Recent Advances in the Application of Control Theory to Network and Service Management

Tarek Abdelzaher, University of Illinois
Yixin Diao, IBM Research
Joseph L. Hellerstein, Google
Chenyang Lu, Washington University
Sharad Singhal, Hewlett Packard Laboratories

June 1, 2009
Tutorial Agenda

- Control theory fundamentals (30 min)
- Managing power and performance in data centers (35 min)
- Self-tuning memory management in IBM’s DB2 (35 min)

BREAK

- Control of real-time systems using model-predictive control (35 min)
- Automated workload management in virtualized data centers (35 min)
- Research challenges (30 min)

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal
Tutorial Agenda

- Control theory fundamentals
  - Control architecture and taxonomy
  - Simple analytics
  - Application summaries
    - Regulating load for IBM’s Lotus Domino email server
    - Optimizing throughput for Microsoft’s .NET thread pool
- Self-tuning memory management in IBM’s DB2
- Control of real-time systems using model-predictive control
- Automated workload management in virtualized data centers
- Managing power and performance in data centers
- Research challenges

*IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal*
Elements of a Control System

Components
- Target system: what is controlled
- Controller: exercises control
- Transducer: translates measured outputs

Data
- Reference input: objective
- Control error: reference input minus measured output
- Control input: manipulated to affect output
- Disturbance input: other factors that affect the target system
- Transduced output: result of manipulation

Given target system, transducer
Control theory finds controller
that adjusts control input
to achieve measured
output in the presence of disturbances.

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal
Closed Loop vs. Open Loop

Closed Loop System

Open Loop System

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal
Types of Control

- Manage to a reference value
- Ex: Service differentiation, resource management, constrained optimization

Regulatory Control

- Eliminate effect of a disturbance
- Ex: Service level management, resource management, constrained optimization

Disturbance Rejection

- Achieve the “best” value of outputs
- Ex: Minimize Apache response times

Optimization

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal
The SASO Properties of Control Systems

Stability  Accuracy  Short Settling  Small Overshoot

Unstable System

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal
Control Theory By Example – IBM Domino Server

Architecture

Admin → Controller → MaxUsers → RPCs → Server → Desired RIS → Actual RIS

Block Diagram

Desired RIS → e(k) → Controller → u(k) → y(k) → MaxUsers → Server → Actual RIS

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal

Graphs:

- Good
- Bad
Dynamical Analysis of Discrete Time Systems

\[ y(k + 1) = ay(k) + bu(k) \]

\[ u(k) \rightarrow Server \rightarrow y(k) \]

Z-Transform

\[ y(k + 1) = ay(k) + bu(k) \iff zY(z) = aY(z) + bU(z) \]

Transfer Function (TF)

\[ F(z) = \frac{Y(z)}{U(z)} = \frac{b}{z - a} \iff (ba^0, ba^1, ba^2, \ldots) \]

Pole: Output at time \( k \) is proportional to \( a^k \), for pole \( a \).

Fast systems have small poles

Oscillations result if neg or im poles

Gain: Ratio of steady state output to steady state input
Control Design

\[ F(z) = \frac{Y(z)}{R(z)} = \text{Closed Loop Transfer Function} \]

**Example: Control Law**

\[ u(k) = u(k - 1) + K_I e(k) \]
Key Results From Linear Systems

Stable if $|a|<1$, where $a$ is the largest pole of $G(z)$

$k_s \approx \frac{-4}{\ln |a|}$, where $|a|$ is the largest pole of $G(z)$

Steady state gain of $G(z)$: $\frac{Y(\infty)}{U(\infty)} = G(1)$

Adding signals:

$\{c(k)=a(k)+b(k)\}$ has Z-Transform $A(z)+B(z)$.

Transfer functions in series:

$U(z) \rightarrow G(z) \rightarrow W(z) \rightarrow H(z) \rightarrow Y(z)$ is equivalent to

$U(z) \rightarrow G(z)H(z) \rightarrow Y(z)$

Transfer function of a feedback loop:

$F_R(z) = \frac{T(z)}{R(z)} = \frac{K(z)G(z)}{1 + H(z)K(z)G(z)}$

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal
Optimizing the Microsoft .NET Thread Pool

- Optimization Objective
  - Choose concurrency level that maximizes throughput

- Intuition: Use “hill climbing”

- But specifics (and effectiveness) depend on
  - Curve shape: concave vs. just unimodal?
  - Transients when concurrency level changes
  - Measurement variability

*IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal*
Steady State System Identification Model

Properties

\[ f_a(x^*) = v^* = f_b(x^*) \]

\[ \frac{df_a}{dx} > 0 \text{ if } \alpha > 0 \quad \frac{df_b}{dx} < 0 \text{ if } \beta > 0 \]

