State-Space Feedback Control for Elastic Distributed Storage in a Cloud Environment

M. Amir Moulavi
Ahmad Al-Shishtawy
Vladimir Vlassov

KTH Royal Institute of Technology, Stockholm, Sweden
ICAS 2012, March 26, St. Maarten, Netherlands Antilles
Agenda

• Introduction & Problem Definition
• System Identification
• Controller Design
• EStoreSim: Elastic Store Simulator
• Evaluation
• Conclusions
Motivation

• Web 2.0 applications
  - WiKis, social networks, media sharing
  - Rapidly growing number of users and the amount of user generated data

• Challenges for a storage service
  - Growing number of users and the amount of data (scalability)
  - Uneven load, user geographically scattered (low request latency, load balancing)
  - Partial failures, very high load (high availability)
  - Acceptable data consistency guarantees (e.g., eventual consistency)
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**
Need for Elasticity

• Web services, e.g. storage, 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 VM instances to scale the service in order to meet SLOs at a minimal cost
Elasticity versus Static Provisioning

Scalability

Elasticity

ICAS 2012, March 26, St. Maarten, Netherlands Antilles
Automation of Elasticity

- Elasticity can be done 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., 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)
Feedback (Closed Loop) Control [Hel2004]
Automatic Control of Storage Elasticity in the Cloud

ICAS 2012, March 26, St. Maarten, Netherlands Antilles
Two Phases of a Feedback Controller Design

**System identification**
- Building a mathematical model of a dynamic system
  - How control outputs depend on control inputs
- Two main approaches:
  - First-principle (e.g., using queuing theory)
  - Black-box (e.g., state-space)

**Controller design**
- Choose a controller type (e.g., PID, State-Space)
- Determine controller gains based on the system model
State-Space Model

**Advantages**
- Provides scalable approach to model systems with large number of inputs/outputs
- Can be extended easily

**State variables**
- Express the dynamics of the system

**Main Steps**
- Study system behavior in order to identify the control inputs, control outputs, and state variables of the system
- Construct the characteristic equations
- Design an experiment to estimate the parameters of the characteristic equations

ICAS 2012, March 26, St. Maarten, Netherlands Antilles
Control Inputs/Outputs of an Elastic Storage System

Cloud-based Elastic Key-Value Store

Input \( u \): Number of Instances

Output \( y = \text{state } x \):
- Average CPU Load
- Total Cost
- Average Response Time

Load
A state space model considers relationship between inputs $\mathbf{u}$, outputs $\mathbf{y}$, and state variables $\mathbf{x}$.

State variables used in two ways:
- Describe dynamics (state changes)
- Determine the measured output from the state

\[
\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k)
\]
\[
\mathbf{y}(k) = C\mathbf{x}(k)
\]

Allows modeling of a MIMO system with multiple inputs and outputs.
Characteristic Equations

\[ x_1(k + 1) = CPU(k + 1) = \]
\[ a_{11} CPU(k) + b_{11} NN(k) + 0 \times TC(k) + 0 \times RT(k) \]

\[ x_2(k + 1) = TC(k + 1) = \]
\[ a_{21} TC(k) + b_{21} NN(k) + 0 \times RT(k) + 0 \times CPU(k) \]

\[ x_3(k + 1) = RT(k + 1) = \]
\[ a_{31} CPU(k) + a_{33} RT(k) + b_{31} NN(k) + 0 \times TC(k) \]

\[ y(k) = I_3 x(k) \]

\[ A = \begin{bmatrix}
    a_{11} & 0 & 0 \\
    0 & a_{22} & 0 \\
    a_{31} & 0 & a_{33}
\end{bmatrix} \]

\[ B = \begin{bmatrix}
    b_{11} \\
    b_{21} \\
    b_{31}
\end{bmatrix} \]

\[ C = \begin{bmatrix}
    1 & 0 & 0 \\
    0 & 1 & 0 \\
    0 & 0 & 1
\end{bmatrix} \]
Identification: Estimate the coefficient matrices $A$, $B$ and $C$ from experimental data

- Feed the system with an input signal and observe outputs and internal state variables periodically.
- Compute the matrices from the collected data using the *multiple linear regression method* - The Matlab `regress(y,X)` function can be used to calculate matrices

\[
A = \begin{bmatrix}
0.9 & 0 & 0 \\
0 & 0.724 & 0 \\
5.927 & 0 & 0.295
\end{bmatrix} \quad B = \begin{bmatrix}
2.3003 \\
0.0147 \\
77.8759
\end{bmatrix}
\]
Controller Design

