DISCRETE TIME ADAPTIVE LINEAR CONTROL FOR CLOUD APPLICATIONS

Martina Maggio
Lund University
“Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems with inputs, and how their behavior is modified by feedback.”
CONTROL ARCHITECTURE

Plant
CONTROL ARCHITECTURE

Actuators → Plant → Measurements
CONTROL ARCHITECTURE

Actuators \[\rightarrow\] Linear Model \[\rightarrow\] Measurements
CONTROL ARCHITECTURE

Objectives → Actuators → Linear Model → Measurements
CONTROL ARCHITECTURE

Objectives → Error → Actuators → Linear Model → Measurements

+ → -
CONTROL ARCHITECTURE

Objectives  Error  Actuators  Measurements

+  -

Controller  Linear Model
CONTROL ARCHITECTURE

Adaptive control: updating the **controller** online based on data from the software.
CONTROL ARCHITECTURE

Adaptation Linearization

Objectives Error Actuators Measurements

Controller

Linear Model

“Automated design of self-adaptive software with control theoretical formal guarantees” A. Filieri, H. Hoffmann, M. Maggio; ICSE 2014
CONTROL CONTRIBUTION

• Startup:
  • we perform experiments on the system
  • we build a first-order linear model, using the results
  • we automatically synthesize a controller

• The controller is up and running autonomously:
  • we keep the model updated during runtime
  • in case of extreme error, we trigger a new startup

“Automated design of self-adaptive software with control theoretical formal guarantees” A. Filieri, H, Hoffmann, M. Maggio; ICSE 2014
FORMAL GUARANTEES

- **Stability**: objectives reached when feasible
- **No-overshooting**: without overshooting
- **Short settling time**: in a short time
- **Robustness**: despite model inaccuracies
CONTROLLING THE CLOUD

BROWNOUT
A brownout is an intentional or unintentional drop in voltage in an electrical power supply system. Intentional brownouts are used for load reduction in emergency conditions.
A brownout is an intentional or unintentional drop in voltage in an electrical power supply system. Intentional brownouts are used for load reduction in emergency conditions.
PREDICTABILITY: MOTIVATION

✓ 82% of end-users give up on a lost payment transaction*
✓ 25% of end-users leave if load time > 4s**
✓ 1% reduced sale per 100ms load time**
✓ 20% reduced income if 0.5s longer load time***

*JupiterResearch, **Amazon, ***Google
STATE OF THE ART

Increase capacity

Public cloud
BROWNOUT
DISABLE NON-ESSENTIAL CONTENT ON DEMAND
CONTROL ARCHITECTURE

- Objectives
- Error
- Actuators
- Measurements

Adaptation Linearization

Controller

Linear Model
CONTROL ARCHITECTURE

Adaptation
Linearization

Controller

Software

Objectives          Error          Actuators          Measurements
The 95th percentile of the responses should be produced in less than 1 second
CONTROL ARCHITECTURE

The dimmer denotes the percentage of requests served with the optional computation enabled.
We need to measure the 95th percentile of the response times to know how far we are from our objective.
CONTROL ARCHITECTURE

The adaptation strategy linearizes the system and takes care of runtime changes (like slow downs due to memory leaks).
CONTROL ARCHITECTURE

- Software equation-based model:

\[ y(k + 1) = y(k) + \alpha u(k) \]
CONTROL ARCHITECTURE

- Software equation-based model:

\[ y(k + 1) = y(k) + \alpha u(k) \]

The value of \( \alpha \) determines the runtime behavior of the metric that is monitored.
CONTROL ARCHITECTURE

- Equation-based controller:

\[ u(k + 1) = u(k) + \frac{1 - p}{\hat{\alpha}} e(k) \]
CONTROL ARCHITECTURE

• Equation-based controller:

\[ u(k + 1) = u(k) + \frac{1 - p}{\hat{\alpha}} e(k) \]

