CSci 5105

Introduction to Distributed Systems

Cloud Computing
Today

• Cloud computing
  – 50K feet

• Berkeley “Above the Clouds” paper
The “Standard” Cloud

Data in

Computation

Results out

“No limits”
- Storage
- Computing
Virtual Containers

Provide Isolation
What is the cloud?
NIST Definition

• On-demand self-service
• Broad network access ~ private?
• Resource pooling (multi-tenancy)
• Rapid elasticity
• Measured service (not yet)
Berkeley Definition

• Utility computing, but “new”:
  – Unlimited resources on-demand
  – No up front commitment
    • Other than trust 😊
  – Pay for use

• Not clear if this definition is crisp enough
Appealing Features

• Scale/consolidation
  – elasticity, lower TCO

• Packaging
  – pour all tools/code/data into a virtual container

• Always On
  – high availability
More Appealing Features

• Strong locality
  – data and computing => great for analytics

• Novel sharing platform
  – data/state and applications => gaming, Web 2.0, social computing, scientific applications
Interesting Facts: Pro/Con

• “Lock-in” is real:


PhotoWorks is closing
but the good news is we’ve found a great place for your photos

• Growth:
  – Amazon adds new computing/day at rate of full Amazon data-center circa 2000
  – YouTube content upload in the last 2 months > content broadcast by ABC & NBC since 1948
Who is Using the Cloud and How?

• Government
  – GSA FedRamp Cloud: consolidate 1000s of DCs
    => cost reduction

• National Labs
  – DOE Magellan (LBL, ANL): computational biology => packaging

• Commercial
  – NY times: converted all microfiche to pdf on Amazon EC-2 in one day
    => reduced TCO
Cloud Landscape

• **IaaS:** Infrastructure as a Service
  • provisions resources within provider's infrastructure upon which they can deploy and run arbitrary software (OS, apps, networking, storage)

• **PaaS:** Platform as a Service
  • can create custom applications using programming tools supported by the provider and deploy them onto the provider's cloud infrastructure

• **SaaS:** Software as Service
  • use provider’s applications running on provider's cloud infrastructure
Intuitive Examples

• IaaS
  – facilities to create and deploy a VM, access storage blobs

• PaaS
  – language to program application; components of which get mapped to resources

• SaaS
  – web interface to an application running in the cloud
Cloud Landscape: User-View

• Assembly, low-level clouds IAAS (x86/IA32)
  – want control of all details
  – raw infrastructure => Amazon EC-2/S3
  – few “power” users or groups

• Medium-level clouds PAAS (Java/C)
  – hides raw infrastructure
  – programmable via restricted language =>
    Google App Engine, Microsoft Azure
  – more users, more restrictive
Cloud Landscape: User-View

• High-level “clouds” (SQL service)
  – hides “cloud” itself
  – focus is on user-facing applications
  – Gmail, Gmaps, DropBox, FaceBook, ...
  – most users
  – greatest growth potential
Service Model Architectures

Software as a Service (SaaS) Architectures

Platform as a Service (PaaS) Architectures

Infrastructure as a Service (IaaS) Architectures
Characteristics of IaaS

• Utility computing and billing model
• Automation of administrative tasks
• Dynamic scaling
• Policy-based services
• Internet connectivity
IaaS Example: Amazon Cloud

+ SimpleDB, Glacier, ....
IaaS Example: Amazon Cloud (cont’d)

• Amazon cloud components
  – Elastic Compute Cloud (EC2)
  – Simple Storage Service (S3)
  – Elastic block storage (EBS)
  – SimpleDB

• Some Features
  – Load balancing, auto-scaling, monitoring
  – Availability zones
  – Elastic IP addresses
Research Side

• Cloud toolkits
  – Eucalyptus (UCSB)
  – Globus/Nimbus (ANL)

• Infrastructure
  – FutureGrid
  – OpenCirrus
PaaS Example: Google App Engine

• Service that allows user to deploy user’s Web applications on Google's very scalable architecture

• Providing user with a sandbox for user’s Java and Python applications that can be referenced over the Internet

• AppEnginecode
PaaS Example: Windows Azure

Three core components: *Compute, Storage* and *Fabric*

---

**Compute**

**Storage**

**Fabric**

---

azure programming
Google SaaS Cloud

• Office software => Service
  – Gmail
  – Google docs
  – Google video
  – Google sites
  – ...

