OPERATING SYSTEMS

Kernel Data Structures and Computing Environments

Chandravva Hebbi
Department of Computer Science

OPERATING SYSTEMS

Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

OPERATING SYSTEMS

Kernel Data Structures

Chandravva Hebbi
Department of Computer Science

Kernel Data Structures

Array:

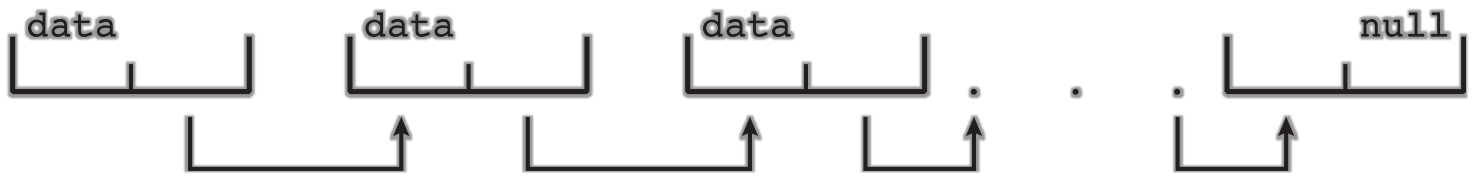
- An array is a simple data structure in which each element can be accessed directly.
- Main Memory constructed with array.
- How the data is accessed?
- Items with multiple bytes are accessed as item number × item size
- But what about storing an item whose size may vary?
- what about removing an item if the relative positions of the remaining items must be preserved?

Lists

- Standard programming data structures are used extensively in OS

Singly linked list

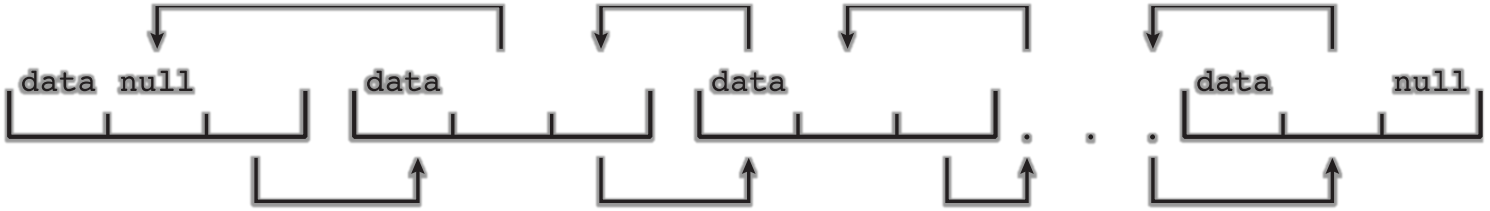
- The items in a list must be accessed in a particular order.
- common method for implementing this structure is a linked list



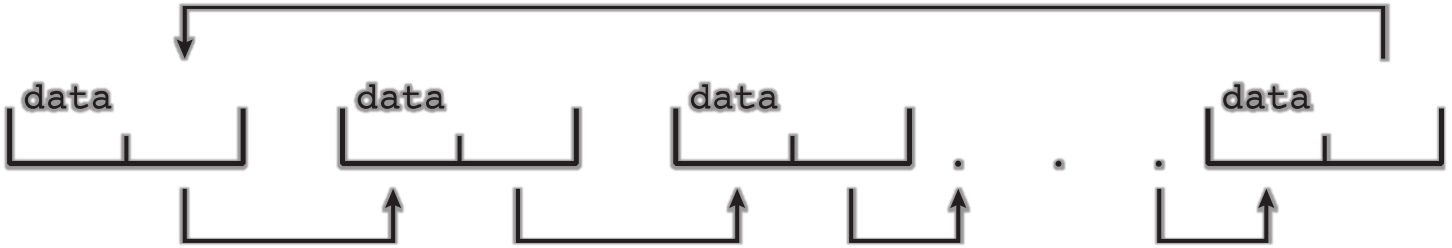
- In a **singly linked list,** each item points to its successor.
- In a **doubly linked list**, a given item can refer either to its predecessor or to its successor.
- In a **circularly linked list**, the last element in the list refers to the first element, rather than to null.

