Data-intensive computing systems

Cloud Computing

University of Verona
Computer Science Department

Damiano Carra

Acknowledgements

- *Credits*
  - *Part of the course material is based on slides provided by the following authors*
    - *Altaf Hussain, Andreas Haeberlen, Zachary G. Ives*

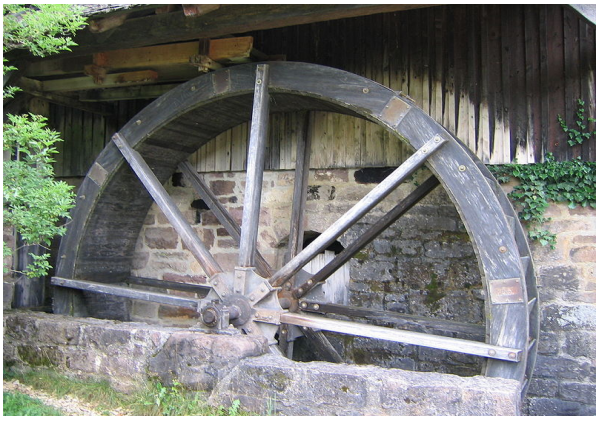
History: The early days

- Cloud computing: A new term for a concept that has been around since the 1960s

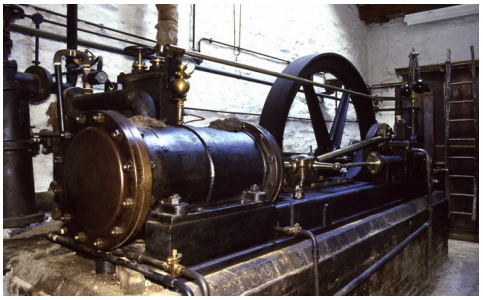
- Who invented it? No agreement. Some candidates:
  - 1961: John McCarthy (Stanford professor and inventor of Lisp)
    - Computing may someday be organized as a public utility, just as the telephone system is organized as a public utility
  - 1963: J.C.R. Licklider (contributed key ideas to ARPANET)
  - 1966: Douglas Parkhill
    - published a book on "The Challenge of the Computer Utility" in 1966

The power plant analogy

- It used to be that everyone had their own power source
  - Challenges are similar to the cluster: Needs large up-front investment, expertise to operate, difficult to scale up/down...



Waterwheel at the Neuhausen ob Eck Open-Air Museum



Steam engine at Stott Park Bobbin Mill

Scaling the power plant

- Then people started to build large, centralized power plants with very large capacity...

Metered usage model

- Power plants are connected to customers by a network
- Usage is metered, and everyone (basically) pays only for what they actually use

Power source **Network** **Metering device** **Customer**

Why is this a good thing?

Electricity	Computing
❑ Economies of scale	
– Cheaper to run one big power plant than many small ones	– Cheaper to run one big data center than many small ones
❑ Statistical multiplexing	
– High utilization!	– High utilization!
❑ No up-front commitment	
– No investment in generator; pay-as-you-go model	– No investment in data center; pay-as-you-go model
❑ Scalability	
– Thousands of kilowatts available on demand; add more within seconds	– Thousands of computers available on demand; add more within seconds

What is Cloud Computing?

- ❑ When computing services are provided over the Internet rather then locally on a user's own machine
- ❑ Computation is run on an supporting infrastructure which is independent of the applications themselves
- ❑ The infrastructure can take on many forms, but to the end user, the implementation is irrelevant, hence the "cloud" abstraction
- ❑ In addition, the platform provides on demand services, that are always on, anywhere, anytime and any place
- ❑ Pay for use and as needed, elastic (scale up and down in capacity and functionalities)
- ❑ The hardware and software services are available to the general public, enterprises, corporations and businesses markets