The controller uses an online estimate of \( \alpha \) (corresponds to linearizing around the operating point)
CONTROL ARCHITECTURE

• Closed loop system:

\[ G(z) = \frac{1 - p}{z - p} \]
CONTROL ARCHITECTURE

• Closed loop system:

\[ G(z) = \frac{1 - p}{z - p} \]

The value of \( p \) controls the speed of convergence.
CONTROL ARCHITECTURE

• Robustness to model inaccuracies: \( \alpha = \hat{\alpha} \cdot \Delta \alpha \)
CONTROL ARCHITECTURE

- Robustness to model inaccuracies: \( \alpha = \hat{\alpha} \cdot \Delta \alpha \)

\[
0 \leq \Delta \alpha \leq \frac{2}{1 - p}
\]
BROWNOUT

- Single replica
- Graceful degradation
  - Minimally intrusive (<200 lines of code)
  - Application developers mark optional code
  - **Automatic control strategy** to select when the optional computations should be turned on and off
EXAMPLE: E-COMMERCE WEBSITE
EXAMPLE: E-COMMERCE WEBSITE

![Graph showing performance metrics for an e-commerce website. The graph plots latency against resources used, with a spike in latency observed at high resource usage.](image-url)
EXAMPLE: E-COMMERCE WEBSITE

![Graph showing latency versus resources. The top graph indicates latency in seconds, while the bottom graph shows the dimmer level. The x-axis represents resources as a percentage of CPU, ranging from 0 to 1000. The graphs illustrate the impact of increasing resources on latency and dimmer level.]
AUTOMATIC CONTROL STRATEGY

• Compute **statistics** from response times

• Use statistics as a **feedback** signal

• Update the current model of the system (linearization)

⇒ Control strategy: selects the probability of executing the optional code (dimmer)

“Brownout: building more robust cloud applications”
C. Klein, M. Maggio, K.-E. Årzén, F. Hernández-Rodriguez, ICSE 2014
AUTOMATIC CONTROL STRATEGY

- Implemented and tested:
  - Adaptive PI controller
  - Adaptive PID controller
  - Deadbeat controller
  - Feedforward plus feedback controller

“Control strategies for predictable brownouts in cloud computing”
M. Maggio, C. Klein, K.-E. Årzén, IFAC WC 2014
AUTOMATIC CONTROL STRATEGY

- Implemented and tested:
  - **Adaptive** PI controller
  - Adaptive PID controller
  - Deadbeat controller
  - Feedforward plus feedback controller

“Control strategies for predictable brownouts in cloud computing”
M. Maggio, C. Klein, K.-E. Årzén, IFAC WC 2014
EXPERIMENTAL VALIDATION

Experiment with RUBBoS and varying users/resources
STATISTICAL VALIDATION

With control

Without control
SCALING UP

- Multiple replicas
- Need load balancing:
  - Based on latencies
  - SQF
  - Brownout-aware (PIBH, EPBH)

“Control-theoretical load-balancing for cloud applications with brownout”
“Improving Cloud Service Resilience using Brownout-Aware Load-Balancing”
CONCLUSION

• Bounded response times
• Improved fault tolerance
• Formal guarantees
• Minimally intrusive

https://github.com/cloud-control/
AND FUTURE WORK
Currently ongoing work: using the scenario theory to obtain probabilistic guarantees for self-adaptive software.
SCALING UP

- Multiple replicas
- Need migration:
  - Analysis of migration strategies and development of control-based migration management

“Virtual Machine Migration in Cloud Infrastructures: Problem Formalization and Policies Proposal”
A. V. Papadopoulos and M. Maggio, CDC 2015
MIGRATION

• Preliminary result show a comparison of different proposals (e.g. Sandpiper - T. Wood et al. USENIX NSDI 2007)

• Long way to the development of proper control-theoretical proposals for migration management (for example: no technique yet takes into account the difference between pre-copy and post-copy)
THANKS! :)