Performance Modelling Using Discrete Event Simulation

Benny Mathew

Tata Consultancy Services
benny1.m@tcs.com

September 2013

www.cmgindia.org
Agenda

Introduction to Discrete Event Simulation

Simulation of IT Systems

Simulation of Call Centres
Performance Modelling Techniques

Analytical Modelling
- Abstract the system into network of queues
- Establish queuing relationships using mathematical expressions
- Solution based on
  - Operational laws
  - Queuing theory
  - Markov Chains
  - Stochastic Petri nets

Simulation Modelling
- Abstract the system into network of resources
- Relationships and interaction between resources modelled using “code”
- Interaction between model in a simulation environment used to get the desired data
### Performance Modelling Techniques

<table>
<thead>
<tr>
<th></th>
<th>Measurements</th>
<th>Analytical Modelling</th>
<th>Simulation Modelling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flexibility</strong></td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
</tr>
</tbody>
</table>

* These are for typical scenarios
Discrete Event Simulation

- Entire system to be modeled is broken down into resources
- Each resource can have highly complex behavior using internal events
- Each resource is a usually a C++/Java class
Applications of Discrete Event Simulation

Industrial
• Production Lines and Flow

Logistics
• Maritime
• Hospital (Beds, Medicines, Ambulances)

Staff Scheduling
• Call Centres, Hospitals

Business Processes
• Vendor Management /Outsourcing
• Workflows (Service Request, Appraisals)
• Recruitment
DES for Sizing/Design Decisions
Sizing and Capacity Planning Methods

Sizing – RFP/Design Stage

• Based on complexity vis-à-vis standard benchmark
• Thumb Rules – example 1 CPU can support 100 web users

Capacity Planning – Testing/Production Stage

• Extrapolation / Regression – range from transaction agnostic to more complex transaction aware
• Queuing Network model – Build queuing network model from transaction-wise demand. Can range from simple Markov model to more complex Petri-Nets or Colored Petri Nets
• Discrete Event Simulation : Build discrete event simulation model from transaction-wise demand
Problem Statement

Sizing
• Large factor of safety has to be kept when using relative complexity / thumb rules

Can we use more sophisticated techniques
• Improve sizing accuracy at RFP stage
• Influence design decisions at RFP/Design stage
Approach

Create Building Block Library
- Demand of CPU/memory/disk/network

Build Simulation Model

Run the Simulation
- Define workload
- Define transactions with respect to building blocks
- Simulate/Do what-if
Creating Building Block Library: Steps

1. Take existing Applications
2. Break it down into its building blocks
3. Profile each building block
4. Add to Library
Creating Building Block Library: Profiling

Complexity of Profiling

\[ \text{Demand} = f \left\{ \begin{array}{l}
\text{block complexity} \\
\text{inherent variations} \\
\text{server specifications} \\
\text{concurrency}
\end{array} \right\} \]
Building Simulation Model – Overall Resources and Flow

- Workload Transaction 1
- Workload Transaction 2
- Workload Transaction n

Load Balancer

Web/App Server 1

Database Server

Web/App Server 1
Building Simulation Model – Application Server

Requests From Load Balancer → Network → Application Threads → CPU → Reply to Load Balancer

To Database Server → Disk → Memory → From Database Server
Simulating the IT System

- Workload is set of transactions with certain mix and think time
- Each transaction is defined as **series of building blocks**

**Define Workload**

**Set Parameters**
- Block complexity
- Server Factor
- Run Duration

**Run Simulation**
- Reports Response time, Throughput and system utilization

Computer Measurement Group, India
Simulating a Online Store : Throughput

Test Details
No. of users : 10 – 250

Transaction Mix
1. Transaction 1 (50%)
2. Transaction 2 (30%)
3. Transaction 3 (20%)

<table>
<thead>
<tr>
<th>#users</th>
<th>Measured X</th>
<th>Sim X</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>50.00</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>150.00</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>200.00</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>250.00</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>300.00</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>350.00</td>
<td></td>
</tr>
</tbody>
</table>
Simulating a Online Store: Response Time

![Graph showing response time](image)

![Graph showing response time](image)

![Graph showing response time](image)

![Graph showing response time](image)
DES for Call Centre Manpower Planning
Call Centres End to End Picture

Telephone Network (PSTN)