- Dynamic State Feedback
  - a State-Space analogous to PI (Proportional Integral) control
- Has good disturbance rejection properties
- the control error is
  \[ e(k) = r - y(k) \]
- The integrated control error is
  \[ x_I(k + 1) = x_I(k) + e(k) \]
- The control law is
  \[ u(k) = - [K_p \quad K_I] \begin{bmatrix} x(k) \\ x_I(k) \end{bmatrix} \]
LQR Controller Design

• LQR: Least Quadratic Regulation
• An approach to controller design is to focus on the tradeoff between control effort and control errors
• Minimizing control errors (Defined by matrix R):
  • Improve accuracy and reduce both settling times and overshoot
• Minimizing control effort (Defined by matrix Q):
  - Sensitivity to noise is reduced

\[
Q = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\quad R = [1]
\]
LQR Controller Design

- **Given:** $A$ and $B$ (from system identification), weighting matrices $R$ and $Q$ (state/input and output, respectively), and the quadratic cost function $J$;

  \[ J = \frac{1}{2} \sum_{k=0}^{\infty} \left[ x^T(k)Qx(k) + u^T(k)Ru(k) \right] \]

- **Find:** The controller gain vector $K$ (for three outputs) that minimizes the quadratic cost function $J$ for given $R$ and $Q$;

- **Use** Matlab `dlqr()` function: $K = \text{dlqr}(A, B, Q, R)$

  \[ K = \begin{bmatrix} 0.134 & 1.470162e-06 & 0.00318 \end{bmatrix} \]
EStoreSim: Elastic Key-Value Store Simulator

- Simulates a Cloud environment
  - VMs (CPU & Memory)
  - Network (Upload bandwidth)
  - ...
- Generates various workload patterns
- Supports controller design
  - Run system identification experiments and gather data
  - Experiment with different controller designs
EStoreSim: Elastic Key-Value Store Simulator

Implementation

- Based on Kompics
- Written in Java, Scala
- Publicly available on
  https://github.com/amir343/ElasticStorage
Evaluation

- **SLO Requirements**
  - Average CPU Load ≤ 55%
  - Response Time ≤ 1.5 seconds
  - Average Bandwidth per download > 200 KB/s

- **Two Experiments:**
  - SLO Experiment
  - Cost Experiment
SLO Experiment

- Workload (interarrival time)
SLO Experiment

Average CPU Load

Time (s)

Average CPU Load (%)

NL

HL

JSLA Requirement < 55%

w/ controller

w/o controller
SLO Experiment

Average Bandwidth per download

- w/ controller
- w/o controller

Time (s)

Average Bandwidth (B/s)

KTH Royal Institute of Technology

ICAS 2012, March 26, St. Maarten, Netherlands Antilles
SLO Experiment

![Interval Total Cost Graph]

- **w/ controller**
- **w/o controller**

- NL (Normal Load)
- HL (High Load)

ICAS 2012, March 26, St. Maarten, Netherlands Antilles
SLO Experiment

![Graph showing number of nodes over time with and without a controller. The graph has two lines: one for 'w/ controller' and one for 'w/o controller.' The x-axis represents time in seconds, ranging from 0 to 2000. The y-axis represents the number of nodes, ranging from 3 to 9. The graph shows fluctuations in the number of nodes, with 'NL' and 'HL' indicating different states or events.]
Cost Experiment

<table>
<thead>
<tr>
<th>Workload</th>
<th>w/ controller</th>
<th>w/o controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Cost ($)</td>
<td>10,5</td>
<td>16,5</td>
</tr>
</tbody>
</table>

ICAS 2012, March 26, St. Maarten, Netherlands Antilles
Conclusions

• **Elasticity** in Cloud computing is an ability of a system to **scale up and down** in response to changes in its environment and workload
  - Improves Cloud-based systems by **reducing the total cost** for the system while meeting SLOs
• Described the **steps** in designing an elasticity controller for a Cloud-based **key-value store**
• **EStoreSim**: Open source simulation framework for Cloud systems
• Experiments have shown the **feasibility** of our approach to **automate** elasticity control of storage services using **state-space** feedback control.