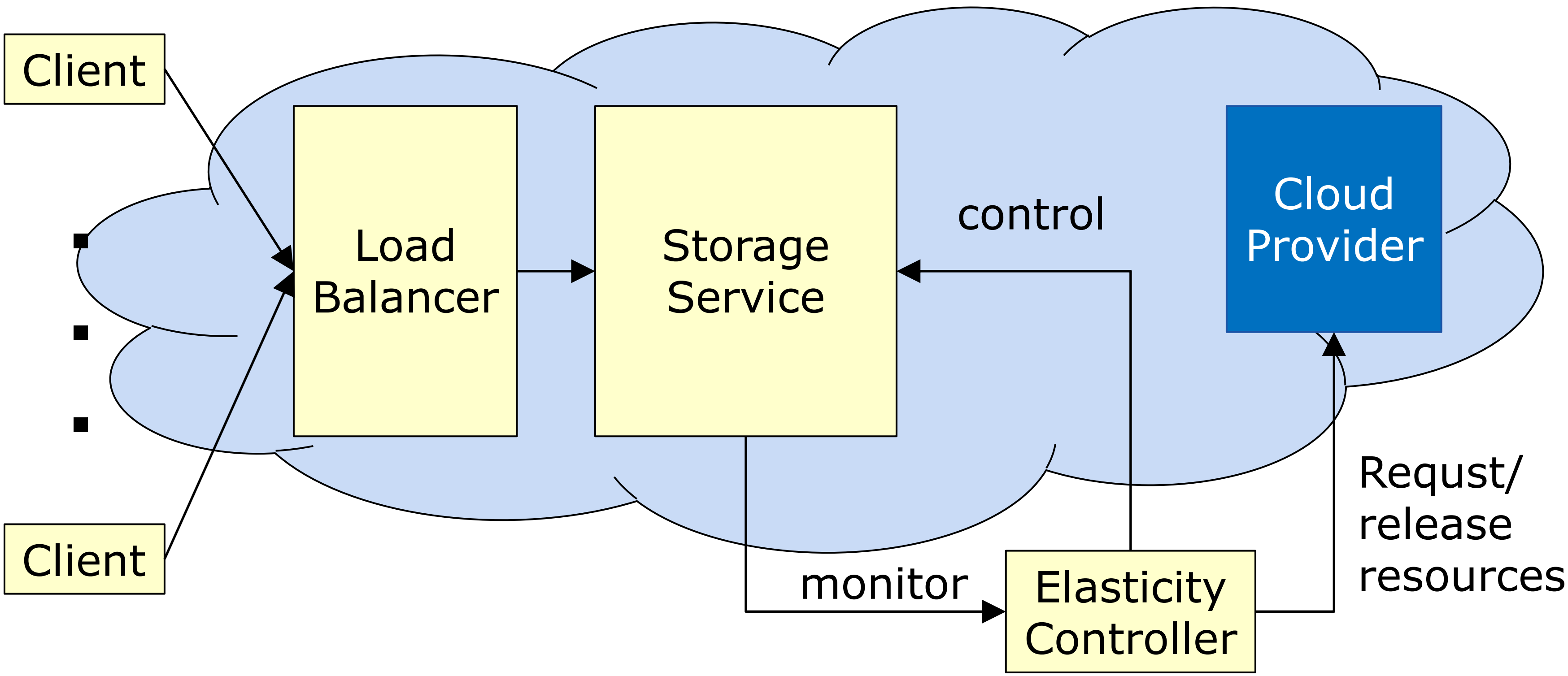


- Model Preditive Control
  e.g., [Tru2011]

