Simulator for Performance Evaluation of Distributed Software Systems

Bhavesh A. Kotak 03305052
Guide: Prof. Varsha Apte
Dept. of Computer Science
IIT Bombay

25th February 2005
Outline

1. **Introduction and Motivation**
   - Introduction
   - Motivation for Performance Analysis tool

2. **Approach to tool building**
   - Stage I
   - Stage II

3. **Simulator Implemented**
   - Simulator Features
   - Simulator’s Output

4. **Design Example**
   - Overview
   - Message Flow Diagrams

5. **Issues or limitations in current implementation**

6. **Future Work and Conclusion**
   - Future Work
Outline

1. Introduction and Motivation
   - Introduction
   - Motivation for Performance Analysis tool

2. Approach to tool building
   - Stage I
   - Stage II

3. Simulator Implemented
   - Simulator Features
   - Simulator’s Output

4. Design Example
   - Overview
   - Message Flow Diagrams

5. Issues or limitations in current implementation

6. Future Work and Conclusion
Outline

1 Introduction and Motivation
   - Introduction
   - Motivation for Performance Analysis tool

2 Approach to tool building
   - Stage I
   - Stage II

3 Simulator Implemented
   - Simulator Features
   - Simulator’s Output

4 Design Example
   - Overview
   - Message Flow Diagrams

5 Issues or limitations in current implementation

6 Future Work and Conclusion
   - Future Work
Outline

1. Introduction and Motivation
   - Introduction
   - Motivation for Performance Analysis tool

2. Approach to tool building
   - Stage I
   - Stage II

3. Simulator Implemented
   - Simulator Features
   - Simulator’s Output

4. Design Example
   - Overview
   - Message Flow Diagrams

5. Issues or limitations in current implementation

6. Future Work and Conclusion
Outline

1. Introduction and Motivation
   - Introduction
   - Motivation for Performance Analysis tool

2. Approach to tool building
   - Stage I
   - Stage II

3. Simulator Implemented
   - Simulator Features
   - Simulator’s Output

4. Design Example
   - Overview
   - Message Flow Diagrams

5. Issues or limitations in current implementation

6. Future Work and Conclusion
Outline

1. Introduction and Motivation
   - Introduction
   - Motivation for Performance Analysis tool

2. Approach to tool building
   - Stage I
   - Stage II

3. Simulator Implemented
   - Simulator Features
   - Simulator’s Output

4. Design Example
   - Overview
   - Message Flow Diagrams

5. Issues or limitations in current implementation

6. Future Work and Conclusion
   - Future Work
Introduction

Title

“To develop a tool for performance analysis of distributed software systems.”

A distributed software system design will be evaluated to check for its performance capabilities. This would be done before deploying the system. The tool’s output would help to predict the performance that can be achieved by the system, thus rendering it either fit or unfit for deployment...
Designer’s questions that motivated the simulator

- What external arrival rate can my system support?
- What is the bottleneck resource?
- Should one more physical server be introduced?
- Should the software server be moved to another physical server?
- Will a couple of more CPUs or Disks do the trick?

A *performance analysis tool is needed that can easily capture such ‘what-if’ questions and provide relevant performance measures*
Stage I work done

Literature Survey

- ‘Software Architecture (SA)’ specification mechanisms viz. Message Sequence Charts (MSC), Use Cases, Use Case Maps (UCM)
- ‘Performance Models (PM)’ viz. Markov Chains, Petri Nets, Queuing Networks and their variants
- Conversion mechanism from SAs to PMs
Focus of Stage II work

- Implementation of a simulation tool
- *Why?*
  - A simple analytical model already implemented here
  - The model is being extended to cover more design features
  - Results of the extended analytical model need verification
- Simulator can help to verify the extended analytical solver
- Can be used independently as a performance evaluator

---

* at Networks Group - CSE, IIT Bombay by Ms. Rukma P. Verlekar
Simulator Features

**Model**
- SA description: Similar to Message Sequence Charts
- Performance Model: Layered Queuing Network

**Design features evaluated**
- Open queuing network
- Multiple callflows
- Deterministic/Probabilistic sequence
Simulator Features

Model
- SA description: Similar to Message Sequence Charts
- Performance Model: Layered Queuing Network

Design features evaluated
- Open queuing network
- Multiple callflows
- Deterministic/Probabilistic sequence
Design features evaluated...

- Asynchronous calls, Synchronous calls
- Software servers modeled as queuing entities
- Multiple s/w servers with multiple s/w services
- Software servers mapped to Physical servers
- Multiple physical servers with multiple resources
Simulator Features...

