Elasticity Manager for Elastic Key-Value Stores in the Cloud

ROYAL INSTITUTE OF TECHNOLOGY

Vladimir Vlassov vladv@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

Joint work with: Ahmad Al-Shishtawy, SICS

Cloud Control Workshop, Lund University, 7-9 May 2014

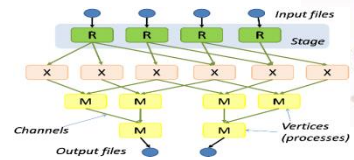
Motivation

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

- Challenges
  - scalability, elasticity
    - Rapidly growing number of users and amount of user-generated data, data-intensive applications
  - load balancing, latency
    - Uneven load; users are geographically scattered
  - availability
    - Partial failures, very high load, load spikes
  - consistency guarantees
    - Data-centric applications and services

Pregel

Pig



Dryad

ROYAL INSTITUTE OF TECHNOLOGY

Cloud-Based Services and Applications

- **Clouds** provide the illusion of the infinite amount of resources
- Pay-as-you-go pricing model
  - High load: Allocate more resources to improve performance
  - Low load: Release resources to save money
- Enables **Cloud-based *Elastic* Services and Applications**

The Need for Elasticity

- Web services and applications frequently experience high workloads
  - A service can become popular in just an hour
- 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 VMs to scale/resize the service in order to meet SLOs at a minimal cost

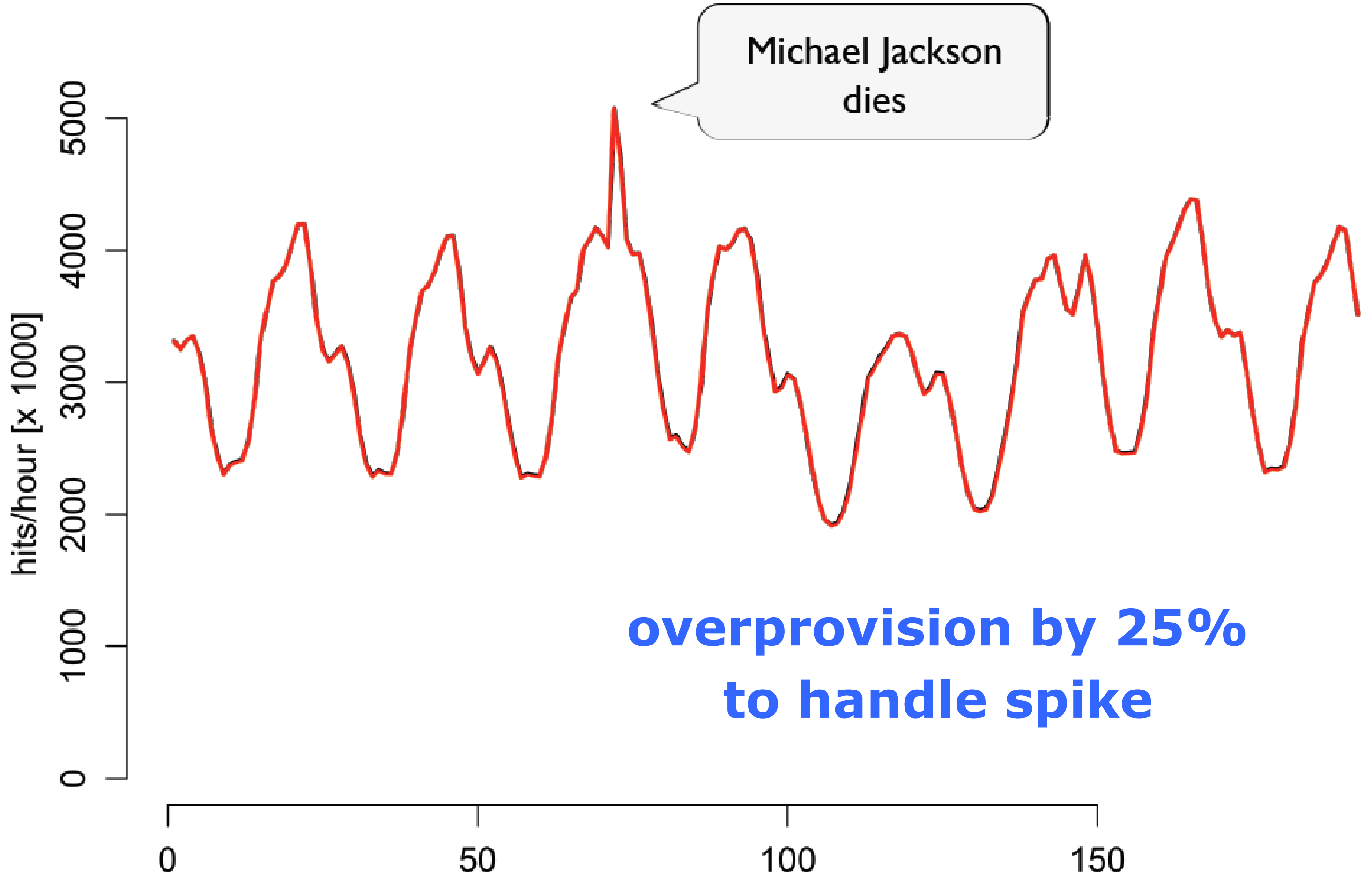
ROYAL INSTITUTE OF TECHNOLOGY

Elasticity

- 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]

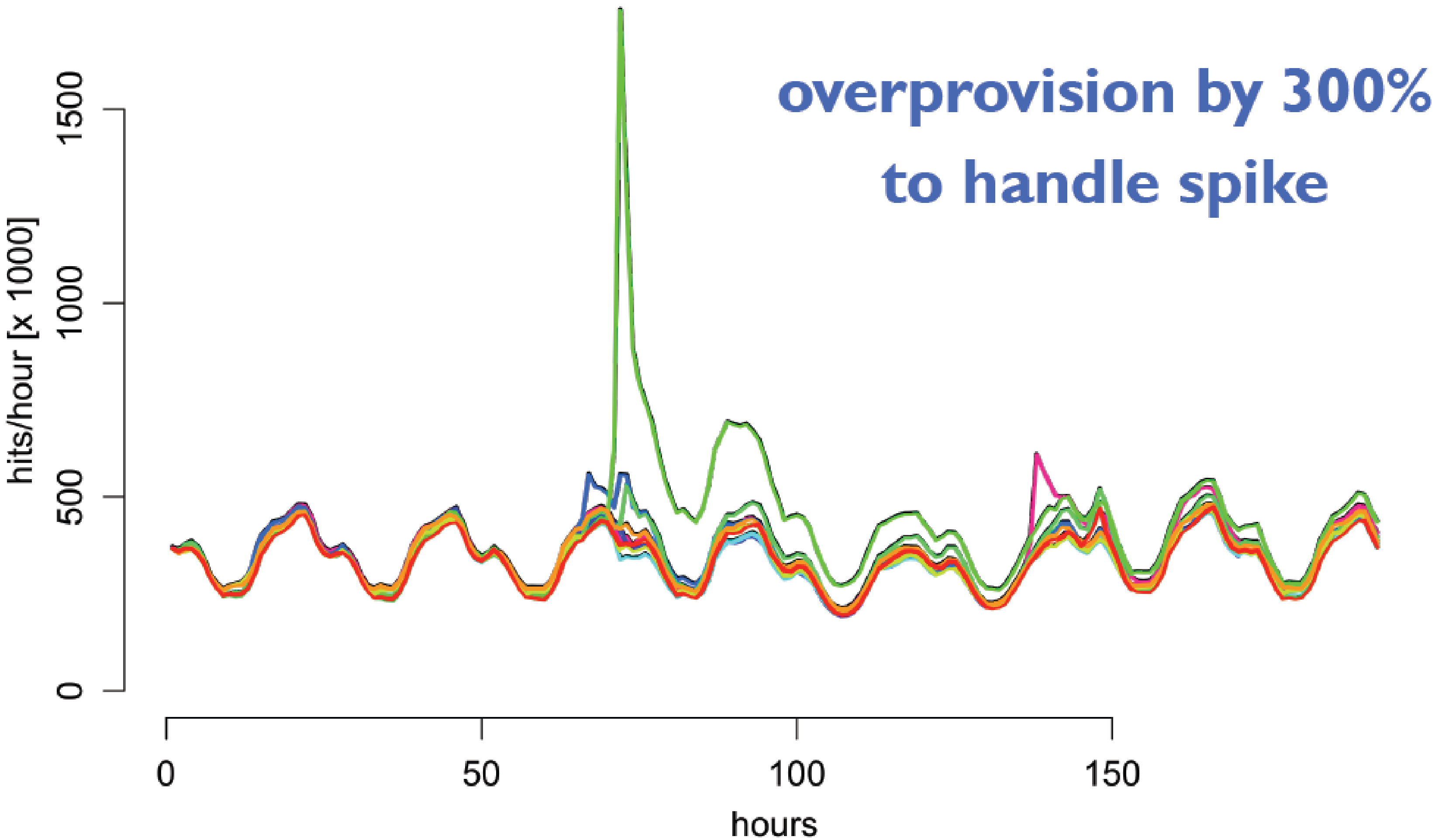
- In Cloud computing, **Elasticity** is the ability to scale resource usage up and down [rapidly] according to [instantaneous] demand
  - The ability of a system to scale up and down (grow and shrink by requesting and releasing resources) in response to changes in its environment, workload, and QoS requirements

