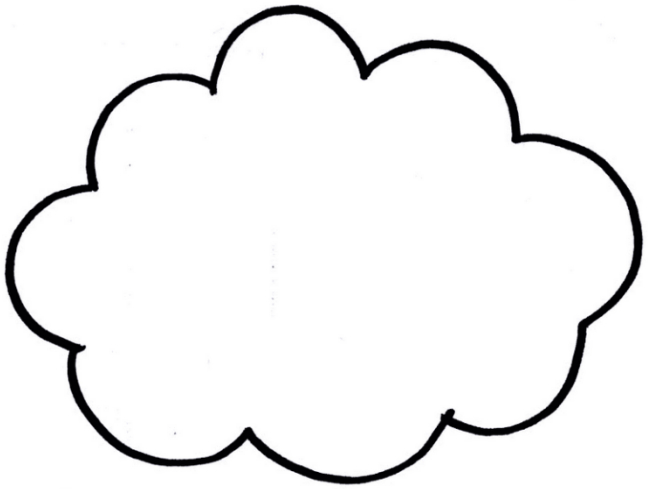
Above the Clouds: A Berkeley View of Cloud Computing

February 10, 2009

Authors: Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zahari



Avner May
September 14, 2011
COMS-E6998-7

Outline

- Paper goes over 8 questions (top pg 4), which I'll distill down to:
  - What is Cloud Computing? (basic terms)
  - Why does it make sense economically?
    - Cloud provider
    - Cloud user
  - How can we classify current cloud offerings?
  - Forward-looking
    - 10 obstacles/opportunities to Cloud Computing
    - What new types of applications does it enable?
    - How should software/hardware change as a result of the Cloud?

Outline

- Paper goes over 8 questions (top pg 4), which I'll distill down to:
  - What is Cloud Computing? (basic terms)
  - Why does it make sense economically?
    - Cloud provider
    - Cloud user
  - How can we classify current cloud offerings?
  - Forward-looking
    - 10 obstacles/opportunities to Cloud Computing
    - What new types of applications does it enable?
    - How should software/hardware change as a result of the Cloud?

What is Cloud Computing?

"Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services."

Important Terms

- **“Software as a Service”** (“SaaS”): Applications delivered as services over the internet (this has existed long before cloud computing)
- **"Cloud"**: A datacenter's hardware and software.
- **"Public Cloud"**: When a Cloud is made available on a pay-as-you-go manner to the general public.
- **"Utility Computing"**: This is the term for the service being provided by Cloud providers.
- **"Private Cloud"**: Internal datacenters of a business or organization (not available to public)
- **"Cloud Computing"** = "Saas" + "Utility Computing" --- eg, when software is provided as a service on the public cloud. (doesn't include "Private Cloud")

** Cloud ≠ SaaS because SaaS can be hosted on private datacenter

Outline

- What is Cloud Computing? (basic terms)
- Why does it make sense economically?
  - Cloud provider
  - Cloud user
- How can we classify current cloud offerings?
- Forward-looking
  - 10 obstacles/opportunities to Cloud Computing
  - What new types of applications does it enable?
  - How should software/hardware change as a result of the Cloud?

Why become Cloud Provider? Economies of Scale

- Large vs Medium DC (5x-7x cost difference for resources)
  - Resources: Network, storage, administration
- Electricity and cooling (up to 1/3 cost of DC): highly variable based on **location**
- Additionally, Cloud Provider can use statistical multiplexing among a large group of customers to increase utilization of their resources.
- This allows price point to exist where being a Cloud Provider is profitable.

Table 2: Economies of scale in 2006 for medium-sized datacenter (≈1000 servers) vs. very large datacenter (≈50,000 servers). [24]

Technology	Cost in Medium-sized DC	Cost in Very Large DC	Ratio
Network	$95 per Mbit/sec/month	$13 per Mbit/sec/month	7.1
Storage	$2.20 per GByte / month	$0.40 per GByte / month	5.7
Administration	≈140 Servers / Administrator	>1000 Servers / Administrator	7.1

Table 3: Price of kilowatt-hours of electricity by region [7].