- Physical resources also modeled as queuing entities
- Finite/Infinite queue capacity support on each queuing entity
- Finite/Infinite number of threads on each s/w server
- Various scheduling policies supported at the resources viz. FCFS, Round Robin, Priority-Based and SJF
- Network channel delay and Mean Transfer Unit restrictions are captured
Simulator’s Output

<table>
<thead>
<tr>
<th>Response Times for</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CallFlows</td>
</tr>
<tr>
<td>2. Software servers</td>
</tr>
<tr>
<td>3. Software services</td>
</tr>
<tr>
<td>4. Physical servers</td>
</tr>
<tr>
<td>5. Physical resources</td>
</tr>
<tr>
<td>6. Individual software queues on physical servers</td>
</tr>
</tbody>
</table>
Waiting Times for

1. Software servers
2. Physical servers
3. Physical resources
4. Individual software queues on physical servers
Simulator’s Output...

Average Queue Lengths for

1. Software servers
2. Physical servers
3. Physical resources
4. Individual software queues on physical servers
Simulator’s Output...

Throughputs of:

1. Software servers
2. Physical servers
3. Physical resources
4. Individual software queues on physical servers
Simulator’s Output...

**Utilizations of**

1. Software servers
2. Physical servers
3. Physical resources
4. Individual software queues on physical servers

- Helps to identify the bottleneck
- Effect of different arrival rates on a given software system
Web-based E-mail System

Overview
Message Flow Diagrams

Browser → WEB → AUTH → IMAP → SMTP

User Actions:
1. Login
2. Read message
3. Send message
4. List message
5. Delete message
**Designer’s questions**

- What arrival rate can my system support?
- What throughput will be achieved?
- What will be the utilization of my physical servers?
- What is the bottleneck resource, if any?

*To answer these, we need to know the message flow of each transaction, their arrival rates, deployment of software and physical servers, resources used, service rates of these resources, and other things...*
Login Message Flow Diagram

Browser

User_id/passwd

Web

Send_to_auth

Authentication

Not_OK

Verify_passwd

OK

GenerateHtml

IMAP

0.1

SMTP

0.9

Send_to_imap

ListMsg

MessagesList

GenerateHtml
Read Message Flow Diagram
Send Message Flow Diagram

Overview
Message Flow Diagrams

Send Message Flow Diagram

Browser  Web  Authentication  IMAP  SMTP

- Session_id
- Send_to_auth
- NOK
- Change_to_html

- OK
- 0.8
- Message

Send_to_smtp

- List_msg
- Change_to_html

Send_msg
List Message Flow Diagram

List Messages

<table>
<thead>
<tr>
<th>Browser</th>
<th>Web</th>
<th>Authentication</th>
<th>IMAP</th>
<th>SMTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session_id</td>
<td></td>
<td>Verify_session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send_to_web</td>
<td>NOK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change_to_html</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send_to_imap</td>
<td>OK</td>
<td></td>
<td></td>
<td>List_message</td>
</tr>
<tr>
<td>Change_to_html</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Delete Message Flow Diagram

Delete MFD

<table>
<thead>
<tr>
<th>Browser</th>
<th>Web</th>
<th>Authentication</th>
<th>IMAP</th>
<th>SMTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session_id</td>
<td>Send_to_auth</td>
<td>Verify_session</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOK</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OK</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send_to_imap</td>
<td>Delete_msgg</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GenerateHtml</td>
<td>Read_msgg</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GenerateHtml</td>
<td>List_msgg</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bhavesh A. Kotak
Performance Evaluation of Distributed Software Systems
### Overview

**Deployment and Configuration Scheme I**

#### Physical Servers

<table>
<thead>
<tr>
<th>Name</th>
<th>Quantity</th>
<th>CPU</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUN</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>HP</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

#### Software Servers

<table>
<thead>
<tr>
<th>Name</th>
<th>Threads</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB1</td>
<td>4</td>
<td>send_to_auth, send_to_imap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>send_to_smtp, change_to_html</td>
</tr>
<tr>
<td>AUTH1</td>
<td>2</td>
<td>verify_passwd, verify_session</td>
</tr>
<tr>
<td>IMAP1</td>
<td>2</td>
<td>read_messg, list_messg, delete_messg</td>
</tr>
<tr>
<td>SMTP1</td>
<td>2</td>
<td>send_messg</td>
</tr>
</tbody>
</table>
Deployment and Configuration Scheme I

### Physical Servers

<table>
<thead>
<tr>
<th>Name</th>
<th>Quantity</th>
<th>CPU</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUN</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>HP</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

### Software Servers

<table>
<thead>
<tr>
<th>Name</th>
<th>Threads</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB1</td>
<td>4</td>
<td>send_to_auth, send_to_imap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>send_to_smtp, change_to_html</td>
</tr>
<tr>
<td>AUTH1</td>
<td>2</td>
<td>verify_passwd, verify_session</td>
</tr>
<tr>
<td>IMAP1</td>
<td>2</td>
<td>read_messg, list_messg, delete_messg</td>
</tr>
<tr>
<td>SMTP1</td>
<td>2</td>
<td>send_messg</td>
</tr>
</tbody>
</table>
## Mean Service Times