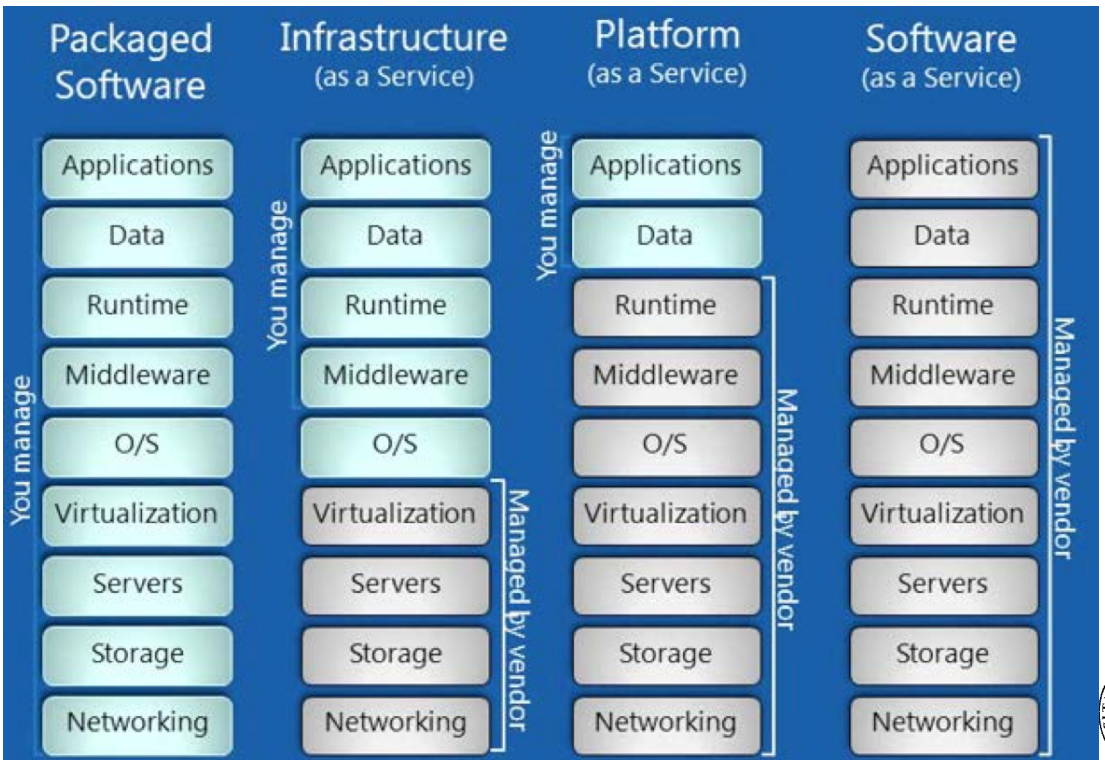
Another definition

- According to NIST:
  - 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.
- Essential characteristics:
  - On-demand self service
  - Broad network access
  - Resource pooling
  - Rapid elasticity
  - Measured service

Everything as a Service

- What kind of service does the cloud provide?
  - Does it offer an entire application, or just resources?
  - If resources, what kind / level of abstraction?
- Three types commonly distinguished:
  - Software as a service (SaaS)
    - Analogy: Restaurant. Prepares&serves entire meal, does the dishes, ...
  - Platform as a service (PaaS)
    - Analogy: Take-out food. Prepares meal, but does not serve it
  - Infrastructure as a service (IaaS)
    - Analogy: Grocery store. Provides raw ingredients
  - Other xaaS types have been defined, but are less common
    - Desktop, Backend, Communication, Network, Monitoring, ...

Delivery models: who manages what



Software as a Service (SaaS)

- What is SaaS?
  - Software is provided to end users in an "On-demand" fashion
  - Reduces upfront costs, i.e. buying multiple licenses
  - "Utility-based" computing

- SaaS Delivery Model
  - Increasingly popular with Small/Medium Enterprises (SMEs) and Independent Software Vendors (ISV)
  - No hardware or software (e.g. OS) to manage
  - Service delivered through a browser

Software as a Service (SaaS)

- Advantages
  - Pay per use
  - Instant Scalability
  - Security
  - Reliability
  - APIs
- Examples
  - iCloud
  - Drop box
  - Amazon S3
  - Google Docs

Platform as a Service (PaaS)

