CSE6331: Cloud Computing

Leonidas Fegaras

University of Texas at Arlington

Cloud Computing Fundamentals

Based on:
J. Freire's class notes on Big Data `http://vgc.poly.edu/~juliana/courses/BigData2016/`
D. Abadi: Data Management in the Cloud: Limitations and Opportunities. IEEE Data Eng. Bull. 2009
D. Maier: Data Management in the Cloud `https://piazza.com/pdx/spring2018/cs410510cloud/home`
VMWare white paper: Virtualization Overview

Computing as a Utility

- Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that are readily available from a large infrastructure (data center)
- What kind of resources?
  - network bandwidth
  - data storage
  - applications
  - services

  these resources can be rapidly provisioned and released with minimal management effort or service provider interaction
- Service-based view: everything as a service
- Key ideas:
  - Move computing into large shared data centers
  - Gives the illusion of infinite resources on demand
  - Utility computing: pay-as-you-go (as needed), no upfront cost
  - Elasticity: scale up or down, based on current needs
  - Automatically manages fault tolerance, redundancy, storage, load balancing, geographic distribution, security, software updates, etc

Why?

- Sharing resources among applications utilizes these resources better
  - while the resource needs of an application may vary dramatically, the sum of the resource needs of many independent applications varies much less
- Upfront setup cost is amortized over many applications
- Illusion of infinite computing resources
- No up-front cost or commitment by users
  - companies can start small (demand unknown in advance)
  - increase resources only when there is an increase in need
- Cost is proportional to use
  - "cost associativity": 1000 EC2 machines for 1 hour = 1 EC2 machine for 1000 hours
- Rapid prototyping and market testing
- Improved service levels, fault tolerance, and availability
- Better security infrastructure, better protection against network attacks
- Allows self-service deployment

Good for...

- Web hosting
- Deployment of data-intensive web services
    - especially those with unpredictable load
    - ... that require high availability
- Storing massive amounts of data (backups, videos, logs, etc)
- Off-line analysis of massive data (map-reduce, Spark)
- Large computing jobs on demand
    - jobs that require many CPUs

Alternatives

- HPC (High-Performance Computing)
- Distributed Computing
- Distributed Databases
- Grid computing
- Peer-to-peer computing
  - BitTorrent, multi-user games, Skype

Current Trends

- Cloud providers: Amazon, Google, Microsoft, ...
- Public clouds: Amazon Web Services, Google Cloud Platform, Microsoft Azure, ...
  - they provide infrastructure and/or deployment platform
  - they offer them as web services and are billed on a utility basis
- Amazon web services
  - Simple Storage Service (SS3)
  - Elastic Compute Cloud (EC2)
  - Elastic MapReduce
- Google App Engine
  - plus many office applications (Google docs, gmail, etc)

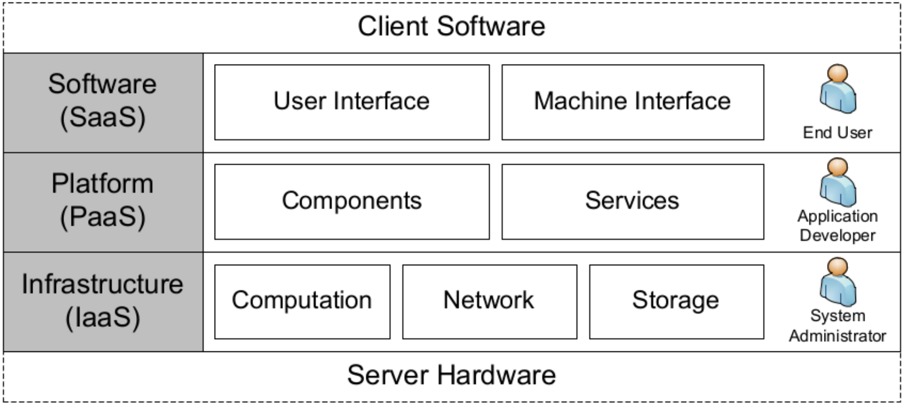
Challenges

- Data security
  - Have to depend on the security provided by the cloud host
  - Data protection against other applications
  - Protection against the cloud host
  - A cloud is a high value target to hackers
  - Data ownership issues
  - Moving sensitive data to the cloud
- Performance variability
  - Lack of Quality of Service (QoS) guarantees
- Data transfer bottlenecks
- Financial cost