Lists

Doubly linked list



Circular linked list



Lists

Lists Adavantages:

- Linked lists accommodate items of varying sizes.
- Allow easy insertion and deletion of items.

Lists Adavantages:

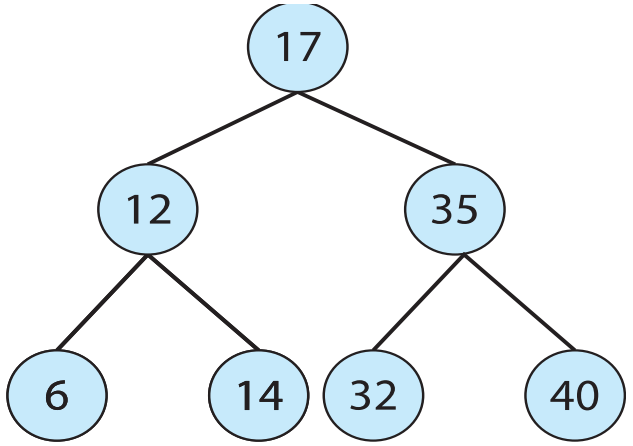
- performance for retrieving a specified item in a list of size n is linear — O(n) , worst case.
- Usage
- List are used by the some of the kernel algorithms,
- Constructing more powerful data structures such as stacks and queues

Stacks & Queues

- Stack uses **LIFO** principle
  - OS often uses a stack when involving function calls.
  - Parameters, local variables and the return address are pushed onto the stack when a function is called
  - Return from the function call pops those items off the stack
- Queue uses **FIFO** principle
  - Tasks waiting to be run on an available CPU are organized in queues
  - Print jobs sent to a printer are printed in the order of submission

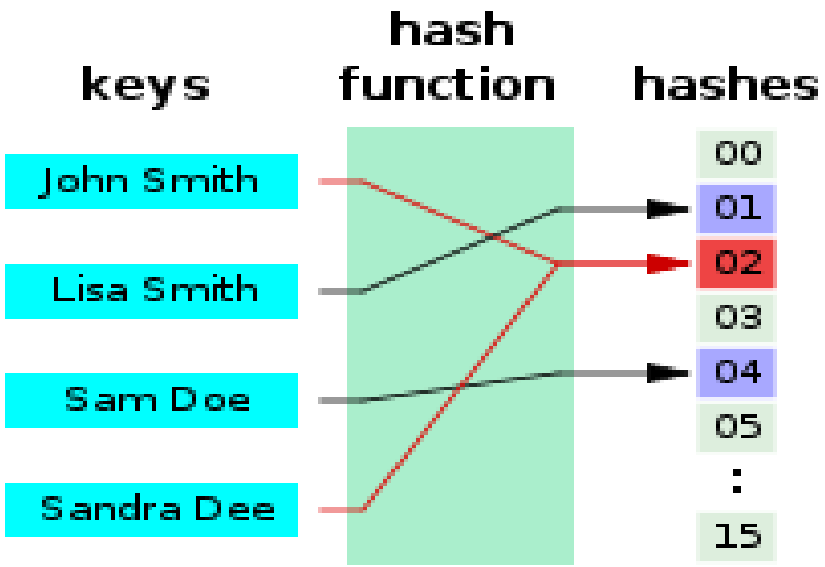
Trees

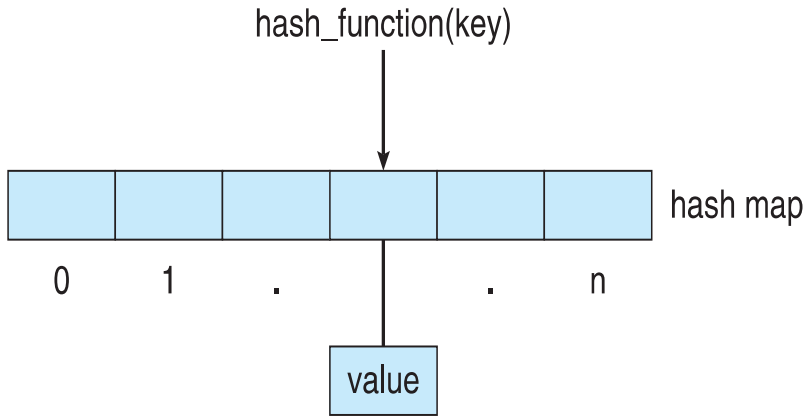
- **Data structure used to represent data hierarchically**
- **Binary search tree**
  ordering between 2 children: left <= right
  - Search performance is *O(n)*
  - **Balanced binary search tree** is *O(lg n)*
    - *Used by Linux as part its CPU-Scheduling algorithm for selecting which task to run next*



