Schedule Distributed Virtual Machines in a Service Oriented Environment

Lizhe Wang
Gregor von Laszewski
Suresh Marru (speaker)
Pervasive Technology Institute
Indiana University, US

Jie Tao
Marcel Kunze
Steinbuch Center for Computing
Karlsruhe Inst. of Tech., Germany
Outline

• Introduction
• Problem definition
• Schedule distributed VMs in a SOA environment
• Performance study
• Conclusion
Introduction

- Service oriented architecture
- Virtual machine
- Task scheduling
Service Oriented Architecture

Service broker

WSDL

UDDI

f(x)

Service requester

SOAP

Service provider
Virtual machine

- A build block for modern IT infrastructures and Clouds
- Advantages:
  - On-demand creation and customization
  - QoS guarantee and performance isolation
  - Legacy software support
  - Easy management
- Popular VMMs
  - Xen
  - VMware
Task scheduling

• Static scheduling
  – the assignment of tasks to resources is performed before applications are executed
  – all the overhead of the scheduling process is incurred at compilation time
  – Sometime, Hard to get resource/task information before execution
  – Cannot adaptive to dynamic changed environment

• Dynamic scheduling
  – redistribution of tasks among PEs during execution time.

• VM is suitable for static scheduling
  – VM as a task, information can be achieved before execution
  – Dynamic migration of VM across SOA maybe expensive
Schedule VM in a SOA

• Computing resources required to be scheduled are multiple dimension, for example, CPU bandwidth, memory, and software licenses. In traditional scheduling environment, only processor resources are considered for resource allocation.

• Resource allocation should be considered with more restrictions, for example, some applications can only be scheduled to certain virtual machines that provide the required application execution environment. This scenario does not exist in the traditional distributed system.
Formal definition

• SOA = \{\text{Host, VM, VM affiliation}\}
• Task= \{\text{Job, job dependency}\}
  – A job can have multiple resource requirements:
    • CPU number
    • Memory,
    • OS
    • Software licenses
    • ……

• Problem definition:
  – Find a map from Task to SOA
  – Minimize Task Execution Time
The scheduling framework

Job submission

Job queue

Virtual machine pool

Scheduler (M-DSA)

Resource information

Job scheduling
Task scheduling algorithm

• Generate DAGs form parallel tasks
• Sort jobs in a parallel task in priority with job dependency
• Scheduling algorithm
  – Greedy algorithm
  – Get the first job from sorted job list
  – Find a resource that fit job’s requirement in multi-dimension
An demonstration example
Simulation results
A test SOA environment

Host

Host

Host

... Register

UDDI

Response

Query

Reply

Submit & monitor

Clients

UDDI

Host resource

1

2

3

4

5

6
Implementation
Implementation SOA execution sequence diagram
Bio-sequence Alignment: a Use Case

• The process of bio-sequence alignment is to execute the Smith-Waterman algorithm and compare the query sequence with the bio-database.

• Master-slave paradigm

• Use distributed VM to execute slave jobs
Bio-sequence Alignment: a Use Case
Test bed

IWR

Virtual machine pool
vm vm ...... vm

VMware infrastructure
host host ...... host

RZ

Virtual machine pool
vm vm ...... vm

Xen servers
host host ...... host
Table 1: Test results of bio-sequence alignment

<table>
<thead>
<tr>
<th>Task Name</th>
<th>CPU (GHz)</th>
<th>Memory (MB)</th>
<th>Req. TET (Second)</th>
<th>Actual TET (Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>divide</td>
<td>0.5</td>
<td>500</td>
<td>60</td>
<td>52</td>
</tr>
<tr>
<td>align</td>
<td>1</td>
<td>1000</td>
<td>5800</td>
<td>5632</td>
</tr>
<tr>
<td>collect</td>
<td>0.5</td>
<td>500</td>
<td>60</td>
<td>44</td>
</tr>
</tbody>
</table>
Conclusion

• Multiple dimension VM scheduling in a SOA
• Implement in a SOA environment
• Make a performance study in a wide area