Price per KWH	Where	Possible Reasons Why
3.6¢	Idaho	Hydroelectric power; not sent long distance
10.0¢	California	Electricity transmitted long distance over the grid; limited transmission lines in Bay Area; no coal fired electricity allowed in California.
18.0¢	Hawaii	Must ship fuel to generate electricity

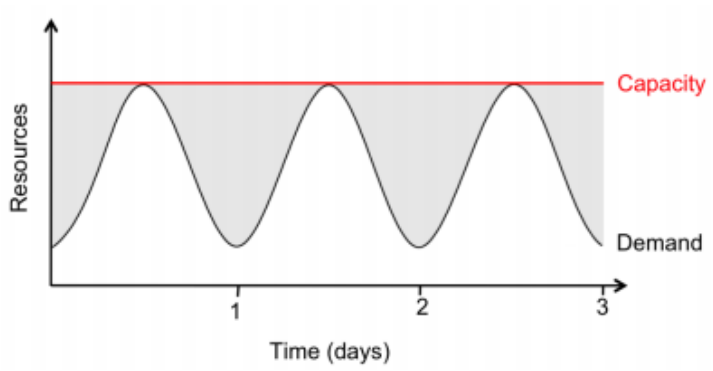
Outline

- What is Cloud Computing? (basic terms)
- Why does it make sense economically?
  - Cloud provider
  - Cloud user
- How can we classify current cloud offerings?
- Forward-looking
  - 10 obstacles/opportunities to Cloud Computing
  - What new types of applications does it enable?
  - How should software/hardware change as a result of the Cloud?

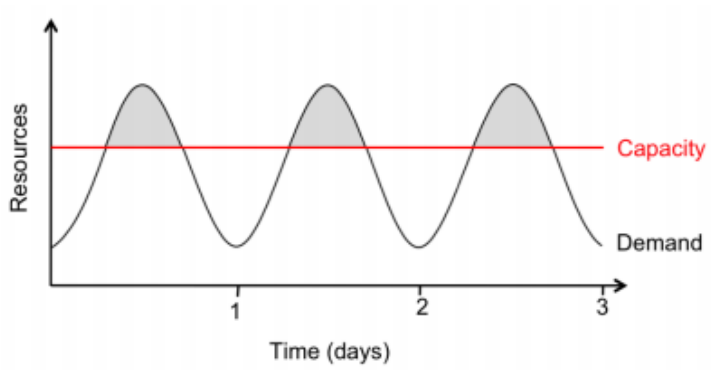
Why move to Cloud?

- Basic Idea
  - Cloud
    - No upfront costs
    - Pay-as-you-go (scale up and down automatically)
    - Consume computational resources as a utility
  - Private datacenter
    - Upfront costs to setting up datacenter (risky! What if startup fails?)
    - Provisioning issues: How many servers should you buy?
      - Provision correctly --> waste (allocate enough resources for peak demand, but these resources then lie idle during calmer hours). The higher the peak to mean ratio, the more waste there is. Note: daily + seasonal + unpredictable peaks…how provision?
      - Overprovision --> more waste
      - Underprovision --> miss out on potential revenue, potentially losing customers long-term due to poor service.