Hash Functions and Maps

- Hash functions can result in the same output value for 2 inputs
- **Hash function** can be used to implement a **hash map**
  - Maps or associates key:value pairs using a hash function





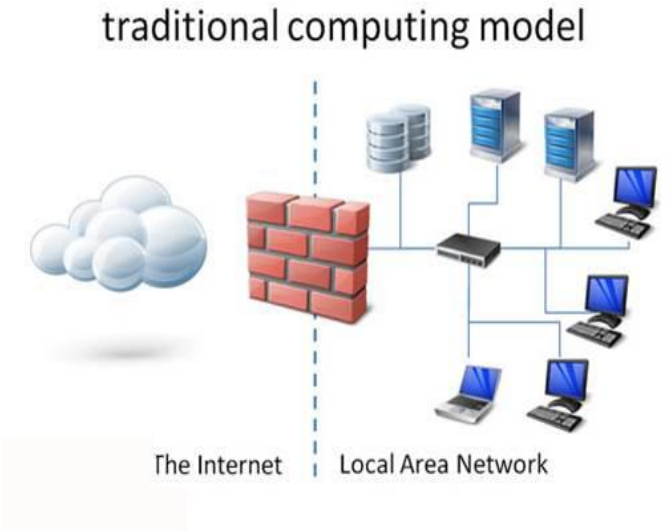
Bitmaps

- **Bitmap** - string of *n* binary digits representing the status of *n* items
- Availability of each resource is indicated by the value of a binary digit
  - 0 – resource is available
  - 1 – resource is unavailable
- Value of the **i**th position in the bitmap is associated with the **i**th resource
  - Ex: bitmap 001011101 shows resources 2, 4, 5, 6, and 8 are unavailable; resources 0, 1, 3, and 7 are available

OPERATING SYSTEMS

Computing Environments – Traditional

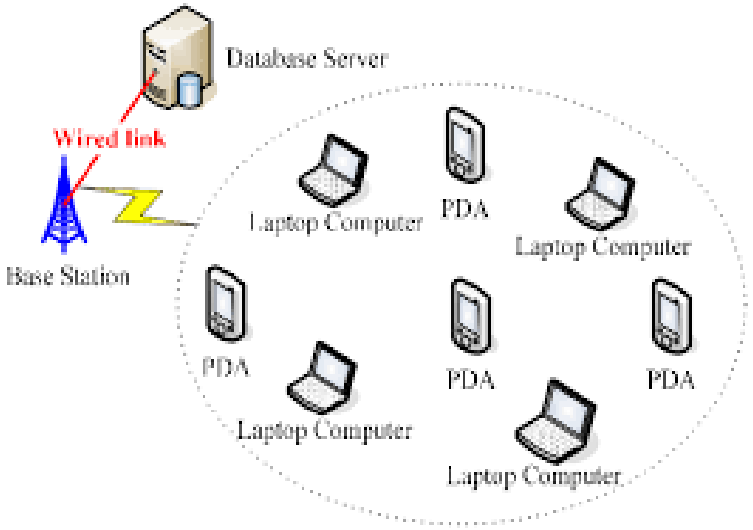
- Stand-alone general purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients**) are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks



OPERATING SYSTEMS

Computing Environments – Mobile

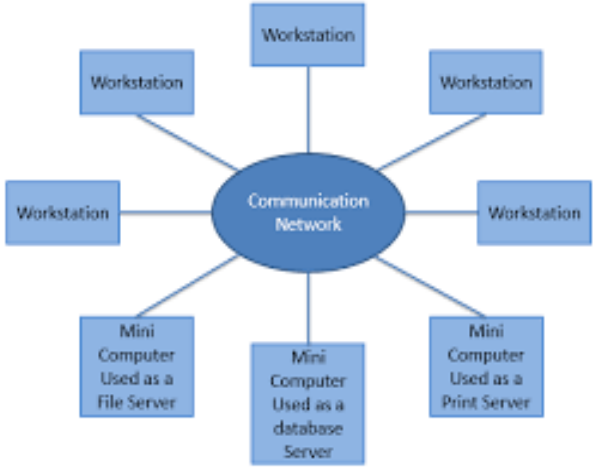
- Handheld smartphones, tablets, etc
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like ***augmented reality***
- Use IEEE 802.11 wireless, or cellular data networks for connectivity
- Leaders are **Apple iOS** and **Google Android**



OPERATING SYSTEMS

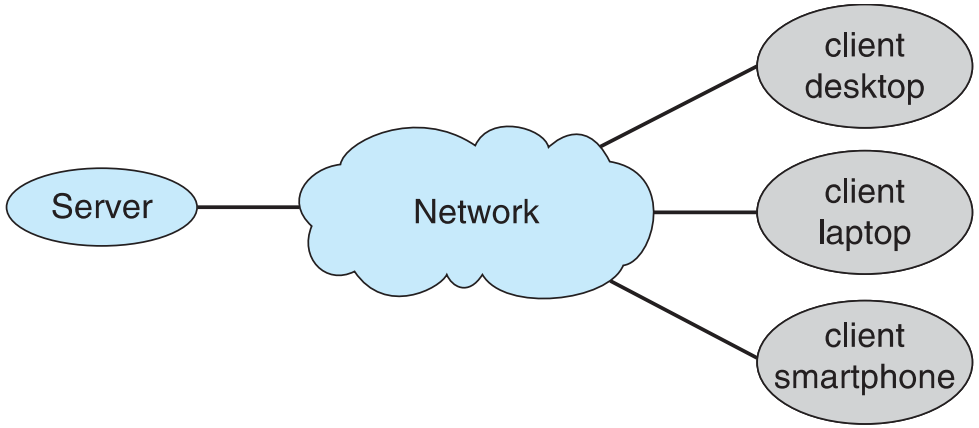
Computing Environments – Distributed

- Distributed computing
  - Collection of separate, possibly heterogeneous, systems networked together
    - **Network** is a communications path, **TCP/IP** most common
      - **Local Area Network** (**LAN**)
      - **Wide Area Network** (**WAN**)
      - **Metropolitan Area Network** (**MAN**)
      - **Personal Area Network** (**PAN**)
  - **Network Operating System** provides features between systems across network
    - Communication scheme allows systems to exchange messages
    - Illusion of a single system



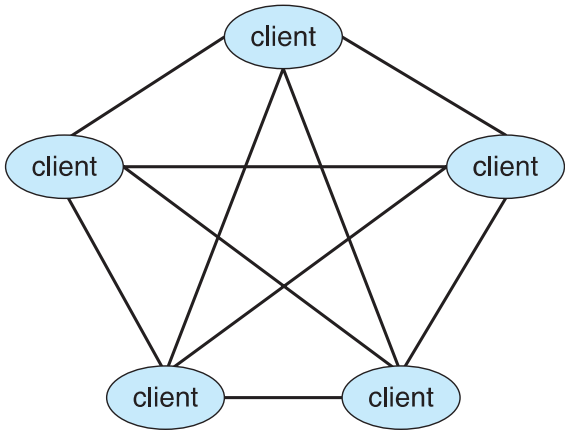
Computing Environments – Client-Server

- Client-Server Computing
  - Dumb terminals replaced by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for clients to store and retrieve files



Computing Environments – Peer-to-Peer

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via ***discovery protocol***
  - Examples include Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype



Computing Environments – Virtualization

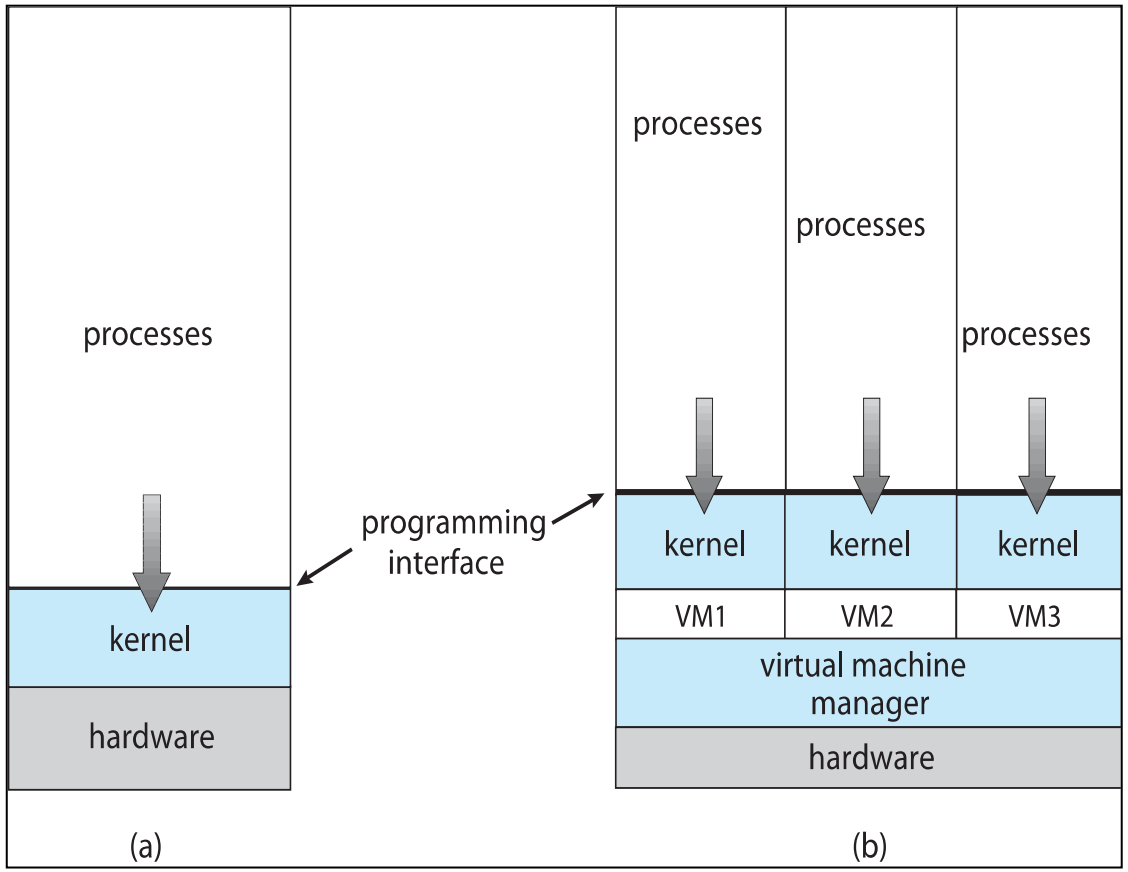
- Allows operating systems to run applications within other OSes
  - Vast and growing industry
- **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
  - Generally slowest method
  - When computer language not compiled to native code – **Interpretation**
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - **VMM** (virtual machine Manager) provides virtualization services

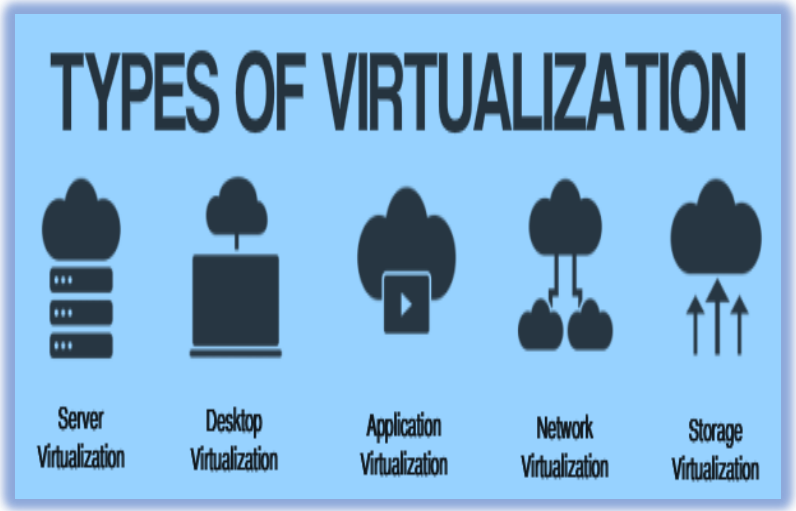
Computing Environments – Virtualization

- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - Apple laptop running Mac OS X host, Windows as a guest
  - Developing apps for multiple OSes without having multiple systems
  - QA testing applications without having multiple systems
  - Executing and managing compute environments within data centers
- VMM can run natively, in which case they are also the host
  - There is no general purpose host then (VMware ESX and Citrix XenServer)

