Introduction to Big Data

Dr. Animesh Chaturvedi

Assistant Professor: **IIIT Dharwad**

Young Researcher: **Heidelberg Laureate Forum** and **Pingala Interaction in Computing**

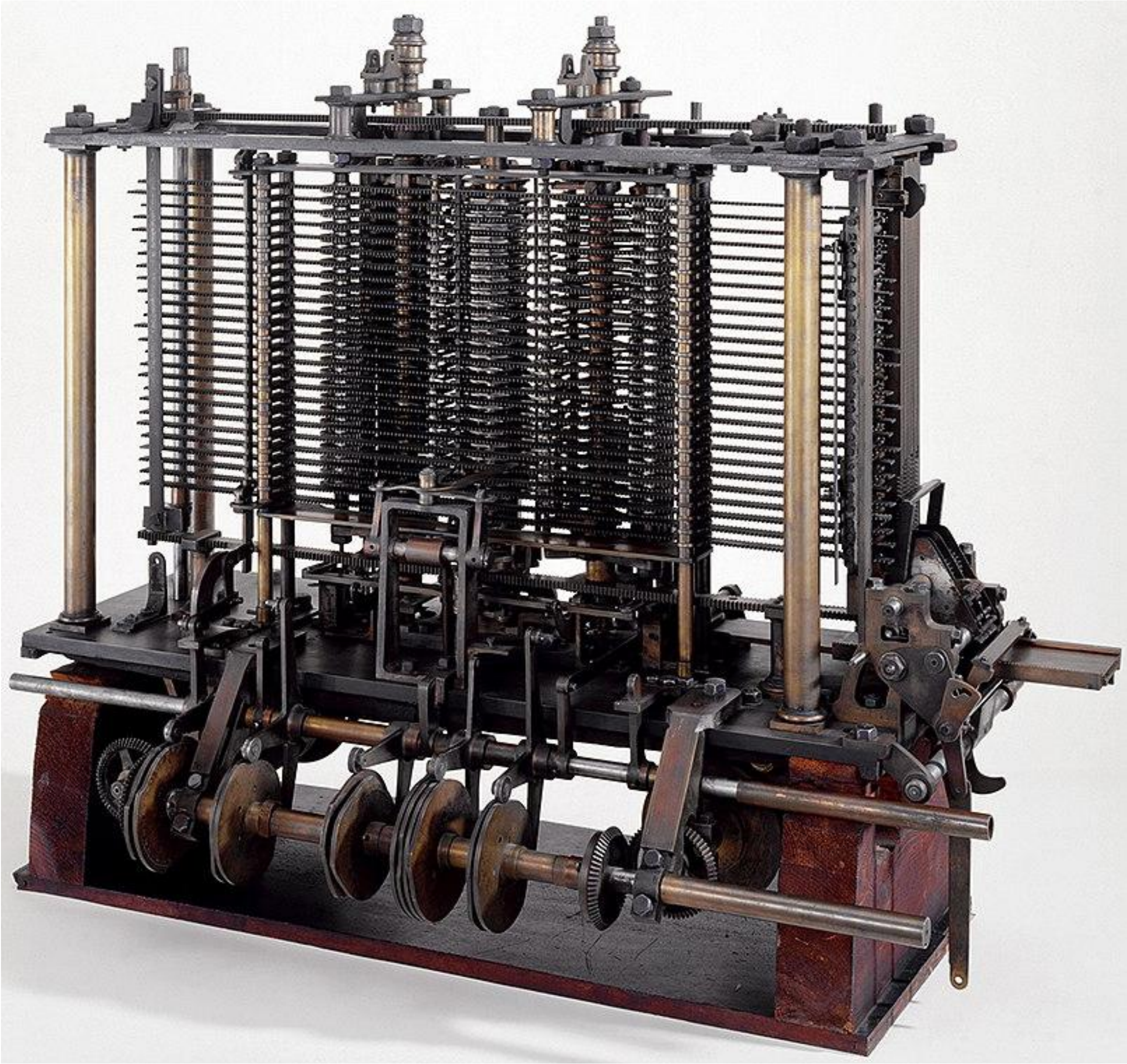
Young Scientist: **Lindau Nobel Laureate Meetings**

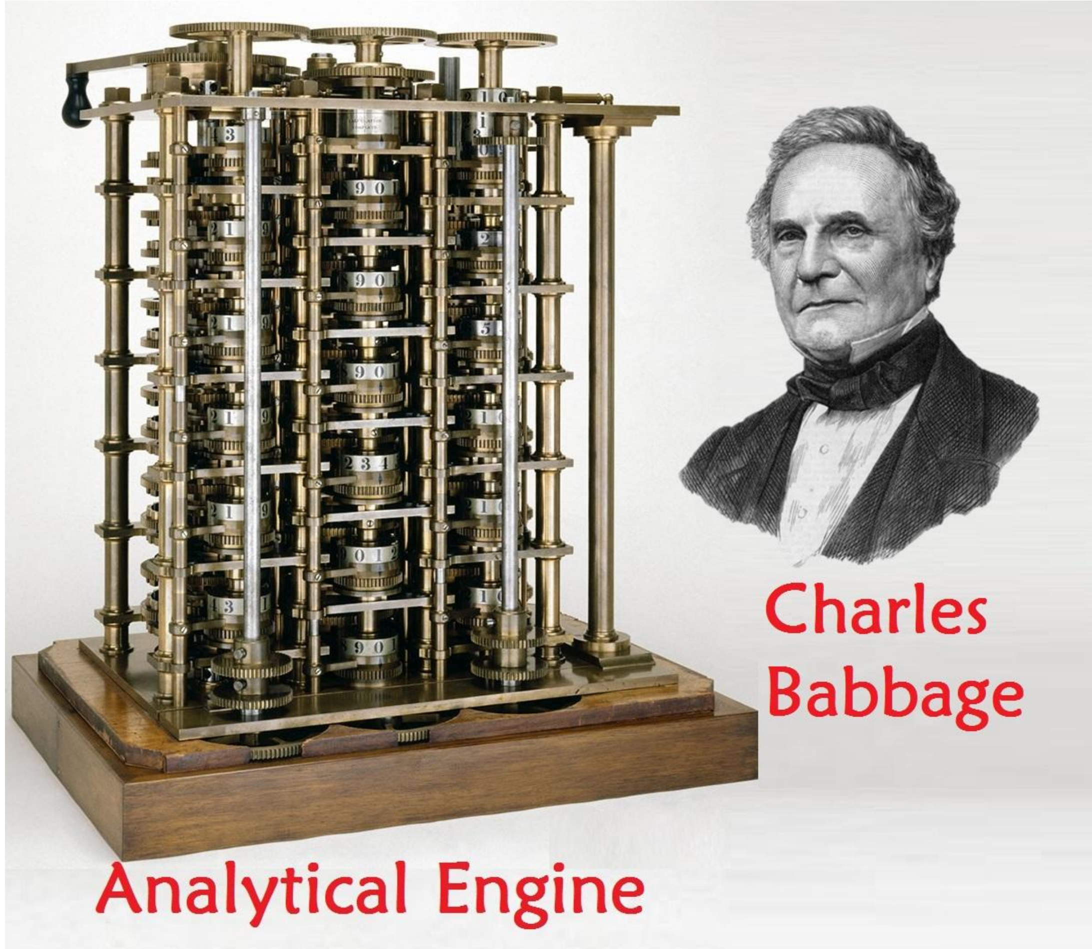
Postdoc: **King's College London** & **The Alan Turing Institute**