Wikipedia workload trace - June 2009



[From: "Solving the Scalability Dilemma with Clouds, Crowds, and Algorithms", Michael Franklin, UC Berkeley]

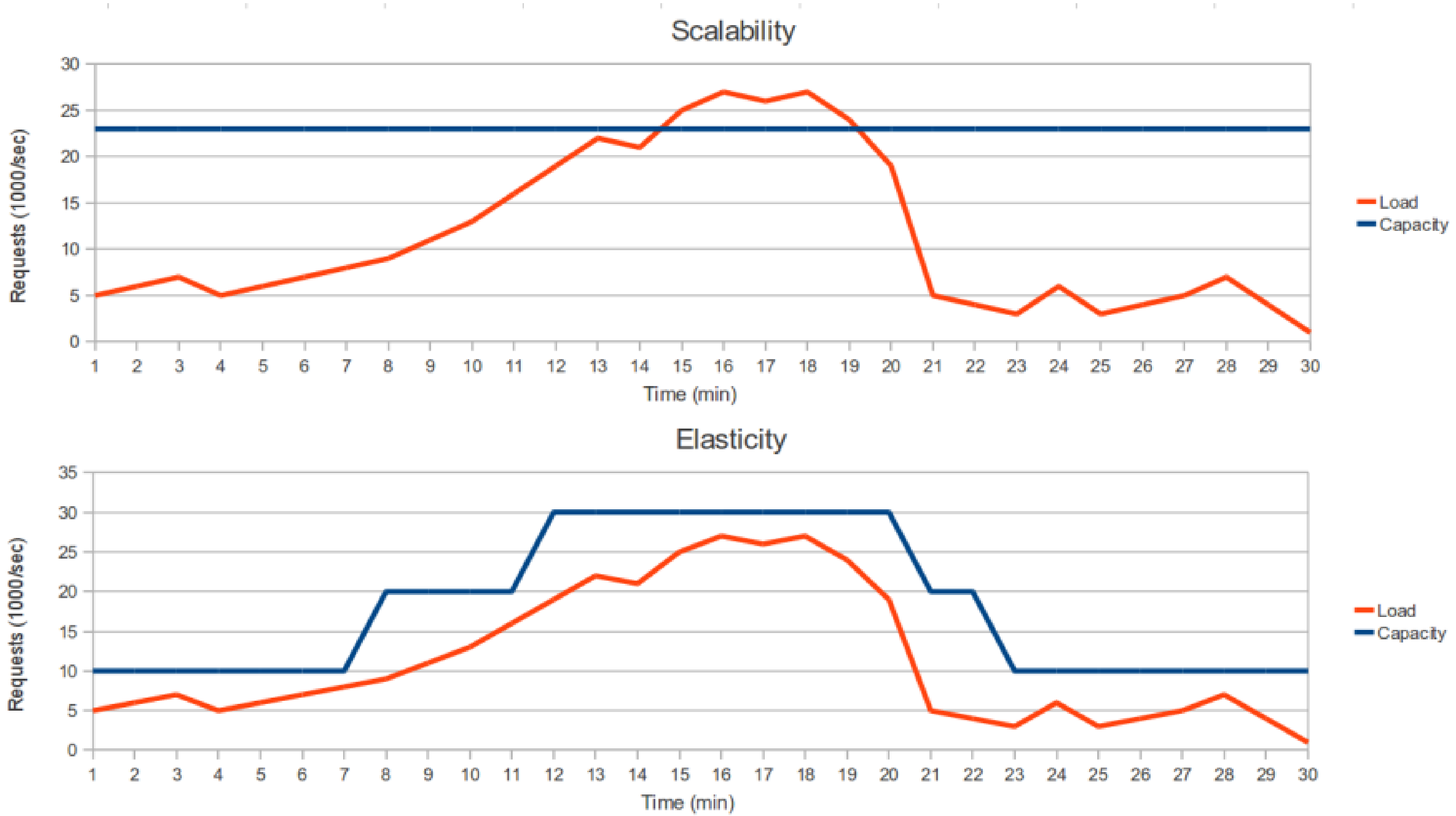
Over-provisioning a Storage System



(assuming data stored on ten servers)

[From: “Solving the Scalability Dilemma with Clouds, Crowds, and Algorithms”, Michael Franklin, UC Berkeley]