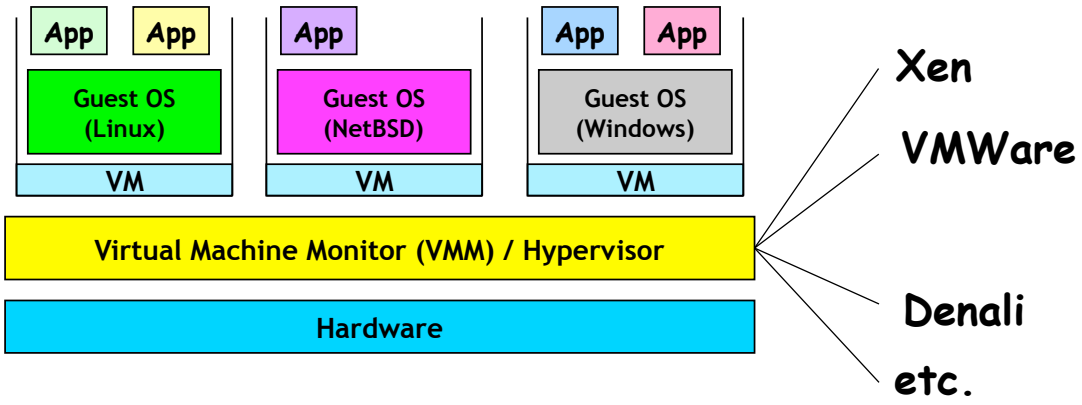
- What is PaaS?
  - Service Provider provide cloud environment for development and deployment software
  - Consumer can avoid preparing and manage complex environment
  - Some PaaS offers, the underlying compute and storage resources scale automatically to match application demand such that the cloud user does not have to allocate resources manually
  - With the model user can run web service, database, and development and testing tools
- PaaS Service
  - Storage Service
  - Compute Service

Infrastructure as a Service (IaaS)

- What is IaaS?
  - User can get dynamic infrastructure facility by the provider in virtualized form
    - Computing hardware
      - E.g. Client Storage Space, Processing power needs
    - Virtualization
      - E.g. VM Ware, VirtualBox
    - Networking
      - Security, communication speeds, servers
    - Utility Computing
      - E.g. Charging by hour, gigabyte, process load
- PaaS Providers:
  - Microsoft
  - Amazon
  - IBM

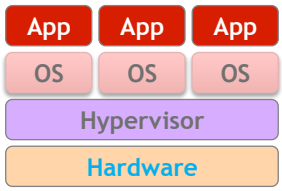
Enabling Technologies: Virtual Machines

- VM technology allows multiple virtual machines to run on a single physical machine



Providing and Sharing Resources: Virtualization

- Virtual workspaces:
  - An abstraction of an execution environment
  - Resource quota (e.g. CPU, memory share),
  - Software configuration (e.g. OS, provided services).
- Implement on Virtual Machines (VMs):
  - Abstraction of a physical host machine,
  - Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs,
  - VMWare, Xen, IBM Virtual BOX etc.
- Provide infrastructure API:
  - Plug-ins to hardware/support structures



Virtualized Stack

Virtualization in the cloud

- Gives cloud provider a lot of flexibility
  - Can produce VMs with different capabilities
  - Can migrate VMs if necessary (e.g., for maintenance)
  - Can increase load by overcommitting resources
- Provides security and isolation
  - Programs in one VM cannot influence programs in another
- Convenient for users
  - Complete control over the virtual 'hardware' (can install own operating system own applications, ...)
- But: Performance may be hard to predict
  - Load changes in other VMs on the same physical machine may affect the performance seen by the customer

Types of Cloud (According to Service Availability)

- Private Cloud:
  - Access to these services are provided through a dedicated Government Intranet/ Organization and is not accessible from the Public Internet

- Public Cloud:
  - The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services

Types of Cloud (According to Service Availability)