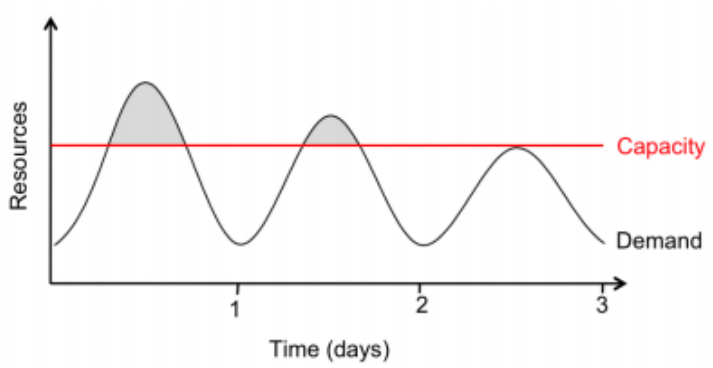
Provisioning Issues



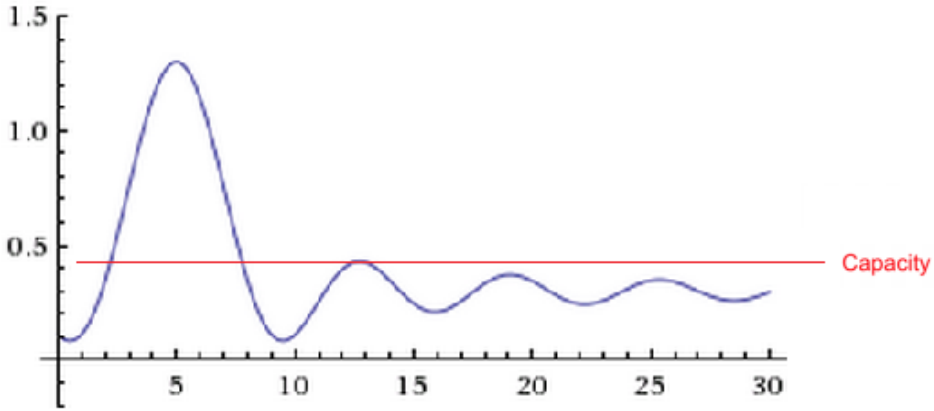
(a) Provisioning for peak load



(b) Underprovisioning 1



(c) Underprovisioning 2



(d) Underprovisioning 3 (unexpected peak)

Cloud Economics

$$\text{UserHours}_{\text{cloud}} \times (\text{revenue} - \text{Cost}_{\text{cloud}}) \geq \text{UserHours}_{\text{datacenter}} \times \left(\text{revenue} - \frac{\text{Cost}_{\text{datacenter}}}{\text{Utilization}}\right)$$

Automatic Scalability! (accommodates peaks and troughs)

- Absorbs changing costs
- Pay per resource
- Benefits from economies of scale, low power costs… (see next slide…)

Could be made smaller if underprovision

- Always < 1 due to idle resources and physical limitations of hardware (.6-.8)
- Highly variable based on peaks/troughs and provisioning choices

Note: Revenue and Costs are hourly
Cost_datacenter = amortized cost of hardware over 3 years, per server-hour (assuming 100% utilization at all times)

Cloud Economics
Summary

Pros (Cloud)

- No upfront costs
- Elasticity (variable demand)
  - Transfer Risk of misprovisioning
- Faster if parallel! (cost associativity)
- As hardware/software improves, get these benefits automatically (more/better services, lower prices)
- Possibly cheaper
  - Pay per resource, as used (resource costs change over time. Get these benefits)
  - Lower operation costs

Cons (Cloud)

- Data transfer time + costs
- Migration costs
- See 10 obstacles…

Cloud Economics
Summary

Pros (Private Datacenter)

- Potentially lower cost (at least per server-hour, ignoring utilization issues)
- Computation right beside data (no latency or data transfer costs)
- Security?

Cons (Private Datacneter)

- Upfront cost
- Risk of over/underutilization, not provisioning correctly for surges in demand, **waste of idle resources…**
- Inability to scale down
- Limits on utilization ratio of resources (.6 to .8 max)
- Slow to absorb improvements in technology
- Operation costs
- Power, cooling, and physical plant costs

Discussion (Economic Model)

- Does it express the main trade offs well?
- Does it fail to include some costs/benefits which in practice affect the decision to move to the cloud?
- Are its assumptions valid? (eg, revenue proportional to user hours)
- Is it reasonable for companies considering private datacenter to treat cost_datacenter as hourly cost?

Outline

- What is Cloud Computing? (basic terms)
- Why does it make sense economically?
  - Cloud provider
  - Cloud user
- How can we classify current cloud offerings?
- Forward-looking
  - 10 obstacles/opportunities to Cloud Computing
  - What new types of applications does it enable?
  - How should software/hardware change as a result of the Cloud?