OPERATING SYSTEMS

Computing Environments – Virtualization





Computing Environments – Cloud Computing

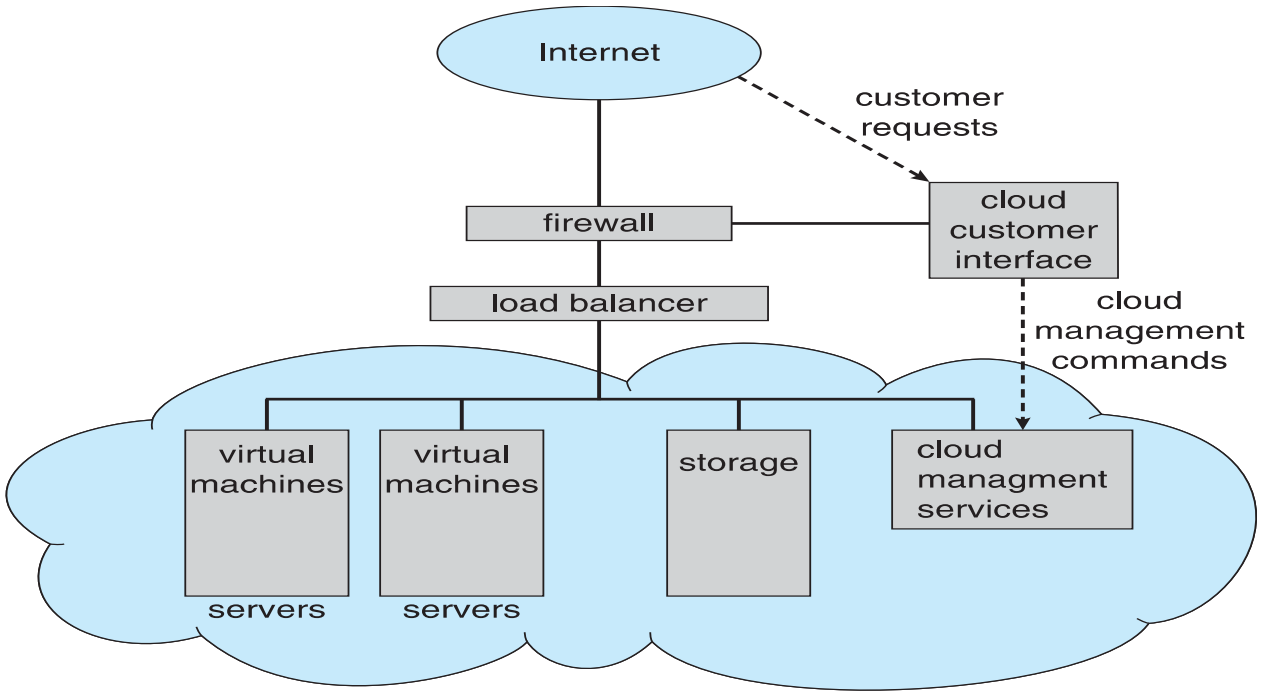
- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for it functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components

Computing Environments – Cloud Computing

- Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor)
- Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server)
- Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use)

Computing Environments – Cloud Computing

- Cloud computing environments composed of traditional OSes, plus VMMs, plus cloud management tools
  - Internet connectivity requires security like firewalls
  - Load balancers spread traffic across multiple applications



Computing Environments – Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
  - Vary considerable, special purpose, limited purpose OS, **real-time OS**
  - Use expanding
- Many other special computing environments as well
  - Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints
  - Processing ***must*** be done within constraint
  - Correct operation only if constraints met

THANK YOU

Chandravva Hebbi
Department of Computer Science Engineering

chandravvahebbi@pes.edu