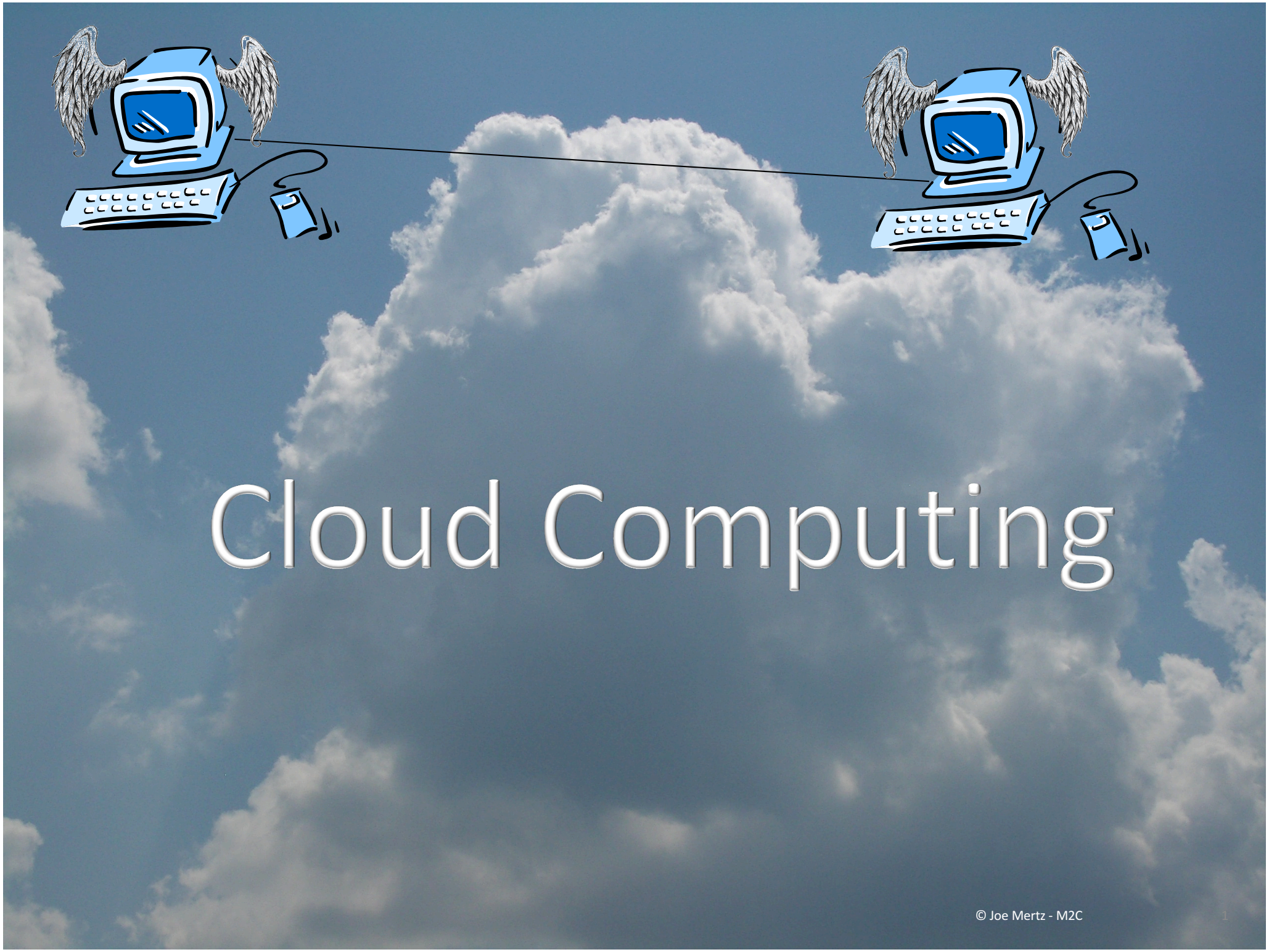


Learning Goals

- To have a general understanding of what falls under the nebulous definition of Cloud Computing
  - Not to be confused with nebulous clouds nor the cloud nebula.
- To be familiar with the concepts of SaaS, IasS, PaaS, (and HuaaS).
- To understand the benefits and risks of using cloud computing
- To be aware of the breadth of services that are available as XaaS
  - Including business models and software development support

What is Cloud Computing?

- Cloud computing is a model for enabling
  - convenient, on-demand network access
  - to a shared pool of configurable computing resources
    - (e.g., networks, servers, storage, applications, and services)
  - that can be rapidly provisioned and released
    - with minimal management effort
    - or service provider interaction

Source: US National Institute of Standards and Technology, 2009

Example Use Cases

- Payroll company – monthly variation
  - Has peak loads on its web applications on the last working day of the month
  - Traffic tails off the rest of the month
- Weather company – special events
  - Fairly steady state load most of the time
  - Extreme peak loads when there is a weather event (hurricane, ice storm, etc.)