Classification of Cloud Offerings

- Applications need:
  - Model of computing, storage, and communication
- Differentiators
  - Level of abstraction presented to developer
  - Level of management of resources

Flexibility
Minimal abstraction
Minimal management of resources

Developer Convenience
Highly abstracted
Highly managed

Amazon EC2

amazon web services™

Sign in to the AWS Management Console | Create an AWS Account | English

Search: Entire Site

AWS | Products | Developers | Community | Support | Account

1 2 3 4 5

Sign up for a free Amazon Web Services Account

Sign Up Now

News & Events

What's New? | Media Coverage | Upcoming Events

Sep 08, 2011 AWS Direct Connect announces general availability in the US West Region

Sep 08, 2011 Announcing the AWS Toolkit for Visual Studio

Sep 07, 2011 AWS Mobile SDKs Exit Beta With Support for Credential Management

Aug 24, 2011 VM Import announces support for Windows Server 2003 and new virtualization platforms

Aug 22, 2011 Announcing Amazon ElastiCache

Aug 18, 2011 Amazon Elastic MapReduce Now Supports Spot Instances

RSS

› View all

Products & Services

Get Started

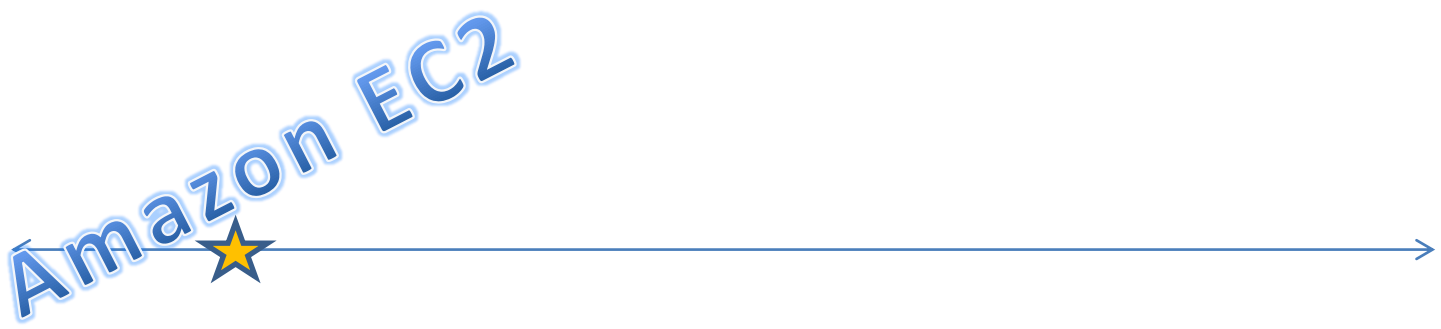
Business Managers

Learn how Amazon Web Services enables you to reach business goals faster:

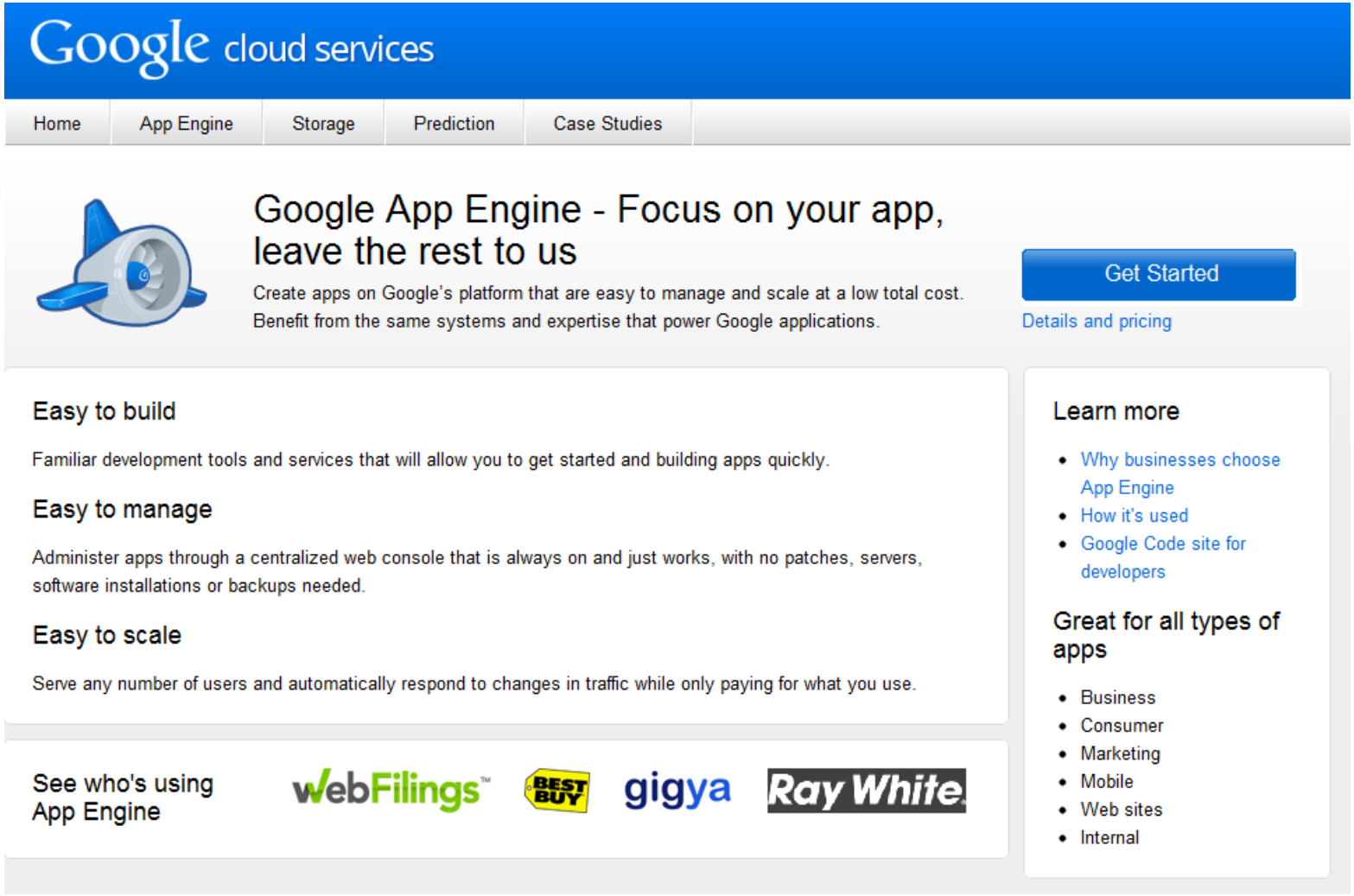
- Solutions & Use Cases
- Security & Compliance
- Economics Center
- Case Studies
- Service Health Dashboard
- Solution Providers
- Videos & Webinars
- AWS Blog

Amazon EC2

- Users control entire software stack (kernel up)
  - Developer can do anything
  - Very difficult to offer automatic scalability and failover (replication/state management highly application dependent)

