Cloud technology for algorithm preservation

Cécile Cavet

cecile.cavet at apc.univ-paris7.fr

Centre François Arago (FACe), Laboratoire APC, Université Paris Diderot
LabEx UnivEarthS

November 5, 2014
Plan

1. Cloud computing

2. StratusLab

3. Algorithm preservation in the Cloud

4. Conclusion
What is Cloud computing?

Provides IT resources on-demand.

Full definition of Cloud Computing: NIST report [1].
What is Cloud computing? A mature project

1. **Hardware**: virtualization of all resources of commodity hardware.
The virtualization revolution

Figure: Principle of virtualization [2].
What is Cloud computing? A mature project

1. **Hardware**: virtualization of all resources of commodity hardware.

2. **Software**: simplified APIs (software interface) to end-user client and Web interface.
C. Cavet  Cloud technology for algorithm preservation
What is Cloud computing? A mature project

1. **Hardware**: virtualization of all resources of commodity hardware.

2. **Software**: simplified APIs (software interface) end-user client and Web interface.

3. **Resources**: excess of commercial computing resources.

   ➔ In 2006: 50% of Amazon’s resources was not used.

   ➔ Now: growing infrastructures (academic and commercial).
What is Cloud computing? Essential characteristics

Figure: Attributes of Cloud computing.

⇒ Huge flexibility for scientific applications.

C. Cavet Cloud technology for algorithm preservation
What is Cloud computing? A clump of Clouds...

Figure: Different models of Cloud infrastructure (from [3]).

We will focus on academic public IaaS Cloud.

C. Cavet
Cloud technology for algorithm preservation
Cloud solution & ressources

Clouds in production phase:

- **StratusLab@LAL** [4] (EU, 2010):
  
  http://stratuslab.eu/index.html

- **OpenStack@CC-IN2P3** (FR, 2012):
  
  http://ccwiki.in2p3.fr/infrastructure:cloud:start

→ Performance benchmarks, tests and scientific applications.

C. Cavet  Cloud technology for algorithm preservation
StratusLab Cloud:

- Cloud manager: **StratusLab** with libvirt (in dec. 2014, [OpenNebula](http://opennebula.org)).
- Resources **LAL@Orsay**: 16 nodes, 440 cores, 772 GB of memory, 15 TB of disk space, 1 GbE/s inter-node connexion.
- Utilization: $\sim 90\%$.
- Services:
  - End-user client ([http://stratuslab.eu/index.html](http://stratuslab.eu/index.html)).
  - Persistent disk Web interface ([https://pdisk.lal.stratuslab.eu/svc-pdisk.html](https://pdisk.lal.stratuslab.eu/svc-pdisk.html)).
  - Ressource monitoring ([https://cloud.lal.stratuslab.eu/load.txt](https://cloud.lal.stratuslab.eu/load.txt)).
How to use the **StratusLab** Cloud

**Figure:** Virtual machine life-cycle @StratusLab.

C. Cavet  
Cloud technology for algorithm preservation
How to use the StratusLab Cloud

- Choice of the OS.
Cloud computing
What is it?
Characteristics
Models
Solutions & ressources

StratusLab
Ressources
How to use a Cloud

MarketPlace
Monitoring
End-user client
Connexion
Persistent disk

Algorithm preservation in the Cloud
Requirements
Tools

Conclusion
References

MarketPlace: share Disk Image

Figure: MarketPlace Website.
C. Cavet  Cloud technology for algorithm preservation
MarketPlace: share Disk Image

- Disk image creation is time consuming:
  - Need to have access to a catalog of images.
  - Need to share disk images in project.
- To boot an image: identifier ➞ NYUUiZmum8Tj77kMhkxbPwxl5tA.
- Cloud dependent.
- Disk images have 6 months of validity ➞ OS update/upgrade for security.
How to use the StratusLab Cloud