Service Offering Models

- Software as a Service (SaaS)
  - applications running on a cloud infrastructure
  - users: any end user
  - eg, some web applications, such as Google Gmail
- Platform as a Service (PaaS)
  - deploy onto the cloud infrastructure user-defined applications created using a set of provided libraries, services, and tools
  - users: application developers
  - eg, Google App Engine
- Infrastructure as a Service (IaaS)
  - deploy and run arbitrary software, including operating systems and applications; share computing resources, storage, network
  - users: IT architects, data analysts
  - eg, Amazon Web Services (AWS)

Service Offering Models



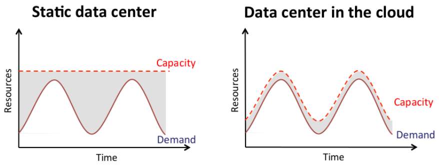
Source: Wikipedia (http://www.wikipedia.org)

Horizontal vs. Vertical Scaling

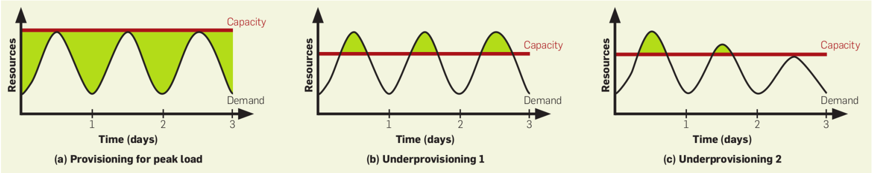
- Horizontal scaling (scale out): Ability to run many instances of a task in parallel
  - Usually with modest amounts of data each
  - Often with relaxed consistency
  - Focus of many NoSQL systems
- Vertical Scaling (scale up): Ability to run large instances of a task
  - Large amounts of data
  - Queries that last for hours, hence fault tolerance is important
  - Focus of Map-Reduce and some scalable SQL systems

Economics of Cloud Providers

- Pay by use instead of provisioning for peak

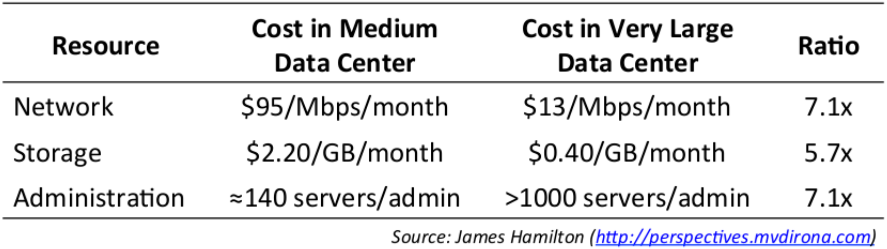


- Risk of over-provisioning: underutilization
- Heavy penalty for under-provisioning



Slide Credit: RAD Lab, UC Berkeley

Economics of Cloud Providers (cont.)



Resource	Cost in Medium Data Center	Cost in Very Large Data Center	Ratio
Network	\$95/Mbps/month	\$13/Mbps/month	7.1x
Storage	\$2.20/GB/month	\$0.40/GB/month	5.7x
Administration	≈140 servers/admin	>1000 servers/admin	7.1x

Source: James Hamilton (http://perspectives.mvdirona.com)

- Cloud computing is 5-7x cheaper than traditional in-house computing
- Power/cooling costs: approx double cost of storage, CPU, network
- Added benefits (to cloud providers)
  - utilize off-peak capacity (Amazon)
  - sell .NET tools (Microsoft)
  - reuse existing infrastructure (Google)