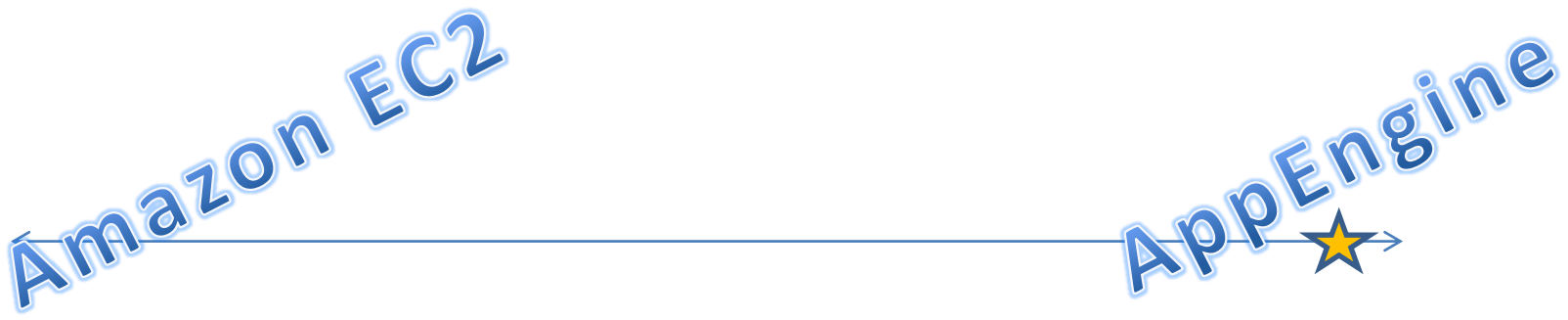


Google AppEngine



Google AppEngine

- Targetted exclusively at **traditional web applications**
- Enforces application structure:
  - Stateless computation tier
  - Stateful storage tier
  - Must be request/reply based
- Excellent automatic scaling and high-availability (but not suitable for general purpose computing)



Microsoft's Azure

Microsoft's Azure

- Must write applications with .NET libraries, and run them on CLR.
- Intermediate failover/scalability

Outline

- What is Cloud Computing? (basic terms)
- Why does it make sense economically?
  - Cloud provider
  - Cloud user
- How can we classify current cloud offerings?
- Forward-looking
  - 10 obstacles/opportunities to Cloud Computing
  - What new types of applications does it enable?
  - How should software/hardware change as a result of the Cloud?

Top 10 Obstacles/Opportunities?

- 1-3: technical obstacles to the **adoption** of Cloud Computing
- 4-8: technical obstacles to the **growth** of Cloud Computing
- 9-10: **Policy** and **business** obstacles to the adoption of Cloud Computing

This list is **RANKED**!

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

Obstacle 1: Availability

- Obstacle:
  - Availability: What if service goes down, either temporarily or permanently?
    - Even if a company has datacenters all over the world, it still probably has common software infrastructure and accounting systems, and can even go out of business.
  - DDoS?
- Opportunity:
  - “No single point of failure”: Spread service over multiple offerings.
  - Automatic scaling of Cloud, and the Cloud Provider’s expertise in defending against these attacks, provides defense. Transfer of target from SaaS provider to Cloud Provider.

Obstacle 2: Data Lock-In

- Obstacle: Data Lock-In
  - No easy way to extract data from the cloud. Thus, customers get locked into a Cloud Provider. This puts customer data at risk.
  - APIs for Cloud Computing have as of yet not been standardized, thus limiting potential of interoperability.
- Opportunity: Standardize APIs
  - With standardized APIs, a SaaS provider could spread their service and data across multiple Cloud providers.
  - Con: fear of “race-to-the-bottom” of cloud pricing
  - Counter: a) Quality as differentiator, b) enables “Surge Computing” (mix between private and public)

