Distributed Data Management
Summer Semester 2013
TU Kaiserslautern

Dr.-Ing. Sebastian Michel

smichel@mmci.uni-saarland.de
Lecture 8

CLOUD COMPUTING
Imagine

• You are developing a cool new app
• Testing seems promising
• Friends love it!
• Let’s go public/viral!

Distributed Data Management, SoSe 2013, S. Michel
Start Up Cost

• How much servers do we need?
• Want to handle system also at peak time?!
• The young startup Animoto went from 50 to 3,500 machines in few days.
• What if demand drops? What do to?
Cloud Computing

• Outsource computation, data storage, applications to run on machines of a (cloud) service provider: Hosted services.

• Pay-as-you-go renting of services; hardware, tools or end-user software

• High utilization of services through virtualization and resource sharing

• Distributed computing (how and where is hidden to end user)

• Unclear origin of term “cloud”
Sharing Resources/On Demand Access

• In the 1960s, several companies started providing time-sharing services as service bureaus.

• Shared (rented) via remote login through modem or direct use. Hardware was expensive.

• Later: HW affordable. Ok, if machine was idle.
SaaS, PaaS, and IaaS

• Grouping of provided services in categories

• Ranging from high level (full apps) to low level services (like storage)

• For end users, application developers, and system architects, respectively.

Software as a Service (SaaS)
email, games, word processing, ...

Platform as a Service (PaaS)
Database system, NoSQL store, web server, ...

Infrastructure as a Service (IaaS)
Storage, Servers, Queuing, Load Balancing, ....
Examples of Large Cloud Platforms/Providers

• Microsoft Azure:
  http://www.windowsazure.com

• Amazon Web Service (AWS):
  http://aws.amazon.com/

• Google Cloud Platform:
  https://cloud.google.com/

Each comes with whole suite of different services.
Example: Amazon Web Services

For instance, have seen in this lecture already NoSQL Data Store, MapReduce (Hadoop)
Cloud Computing Promises

• Cloud services offer on-demand availability of servers, software, services.
• No or little startup cost for your (young) business; “pay as you go”
• Don’t have to administrate hardware
• Elasticity: rent more when you need more, give back instances when not needed (so you don’t pay for it) Both for hardware and software!
CC Promises (Cont’d)

• **Fault tolerant** services (and you don’t have to care about it, just pay)
• **Availability.** In general, through **Service Level Agreements** (SLAs)
• There are **prominent showcases** for scalability (e.g., Animoto) on Amazon’s Elastic Compute Cloud (EC2)
Potential Threats

• **Privacy**
  – your (or your customer data) is (entirely) stored at provider

• **Dependency** on cloud provider
  – non physical control over hardware
  – what if foreign government decides to shut down (their) cloud?
  – or to force provider to unveil data / usage information of your customers?
  – what if their datacenter breaks down?

• Somewhat bound to provider (aka. vendor lock-in) by **tailoring solutions** to specific (non standard) services.

• Have to be **constantly online** for SaaS
Infrastructure as a Service (IaaS)

- Provides low level services, like simply hardware (machines).
- Virtual machines
- Cost depends on CPU, RAM, disk space
- Pay by hour (time)

- Amazon EC2  *(will have a closer look later)*
- Google compute engine
- Microsoft Azure services platform
Realization/Paradigms

• Use of virtualization: Putting several virtual machines (VMs) on physical instances

• Aims at high utilization of physical machines

• Provided services play together. E.g., load balancing and dynamic starting of new machines. EC2 instances and storage/tools. S3 storage and Hadoop MapReduce.
Virtualization

- Guest OS is running on
- Hardware that is virtualized
Virtualization Variants

- Hypervisor runs directly on machine without full-fledged OS
- or Hypervisor runs in OS

https://en.wikipedia.org/wiki/Hypervisor
ParaVirtualization

• Avoid virtualization overhead for access of guest OS to underlying physical hardware
• Hypervisor provides API that guest OS can call instead of fully simulating “hardware”
• For this, OS code is adapted
• Increased efficiency (toward near-native performance)
• VMs with paravirtualization available for Linux (as OS needs to be modified/adapted)

• See also hardware support for virtualization.
Amazon EC2

- **Elastic Compute Cloud** (EC2)
- Renting (virtual) machine instances. Choosing from several AMIs: Amazon Machine Images
- Can choose between different operating systems, and “hardware” configurations
  - RAM
  - CPU (cores)
  - Disk
- Use pre-defined OS images or tailor your own one
Request Instances Wizard