Virtualization

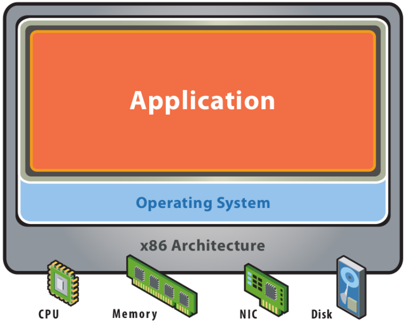
- The underlying core technology of cloud computing
- Virtual resources are abstracted from physical resources
    - hardware platform, software, memory, storage, network
    - fine-grain, lightweight, flexible and dynamic
- Relevance to cloud computing:
    - centralize and ease administrative tasks
    - improve scalability and work loads
    - increase stability and fault-tolerance
    - provide standardized, homogenous computing platforms through hardware virtualization (virtual machines)

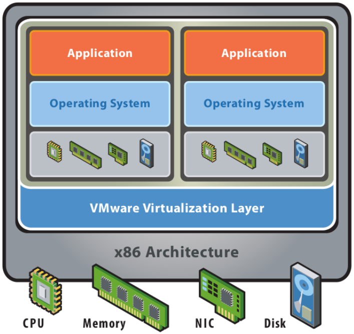
Virtualization (cont.)

- The physical infrastructure owned by the service provider is shared among many users, increasing the resource utilization
- Partitioning: may run multiple operating systems on a single physical system and share the underlying hardware resources
- Benefits: isolation of virtual machines and hardware-independence
- Virtual machines are highly portable
- Hypervisor: software that provides virtual partitioning capabilities
    - bare metal approach: runs directly on hardware

Virtualization Overview

Slide credit: VMWare white paper: Virtualization Overview





Before Virtualization:

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure

After Virtualization:

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

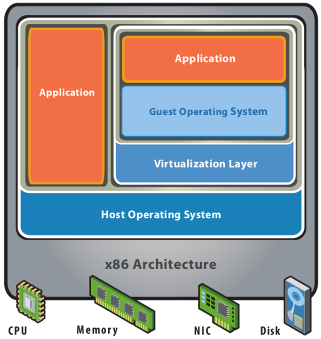
Virtualization (cont.)

- Hosted Virtualization
    - The virtualization software relies on the host OS to provide the services to talk directly to the underlying hardware
    - The guest OS cannot communicate directly with the underlying physical infrastructure
    - The hypervisor interacts with the host's hardware resources (through the host OS) and acts as a platform for the guest OS
    - VirtualBox
- Paravirtualization
    - The guest OS is aware that it is running within a virtualized environment, and has been modified to exploit this
    - VirtualBox guest editions

Virtualization (cont.)

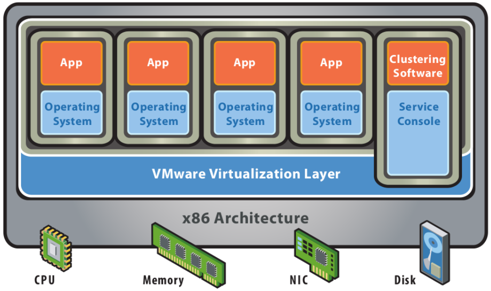
- Bare-Metal (Hypervisor) Virtualization
  - No host OS
  - The guest OS access the underlying hardware directly through an API
  - More efficient than hosted architectures: greater scalability, robustness, and performance
  - VMWare, Xen
- Hardware-Assisted Virtualization
  - Current CPUs support virtualization through hardware extensions
  - Intel VT-x, and AMD AMD-v

Virtualization Architecture
Slide credit: VMWare white paper: Virtualization Overview



Hosted Architecture

- Installs and runs as an application
- Relies on host OS for device support and physical resource management