<table>
<thead>
<tr>
<th>Component</th>
<th>Service Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB1</td>
<td>{0.001, 0.007}, {0.00098, 0.007},</td>
</tr>
<tr>
<td></td>
<td>{0.002, 0.008}, {0.001, 0.008}</td>
</tr>
<tr>
<td>AUTH1</td>
<td>{0.0072, 0.006}, {0.001, 0.008}</td>
</tr>
<tr>
<td>IMAP1</td>
<td>{0.007, 0.003}, {0.0002, 0.007}, {0.004, 0.006}</td>
</tr>
<tr>
<td>SMTP1</td>
<td>{0.001, 0.007}</td>
</tr>
</tbody>
</table>

## Allocation Table

<table>
<thead>
<tr>
<th></th>
<th>SUN_1</th>
<th>SUN_2</th>
<th>HP_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AUTH</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IMAP</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SMTP</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Deployment and Configuration Scheme I

### Mean Service Times

<table>
<thead>
<tr>
<th>Service</th>
<th>Mean Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB1</td>
<td>{0.001, 0.007}, {0.00098, 0.007}, {0.002, 0.008}, {0.001, 0.008}</td>
</tr>
<tr>
<td>AUTH1</td>
<td>{0.0072, 0.006}, {0.001, 0.008}</td>
</tr>
<tr>
<td>IMAP1</td>
<td>{0.007, 0.003}, {0.0002, 0.007}, {0.004, 0.006}, {0.001, 0.007}</td>
</tr>
<tr>
<td>SMTP1</td>
<td></td>
</tr>
</tbody>
</table>

### Allocation Table

<table>
<thead>
<tr>
<th>Service</th>
<th>SUN_1</th>
<th>SUN_2</th>
<th>HP_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AUTH</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IMAP</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SMTP</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Input File Format</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOURCES CPU,DISK;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHYSICAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 SUN 2,1;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 HP 2,2;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOFTWARE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEB1 4 send_to_auth, send_to_imap, send_to_smtp, change_to_html;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTH1 2 verify_passwd, verify_session;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMAP1 2 read_message, list_message, delete_message;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMTP1 2 send_message;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deployment and Configuration Scheme I

Input File Format

MAPPINGS
WEB1 SUN_1, HP_1;
AUTH1 SUN_2, HP_1;
IMAP1 SUN_1, SUN_2;
SMTP1 HP_1;

TIMES
WEB1 {0.001,0.007} {0.00098,0.007} {0.002,0.008} {0.001,0.008};
AUTH1 {0.0072,0.006} {0.001,0.008};
IMAP1 {0.007,0.003} {0.0002,0.007} {0.004,0.006};
SMTP1 {0.001,0.007};
Throughput of Software Servers

Throughput (requests per second) vs Arrival Rate (requests per second)

- ID 0
- ID 1
- ID 2
- ID 3

Bhavesh A. Kotak
Performance Evaluation of Distributed Software Systems
Result

Average Queue length on Software Servers

- ID 0
- ID 1
- ID 2
- ID 3

Average queue length (requests per second) vs. Arrival Rate (requests per second)
1. The simulator started running out of Java Heap Space beyond 62 arrivals per second to the system.

2. This indicates ‘Stale-Mate’ situation.

3. Throughputs started dipping and behaving unpredictably after 62 arrivals per second.

4. Queue Lengths suddenly increased mammothly for WEB1 and AUTH1 servers.

5. Either WEB1 or AUTH1 is the bottleneck.
Observations and Inferences

- Accept Design Scheme I and prevent more than 60 arrivals per second to the system
- Make Design Scheme II with more number of threads for AUTH1 and WEB1
- Make Design Scheme III with faster service rate on AUTH1 and WEB1
- Make Design Scheme IV with lesser number of Synchronous calls
- Make Design Scheme V with more physical servers
Observations and Inferences

- Make Design Scheme VI with more number of resources on physical server bottleneck
- Make Design Scheme VII with different mapping between s/w and physical servers
- Make Design Scheme VIII as a combination of any of the above schemes
Increased number of threads of AUTH1 and WEB1 from 2 and 4 respectively to 10 each
Deployment and Configuration Scheme II

Average Queue length on Software Servers

- ID 0
- ID 1
- ID 2
- ID 3

Arrival Rate (requests per second)

Average queue length (requests per second)
Response Time on Software Servers

- ID 0
- ID 1
- ID 2
- ID 3

Graph shows the response time (seconds) on software servers against the arrival rate (requests per second).
Utilization of Software Servers

- Utilization (%)
  - ID 0
  - ID 1
  - ID 2
  - ID 3

- Arrival Rate (requests per second)
The simulator ran smoothly, no ‘Stale-Mate’ situation encountered. (Good News !)

