Engineering Cloud Applications 1

Cloud Computing

Dr. Philipp Leitner

University of Zurich, Switzerland
What is Cloud Computing? (1)

- From a user perspective:
  - Software and data is access over the Internet
What is Cloud Computing? (2)

• From a user perspective:
  • Software and data is access over the Internet

• From a software developer perspective:
  • Developer “rents” computing resources
  • Application is running on the cloud provider’s machines
  • Fundamentally about loss of control
Why is CC so popular?

• Let’s assume you are a small(ish) company

• You probably have a Web-based product
• You probably don’t have too much money
“In the past, if your application became popular and your systems or your infrastructure did not scale you became a victim of your own success. Conversely, if you invested heavily and did not get popular, you became a victim of your failure.”

Jinesh Varia, Amazon
Cloud Technology Stack

IaaS  PaaS  SaaS

Less Restrictions

Managed by Client  Managed by Cloud Vendor
Example Services

- **IaaS** (Infrastructure as a Service):
  - Application
  - Data
  - Application Runtime
  - Database
  - Managed by Client

- **PaaS** (Platform as a Service):
  - Application
  - Data
  - Application Runtime
  - Managed by Cloud Vendor
  - EC2

- **SaaS** (Software as a Service):
  - Application
  - Data
  - Managed by Cloud Vendor
  - Salesforce

Higher Abstraction

Less Restrictions

- Managed by Client
- Managed by Cloud Vendor
AWS Lambda
(PaaS as well)
Other Cloud Providers

- Heroku
- IBM Bluemix™
- DigitalOcean
- Rackspace®
- Google Cloud Platform
- SoftLayer®, an IBM Company
- Azure
Basic Cloud Computing Concepts

• Core concepts:
  • Resource virtualization / pay-per-use
  • Elasticity
  • Multi-Tenancy
  • Opaque infrastructure
  • “Illusion of Infinite Resources”
Basic Cloud Computing Concepts

- Core concepts:
  - Resource virtualization / pay-per-use
  - Elasticity
  - Multi-Tenancy
  - Opaque infrastructure
  - “Illusion of Infinite Resources”
Virtualization of Computing Resources

Virtualization is at the core of cloud computing
Network Virtualization

- Software-Defined Networks (SDN)
Virtualization of Storage

- "Ephemeral" Storage
  - Storage on the same physical host
- Block Storage
  - Virtualized NAS
Basic Cloud Computing Concepts

- Core concepts:
  - Resource virtualization / pay-per-use
  - Elasticity
  - Multi-Tenancy
  - Opaque infrastructure
  - “Illusion of Infinite Resources”
“elasticity (or stretchiness) is the physical property of a material that returns to its original shape after the stress (e.g. external forces) that made it deform or distort is removed” – http://en.wikipedia.org/wiki/Elasticity_(physics)
Overprovisioning

[Graph showing capacity and demand with unused resources shaded.]
Underprovisioning
Elastic Provisioning
Real Example (from Segment)

https://segment.com/blog/the-million-dollar-eng-problem/
In practice, how “elastic” one can really be depends on three factors:

- Resource granularity (how fine-grained can we get resources?)
- Billing granularity (“billing time unit”, for instance 1 hour)
- Provisioning time (how long does it take to get and release resources?)

→ all of those depend on the service!
Basic Cloud Computing Concepts

• Core concepts:
  • Resource virtualization / pay-per-use
  • Elasticity
  • Multi-Tenancy
  • Opaque infrastructure
  • “Illusion of Infinite Resources”
Multi-Tenancy

- Resources are shared among many independent parties
  - *e.g.*, the physical host that runs your application also hosts VMs for Netflix

- Fundamentally how the cloud makes money
  - Allows cloud to efficiently use its physical infrastructure and leverage economies of scale
Main Issues

- "Noisy Neighbours" (resource inference)
  - Not all computational resources can be cleanly divided
  - Most importantly: I/O and network access

- "Nosy Neighbours" (information leakage)
  - E.g., VM placement attacks
Potential Fixes

- “Dedicated” instances are guaranteed to be only co-located with resources of the same tenant
  - Not the same as non-virtualized resources

- **Bare Metal Cloud**
  - Non-virtualized resources that are served on demand
Basic Models

**IaaS vs. PaaS vs. SaaS**
What level of abstraction is provided by the cloud?

**Public vs. Private vs. Hybrid Cloud**
Who is hosting the cloud?
So far we have always assumed that “cloud computing” means public cloud.

The entity providing the cloud is separate from the consumer.