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

**Number of Instances:** 1  
**Instance Type:** C1 High-CPU Medium (c1.medium, 1.7 GiB)

Launch as an EBS-Optimized instance (additional charges apply):

- Not supported for this instance type

**Launch Instances**

EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

**Launch into:**
- EC2-Classical
- EC2-VPC

**Availability Zone:** eu-west-1a

**Request Spot Instances**
Time vs. Money / Pricing Policies

- Choice of computing tools/machines gives tradeoff between time and money
- “Exchange” time on y-axis with $.
- Example: Amazon EC2

### On-Demand Instance Prices

<table>
<thead>
<tr>
<th>Region: US East (N. Virginia)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux/UNIX Usage</strong></td>
</tr>
</tbody>
</table>

#### Standard On-Demand Instances

- Small (Default): $0.060 per Hour
- Medium: $0.120 per Hour
- Large: $0.240 per Hour
- Extra Large: $0.480 per Hour

#### Second Generation Standard On-Demand Instances

- Extra Large: $0.500 per Hour
- Double Extra Large: $1.000 per Hour

#### Micro On-Demand Instances

- Micro: $0.000 per Hour

As of June 12, 2013
## Pricing: VMs at Azure

**Standard Instances**

Provide optimal set of compute, memory and IO resources for running a vast array of applications. Detailed configuration of the instances is available [here](http://www.windowsazure.com/en-us/pricing/details/virtual-machines/).

<table>
<thead>
<tr>
<th>COMPUTE INSTANCE NAME</th>
<th>VIRTUAL CORES</th>
<th>RAM</th>
<th>PRICE PER HOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra Small (A0)</td>
<td>Shared</td>
<td>768 MB</td>
<td>$0.02 (~$15/month)</td>
</tr>
<tr>
<td>Small (A1)</td>
<td>1</td>
<td>1.75 GB</td>
<td>$0.06 (~$45/month)</td>
</tr>
<tr>
<td>Medium (A2)</td>
<td>2</td>
<td>3.5 GB</td>
<td>$0.12 (~$89/month)</td>
</tr>
<tr>
<td>Large (A3)</td>
<td>4</td>
<td>7 GB</td>
<td>$0.24 (~$179/month)</td>
</tr>
<tr>
<td>Extra Large (A4)</td>
<td>8</td>
<td>14 GB</td>
<td>$0.48 (~$357/month)</td>
</tr>
</tbody>
</table>

* Based on 744 hours per month

Prices above are effective as of April, 16th, 2013.
EC2 Storage Possibilities

• **EBS: Elastic Block Storage**
  – block-level storage
  – attached to a single EC2 instance
  – durable/replicated
  – can create snapshots of content (to S3)

• **Physical hard disk:**
  – not durable (after end of EC2 instance)

• **S3:**
  – “highly durable, highly available” object store
Amazon S3 Storage

- **Simple Storage Service (S3)**
- Stores objects (up to several TB) organized in buckets
- Accessed via simple HTTP/SOAP interfaces (and BitTorrent for downloading)

- **Charge on GB/month traffic** consumption for outgoing traffic.
- Several datasets available on demand (Google n-grams, Million song dataset, Wikipedia traffic, ...)

Distributed Data Management, SoSe 2013, S. Michel 23
S3 Pricing

• For region EU (Ireland) (others might vary slightly)
• For standard storage
  – First 1 TB/month: $0.095 per GB, ...
  – Over 5000 TB/month: $0.055 per GB
• Request pricing:
  – PUT, COPY, POST, or LIST requests: $0.005 per 1000 requests, ...
• Data transfer:
  – upload: free, download to EC2 (same region), free
  – download to diff. region not free
  – same as to Internet (first 1GB free, then e.g., <10TB/month= $0.120 per GB

numbers as of June 19, 2013
Service Level Agreements

• From the perspective of clients (users, application developers and providers) of cloud services
  – crucial to understand performance of cloud services
  – predictability is key
  – kind of contract between cloud provider and customer
Service Level Agreements (2)

• Description of (minimum) of service qualities
• In terms of query response time, throughput, availability (mean time between failures; mean time to recovery)
• Usually defines as percentiles, 99.99% of all requests are within 300ms.

