Resource Management in IaaS Cloud Platforms made Flexible through Programmability

Juliano Araujo Wickboldt

XV Workshop da Rede Nacional de Ensino e Pesquisa (WRNP 2014)

May 5th, 2014
1. Introduction
2. Concepts
3. Prototype
4. Evaluation & Results
5. Demo
Introduction
Introduction
Background

End User

Operator

Cloud Controller Platform

Computing

Storage

Network
Cloud computing platforms tend to separate resource management concerns into computing, storage, and network.

Computing and storage management in current clouds can be considered stable.

The network management part, on the other hand, is rather incipient.

Most platforms rely on external systems for any complex network management task, such as:

- Creating dynamic overlay network topology
- Setting up communication requirements (e.g., bandwidth)
Moreover, cloud platforms are conceived as one black-box-like centralized controlling systems.

This configures two different flexibility shortcomings:

- Poor description of application specific requirements (affects the end-user)
- Hard-coded resource allocation strategies (affects the cloud manager)
• Take a new approach to cloud platform design considering:
  – Robust network module for allying cloud computing with modern network paradigms
  – Specification of complex virtual infrastructures including all kinds of virtual resources and application specific requirements
  – Programmability at the core of the platform with a simplified API, to allow personalized programs to manage resources
• Benefits of customization in resource allocation are under two perspectives:

1. To achieve environment specific objectives
   • Optimizing energy consumption
   • Reducing expensive link utilization

2. To enable application oriented resource allocation
   • Placing connected virtual machines closer to one another in the data center to reduce communication delay
Concepts
### Concepts

#### Architecture

- **GUI**
  - User Interface
  - Administrative Interface

- **Slice Space**
  - Cloud Slice Manager
  - Deployment Engine
  - Optimization Engine
  - Metrics Engine
  - Event Manager

- **Unified API**
  - Computing
  - Storage
  - Networking
  - Monitoring

- **Drivers**
  - Libvirt
  - OpenFlow
  - Open vSwitch
  - FlexCMS
  - Nagios

- **Managed Infrastructure**

- **Resource Management**
  - Programs & Metrics

- **Monitoring Infrastructure**
  - Alarm Trigger
  - Configuration Manager
  - Monitoring Engine

---

**XML Version Description**

```
<?xml version="1.0" ...
```

**Include Files**

```
#include <stdio.h>
```

**Programs & Metrics**

```
#include <stdio.h>
#include <stdio.h>
```

**Configuration Manager**

```
```

**Monitoring Engine**

```
```
Prototype
Prototype Platform Interactions

Aurora Cloud Manager
- GUI
- Slices
- Programs
- Events
- Unified API
- Drivers
- Libvirt
- OFlow
- FlexCMS
- OVSwitch
- Nagios

FlexCMS
- GUI
- Change Manager
- Configuration Manager
- Gatherers
- Aurora
- Nagios

Nagios
- GUI
- Event Logic
- Monitoring Logic
- Configurators
- Plugins
- virsh

Information Requests

Managed Infrastructure

Cloud Slices

Physical Network

Administrator
3 Prototype Technologies

- Web based application
  - Python-Django (MVT)
- Virtual infrastructure spec. with VXDL
- Framework for execution of resource allocation programs and metrics
- Unified API Implementation

**Computing**
- VMs
- Images
  - create, start, stop, destroy
  - migrate ...

**Storage**
- VDisk
- VDiskPool
  - create, attach, destroy ...

**Network**
- VLink
- VRouter
  - create, establish, disable, remove ...

**Monitoring**
- Physical
- VirtualDev
- Events
  - status, memory, cpu and disk utilizations and allocations ...

- `<vNode id="Node0">`
  - `<cpu>`
  - `<cores>`
    - `<simple>1</simple>`
  - `<frequency>`
    - `<simple>1</simple>`
    - `<unit>GHZ</unit>`
  - `<memory>`
    - `<simple>128</simple>`
    - `<unit>MB</unit>`
  - `<storage>`
  - `<interval>`
    - `<min>500</min>`
  - `<interface>`
    - `<alias>net0</alias>`
    - `<type>bridge</type>`
• OptimizeBalance
  – Tries to spread virtual machines of a Cloud Slice over the managed infrastructure