Example Use Cases

- Short-Term Campaign
  - You have a campaign (e.g. Superbowl commercial) that needs a short-term increased capacity to manage.

Anonymous Pharmaceutical Company

- Problem:
  - Simulate the interaction of each of millions of compounds with a cancer-related protein.
  - Estimated 341,700 hours of computing
    - I.e. 39 years!
- Solution:
  - Used 10,600 cloud-based compute instances
- Physical equivalent:
  - 12,000 sq feet data center
  - Would cost $44 million
- Result
  - 2 hour setup
  - 9 hours use
  - Peak cost $549.72/hour
  - Total cost $4,362!

Source: http://blog.cyclecomputing.com/2013/02/built-to-scale-10600-instance-cyclecloud-cluster-39-core-years-of-science-4362.html

Example Use Cases

- Startup company - scalability
  - Minimal capital available for equipment
  - Minimal capital available to hire tech support staff
  - No way to predict when their new service will go viral.

E.g. Animoto (cloud-based video creation service)

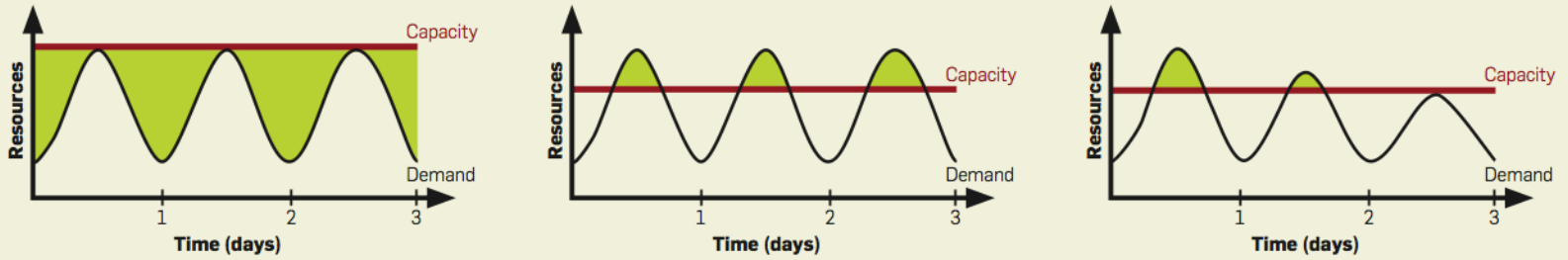
- Animoto made its service available via Facebook
- Resource needs doubled every 12 hours for three days.
- Demand surged from 50 servers to 3,500 servers
- After the peak subsided, traffic fell to a lower level.

Source: Michael Armbrust et al.

» http://doi.acm.org/10.1145/1721654.1721672

Over / Under Provisioning

Figure 2. (a) Even if peak load can be correctly anticipated, without elasticity we waste resources (shaded area) during nonpeak times. (b) Underprovisioning case 1: potential revenue from users not served (shaded area) is sacrificed. (c) Underprovisioning case 2: some users desert the site permanently after experiencing poor service; this attrition and possible negative press result in a permanent loss of a portion of the revenue stream.



– Source: Michael Armbrust et al.
  » http://doi.acm.org/10.1145/1721654.1721672

Infrastructure is Not Your Core Business

- You have a need for an extensible software application that scales indefinitely (from your perspective) and is available 24/7 worldwide.
- And your core business is not distributed software development.

Essential Characteristics

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service
  - US National Institute of Standards and Technology, 2009

On-demand self-service

- Consumer can unilaterally:
  - Provision computing capabilities,
    - E.g. server time and network storage,
  - As needed
  - Automatically
  - Without requiring human interaction

Broad network access

- Capabilities are available
  - Over the network
  - Accessed through standard, published APIs

Resource pooling

- Provider's computing resources are pooled
  - Serving multiple consumers using a multi-tenant model
  - With different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
- Customer generally has no control or knowledge over the exact location of the provided resources
  - But may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).
- Examples of resources include
  - storage
  - processing
  - memory
  - network bandwidth
  - virtual machines.

Resource Pooling - Business model

- Resource pooling is key to the business model
  - Large scale data centers
  - In low-cost geographic locations
    - Real estate
    - Power
    - Labor
  - Statistical multiplexing to increase utilization
  - Resulted in significant decrease in costs
    - Decrease factor of 5 to 7- Armbrust et al
- Therefore, cloud computing could was able provide better software and computing services cheaper than medium and small sized data centers.

