Performance Modelling Techniques: 
Markov Chains and MVA

Rajesh.Mansharamani@gmail.com

September 2013

www.cmgindia.org
Problem Statement: Large Retail

Payload = avg of 11KB per user request (incl ovhds)

$Z = 8$

$N = 5$ per branch

5000 branches

$R_{DC} = 2$

$R_{NW} = 1$

128Kbps available at branch.
How much additional bandwidth required?
Recap: Fundamental Definitions

Response Time = Exit Time – Entry Time

\[ R = \text{average system response time} \]

\[ N = \text{Average Number in the System} \]

\[ \text{Throughput } X = \text{Number of Completions per Unit Time} \]
Recap: Little’s Law

\[ N = X \times R \]

- \( N \): Average Number in the System
- \( R \): Average System Response Time
- \( X \): Throughput (Number of Completions per Unit Time)
Recap: Little’s Law for a Closed System

\[ N = X (R+Z) \]
Average Service Time $S = \text{Average Response Time in Resource outside of queueing/waiting} = \text{Single User Response Time at Resource}
Now Let’s Revert to the Problem Statement

Payload = avg of 11KB per user request (incl ovhds)

$Z = 8$

$N = 5$ per branch

5000 branches

$R_{DC} = 2$

$R_{NW} = 1$

Data Centre

128Kbps available at branch. How much additional bandwidth required?
Now Let’s Revert to the Problem Statement

As far as a given branch is concerned:

\[ Z = 8 \]

\[ N = 5 \text{ per branch} \]

\[ R_{DC} = 2 \]

\[ R_{NW} \text{ should be } \leq 1 \]

\[ S_{NW} = \frac{\text{Payload}}{\text{Bandwidth}} = 11\text{KB}/B = 88\text{Kb}/B \]

If we had \( N = 1 \) user, then \( R_{NW} = S_{NW} = 1 \)

\[ \text{If } S_{NW} = 1, \ B = \frac{88\text{Kb}}{S_{NW}} = 88\text{Kbps} \]
Naive Approach Taken By Vendor

As far as a given branch is concerned:

\[ Z = 8 \]

\[ N = 5 \text{ per branch} \]

\[ B = 88\text{Kbps for one user,} \]

Therefore for \( N = 5 \) users, we need \( 88 \times 5 = 440\text{Kbps} \)

Round to 512Kbps

\[ R_{NW} \text{ should be} \leq 1 \]

\[ R_{DC} = 2 \]

Cost of upgrade is just $2000 per year from 128Kbps to 512Kbps, so what’s the problem?
Naive Approach Taken By Vendor

As far as a given branch is concerned:

\[ Z = 8 \]

\[ N = 5 \text{ per branch} \]

\[ R_{DC} = 2 \]

\[ R_{NW} \text{ should be } \leq 1 \]

\[ B = 88 \text{Kbps for one user,} \]

\[ \text{Therefore for } N = 5 \text{ users, we need } 88 \times 5 = 440 \text{Kbps} \]

\[ \text{Round to 512Kbps} \]

Cost of upgrade is just $2000 per year from 128Kbps to 512Kbps, so what’s the problem?

The problem is that we have 5000 branches and that will cost $2000 \times 5000 = $10M !!!
At Design Time We Cannot Test: We Need Models

\[ Z = 8 \]

\[ N = 5 \text{ per branch} \]

\[ R_{NW} \text{ should be} \leq 1 \]

\[ R_{DC} = 2 \]
At Design Time We Cannot Test: We Need Models

\[ Z = 8 \]

\[ N = 5 \text{ per branch} \]

\[ R_{DC} = 2 \]

\[ Z = 10 \]

\[ N = 5 \text{ per branch} \]
Bandwidth Model Solution Approach

\[ Z = 10 \]
\[ N = 5 \text{ per branch} \]
\[ S = 88\text{Kb/B} \]

\[ R_{NW} = f(B) \]

For a given \( B \) we can determine \( R \). How to determine \( B \) for a given \( R \)?
**Bandwidth Model Solution Approach**

- **N** = 5 per branch
- **Z** = 10
- **S** = 88Kb/B

Lower bound on **B** is 88Kbps
Upper bound on **B** is 5x88Kbps = 440Kbps

Apply binary search on **B** and keep iterating until **R(B) = 1**

\[ R = f(B) \]

\[ 88 \rightarrow B \rightarrow 440 \]

\[ R = 1 \]
Bandwidth Model Solution Approach

For a given $B$, determine $R$

Determine lower and upper bounds of $B$

Using binary search approach converge to $B$ such that $R(B) = 1$

So now need to determine $R$ for a given $B$
Markov Chains

**Step 1:** Define System State

**Step 2:** Create State Transition Diagram

**Step 3:** Augment with Rates

**Step 4:** Solve
Markov Chains: Step 1 – Define System State

System State = (Number of Users in Think State, Number of Users in Network)

Can this be made simpler?

