The University for
world-class professionals

Manchester
Metropolitan
University

Advanced Operating Systems
Lecture : Cloud computing

Soufiene Djahel
Office: John Dalton E151

Email: s.djahel@mmu.ac.uk

Telephone: 0161 247 1522

Office hours: Monday 10 -11, Thursday 11-13

This lecture was adapted and extended from the slides originally designed by Emma Norling, MMU, UK

Recap from last week

- Processes
- Threads
- Concurrency
  - Types of process interaction
  - Race conditions & critical sections
  - Mutual exclusion by busy waiting
  - Mutual exclusion & synchronization
- Deadlock

Today's objectives

- To examine the "big picture" of cloud computing
- To explore the implications of cloud computing for operating systems

A brief history

- The idea of computing in a "cloud" traces back to the origins of utility computing
  - If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility. … The computer utility could become the basis of a new and important industry, (John McCarthy, 1961)

A brief history (cont.)

- In 1969, Leonard Kleinrock, a chief scientist of the Advanced Research Projects Agency Network or **ARPANET** project that seeded the Internet, stated:
  - As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of 'computer utilities' ….

A brief history (cont.)

- Late 1990: Salesforce.com pioneered the notion of bringing remotely provisioned services into the enterprise
- 2002: Amazon.com launched the Amazon Web Services (AWS) platform
- 2006: Amazon launched its Elastic Compute Cloud (EC2) services
  - Emergence of the term "Cloud computing"

What is Cloud Computing?

What is Cloud Computing?

- IBM: "Cloud computing, often referred to as simply "the cloud," is the delivery of on-demand computing resources:
  – everything from applications to data centers
  – over the internet on a pay-for-use basis

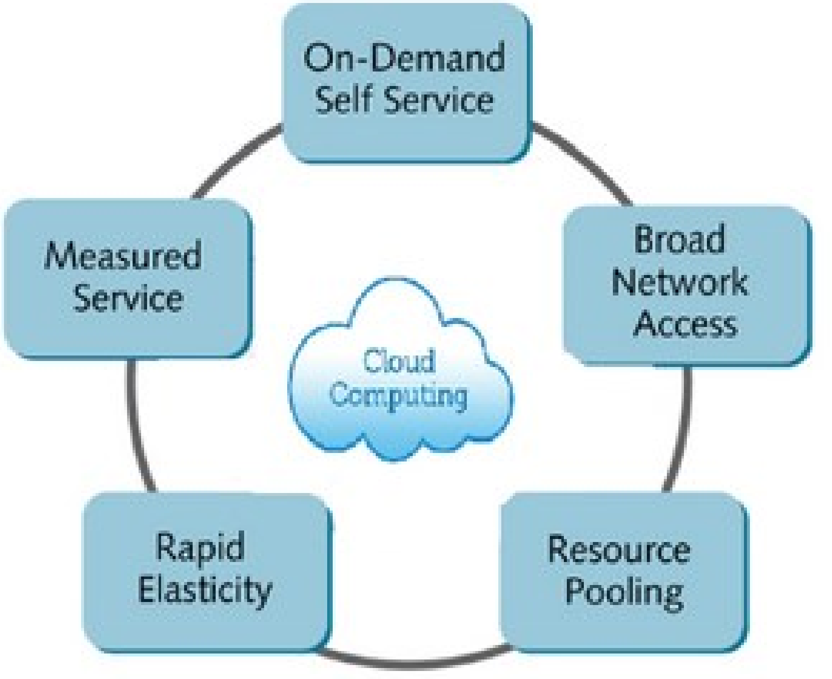
What is Cloud Computing?

- Cloud computing is a model for enabling ubiquitous, 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: NIST(National Institute of Standards and Technology)

Cloud Computing : the Essential Characteristics

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service



On-Demand Self-Service

A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad Network Access

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource Pooling

- The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources:
  - dynamically assigned and reassigned according to consumer demand.