• OptimizeGroup
  – Goes into the opposite direction by mapping virtual machines into the smallest subset of physical nodes

• OptimizeHops
  – Reduces the number of hops between linked virtual machines of a Cloud Slice
Evaluation & Results
Evaluation & Results
Emulated Infrastructure
- NetInf an Information Centric Networking (ICN) application
- Three topology with caching
- Deployment
  - 7 NetInf Routers
  - 4 Clients
  - Debian Squeeze (330MB)
  - Deployment time ~60sec
Evaluation & Results

Topology Examples

(a) Tree topology with 15 virtual machines

(b) Tree topology with 7 virtual machines

(c) Ring topology with 4 virtual machines
### Evaluation & Results

**Some Deployment Results**

#### Deployment Statistics: Overall Time

<table>
<thead>
<tr>
<th></th>
<th>Info. gathering</th>
<th>Reasoning</th>
<th>Image copy</th>
<th>VM define</th>
<th>VM Start</th>
<th>Link estab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring 4</td>
<td>0.17s</td>
<td>0.94s</td>
<td>1.95s</td>
<td>2.04s</td>
<td>6.67s</td>
<td>0.26s</td>
</tr>
<tr>
<td>Tree 7</td>
<td>0.17s</td>
<td>2.12s</td>
<td>3.58s</td>
<td>10.95s</td>
<td>14.21s</td>
<td>0.81s</td>
</tr>
<tr>
<td>NetInf</td>
<td>0.20s</td>
<td>4.65s</td>
<td>5.71s</td>
<td>17.88s</td>
<td>23.81s</td>
<td>5.56s</td>
</tr>
<tr>
<td>Tree 15</td>
<td>0.25s</td>
<td>8.52s</td>
<td>7.80s</td>
<td>27.51s</td>
<td>30.98s</td>
<td>0.99s</td>
</tr>
</tbody>
</table>

#### Deployment Statistics: Per VM/Link

<table>
<thead>
<tr>
<th></th>
<th>Reasoning/ VM/Link</th>
<th>Image copy/VM</th>
<th>VM define/ VM</th>
<th>VM Start/ VM</th>
<th>Link estab./ Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring 4</td>
<td>0.06s</td>
<td>0.49s</td>
<td>0.51s</td>
<td>1.67s</td>
<td>0.06s</td>
</tr>
<tr>
<td>Tree 7</td>
<td>0.05s</td>
<td>0.51s</td>
<td>1.56s</td>
<td>2.03s</td>
<td>0.14s</td>
</tr>
<tr>
<td>NetInf</td>
<td>0.04s</td>
<td>0.52s</td>
<td>1.63s</td>
<td>2.16s</td>
<td>0.56s</td>
</tr>
<tr>
<td>Tree 15</td>
<td>0.04s</td>
<td>0.52s</td>
<td>1.83s</td>
<td>2.07s</td>
<td>0.07s</td>
</tr>
</tbody>
</table>

- Most expensive task
- Weird variations
• Optimization Scenarios

1. A previously allocated Cloud Slice expires and releases resources
2. A Cloud Slice is modified by the addition of virtual machines
3. The removal of virtual machines from a Cloud Slice makes room for reconfigurations similarly to item 1;
4. The Administrator performs maintenance in the infrastructure, for example, to install software updates in the nodes
5. The Administrator replaces equipment to increase the capacity of the infrastructure
Evaluation & Results
Some Optimization Results

OptimizeGroup Results

OptimizeBalance Results
HopCount Results

- Max
- Min
- Mean
- Stdv
• Resource Management in IaaS Cloud Platforms made Flexible through Programmability
  – Juliano Araujo Wickboldt, Rafael Pereira Esteves, Márcio Barbosa de Carvalho, Lisandro Zambenedetti Granville
  – Published in Elsevier Computer Networks (in press)

DOI: http://dx.doi.org/10.1016/j.comnet.2014.02.018
DEMO NOW!