The Sirocco multi-cloud management framework

Frédéric Dang Tran
Paweł Rubach

Orange Labs

{frederic.dangtran,pawel.rubach}@orange.com
Outline

- Context and objectives
- Sirocco architecture overview
- Focus on the DMTF CIMI standard
- Placement management
- Conclusion
Context and Objectives (1/2)

- Accessing a single Cloud provider is not enough
  - Cloud bursting from a private to a public cloud during load peaks
  - Satisfy application specific constraints such as location, performance, costs, ...
  - Means to implement disaster recovery and HA configurations by replication over multiple clouds

- Hiding IaaS provider heterogeneity
  - Different APIs
  - Different infrastructure models: compute, storage, network
  - Different format of VM images
Context and Objectives (2/2)

- **What is Sirocco?**
  - Multi-cloud management platform
  - Single point of entry to access multiple cloud providers
  - Unified IaaS API based on the DMTF CIMI API standard
  - Extensible and open middleware framework
Sirocco Architecture

Sirocco multi-cloud management framework

DMTF CIMI API Server

Request orchestration

User & Tenant management

Resource Provisioning & Lifecycle mgmt

Provider & Placement Management

Image Management

Monitoring & Metering

Cloud Provider API gateway and connection management
DMTF CIMI API (1/3)

- DMTF Cloud Management Working Group (CMWG)
  - Focus on Cloud IaaS API
  - Cloud Infrastructure Management Interface (CIMI) 1.0 API specification released 29 August 2012

- 34 involved companies and 10+ academic or alliance partners:
  - IBM, VMware, CA, Cisco, Citrix, Fujitsu, HP, Huawei, ZTE, Microsoft, Oracle, Red Hat, Telecom Italia, Telefonica, France Telecom…
DMTF CIMI API (2/3)

CIMI Provider

- Machine: Compute resource with CPU, memory and local disks
- Volume: Persistent storage that can be attached/detached to/from a Machine Block or file-system level
- Network: Level 2 network Broadcast domain
- System: Appliance grouping machines, volumes and networks Managed as single unit

OVF import/export
Mapping CIMI API → IaaS API

CIMI model
System
Network

vCloud Director
Amazon VPC
OpenStack Quantum
Placement Management – Motivation

How to choose the cloud provider or data center where an app is deployed

For a single cloud operator:

> Operating geographically distributed data centers with potentially differentiated service offers
> Different policies: load balancing, energy management, disaster recovery (deploy on two clouds separated by a distance), deploy close to active users, deploy within a network distance (latency, bandwidth)

For a cloud broker

> Mediate in face of cloud providers with different service offers, prices,
> Maximize revenue
Placement Management - matching provider “iaas” offers based on virtual machine configuration

### Cloud Provider "A"

<table>
<thead>
<tr>
<th>BUNDLE NAME</th>
<th>CPU</th>
<th>MEM (GB)</th>
<th>DISK (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALL</td>
<td>{1, 2}</td>
<td>(0.5 – 2.0)</td>
<td>(2 – 40)</td>
</tr>
<tr>
<td>MEDIUM</td>
<td>{2, 4, 8}</td>
<td>(1.0 – 8.0)</td>
<td>(4 – 200)</td>
</tr>
<tr>
<td>LARGE</td>
<td>{4, 8, 16}</td>
<td>(2.0 – 16.0)</td>
<td>(16 – 800)</td>
</tr>
</tbody>
</table>

### Cloud Provider "B"

<table>
<thead>
<tr>
<th>CPU</th>
<th>MEM (GB)</th>
<th>DISK (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1,2,4,8,16}</td>
<td>(0.25-32.0)</td>
<td>(1-1000)</td>
</tr>
</tbody>
</table>

Which offer to choose?
Placement Management – Requirements for Architecture

- Flexible and extensible: facilitate administration of placement policies
  - Adapt to evolving operator goals and policies
  - Evolving constraints
  - Allow adaptation to changes in demand, changes in cloud provider offers, prices, policies
Placement Management – Architecture
Placement Management – Current Status

- Developed extensions to the CIMI model to allow for:
  - matching by capabilities (including bundles)
  - matching by locations (including network distance etc.)
  - passing placement requests and decisions
- Developed PoC according to the presented architecture with simplified matching and optimization rules
Conclusion: next steps

- Complete DMTF CIMI 1.0 API implementation
- Integrate third-party VM Image Factories (e.g. UShareSoft)
- Integrate Application Lifecycle Management service based on OVF

- IaaS provider support: complete mapping of CIMI network model to
  - VMware vCloud Director
  - OpenStack Quantum

- Placement management:
  > Enhance matching rules
  > Provide a mechanism to change rules dynamically
  > Implement feedback mechanisms
Points

- OW2 Sirocco project home page
  > http://sirocco.ow2.org

- SCM repositories
  > svn://svn.forge.objectweb.org/svnroot/sirocco/cloudmanager/trunk
  > git://gitorious.ow2.org/ow2-sirocco

- OpenCloudware collaborative project (FSN)
  > http://www.opencloudware.org

- LGPL open-source license