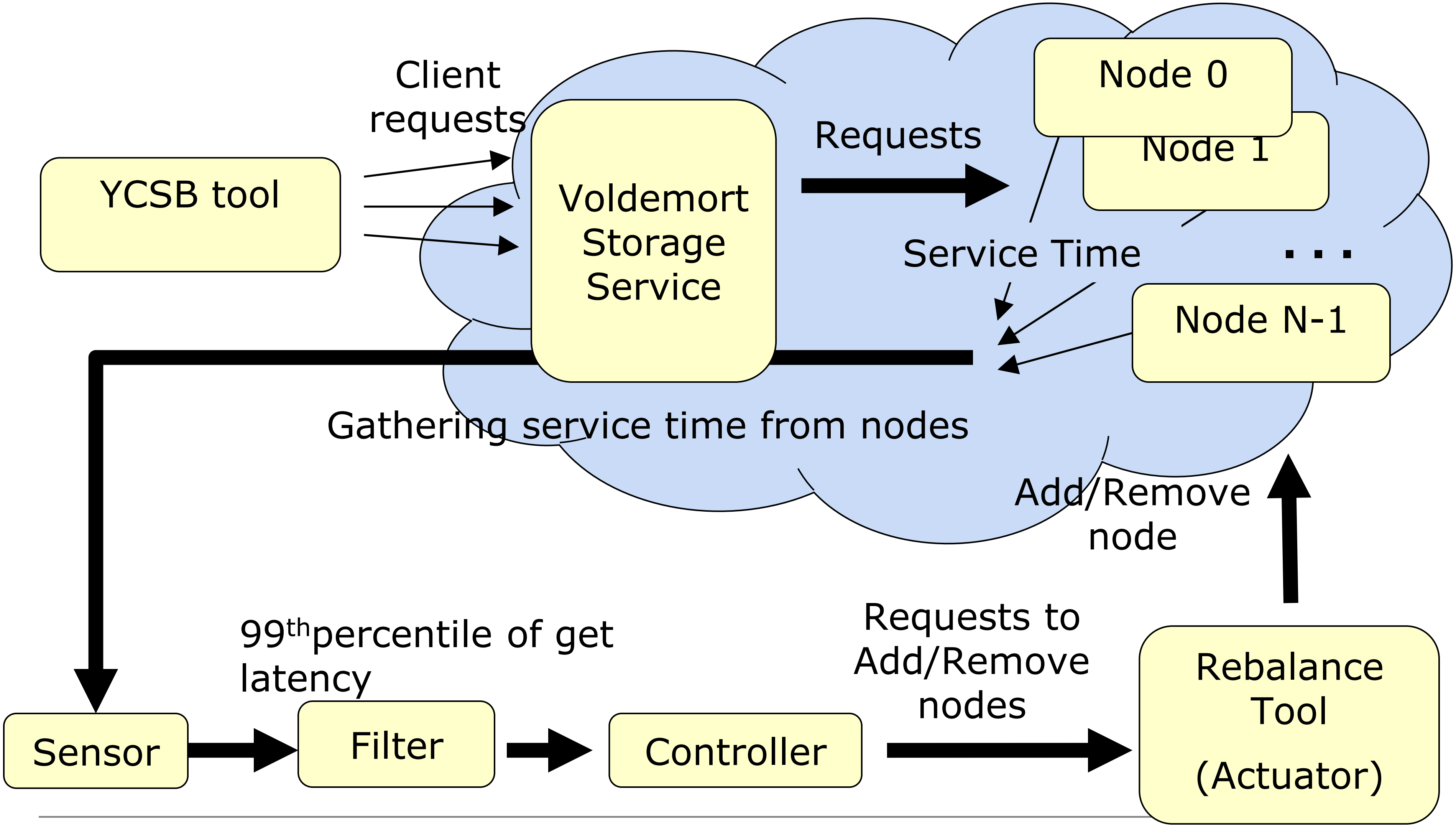


Example 1: An Elastic Cloud-Based Storage with Feedback Elasticity Controller [Mo2012]

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



Example 2: An Elasticlity Controller for the LInkedIn Voldemort k-v Store



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 Automation of Compute Services

- The **mismatch** between the allocation **granularity** of Clouds (VMs) and of cluster computing frameworks (jobs/tasks)
  - Inefficient resource utilization
  - Inefficient data sharing across jobs, applications, and frameworks
- Jobs with timing constrains require elasticity

- **The demand may exceed the supply**

Solutions

- Multiple frameworks in a single cluster with fair sharing, e.g., LXC-based Mesos [Hin2011]
- Resource allocation across clouds; Cloud federations

P2P and Clouds

- P2P: Free, self-organizing, but unreliable
- Clouds: Cost money, but reliable
- P2P systems enhanced with **Cloud helpers**
  - Compute Cloud: active (push)
  - Storage Cloud: passive (pull)
- Example: P2P CDN supported by Cloud helpers [Mon2011]

Challenges

- QoS requirements include timing constraints (e.g., live streaming)
- Requires model predictive control for optimal use of Cloud helpers
- Model of a P2P CDN integrated with cloud helpers
- All related issues: touch-points (monitoring, actuation), etc.