

# Lecture #17: Cloud Computing and The Internet of Things

*CS106E, Young*

---

*In this lecture, we study Cloud Computing. We consider what it means for something to “be on the cloud” and look at Utility Computing and Grid Computing – two early paradigms for the power that Cloud Computing can bring. We then study four different types of Cloud Computing in use today: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Serverless, and Software as a Service (SaaS).*

*Next, we look at a related but distinct type of computing – variously referred to using the interrelated terms Edge Computing, Fog Computing, or Mesh Computing. As we discover, this type of computing has some real advantages in certain distinct cases, including remote locations (such as offshore platforms) and for the military. In addition, this type of computing has important uses for the Internet of Things.*

*We conclude the lecture with a discussion of the Internet of Things (IoT). We look at what it is and what are some of its uses. We take a look at some issues that may occur with IoT, including heightened security and privacy issues.*

---

## **Cloud Computing Basics**

- The basic concept of cloud computing is that instead of limiting ourselves to what’s on our own personal computer, we instead access computing resources available via the network.
- Under this basic definition, anything we access out on the network could be considered cloud computing. This could include:
  - o file servers
  - o mail servers
  - o media servers
- So the following phrases would be accurate:
  - o When I take a photo with my iPhone, it gets automatically stored on the Cloud where I can access it from my computer or other devices.
  - o I backup critical documents to the Cloud by storing them on DropBox.

## **Providing Virtual Computing Devices for Business**

- Often however, when we hear the term “Cloud Computing,” particularly in a business context, what’s being discussed is replacing direct physical installation, storage, and maintenance of computers on premise with remote resources partly managed by a third party.
  - o There are lots of different models for how this might work, and we’ll explore some of the most common.
  - o Keep in mind when we look at these models, the comparison point is the traditional (non-Cloud) approach where:
    - The company manages and stores its own computers.
    - It is responsible for making sure that the computers are secure, that they have sufficient electricity, that they stay connected to the network.

- It is responsible for buying more computers as needed, and installing those computers.
  - It is responsible for maintaining the computers, making sure that they are patched.
- Early on the paradigms of **Utility Computing** or **Grid Computing**<sup>1</sup> provided a conceptual view of what cloud computing might look like.
  - These terms are closely tied together.
    - The term Utility Computing comes from the concept of a public utility such as a water or electric utility.
    - Grid Computing comes from the concept of the electric grid.
  - In both cases, the idea is that there should be computing capability available to users that users can tap into as simply as we can access electricity from the electric grid.
  - Under these models
    - Computing is available on demand. When we need more computing power, it's just there for us, just as if I buy a new television or a new air conditioner, I simply plug it into the wall and the electricity is there as needed.
    - It's very easy to scale up or scale down.
      - If it's no longer winter, I stop using my electric heaters. I don't have to contact anyone to tell them to reduce the electricity provided to my house. If it gets warm, I can just turn on an air conditioner pulling more electricity without calling the electric company.
    - I don't need to determine my needs in advance.
      - I don't have a contract that sets out how much electricity I'm planning to use for a given month.
      - I draw as much as I need and pay for just as much as I use.
  - Computing under the Grid Computing and Utility Computing models provides tremendous benefits to companies.
    - They provide **tremendous flexibility and agility** to a company.
    - This is in very sharp contrast to the traditional model where I would have to estimate how many computers I needed in advance.
      - If I overestimated, I would be wasting resources.
      - If my needs changed, I might be wasting resources.
      - If I underestimated, I would need to order more and would have to wait until they arrived and were installed before I could meet customer demand.
  - Another potential benefit provided by these models is that the computers aren't located at my company.
    - My provider is likely to provide a more secure location than a small company can provide.
    - Similarly, the computers are likely to have more reliable access to electricity (along with backup generators) and more reliable Internet access than many individual companies would have for their own computers.
    - I can have redundant computers at multiple locations, ensuring that my own customers have access even if one of the computers fails or a natural disaster such as a hurricane occurs.

