Summary of Cloud Computing (CC)

from the paper

Abovce the Clouds: A Berkeley View of Cloud Computing (Feb. 2009)

Definitions (I)

- **Cloud Computing refers to both the applications delivered as services over the Internet and the scalable hardware and systems software that provide those services.** The services themselves have long been referred to as *Software as a Service (SaaS).*

Definitions (II)

- The **scalable hardware and software** is what we will call a ***Cloud***.
- When a Cloud is made **available** to the general public, we call it a ***Public Cloud***.
  This often means that one can access a cloud for free or in a pay-as-you-go manner.
- The **service being sold is *Utility Computing***.

Definitions (III)

- We use the term ***Private Cloud*** to refer to scalable hardware and software of a business or other organization, **not made available to the general public**.

Characteristics of clouds

- **illusion of infinite computing resources** (because of scalability)
- ***elimination of an up-front commitment*** *by Cloud users,* thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs
- *The **ability to pay** for use of computing resources **on a short-term basis** as needed* (e.g., processors by the hour and storage by the day)

Why CC now? (I)

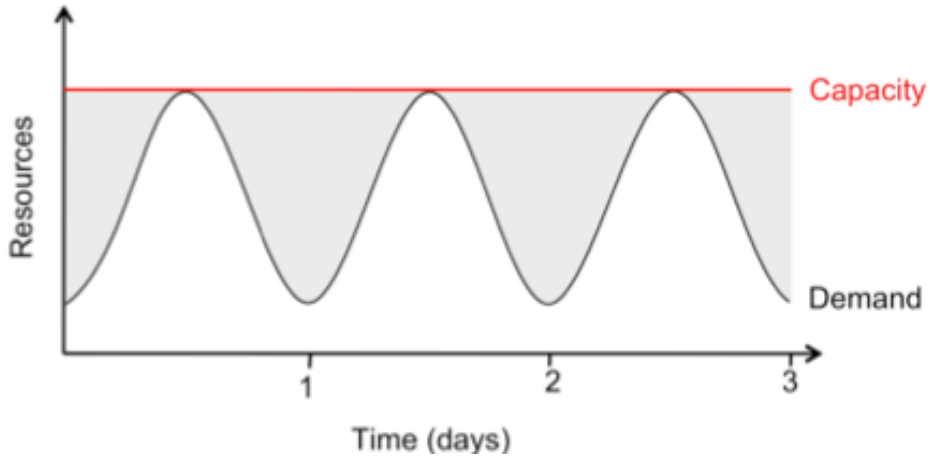
- economies of scale (prices of 2006):

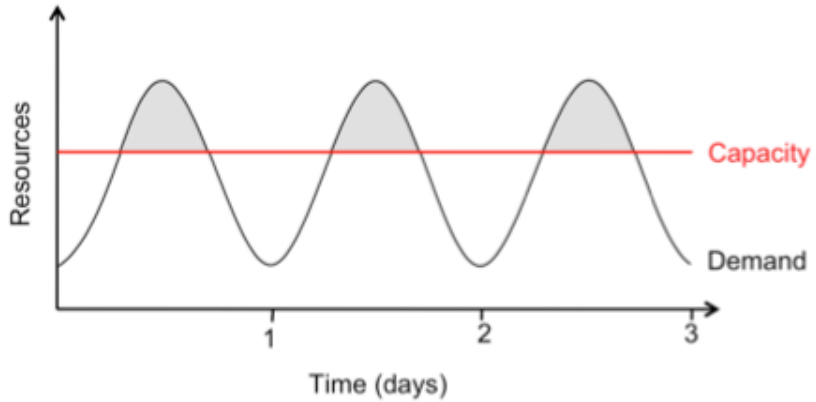
Technology	Cost in Medium-sized DC	Cost in Very Large DC	Ratio
Network	$95 per Mbit/sec/month	$13 per Mbit/sec/month	7.1
Storage	$2.20 per GByte / month	$0.40 per GByte / month	5.7
Administration	≈140 Servers / Administrator	>1000 Servers / Administrator	7.1

- infrastructure costs (eg, electricity)

Why CC now? (II)

- load balancing difficult in private data centers:





(b) Underprovisioning 1

Why CC now?

