Case Study: Resource Provisioning

Joseph L. Hellerstein, Jie Liu
Microsoft

March 10, 2008
Today’s Agenda

• Model Predictive Control (MPC)
• Case Study: Energy –Aware Server Provisioning
• Paper discussions
Open Loop Controller

- If desired closed-loop transfer function is $H(z)$, then $G(z) = H(z)/P(z)$. QED.

\[
y(k+1) = 0.5 y(k) + 0.3 y(k-1) + x(k) \\
y(0) = 0; \ y(1) = 0; \ r(k) = 1 \\
\]

\[
y(2) = x(1) = 1 \Rightarrow x(1) = 1 \\
y(3) = 0.5 y(2) + x(2) = 1 \Rightarrow x(2) = 0.5 \\
y(4) = 0.5 y(3) + 0.3 y(2) + x(3) = 1 \Rightarrow x(3) = 0.2 \\
\]

- Need accurate model of $P(z)$
- Need known initial state
- Cannot have disturbances
MPC Basic Idea

- Predict the output of the plant under all possible controls.
- Pick the best one.
- Check the next output for disturbance.

- As a heuristic, works for nonlinear and empirical models
Predictive Control

\[ y(k + 1) = 0.5 y^2(k) + 0.3 y(k - 1) + x(k) + d(k) \]  \hspace{1cm} \text{Non-linear!}
\[ y(0) = 0; \ y(1) = 0; \ r(k) = 0.5 \]

\[ \hat{y}(1) = 0.5 \Rightarrow x(1) = 0.5 \]
\[ y(1) = 0.45 \]
\[ \hat{y}(2) = 0.5 \Rightarrow x(2) = 0.5 - 0.5 \times (0.45)^2 = 0.39875 \]
\[ y(2) = 0.6 \]

\[ \ldots \]

- What if control output has constraint: \( |x(k)| \leq 0.4 \)

Assume \( d(k)=0 \)

\[ x(1) = 0.4 \Rightarrow y(2) = 0.4 \]
\[ x(2) = 0.4 \Rightarrow y(3) = 0.48 \]
\[ x(3) = 0.2648 \Rightarrow y(4) = 0.5 \]

\[ \ldots \]

Three-step lookahead.
Model Predictive Control

Time = $t_k$

Time = $t_{k+1}$
MPC Formulation

• Predict outputs from inputs: \((p\) steps into the future\)

\[
\begin{bmatrix}
    y(k+1) \\ y(k+2) \\ \vdots \\ y(k+p)
\end{bmatrix}_{k} = F^x \left( \begin{bmatrix}
    y(k-m) \\ y(k-m+1) \\ \vdots \\ y(k)
\end{bmatrix}, \begin{bmatrix}
    u(k-r) \\ u(k-r+1) \\ \vdots \\ u(k-1)
\end{bmatrix} \right) + F^u \left( \begin{bmatrix}
    u(k) \\ u(k+1) \\ \vdots \\ u(k+p-1)
\end{bmatrix} \right)
\]

• Minimize a cost function: e.g. (quadratic form)

\[
J(k) = \sum_{i=1}^{p} (y(k+i)\big|_k - y^*)^T P(y(k+i)\big|_k - y^*) + \sum_{i=0}^{p-1} u(k+i)^T Qu(k+i)
\]

– subject to constraints

• Find the optimal \(u\) and apply \(u(k)\)

• Repeat

There is no control law in the conventional sense!
MPC Results Highlights

- Unconstrained, $\infty$-horizon MPC is equivalent to an optimal control. Its stability can be proved.

- The stability of general MPC is hard to show, since the control law is non-linear and implicitly defined by the optimization.

- Unconstrained quadratic optimization is easy.

- In general, constrained quadratic optimizations have to be solved numerically.

- Hugh amount of application in process control. Standard packages available in Matlab and other industrial control software.
  - 4500 applications reported by 1999.
Power-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services
Energy Consumption of Data Centers

- **US alone in 2006:**
  - 61.4 billion kWh
  - $4.5 billion cost

- **Equivalent to:**
  - %1.5 total US consumption
  - %5 total US households
  - Entire US transportation manufacturing industry

- **Trend:**
  - Doubled since 2000
  - Will double again by 2011

Source: *EPA Report to Congress on Server and Data Center Energy Efficiency, August 2007*
Dynamic Server Provisioning

Idea: turning off unnecessary servers to save energy
Challenges for MSGR Servers

• Two aspects of load
  – $N$: Number of connections
  – $L$: Login rate (per second)
    \[N < 100,000, \quad L < 70/\text{sec}\]

• Dynamic provisioning challenges
  – Turning on: allow slow integral dynamics to fill server capacity
  – Turning off: minimize Server-Initiated Disconnections (SID)
  – Provisioning coupled with load dispatching
Modeling CPU utilization

Performance modeling

\[ CPU \% = 0.00028*N + 0.55*L + 0.82 \]
Supervisory Control Architecture

Load prediction

Provision Controller

\( L_{tot}(t) \)

Dispatcher

\{on, off\}

\{N_i\...\}

\{N_i\...\}

CS

CS

...

CS
Provisioning with Two Constraints

\[ K(t) = \max \left\{ \frac{L_{\text{tot}}(t)}{L_{\text{max}}}, \frac{N_{\text{tot}}(t)}{N_{\text{max}}} \right\} \]

to satisfy:

\[ L_i(t) \leq L_{\text{max}} \]
\[ N_i(t) \leq N_{\text{max}} \]

- Problems with this formula
  - Prediction or estimation errors
  - Slow dynamics: can only fill a server slowly
  - Slow draining process and SID
Provisioning with Safety Margins

\[ K(t) = \max \left\{ \gamma_L \frac{L_{\text{tot}}(t)}{L_{\text{max}}}, \quad \gamma_N \frac{N_{\text{tot}}(t)}{N_{\text{max}}} \right\} \]

\[ \gamma_L = \gamma_L^{\text{frc}} \cdot \gamma_L^{\text{dyn}} \quad \gamma_N = \gamma_N^{\text{frc}} \cdot \gamma_N^{\text{dyn}} \]

• Split into two types of factors (all > 1)
  – frc: Forecasting factors to account for forecasting errors
  – dyn: Dynamics factors to compensate for slow dynamics

• Starving before turning off (sleep) to reduce SID
Short-Term Load Forecasting

- Time series prediction
  - “Seasonal” time series models
  - Parameter estimation
  - Time series prediction

\[
N(t + 1) = \sum_{k=0}^{n} a_k N(t - kT) + \sum_{j=0}^{m} b_j \left( N(t - j) - \frac{1}{n} \sum_{k=1}^{n} N(t - j - kT) \right)
\]
Short-Term Load Forecasting

- Number of Connections
  - Observed value
  - Forecasted value

- Login Rates (per second)
  - Observed value
  - Forecasted value
Load Balancing

• Dynamic system modeling

\[ N_i(t + 1) = N_i(t) + L_i(t) - D_i(t) \]

where

\[ D_i(t) = \beta(t) \cdot N_i(t) \]

\[ L_i(t) = L_{tot}(t) \cdot p_i(t) \]

• Load balancing
  – Idea: make all Ni(t) same
  – Problem: large number of SIDs
System Dynamics

- All servers converge at the same rate.

\[ p_i(t) = \frac{1}{K(t)} + \alpha \left( \frac{1}{K(t)} - \frac{N_i(t)}{N_{tot}(t)} \right) \]

\[ N_i(t+1) = N_i(t) + p_i(t)L_{tot}(t) - \beta_i(t)N_i(t) \]

\[ = (1 - \alpha \frac{L_{tot}(t)}{N_{tot}(t)} - \beta_i(t))N_i(t) + \frac{1+\alpha}{K(t)}L_{tot}(t) \]

- The system is stable if \[ \left| 1 - \alpha \frac{L_{tot}(t)}{N_{tot}(t)} - \beta_i(t) \right| < 1 \]