Obstacle 3: Data Confidentiality and Auditability

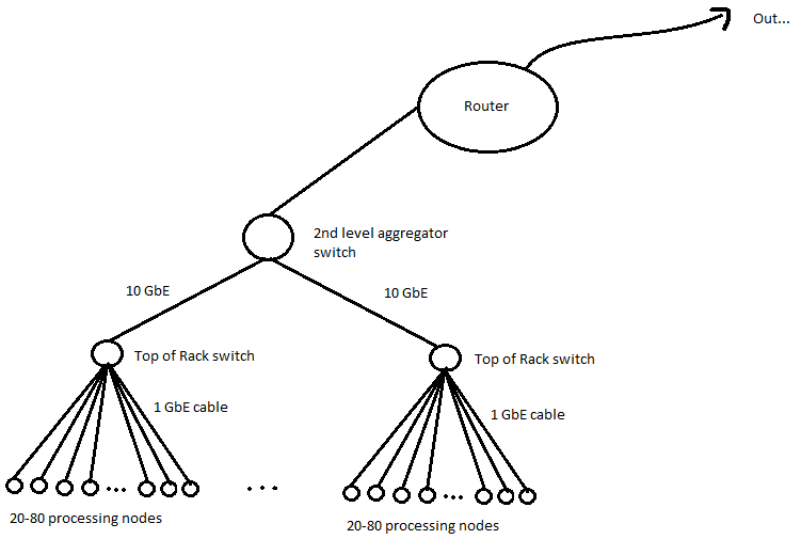
- Authors believe there are no fundamental obstacles in the way of making the Cloud as secure as the vast majority of private datacenters.
- For example, encrypting data before placing it in the cloud may be more secure than unencrypted data in a local datacenter.
- Auditability and confidentiality could be centralized in a single layer beneath the virtualized "guest OS", and in this manner provided as a service.
- Related issue:
  - National Laws (data must be within borders, Patriot Act…)
  - Cloud can accommodate these issues via Geographical Data Storage.

Obstacle 4: Data Transfer Bottlenecks

- Data transfer is a bottleneck in various senses
  - Cost: Expensive to move data around. (eg, $100-$150 per TB)
  - Time: Takes long time to transfer large quantities of data (eg, 45 days to transfer 10 TB at 20 Mbits/sec)
  - Change over time: Technology hasn't been advancing as quickly as others, in terms of performance and cost (< 3x vs 10x)

Obstacle 4: Data Transfer Bottlenecks

- Opportunities
  - Shipping disks (ex: $400 and < 24 hours vs. $1000 and 45 days)
  - Make cloud storage attractive
    - Cloud providers could offer affordable off-premise backup services. (ship physical disks weekly, daily compressed update)
    - Once data is in cloud, new services can be offered/performed, which use CPU cycles.
  - Reduce cost of WAN bandwidth more quickly
    - Reduce cost of high-end routers (only 1/3 cost is fiber optics)
  - Improve Intra-cloud networking
    - The networking cables which aggregate the nodes in a rack are 1 Gbit. Potential bottleneck (ex, aggregation step between Map and Reduce)
    - Cost of higher throughput cables expected to drop.



Obstacle 5: Performance Unpredictability

- VMs share CPU and RAM well, but struggle sharing IO (IO interference), leading to unpredictable IO rates (st. dev./mean ~ 16%)
- Opportunities
  - Efficiently virtualize interrupts and IO channels
  - Incorporate flash memory into stack to decrease interference
- Related problem
  - High performance computing requires that some computations occur simultaneously across various threads. This type of guarantee can't be offered on VMs. (note: HPC has applications with high financial returns)
    - A common pattern in parallel programming is for all threads to compute something, wait for all the others to complete, and then communicate.
  - Opportunity: "Gang scheduling" for Cloud Computing
    - Provide option for Cloud users to schedule n VMs at once, with some synchronization guarantees.

Obstacle 6: Scalable Storage

- Would like storage model, which like computation model in the cloud, supports:
  - Short-term usage (scaling up and down based on demand)
  - No upfront costs
  - Infinite capacity on demand
- Opportunity: "Create a storage system [which] would not only meet these needs but combine them with the cloud advantages of scaling arbitrarily up and down on-demand, as well as meeting programmer expectations in regard to resource management for scalability, data durability, and high availability."

Obstacle 7: Bugs in Cloud

- Sometime bugs only occur when application is deployed in large-scale production environment.
- Opportunity
  - Invent debugger that relies on distributed VMs
  - The fact that VMs are being used could allow for valuable debugging information to be gathered which would not be possible on a physical machine.

Obstacle 8: Scaling Quickly

- Pay-as-you-go applies easily to storage and network bandwidth, but not always as cleanly to computation (depending on model)
  - For example, AWS charges per hour regardless of whether VM is idle.
  - Google AppEngine, on the other hand, does follow the "Pay-As-You-Go" model. Automatically scales up/down in response to demand, and users are charged by cycle used.
- Opportunity
  - Figuring out a way (in these imperfect "pay-as-you-go" computation models which leave computational resources idle) to scale quickly up and down in response to load to save money and energy.
  - Fast and easy-to-use snapshot/restart tool.