Elasticity versus Static Provisioning



ROYAL INSTITUTE OF TECHNOLOGY

Outline

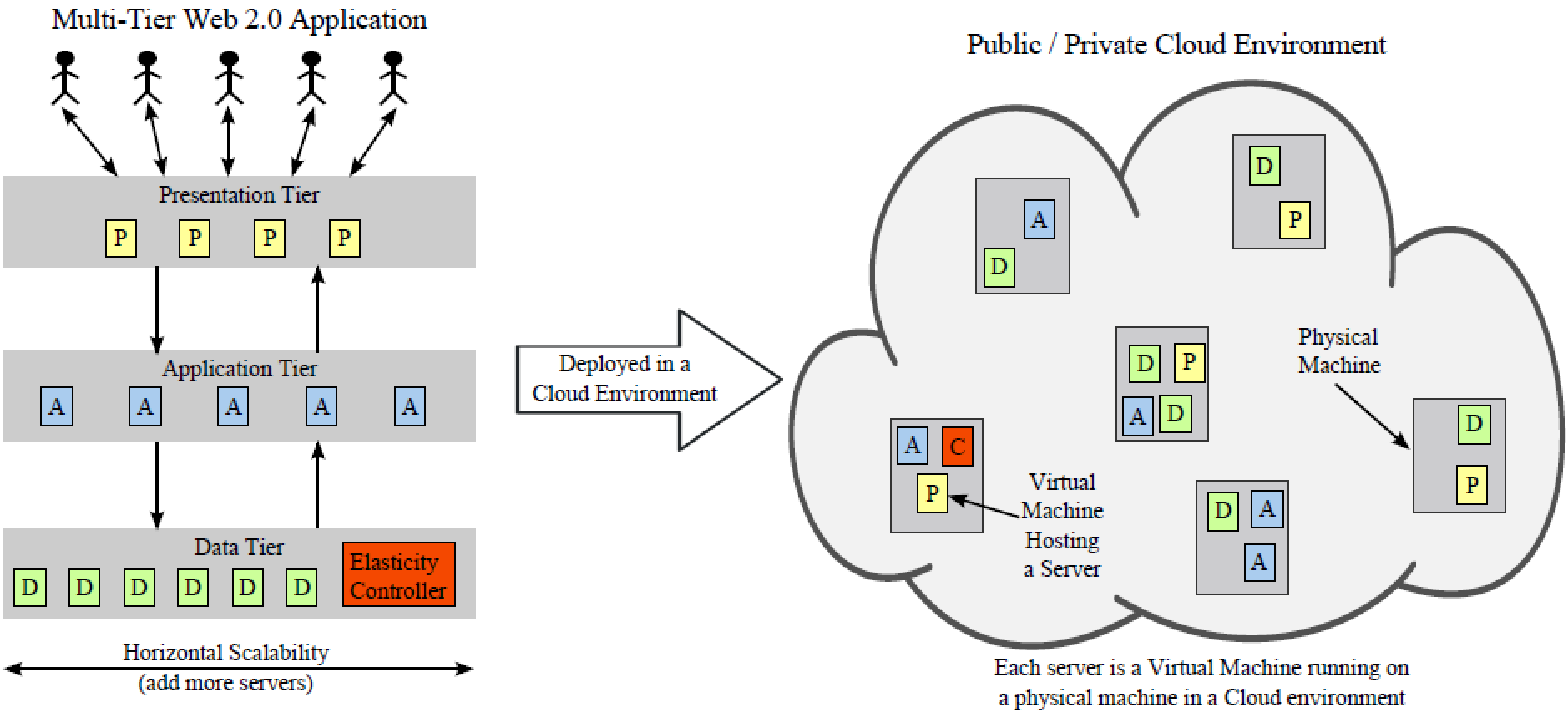
- Automation of elasticity
- Elasticity control for a cloud-based elastic storage
  - Requirements; challenges; approaches
- ElastMan: an elasticity manager for Cloud-based key-value stores
  - Feedforward control to handle workload spikes
  - Feedback control to handle gradual workload changes
- Evaluation of ElastMan in Voldemort key-value store
- Conclusions

KTH VETENSKAP OCH KONST
ROYAL INSTITUTE OF TECHNOLOGY

Automation of Elasticity

- Elasticity can be controlled either manually (by the sys-admin) or automatically (by a autonomic manager)
- Automation of elasticity can be achieved by providing an **Elasticity Controller**
  - Helps to avoid SLO violations while keeping the cost low
  - Automatically adds/removes VMs (servers, service instances) in response to changes in some SLO metrics, e.g., access latency, caused by changes in workload
- Can be built using elements of Control Theory and/or Machine Learning techniques
  - Feedback-loop (a.k.a. closed-loop) control
  - Model Predictive Control (MPC)

Target System



Storage Services. Key-Value Stores

- 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

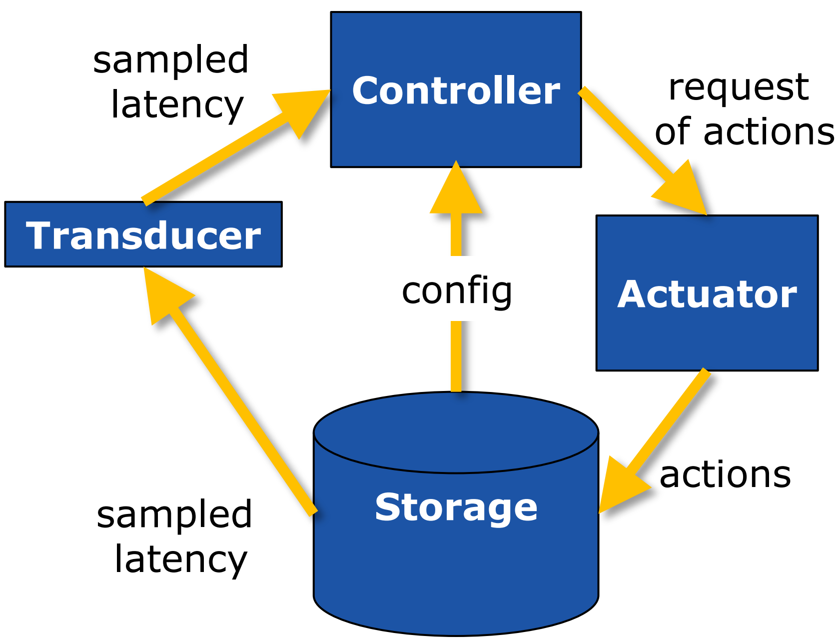
ROYAL INSTITUTE OF TECHNOLOGY

An Elasticity Controller for Cloud-Based Key-Value Stores

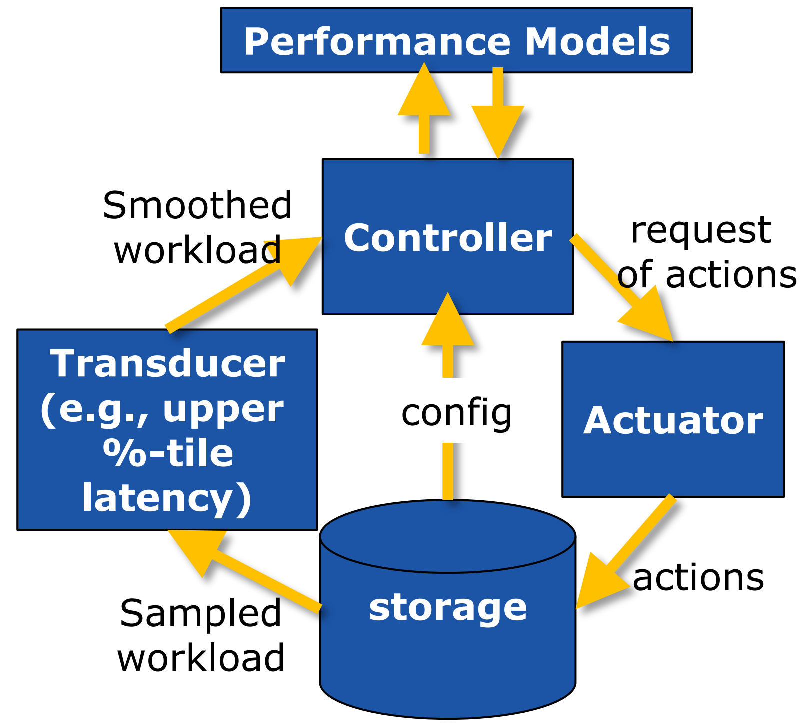
- Objective
  - To meet SLOs at a minimal cost by controlling the elasticity (size) of Cloud-based key-value stores by adding/removing resources (VMs, storage nodes)
- SLO Examples
  - Average read latency in one minute interval is less than 10ms
  - 99% of reads in one minute interval are performed in less than 10ms per read

Approaches to Automated Elasticity Control for a Cloud-Based Storage

- Closed-loop control
  e.g., [Lim2010][Mou2012]



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



Requrements for Elastic Storage

- **Horizontal scalability**
  - The storage and I/O capacity scales (roughly linearly) with the number of the active servers.
- **Touch points**
  - Sensors to monitor workload and performance (e.g., read latency);
  - Actuators to add/remove resources and service instances
- **Load balancing (re-balancing)**
  - Distribute data across servers to effectively balance load;
  - Redistribute (rebalance) data in response to join/leave events
- **Replication**
  - For robust availability: enough to avoid interruptions on leave events

Challenges of Elasticity Control for a Cloud-Based Storage

- Actuator delays due to data movement (rebalancing)
- Interference with applications and sensor measurements
- Discrete storage units
- Nonlinearity due to diminishing reward of adding a storage unit with increasing scale
- 99th percentile of access latency is a relatively noisy signal
- VM performance is difficult to model and predict
- Highly dynamic workload that is composed of both gradual (diurnal) and sudden (spikes) variations