Resource Pooling (cont.)

- Location independence: the 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 data-center).
- Examples of resources include storage, processing, memory, and network bandwidth.

Rapid Elasticity

- Capabilities can be elastically provisioned and released, in some cases automatically, to:
  - scale rapidly outward and inward commensurate with demand.
- To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured Service

- Cloud systems automatically control and optimize resource use by leveraging:
  - 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.

Classical computing vs. Cloud computing

Classical computing	Cloud computing
Repeat the following cycle every 18 months	Pay as you go per each service provided
Buy and own Hardware, system software, applications to meet peak needs	**Subscribe** -----
Install, configure, test, verify, evaluate, manage ------	**Use** (Save about 80-95% of the total cost according to experts at IBM) ------
Use	-------
------ **Pay $$$$$ (High cost)**	**$ - Pay for what you use** Based on the QoS

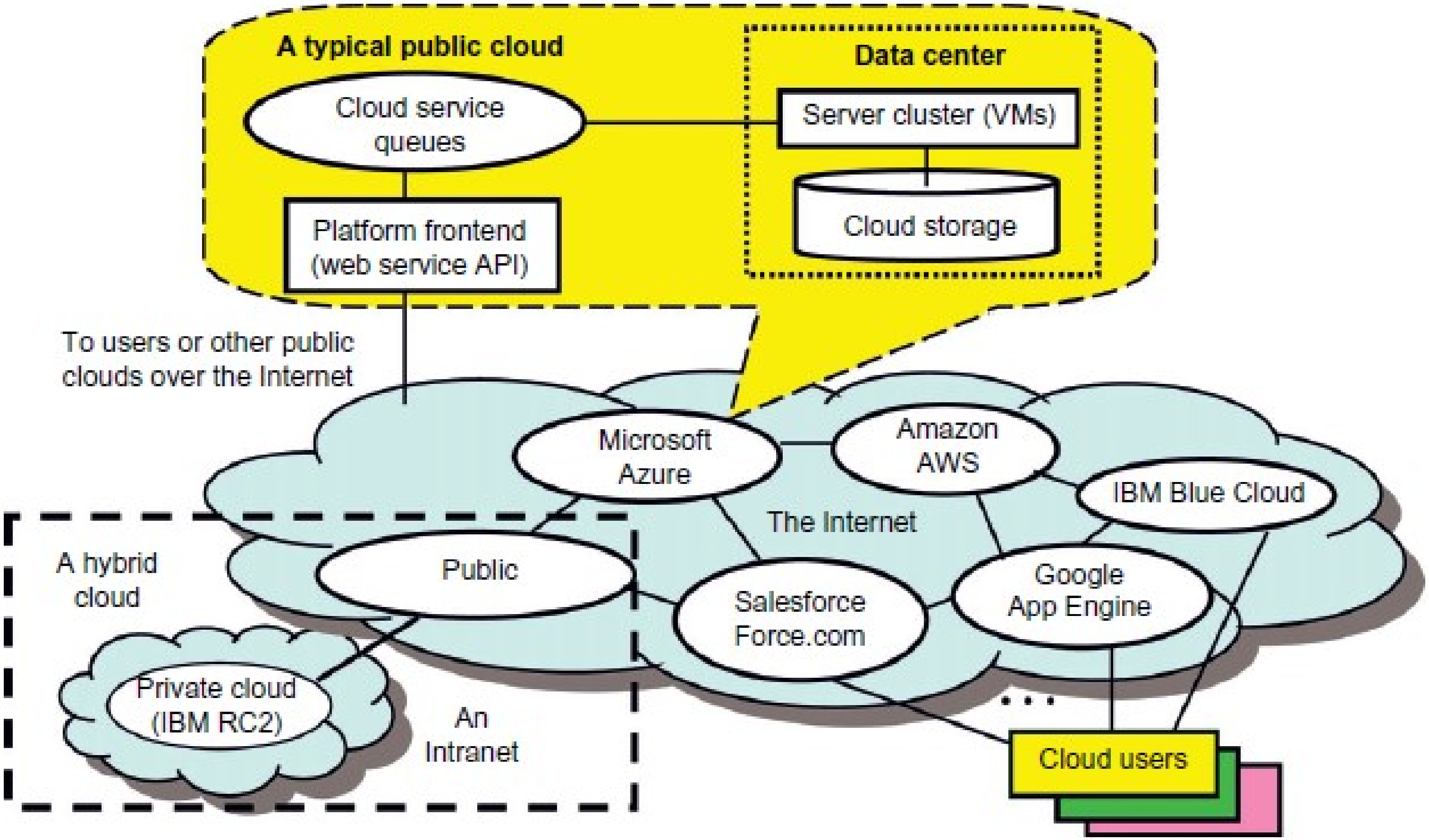
Source: Distributed and Cloud Computing: From Parallel Processing to the Internet of Things