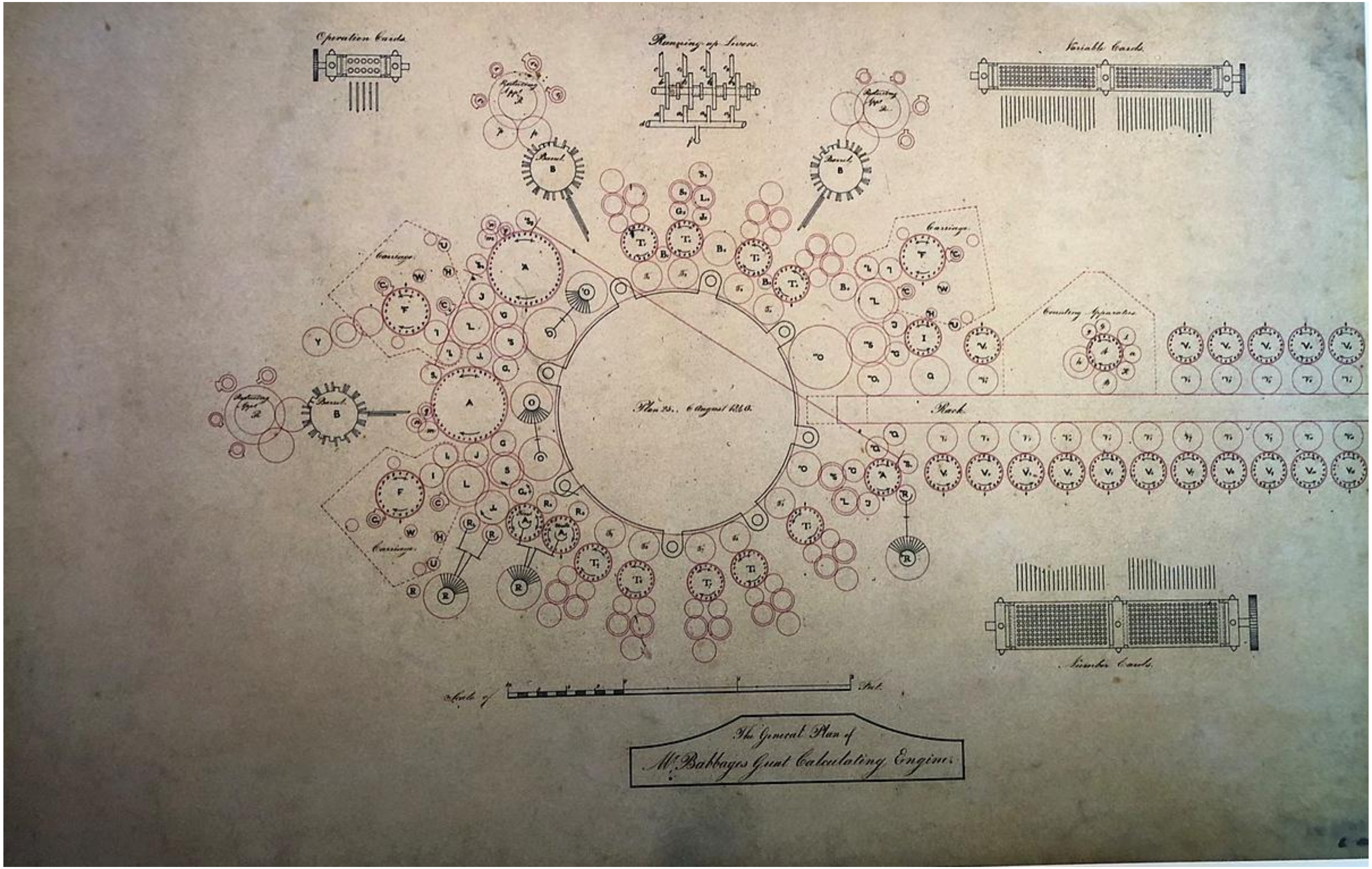
PhD: **IIT Indore** MTech: **IIITDM Jabalpur**

History of Computing

Analytical Engine 1837



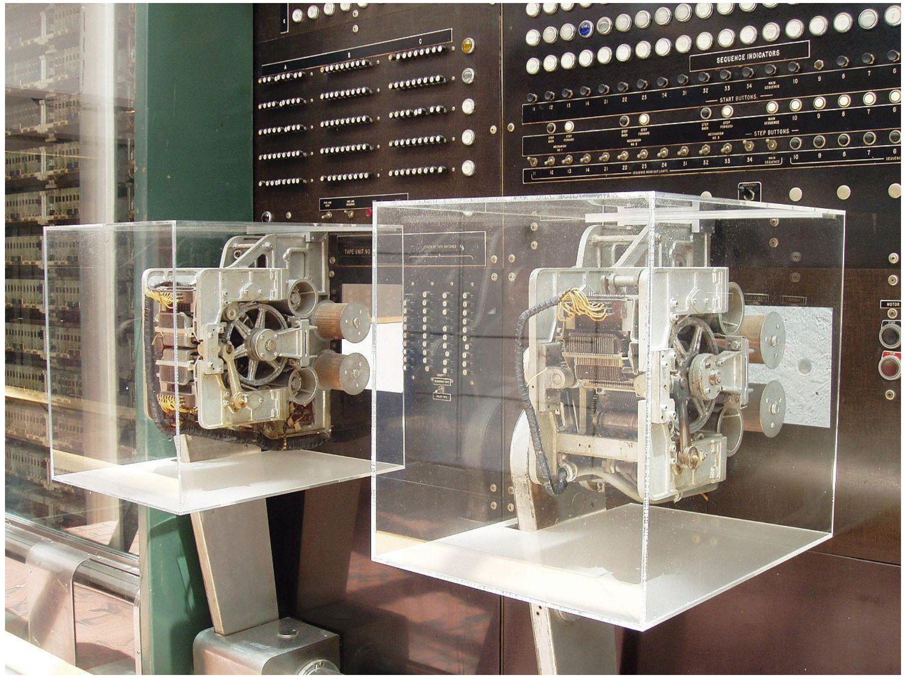




Plan diagram of the Analytical Engine from 1840

Harvard Mark-I, 1944

- IBM Automatic Sequence Controlled Calculator (ASCC)
- General-purpose electromechanical computer
- One of the first programs to run on the Mark I was initiated on 29 March 1944 [1] by John von Neumann.



Mainframe computers and Time Sharing

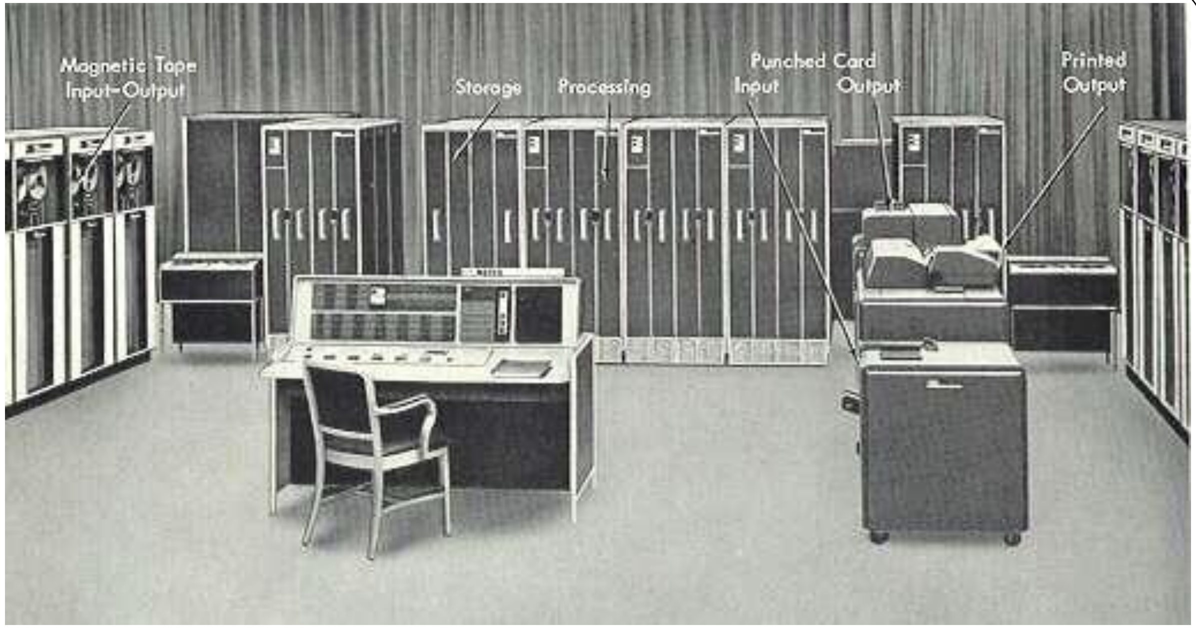
- **The 1950s**
  - large-scale mainframe computers
  - Practice of sharing CPU time on a mainframe became known in the industry as time-sharing.
- **Mid 70s**
  - time-sharing was popularly known as RJE (Remote Job Entry) by IBM and DEC.

http://en.wikipedia.org/wiki/Cloud_computing

Joe Thompson and Whirlwind, 1951 | MIT

- Among the first digital electronic computers that operated in real-time for output, and the first that was not simply an electronic replacement of older mechanical systems.
- One of the first computers to calculate in parallel (rather than serial), and was the first to use magnetic-core memory.