System State = (Number of Users in Network)

since Number of Users in Think State = 5 – Number of Users in Network
Markov Chains: Step 2 – State Transition Diagram

System State = (Number of Users in Network)

Assuming no two arrivals to queue at the same time and no two departures from queue at same time.
Markov Chains: Step 3 – Augment with Rates

Let $\lambda = 1/Z = \text{‘think’ rate}$
Let $\mu = 1/S = \text{service rate}$

What is the rate of transitioning from State 0 to State 1?

0 in network queue = how many thinking?

What is the rate of transitioning from State 1 to State 0?

$\mu$
Markov Chains: Step 3 – Augment with Rates

Let $\lambda = 1/Z = \text{‘think’ rate}$

Let $\mu = 1/S = \text{service rate}$
Markov Chains: Step 3 – Augment with Rates

Let $\lambda = 1/Z = \text{‘think’ rate}$
Let $\mu = 1/S = \text{service rate}$

$N=5$ per branch
Markov Chains: Step 4 – Solve

What does solve mean?

Get steady state probability of being in each state by means of flow balance equations
Markov Chains: Step 4 – Flow Balance Equations

Rate of Arrival in to each State = Rate of Departure from each State

Rate of Arrival in to State 0 = \( \mu p_1 \)

Rate of Departure from State 0 = \( 5\lambda p_0 \)

\[ 5\lambda p_0 = \mu p_1 \]
Markov Chains: Step 4 – Flow Balance Equations

Rate of Arrival in to each State = Rate of Departure from each State

Rate of Arrival in to State 1 = ?  
\[ 5\lambda \ p_0 + \mu \ p_2 \]

Rate of Departure from State 1 = ?  
\[ \mu \ p_1 + 4\lambda \ p_1 \]

\[ \therefore \ 5\lambda \ p_0 + \mu \ p_2 = \mu \ p_1 + 4\lambda \ p_1 \]
Rate of Arrival in to each State =
Rate of Departure from each State

In this way we get 6 equations in 6 unknowns, but one of them will be redundant.

We therefore add the equation:
\[ p_0 + p_1 + p_2 + p_3 + p_4 + p_5 = 1 \] (since all are probabilities)
Rate of Arrival in to each State = Rate of Departure from each State

We can now solve these simultaneous equations, using any package such as MATLAB.

Or we can simply use a Markov Chain solver.
Markov Chains: Step 4 – Inputs

\[ Z = 10 \Rightarrow \lambda = 1/Z = 0.1 \]

\[ S = 88\text{Kb}/B, \text{ let’s start with } B=88\text{Kbps or } S = 1 \]

\[ \Rightarrow \mu = 1/S = 1 \]
Markov Chains: Step 4 – ORSTAT MCQueue Solver

<table>
<thead>
<tr>
<th>Transition Rate Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of states (1 to 100): 6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transition Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>
Markov Chains: Step 4 – ORSTAT MCQueue Solver

Choice of output:
- Chain Structure
- Steady-state behaviour
- Transient behaviour
- First-passage Time Probabilities

Run
 Continuous Time Markov Chains

-----------------------------
STEADY-STATE BEHAVIOUR
-----------------------------

The solution is:

State 1: \( p[1] = 0.5639521769 \)
State 2: \( p[2] = 0.2819760884 \)
State 3: \( p[3] = 0.1127904354 \)
State 4: \( p[4] = 0.0338371306 \)
State 5: \( p[5] = 0.0067674261 \)
State 6: \( p[6] = 0.0006767426 \)
We have solved for the steady state probabilities $p_0$ to $p_5$.

**But how does that help us determine avg response time in the network $R_{NW}$?**

- $Z = 10$
- $N = 5$ per branch
- $S = 1$
- $B = 88$Kbps
Markov Chains: Step 4 – ORSTAT MCQueue Solver

We have solved for the steady state probabilities $p_0$ to $p_5$

$Z = 10$

$N=5$ per branch

S = 1
B = 88Kbps

**Hint:** Using probabilities determine average number in the network.

Since state space = number in network.
Markov Chains: Step 4 – ORSTAT MCQueue Solver

We have solved for the steady state probabilities $p_0$ to $p_5$

Avg number in network
$N_{Nw} = \sum (k p_k) \quad \{k=0,...,5\}$

Using the probability values obtained through MCQueue we get

$N_{Nw} = 0.64$
We have average number in network as
\[ N_{\text{NW}} = 0.64 \]

How do we arrive at avg response time in network \( R_{\text{NW}} \)?

Average number of users in think state \( N_Z = ? \)
\[ N - N_{\text{NW}} = 5 - 0.64 = 4.36 \]

Avg response time at user terminal = ?
\( Z = 10 \)

Throughput at user terminals = ?
\[ X = \frac{N_Z}{Z} \text{ by Little's Law} \]
\[ = \frac{4.36}{10} = 0.436 \]
Markov Chains: Step 4 – ORSTAT MCQueue Solver

We have average number in network as
\[ N_{NW} = 0.64 \]

And throughput = 0.436

Avg Network Response Time \( R_{NW} = \) ?

\[ \frac{N_{NW}}{X_{NW}} \]

\[ = \frac{0.64}{0.436} \]

\[ = 1.467 \text{ sec} > 1 \text{ sec} \]
We repeat the same process for $B = 128$Kbps to get

\begin{align*}
\text{a) } R_{NW} &< 1 & R_{NW} &= 0.89988 \\
\text{b) } R_{NW} &= 1 & \text{No need for client to increase bandwidth at all !!!} \\
\text{c) } R_{NW} &> 1 \\
\end{align*}

Recall vendor had proposed $B = 512$Kbps to meet response time requirement at $10$M.
Mean Value Analysis

Since we are dealing with queues is there is a simpler way to arrive at the same answer?

Use Mean Value Analysis (MVA) from Queueing Theory
Mean Value Analysis

N=5 per branch

Z = 10

S = 88Kb/B
B = 128Kbps

Average response time at queue when there are N users in the system

R(N) = S + Average Waiting Time
Mean Value Analysis

We have now recursively expressed \( R(N) \) in terms of \( R(N-1) \) with the termination condition as \( R(1) = S \)

\[
R(N) = S + Q(N-1) S
\]

\( Z = 10 \)

\( N = 5 \) per branch

\( S = 88\text{Kb/B} \)

\( B = 128\text{Kbps} \)
Mean Value Analysis: Solver Code

N = 5 per branch

Z = 10

S = 88Kb/B
B = 128Kbps

\[ R(N) = S + Q(N-1) \times S \]
\[ Q(N-1) = X(N-1) \times R(N-1) \]
\[ X(N-1) = \frac{N-1}{[R(N-1) + Z]} \]

Same as Markov Chain Analysis result

<table>
<thead>
<tr>
<th>i</th>
<th>R</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6875</td>
<td>0.9356</td>
</tr>
<tr>
<td>2</td>
<td>0.7317</td>
<td>0.1864</td>
</tr>
<tr>
<td>3</td>
<td>0.7813</td>
<td>0.2783</td>
</tr>
<tr>
<td>4</td>
<td>0.8370</td>
<td>0.3691</td>
</tr>
<tr>
<td>5</td>
<td>0.8999</td>
<td>0.4587</td>
</tr>
</tbody>
</table>
Concluding Remarks: Markov Chains

• Why is a Markov Chain called a Markov Chain?

Because of the Markov Property:
Transition to next state is memoryless: it depends only on the current state and not on the past history
Concluding Remarks: MVA

- Extends to multiple queues

- Exact solution works for *Product Form Queueing Networks*
  - Memoryless distribution for FCFS, general distribution for processor sharing or delay centres

- Extends to multiple classes as well
  - For example out of N=5, we have 2 users with payload 5KB and three with payload 15KB

- State space can explode with multiple classes, but a lot of approximation techniques are available
Concluding Remarks: Other Modelling Techniques

- Petrinets – higher level abstraction that translates to Markov Chains
- Discrete Event Simulation – see next talk
- Measurement Based Modelling – prototyping and extrapolation
References

- Kishor Trivedi: *Probability & Statistics ... popular book*
- Lazowska et al: *Quantitative System Performance ... popular book*
- MCQueue: [http://staff.feweb.vu.nl/tijms/](http://staff.feweb.vu.nl/tijms/)
- Introduction to Performance Modelling, Little’s Law etc: [www.softwareperformanceengineering.com](http://www.softwareperformanceengineering.com)