\( f_a(f_b) \) is linear if \( \alpha = 1 (\beta = 1) \)

\( f_a(f_b) \) is constant if \( \alpha = 0 (\beta = 0) \)

Concave if \( \alpha, \beta > 1 \)

Parameters: \( x^*, x^{MAX}, v^*, \alpha, \beta \)

\[ \log(1 - \frac{f_a(x)}{v^*}) = \alpha \log(1 - \frac{x}{x^*}) \]

\[ f_a(x) = v^*(1 - (1 - \frac{x}{x^*})^\alpha) \quad f_b(x) = v^*(1 - (1 - \frac{x-x^*}{x^{MAX}-x^*})^\beta) \]

\[ f(x) = f_a(x), x \leq x^*, f_b(x), x \geq x^* \]
Steady State System Identification

\( \alpha = 0.99, \beta = 1.3 \)

\( \alpha = 1 \)

\( \alpha = 1.2, \beta = 0.28 \)

\( \alpha = 2.5, \beta = 0.65 \)

\( \alpha = 1.3, \beta = 0.74 \)

\( \alpha = 1.3, \beta = 0.58 \)

\( \alpha = 1.1 \)

\( \alpha = 1 \)

\( \alpha = 1.6, \beta = 2.1 \)

\( \alpha = 1.4, \beta = 1.6 \)

\( \alpha = 1 \)

\( \alpha = 3.1 \)

\( R^2 = 0.99, x^{MAX} = 58 \)

\( R^2 = 1 \)

\( R^2 = 0.98, x^{MAX} = 2601 \)

\( R^2 = 0.96, x^{MAX} = 89 \)

\( R^2 = 1 \)

\( R^2 = 1 \)

\( R^2 = 0.98, x^{MAX} = 599 \)

\( R^2 = 1 \)

\( R^2 = 0.98, x^{MAX} = 51 \)

\( R^2 = 0.99, x^{MAX} = 30 \)

\( R^2 = 0.91 \)

\( \)
Hill Climbing Using Stochastic Gradient Approximation

Want large gain so move quickly, but not overshoot.

Making good moves depends on:
- throughput variance
- shape of curve

Last #Threads = History mean

IM 2009: Recent Advances in Control Theory.
Abdelzaher, Diao, Hellerstein, Lu, Singhal
Control State Diagram

State 1 - Initializing LastHistory.
LastHistory.Add(data)

State 1a – InTransition.

ChangePointWhileInitializing
IsChangePoint(LastHistory):
  LastHistory = data
  CurrentControlSetting = ExploreMove()

WaitForSteadyState
IsInTransition()

CompletedInitializing
IsStableHistory(LastHistory):
  LastControlSetting = CurrentControlSetting
  CurrentControlSetting = ExploreMove()

ChangePointWhileLookingForMove
Same as ChangePointWhileInitializing

State 2 – Looking for move.
CurrentHistory.Add(data)

State 2a – InTransition
CurrentHistory.Add(data)

DirectedMove
IsSignificantDifference(CurrentHistory, LastHistory):
  LastControlSetting = CurrentControlSetting
  CurrentControlSetting = DirectedMove()
  LastHistory = CurrentHistory
  CurrentHistory = null

StuckInState
IsStableHistory(CurrentHistory) \&
  CurrentHistory.Count > SufficientlyLargeHistory:
  LastControlSetting = CurrentControlSetting
  CurrentControlSetting = ExploreMove()
  LastHistory = CurrentHistory
  CurrentHistory = null

ReverseBadMove
CurrentHistory.Count > MinimumHistory
\& LastHistory.Mean() > CurrentHistory.Mean():
  Swap(CurrentControlSetting, LastControlSetting)

ChangePointInQueueWaiting
IsChangePoint(QueueOfWaiting)
Example of Controller Assessments

<table>
<thead>
<tr>
<th>Controller</th>
<th>throughput</th>
<th>#threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8.2</td>
<td>8.1</td>
</tr>
<tr>
<td>B</td>
<td>18.5</td>
<td>20.6</td>
</tr>
<tr>
<td>C</td>
<td>19.6</td>
<td>22.1</td>
</tr>
<tr>
<td>D</td>
<td>18.5</td>
<td>19.7</td>
</tr>
<tr>
<td>E</td>
<td>7.8</td>
<td>7.7</td>
</tr>
</tbody>
</table>

Challenge: assess tens of thousands of cases
- Use system ID to generate synthetic workloads efficiently as concurrency-throughput curves
- Real controller operates on synthetic workloads
- Can assess optimality of controller since know peak of curve.