Timesharing (1960's)



- MIT IBM 7090 hardware
  - Transistorized version of IBM 709
  - 32K of 36-bit words of magnetic storage
  - 3 channels with 19 tape units
  - 60 cycle/sec accounting and interrupt clock
  - Special mode for memory protection, dynamic relocation and trapping of all user I/O
  - Time-sharing between 4 users (3 typewriters)
  - Single OS, a few simultaneous users
- Four significant design features of the foreground system for users:
  1. Enables development of programs in languages compatible with the background system
  2. Can develop a private file of programs
  3. Enables debugging sessions at the state of the previous session
  4. Setting their own pace without wasting computer time

IBM System/360 Model 67 (Aug'65)

- First IBM system with virtual memory capabilities
  - addition of the Dynamic Address Translation (DAT)
  - 16 MB (24-bit addresses)
  - Supported segmentation and paging
  - Memory virtualization enables multiple applications
  - Very high reliability/availability
  - Single OS, many simultaneous users

DEC PDP-11 (1970)

- 16-bit successor to the PDP-8
- UNIX provided virtualization (strong isolation) of memory, computation, storage
  - Single OS, many users
  - Low-cost alternative to mainframe-based computing, but less capable

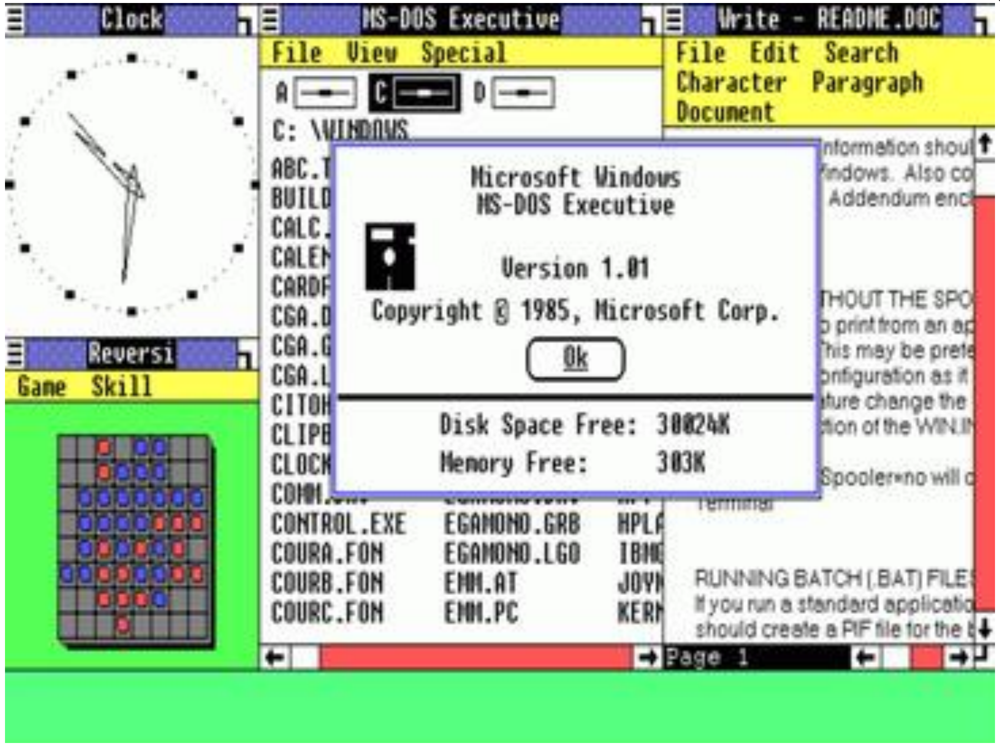
IBM System/370 (Aug'72)

- Virtual memory
  - 2KB or 4KB pages of memory,
  - 64KB or 1MB segment sizes
  - High performance Virtual Address translation
    - using "Translation-Lookaside Buffer" (TLB)
- VM/370 – Virtual Machine Manager (VMM)
  - First instance of hardware assisted virtualization
  - Privileged I/O: special-purpose control instructions, privileged I/O instructions
  - Privileged System Control: privileged system-control instructions
  - Complete virtualization (strong isolation) of memory, computation, storage
- *Multiple simultaneous OSes (VMs)*, 100's-1,000's simultaneous users
  - But very, very expensive platform

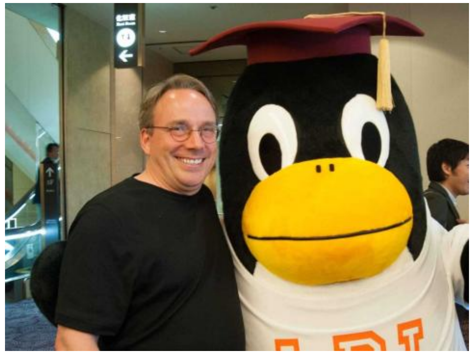
Personal computing era

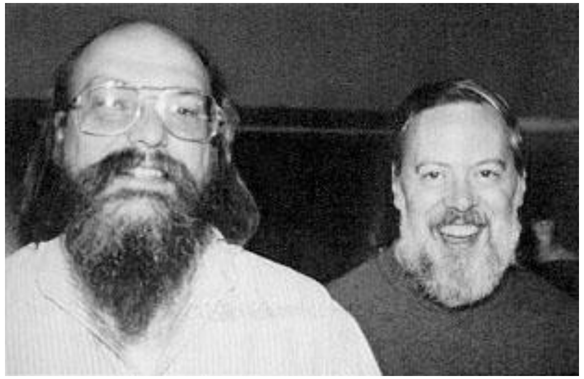
ATARI **micro**computers Gaming + home computing 1979.

Apple Macintosh, 1984



Microsoft Windows 1.0, 1985





Linux began in 1991, Linus Torvalds began a project that later became the Linux kernel. Based on Unix developed by Ken Thompson (left) and Dennis Ritchie (right), creators of the Unix operating system 1969, written in C and assembly language

Intel Pentium Processor, 1993

Internet search and services

Yahoo 1994

Google 1998

Operating System and Computer Network

➔ **Since 1990s**

- Telecommunications companies, provided point-to-point data circuits, Virtual Private Network (VPN).
- Efficient bandwidth.
- Big servers and network infrastructure.
- Scientists and technologists need for large-scale computing through time-sharing.
- Codes to optimize the infrastructure.
- Better OS platform, and applications to prioritize CPUs.
- **Origin of the term *Cloud*** is unclear.
  - ***Cloud*** was used as a metaphor for the ***Internet***.

http://en.wikipedia.org/wiki/Cloud_computing

Grid Computing

- Grid Computing Criteria (Ian Foster 2004)
  - **Coordination:** A grid must coordinate resources that are not subject to centralized control
  - **Open APIs:** A grid must use standard, open, general-purpose protocols and interfaces
  - **QoS:** A grid must deliver nontrivial qualities of service (e.g., relating to response time, throughput, availability, and security) for co-allocating multiple resource types to meet complex user demands
- Promise of ubiquitous grid computing (utility)
  - Reality is specialized grids
    - TeraGrid, Open Science Grid, LHC Grid
  - Grid provides "library level" service customized to HW
    - Ensuring consistent libraries across HW is hard!

Grid Architecture

- *Fabric layer* grids provide access to different resource types such as compute, storage and network resource, code repository, etc.
- *Connectivity layer* defines core communication and authentication protocols for easy and secure network transactions.
- *Resource layer* defines protocols for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations on individual resources.
- *Collective layer* captures interactions across collections of resources, directory services such as MDS (Monitoring and Discovery Service)
- *Application layer* comprises whatever user applications built on top of the above protocols and APIs and operate in VO environments.

Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." *Grid Computing Environments Workshop, 2008. GCE'08*. IEEE, 2008.

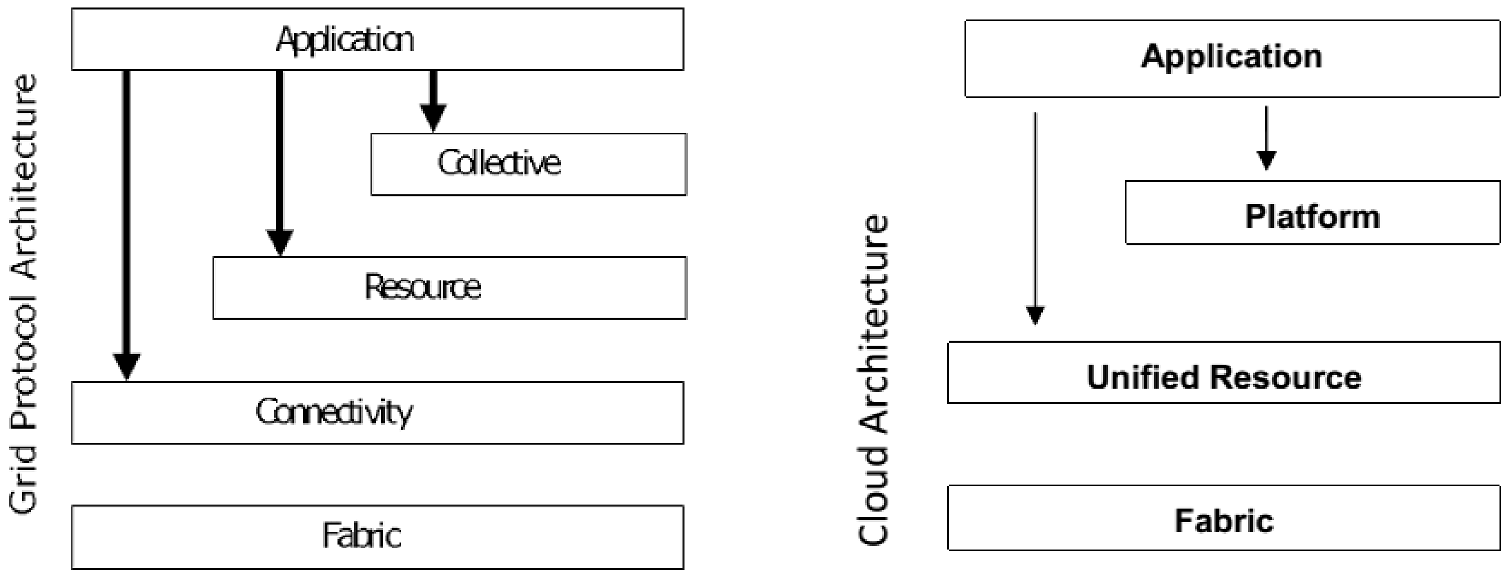
Cloud Architecture

- *Fabric layer* contains the raw hardware level resources, such as compute resources, storage resources, and network resources.
- *Unified resource layer* contains resources that have been abstracted/encapsulated (usually by virtualization) so that they can be exposed to upper layer and end users as integrated resources, for instance, a virtual computer/cluster, a logical file system, a database system, etc.
- *Platform layer* adds on a collection of specialized tools, middleware and services on top of the unified resources to provide a development and/or deployment platform. For instance, a Web hosting environment, a scheduling service, etc.
- *Application layer* contains the applications that would run in the Cloud

Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." *Grid Computing Environments Workshop, 2008. GCE'08*. IEEE, 2008.

Grid and Cloud Architecture

- All the layers are being influenced with the fundamental concepts of *Services* technology.



Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." *Grid Computing Environments Workshop, 2008. GCE'08*. IEEE, 2008.

Cloud Computing vs. Grid Computing

	Grid Computing	Cloud Computing
Platform	Custom node/network HW	Commodity node/network HW
Environment	Library-based and customized to HW, hard to ensure consistent libraries across HW domains	**Virtualized: Exact execution environment can be created and cloned in the cloud, arbitrary apps supported**
Resource allocation	Whole machine unit of allocation	**HW resources can be fractionally allocated, maximizing utilization**
Quality of Service	**Strong CPU and I/O performance guarantees**	Only CPU-based QoS guarantee (some variation)
Capacity	Finite allocation of resources	**“Infinite” resources available**

Cloud Computing History

➔ Since 2000

In early 2008

- Eucalyptus became the first open-source,
- AWS API-compatible platform for deploying private clouds.
- OpenNebula, became the first open-source software for deploying private and hybrid clouds, and for the federation of clouds.
- Quality of service guarantees in cloud-based infrastructures.

In 2006 Amazon.com introduced the Elastic Compute Cloud.

In July 2010, Rackspace Hosting and NASA jointly launched OpenStack.

On March 1, 2011, IBM SmartCloud framework to support Smarter Planet project.

On June 7, 2012, Oracle announced the Oracle Cloud

http://en.wikipedia.org/wiki/Cloud_computing

Cloud Computing Definition

"A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet."

By Foster, Y. Zhau, R. Ioan, and S. Lu. "Cloud Computing and Grid Computing: 360-Degree Compared." *Grid Computing Environments Workshop*, 2008.

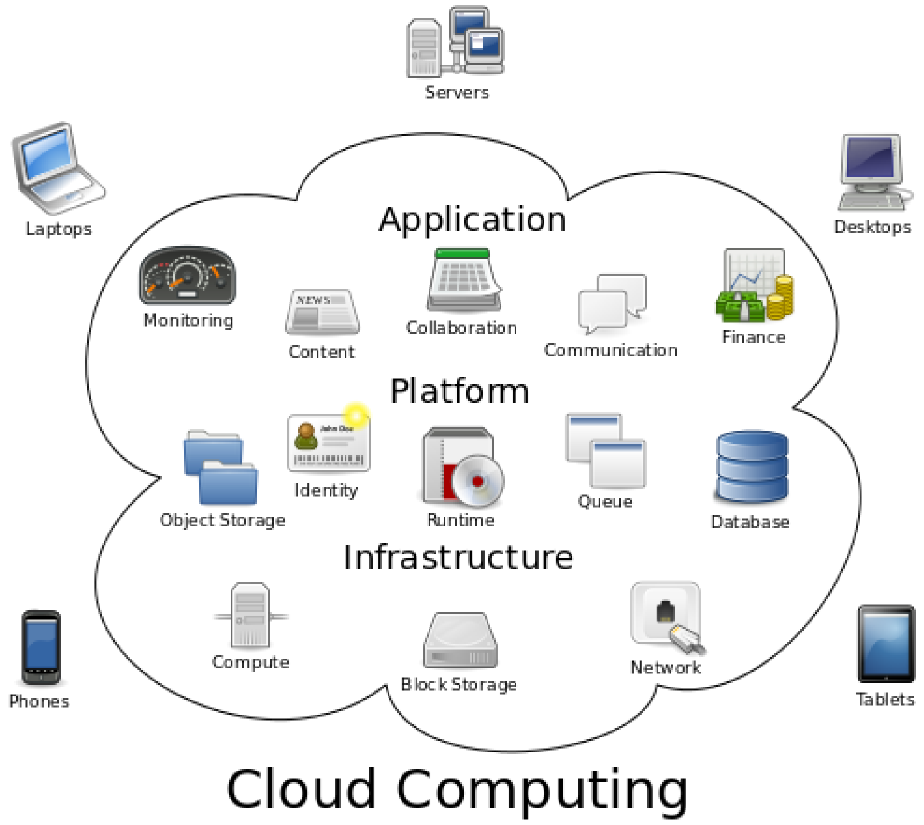
"A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

By *National Institute of Standards and Technology (NIST)*, 2011.

Cloud Computing

- **Cloud computing** (Utility and Grid Computing).
- Clouds are of three types
  - public, private or hybrid.