- The server that gets max load is the newly turned on server:

\[ N_i(t) = 0 \Rightarrow p_i(t) = \frac{1+\alpha}{K(t)} \]

\[ L_i(t) = \frac{1+\alpha}{K(t)} L_{tot}(t) \]

\[ K(t) = (1+\alpha) \frac{L_{tot}(t)}{L_i(t)} \]

\[ \gamma_{L}^{dyn} = 1+\alpha \]
System Dynamics

• **Steady state** \( N_i(t + 1) = N_i(t) \)

• **So,**

\[
N_i(t) = \frac{1 + \alpha}{D_{tot}(t)} + \alpha \frac{N_{tot}(t)}{K(t)} \leq N_{max}
\]

\[
K(t) \geq \frac{1 + \alpha}{D_{tot}(t)} + \alpha \frac{N_{tot}(t)}{N_{max}}
\]

\[
\gamma^\text{dyn}_N = \frac{1 + \alpha}{\min\left(\frac{D_{tot}(t)}{L_{tot}(t)}\right)} + \alpha
\]
Load Skewing

- Route logins to busy servers as long as they can handle
- Actively maintain small number of tail servers
  - Serve as reserve when logins surge
  - Minimize SIDs when turning off
- A simple algorithm:
  - Set $N_{tgt}$ slightly less than $N_{max}$
  - Rank servers with $N_i(t) < N_{tgt}$
  - Evenly distribute logins to the top half
Experimental Setup

- Cluster of 60 connection servers
  \[ N_{\text{max}} = 100,000 \quad L_{\text{max}} = 70/\text{sec} \]
  \[ \text{CPU}\% = 0.00028\times N + 0.55\times L + 0.82 \]
  \[ \text{Power} = \begin{cases} 150 + 0.75\times \text{CPU}\% & \text{if active} \\ 3 & \text{if stand-by} \end{cases} \]

- Provisioning method

\[
K(t) = \max \left\{ \gamma_L^{\text{frc}}, \gamma_N^{\text{frc}} \, \frac{L_{\text{tot}}(t)}{L_{\text{max}}}, \, \gamma_L^{\text{frc}}, \gamma_N^{\text{dyn}} \, \frac{N_{\text{tot}}(t)}{N_{\text{max}}} \right\}
\]
## Load Balancing vs Load Skewing

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Energy (KWH)</th>
<th>Savings</th>
<th>SID</th>
</tr>
</thead>
<tbody>
<tr>
<td>No provisioning + Balancing (NB)</td>
<td>478</td>
<td>---</td>
<td>0</td>
</tr>
<tr>
<td>Forecast + Balancing (FB)</td>
<td>331</td>
<td>30.8%</td>
<td>3,711,680</td>
</tr>
<tr>
<td>Forecast + Balancing + Starving (FBS)</td>
<td>343</td>
<td>28.2%</td>
<td>799,120</td>
</tr>
<tr>
<td>Forecast + Skewing (FS)</td>
<td>367</td>
<td>23.3%</td>
<td>597,520</td>
</tr>
<tr>
<td>Forecast + Skewing + Starving (FSS)</td>
<td>381</td>
<td>20.2%</td>
<td>115,360</td>
</tr>
<tr>
<td>Reactive Load Skewing (RLS)</td>
<td>375</td>
<td>21.5%</td>
<td>48,160</td>
</tr>
</tbody>
</table>

![Graph showing number of active servers over time for different algorithms](image1)

![Graph showing number of active servers over time for different algorithms](image2)
Load Profiles

NB

 FB

 FBS

 FS

 FSS

 RLS
Discussions

• Unique challenges for power-aware provisioning of connection-intensive servers (MSGR)
  – Turning on: slow dynamics to fill to server capacity
  – Turning off: server-initiated-disconnections (SID)

• Unified framework for joint analysis and design of server provisioning and load dispatching

• Load balancing versus load skewing
  – Load balancing: better energy efficiency, high SIDs
  – Load skewing: less energy efficiency, low SIDs