A network transparent solution for flexibly provisioning connected virtual infrastructure across multiple data centers

Huan Zhou, Paul Martin, Cees de Laat, Zhiming Zhao
University of Amsterdam, Amsterdam, the Netherlands

Background

This poster presents a network-transparent flexible provisioning agent for complex virtual infrastructure distributed over multiple data centers or cloud providers. This work is conducted in the context of the EU H2020 project SWITCH as part of a dynamic real-time infrastructure platform for cloud applications. This technology can be used by system developers to customize and provision virtual infrastructure for applications with critical performance constraints that may influence how and where application components are hosted.

Research Problems

Quality critical applications such as disaster early warning, live event broadcasting or real-time business collaboration often require customized virtual infrastructure with tailored service level agreements (SLAs) when migrated into a cloud environment. A typical architecture for a big data quality critical application is shown in the figure below.

According to the current state of the Cloud in industry, we infer the following challenges and gaps when migrating this kind of quality-critical application onto Cloud, focusing mainly on infrastructure provisioning.

Networked infrastructure
- The application workflow becomes more complex with a lot of components that need to communicate with each other. The virtual infrastructure must therefore realize a particular network topology.
- Most current cloud providers cannot support this; for example, Amazon EC2 can only allow users to describe private subnets, making it hard to build a complete topology.

Nearly real-time constraints
- Static constraints (e.g. network transmission time as data is processed), which restrict task scheduling before provisioning.
- Runtime constraints (e.g. on failure recovery time), which affect how applications are managed at runtime, especially where the Cloud is not wholly reliable.

Geography
- In the figure above, not all the components of an application are on the Cloud.
- Virtual infrastructure should be provisioned close to external components in order to satisfy any nearly real-time constraints on data delivery.

Auto-provisioning and federated cloud
- We need a way to provision the whole infrastructure and deploy applications automatically. Currently, some tools can only provision automatically at instance level, for example Chef.
- The application may need more resources from other Clouds to provision a large scale infrastructure. It is a problem to combine these resources across multiple locales however.

Approach

Connection Methods
- NAT Based
- Tunnel Based

Virtual Cloud

Data Analysis And Storage

Data Pre-processing

Data Acquisition

Virtual Machine

Data Collector

Data Collector

Data Collector

Cloud Provider

Provider

Developer

REST APIs

Scenario 1:
Infrastructure-level automatic provisioning of VMs and network links.

Scenario 2:
Fast recovery from some sudden failures.

Scenario 3:
Auto-scaling across data centers or Clouds.

Networked infrastructure
- The application workflow becomes more complex with a lot of components that need to communicate with each other. The virtual infrastructure must therefore realize a particular network topology.
- Most current cloud providers cannot support this; for example, Amazon EC2 can only allow users to describe private subnets, making it hard to build a complete topology.

Nearly real-time constraints
- Static constraints (e.g. network transmission time as data is processed), which restrict task scheduling before provisioning.
- Runtime constraints (e.g. on failure recovery time), which affect how applications are managed at runtime, especially where the Cloud is not wholly reliable.

Geography
- In the figure above, not all the components of an application are on the Cloud.
- Virtual infrastructure should be provisioned close to external components in order to satisfy any nearly real-time constraints on data delivery.

Auto-provisioning and federated cloud
- We need a way to provision the whole infrastructure and deploy applications automatically. Currently, some tools can only provision automatically at instance level, for example Chef.
- The application may need more resources from other Clouds to provision a large scale infrastructure. It is a problem to combine these resources across multiple locales however.

Experiment Results

We have conducted experiments to study the impact of our solution architecture on application performance, testing the influence of task placement on latency and bandwidth.

<table>
<thead>
<tr>
<th>Object Number</th>
<th>Subject</th>
<th>CPU Core</th>
<th>OS</th>
<th>Memory</th>
<th>VM</th>
<th>Hourly</th>
<th>Upload</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Laptop</td>
<td>1 GB</td>
<td>MacOS</td>
<td>8 GB</td>
<td></td>
<td>10 Mbps</td>
<td>8.59 Mbps</td>
<td>193 Mbps</td>
</tr>
<tr>
<td>2</td>
<td>Laptop</td>
<td>1 GB</td>
<td>MacOS</td>
<td>8 GB</td>
<td></td>
<td>10 Mbps</td>
<td>8.59 Mbps</td>
<td>193 Mbps</td>
</tr>
<tr>
<td>3</td>
<td>Virtual Machine</td>
<td>1 GB</td>
<td>Ubuntu 14.04</td>
<td>3 GB</td>
<td>Ethernet</td>
<td>15 Mbps</td>
<td>385 Mbps</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Virtual Machine</td>
<td>1 GB</td>
<td>Ubuntu 14.04</td>
<td>3 GB</td>
<td>Ethernet</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

There are two main scenarios we need to compare:
- Scenario “S1” in the figure, is the deployment of all the components in one data centre without use of our agent.
- Scenario “S2” in the figure, is to adopt our solution, which is to distribute the components among multiple locales with our agent.

Application Scenarios

Our provisioning agent can distribute an application workflow across multiple locales while preserving network interconnectivity between tasks. This is done via a choice of methods, and requires no special effort by the application developer or cloud provider.

Experiment Results

We have conducted experiments to study the impact of our solution architecture on application performance, testing the influence of task placement on latency and bandwidth.

Table: Properties of objects used to test performance of linked tasks hosted in different locales.