Touch-points

Sensors

- To monitor workload and performance (e.g., access latency)
- A controller input includes one or more system performance metrics (system outputs)
- Requirements to system metrics [Lim 2010]
  - Easy to measure accurately
  - Should expose the system(tier)-level behavior or performance
  - Should be reasonably stable
  - Should correlate to the measure of service level specified in SLO

Actuators

- To add/remove resources and service instances
  - Cloud APIs to request/release server instances (VMs)
  - Storage API to request handling joins and leaves and rebalancing

ROYAL INSTITUTE OF TECHNOLOGY

Sensors (1/3)

SLO metrics (what to sense/measure/monitor)

- Request latency
  - response time, service latency, download latency, etc.
  - End-user /client experience, QoS
- System-wide performance metrics
  - CPU utilization, memory usage, network utilization
- Cost
- Other metrics

Sensors (2/3)

- In many cases, a system performance metric strongly correlates with the overall request latency (response time)
- Performance counters (CPU/memory/network utilizations), can be used as sensors

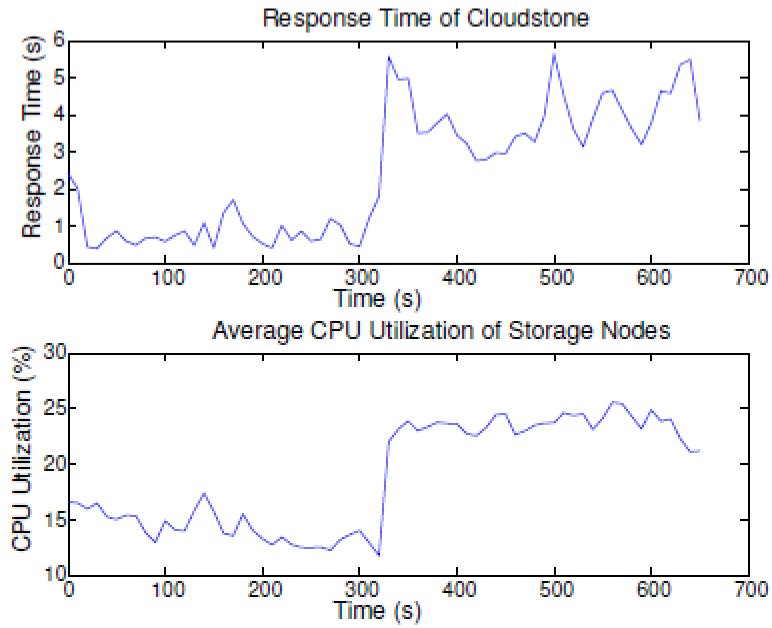


Figure 2: Cloudstone response time and average CPU utilization of the storage nodes, under a light load and a heavy load that is bottlenecked in the storage tier. CPU utilization in the storage tier correlates strongly with overall response time (the coefficient is .88), and is a more stable feedback signal.

[Lim2010]

ROYAL INSTITUTE OF TECHNOLOGY

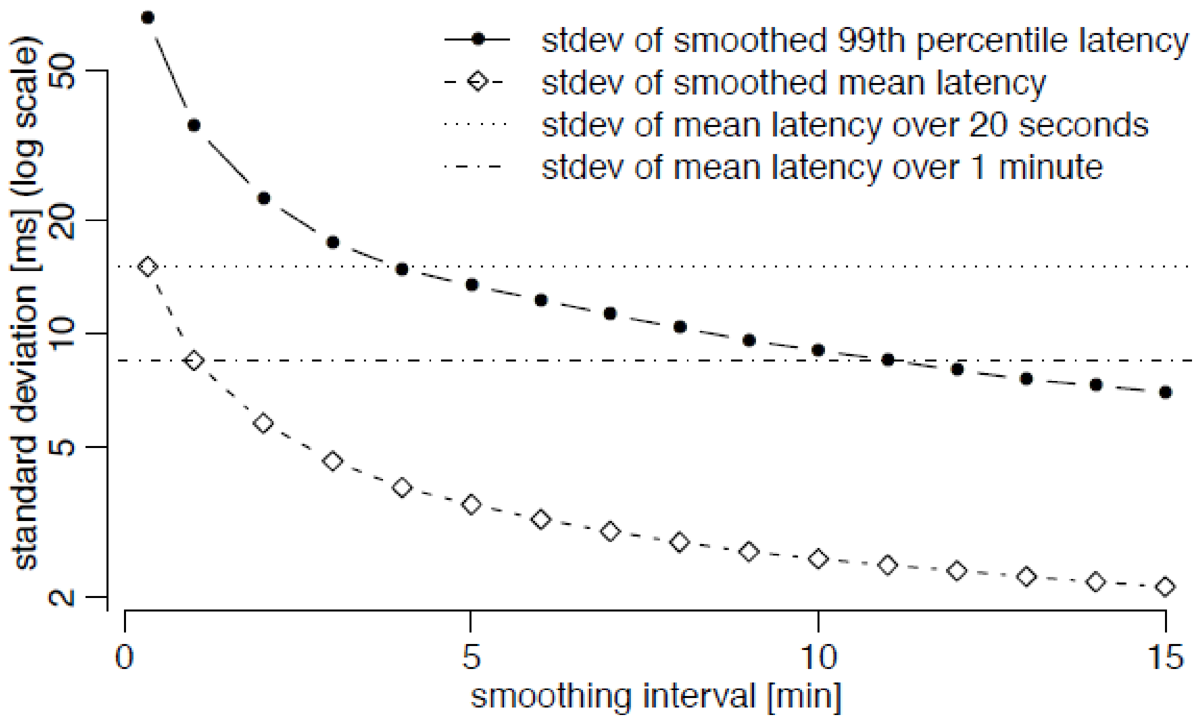
Sensors (3/3)

SLO requirements (what to regulate)

- Average values (means) of the SLO metrics
  - E.g., an average response time, an average CPU utilization
  - Classical closed-loop control
  - Example: HSC integral controller of HDFS-based storage [Lim2010]
- Upper quantiles of the SLO metrics
  - E.g. 99^{th} percentile of latency
    "99% of all requests must be answered within 100ms"
  - Model Predictive Control
  - Example: SCADS Director for SCADS storage (UCB) [Tru2011]

Mean versus Upper Quatile

- Usually, upper percentiles of latency measurements are very “noisy”
- A noisy latency signal can cause oscillations in classical closed-loop control



Standard deviation for the mean and 99th percentile of latency for increasing smoothing window sizes. [Tru2011]

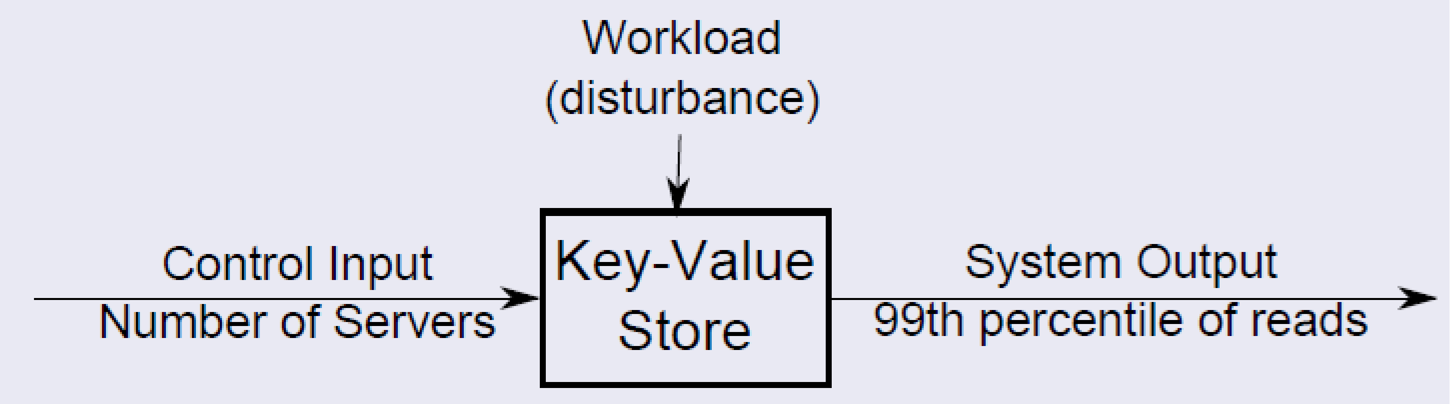
ROYAL INSTITUTE OF TECHNOLOGY

ElastMan: Elasticity Manager for Cloud-Based Key-Value Stores

- Addresses the challenges
  - the variable performance of VMs,
  - dynamic workload (spikes, diurnal changes),
  - stringent performance requirements,
- Combines and leverages the advantages of feedback and feedforward control
- Once designed, the controller can operate for different sizes of the key-value store
- Evaluated with LinkedIn's Voldemort key-value store deployed in our private OpenStack Cloud