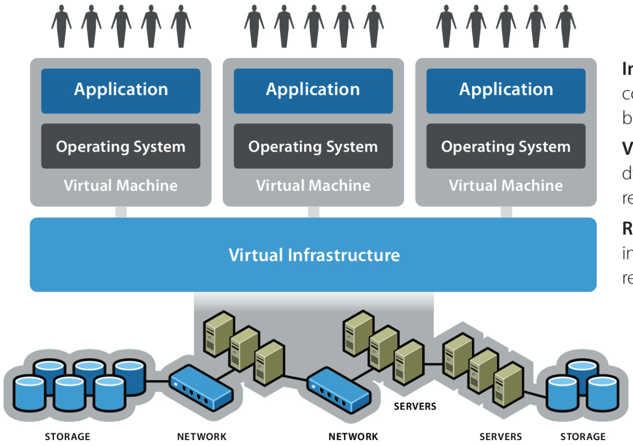


Bare-Metal (Hypervisor) Architecture

- Lean virtualization-centric kernel
- Service Console for agents and helper applications

Virtualization in a Data Center

Slide credit: VMWare white paper: Virtualization Overview



Infrastructure is what connects resources to your business.

Virtual Infrastructure is a dynamic mapping of your resources to your business.

Result: decreased costs and increased efficiencies and responsiveness

Transforms farms of individual servers, storage, and networking into a pool of computing resources

Large Scale Computing

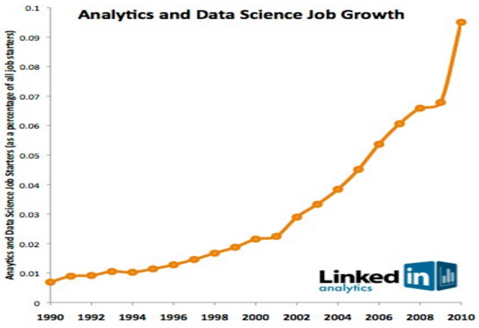
- Distributed systems
  - Parallel or distributed computations on multiple disks and nodes
    - often using message passing interface (MPI)
    - and parallel virtual machines (PVM)
  - Data are stored in a central location and copied to processing nodes when needed
  - Brings the data to the computation
  - Good for computationally intense tasks
- Cloud Computing
  - Run on clusters of share-nothing commodity computers
  - Use distributed storage
  - The data are already distributed when they are stored
  - Distributed computation: run computations where the data are
  - Brings the computation to the data
  - Good for data intense tasks

What is Big Data?

The three V's of big data: Volume, Variety, and Velocity

- High Volume (too big): petabyte-scale collections
- High Variety (too hard): data comes from a variety of sources in different formats
  - irregular, unstructured, or incomplete data
  - Data need to be cleaned, processed, and integrated
- High Velocity (too fast): data need to be processed quickly (throughput) with low latency
  - Batch streaming, Real-time analytics
- Also veracity: correctness and accuracy of data

High Demand for Data Scientists



Linkedin Dec 2017: From the top 10 emerging positions:

1) Machine Learning Engineer (9.8X growth)
2) Data Scientist (6.5X)
5) Big Data Developer (5.5X)
8) Director of Data Science (4.9X)

How Big?

From Lin and Dyer, 2010:

- Google:
  - grew from processing 100 TB of data a day with MapReduce in 2004 to processing 20 PB a day with MapReduce in 2008
- Facebook:
  - 1.1 billion users the site each month
  - 2.5 PB of user data, growing at about 15 TB per day
- Twitter:
  - 517 million accounts, 250 million tweets/day
- eBay's two enormous data warehouses:
  - one with 2 PB of user data, and the other with 6.5 PB of user data spanning 170 trillion records and growing by 150 billion new records per day

Data-intensive Scientific Computing

- Current scientific applications must analyze enormous amounts of data using complex mathematical data processing methods
- Necessary for scientific breakthroughs
- Capable of generating several petabytes of data per
  - Large Hadron Collider at CERN
  - Astronomy's Pan-STARRS5 array of celestial telescopes
  - Sloan Sky Survey, genome, climate data, oceanography

Big Data is not New, but has Become More Important

- More data are now widely available
  - Web, scientific data, social network data, etc
- Increased variety
  - logs, XML, JSON, ...
- Many more companies try to collect data and turn them into value
  - targeted advertising
- Computing is cheaper and easier to access
  - server with 64 cores, 512GB RAM $11k
  - cluster with 1000 cores $150k
  - cloud computing: Amazon EC2, Microsoft Azure
- Very large data centers
- Better virtualization technology
- Broadband Internet