Creating a groups of scalable hardware and software that allow remote computation, data storage and access using “Services”.



Whether cloud stands on existing technology?

Yes, Hypervisor, Virtualization, Load Balancing, VPN, SAN, SOA, Web Service and many more are the technologies behind Cloud.

Larry Ellison, CEO, Oracle (Wall Street Journal, Sept. 26, 2008) “we have redefined Cloud Computing to include everything that we already do.... change the wording of some of our ads.”

Richard Stallman, Founder, Free Software Foundation (The Guardian, Sept. 29, 2008). *said* “it’s marketing hype. Somebody is saying this is inevitable ….. it’s very likely to be a set of businesses campaigning to make it true.”

Mell, Peter, and Tim Grance. "The NIST definition of cloud computing. "*National Institute of Standards and Technology* 53.6 (2009): 50.

Turing Award 2013

- "Leslie Lamport, a Principal Researcher at Microsoft Research, as the recipient of the 2013 ACM A. M. Turing Award." Lamport's proposed widely used algorithms and tools that have applications in cloud computing.
- His 1978 paper "Time, Clocks, and the Ordering of Events in a Distributed System" is one of the most cited in the history of computer science.
- "Turing Award for imposing clear, well-defined coherence on the seemingly chaotic behavior of distributed computing systems, in which several autonomous computers communicate with each other by passing messages. He devised important algorithms and developed formal modeling and verification protocols that improve the quality of real distributed systems. These contributions have resulted in improved correctness, performance, and reliability of computer systems."

http://www.acm.org/press-room/news-releases/2014/turing-award-13

Turing Award 2013

- "Google Vice President of Research Alfred Spector noted that "with the growing shift to ever-larger scale distributed systems and cloud computing, Lamport's work has taken on a significantly increased role. His results have benefited many research communities including those in parallel and high performance computing systems, concurrent algorithms, and software reliability. And, his work has had implications not just in the theoretical community, but also with the engineers and programmers who design and implement many types of systems."
- Harry Shum, Microsoft executive vice president of Technology and Research: "I really started to appreciate the incredible contribution his work has made to our industry, especially in cloud computing and distributed systems, when I worked at Bing. At Bing, we studied his paper on Paxos and applied his technology to build the distributed systems that we still use today."

http://www.acm.org/press-room/news-releases/2014/turing-award-13
http://research.microsoft.com/en-us/news/features/lamport-031814.aspx

Historical Evolution of Terminologies

Supercomputers

Grid Computing

Utility Computing

- pay-as-you-go computing
- Illusion of infinite resources
- Demanded billing like hourly, daily, number of users

Cloud Computing

(Infrastructure, Hardware, Platform, Software) as a service → "X as a service"

Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)

Requirements of CAPEX and OPEX

Cloud computing is required to meet the requirement of scalability, cost efficiency, legal agreement and business.

CAPEX and OPEX

- CAPEX model (Capital expenditure)
  - buy the dedicated hardware over a period of time.
- OPEX model (Operational expenditure)
  - use a shared cloud infrastructure and pay as one uses it.
- **Moving to cloud:** CAPEX ➔➔➔ OPEX model
- **Results:** Cloud vendors are experiencing high growth rates annually
- Business in profit.

CAPEX to OPEX

- **Benefit of OPEX and why cloud?**
  1. Sharing of resources to achieve coherence and economies of scale.
  2. Avoid upfront infrastructure costs
  3. Focus on code development instead of infrastructure.
  4. Applications can run faster, with improved manageability and less maintenance, and enables IT.
  5. Adjustment with fluctuating and unpredictable business demand.
  6. Cloud providers use "pay as you go" model. Charges may be high if administrators do not used cloud.
  7. Reason for the growth of cloud computing: availability of high-capacity networks, low-cost computers and storage devices, hardware virtualization, service-oriented architecture, utility computing.

Requirements of Cloud Computing

- Cloud computing in computer science is analogous to electricity grid over an electric network.
- Concept of converged infrastructure and shared services.
- Maximizing the effectiveness of the shared resources.
- Resources are shared by multiple users.
- Resources are dynamically reallocated as per demand.
- Efficient use of computing power thus being eco-friendly (less power, air conditioning, rackspace, etc.).
- Multiple users can work on a machine.
- Users can retrieve and update the data with flexible licenses for different applications.

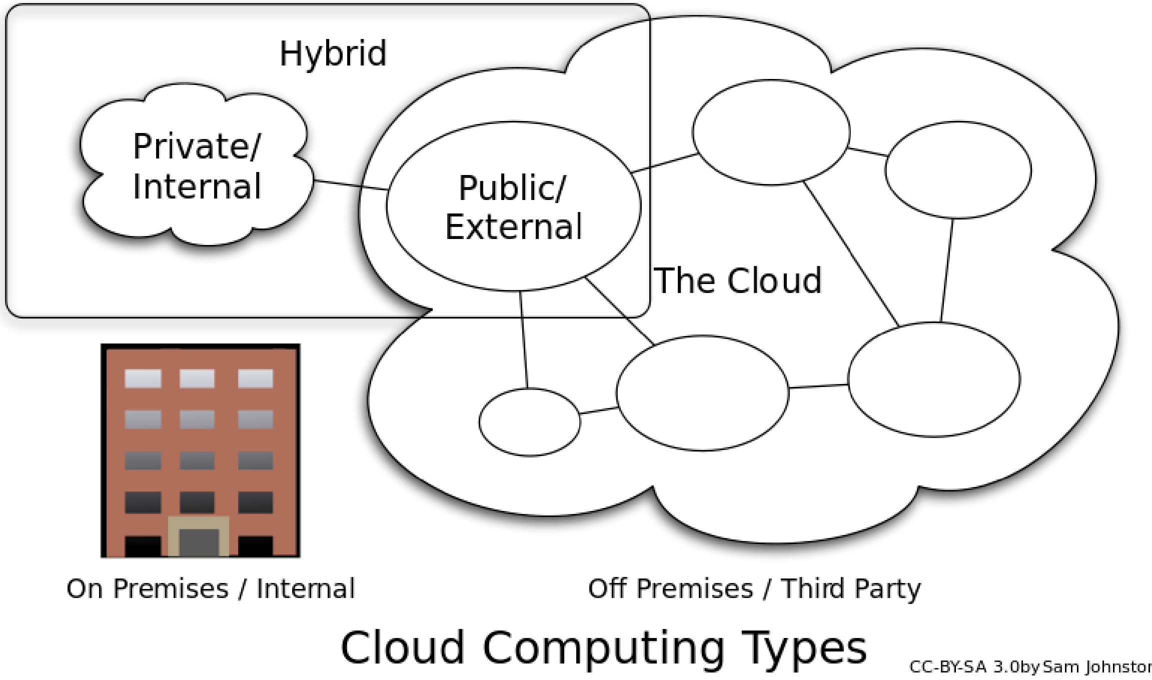
Technologies involved in Cloud

- SaaS
- Inexpensive storage
- Inexpensive and plentiful client CPU bandwidth to support significant client computation
- Sophisticated client algorithms, including HTML, CSS, AJAX, REST
- Client broadband
- SOA (service-oriented architectures)
- Large infrastructure implementations from Google, Yahoo, Amazon, and others that provided real-world, massively scalable, distributed computing
- Commercial virtualization
- Cloud computing is upgraded version kind of grid computing
- Cloud has evolved by addressing the QoS (quality of service) and reliability problems.
- Cloud computing provides the tools and technologies to build data/compute intensive parallel applications.

Creeger, Mache. "Cloud Computing: An Overview." *ACM Queue* 7.5 (2009): 2.

https://sites.google.com/site/animeshchaturvedi07/academic-teaching/cloudcomputing

Private, Public and Hybrid Cloud



Private, Public and Hybrid Cloud

- **Private cloud:** "The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises".
- **Public cloud:** "The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider".
- **Hybrid cloud:** "The cloud infrastructure is a composition of two or more distinct cloud infrastructures (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 for load balancing between clouds)".

