THE ROLE OF PERFORMANCE TOOLS IN MODELING COMPLEX SYSTEMS

C.S. Kim, J. Sztrik *

* Faculty of Informatics, University of Debrecen, Hungary

e-mail: jsztrik@inf.unideb.hu
http://irh.inf.unideb.hu/user/jsztrik/
OUTLINE

• Modeling tools
• Finite-source retrial queueing systems
• MOSEL tool
• Case studies
• References
Modeling tools

- University of Dortmund: \textit{HIT, MACOM, HiQPN, DSPNexpress}
  
  \url{http://ls4-www.informatik.uni-dortmund.de/tools.html}

- University of Erlangen: \textit{PEPSY, MOSEL}
  
  \url{http://www4.informatik.uni-erlangen.de/Projects/MOSEL/}

- Duke University: \textit{SHARPE, SPNP, iSPN, RAFT}
  
  \url{http://www.ee.duke.edu/ kst/tools.html}

- University of Illinois at Urbana-Champaign: \textit{MÖBIUS}
  
  \url{http://www.mobius.uiuc.edu/}
Finite-source retrial queueing system
Mathematical model

The system state at time $t$ can be described with the process

$$X(t) = (Y(t); C(t); N(t))$$

where $Y(t) = 0$ if the server is up, $Y(t) = 1$ if the server is failed,

$C(t) = 0$ if the server is idle, $C(t) = 1$ if the server is busy,

$N(t)$ is the number of sources of repeated calls at time $t$. 
We define the stationary probabilities:

\[ P(q; r; j) = \lim_{t \to \infty} P(Y(t) = q, C(t) = r, N(t) = j) \]

\[ q = 0, 1, \quad r = 0, 1, \quad j = 0, \ldots, K^*, \]

where \( K^* = \begin{cases} 
K - 1 & \text{for blocked case,} \\
K - r & \text{for unblocked case.} 
\end{cases} \)

Once we have obtained these limiting probabilities the **main system performance measures** can be derived in the following way.
1. Utilization of the server

\[ U_S = \sum_{j=0}^{K-1} P(0, 1, j) \]

2. Utilization of the repairman

\[ U_R = \sum_{q=0}^{1} \sum_{j=0}^{K^*} P(1, q, j) \]

3. Availability of the server

\[ A_S = \sum_{q=0}^{1} \sum_{j=0}^{K^*} P(0, q, j) = 1 - U_R \]
4. The mean number of calls staying in the orbit or in service

\[ M = E[N(t) + C(t)] = \sum_{q=0}^{1} \sum_{r=0}^{1} \sum_{j=0}^{K^*} jP(q, r, j) + \sum_{q=0}^{1} \sum_{j=0}^{K-1} P(q, 1, j). \]

5. Utilization of the sources

\[ U_{SO} = \begin{cases} \frac{E[K-C(t)-N(t);Y(t)=0]}{K} & \text{for blocked case,} \\ \frac{K-M}{K} & \text{for unblocked case.} \end{cases} \]

6. Overall utilization

\[ U_O = U_S + KU_{SO} + U_R. \]
7. The mean rate of generation of primary calls

\[ \overline{\lambda} = \begin{cases} \lambda E[K - C(t) - N(t); Y(t) = 0] & \text{for blocked case,} \\ \lambda E[K - C(t) - N(t)] & \text{for unblocked case.} \end{cases} \]

8. The mean response time

\[ E[T] = \frac{M}{\overline{\lambda}} \]

9. The blocking probability of a primary call

\[ B = \begin{cases} \frac{\lambda E[K - C(t) - N(t); Y(t) = 0; C(t) = 1]}{\overline{\lambda}} & \text{for blocked case,} \\ \frac{\lambda E[K - C(t) - N(t); C(t) = 1]}{\overline{\lambda}} & \text{for unblocked case.} \end{cases} \]
Evaluation Tool MOSEL

MOSEL (Modeling, Specification and Evaluation Language) developed at the University of Erlangen, Germany, is used to formulate and solved the problem.

Case studies
## Input system parameters

<table>
<thead>
<tr>
<th></th>
<th>K</th>
<th>λ</th>
<th>μ</th>
<th>ν</th>
<th>δ, γ</th>
<th>τ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>6</td>
<td>0.8</td>
<td>4</td>
<td>0.5</td>
<td>x axis</td>
<td>0.1</td>
</tr>
<tr>
<td>Figure 2</td>
<td>6</td>
<td>0.1</td>
<td>0.5</td>
<td>0.5</td>
<td>x axis</td>
<td>0.1</td>
</tr>
<tr>
<td>Figure 3</td>
<td>6</td>
<td>0.1</td>
<td>0.5</td>
<td>0.05</td>
<td>x axis</td>
<td>0.1</td>
</tr>
<tr>
<td>Figure 4</td>
<td>6</td>
<td>0.8</td>
<td>4</td>
<td>0.5</td>
<td>0.05</td>
<td>x axis</td>
</tr>
<tr>
<td>Figure 5</td>
<td>6</td>
<td>0.05</td>
<td>0.3</td>
<td>0.2</td>
<td>0.05</td>
<td>x axis</td>
</tr>
<tr>
<td>Figure 6</td>
<td>6</td>
<td>0.1</td>
<td>0.5</td>
<td>0.05</td>
<td>0.05</td>
<td>x axis</td>
</tr>
</tbody>
</table>
Mean response time depending on server's failure rate

$E[T]$ versus server’s failure rate
Overall utilization depending on server’s failure rate

Legend
- ■ reliable server
- ▲ non-reliable server (continuous)
- ○ non-reliable server (non-continuous)
- △ non-reliable server (continuous, intelligent)
- ● non-reliable server (non-continuous, intelligent)

$U_O$ versus server’s failure rate
Mean number of requests staying in the orbit or in service depending on server’s failure rate

Legend
- reliable server
- non-reliable server (continuous)
- non-reliable server (non-continuous)
- non-reliable server (continuous, intelligent)
- non-reliable server (non-continuous, intelligent)

$M$ versus server’s failure rate
Mean response time depending on server's repair rate

Legend
- ■ reliable server
- △ non-reliable server (continuous)
- ○ non-reliable server (non-continuous)
- ▲ non-reliable server (continuous, intelligent)
- ● non-reliable server (non-continuous, intelligent)

Mean response time

$E[T]$ versus server’s repair rate
Overall utilization depending on server's repair rate

$U_O$ versus server’s repair rate
Mean number of requests staying in the orbit or in service depending on server’s repair rate

$M$ versus server’s repair rate


References