Private cloud:

- Use cloud principles on your own hardware
- Largely out of scope in this lecture
Cloud Bursting
A hybrid cloud system is a system that makes use of multiple clouds at the same time
(typically a combination of private and public clouds)
Cloud Bursting
Advantages

• Best of **two worlds**:
  
  • (1) application mostly runs on top of (cheap, fast and relatively reliable) internal data centre —> private cloud
  
  • (2) load spikes can be covered by resources from a public cloud provider —> public cloud
Cloud Bursting
Disadvantages

• **Technical issues:**
  • Network latency between instances in your private cloud and those in the public cloud is very large
  • Technical capabilities of the different clouds may be different

• **Legal issues:**
  • Oftentimes, not all data is allowed to be transferred to a public cloud
Opportunities / Benefits

Cost- and Product-Related

- Focus on Product, Productivity: 10%
- Faster Delivery, Time-to-market: 10%
- Lower Costs/TCO (Total Cost of Ownership): 11%
- Automation, Deployment Pipelines: 22%
- Easier Infrastructure Setup and Maintenance: 30%
- Scalability / Elasticity: 31%

(survey responses of 294 cloud developers)

Business Opportunities

- Lower TCO (presumably)
- CapEx -> OpEx
  - Pay little as long as you don’t have users
- DevOps / NoOps
  - Less administration, more building features
- Business Agility
  - Faster feedback, continuous delivery
TCO

• “Total Costs of Ownership”
  • Acquisition + Operation + Change
  • Costs of buying the hardware
  • + Operational costs (utilities, admin salaries)
  • + Maintenance and repair
## Simple Calculation Example

<table>
<thead>
<tr>
<th>$ in 1,000s</th>
<th>Acquisition Costs</th>
<th>Operating Costs</th>
<th>Change Costs</th>
<th>Total</th>
<th>% of TCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>274</td>
<td>82</td>
<td>138</td>
<td>494</td>
<td>2.9%</td>
</tr>
<tr>
<td>Hardware</td>
<td>539</td>
<td>97</td>
<td>71</td>
<td>707</td>
<td>4.1%</td>
</tr>
<tr>
<td>Personnel</td>
<td>55</td>
<td>8,873</td>
<td>5,952</td>
<td>14,879</td>
<td>86.2%</td>
</tr>
<tr>
<td>NW &amp; Comm</td>
<td>146</td>
<td>543</td>
<td>459</td>
<td>1,149</td>
<td>6.7%</td>
</tr>
<tr>
<td>Facilities</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>29</td>
<td>0.2%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1,104</strong></td>
<td><strong>9,610</strong></td>
<td><strong>6,634</strong></td>
<td><strong>17,258</strong></td>
<td>—</td>
</tr>
<tr>
<td><strong>% of TCO</strong></td>
<td><strong>5.9%</strong></td>
<td><strong>55.7%</strong></td>
<td><strong>38.4%</strong></td>
<td>—</td>
<td><strong>100.0%</strong></td>
</tr>
</tbody>
</table>

CapEx -> OpEx

- Acquisition costs -> **capital expenses** (CapEx)
  - Requires money upfront
  - Cashflow problem (especially for startups)
- Through pay-per-use, you essentially have no CapEx anymore, only operational expenses (OpEx)
- Compare: car or personnel leasing
Opportunities / Benefits


![Chart showing percentages of cloud developers' responses]

- Focus on Product, Productivity: 9%
- Faster Delivery, Time-to-market: 10%
- Lower Costs/TCO (Total Cost of Ownership): 22%
- Automation, Deployment Pipelines: 22%
- Easier Infrastructure Setup and Maintenance: 30%
- Scalability / Elasticity: 31%

(survey responses of 294 cloud developers)
Automation

- Central concept in IaaS:
  - Instances are volatile
  - No data stored on instance
  - Can be terminated any time
  - Setup completely automated

"One interesting thing that is very cloud specific and influenced our architecture is, that the cloud provider tells you, we can kill your machine any time we want."
-P9IaaS

"We have now moved to strict Immutable Infrastructure in our deployment. We don’t even put SSH keys into instances anymore, making changes to existing infrastructure impossible.” -P12IaaS

Example Technology - Opscode Chef

Interacts with the IaaS cloud via API

```
knife ec2 server create \
  --availability-zone us-east-1d \
  --node-name redis.learnchef.demo \
  --flavor t1.micro \
  --image ami-fd20ad94 \
  --identity-file ~/.ssh/aws.pem \
  --run-list "role[redis]" \
  --ssh-user ubuntu
```

Configures the server via Ruby code

```ruby
name 'redis'
description 'A redis master on slave (on a single machine)'
default_attributes({
  'redisio' => {
    'servers' => [
      { 'port' => '6379' },
      { 'port' => '6380', 'slaveof' => { 'address' => '127.0.0.1', 'port' => '6379' } }
    ]
  }
})
run_list(
  'recipe[redisio::install]',
  'recipe[redisio::enable]'
)
```
• (Public) cloud computing is about doing your computing on somebody else’s hardware
  • —> Loss of control

• Basic models:
  • IaaS vs. PaaS vs. SaaS
  • Private vs. Public Cloud
• Core advantages are **technical** and **economical**

• **Economical advantages:**
  • Reduced TCO
  • No investment necessary

• **Technical advantages:**
  • Elasticity
  • Automation