Modeling the Store (1/2)

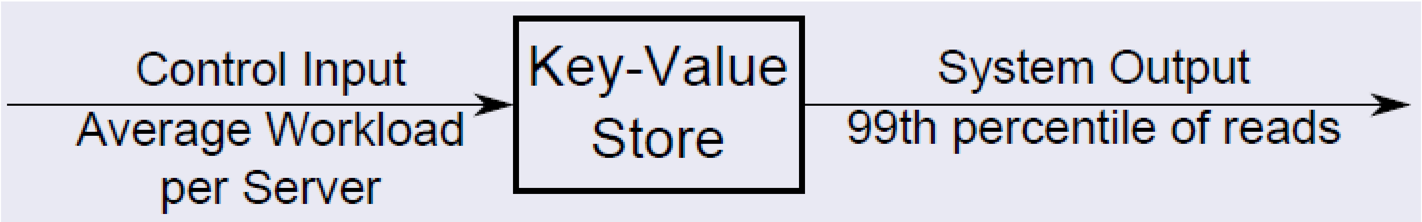
Typical approach



- Non linear Model
  - Adding 1 node to a 1 node system –> doubles the performance
  - Adding 1 node to a 100 nodes system –> only 1% improvement
- Workload treated as disturbance

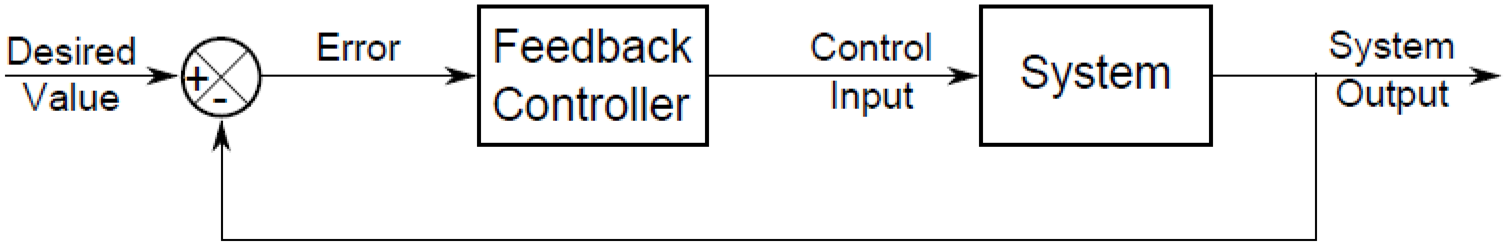
Modeling the Store (2/2)

ElastMan approach



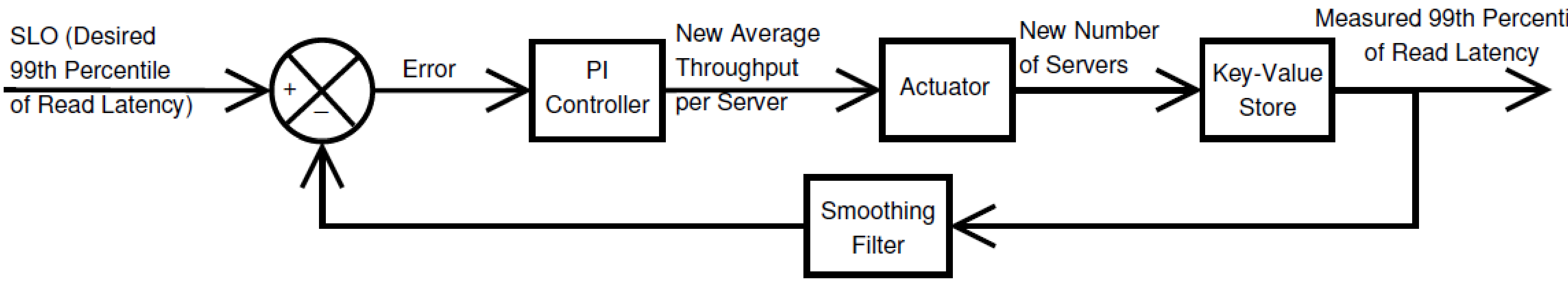
- Control the number of servers indirectly by controlling the average workload per server
- Relies on near linear scalability of key-value stores

Feedback Control [Hel2004]

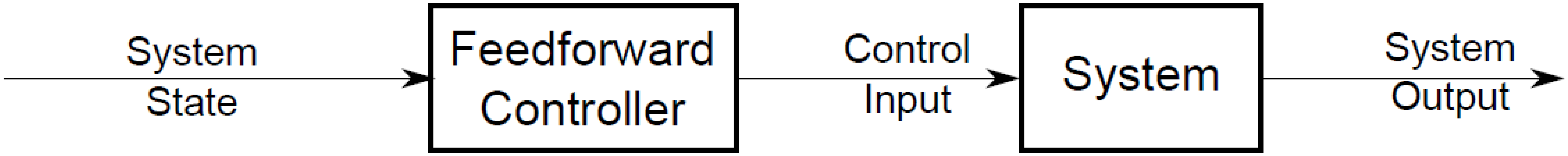


- The system's output (e.g., response time) is being monitored
- Controller calculates the control error
- Controller changes the control input (e.g., number of servers to add or remove) according to the amount and sign of the control error
- Advantage: controller can adapt to disturbance
- Disadvantages: oscillation, overshoot, possible instability

ElastMan Feedback Controller

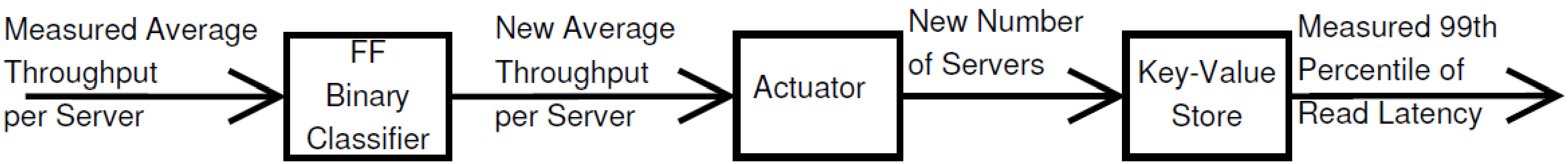


Feedforward Control [Hel2004]