Rapid elasticity

- Capabilities can be rapidly and elastically provisioned
  - In some cases automatically
  - To quickly scale out and rapidly released to quickly scale in.
- To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Measured Service

- Providers use a metering capability at some level of abstraction appropriate to the type of service
  - (e.g., storage, processing, bandwidth, and active user accounts).
- Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.
- 1000 processors for 1 hour is no more expensive than 1 processor for 1000 hours

These are worth knowing…

- Essential characteristics of cloud computing
  - On-demand self-service
  - Broad network access
  - Resource pooling
  - Rapid elasticity
  - Measured service
    - US National Institute of Standards and Technology, 2009

Service Models

- IaaS - Infrastructure as a Service
- PaaS - Platform as a Service
- SaaS - Software as a Service
- HuaaS - Humans as a Service

Infrastructure as a Service (IaaS)

- Processing, storage, networks, and other fundamental computing resources
- The consumer can run arbitrary software
  - Including operating systems and applications

Amazon Elastic Compute Cloud (EC2)

- IaaS example
- Multiple instance types, ranging from
- Micro Instance
  - 613 MB memory
  - up to 2 ECUs
  - Network storage
- Extra Large Instance
  - 15 GB memory
  - 8 ECUs
  - 1690 GB local storage
  - Higher network performance

Region: US East (Virginia)

	Linux/UNIX Usage	Windows Usage
Standard On-Demand Instances		
Small (Default)	$0.080 per Hour	$0.115 per Hour
Medium	$0.160 per Hour	$0.230 per Hour
Large	$0.320 per Hour	$0.460 per Hour
Extra Large	$0.640 per Hour	$0.920 per Hour
Micro On-Demand Instances		
Micro	$0.020 per Hour	$0.030 per Hour
Hi-Memory On-Demand Instances		
Extra Large	$0.450 per Hour	$0.570 per Hour
Double Extra Large	$0.900 per Hour	$1.140 per Hour
Quadruple Extra Large	$1.800 per Hour	$2.280 per Hour
Hi-CPU On-Demand Instances		
Medium	$0.165 per Hour	$0.285 per Hour
Extra Large	$0.660 per Hour	$1.140 per Hour

Simple Storage Service (S3)

- Web-based storage
- Web Services interface
  - SOAP and REST

    (Two varieties of machine-to-machine communication)

Platform as a Service (PaaS)

- Programming languages, tools, and/or software systems provide a platform upon which a customer can build an applications.
- The consumer does not manage or control the underlying computing infrastructure, but has control over the deployed applications.

Force.com - PaaS

- Force.com development platform
- Apex programming language
- API
- Eclipse IDE integration
- Database, security, workflow, and user interface tools
- Free for developers

Google App Engine – PaaS

- Run web apps on Google infrastructure
  - Automatic scaling and load balancing
- Security and Authentication
- Work queues for scheduled tasks
- Messaging

Google App Engine

- Provides Java, Python, Go, and PHP platforms
- Eclipse IDE integration
- Persistent storage via:
  - Cloud Datastore – NOSQL
  - Cloud SQL – based on MySQL
  - Cloud Storage – objects & files up to 1TB

Software as a Service (SaaS)

- Provides the capability to use software applications running on cloud infrastructure.
- Applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email).

*aaS Customer Base

- While with IaaS and PaaS, the consumer is an application developer, with SaaS, the consumer can be an end user (or application developer).
- E.g.
  - Companies can use the SaaS named CakeHR
    - Human resources software
  - CakeHR uses the PaaS from CenturyLink/AppFog
    - PHP platform as a service
  - AppFog uses the IaaS from Amazon Web Services
    - EC2, S3

Example SaaS

- SalesForce.com – CRM
- Flickr – Photo Management
- YouTube, Vimeo – Video Streaming
- Piazza – Course forums
- Others you use?

Human as a Service (Huaas)

- Less common association with cloud computing
- E.g.
  - YouTube crowdsourcing of “newsworthy” videos
  - Amazon product reviews
  - Amazon Mechanical Turk
  - ReCAPTCHA
  - duoLingo
  - Uber
  - Burpy
  - Thumbtack
  - Favor Delivery
  - HomeAdvisor

Public & Private Clouds

- Public cloud:
  - Cloud computing provided to public customers
  - Service aka *utility computing*
- Private cloud:
  - Cloud computing only within a firm
  - Only sensible when economies of scale are big enough to justify
    - Else you just have a "data center".

Where can we run Node.js apps?

- Laptop
  - No static IP address
  - Laptop not awake when TA wants to test
- Andrew.cmu.edu
  - Only static web pages
- Traditional web hosting (e.g. DreamHost, BlueHost)
  - Do not allow continuous processes
  - Node.js is a running process, not something a web server invokes
- IaaS – E.g. AWS
  - Possible, but need to create OS stack to run Node on.
  - More work than is needed
- PaaS
  - Best alternative for what we need.

Node.js PaaS Options

- Zeit Now
  - Free
  - Amazingly easy CLI to deploy
- Heroku
- Redhat OpenShift
- Microsoft Azure
- IBM Bluemix
- Modulus
- Nodejitsu
- AppFog
- EngineYard

And others…