Business drivers for cloud computing

- **Capacity Planning**
  - planning for capacity can be challenging because it requires estimating usage load fluctuations.
- **Cost Reduction**
  - cost of acquiring new infrastructure and its ongoing ownership
- **Agility**
  - Businesses need to adapt and evolve to successfully face change
  - respond to business change by scaling its IT resources

Types of cloud deployment

- Public
- Private
- Hybrid



Public, private, and hybrid clouds illustrated by functional architecture and connectivity of representative clouds available by 2011

Public clouds

- Public clouds are built over the Internet and owned by service providers
- Accessible through a subscription
- Promote standardization, preserve capital investment, and offer application flexibility
  - Examples: Google App Engine (GAE), Amazon Web Services (AWS), Microsoft Azure, …

Private clouds

- A private cloud is built within the domain of an intranet owned by a single organization
- Its access is limited to the owning clients and their partners
- Attempt to achieve customization and offer higher efficiency, resiliency, security, and privacy
  - Example: IBM Research Compute Cloud (RC2)

Hybrid clouds

- Built with both public and private clouds
- Operate in the middle, with many compromises in terms of resource sharing
- Private clouds can also support a hybrid cloud model by
  - supplementing local infrastructure with computing capacity from an external public cloud

Cloud service models

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

Differ in what is provided to the consumer

IaaS

- Delivery of computer infrastructure (typically platform virtualization environment) as a service
- Buy resources such as servers, software, data center space and network equipment as fully outsourced services
  - i.e., the service is performed by rented cloud infrastructure
- e.g., Amazon EC2, GoGrid, FlexiScale, etc,

IaaS: Virtualisation

- Virtualisation technology is a major enabler of IaaS
  - It is a path to share IT resource pools: web servers, storage, data, network, software and databases.
  - Higher utilization rates
  - *More about this in week 5*

PaaS

- Delivery of an integrated computing platform to build/test/deploy custom apps
- Deploy your applications & do not worry about buying & managing the underlying hardware and software layers

PaaS (cont.)

- Customer enters into a Service Level Agreement (SLA) with provider about the scalability of the apps
- Customer provided with API for underlying infrastructure
  - no need to know OS
  - underlying OS as for SaaS
- e.g., Google App Engine (Python, Java,..), MapReduce (Java, Ruby, …), Aneka, etc

SaaS

- e.g. Google Docs, Office 365
  - Customer connects to service via cross-platform interface
- Customer should be able to use apps anywhere, anytime
- Cloud storage of data
  - e.g. GFS (Google File System)
- “The data centre is the computer”
  - In fact, a network of data centres are the computer

Where does the OS come in?

- Operating systems are involved in different ways, depending on the provision
- Underlying everything is an OS
  - <u>SaaS</u>: OS is “hidden” from customers, they just use apps
  - <u>PaaS:</u> OS is provided, customer uses platform
  - <u>IaaS</u>: customer has a range of (virtual) machines. OS support may be provided, or customer may be responsible for OS management

What is its Operating System?