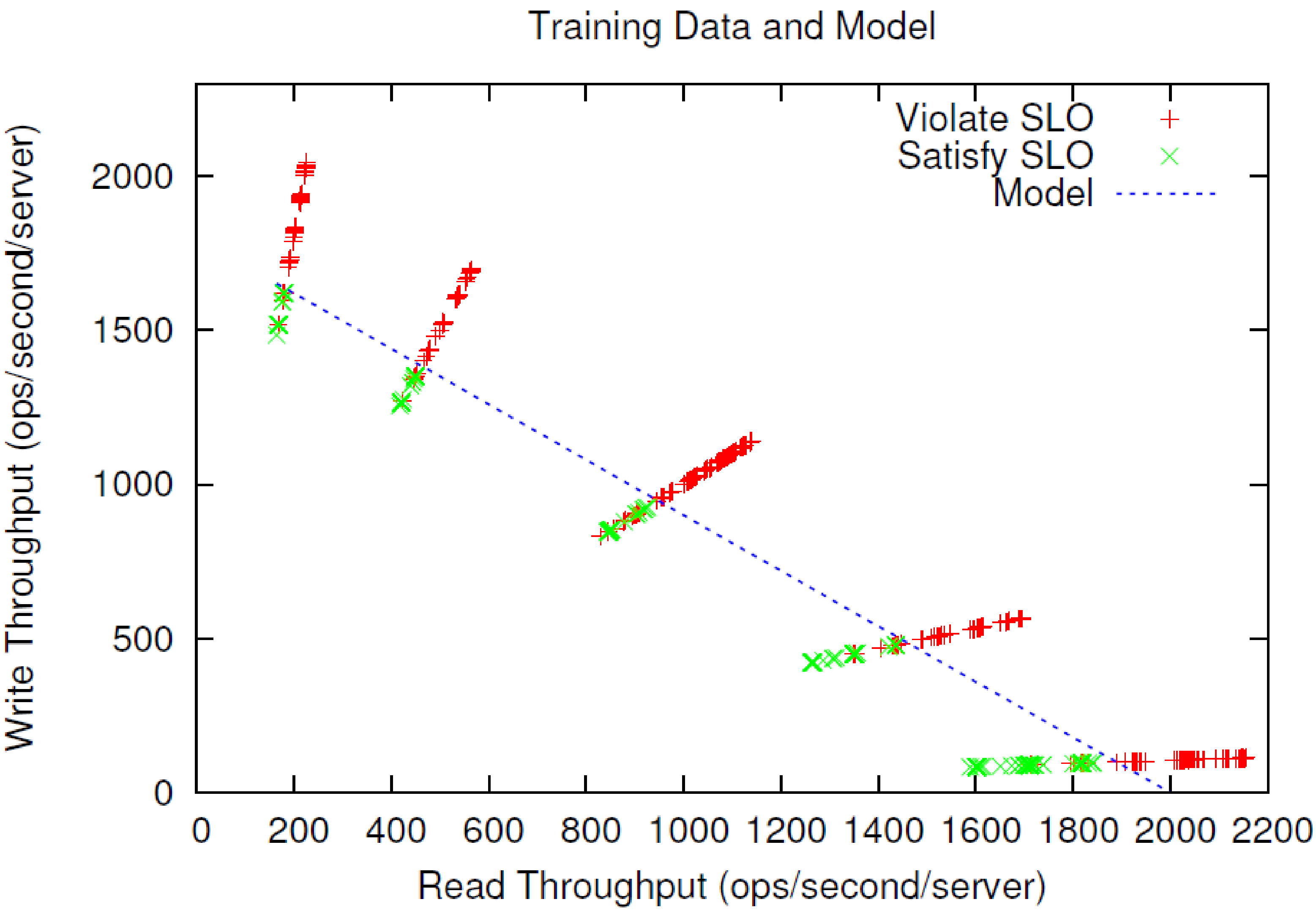


- The system's output is not monitored
- Other system states and variables are monitored
- Controller relies on a model of the system to calculate necessary
- change
- Advantages: faster and avoids oscillation and overshoot
- Disadvantages: sensitive to unexpected disturbances that are not modeled

ElastMan Feedforward Controller



Binary Classifier



ElastMan: Combining Feedback and Feedforward Control

Feedback

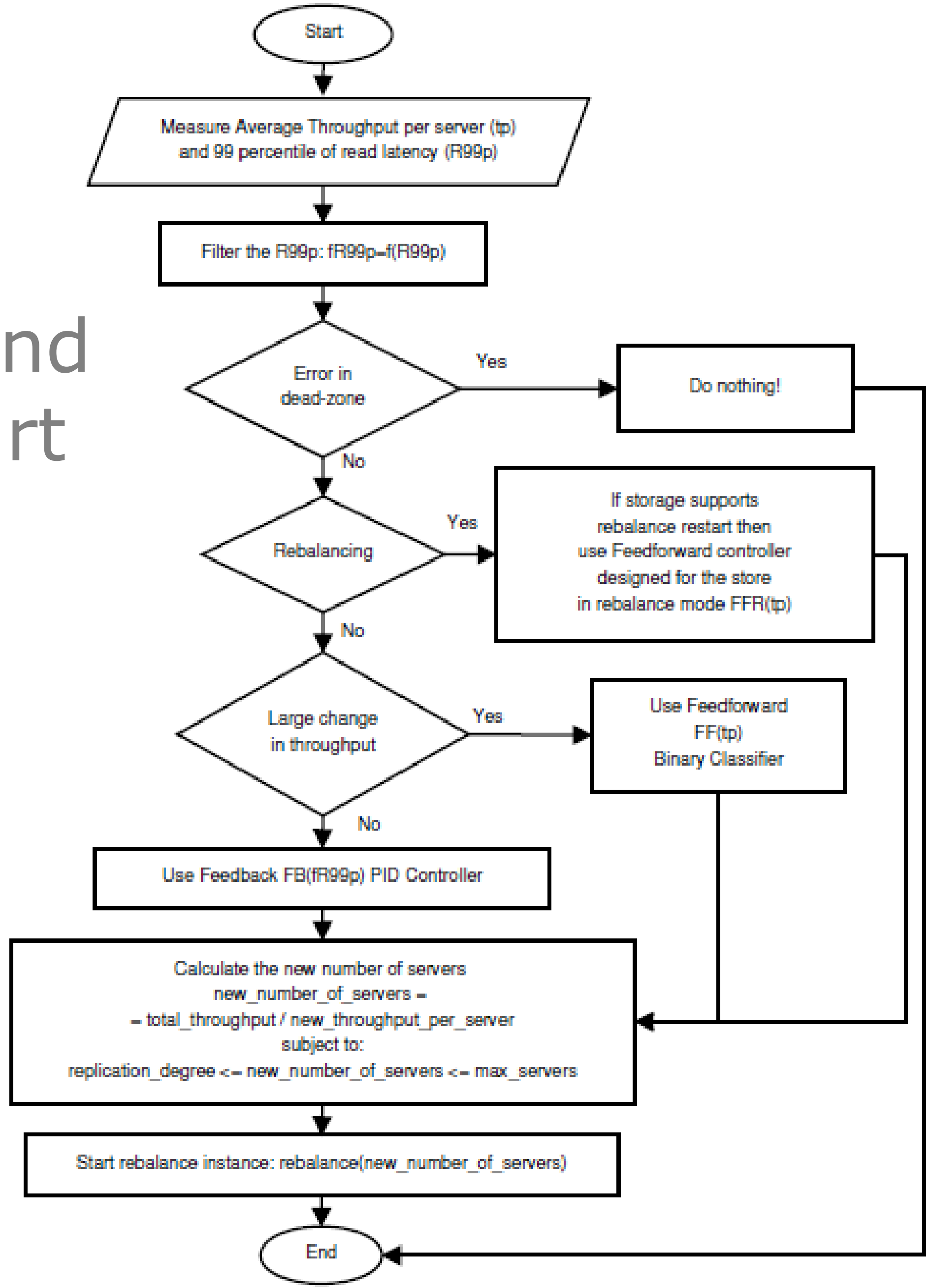
- Classical PI controller
- Deals with diurnal workloads (e.g., slow day-night changes)
- Monitors 99th percentile of read latency
  - E.g., the read latency of 99% of reads in a 1 min interval is at most 10ms
- Can tolerate and adapt to modeling errors