Arrival rates above 130 per second are now supported without any problem. (More Good News !)

Throughputs increased multi-fold, linearly (from 185 to upto 400 requests per second for WEB1)

Queue Lengths for s/w servers and physical servers dipped down to near 0 values, confirming that indeed the bottleneck was a s/w server

Utilization of WEB1 and IMAP1 is reaching 100%, so is the case with all the 3 physical servers

IMAP1, WEB1 and all 3 phy. servers are the new bottlenecks
Increased number of threads on all S/W servers, increased number of resources on all physical servers
Utilization (%)

Arrival Rate (requests per second)
Throughput went even higher (450 req/sec for WEB1)
Arrival rates upto 150 per second supported
Utilization of WEB1 and IMAP1 is reaching 100% at a higher arrival rate (150 req/sec compared to 130 req/sec for WEB1 as well as IMAP1)
The initial design supported maximum 60 arrivals per second
WEB1, AUTH1 were candidates for being the bottleneck
Deployment and Configuration SCHEME II: increased number of threads on WEB1 and AUTH1
Throughput on WEB1 increased from 185 to 400 requests per second
Supported arrival rate increased from 62 to over 130 arrivals per second
Deployment and Configuration SCHEME III: increased number of threads as well as number of resources

- Throughput on WEB1 increased to from 185 to 450 requests per second (243% gain over original design)
- Supported arrival rate increased to from 62 to 150 arrivals per second (241% gain over original design)

Identical trend of improved performance was observed in physical servers as well
Issues or limitations in current implementation

- Need to detect stale-mates and INFORM the same along with the output.
- If arrival rate is very large, and has a non-Poisson distribution, the simulator consumes a large amount of memory.
- For finite queue length implementation, need to implement the action to be taken when a request is dropped.
- Currently, no action is taken, request is simply dropped. Won't work for synchronous forwarded calls.
Future Work

- Verification has been done for Asynchronous calls and Non-layered design. Need to verify for Synchronous and Layered design.
- Add following features
  - Closed queueing network
  - Forks and Joins
  - Special resources like logs, buffers
  - Constructs like loops, synchronization, triggers
- Intense verification of implemented features using Measurement technique
- Read design from a user interface and automate conversion of the design into input file format of the simulator
Currently implemented simulator is giving fairly accurate results

The simulator results help in making performance critical decisions in the design

This influences the decision as to whether the system is fit to be deployed or not

More features are to be added

The simulator can then be used to verify results of analytical model
Thank You
Ids 0-3 are simulation results, Ids 4-7 are analytical results.

Response Time on Physical Servers

<table>
<thead>
<tr>
<th>Arrival Rate (requests per second)</th>
<th>Response Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>1.5</td>
<td>0.15</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
</tr>
<tr>
<td>2.5</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
</tr>
<tr>
<td>3.5</td>
<td>0.35</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>4.5</td>
<td>0.45</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
</tr>
<tr>
<td>5.5</td>
<td>0.55</td>
</tr>
</tbody>
</table>

Tested against Analytical Results
...Tested against Analytical Results

IDs 0-3 are simulation results, IDs 4-7 are analytical results

Utilization of Physical Servers

- Utilization (%)
- Arrival Rate (requests per second)

Graph showing utilization of physical servers across different ID ranges (0-3 for simulation, 4-7 for analytical) with respect to arrival rate.
Throughput of Physical Servers

Throughput (requests per second) vs Arrival Rate (requests per second)

IDs 0-3 are simulation results, IDs 4-7 are analytical results

Tested against Analytical Results
...Tested against Analytical Results

IDs 0-3 are simulation results, IDs 4-7 are analytical results

![Average Queue length on Physical Servers](image_url)

- Average queue length (requests per second)
- Arrival Rate (requests per second)

**Future Work and Conclusion**
...Tested against Analytical Results

![Graph showing response time on resources against arrival rate. The x-axis represents arrival rate (requests per second) ranging from 0 to 8, and the y-axis represents response time (seconds) ranging from 0.015 to 0.055. The graph includes data for four different resource IDs: ID 0, ID 1, ID 2, and ID 3, each represented by distinct line styles and markers.]
...Tested against Analytical Results

![Graph showing Response Time of CallFlows](image)

- **Response Time (seconds)**
- **Arrival Rate (requests per second)**

**ID 0**, **ID 1**, **ID 2**, **ID 3**, **ID 4**

**Bhavesh A. Kotak**

**Performance Evaluation of Distributed Software Systems**
Testing the simulator

**Need to test for**

1. Synchronous calls on complex designs
2. Layered architecture

**Testing using Measurement in Stage III**