The NIST Definition of Cloud Computing

Public cloud

- Public cloud services may be free or offered on a **pay-per-usage** model.
- Mostly similar public and private cloud architecture. But difference is for security, storage, and other resources that are usually better in public cloud.
- Amazon AWS, Microsoft and Google operate the infrastructure at their data center.
- **Pay-as-you-go** model for compute and "Intelligence-as-a-Service."
- Default for training massive AI models (LLMs) due to unrivaled GPU access.
- **Global Hyperscale:** Amazon (AWS), Microsoft (Azure), and Google (GCP) dominate, followed by emerging "NeoClouds".
- **Neoclouds** are specialized AI cloud infrastructure providers of high-performance GPU-as-a-Service (GPUaaS) to meet the massive demand for AI training and inference.

Private cloud

- Private cloud establishment requires effort by an organization to and also intelligence to take decisions regarding the resources.
- It improves business and also raises security issues that must be addressed.
- Private data centers are generally capital intensive.
- Requires space, hardware, environmental controls and electricity.
- Operational expenses as well as **Capital expenditures** for renewable.
- Issues "buy, build, manage and expertise”.

Hybrid cloud

- Hybrid cloud will help in extension of the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service. Ability to connect, managed or dedicated services with both private and public cloud resources.
    1. If an organization have some client data on a private cloud application, but want to interconnect that application to a business intelligence application provided on a public cloud as a software service.
    2. If an organizations use public cloud computing resources to meet temporary capacity needs that can not be met by the private cloud.
- **Cloud bursting:** when the demand for computing capacity increases then an application deployment model in a private cloud or data center will "bursts" to a public cloud. During spike in processing demand requirement private cloud infrastructure that supports average workloads will start using cloud resources from public or private clouds together.
- **Advantage:** payment only for the extra compute resources when they are needed.

Hybrid Cloud

- AWS and Microsoft offers direct connect services.
- **Dedicated Interconnects:** High-speed and dedicated private links like
  - **AWS Direct Connect:** connections between on-premises data center to AWS
  - **Microsoft Azure ExpressRoute:** on-premises infrastructure and Azure datacenters
  - are now essential for low-latency Hybrid-AI workflows, often bypassing the public internet entirely for security.
  - High-bandwidth connectivity essential for large data transfers, real-time analytics, and hybrid cloud environments.
- **Architectural:** Public and Private clouds share "Cloud-Native" architectures (Kubernetes/Containers).
- Difference is **Data Sovereignty** and **Zero-Trust control** in Private clouds versus the infinite **GPU-elasticity** of Public clouds.

Other Deployment models

- **Distributed cloud:** Computing done by distributed set of machines that are running at different locations and they are connected to a single network. Examples BOINC and Folding@Home. Example: voluntarily shared resources.
- **Multicloud:** When multiple cloud vendors provides service together in heterogeneous architecture. This will help in reducing reliance on single vendors, increase flexibility through choice, mitigate against disasters, etc.
- **Intercloud** is an interconnected cloud i.e. "cloud of clouds" similar to Interconnected network (Internet) "network of networks". Mainly inter cloud helps in interoperability between public cloud service providers (as is the case for hybrid- and multi-cloud).
- **Community cloud:** "The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises".

Elasticity and Provisioning

Architecture for Elasticity

- **Vertical Scale-Up**
  - Keep on adding resources to a unit to increase computation power.
  - Process the job to single computation unit with high resources.
- **Horizontal Scale-Out**
  - Keep on adding discrete resources for computation and make them behave as in converged unit.
  - Splitting job on multiple discrete machines, combine the output.
  - Distribute database.
- For HPC second option is better than first. Because Complexity and cost of first option is very high.

Cloud Computing Infrastructure

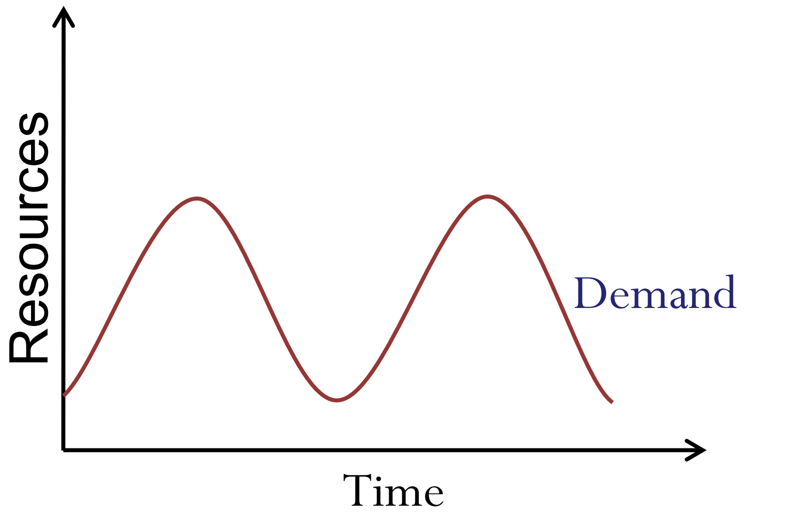
- Scale “out”, not scale “up”. This is due to limitations of
  - Serial Management Protocol (SMP) and
  - Large shared-memory machines
- Move processing to the data
  - Cluster has limited bandwidth
- Process data sequentially, avoid random access
  - Seeks are expensive, disk throughput is reasonable
- Seamless scalability for ordinary programmers
  - From the mythical man-month to the tradable machine-hour

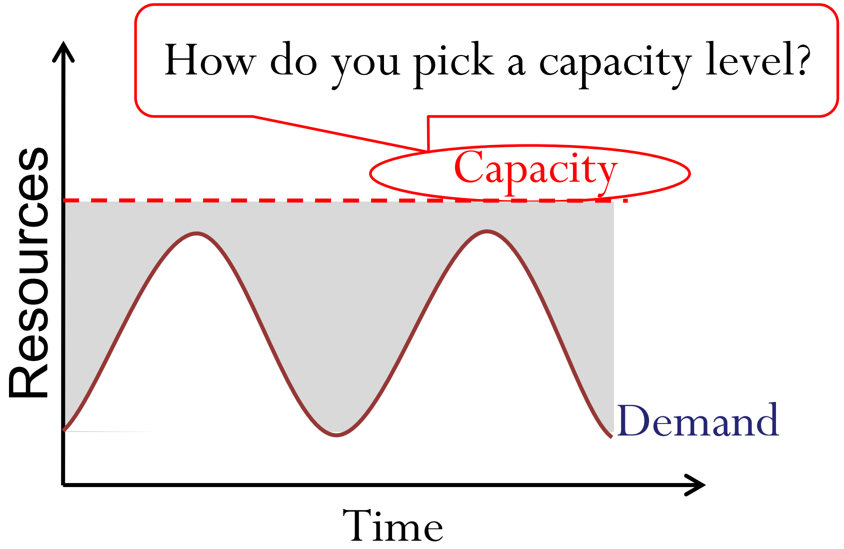
Cloud Computing Computation Models

- Finding the right level of abstraction
  - von Neumann architecture vs cloud environment
- Hide system-level details from the developers
  - No more race conditions, lock contention, etc.
- Separating the *what* from *how*
  - Developer specifies the computation that needs to be performed
  - Execution framework ("runtime") handles actual execution
- Idempotent operations of API (if we make multiple identical requests with the same input parameters and receive the same response every time)
  - Simplifies redo in the presence of failures

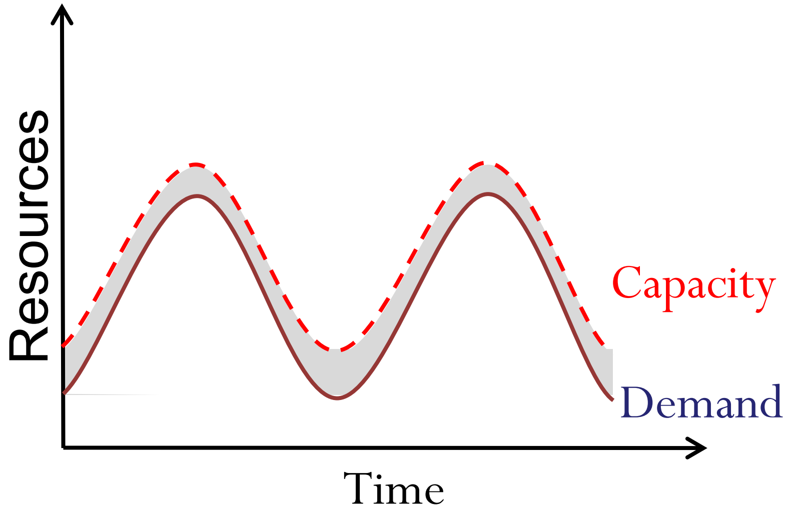
Cloud Application Demand

- Many cloud applications have cyclical demand curves
  - Daily, weekly, monthly, …
- Workload spikes more frequent and significant
- Economics of Cloud Users
  - Pay by use instead of provisioning for peak