Feedforward

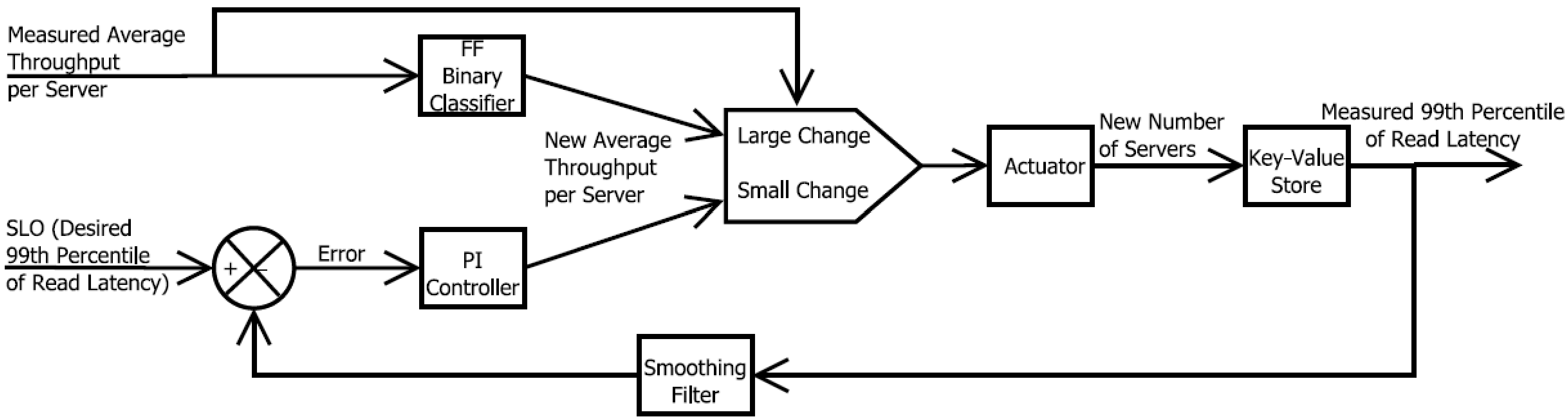
- A binary classifier using logistic regression
- Deals with workload spikes (large rapid changes)
- Monitors workload (intencity of reads and intencity of writes)
- Allows smoothing the noisy 99th percentile signal