---

<sup>1</sup> These terms are more or less used interchangeably. From what I can tell, Utility Computing tends to focus more on the cost structure, whereas Grid Computing tends to focus more on the idea that computers are physically out somewhere on a grid that we can remotely access.

- Moreover, I can have computers around the world, giving global customers low latency access to my own services.
- While not all Cloud Computing provides the benefits envisioned by Utility and Grid Computing, Cloud Computing services provide at least some of the benefits foreseen by these models.
  - These terms are still used and offer a useful way of thinking about the benefits Cloud Computing can give.

### Cloud Computing Models

- Let's take a look at some common cloud computing models in use today.

#### Infrastructure as a Service (IaaS)

- With IaaS, the company is providing basic computing infrastructure on the cloud. It's up to the customer to organize and arrange that infrastructure as needed.
- Amazon Web Services (AWS) Virtual Private Cloud (VPC) is an example of IaaS.
- My company can choose to create an Amazon Virtual Private Cloud.
  - I contract for a number of Amazon Elastic Computing Cloud (EC2) instances.<sup>2</sup>
  - My company's IT professionals setup the EC2 computers with operating systems as desired.
  - They can arrange their network structure and can setup internal subnetworks controlling which EC2 instances have access to the external world and which are private internal servers.
- The Infrastructure as a Service (IaaS) model is the closest of the Cloud Computing models to my company just having its own servers.
  - In both cases, we need an extensive IT staff that is very knowledgeable and is able to configure our computers and keep them well maintained and updated.
  - However, in IaaS, while I don't get the easy "simply plug things in" capability envisioned by Utility and Grid Computing, I do get many of the other benefits.
    - I can easily add or remove computer resources as needed. I do not need to await for the arrival of actual physical hardware, nor am I responsible for physically installing the hardware.
    - My hardware is stored in a well-protected location with good power backup and good Internet access.
    - I can choose to have redundant servers in different locations and I can choose to locate my hardware anywhere around the world.

#### Platform as a Service (PaaS)

- With this model, a computing platform is provided for me. I am not responsible for configuring it or maintaining it. I'm only responsible for the application software running on it.
- Heroku and Amazon Elastic Beanstalk are both examples of PaaS.
- With Heroku
  - I write my application in any of the supported languages.
    - This includes JavaScript Node.js, Ruby, PHP, Python, and Java.
  - I store my code using Git.

---

<sup>2</sup> In cloud computing a computer instance may represent an actual physical computer, or it may represent a part of a computer where my work is placed in a sandbox separate from other users who may also be placed on the same computer. These are sometimes referred to as Virtual Private Servers (VPS).

- Git is a popular version control system, which is used by software developers to store different versions of their source code.
- I instruct Git to send my code to Heroku.
- That's pretty much it, my code is now running live on the Internet.
- I do have the option to add extra computer instances and the option to choose some add on packages (such as choosing to include a database).
- However, in contrast to Infrastructure as a Service (IaaS), I'm not setting up my own network and configuring my own computers.
  - I can't even access many aspects of the Operating System on the Heroku computer instance I'm using.
  - I just put up my application on the instance, and Heroku takes care of the rest.

### Serverless

- In the serverless model, I don't set up any computing instances. Not only am I not responsible for maintaining and keeping computer instances updated, this model isn't concerned with computer instances at all.
- Amazon AWS Lambda is an example of the serverless model.
- With AWS Lambda
  - I write a function that I want to be able to call remotely.
  - I upload the function code up to Amazon.
  - I can call the function remotely as needed.
  - I pay for as much computing resources as I use.

