Contents

1 Introduction
2 Problem Definition
3 Design
4 Implementation
5 Results
6 Future Work
7 References
Cloud computing: Computing as Utility

- Computing: Hardware, Software, Platform

- Pay as you use [1]
Virtualization

- Components: Physical resources, Virtualization, Applications

Figure: virtualization layers

- Testing of policies difficult in real cloud environment
Software tools

- **Tools**: Eucalyptus [3], OpenNebula

**Figure**: Eucalyptus components

- **Features**
  - user interface for monitoring
  - multiple virtualization technique
  - resource scheduling
  - integration with public cloud

- Requires large number of physical resources
Motivation

Cloud computing challenges

- How to maintain quality of service?
- How to utilize resources efficiently?
- How to meet varying demand of applications?
- How to meet service level agreement?

Possible solution approaches

- Placement policies
- Resource allocation policies

Difficulties

- Requires large number of physical resources
- Needs low level access
Problem definition

Design and implementation of cloud computing environment simulator, which aims to help developers to model and test policies:

- Modelling cloud computing environment
- Modelling applications
- Testing of properties or behaviours of cloud computing
- Quantifying the performance of policies
  - resource allocation, application scheduling and migration policies
- Testing Cloud computing environments for different application and service models under varying load
## Related work

<table>
<thead>
<tr>
<th>Features</th>
<th>CloudSim [2]</th>
<th>VirtualCloud</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td>Modelled as number of instructions, list of files</td>
<td>collection of pairs (time, value) representing the resource requirement</td>
</tr>
<tr>
<td><strong>Execution model</strong></td>
<td>Maximum utilization model</td>
<td>Different execution models (Distribution based model, Fixed model, Maximum model)</td>
</tr>
<tr>
<td><strong>Policies</strong></td>
<td>Implements static placement and fixed resource allocation policy</td>
<td>Placement and resource allocation policies can be modelled (hard and soft allocation policy implemented)</td>
</tr>
<tr>
<td><strong>Virtualization technique</strong></td>
<td>Not implemented and no dependency, interference between different resources</td>
<td>Currently in design, dependency and interference among resources. Future implementation of Xen, KVM</td>
</tr>
</tbody>
</table>
VirtualCloud block diagram

Figure: Block diagram of Virtualcloud
Cloud computing abstraction

- Cloud computing environment: interconnected physical machines running different applications using some virtualization technique
- Application: resource consumption, execution length, execution pattern
- Physical and virtual machine: resource configuration, list of vm or applications running
- Communication: transmission delay between two entities
- Policies: set of conditions and corresponding actions
Figure: uml class diagram
Implementation

- Application class
  - Resource requirement, execution policies
- Virtual machine
  - configuration of resource limit, applications
- Physical machine
  - resource configuration, virtualization technique, resource allocation policy
- Allocation policies
  - Hard and soft allocation
- Input and output specification
  - Application characteristics and monitoring parameters
Testing of features

Features

- Abstraction of applications
- Abstraction of virtual machines and physical machines
- Execution model
- Configuration, input and output

Test

- deterministic regression test: done
- validation using experiment: pending
Experiment setup

- 4 virtual machines in a single physical host
- Each vm has same priority
- Each vm is running cpu-centric application with different utilization level
- Resource allocation policy
  - Hard: Each vm has maximum fixed resource utilization
  - Soft: Vm can have resource utilization higher than limit, when other vms are not using their portion

Measured output

- Resource requirement and availability at application level
- Resource utilization at host level
• **Goal:** Variation of resource availability and utilization on different allocation policies

![Experiment result](image)

(a) Virtual machine 1  
(b) Virtual machine 2

**Figure:** Experiment 1
Experiment result

Allocation policies

- Resource utilization at soft allocation is better than hard allocation
- Resource availability at vm level is higher in soft allocation
- Difference between demand and available amount is less in case of soft allocation
- Applications will finish earlier in soft allocation than hard allocation
Validation

Goal

Variation of resource allocation among virtual machines by hypervisor under different conditions

Setup

- 2 virtual machines on a single physical host
- Virtualization technique: KVM
- tools: lookbusy to generate cpu load
- allocation policy: hard and soft
- resource allocation at vm and pm level

Input

Applications having different varying demand. Application will follow different execution model
Implementation

- Different virtualization technology
- Migration process
- Manager module
- Dependency and interference among resources
Thank You!
[Michael Armbrust et al.]
Above the clouds: A berkeley view of cloud computing.

[Rajkumar Buyya et al.]
Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportuniess.

[Daniel Nurmi et al.]