- Choice of the OS.
- Choice of the resources.
Monitoring

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>RVM</th>
<th>TCPU</th>
<th>FCPU</th>
<th>ACPU</th>
<th>TMEM</th>
<th>FMEM</th>
<th>AMEM</th>
<th>STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>onehost-31.lal.</td>
<td>7</td>
<td>3200</td>
<td>2896</td>
<td>0</td>
<td>63G</td>
<td>13.1G</td>
<td>3.6G</td>
<td>on</td>
</tr>
<tr>
<td>21</td>
<td>onehost-30.lal.</td>
<td>5</td>
<td>3200</td>
<td>3034</td>
<td>0</td>
<td>63G</td>
<td>9.2G</td>
<td>3.7G</td>
<td>on</td>
</tr>
<tr>
<td>23</td>
<td>onehost-15.lal.</td>
<td>6</td>
<td>2400</td>
<td>2396</td>
<td>400</td>
<td>35.4G</td>
<td>10.7G</td>
<td>3.1G</td>
<td>on</td>
</tr>
<tr>
<td>24</td>
<td>onehost-16.lal.</td>
<td>7</td>
<td>2400</td>
<td>2247</td>
<td>400</td>
<td>35.4G</td>
<td>16.1G</td>
<td>6.9G</td>
<td>on</td>
</tr>
<tr>
<td>25</td>
<td>onehost-17.lal.</td>
<td>6</td>
<td>2400</td>
<td>2396</td>
<td>600</td>
<td>35.4G</td>
<td>12.5G</td>
<td>9G</td>
<td>on</td>
</tr>
<tr>
<td>26</td>
<td>onehost-19.lal.</td>
<td>6</td>
<td>2400</td>
<td>2398</td>
<td>400</td>
<td>35.4G</td>
<td>28.4G</td>
<td>15.6G</td>
<td>on</td>
</tr>
<tr>
<td>27</td>
<td>onehost-20.lal.</td>
<td>4</td>
<td>2400</td>
<td>1299</td>
<td>0</td>
<td>35.4G</td>
<td>17.7G</td>
<td>18.3G</td>
<td>on</td>
</tr>
<tr>
<td>31</td>
<td>onehost-25.lal.</td>
<td>5</td>
<td>3200</td>
<td>2896</td>
<td>500</td>
<td>63G</td>
<td>17G</td>
<td>365.2M</td>
<td>on</td>
</tr>
<tr>
<td>32</td>
<td>onehost-26.lal.</td>
<td>6</td>
<td>3200</td>
<td>3197</td>
<td>1800</td>
<td>63G</td>
<td>23.5G</td>
<td>2.5G</td>
<td>on</td>
</tr>
<tr>
<td>33</td>
<td>onehost-27.lal.</td>
<td>6</td>
<td>3200</td>
<td>3095</td>
<td>0</td>
<td>63G</td>
<td>51.2G</td>
<td>2.5G</td>
<td>on</td>
</tr>
<tr>
<td>34</td>
<td>onehost-28.lal.</td>
<td>6</td>
<td>3200</td>
<td>3098</td>
<td>100</td>
<td>63G</td>
<td>56.5G</td>
<td>23.9G</td>
<td>on</td>
</tr>
<tr>
<td>36</td>
<td>onehost-21.lal.</td>
<td>6</td>
<td>2400</td>
<td>2396</td>
<td>1000</td>
<td>35.4G</td>
<td>11G</td>
<td>21.5G</td>
<td>on</td>
</tr>
<tr>
<td>37</td>
<td>onehost-14.lal.</td>
<td>7</td>
<td>2400</td>
<td>2396</td>
<td>1000</td>
<td>35.4G</td>
<td>19.1G</td>
<td>9.4G</td>
<td>on</td>
</tr>
<tr>
<td>38</td>
<td>onehost-23.lal.</td>
<td>3</td>
<td>2400</td>
<td>2398</td>
<td>600</td>
<td>35.4G</td>
<td>24.1G</td>
<td>77.9M</td>
<td>on</td>
</tr>
<tr>
<td>39</td>
<td>onehost-24.lal.</td>
<td>6</td>
<td>2400</td>
<td>1961</td>
<td>500</td>
<td>35.4G</td>
<td>9.1G</td>
<td>397.9M</td>
<td>on</td>
</tr>
</tbody>
</table>

Figure: Physical servers of the StratusLab Cloud: total and used resources (CPU and memory).
How to use the StratusLab Cloud