Static data center

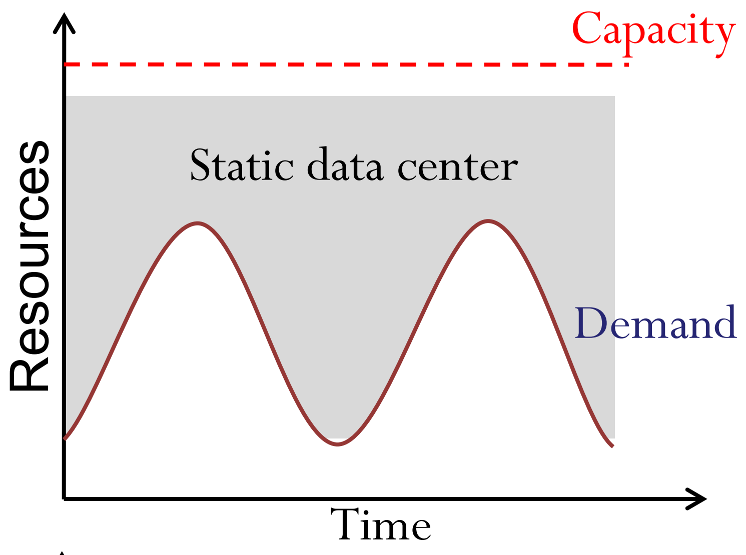


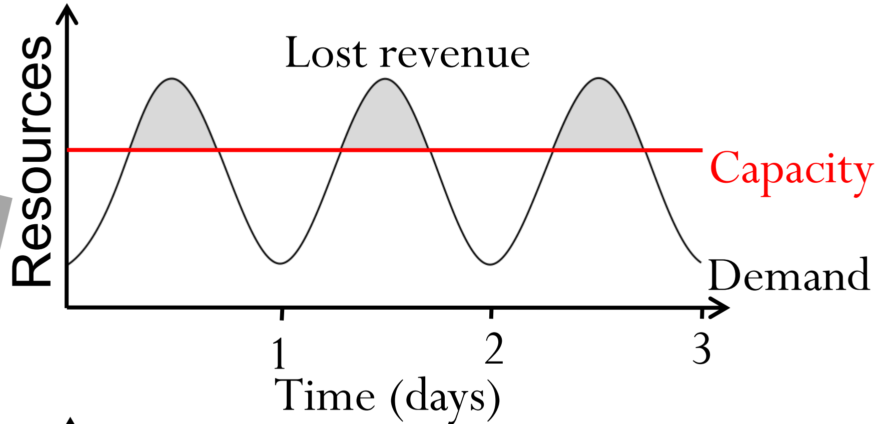
Data center in the cloud

Unused resources

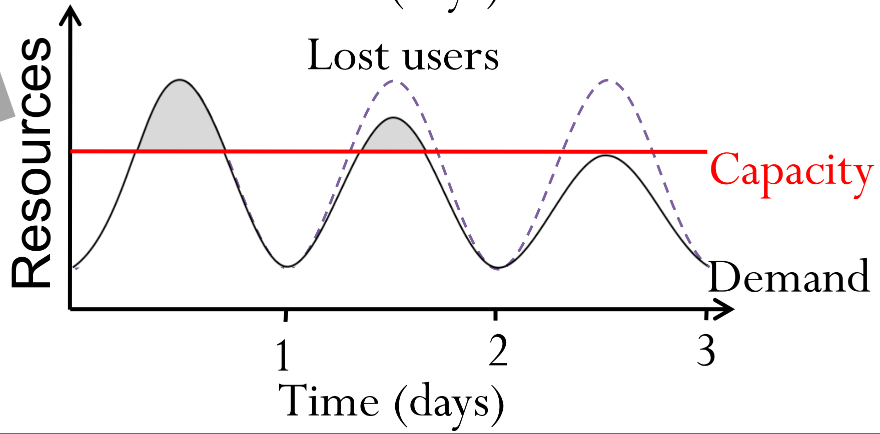
Economics of Cloud Users

- Risk of over-provisioning: underutilization
- Huge sunk cost in infrastructure
- Heavy penalty for under-provisioning





Risk of underutilization if peak predictions are too optimistic
– Wasted CapEx
Hard to provision for spiky workloads



Unused resources

Resource Management & Provisioning

Public cloud services can be used with three Cloud provisioning:

- **Consumer self-provisioning:** Consumer contract and pay as per usage for cloud services directly to provider,
  - e.g. Institute Google or Microsoft domain (email, form, docx, excel etc.).
- **Advanced provisioning:** Consumer contract and pay in advance for resources and services
  - e.g. online event management system
- **Dynamic provisioning:** Provider allocates resources as per consumer usage, then de-provisioning when resources are not in use.
  - Consumer pays as per usage
- **Provisioning and orchestration:**
  - create, modify, and delete resources
  - orchestrate workflows and management of workloads

https://en.wikipedia.org/wiki/Cloud_management

Elasticity and Resource Provisioning

Elasticity: provisioning and de-provisioning resources in an autonomic manner,

For Elasticity **Avoid**

- Over-provisioning: allocating more resources than required,
  - Issue: pay for the useless resources
- Under-provisioning: allocating fewer resources than required
  - Issue: poor service performance, e.g. slow or unreachable
  - Issue: loses customers

https://en.wikipedia.org/wiki/Elasticity_(cloud_computing)

Data Center
(Hardware & software Infrastructure)

https://sites.google.com/site/animeshchaturvedi07/academic-teaching/cloudcomputing

Scale of Cloud in industries?

10s of CPUs?

100s of CPUs?

Scale of Cloud in industries?

- Example: Large scale log processing
  - Clusters vary from 1 to thousands of nodes.
  - Thousands of CPUs and computers

- Cluster has hundreds to thousands of nodes
  - Each node
    - A×B CPU boxes
    - A×B TB disk
    - X GB RAM

"Cabinet Asile" by Robert.Harker - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons

Hyperscale Industry Metrics

- **Massive Cluster Growth:** Modern AI workloads now run on clusters exceeding **65,000 to 130,000 nodes** in leading GCP/Azure environments.

- **Compute Density:** Individual nodes have evolved significantly:
  - **CPUs:** 64 -128+ cores per socket.
  - **RAM:** 1TB - 2TB+ per node is common for high-memory AI inference.
  - **Accelerators:** 8x NVIDIA H100/H200 or Microsoft Maia GPUs per rack.

- **Storage:** Shift from 1TB disks to high-speed **NVMe SSDs (Non-volatile Memory Express Solid State Drives)** (4TB–15TB per drive) to feed data to GPUs at wire speed.

Data Centers

- A **data center** is a facility used to house computer systems and associated components, such as telecommunications and storage systems.
- Includes backup power supplies, redundant data communications connections, environmental controls (e.g., air conditioning, fire suppression) and various security devices.
- Large data centers are industrial scale operations using as much electricity as a small town.

Data Centers

- **The Power Shift:** Modern data centers are **Industrial-Scale Gigawatt (GW) Campuses**. By 2026, global consumption is expected to exceed **1,000 TWh**, equivalent to the annual energy use of Japan.
- **Sustainable Infrastructure:** Focus on **Liquid Cooling** (direct-to-chip or immersion) to handle high-density AI racks (60kW–100kW+ per rack).
- **Energy Mix:** Leading providers now use **Small Modular Reactors (SMRs)** or hydrogen fuel cells to ensure 24/7 carbon-free power.
- **Advanced Controls:** AI-driven "Digital Twins" manage heat flow and workload distribution in real-time.