• 500,000+ organizations use Google Apps

• GE moved 400K desktops from Office to Google Apps
Cloud Landscape: System-View

Cloud Computing Types

Ad-hoc cloud
4 Cloud Deployment Models

• Private cloud
  – enterprise owned or leased

• Community cloud
  – shared infrastructure for specific community

• Public cloud
  – sold to the public, mega-scale infrastructure

• Hybrid cloud
  – composition of two or more clouds
The NIST Cloud Definition Framework

**Deployment Models**
- **Private Cloud**
- **Community Cloud**
- **Public Cloud**

**Service Models**
- **Software as a Service (SaaS)**
- **Platform as a Service (PaaS)**
- **Infrastructure as a Service (IaaS)**

**Essential Characteristics**
- On Demand Self-Service
- Broad Network Access
- Resource Pooling
- Rapid Elasticity
- Measured Service

**Common Characteristics**
- Massive Scale
- Homogeneity
- Virtualization
- Resilient Computing
- Geographic Distribution
- Service Orientation
- Advanced Security

Based upon original chart created by Alex Dowbor - http://ornot.wordpress.com
Why?
Why? Resource Efficiency

- Pay by use instead of provisioning for peak

Static data center vs. Data center in the cloud

Unused resources
Why? Resource Efficiency (cont’d)

• Risk of over-provisioning: underutilization
Why? Resource Efficiency (cont’d)

- Risk of under-provisioning
Why? Applications

• New Application Opportunities
  – Mobile and web applications
    • Mobile devices: low memory & computation power
  – Extensions of desktop software
    • Matlab, Mathematica
  – Batch processing
    • The New York Times used 100 Amazon EC2 instances to recognize 4TB of raw TIFF image into 1.1 million PDFs in 24 hours ($240)
  – Big Data!!
## Berkeley Challenges

<table>
<thead>
<tr>
<th>Obstacle</th>
<th>Opportunity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability of Service</td>
<td>Use Multiple Cloud Providers to provide Business Continuity; Use Elasticity to Defend Against DDOS attacks</td>
</tr>
<tr>
<td>Data Lock-In</td>
<td>Standardize APIs; Make compatible software available to enable Surge Computing</td>
</tr>
<tr>
<td>Data Confidentiality and Auditability</td>
<td>Deploy Encryption, VLANs, and Firewalls; Accommodate National Laws via Geographical Data Storage</td>
</tr>
<tr>
<td>Data Transfer Bottlenecks</td>
<td>FedExing Disks; Data Backup/Archival; Lower WAN Router Costs; Higher Bandwidth LAN Switches</td>
</tr>
<tr>
<td>Performance Unpredictability</td>
<td>Improved Virtual Machine Support; Flash Memory; Gang Scheduling VMs for HPC apps</td>
</tr>
<tr>
<td>Scalable Storage</td>
<td>Invent Scalable Store</td>
</tr>
<tr>
<td>Bugs in Large-Scale Distributed Systems</td>
<td>Invent Debugger that relies on Distributed VMs</td>
</tr>
<tr>
<td>Scaling Quickly</td>
<td>Invent Auto-Scaler that relies on Machine Learning; Snapshots to encourage Cloud Computing Conservationism</td>
</tr>
<tr>
<td>Reputation Fate Sharing</td>
<td>Offer reputation-guarding services like those for email</td>
</tr>
<tr>
<td>Software Licensing</td>
<td>Pay-for-use licenses; Bulk use sales</td>
</tr>
</tbody>
</table>
Evolution: Drivers

• Economics
  – power and cooling “green clouds”

• Technology
  – devices: smart phones, ipods, tablets, sensors

• Big data
  – 4th paradigm for scientific inquiry

• Privacy/trust
  – local clouds