Performance Management For Cluster Based Web Services

Ron Levy, Jay Nagarajarao, Giovanni Pacifici, Mike Spreitzer, Asser Tantawi, and Alaa Youssef
ayoussef@us.ibm.com
Outline

- Motivating Scenario
- Objectives
- Architecture and Prototype Implementation
- System Model
- Experimental Results
- Summary and Conclusions
Motivating Scenario

App 1

App 2

Application Hosting Environment

Time Sensitive Service

Bulk Service

Request Flood

Fortune 500 Customer

Small Customer
Motivating Scenario (cont.)

Fortune 500 Customer

Small Customer

Application Hosting Environment

Request Flood

Time Sensitive Service

App 1

App 2

Bulk Service
Need For Overload Control

![Graph showing the relationship between number of concurrent requests and throughput. The x-axis represents the number of concurrent requests, ranging from 0 to 25. The y-axis represents throughput (req/sec), ranging from 126 to 146. The data points are scattered with error bars indicating variability.](image-url)
Objectives

- Classify requests based on client identities and invoked operations
- Isolate requests belonging to different classes
- Allocate resources in real-time and on a per request class basis
- Provide performance guarantees for critical services
- Provide different performance guarantees to different customers
- Use SLAs to represent performance guarantees
- Avoid specific hardware or OS prerequisites to achieve these goals

Benefits:

- Infrastructure efficiency and flexibility
- Ability to handle rapid changes in traffic loads and patterns
- High resource utilization
- Hardware and OS independent solution
Architecture

Real-time mechanisms that act on each request

Event-based control network for asynchronous communications among controllers

Resource manager coordinates controllers to achieve global optimum

Control Network

Global Manager

Management Console
Architecture: Control Mechanisms and Global Resource Manager

- Classification and Admission Control
- Throughput Control
- Scheduling and Flow Control
- Routing and Load Balancing

SLAs
Operational Goals

Global Manager

Traffic Measurements
Service Measurements
Server Measurement
Prototype Implementation: Modular Design Takes Advantage of Axis Handlers or WS Gateway Filters

- Request Queue Manager/Scheduler
- Authentication Handler
- Classification Handler
- Throughput Policing Handler
- Request Queue Handler
- Dispatch Handler
- Response Handler
- Other Handler

Request flow:
- Request arrives
- Auth Handler
- Classification Handler
- Throughput Policing Handler
- Request Queue Handler
- Dispatch Handler
- Response Handler
- Other Handler

Response flow:
- Response
- Resource release

Dispatch requests to servers:
Global Resource Manager

- Resource Configuration
- SLA Contracts
- Requests Scheduler
  - Traffic Load
  - Service Time
- Management Parameters
  - Utility
- Resource Monitoring
  - Available Capacity
- Global Resource Manager
  - Scheduling Weights
  - Max Concurrency
  - Requests Scheduler
Utility Function

- Function of service performance level (avg response time)
- Maps the service performance level into a business value measure
- May capture customer expectation and satisfaction
- May implement a business game plan

Utility Function $U_c(\tau_c, t_c)$

- $\alpha_c = 2$
- $\alpha_c = 1.5$
- $\alpha_c = 1$
- $\beta_c = 1$
- $\beta_c = 1.5$
- $\beta_c = 2$
Optimization Criteria

- **Cumulative:**
  maximize sum of utilities

  \[
  \text{Max} \quad U_1 + U_2
  \]

- **Equalize:**
  maximize min of utilities

  \[
  \text{Max} \quad \text{Min} (U_1, U_2)
  \]
Management Console

In manual mode
System Model

- GRM uses system model to predict response time given a certain allocation vector
- Model continuously tuned based on measured performance parameters
- Experimented with several queuing models

\[ \text{Lambda} = \frac{\lambda}{n} \]
\[ \mu = \frac{1}{T} + \frac{\lambda}{n} \]
Experimental Results

System Behavior
Closed Loop Model

- **Clients**
  - Think Time
  - Throughput

- **System**
  - Response Time
  - Service Time
  - Utilization
  - Queue Length
Demo Configuration and Assumptions

- Closed loop with finite clients
- Max service request concurrency (N = 10)
- Number of classes (K = 2)
- Mean response time SLO (t)
- M/M/1 Queue Model
- Linear utility function u(t)
- MaxMin utility optimization criterion
## Class Parameters

<table>
<thead>
<tr>
<th></th>
<th>Gold</th>
<th>Silver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Time ((1/\lambda))</td>
<td>0.5 sec</td>
<td>0.5 sec</td>
</tr>
<tr>
<td>Think Time ((Z))</td>
<td>1.0 sec</td>
<td>1.0 sec</td>
</tr>
<tr>
<td>Min Allocation</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Target ((\tau))</td>
<td>1.0 sec</td>
<td>3.0 sec</td>
</tr>
<tr>
<td>Weight</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Demo Scenario

Silver Clients

Gold Clients

10 10

20 20

30 30

a

c

b

d

ad

cb
Performance Management for Cluster Based Web Services

- Gold Clients: 10, 30
- Silver Clients: 10, 30

Graph: Average Q Length (req)

- Points: a, b, c, d
- Time: 03:18 03:21 03:23 03:24 03:25 03:26 03:27 03:28 03:30 03:31 03:32 03:33 03:34 03:35 03:36 03:37 03:38
Experimental Results

Performance Comparison
Utility Regions Comparison

- Disciplines
  - FIFO (no control)
  - Static Priority
  - GRM controlled
Static Priority: Utility

Minimum Utility (No Control)

Minimum Utility
GRM Controlled: Utility

Minimum Utility (No Control)

Minimum Utility (Control)

Gold Clients

Silver Clients

GRM Controlled:
Utility
Summary and Conclusions

- Designed and prototyped SLA-based performance management system for Web Services
- GRM yields near optimal allocations using simple queueing models with dynamic correction
- The impact of allocation is more significant in heavy load conditions
- Disciplines that are oblivious to performance targets may lead to undesirable results
- System is responsive to coarse fluctuations in traffic characteristics

- WSMM included in IBM WSTK (Web Services ToolKit) available for download at: www.alphaworks.ibm.com/tech/webservicestoolkit