- large Internet companies (amazon, Google, etc.) had to run huge data centers anyway as their core business with excess capacities **=> additional revenue stream that also reduces excess capacities**
- by having to run huge data centers they **significantly improved the technology to manage them**
- **improved internet infrastructure**

Spectrum of CC

computational model (VM):

Amazon Web Services	Microsoft Azure	Google AppEngine
• x86 Instruction Set Architecture (ISA) via Xen VM • Computation elasticity allows scalability, but developer must build the machinery, or third party VAR such as RightScale must provide it	• Microsoft Common Lan- guage Runtime (CLR) VM; common intermediate form executed in managed envi- ronment • Machines are provi- sioned based on declarative descriptions (e.g. which "roles" can be replicated); automatic load balancing	• Predefined application structure and framework; programmer-provided "han- dlers" written in Python, all persistent state stored in MegaStore (outside Python code) • Automatic scaling up and down of computation and storage; network and server failover; all consistent with 3-tier Web app structure

Spectrum of CC

storage model:

Amazon Web Services	Microsoft Azure	Google AppEngine
• Range of models from block store (EBS) to augmented key/blob store (SimpleDB) • Automatic scaling varies from no scaling or sharing (EBS) to fully automatic (SimpleDB, S3), depending on which model used • Consistency guarantees vary widely depending on which model used • APIs vary from standardized (EBS) to proprietary	• SQL Data Services (restricted view of SQL Server) • Azure storage service	•MegaStore/BigTable

Spectrum of CC

networking model:

Amazon Web Services	Microsoft Azure	Google AppEngine
• Declarative specification of IP-level topology; internal placement details concealed • Security Groups enable restricting which nodes may communicate • Availability zones provide abstraction of independent network failure • Elastic IP addresses provide persistently routable network name	• Automatic based on programmer's declarative descriptions of app components (roles)	• Fixed topology to accommodate 3-tier Web app structure • Scaling up and down is automatic and programmer-invisible

Top 10 Obstacles and Opportunities for CC

	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

1) availability

- down-time of dominant cloud providers is extremely low
- but: cloud is a single point of failure; using multiple clouds (redundancy) is not an option today (incompatibility)
- Distributed Denial of Service attacks: target shifted to cloud provider

2) data lock-in

- clouds are currently proprietary
- problem if more specific clouds are used

- speculation: standard APIs not to be expected in the near future (5 years)

3) data confidentiality and auditability

- technology perspective: encrypting data before placing it in a cloud may be even more secure than unencrypted data in a local data center
- political perspective: laws that require customer/citizen data to be kept within national/EU boundaries

4) data **transfer bottlenecks**

- **10 TB:**
  $10 * 10^{12}$ Bytes / (20×10^6 bits/second) = (8×10^{13})/(2×10^7) seconds = 4,000,000 seconds, which is more than **45 days**
- remedy: express mailing of disks

5) performance unpredictibility

- main memory: no problem
- I/O: 16% variability
  remedy: flash memory

- high performance computing: requires synchronized scheduling of tasks, which today's VMs and operating systems do not provide
- remedy: "gang scheduling" for CC

6) scalable storage

- problem: varying richness of query and storage API, of performance guarantees and the complexity of data structures
- research opportunity: create CC storage system that overcomes these limitations

7) debugging in massively distributed systems

- problem: bugs often cannot be reproduced in smaller configurations
- research opportunity: create appropriate VMs/debugging concepts and tools

9) reputation fate sharing

- one cloud user's bad behavior can affect the reputation of the cloud as a whole
- remedies: reputation-guarding services !?

- question of transfer of legal liability: for example, is the conpany that sends spam or the cloud provider liable

10) software licensing

- CC incompatible with one-time purchases
- remedy: pay-as-you-go licensing models