Obstacle 9: Reputation Fate Sharing

- "Reputations do not virtualize well"
- One customer's bad behavior in the Cloud can affect the reputation of the cloud as a whole:
  - For example, EC2 IP addresses getting black-listed by a spam-prevention service
  - Future applications running on those physical machines, with those same IP addresses, get affected by the previous bad behavior of an application at this location.
- Opportunity: Create a reputation-guarding service similar to the "trusted email" services currently offered to services hosted on smaller ISPs.
- Another Issue: question of legal liability – How to hold Cloud users liable for any illegal activity they do.

Obstacle 10: Software Licensing

- Current licensing model: restricts # computers on which software runs, users pay upfront, and users pay maintenance fee.
- Note that this model doesn't fit will with cloud computing (50 customers might each only want a certain software for 1 hour, but across 100 machines simultaneously... How would this be billed using the above model?)
- Opportunity
  - Commercial software companies should change their licensing model to better fit the Cloud.
  - Encourage sales teams to sell to Cloud in bulk (eg, 100,000 user-hours)

Outline

- What is Cloud Computing? (basic terms)
- Why does it make sense economically?
  - Cloud provider
  - Cloud user
- How can we classify current cloud offerings?
- Forward-looking
  - 10 obstacles/opportunities to Cloud Computing
  - What new types of applications does it enable?
  - How should software/hardware change as a result of the Cloud?

Looking ahead: New Types of Applications?

- Mobile Interactive Applications
  - High availability + large data sets
- Parallel batch processing
  - Cost associativity + parallelizable tasks = faster results for same $
- Analytics
  - Growing rapidly today (analyzing customer data instead of simply processing transactions)
- Extension to compute-intensive desktop apps
  - Matlab, Mathematica (symbolic math), 3D rendering
  - Good fit: High computation per byte
- Limitations: “Earthbound” Apps
  - Time sensitive financial calculations are bad fit
  - Data transfer costs + time = bad

Outline

- What is Cloud Computing? (basic terms)
- Why does it make sense economically?
  - Cloud provider
  - Cloud user
- How can we classify current cloud offerings?
- Forward-looking
  - 10 obstacles/opportunities to Cloud Computing
  - What new types of applications does it enable?
  - How should software/hardware change as a result of the Cloud?

Future Software/Hardware?

- General recommendation: Always favor horizontal scalability to many VMs over the efficiency of the system on a single VM.
- Applications Software
  - Will have client and cloud components
  - Cloud: should scale quickly
  - Client: should be useful when offline (not currently the case)
  - Needs pay-for-use model
- Infrastructure Software
  - Aware that it's running on VM
  - Billing built-in
- Hardware
  - Designed/built at level of container
  - Energy proportionality: power-down idle resources
  - Processors should efficiently run VMs
  - Flash memory in memory hierarchy
  - Improved LAN switches and WAN routers

Discussion

- General opinions about article:
  - Does it effectively explain Cloud Computing, and answer the most important questions about it?
  - Strengths of paper?
  - Weaknesses of paper?
- Obstacles
  - Is availability really the most important obstacle, and is spreading your service across multiple offerings the answer? If not, what is?
  - Should/will Cloud APIs be standardized, to allow cloud users to easily switch between providers? Would it lead to a "race-to-the-bottom" and kill cloud profits? Is it even technically possible given the different application models of the various providers?
  - Best solution for data transfer bottlenecks?
  - Most important obstacle? Most challenging?
  - Other obstacles not mentioned? Other barriers to adoption? Opportunities not mentioned?
  - Who is best positioned to overcome these obstacles? Start-ups? Researchers? Or Cloud Providers themselves? If a certain group is not best positioned, what can they contribute to these problems?
- Which Cloud model seems best? Will one win?
- Most exciting types of applications which Cloud enables?
- 2009 vs. today? How has Cloud Computing developed?