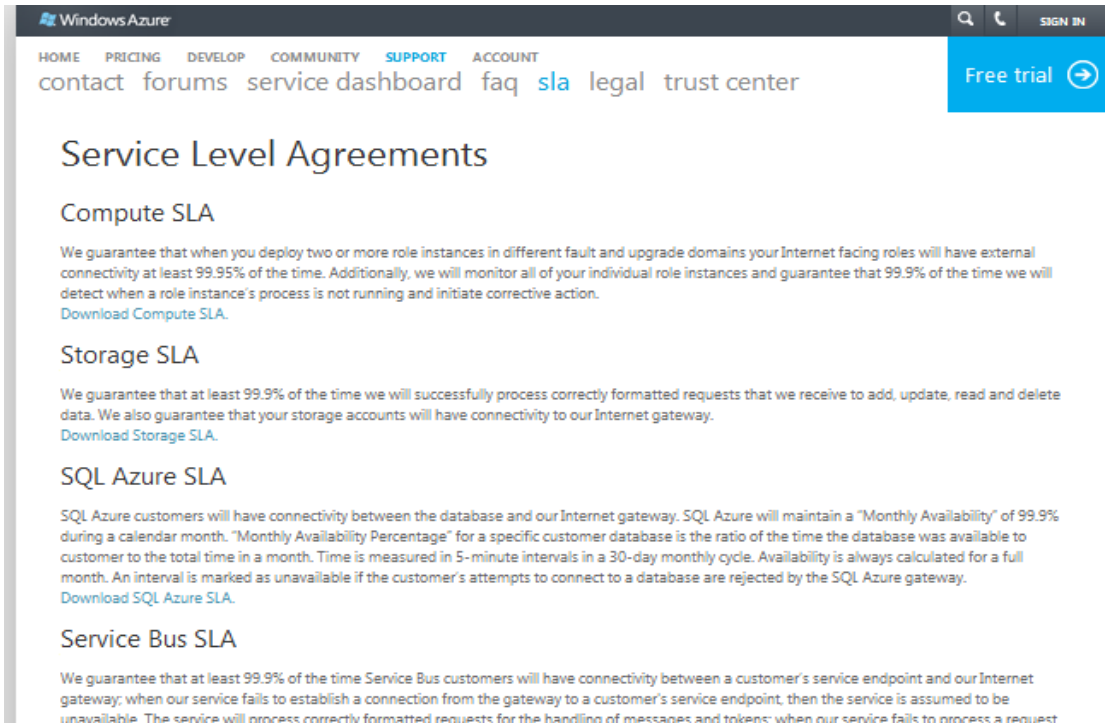
- Community Cloud:
  - The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise

- Hybrid Cloud:
  - The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting)

SLA: Service Level Agreement

- A service-level agreement (SLA) is a part of a service contract where the level of service is formally defined
- In practice, the term SLA is sometimes used to refer to the contracted delivery time (of the service) or performance
- As an example, internet service providers will commonly include service level agreements within the terms of their contracts with customers to define the level(s) of service being sold in plain language terms

SLA: Service Level Agreement



Windows Azure

SIGN IN

HOME PRICING DEVELOP COMMUNITY SUPPORT ACCOUNT

contact forums service dashboard faq sla legal trust center

Free trial

Service Level Agreements

Compute SLA

We guarantee that when you deploy two or more role instances in different fault and upgrade domains your Internet facing roles will have external connectivity at least 99.95% of the time. Additionally, we will monitor all of your individual role instances and guarantee that 99.9% of the time we will detect when a role instance's process is not running and initiate corrective action.
Download Compute SLA.

Storage SLA

We guarantee that at least 99.9% of the time we will successfully process correctly formatted requests that we receive to add, update, read and delete data. We also guarantee that your storage accounts will have connectivity to our Internet gateway.
Download Storage SLA.

SQL Azure SLA

SQL Azure customers will have connectivity between the database and our Internet gateway. SQL Azure will maintain a "Monthly Availability" of 99.9% during a calendar month. "Monthly Availability Percentage" for a specific customer database is the ratio of the time the database was available to customer to the total time in a month. Time is measured in 5-minute intervals in a 30-day monthly cycle. Availability is always calculated for a full month. An interval is marked as unavailable if the customer's attempts to connect to a database are rejected by the SQL Azure gateway.
Download SQL Azure SLA.

