Adaptive Control of Virtualized Resources in Utility Computing Environments

Pradeep Padala, Kang G. Shin

Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant

Kenneth Salem

June 2007
Publisher: ACM

July 8, 2009
OUTLINE

• INTRODUCTION
• MODELING
• DESIGN
• EVALUATION
• CONCLUSIONS
INTRODUCTION

Each application has its own dedicated servers in traditional way.

- Email
- WWW
- DB

Data Center
An example of data center server consumption (each node has 6 CPUs)
INTRODUCTION

Applications shares resources as their demands changes over time

Email  WWW  DB

Next-generation Data Center
INTRODUCTION

(c) Sum of CPU consumptions from both nodes

An example of data center server consumption (each node has 6 CPUs)

Solution: Adaptive Controller
INTRODUCTION

Goals
- Good Utilization
- Good Performance
- QoS Differentiation

Controller

CPU Usage?
Set CPU shares

Xen SEDF scheduler
Virtualized Server

VM 1
50%

VM 2
50%

Goals met? NO

CPU share
50%
50%
INTRODUCTION

Goals
- Good Utilization
- Good Performance
- QoS Differentiation

VM 1
---
VM 2

CPU share
- 80%
- 20%

Control CPU shares based on goals
Enterprise applications typically employ a multi-tier architecture
MODELING

• First define some terminology
  • entitlement \((u)\): the percentage of CPU share allocated to a virtual machine
  • consumption \((v)\): the percentage of total CPU capacity actually used by the VM
  • VM utilization \((r)\): the ratio between consumption and entitlement \((r = v / u)\)
An input-output model for a multi-tier application
Experimental Testbed

- 5 HP proliant servers, each has two processors, 4 GB of RAM, one Gigabit Ethernet interface, and two local SCSI disks
- Used two workload generators
  - RUBiS: an online auction site benchmark
  - TPC-W: a transactional e-Commerce benchmark
Experimental Testbed

MODELING

Client 1
QoS Sensor 1

Client 2
QoS Sensor 2

WWW node
WWW VM (w1)
A
S

WWW VM (w2)
A
S

DB node
DB VM (d1)
A
S

DB VM (d2)
A
S

Measured QoS and resource utilization

Resource Controller

Resource allocation decisions
MODELING

Modeling single multi-tier application

• To know how the changes in the WWW/DB entitlements impact the utilization of virtual machines and QoS metrics

• A single testbed node was used to host a two-tier implementation of RUBiS

• Pinned the WWW VM(20-70%), the DB VM(20%), as well as dom0(remaining) to one processor
MODELING

Modeling single multi-tier application

(a) WWW CPU consumption
(b) DB CPU consumption
(c) dom0 CPU consumption
(d) WWW VM utilization
(e) Throughput
(f) Response time
MODELING

Modeling co-hosted multi-tier applications

Client 1
QoS Sensor 1

Client 2
QoS Sensor 2

Resource Controller

Measured QoS and resource utilization

Resource allocation decisions

WWW node
WWW VM (w1)
A
S

WWW VM (w2)
A
S

DB node
DB VM (d1)
A
S

DB VM (d2)
A
S
MODELING

Modeling co-hosted multi-tier applications

- At any given time either the WWW node or the DB node may become saturated

<table>
<thead>
<tr>
<th></th>
<th>DB node unsat.</th>
<th>DB node sat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWW node unsat.</td>
<td>WU-DU</td>
<td>WU-DS</td>
</tr>
<tr>
<td>WWW node sat.</td>
<td>WS-DU</td>
<td>WS-DS</td>
</tr>
</tbody>
</table>

- Also need a QoS differentiation metric:

$$ yratio = \frac{y_1}{y_1 + y_2} $$

y can be average response time, throughput, or request loss
MODELING

Modeling co-hosted multi-tier applications

Response time ratio is more controllable than loss ratio
DESIGN

Utilization goal = 80%

Controller

Set to 40%

VM

Using 20%

Utilization: 20/40 * 100 = 50%
New Utilization: 20/25 * 100 = 80%

An example of the controller
\[ \text{ureq}(k) = \text{ureq}(k - 1) - K(k)e(k - 1) \]

Adaptive utilization controller
DESIGN

Container consumptions

WWW node
  UC for w1
  UC for w2

Want 40%
Want 70%
{ 110% (Saturated)

DB node
  UC for d1
  UC for d2

All controllers are independent
A two-layered controller architecture
EVALUATION

Throughput (reqs/sec)

Sample point (every 10 secs)

Varying load generated by clients
EVALUATION

CPU entitlement and consumption

Sample point (every 10 secs)

% of total CPU
EVALUATION

CPU entitlement and consumption
EVALUATION

CPU entitlement and consumption

Sample point (every 10 secs)

% of total CPU

- Web II entitlement
- Web II consumption
EVALUATION

Sample point (every 10 secs)

CPU entitlement and consumption
EVALUATION

CPU entitlement and consumption

RT Ratio

Sample point (every 10 secs)
CONCLUSIONS

• Achieves high utilization of the data center while meeting application-level QoS goals

• Be able to provide a specified level of QoS differentiation between applications under overload conditions