Do you known the story of that famous statistician, who was found drowned in a lake of an average depth of 10 centimeters?
Example: SLAs for Amazon S3

“AWS will use commercially reasonable efforts to make Amazon S3 available with a Monthly Uptime Percentage (defined below) of at least 99.9% during any monthly billing cycle (the “Service Commitment”). In the event Amazon S3 does not meet the Service Commitment, you will be eligible to receive a Service Credit as described below.”

- Get service credits for “bad” performance of S3

<table>
<thead>
<tr>
<th>Monthly Uptime Percentage</th>
<th>Service Credit Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal to or greater than 99% but less than 99.9%</td>
<td>10%</td>
</tr>
<tr>
<td>less than 99%</td>
<td>25%</td>
</tr>
</tbody>
</table>

http://aws.amazon.com/s3-sla/
SLAs at Microsoft Azure

• For instance on Storage: *We guarantee that at least 99.9% of the time* we will successfully process correctly formatted requests that we receive to add, update, read and delete data. *We also guarantee that your storage accounts will have connectivity to our Internet gateway.*

• Cloud Services, Virtual Machines, ...: For Cloud Services, we guarantee that when you deploy two or more role instances in different fault and upgrade domains, your Internet facing roles will have external connectivity at least 99.95% of the time.

Higher Level Services

• Beyond IaaS there is demand for higher level services in form of PaaS and SaaS.

• More and more integrated in everyday application like Amazon’s cloud player on smartphones.
Platform as a Service (PaaS)

• Like:
  – Google App Engine
  – Microsoft Azure
  – Amazon Beanstalk

• Allow deployment of applications on top of platform services, like Apache Tomcat, for Java/JSP apps, or in general Apache with PHP, Ruby, etc.
Elastic MapReduce

- Run your Hadoop code in the Cloud
- Few clicks only
- Upload jar and data to the S3 storage
- Tell how many instances and what kind you want
- Find results in S3
Specify the master, core and task nodes to run your job flow. For more than 20 instances, complete the limit request form.

**Master Instance Group:** This EC2 instance assigns Hadoop tasks to core and task nodes and monitors their status.

- **Instance Type:** Small (m1.small)
- **Request Spot Instance**

**Core Instance Group:** These EC2 instances run Hadoop tasks and store data using the Hadoop Distributed File System (HDFS). Recommended for capacity needed for the life of your job flow.

- **Instance Count:** 2
- **Instance Type:** Small (m1.small)
- **Request Spot Instances**

**Task Instance Group (Optional):** These EC2 instances run Hadoop tasks, but do not persist data. Recommended for capacity needed on a temporary basis.

- **Instance Count:** 0
- **Instance Type:** Small (m1.small)
- **Request Spot Instances**
Software as a Service (SaaS)

- Renting/Using software like in:
  - Google Apps
  - Microsoft Office 365
  - Email services

- Aim: Same as for hardware reduce cost of ownership
Amazon Mechanical Turk (AMT)

• Not really related to the aforementioned cloud services but still somehow relevant for making use of “services” on demand.
• “Marketplace for work”
• How does it work?
  – You can rent human workers to do tasks online
  – You can earn money (or gift cards) by solving assigned tasks

https://www.mturk.com/mturk/welcome
HITs

• Human Intelligence Tasks

• Usually small (easy) tasks solvable by non-domain experts

• Payment of (usually) a few cents per task.

• Requires counter measures for finding (and not paying) cheaters (often so called “honey pots” or “trap questions” where the answer is known)
Extract items from shopping receipt

- **Is the image a fake or manipulated receipt?**
  - NO, Image is a real physical receipt

- **Can you read the store, items, & total on receipt?**
  - YES

- **Is the store on the receipt: "City Target"?**
  - YES

Transcribe all items, coupons, & voids:

<table>
<thead>
<tr>
<th>#</th>
<th>Type</th>
<th>Item Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Item</td>
<td>EXAMPLE DESCRIPTION</td>
<td>6.99</td>
</tr>
<tr>
<td>2.</td>
<td>Item</td>
<td>CLOROX BLEACH</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Item</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Item</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Item</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is a preview of hit, click ACCEPT HIT above to complete

Distributed Data Management, SoSe 2013, S. Michel
Summary Cloud Computing

• Basic techniques and paradigms behind Cloud Computing are already some days old
• General concept: pay-as-you-go access to services (infrastructure, platform, software)
• Promises smooth startup cost, elasticity/adaptation to load (pay only what you need)
• Also downsides or points to carefully think about: vendor lock-in, privacy, control