Service Bus SLA

We guarantee that at least 99.9% of the time Service Bus customers will have connectivity between a customer's service endpoint and our Internet gateway; when our service fails to establish a connection from the gateway to a customer's service endpoint, then the service is assumed to be unavailable. The service will process correctly formatted requests for the handling of messages and tokens; when our service fails to process a request

Source: https://www.windowsazure.com/en-us/support/sla/

Is the cloud good for everything?

- Sometimes it is problematic, e.g., because of auditability requirements
- Example: Processing medical records
  - HIPAA (Health Insurance Portability and Accountability Act) privacy and security rule
- Example: Processing financial information
  - Sarbanes-Oxley act
- Would you put your medical data on the cloud?
  - Why / why not?

Is the cloud good for everything? (cont'd)

- Should you go for the public cloud if the unit CPU cycle/bit price is higher than a home- grown solution?
  - Need to pay for Amazon's commodity hardware
  - But also for sophisticated cooling, energy provision, smart distributed systems folks working there, Amazon profits,...
  - Might end up more expensive per CPU/data unit
  - Otherwise, it is a no-brainer to use cloud...
- It depends
  - If cloud costs the same and the load is perfectly smooth then it is the same
  - But what if the cloud is more expensive per CPU/data unit and the load is variable?

Is the cloud good for everything? (cont'd)

- Consider a car
  - Buy (lease) for EUR 10 per day
  - vs. Rent a car for EUR 30 a day
  - If you need a car for 2 days in a month, buying would be much more costly than renting

  → It depends on the load/demand

- Turns out that in many business cases a hybrid solution is very attractive
  - You own a daily commute car
  - But you rent a van to cover unusual demand (e.g., to move)
- Might use public cloud to serve load spikes
  - Christmas shopping time, slashdot effects, etc.

10 obstacles and opportunities

1. Availability
   - What happens to my business if there is an outage in the cloud?

Service	Duration	Date
S3	6-8 hrs	7/20/08
AppEngine	5 hrs	6/17/08
Gmail	1.5 hrs	8/11/08
Azure	22 hrs	3/13/09
Intuit	36 hrs	6/16/10
EBS	>3 days	4/21/11
ECC	~2 hrs	6/30/12

Some recent cloud outages

2. Data lock-in
   - How do I move my data from one cloud to another?
3. Data confidentiality and auditability
   - How do I make sure that the cloud doesn't leak my confidential data?
   - Can I comply with (country-specific) regulations?

10 obstacles and opportunities

4. Data transfer bottlenecks
   - How do I copy large amounts of data from/to the cloud?
   - Example: 10 TB from UC Berkeley to Amazon in Seattle, WA
   - Motivated Import/Export feature on AWS

Method	Time
Internet (20Mbps)	45 days
FedEx	1 day

Time to transfer 10TB [AF10]

5. Performance unpredictability
   - Example: VMs sharing the same disk → I/O interference
   - Example: HPC tasks that require coordinated scheduling

Primitive	Mean perf.	Std dev
Memory bandwidth	1.3GB/s	0.05GB/s (4%)
Disk bandwidth	55MB/s	9MB/s (16%)

Performance of 75 EC2 instances in benchmarks

10 obstacles and opportunities

6. Scalable storage
   - Cloud model (short-term usage, no up-front cost, infinite capacity on demand) does not fit persistent storage well

7. Bugs in large distributed systems
   - Many errors cannot be reproduced in smaller configs

8. Scaling quickly
   - Problem: Boot time; idle power
   - Fine-grain accounting?

10 obstacles and opportunities

9. Reputation fate sharing
   - One customer's bad behavior can affect the reputation of others using the same cloud
   - Example: Spam blacklisting, FBI raid after criminal activity

10. Software licensing
   - What if licenses are for specific computers?
     - Example: Microsoft Windows
   - How to scale number of licenses up/down?
     - Need pay-as-you-go model as well