Traditional OS

- Data sharing
  - Inter-Process Communication, RPC, files,
- Programming Abstractions
  - Libraries (libc), system calls, …
- Multiplexing of resources
  - Scheduling, virtual memory, file allocation/protection, …

Cloud OS

- Data sharing
  - Google File System, key/value stores
- Programming Abstractions
  - MapReduce, PIG, Spark
- Multiplexing of resources
  - Apache projects: Mesos, ZooKeeper, BookKeeper, …

Programming Abstractions

- Designed to work with the underlying distributed file systems
  - specific to particular file system, but many similarities
- Distribution is transparent
- Automatic fault-tolerance
- Automatic scaling
- Automatic load distribution

Data Centre Scheduling

- Rapid innovation in data center computing frameworks
- **No single framework optimal for all applications**
- Want to run multiple frameworks in a single data center
  - to maximize utilization
  - to share data between frameworks

Pig

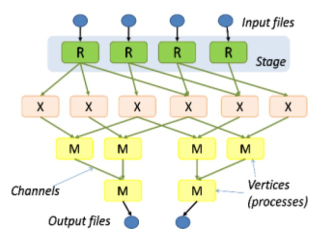
CIEL

RAILS

S4 distributed stream computing platform

Google Pregel

hadoop MapReduce



Dryad

Google Percolator

MPI-2

Example: Apache Mesos

"Program against your data center like it's a single pool of resources"

Apache Mesos abstracts CPU, memory, storage, and other compute resources away from machines (physical or virtual), enabling fault-tolerant and elastic distributed systems to easily be built and run effectively."

Source: https://mesos.apache.org/

What is Mesos?

A distributed systems kernel

Mesos is built using the same principles as the Linux kernel, only at a different level of abstraction.

The Mesos kernel runs on every machine and provides applications (e.g., Hadoop, Spark, Kafka, Elastic Search) with API's for resource management and scheduling across entire data-center and cloud environments.

Source: https://mesos.apache.org/

What is Mesos?

- Project features
  - Scalability to 10,000s of nodes
  - Fault-tolerant
  - Native isolation between tasks with Linux Containers
  - Multi-resource scheduling (memory, CPU, disk, and ports)
  - Java, Python and C++ APIs for developing new parallel applications
  - Web UI for viewing cluster state

Why Cloud Computing?

- Large-Scale Data-Intensive Applications
- Flexibility
- Scalability
- Customized to your current needs:
  - Hardware
  - Software

Why Cloud Computing?

- Effect:
  - Reduce Cost
    - No CAPEX
  - Reduce Maintenance
  - High Utilization
  - High Availability
  - Reduced Carbon Footprint

Why *not* Cloud Computing?

- Security
- Privacy
- Vendor lock-in
- Network-dependent
- Migration

Fog computing

- Key idea: push the computation closer to the edge of the network to minimize latency
- The gateway bears the responsibility for IoT application execution
- Drawbacks:
  - the gateway is single point of failure
  - increased delays in applications involving control

Mist computing

- Key idea: push appropriate computation to the very edge of the network, to the sensor and actuator devices that make up the network
- The Mist computing paradigm decreases latency and further increases the autonomy of a solution.

Cloud, Fog and Mist Computing

- Cloud
  - Server level
- Fog
  - Gateway level
- Mist
  - End device level
    - e.g., an IoT device

Summary

- Cloud computing has a number of potential benefits for business
- But also potentially some issues
- Cloud services can be delivered in a variety of forms
  - Underlying the services is some form of "non-traditional" OS
  - Traditional OSes *may* be used by customers

References

- Thomas Erl et al, "Cloud Computing: Concepts, Technology & Architecture"
- Kai Hwang et al, "Distributed and Cloud Computing: From Parallel Processing to the Internet of Things"

Next lecture: Distributed file systems