### Software as a Service (SaaS)

- SaaS is a bit different from the other Cloud Computing models I've listed here. In those models, my company is creating some sort of unique computing application.
  - In SaaS another company has created an application that my company is simply using.
  - However, I've included it here as it is another model of cloud computing and you will often see SaaS, discussed at the same time as IaaS and PaaS.
- In SaaS, a company has moved user applications to the web.
- Google Docs and DropBox are both examples of SaaS.
- SaaS provides a number of potential advantages for both the company providing it and the company using it.
  - For customers, SaaS means that their IT department does not have to worry about maintaining and patching computers to ensure that the latest version of the software is available.
    - If you're editing a Google Spreadsheet, then you're using the latest version of the software. The software is actually on Google's server (or downloaded via JavaScript to the web browser when you open a Google Spreadsheet). The application is not installed on your own computer, so there's no way for your "version" to end up out of date or missing a security patch.
  - SaaS software often takes advantage of its "on the network" nature to provide core collaborative features that might be simply an afterthought on traditional desktop applications.
  - As SaaS usually requires a subscription, the providing company has a steady income stream.

- SaaS also acts as its own form of copy protection. If you don't have a paid subscription, you can't access the software, and it can't be pirated.

### **Edge Computing / Fog Computing / Mesh Computing**

- Let's take a look at another model of computing. This model falls under the categories of Edge Computing, Fog Computing, or Mesh Computing.
  - Cloud Computing refers to accessing computer resources out somewhere on the wider Internet.
  - In this alternative model, we access computers on the "Edge", where Edge is defined as somewhere close to our originating data sources.
  - This type of computing often involves peer-to-peer computing, although it may also involve servers which are on the edge.
  - The term Fog computing uses the word "Fog" to provide a contrast to "Cloud", evoking the idea that Fog computing is somewhat similar to Cloud Computing, but also has differences.
  - A Mesh Network is a more decentralized network than traditionally networks. It may be formed in an ad hoc manner. It may be more robust than a traditional network.
- What are the benefits of computing on the edge rather than on the cloud?
  - Edge Computing works well in locations that have poor or intermittent Internet access.
    - General Electric GE Digital, for example, notes that Edge Computing can be particularly beneficial in situations such as offshore platforms or remote mines.
  - Because most of the data is being handled close by, rather than not being sent across the wider Internet Edge computing has:
    - reduced bandwidth requirements
    - reduced latency requirements<sup>3</sup>
  - Applications based on this computing model may include:
    - A distributed computing model, meaning that there is not one central point of failure.
    - Self-healing network technology allowing it to respond reasonably if devices drop out.
- In addition to the previously mentioned remote locations, other examples include:
  - Mobile Computing
    - The use of Edge servers located alongside cell towers allows processing to occur near cell phones, reducing the overall network traffic.
      - This is one reason why the European Telecommunications Standards Institute is working on Edge Computing standards.
  - Military Applications
    - The military has a clear interest in a self-healing mesh network that can continue to operate if devices are lost or if access to the wider Internet goes down.
- Edge Computing can be used in conjunction with Cloud Computing, with some of the data handled by relatively local Edge Servers, but additional requests sent on to servers further out in the Cloud.
- Edge Computing often involves devices working in conjunction with more traditional computers or mobile devices. These devices often fit into the category referred to as "The Internet of Things".

---

<sup>3</sup> Remember bandwidth refers to how much and how quickly our network can send data. Latency refers to how much of a delay exists before our data arrives at its destination.

