



UMD DATA605 - Big Data Systems

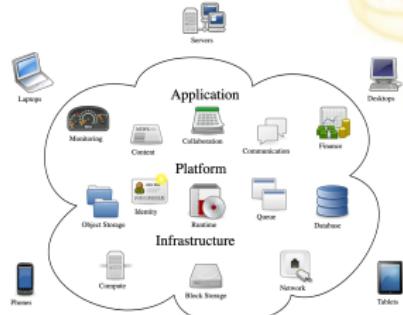
11.1: Cloud Computing

- **Instructor:** Dr. GP Saggese, gsaggese@umd.edu

Cloud Computing

- Computing as "service" instead of "product"

- Storage and computing in the cloud
- Edge devices (e.g., phones, laptops, tablets) interact with the cloud



- Advantages of cloud computing

- Device agnostic: seamless computation across devices
- On demand
- Efficiency / scalability
 - Programming frameworks (e.g., Hadoop, Spark, Dask) enable scalability
- Reliability
- Cost: “pay-as-you-go” for resources
 - Cheaper than building own systems
 - Computing as a commodity (like electricity)

Buying Infrastructure

- **To buy or to rent** computers and infrastructure?
 - Trade-off between time and money
 - Very difficult to decide objectively
- **Buying / building infrastructure**
 - Requires time and capital investments (Capex)
 - Especially at the beginning without revenues
 - Smooth cash flow (constant \$/mo) is better than a lumpy one (big one-time purchase)
 - Buy hardware (e.g., computers, storage, network)
 - Estimate current hardware size
 - Difficult to estimate future demands
 - Update obsolete hardware
 - Cost of owning hardware (Opex)
 - Data center, electricity, cooling, handling faults
 - Administering
 - Install, update, maintain software stack

Renting Infrastructure

- **Renting infrastructure** (i.e., cloud computing)
 - Pay for what you use
 - Low initial capital investment
 - Ready systems with a click
 - No multi-year resource plan needed
 - Choose machines based on your application and data needs

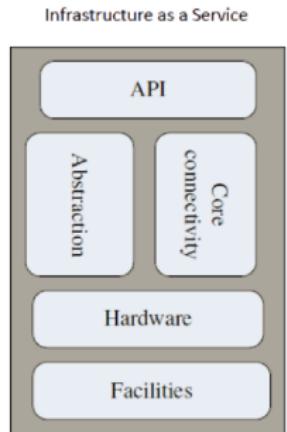
Cloud Computing

- Ideas of cloud computing around for a long time
 - Mainframes + thin clients (1960s)
 - Personal computers (1980s)
 - Grid computing for supercomputers (1990s)
 - Peer-to-peer architecture (early 2000s)
 - Client-server model (Web 1.0 and Web 2.0)
 - Cloud computing (2010s)
- Now, it finally works
- Why now?
 - A convergence of key technologies
 - OS virtualization
 - Large data centers
 - Decreasing hardware costs
 - Big data frameworks



Infrastructure-as-a-Service (IaaS)

- Cloud provides low-level resources like virtual machines, storage, and networking
 - Users install and maintain the operating system and applications
- **Examples**
 - AWS EC2
 - Google Compute Engine
 - Microsoft Azure Virtual Machines
 - IBM Cloud Infrastructure
- **Benefits**
 - Flexibility in managing resources
 - Scalability to meet demand
 - Cost-effective, pay for what you use
- **Use Cases**
 - Hosting websites and applications
 - Data analysis and processing
 - Development and testing environments



Platform-as-a-Service

- **Problem:** assembling your own software stack requires work

- Install
- Configure
- Manage dependencies (e.g., incompatible versions)

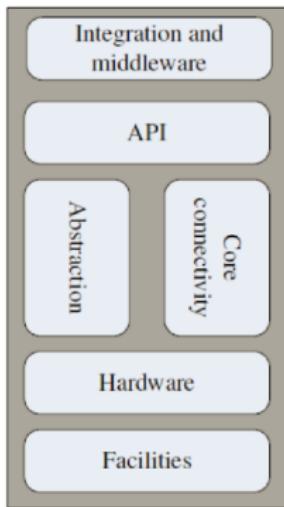
- **Solution:** get a pre-built software stack

- Software stack as a virtualization solution (e.g., VMware or Docker)
 - Pre-installed OS
 - Libraries
 - Application software

- **Business model built around this**

- E.g.,
 - Google App Engine
 - Managed Hadoop
- E.g., pre-built images for Hadoop
 - Hortonworks, Cloudera
- E.g., pre-built distributions for Linux
 - RedHat, Gentoo, CentOS

Platform as a Service



Software-as-a-Service (SaaS)

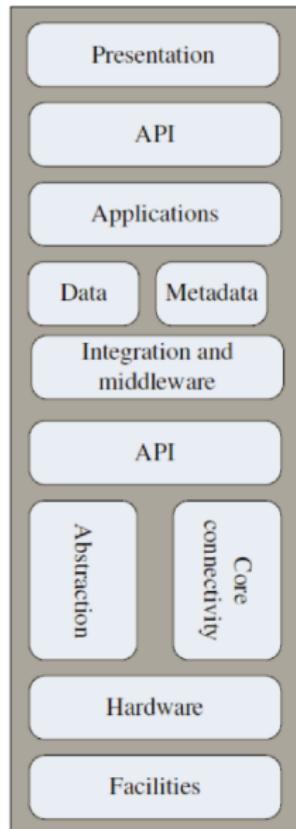
- **Cloud provides the application**

- No need to install on your machine
- Use directly from the cloud
- Examples include:
 - *Dropbox*: Access and share files without local storage
 - *Salesforce*: Manage customer relationships online
 - Any app running in a browser: Google Docs or Microsoft 365

- **Benefits**

- Accessibility from any device with internet connectivity
- Automatic updates and maintenance by the provider
- Scalability for growing user needs
- Cost-effectiveness by reducing physical hardware and software installations

Software as a Service



X-as-a-Service

- After 2010, business model of **X-as-a-Service (XaaS)**
 - Mobility-as-a-Service (e.g., Uber)
 - Gaming-as-a-Service (e.g., Google Stadia)
 - Storage-as-a-Service (e.g., S3, Google Drive)
 - Desktop-as-a-Service (e.g., AWS AppStream)
 - Marketing-as-a-Service
 - Banking-as-a-Service
 - ..

Cloud Deployment Models

- **Private**
 - Internal cloud hosted on organizational premises
 - E.g., company's data center running virtualized services
- **Public**
 - External cloud hosted by third-party providers
 - E.g., AWS, Azure, GCP offering compute and storage to clients
- **Hybrid cloud**
 - Combines private and public environments to optimize cost, security, and scalability
 - Sensitive workloads stay internal
 - Scalable tasks move to the public cloud
- **On-premises resources**
 - E.g., corporate servers in a building
- **Off-premises resources**
 - E.g., cloud provider's distributed data centers