Customers

Trunk Lines

PABX

IVR

ACD

Skill Based Queues

Agents
Modelling Call Centres

Generator (Customers) → Server (PABX + IVR) → Router (ACD) → Priority Queue (SBQ) → Server (Agent Teams) → Sink (Successful Calls)

Sink (Lost Calls) → Server (Agent Teams) → Priority Queue (SBQ) → Server (Agent Teams) → Sink (Abandoned Calls)
Advantages of Simulating Call Centres

More accurate handle times
• Use of distributions, Efficiencies of agents

More detailed Call Flow
• Abandonments and Lost Calls, Call allocation strategy, Misdirected Calls, After call work

More detailed schedule
• Breaks, Meeting and Training

What-if Analysis
• Changing Employee Efficiency
• Staggered breaks
• More cross-trained employees
• Dedicated agents for high-priority customers
• Call-allocation strategy
Simulation Software

General Purpose Freeware
• OMNet++, SimPY

General Purpose Commercial
• OMNest, SimPY, SimEvents, SAS Simulation Studio, Anylogic

IT Simulation
• HyPerformix, OPNET, NS3 (network)

Business Process
• IBM Websphere Lombardi, L-Sim, ARIS

Industrial
• ProModel
Q and A
APPENDIX
Why Model?

Measurements / Prototyping
- May require costly hardware
- Time consuming
- Data collection may disturb normal behavior
- Available late in the software development cycle

Relying on intuition can be costly

Arrival Rate $\lambda = 3 /s$
CPU Speed = $x$ MHz
Response Time = 0.083 sec

Arrival Rate $\lambda = 6 /s$
CPU Speed = $x$ MHz
Response Time = 0.083 sec

New CPU Speed = ?

Should CPU speed also Double?

Ans: Only 20% faster
Analytical or Simulation

Analytical:
• Best for steady state analysis
• Cheaper to build
• Usually faster to solve
• Solution time does not depend on rare events

Simulation:
• Best when there is time varying behaviour
• More expensive to build and takes time to solve
• Solution time can be very high if there are rare events
• Move from Analytical to Simulation when
  • The model is very complex with many variables and interacting components
  • The underlying variables relationships are nonlinear
  • The model contains random distributions
Which Simulation Technique?

System dynamics (SD):
- Aggregated view of a system and is better suited to higher-level modelling
- Focus on flows and not on individual behaviour of entities (model stocks, flows and delays)

Discrete-event simulation (DES):
- Detailed modelling of systems
- Random events and mechanisms play an important part.

Agent Based Simulation (ABS):
- Autonomous (self-directed) agents which follow a series of predefined rules and interact with each other and their environment.
- Eg: Flocking behaviour in birds, spread of cancer cells
DES Terminology

**Resource**

- **Client**
  - Log_on: ...
  - Send_request: ...
  - Receive_reply ...

**Entity**

- **Network**
  - Request for transmit: ...
  - Transmit over: ...

- **Server**
  - Receive Req: ...
  - Send_reply: ...

**Events and Ports**
Modelling CPU in DES

Parameters:
- Number of CPUs
- CPU Speed

Event handlers:

- `Arrival(entity)`
  ```
  { 
  Calculate departure time
  Schedule_departure(time, entity)
  }
  ```

- `Departure(entity)`
  ```
  { 
  send_port(exit, entity);
  }
  ```
Modelling CPU in DES

• Consider that at time 2, a job arrives requiring 5 seconds of CPU time

• So we schedule exit of the job at time = 7

• Another job which requires 8 seconds of CPU time arrives at time=5

• At time=5, residual time for job1 is 2 and for job2 is 8, so reschedule job1 to depart at time = 9

• At time=9, residual time for job2 is 6, so schedule job2 to depart at time =15 (T + 6)

* Calculation of departure time = T + residual_time x num of concurrent jobs
Call Centre Modelling – What-if Examples

![Call Allocation Strategy Chart]

- Wait Time - 80 Percentile (seconds)

- Impact of Distribution:
  - exp(180)
  - logn(5.0679,0.5)
  - logn(4.4117,1.25)
Call Centre Modelling – Success Stories

Bank in Middle East
1. Reduce wait times with existing number of agents
2. Achieve same performance with reduced number of agents

Simulation Results
1. Changed schedule reduced 90 percentile wait times to less than a second from 12 seconds
2. Even after reducing agents from 30 to 25, the 90 percentile was still below 8 seconds