Data Management in the Cloud

Not all Big Data management applications are well-suited for deployment in the cloud:

- Traditional transactional databases (OLTP): NO
  - Good for both read- and write-intensive applications
  - Do not normally use shared-nothing architectures
  - Hard to maintain ACID guarantees in the face of data replication
  - There are enormous risks in storing transactional data on an untrusted host
- Data analysis systems (OLAP): YES
  - Tend to be read-only with occasional batch inserts
  - Shared-nothing architecture is a good match
  - Data analysis workloads consist of many large scans and aggregations
  - ACID guarantees are typically not needed
  - Easy to parallelize
  - Sensitive data can be left out of the analysis

Wish List

- Scalability
  - Scan 100 TB on 1 node @ 50 MB/sec = 23 days
  - Scan 100 TB on 1000-node cluster = 33 minutes
  - Use divide-and-conquer: data partitioning
    - Yahoo! runs a 4000 node Hadoop cluster
    - Overall, there are 38,000 nodes running Hadoop at Yahoo! (2009)
  - Data analysis is elastic, but only if workload is parallelizable
- Fault tolerance
  - Fault tolerant read-only query: does not have to restart it if one of the processing nodes fails
  - Complex queries can involve thousands of nodes and can take hours to complete
  - Clouds are typically built on top of cheap, commodity hardware, for which failure is not uncommon
  - The probability of a failure occurring during a long-running data analysis task is relatively high
  - To contrast: parallel database systems restart a query upon a failure
    - Google reports an average of 1.2 failures per analysis job

Wish List (cont.)

- Performance: optimize resources
  - Since you pay for resources
  - Especially, minimize the number of nodes
- Ability to run in a heterogeneous environment
  - Performance of cloud compute nodes is often not consistent
  - The time to complete a query will be equal to time for the slowest compute node to complete its assigned task
  - How do we tell if a node is slow or faulty?
- Ability to operate on encrypted data
- Ability to work with other data analysis software

Big Data: New Technologies

- Infrastructure
  - New computing paradigms: Hadoop MapReduce, Spark
  - New storage solutions: NoSQL, column stores, Big Table
  - New languages: Hive, JAQL, Pig Latin
  - We will study these and how they relate to previous technologies
- Algorithms and techniques
  - New infrastructure demands new approaches to explore data
  - We will study algorithms to process and explore data in Big-Data environments

NoSQL Systems

- Stands for: Not Only SQL
- They don't use SQL
- No schema
- Simple and scalable data management systems
- Typically oriented towards clusters and cloud (but not all)
- No "one size fits all" philosophy
- Most are open-source
- Various types:
  - Key-value stores
  - Document stores
  - Extensible record stores
- Databases vs NoSQL:
  - Database: complex / concurrent
  - NoSQL: simple / scalable
  - Transactional vs analytical

Why not use a Database?

- They are good for transactional data management
  - banking, airline reservation, e-commerce, ...
- Write-intensive, require ACID
- Do not typically use shared-nothing architectures
- Hard to maintain ACID guarantees in the face of data replication over large geographic distances
- Today's SQL databases cannot scale to the thousands of nodes deployed in the cloud context
- Hard to support multiple, distributed updaters to the same data set
- Hard to replicate huge data sets for availability
- There are risks in storing transactional data on an untrusted host
- Full ACID guarantees are typically not required in analytical applications
- Conclusion: not appropriate for the cloud

How to choose between Databases and NoSQL

- How critical is consistency?
- Do answers need to be precise?
- Interactive vs. batch data analysis
- Does the data partition easily?
- How complex is the data?
  - flat records, raw files, nested structure
- How much heterogeneity in the data?
- How much data is there?
- What is the update pattern?
- How sensitive and valuable is the data?
- How complex is the logic?
- How complex are the access patterns?
- Can the data in use fit in memory?