## The Internet of Things (IoT)

- The Internet of Things consists of devices connected to the Internet that aren't what we would generally think of as computers (or even mobile computers).
- Let's take a look at examples in a few different realms.
  - Devices within the Internet of Things found in a "Smart Home" might include:
    - Smart Lightbulbs,
    - Sensors (which might be used to track which rooms have people in them),
    - Thermostats,
    - Smart Speakers (which allow your music to follow you from room to room).
  - Dorms
    - One of the better uses of IoT that I've seen is that the Stanford dormitories have laundry machines and dryers on the Internet. A student needing to do their laundry can check whether or not a washer is available without having to go down to the basement. They can also check the status of their laundry loads.
  - There are quite a few devices in Cars and Transportation Infrastructure that could count as IoT.
    - The car itself contains multiple devices that could use the IoT to communicate, including navigation systems, entertainment systems, heating a/c, or even engine sensors and actuators.
    - Cars can pass information to one another on road conditions.
    - Sensors can be embedded within roads or at traffic lights.
    - Parking meters can be integrated into the Internet, letting someone see where parking spaces are available or how much time is left on their meter.
    - Sensors can be used to allow engineers to keep track of the health of infrastructure such as bridges.
- Industrial use of the Internet of Things is often distinguished as forming its own category. This is sometimes referred to as **IloT (Industrial Internet of Things)**.
  - Usage scenarios for IloT include:
    - Instrumenting an industrial process and including sensors and Internet actuated controllers.
    - Inventory management potentially done with RFID tags (more on this below).
    - Fleet Monitoring.
    - Security such as motion sensors, cameras, and lights.
  - Radio Frequency IDs (**RFIDs**) are particularly useful for IloT applications.
    - An RFID is similar to the ubiquitous bar codes found on all products sold in grocery stores and many other types of stores.
      - An RFID allows us to give different items an ID number.
      - However, in contrast to a barcode, an RFID can communicate remotely with an RFID reader.
    - Ranges of RFID readers vary, but typically if the reader is within 1 meter it can read the RFID tag. Proposals exist for reader / RFID combinations that have ranges as far as 200 meters.
      - Most RFIDs do not need its own power supply, but rather their circuitry will be triggered and powered by the remote RFID reader.
    - A sample scenario for RFID is inventory management.
      - Currently employees need to individually count boxes on the store shelf to provide an accurate inventory count.
      - With RFIDs, an employee can simply walk down each aisle in the store. The RFID reader will query each product on the shelf, and their RFID tags will respond with their ID number.

- Inventory counting immediately goes from a major undertaking, involving many employees, to just a minor chore any employee can do in just a short time.
    - As RFID prices drop, they will become increasingly useful. Currently they are about 15 cents per RFID, which is still too high for many scenarios. For example using it for store inventory might raise the cost of everything in the store by 15 cents.
- Let's take a look at some issues that arise with IoT.
  - IoT devices obviously need to connect to the Internet.
    - While some connect using WiFi, this takes a fair amount of electric power.
    - One common choice is Bluetooth Low Energy.
  - Once we start putting all manner of devices on the Internet, our usage of Internet numbers will go up astronomically.
    - Use of IPv6 is important for widespread adoption of IoT.<sup>4</sup>
  - **Security is a particular concern with IoT**
    - IoT devices may be particularly vulnerable for a variety of reasons.
      - IoT devices are consumer devices that are often cost sensitive, keeping costs down may result in less attention to security than may be warranted.
      - They are often developed by startups with innovative ideas, but perhaps lacking in specialized security skills.
      - They may not be updated when vulnerabilities are found – after all, who thinks about keeping their lightbulbs updated with security patches.
    - Let's take a look at some sample IoT failures.
      - Smart Light Bulbs [have been hacked](#). While one can laugh off the ability for a hacker to control the lights in someone else's house, gaining access to smart light bulbs may be a way into a home network, bypassing any security measures needed to get in through the cable network or telephone network.
      - Many IoT cameras have been found to be [vulnerable to hacking](#).
      - Finally, one IoT company making Smart Locks managed to [accidentally brick some of the locks](#) with a software update. Customers with these locks were not able to lock their doors.
  - **Privacy also presents concerns with IoT**
    - With IoT devices everywhere, we are leaving more and more electronic breadcrumbs on our activities and our locations.
- Just as there are industry packages designed to provide Cloud Computing support, computer companies are providing support for IoT development.
  - The IBM Watson IoT platform, for example, provides:
    - real-time data collection and analysis from IoT devices,
    - machine learning to process the information gathered,
    - security measures to help keep IoT safe.

---

<sup>4</sup> As you may recall from Lecture #8 Internet Protocol Version 4 (IPv4) only gives us about 4.3 billion addresses. That's not enough if every device in the house and out on the road is on the Internet. IPv6 allows for  $3.4 \times 10^{38}$  addresses.