Automating Resource Selection and Configuration in Inter-Clouds through a Software Product Line Method

Alessandro Ferreira Leite

Paris, September 14th, 2015
Google Cloud Outage: Virtual Networking Breakdown

Google's virtual networking software stopped providing routing updates, and customers lost their connections to the outside world. (http://ubm.io/1XZ0kgx)

The problem with network connectivity in Google Compute Engine is resolved as of shortly after 01:00 US/Pacific. We are sorry for any issues this may have caused to you or your users and thank you for your patience and continued support. Please rest assured that system reliability is a top priority at Google, and we are constantly working to improve the reliability of our systems.

We will provide a detailed analysis of this incident once we have completed our internal investigation. (status.cloud.google.com/incident/compute/15045)
An inter-cloud scenario may help us on dealing with clouds’ outages
There are some challenges to implement an inter-cloud scenario

- Complexity
- Lack of models to describe multiple cloud environments
- Lack of tools that implement automatic resource selection and configuration
- Environment configuration tools still demand programming skills
Why inter-clouds scenarios are complex?

- The users are responsible for:
  - Identify the clouds and the virtual machines types
  - Select the virtual machine images (VMI) in each cloud
  - Deploy the applications, taking into account dependencies and clouds’ constraints
- Even with the pre-configured VMIs configuration involves manual activities (time consuming and error-prone)
- A cloud configuration may demand specialized skills
Example of a networking feature that requires advanced knowledge -- AWS’s enhanced networking feature

Average networking bandwidth (Gbits/sec)

Network throughput when the VM is started with the default parameters

- Public IP: 1.97 Gbits/sec
- Private IP: 5.02 Gbits/sec

Enhanced networking disabled

- Public IP: 7.97 Gbits/sec
- Private IP: 9.33 Gbits/sec

Enhanced networking enabled

Network throughput after enabled a kernel’s driver
A Software Product Line (SPL) is a strategy to design a family of related products using a common architecture, with variations in features.

With a SPL, a single platform can be massively customized to generate a group of products.
SPL relies on feature model to describe the features

- Operating System
  - CentOS
  - Debian
  - Ubuntu

- Processor
  - Ivy Bridge
  - Xeon
  - Sandy Bridge
  - Shared
  - Dedicated

- Processor type
  - One GB
  - Ten GB
  - One TB
  - One hundred GB

- Memory
  - EBS
  - Provisioned

- Storage
  - Server
  - Bootstrap
  - Cluster

- Placement group

Legend:
- Mandatory
- Optional
- Alternative
- Abstract
- Concrete

Constraints:
- c1: Bootstrap \implies Shared \land EBS \land One GB \land One Hundred GB \land \neg Cluster
- c2: Ivy Bridge \lor Sandy Bridge \iff \neg Shared
We developed a SPLE method and a tool to implement the method

- **A Software Product Line Engineering (SPLE) method** for inter-cloud environments. It aims to enable both automatic resource selection and configuration.

- A tool, called Dohko, that implements the proposed method.
Our SPLE Method

1. Create a provider-independent model describing the commonality and variability points of general IaaS clouds.

2. Define an architecture to create the products.

3. Create one concrete model for each target cloud, describing its variation points.

4. Synthesize how each model and the products are instantiated.

5. Selects the models that meet users' requirements.

6. Generates the deployment and configuration plan.

7. Instantiates the selected models.

8. Instantiates the products (deploys and configures the resources).

9. Submits the goals and constraints.

Domain Engineering (At design time)

Product Engineering (At runtime)
Abstract Model

IaaS cloud

Virtual machine

On-demand
Reserved
Spot

Hardware configuration
RAM memory
Network
Compute
General

Instance type
Family type

Accelerator
Memory
Storage
Shared

Virtualization technique
HVM
PVM

Operating system
Platform
Architecture

Region
Africa
Asia
Australia
North America
Europe
South America
Ephemeral
Persistent
Object store
SSD
Standard

Disk type

Disk technology
Requires

Zone

Group

price: integer
maxTps: integer
maxInstances: integer

price: integer

cost: integer
lag: integer

maxSizeGB: integer
minSizeGB: integer
costPerGBMonth: integer

flops: integer
frequency: integer

#vcpu: integer

sizeGB: integer

throughputGbps: integer

ingressCostGB: integer

egressCostGB: integer

Legend:
- Mandatory
- Optional
- Alternative
- Abstract
- Concrete

x64 ↔ b64
zone(disk) ↔ zone(virtual machine) & Zone in zones(Region)
Amazon EC2 Concrete Model

34 instance types x 8 regions
Overview of Dohko

- Implements the runtime part of the SPLE method
- Creates a deployment plan to configure the products across the clouds
- Uses the Configuration Knowledge to execute the actions in the clouds
  - Uses a Constraint Satisfier Problem (CSP) solver to find valid configurations
Dohko’s Architecture
Experimental Setup

* We used to public cloud providers (Amazon EC2 and Google GCE)
* A real bioinformatics application (sssearch36)
* Users’ goals were to minimize monetary cost and maximize resources’ capacity.
Example of an User’s request descriptor (YAML file)

```yaml
1....
2requirements:
3  cpu: 16
4  memory: 90
5  platform: "LINUX"
6  cost: 2.0
7  number-of-instances-per-cloud: 10
8applications:
9  application:
10    name: "ssearch36"
11    command-line: "ssearch36 -d 0 ${query} ${database} >> ${score_table}"
12    file:
13      - name: "query"
14        path: "${HOME}/sequences/060341.fasta"
15        generated: "N"
16      - name: "database"
17        path: "${HOME}/uniprot_sprot.fasta"
18        generated: "N"
19      - name: "score_table"
20        path: "${HOME}/scores/060341.fasta_scores.txt"
21        generated: "Y"
22...
23on-finished: "TERMINATE"
```
## Multiple VMs selection – single and inter-cloud scenarios

<table>
<thead>
<tr>
<th># Req.</th>
<th># vCPU</th>
<th>Memory (GB)</th>
<th>Cost (USD/hour)</th>
<th># VM</th>
<th>Cloud provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>0.2</td>
<td>5</td>
<td>EC2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
<td>0.2</td>
<td>10</td>
<td>GCE</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>6</td>
<td>1.0</td>
<td>5</td>
<td>EC2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>6</td>
<td>1.0</td>
<td>10</td>
<td>GCE</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
<td>1.0</td>
<td>5</td>
<td>EC2 and GCE</td>
</tr>
</tbody>
</table>

### User Request

### Inter-Cloud

<table>
<thead>
<tr>
<th># Req.</th>
<th>Instance type</th>
<th># vCPU</th>
<th>RAM (GB)</th>
<th>Cost ($/hour)</th>
<th>Family type</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>m3.large</td>
<td>2</td>
<td>7.5</td>
<td>0.14</td>
<td>General</td>
<td>EC2</td>
</tr>
<tr>
<td>2</td>
<td>n1-standard-2</td>
<td>2</td>
<td>7.5</td>
<td>0.14</td>
<td>Memory</td>
<td>GCE</td>
</tr>
<tr>
<td>3</td>
<td>c3.xlarge</td>
<td>4</td>
<td>7.5</td>
<td>0.21</td>
<td>Compute</td>
<td>EC2</td>
</tr>
<tr>
<td>4</td>
<td>n1-standard-4</td>
<td>4</td>
<td>15</td>
<td>0.28</td>
<td>General</td>
<td>GCE</td>
</tr>
<tr>
<td>5</td>
<td>c3.xlarge</td>
<td>4</td>
<td>15</td>
<td>0.21*</td>
<td>General</td>
<td>EC2</td>
</tr>
<tr>
<td></td>
<td>n1-standard-4</td>
<td>4</td>
<td>15</td>
<td>0.28</td>
<td>General</td>
<td>GCE</td>
</tr>
</tbody>
</table>

*3 c3.xlarge and 2 n1-standard-4

*Cloud provider*
It takes less than 10 minutes to automatically select and to deploy a real inter-cloud environment.
We proposed a SPLE method and a tool to help the users on instantiating inter-cloud environments.

Our SPLE Method relies on:
- Abstract Feature Model
- Concrete Feature Model
- Configuration Knowledge
- Well-known benchmarks to uniform clouds’ performance description
- A tool that implements the method, and enables automatic inter-cloud configuration
That's all Folks!
Automating Resource Selection and Configuration in Inter-Clouds through a Software Product Line Method

Alessandro Ferreira Leite