ROYAL INSTITUTE OF TECHNOLOGY

Combined Feedback and Feedforward Flow Chart



Combined Feedback and Feedforward Controller

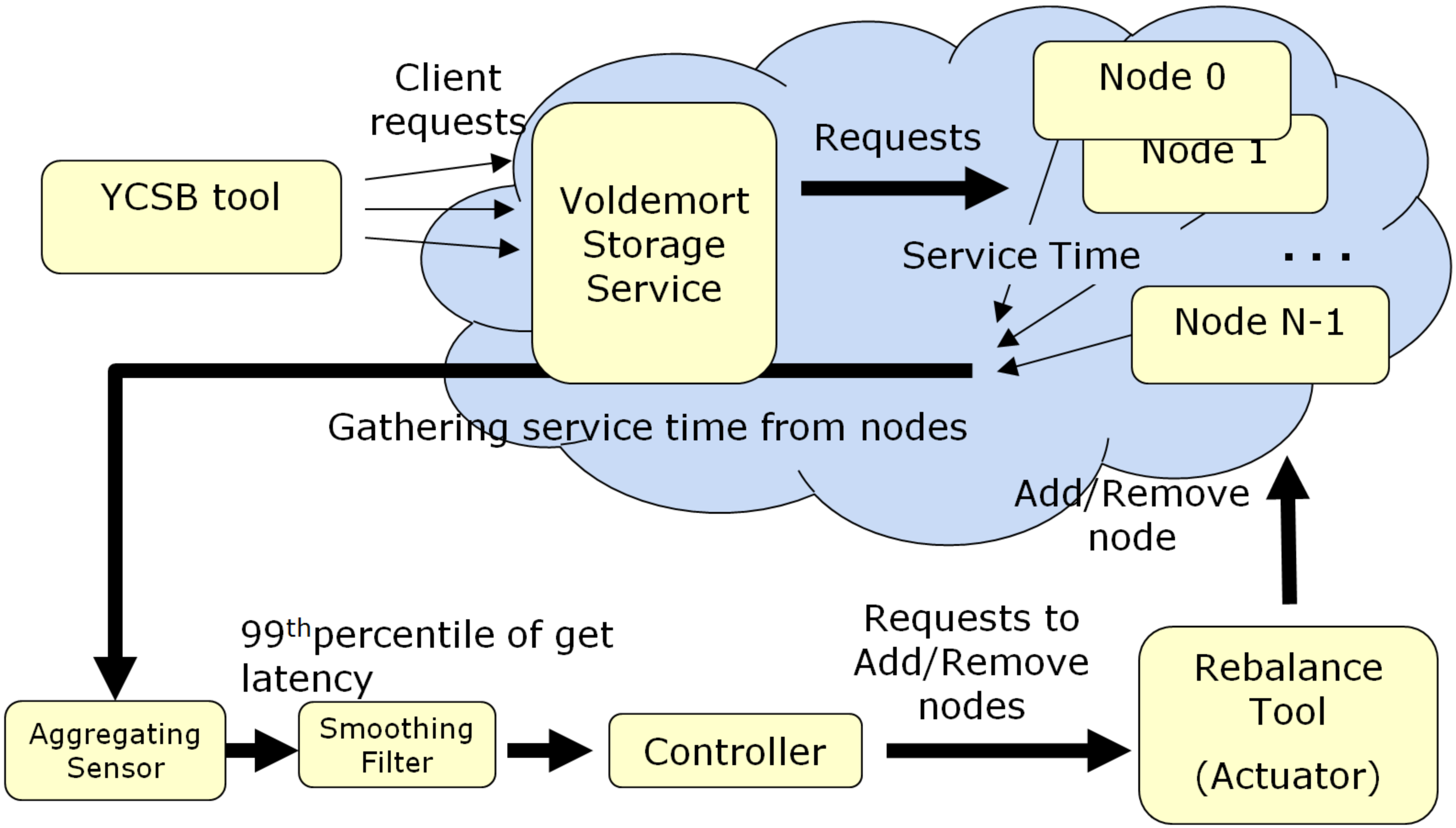


ROYAL INSTITUTE OF TECHNOLOGY

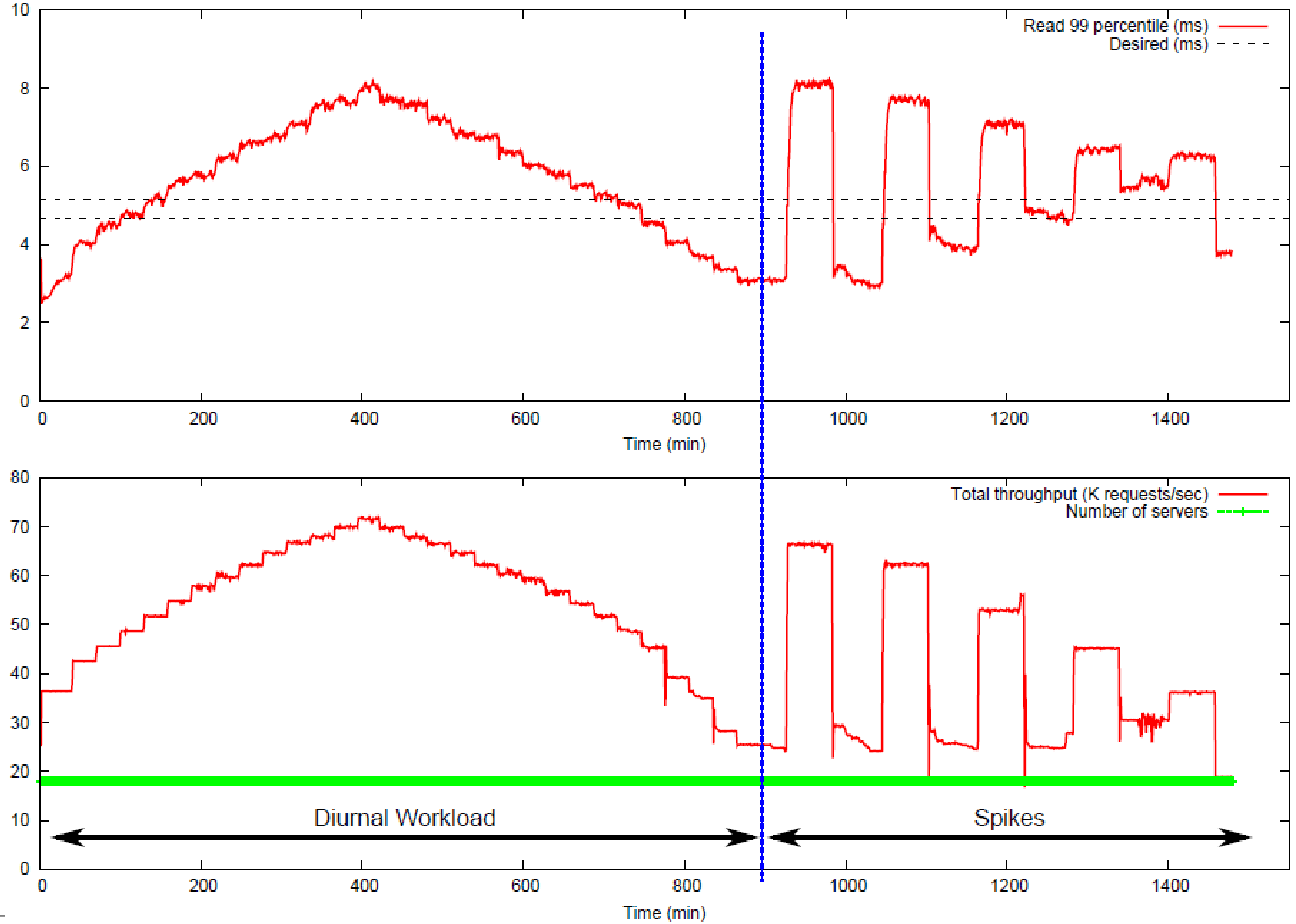
Evaluation

- Implemented a prototype of the ElastMan Elasticity Controller
- Evaluated with LinkedIn’s Voldemort key-value store
- Deployed in our private OpenStack Cloud

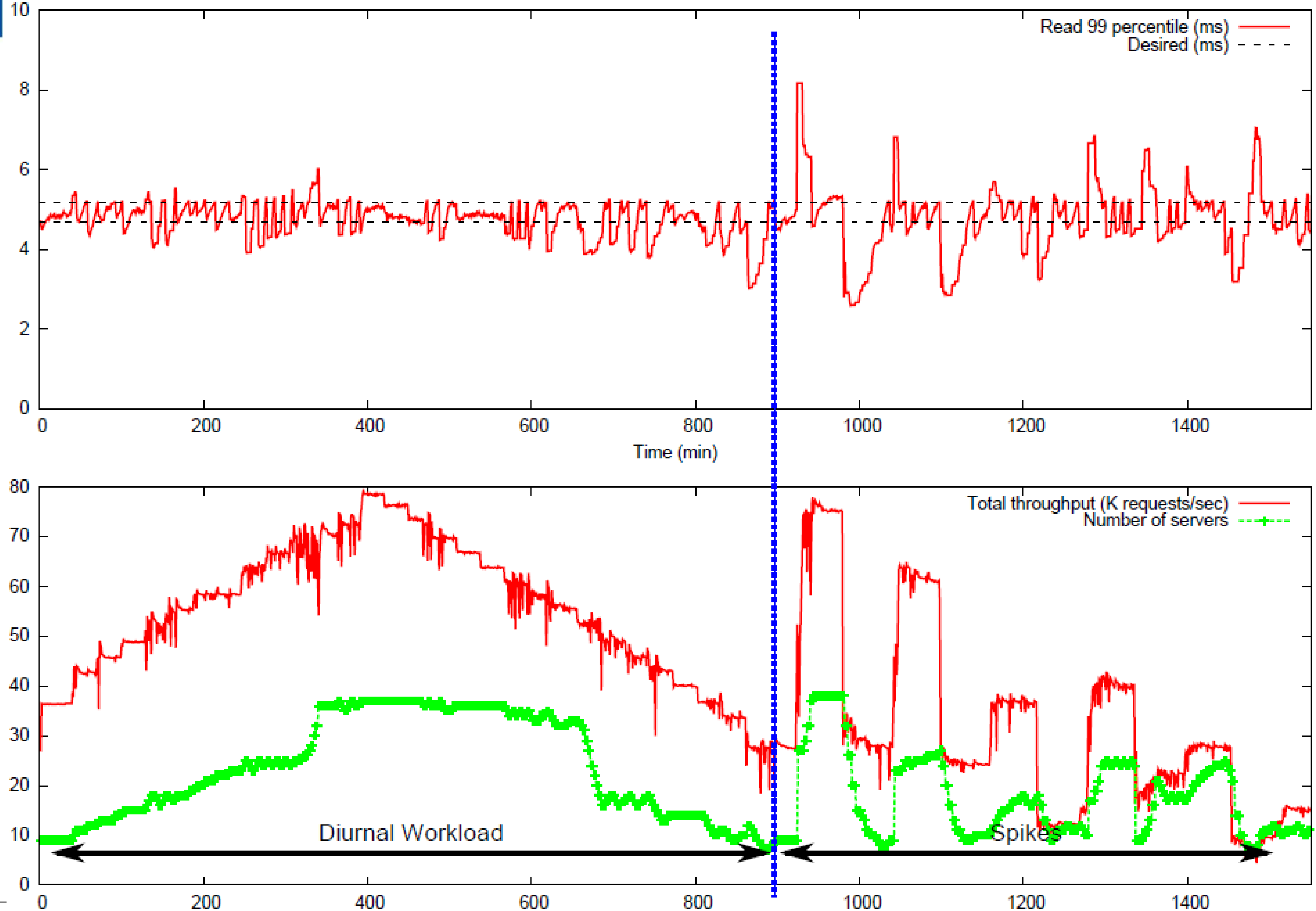
Voldemort Key-Value Store with the ElastMan Elasticlity Controller



Voldemort performance without ElastMan



Voldemort performance with ElastMan



ROYAL INSTITUTE OF TECHNOLOGY

Conclusions

- ElastMan addresses the challenges of the variable performance of Cloud VMs, dynamic workload, and stringent performance requirements
- ElastMan combines and leverages the advantages of both feedback and feedforward control
  - feedforward control quickly responds to rapid changes in workload
  - feedback controller handles diurnal workload and to correct modeling errors in the feedforward control
- Evaluation results show the feasibility of the ElastMan approach

ROYAL INSTITUTE OF TECHNOLOGY

ElastMan References

- Ahmad Al-Shishtawy, Vladimir Vlassov, ***ElastMan: Elasticity Manager for Elastic Key-Value Stores in the Cloud***, The ACM Cloud and Autonomic Computing Conference (CAC 2013), Miami, FL, USA, August 5-9, 2013.
- Ahmad Al-Shishtawy, Vladimir Vlassov, ***ElastMan: Autonomic Elasticity Manager for Cloud-Based Key-value Stores***,The 22nd ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC ’13). ACM, New York, NY, USA, pp. 115-116.

- http://www.ict.kth.se/~ahmadas/ElastMan/

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, 2010
- **[Mou2012]** M. A. Moulavi, A. Al-Shishtawy, and V. Vlassov, *State-Space Feedback Control for Elastic Distributed Storage in a Cloud Environment*, ICAS 2012
- **[Tru2011]** Beth Trushkowsky, et al., *The SCADS director: scaling a distributed storage system under stringent performance requirements*. The 9th USENIX Conf on File and Storage Technologies (FAST'11), 2011