- Choice of the OS.
- Choice of the resources.
- Run VM.
End-user client: run Virtual Machine

```
-bash-3.2$ stratus-run-instance --cpu=2 --ram=4096 --swap=2048 HKDKNwTo_j0y305dk
```

```
***
:: Starting machine(s):
***
:: Starting 1 machine
:: Machine 1 (vm ID: 977)
  Public ip: 134.158.75.228
:: Done!
-bash-3.2$ 
```

**Figure:** Command-line client: run a virtual machine.

```
bash-3.2$ stratus-describe-instance
```

```
+----+-----+----+-----+-------------+-----------------+
<table>
<thead>
<tr>
<th>id</th>
<th>state</th>
<th>vcpu</th>
<th>memory</th>
<th>cpu%</th>
<th>host/ip</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>621</td>
<td>Pending</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>vm-236,lal.stratuslab.eu</td>
<td>one-621</td>
</tr>
<tr>
<td>637</td>
<td>Running</td>
<td>8</td>
<td>16777216</td>
<td>1</td>
<td>vm-89,lal.stratuslab.eu</td>
<td>one-637</td>
</tr>
<tr>
<td>895</td>
<td>Running</td>
<td>2</td>
<td>4194304</td>
<td>0</td>
<td>vm-206,lal.stratuslab.eu</td>
<td>one-895</td>
</tr>
</tbody>
</table>

bash-3.2$ 
```

**Figure:** Command-line client: describe the state of virtual machines.

C. Cavet  Cloud technology for algorithm preservation
How to use the **StratusLab** Cloud

- Choice of the OS.
- Choice of the ressources.
- Run VM.
- Connect to VM and run the application.
Connexion

Figure: Connexion to the virtual machine: by SSH without password (key identification) and as the Root user.

- Virtual machine ready in 5min!
- Install all scientific/administration Linux packages as Root.
- Launch scientific applications as User.
- Run your jobs.
How to use the **StratusLab** Cloud

- Choice of the OS.
- Choice of the resources.
- Run VM.
- Connect to VM and run the application.
- Save/Use the data.
Figure: Persistent disk Web interface.
Persistent disk: manage volumes

- Persistent storage (it survives to the VM).
- Not made for long term storage.
  - Not true for commercial Cloud (AWS, OVH...) but expensive and security issues.
  - Exploration of connection with non virtual storage for academic Cloud (OpenStack@CC-IN2P3).
- Cloud dependent:
  - If Cloud computing part (MV) is down/full = unreachable data.
  - Need to transfer data on each infrastructures.
Algorithm preservation in the Cloud: requirements

**Data preservation**: challenge is also preserving software and algorithm.

We need to:

- be able to run the old software avoiding:
  - old computer dependency.
  - compiler errors, missing libraries...
  - re-writing of the code.
- be able to re-run analysis:
  - to re-obtain results.
  - by updating the code to compare new results with previous studies.
Tools from the Cloud

- **Virtualization:**
  - Encapsulation of the algorithms within the environment (OS + packages + libraries).
  - Huge portability.
  - But... no long term studies (hypervisors will change, disk image standards will evolve...) ➔ Need to fix standards.

- **Cloud infrastructure:**
  - Running algorithms on-demand.
  - Huge flexibility.
  - But... solutions are always evolving (and connected tools also).
Tools from the Cloud

- **MarketPlace:**
  - Supermarket for disk images.
  - Ideal for sharing projects.
  - But... only 6 months of validity ➔ We can imagine a MarketPlace only for **long term preservation** of disk images (not for every day applications).

- **Persistent disk:**
  - Possibility to store data in the Cloud.
  - But... no long term preservation ➔ Specific storage in the Cloud? Storage outside the Cloud?
Conclusion

- **New subject**: everything has to be done.
- Cloud is very adaptative to the algorithm preservation problem:
  - Easy and quick access.
  - Flexibility.
  - Based on virtualization.
  - Already developed tools (*MarketPlace*, *Persistent disk*...).
- Future:
  - Growing infrastructures and community.
  - Federated Clouds.
  - Multi-cloud solution.
  - Algorithm preservation?
Thank you for your attention.
References

End-user client installation and full command description: FACe Wiki


