Achieving Adaptation Through Live Virtual Machine Migration in Two-tier Clouds

Hongbin Lu
Supervisor: Marin Litoiu
Outline

* Introduction.
* Background.
* Multi-cloud deployment.
  * Architecture.
  * Implementation.
  * Experiments.
* Adaptation through VM migration.
  * Framework.
  * Experiments.
* Conclusions and Future Work.
Introduction

* The **multi-cloud** means a combination of computing resources from multiple clouds.
* The **two-tier cloud** refers to a special type of multi-cloud, where resources from multiple cloud providers are divided into two groups.
* Deploying or managing applications in two-tier clouds or multi-clouds is challenging.
Main contributions

* Multi-cloud deployment.
  * A deployment architecture.
  * An implementation of the architecture.
  * A domain specific language.
  * Experimental evaluations of the implementation.
* Adaptation through VM migration.
  * A live VM migration management framework.
  * Experimental evaluations of the framework.
Cloud computing refers to the delivery of computing resources over the Internet as a service.

Virtualization is an enabling technology of cloud computing.

Live virtual machine (VM) migration is a technique to re-locate a VM without stopping its execution (or minimizing the disruption).

Autonomic computing refers to a computing system that manages itself.
Multi-cloud Deployment: Architecture
Multi-cloud Deployment: The Deployment Model
Multi-cloud Deployment: The Metamodel

- **Topology**
  - +name
  - +deploy()
  - +undeploy()

- **Node**
  - +name
  - +cloud
  - +instance_type
  - ...
  - +migrate()

- **Service**
  - +type
  - +war_file
  - +script
  - ...

- **Action**
  - +type
  - +name
  - +operation
  - ...

- **Container**
  - +num_of_copies
  - +scale()
Multi-cloud Deployment: The Deployment Model

**Web host**
- cloud: EC2
- image_id: ami-3fec7956
- Instance_type: t1.micro
- ...

**Data host**
- cloud: OpenStack
- image_id: 8
- Instance_type: 2
- ...

**web_server**
- application_server: Tomcat6
- war_file: petstore.war
- ...

**database_server**
- database_system: MySQL
- database_name: mydb
- ...

**Legend**
- connection
- service
- node
- topology
Multi-cloud Deployment: Pattern Deployment Service (PDS)
Multi-cloud Deployment: Pattern Deployment Service (PDS)

- Open source software.
- Have been used by SAVI research community.
- Features:
  - Domain specific language.
  - Restful web service.
  - Parallel and distributed deployment.
  - Provide real-time deployment status.
  - Recoverable from deployment failures.
  - Support user authentication / admission.
Multi-cloud Deployment: Experiment1

**SAVI (a small cloud)**

<table>
<thead>
<tr>
<th>Number of concurrent users</th>
<th>Elapsed time (in second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>1500</td>
</tr>
<tr>
<td>10</td>
<td>2000</td>
</tr>
</tbody>
</table>

**EC2 (a large cloud)**

<table>
<thead>
<tr>
<th>Number of concurrent users</th>
<th>Elapsed time (in second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
</tr>
<tr>
<td>10</td>
<td>1000</td>
</tr>
</tbody>
</table>

- **Deploy**
- **Scale out**
- **Scale in**
- **Undeploy**
Multi-cloud Deployment: Experiment 2

Elapsed time (in second)

Number of concurrent users

Deploy on pre-installed image (SAVI)
Deploy on standard image (SAVI)
What Is Next

* Introduction.
* Background.
* Multi-cloud deployment.
  * Architecture.
  * Implementation.
  * Experiments.
* Adaptation through VM migration.
  * Framework.
  * Experiments.
* Conclusions and Future Work.
The underlying infrastructure consists of two clouds: A core and a smart edge.

- A **core**: a large datacenter.
- A **smart edge**: a small datacenter.

Two types of VM migration:

- **Local migration**: VMs are migrated within the smart edge.
- **Cross-cloud migration**: VMs are migrated between the core and the smart edge.

Managing cross-cloud migration is our focus.
Adaptation through VM migration: Cross-cloud Migration
Adaptation through VM migration: Architecture
Adaptation through VM migration: Migration Execution

```
Topology
+name
+deploy()
+undeploy()

Node
+name
+cloud
+instance_type
+...
+migrate()

Service
+type
+war_file
+script
+...

Action
+type
+name
+operation
+...

Container
+num_of_copies
+scale()
```
Adaptation through VM migration: Workloads of Experiments

Number of users

Hours

App1 (E-commerce)

App2 (General web traffic)

App3 (Media traffic)
Adaptation through VM migration: Setup of Experiments
Adaptation through VM migration: Migration Process
Adaptation through VM migration: The Two Experiments

Experiment 1

* **Goal:** Evaluate performance of different migration policies.

* **Process:**
  * Round 1: Migration was disabled.
  * Round 2: Least-utilized first (LUF).
  * Round 3: Most-utilized first (MUF).

Experiment 2

* **Goal:** Validate if prediction can improve the performance.

* **Process:**
  * Round 1: Workload-prediction was disabled.
  * Round 2: Linear regression was used.
Adaptation through VM migration: Results of Experiment 1

Round 1 (baseline) | Round 2 (LUF) | Round 3 (MUF)

Response time (in milliseconds)

Elapsed time (in minutes)

App1 (E-commerce)  |  App2 (General traffic)  |  App3 (Media traffic)
Adaptation through VM migration: Results of Experiment 2

Round 1 (baseline) vs Round 2 (predict)

- Response time (in milliseconds)
- Elapsed time (in minutes)

Graph showing response time for different applications across two rounds.
Conclusions and Future Work

Contributions

* A multi-cloud deployment architecture.
* The PDS, an implementation of the architecture.
* A domain specific language.
* Experimental evaluations of PDS.
* A live VM migration management framework.
* Experimental evaluations of the framework.

Future Work

* Extend the functionalities of PDS.
* Improve the scalability of PDS.
* Port the migration experiments to SAVI testbed.
* Evaluate more migration policies and prediction model.
Publications:


Software Product (Open source):


Tutorials/Workshops:

