Lecture 01
Introduction

Dr. Ehtesham Zahoor

About me

2008-2011	**Doctorant en Informatique (Ph.D Computer Science)** Université de Lorraine, INRIA/LORIA, France
2007-2008	**Master en Informatique (MS Computer Science)** Université Henri Poincaré, Nancy 1, France
2002-2004	**MS (Computer Science)** National University of Computer and Emerging Sciences, NUCES, Islamabad, Pakistan.

2011-2012	Assistant Professor (Attaché Temporaire d'Enseignement et de Recherche (ATER)) at Université de Lorraine, France.
2005-2007	Software Engineer at Knowledge Platform, Pakistan http://www.knowledgeplatform.com/. Job responsibilities included development of Web and e-Learning applications.
2004-2005	Software Developer at Jin Technologies, Pakistan http://www.jintech.com/. Job responsibilities included development of various web applications for leading insurance companies.

About me

- HEC Pakistan and French Government grant for pursuing MS leading to Ph.D program from France (2007-2011)
- Winner of the Pakistan Software Export Board (PSEB) Incubator Program grant for the Project ECO-Ride (2012-2013)

- Invited committee member for the Ph.D Thesis defense of Mr. Ahmed Bouchami, Lorraine University, France.
- Invited researcher for OpenPaas::NG: A new-generation collaboration platform at INRIA France.
- Founding member and Chief Scientist of Secure Networks and Distributed Systems (SENDS) lab at FAST NUCES, 2016.

Research contributions

- Focus has been on high quality conference publications.
- Multiple high-quality research publications in A ranked conferences, such as IEEE ICWS, WISE, CAiSE and others.
- Recent work includes addressing security aspects related to large scale distributed systems including major Cloud providers and blockchain frameworks.

Teaching experience

- Undergraduate Courses:
  - Computer Programming, Data Structures, Operating Systems, Computer Networks, Parallel and Distributed Computing, Introduction to Blockchain and Cryptocurrency
- Graduate Courses:
  - Advance Operating Systems, Cloud Computing, Blockchain and its Applications, Distributed Data Engineering

About you?

- You must have already studied the Operating Systems course
  - You need to have strong grasp on core operating systems concepts such as OS structure, processes/threads, memory management and others
  - I will provide you some revision material but it is important to cover all basics as early as possible

Some Rules

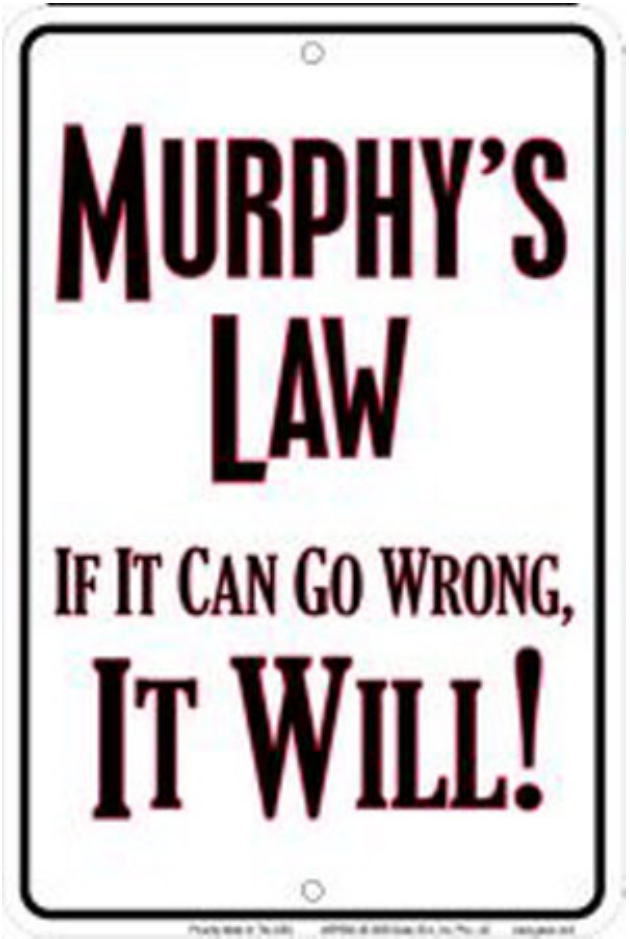
- *Raise your hand before asking any question and then WAIT for the permission*
- *Never ever miss a class*
- *Never ever "sleep" in the class*
- *Never use mobile phones in the class*

- Above all, whatever you do, please do not disturb others

Policy about missed assessment items

- Retake of missed assessment items (other than midterm/ final exam) will not be held.
- One minute (second) late on the deadline is LATE. It is YOUR responsibility to respect the deadlines.

Policy about missed assessment items



Dishonesty, Plagiarism

- Plagiarism in an assignment will result in zero marks in the whole assignments category.

- Plagiarism in the course project will result in zero marks in the project and also the deduction of -75% of the total marks for the assessment from other evaluations. For instance, plagiarism in the course project having 10 absolutes would result in 0 points for the project and -7.5 absolutes would be deducted from scores achieved in other assessment items.

- Plagiarism in the midterm and the final exam would result in a disciplinary case forwarded to the department disciplinary committee.

Dishonesty, Plagiarism

You can fool some of the people all of the time,
and all of the people some of the time,

but you can not fool all of the people all of the time.

Abraham Lincoln,
16th president of US (1809 - 1865)

Some Bonuses

- Class participation - 1 absolute for each correctly answered bonus question

Tentative Evaluation Breakdown

Assignments	15
Project/Presentations	15
Mid-exam	25
Final	45
Total	100

Tentative Course outline

- **PartA – Introduction and Background**
  - Introduction, some overview of basic concepts including processes, distributed systems, Cloud computing and virtualization. …
- **PartB – Process, Threads, IPC in distributed systems**
  - Traditional process and thread management including case studies on processes/thread creation in Unix and IPC using pipes. Introduction to distributed systems, IPC using MPI and network programming.
- **PartC – Memory management in virtualized systems**
  - Memory management in traditional systems, multi-level paging, memory management in virtualized systems, shadow page tables and EPT MMU, VM live migration approaches

Tentative Course outline

- **PartD – Synchronization/consensus in distributed systems**
  - Overview of basic concepts, clock synchronization, consistency and fault models, RAFT
- **PartE – Distributed File Systems - HDFS**
  - File system management by the traditional and distributed operating systems. Hadoop HDFS/MapReduce

Reference Books:

- Operating System Concepts, 10th Editon, Silberschatz Galvin Gagne
- Modern Operating Systems, Andrew S. Tanenbaum
- Operating Systems, William Stallings

AOS on Google Classroom

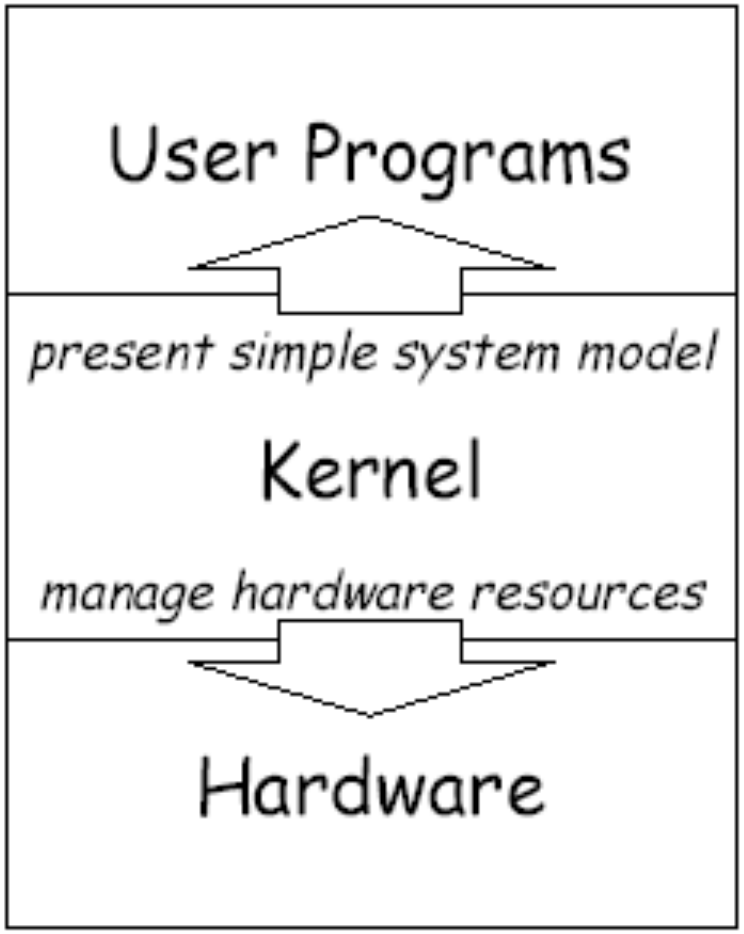
tnl6dwu

What is an Operating System?

- It is a program!
- Top-down view
  - Provides an abstraction to user programs
    - Easier to program than the underlying hardware.
    - All services are invoked and accomplished through system calls.
- Bottom-up view
  - Acts as a resource manager of a complex system
  - Resources consist of processors, memories, timers, disks, keyboard, network interfaces, printers etc.
  - OS manages allocation of these resources to user programs in an orderly and controlled manner

Two Goals of Operating Systems

- Manage hardware resources
  - System operates smoothly: efficiently, reliably, securely
- Present abstract system model to programmer
  - Simple and convenient access to and control of resources



A Question

- The operating system
  - Gets an input
  - Performs a computation
  - Produces an output
  - And Quits
  - Yes or no?

The answer: No

- The operating system is a *Reactive Program*

What is an Operating System?

- OS do nothing by themselves.
- Similar to subroutine libraries, do nothing unless they are invoked by programs
- Act as an intermediary between users and the hardware

Operating System

- Modern Operating Systems are ***Interrupt driven***
- If
  - No process to execute
  - No I/O device to service
  - No user to whom to respond
- Then
- OS will sit quietly, waiting for something to happen
- This something is ***Interrupt***

Interrupts

- There are three (or broadly two) types of interrupts:
- **Hardware Interrupts** - generated by hardware devices to signal that they need some attention from the OS.
  - They may have just received some data (e.g., keystrokes on the keyboard or a data on the ethernet card);
  - or they have just completed a task which the operating system previous requested, such as transferring data between the hard drive and memory.

Interrupts

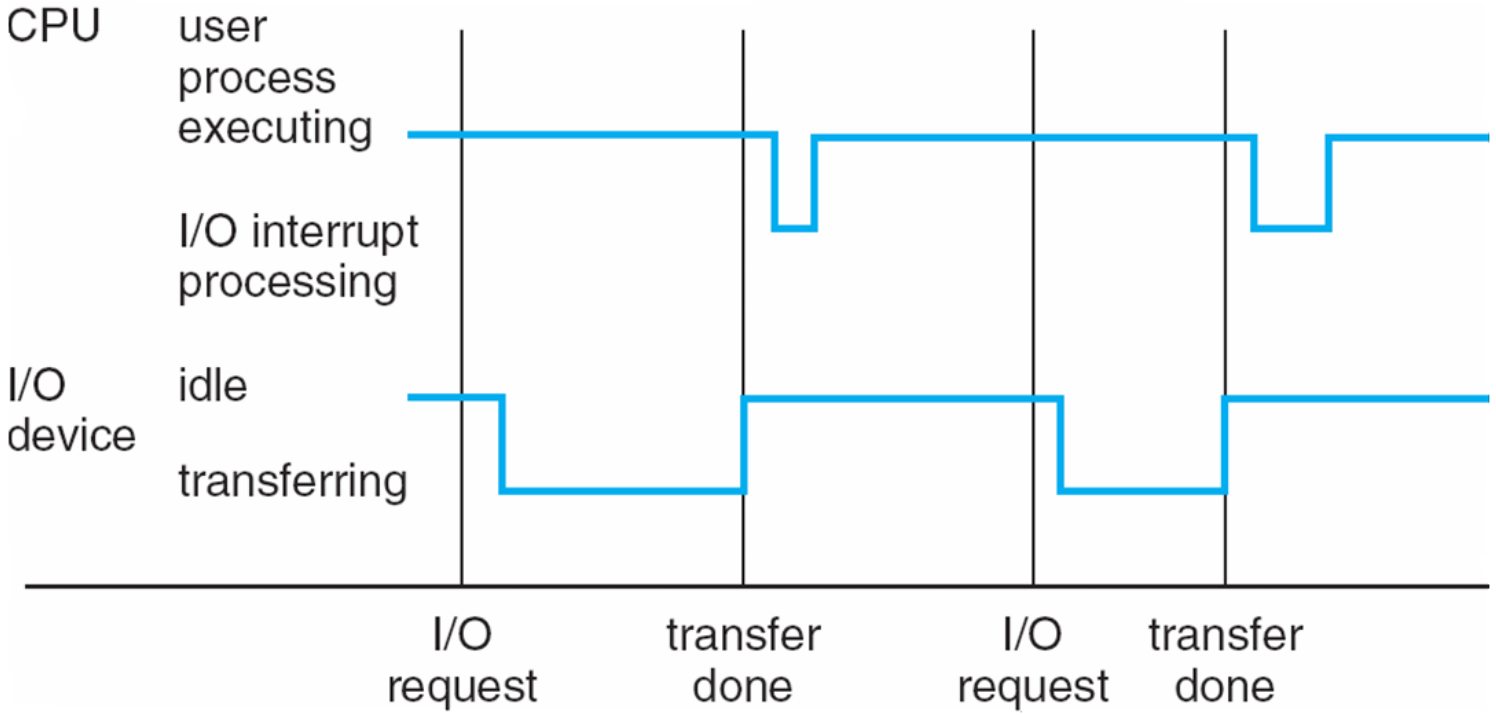
- **Software Interupts** are generated by programs when they want to request a system call to be performed by the operating system.

- **Traps** are generated by the CPU itself to indicate that some error or condition occured for which assistance from the operating system is needed.

Interrupt Handling

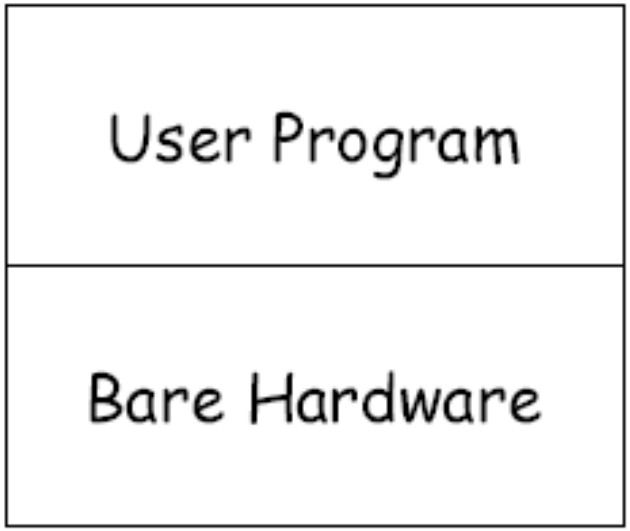
- Different routines handle different type of interrupts.
- Called ***Interrupt Service Routine (ISR)***
- When the CPU is interrupted it stops what it is doing
- The address of the interrupted instruction is saved
- After the interrupt is serviced, the saved address is loaded to the Program Counter
- The Interrupted computation can resume as though the interrupt had not occurred.

Interrupt Timeline



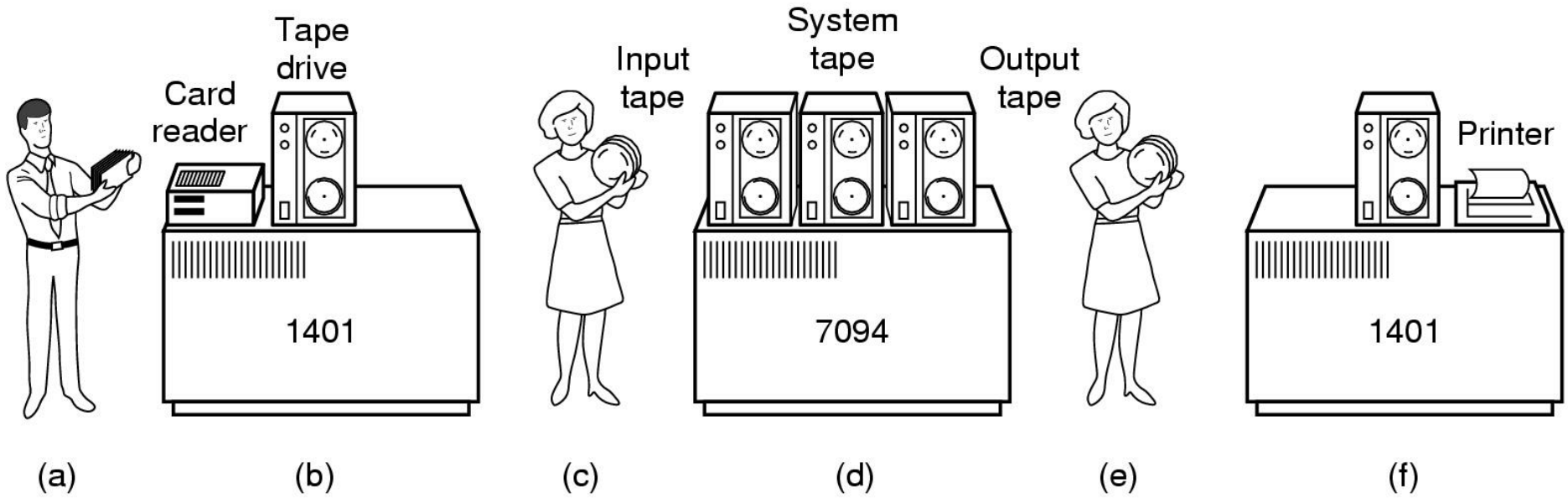
What If No Operating System?

- All we have is bare hardware
- You want to run a program
  - How do you load it?
  - How do you run it?
  - What happens when it completes?
- Need at least some minimal OS to do these functions



…Lets go back to the “stone age” of computing…

Early batch system



- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

History of Operating Systems

- Guided by the history of computer systems in general.
- By the late 1960's operating systems designers were able to develop the system of multiprogramming
- Minicomputers lead the way to Personal computers and the birth of Microsoft and the Windows operating system.

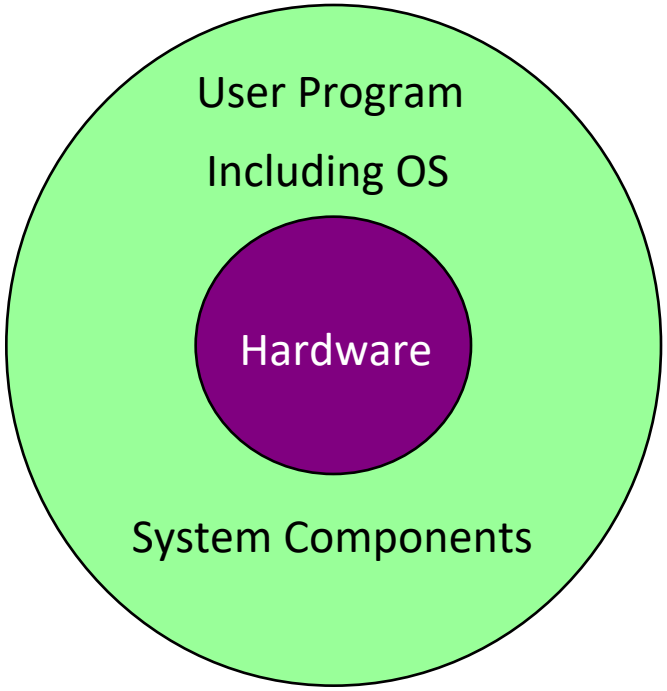
- These days, operating systems come in different falvors

Real-time operating systems

- When correctness of results depend on content and time
- Hard or Soft: indicates how forgiving the system is
  - Nuclear reactor control, Flight control
  - Basically any safety critical system

Embedded Systems

- Embedded system: is a computer system that performs a limited set of specific functions.



Distributed Systems

- *"A distributed system consists of a collection of autonomous computers, connected through a network and distribution middleware, which enables computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. "*

In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox.

We shouldn't be trying for bigger computers, but for more systems of computers.

--Grace Hopper

- Yes, but why dont we grow a large Ox?

Distributed Systems

- Huge amounts of computing are now distributed...
- A few years ago, Intel threw its hands up in the air: couldn't increase GHz much more without CPU temperatures reaching solar levels

- But we can still stuff more transistors
  - Result: Multi-core and GPUs.
  - Your computer has become a parallel/distributed system. In a decade, it may have 128 cores.

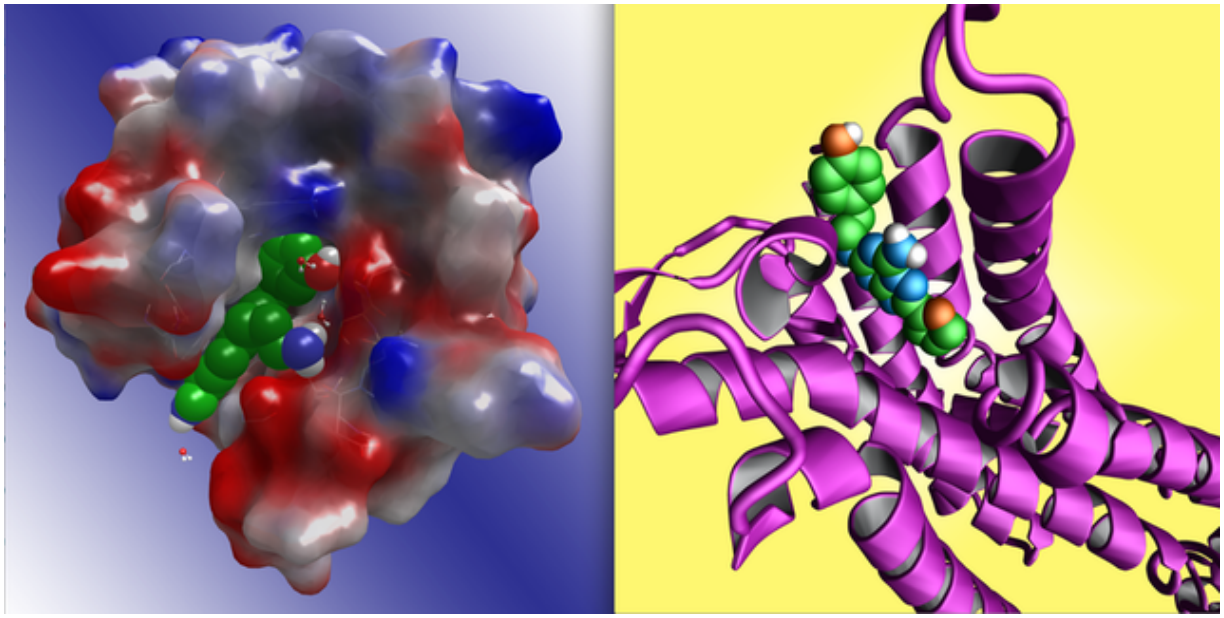
Lecture 01: Introduction

Cloud Computing

- Definition from ***NIST*** ***(National Institute of Standards and Technology)***
  - 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.

Extreme scaling

- $4,829-per-hour supercomputer built on Amazon cloud to fuel cancer research



Cutting corners

- Cancer research is one of the last areas of human endeavor in which anyone would want to "cut corners."

- Yet, that's exactly what scientists must often do when the computational resources available to them aren't sufficient to handle all the components of experiments they want to run.

Cutting corners

- "We cut all kinds of corners. We use less accurate scoring functions. We do less sampling of the conformations of a compound. We are always having to make decisions like that,"
  - Ramy Farid, president of a firm called Schrödinger

- The company operates a 1,500-core cluster to perform research, but it’s often not enough.

A 50,000-core supercomputer

- Recently, they decided to avoid cutting the corners.

- The key was that instead of using Schrödinger's internal cluster, they opted to build a 50,000-core supercomputer on the Amazon Elastic Compute Cloud.

A 50,000-core supercomputer

- It ran for three hours on the night of March 30, 2012
  - at a cost of $4,828.85 per hour.
  - Getting up to 51,132 cores required spinning up 6,742 Amazon EC2 instances running CentOS Linux.
  - This virtual supercomputer spanned the globe, tapping data centers in four continents and every available Amazon region, from Tokyo, Singapore, and Sao Paolo, to Ireland, Virginia, Oregon, and California.

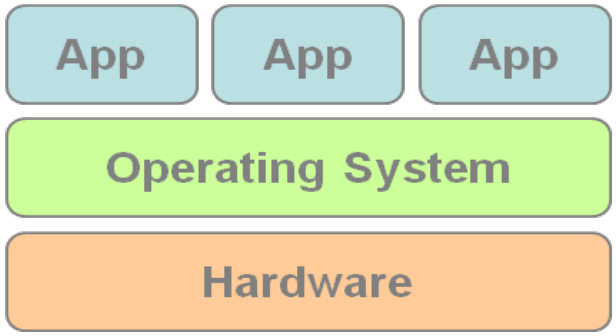
- As impressive as it sounds, such a cluster can be spun up by anyone with the proper expertise, without talking to a single employee of Amazon.

A 50,000-core supercomputer

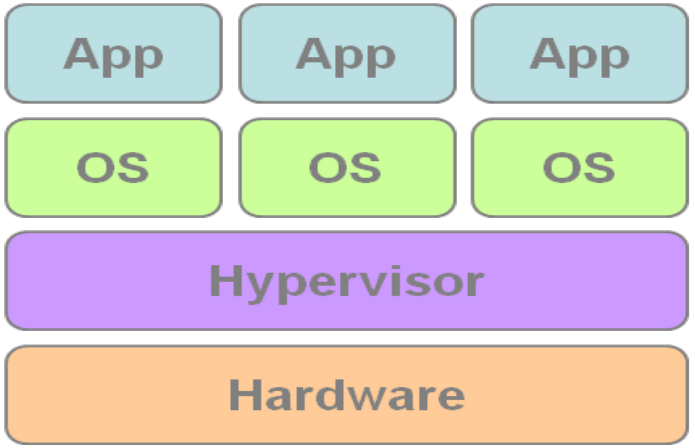
- The 51,132-core cluster didn’t start that large—
  - it scaled up steadily, hitting its peak somewhere in the third hour.
- The cluster used 58.78TB of RAM, and was secured with HTTPS, SSH, and 256-bit AES encryption.
- The cluster used a mix of 10 Gigabit Ethernet and 1 Gigabit Ethernet interconnects.
- However, the workload was what’s often known as "embarrassingly parallel," meaning that the calculations are independent of each other. As such, the speed of the interconnect didn’t really matter.

Cloud Computing

- Key enabler technology: Virtualization



Traditional Stack



Virtualized Stack

Why hypervisor ?

- What is an operating system?
- An operating system is sometimes referred to as a supervisor, so the virtualization software acts as a supervisor of the supervisors and is therefore dubbed a hypervisor;

Virtualization

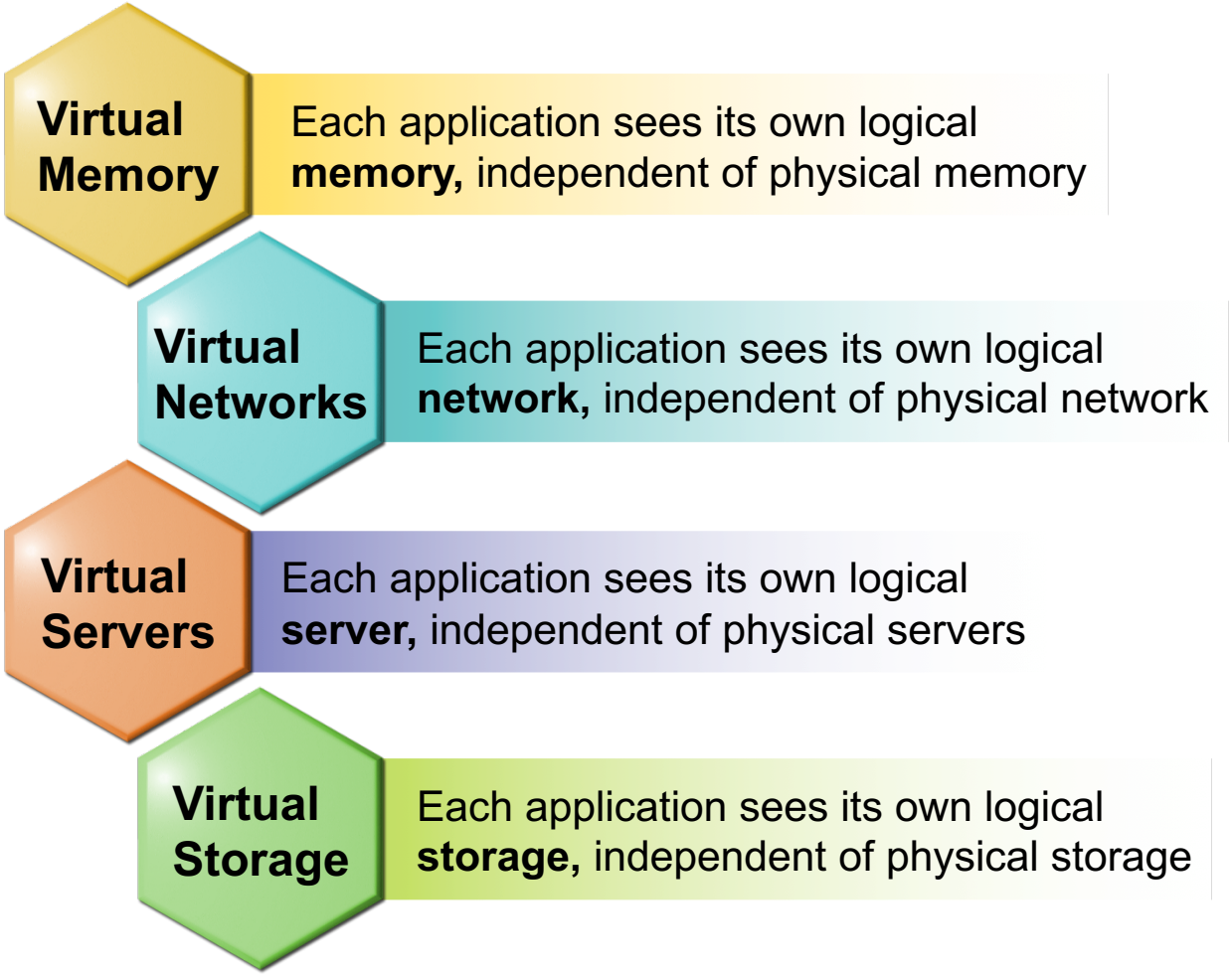
- In a general sense, virtualization, is the creation of a virtual, rather than an actual, version of something.
  - For example, you can take a virtual tour of the White House by going to http://www.whitehouse.gov/about/inside-white-house/interactive-tour
  - In other words, you can take a tour of the White House without actually going to the White House and taking the tour.

Virtualization

- From a computing perspective, you might have already done some virtualization if you've ever partitioned a hard disk drive into more than one "virtual" drive.

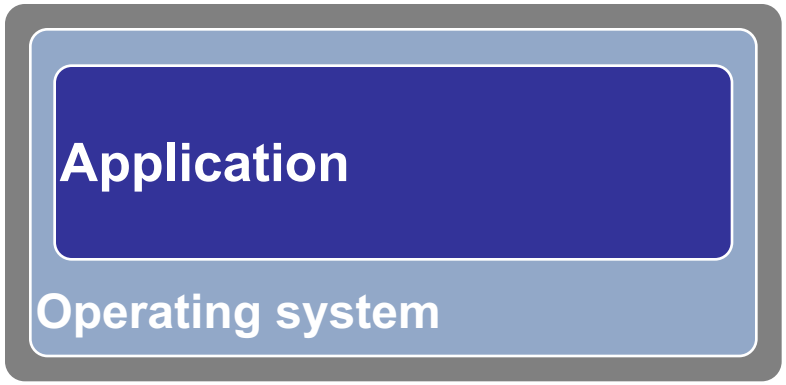
- What other virtualized resources you have already used?

Virtualization Comes in Many Forms



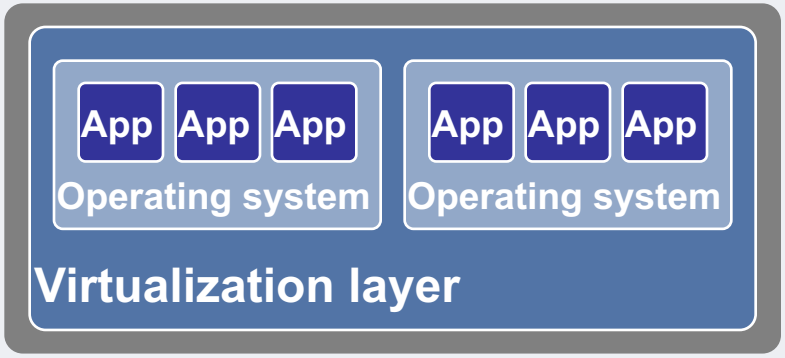
Server Virtualization

Before Server Virtualization:



- Single operating system image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources

After Server Virtualization:



- Virtual Machines (VMs) break dependencies between operating system and hardware
- Manage operating system and application as single unit by encapsulating them into VMs
- Strong fault and security isolation

Quote of the day

- *He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.*

Chinese proverb