Virtualization & Services

- Cloud computing is a combination of existing technologies. User can use this combination without the technical knowledge or expertise.
- **Virtualization:** It is a process of creating a illusion of a physical computing device, into one or more low-configuration virtual device, such that they can easily be used and managed.
- **Service-oriented Architecture (SOA):** Cloud computing adopts concepts of SOA that to create services for sub-problems and these services can be integrated to solution the problem. Moreover, these services are further divided into individual operations (procedures).
- Cloud computing provides its resources in the form of services. Use the well-established technology domains like SOA and Virtualization. This results in globalized and user-friendly technology.

Other relevant technologies

- Mainframe computer — Powerful computers used for critical applications for bulk data processing.
- Peer-to-peer — Distributed Architecture without central coordination. Participants are both suppliers and consumers of resources (in contrast to the traditional client–server model).
- Grid computing — computer are composed into a cluster using networked, cluster is acting as converged resource.
- Utility computing —The package of computing resources (CPUand storage) are charged according to the pay-as-you-go.
- Service oriented Computing: This organize and utilize the distributed services offered by different owners. It gives a formal way of offering, discover, interaction with the flexibility to orchestrate according to requirement.

Can we practice it in our lab?

Yes, we can practice with few machines.

- Containerized Deployment (Using Docker for more portable, scalable apps).
- *Hadoop*, *Storm, Spark* etc.
- Services Oriented Computing
- Cloud, Grid, and Distributed Clusters

Lab Practice

- **Cloud-Native Engines:** Set up small-scale clusters with *Eucalyptus*, *OpenStack*, *OpenNebula*, and *Kubernetes (K8s)* for local cloud environments.
- **Application Deployment:**
  - Deploy **Serverless functions** (**AWS Lambda**, **Google Cloud Functions**).
  - Build **AI-Agents** using Google Cloud or Azure Container Apps.
- **Big Data & AI Tools:** Practice distributed processing with *Hadoop* (*MapReduce*), *Ray*, Apache *Spark*, or *Kafka* for batch and real-time streaming data.
- Move beyond "servers" to Microservices and Serverless architectures.
- Move toward **Service-Oriented** and **Edge Computing** - practicing how to deploy containers from a local lab to a public cloud seamlessly (Hybrid Cloud).

Lab Practice

- **Google Ecosystem Practice:**
  - Develop modern web apps with Angular or Flutter.
  - Deploy containerized microservices (Docker) to Google Cloud Run or Firebase App Hosting.
  - Integrate Generative AI via the Vertex AI Gemini API.
- ***Next.js* or *React*** (Modern frameworks with built-in server-side rendering).
- **Ruby on Rails** as a web framework and **Engine Yard** as a cloud hosting platform for Rails applications.

Web Services

- **SOAP/WSDL Web Service:** Interoperable machine-to-machine interaction over a network with WSDL using SOAP messages, conveyed using HTTP with XML standards.
  - **WSDL:** Description of a web service in XML
  - **SOAP:** Packet of messages in XML

- **RESTful Web services** allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations.
  - Representational State Transfer (REST)
  - Stateless protocol and standard operations,
  - Rest API: The operations HTTP methods available are GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS and TRACE.

Hypervisors

Virtualization: A virtual machine (VM) is a software implementation of a machine (i.e., a computer) that executes programs like a physical machine.

- **Hypervisor:** provides a uniform abstraction of the underlying physical machine.

- **Hypervisor Types**
  - Type 1
    - **Bare Metal:** Runs directly on Hardware. On top Hypervisor the OS is loaded. Example Mircrosoft Hyper V, Citrix Xen Server,
  - Type2
    - **Hosted:** The Hypervisor runs on a guest OS. Example KVM.

Big Data on Distributed Computing Systems

- Batch processing of big data
  - Google File System (GFS) (Map-Reduce)
  - Hadoop Distributed File System (HDFS) (Map-Reduce)
  - Apache Spark
- Real time Stream processing
  - Storm (Topology is a graph of Spouts and Bolts)
  - Apache Spark
- Apache Spark
  - Resilient Distributed Datasets (RDD), Directed Acyclic Graph (DAG)
  - Libraries: SQL, DataFrames, MLlib for machine learning, GraphX, and Spark Streaming
  - Access data in OS file system, HDFS, etc.

5V's and Societies of Big Data

5V’s of Big Data

- Big data can be described by the following characteristics:
  - Volume: size large than terabytes and petabytes
  - Variety: type and nature, structured, semi-structured or unstructured
  - Velocity: speed of generation and processing to meet the demands
  - Veracity: the data quality and the data value
  - Value: Useful or not useful
- The main components and ecosystem of Big Data
  - Data Analytics: data mining, machine learning and natural language processing
  - Technologies: Business Intelligence, Cloud computing & Databases
  - Visualization: Charts, Graphs etc.

Big Data Societies & Communities

- **IEEE Big Data Technical Community:** A multidisciplinary group focused on solving large-scale data challenges through a global network of professionals in industry, academia, and government. It hosts the annual **IEEE International Conference on Big Data** (IEEE BigData), which has been a top-tier research venue since 2013.
- **ACM SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining):** The premier professional society for data science and data mining. It oversees the KDD Conference, manages the ACM Transactions on Knowledge Discovery from Data (TKDD), and promotes ethical data use.
- **IEEE Communications Society (ComSoc) Technical Committee on Big Data (TCBD):** Focuses on the intersection of big data and communications technology, including 5G/6G, IoT, and cloud computing. Membership is free and open to non-IEEE members.
- **International Society of Big Data and Bioinformatics (ISBDB):** Dedicated to promoting excellence in big data research to benefit humanity, particularly in healthcare and medicine.
- **IEEE Technical Community on Data Engineering (TCDE):** Connects researchers working on data-intensive systems, analytics, and scalability. It sponsors the International Conference on Data Engineering (ICDE).

Big Data Societies & Communities

- **Very Large Data Base Endowment Inc. (VLDB Endowment)** promoting and exchanging scholarly work in databases. VLDB Conference
- UN Committee of Experts on Big Data and Data Science for Official Statistics: A global body investigating how big data can be used for monitoring Sustainable Development Goals (SDGs) and informing public policy.

Key Activities of These Societies

- Conferences: Organizing global forums like IEEE BigData, ACM KDD, and ICDE.
- Publications: Managing high-impact journals and newsletters such as SIGKDD Explorations and the IEEE Data Engineering Bulletin.
- Standardization: Developing terminology and methodological standards for the analytics process.
- Professional Growth: Providing awards, distinguished lecturer programs, and networking for young researchers.

References

1. Fox Armando, Rean Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica. "Above the clouds: A Berkeley view of cloud computing." Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28 (2009): 13.
2. Creeger, Mache. "Cloud Computing: An Overview." ACM Queue 7.5 (2009): 2.
3. http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211
4. http://en.wikipedia.org/wiki/Web_Services_Description_Language,
5. http://en.wikipedia.org/wiki/Cloud_computing
6. http://aws.amazon.com/what-is-cloud-computing/
7. http://en.wikipedia.org/wiki/Data_center
8. https://www.microsoft.com/en-us/research/blog/leslie-lamport-receives-turing-award/
9. https://en.wikipedia.org/wiki/Hypervisor

תודה רבה
Hebrew

Ευχαριστώ
Greek

Danke
German

Спасибо
Russian

Merci
French

ধন্যবাদ
Bangla

धन्यवादः
Sanskrit

நன்றி
Tamil

شكرا
Arabic

ಧನ್ಯವಾದಗಳು
Kannada

Thank You
English

നന്ദി
Malayalam

Grazie
Italian

多謝
Traditional Chinese

ధన్యవాదాలు
Telugu

આભાર
Gujarati

Gracias
Spanish

ਧੰਨਵਾਦ
Punjabi

धन्यवाद
Hindi & Marathi

多谢
Simplified Chinese

https://sites.google.com/site/animeshchaturvedi07

Obrigado
Portuguese

ありがとうございました
Japanese

ขอบคุณ
Thai

감사합니다
Korean