



Operating System (OS)

National Tsing Hua University
2022, Fall Semester



Historical Prospective

Mainframe Systems

Computer-system architecture

Special-purpose Systems

System Category

- Mainframe Systems

- Batch
- Multi-programming
- Time-sharing

- Computer-system architecture

- Special-purpose Systems

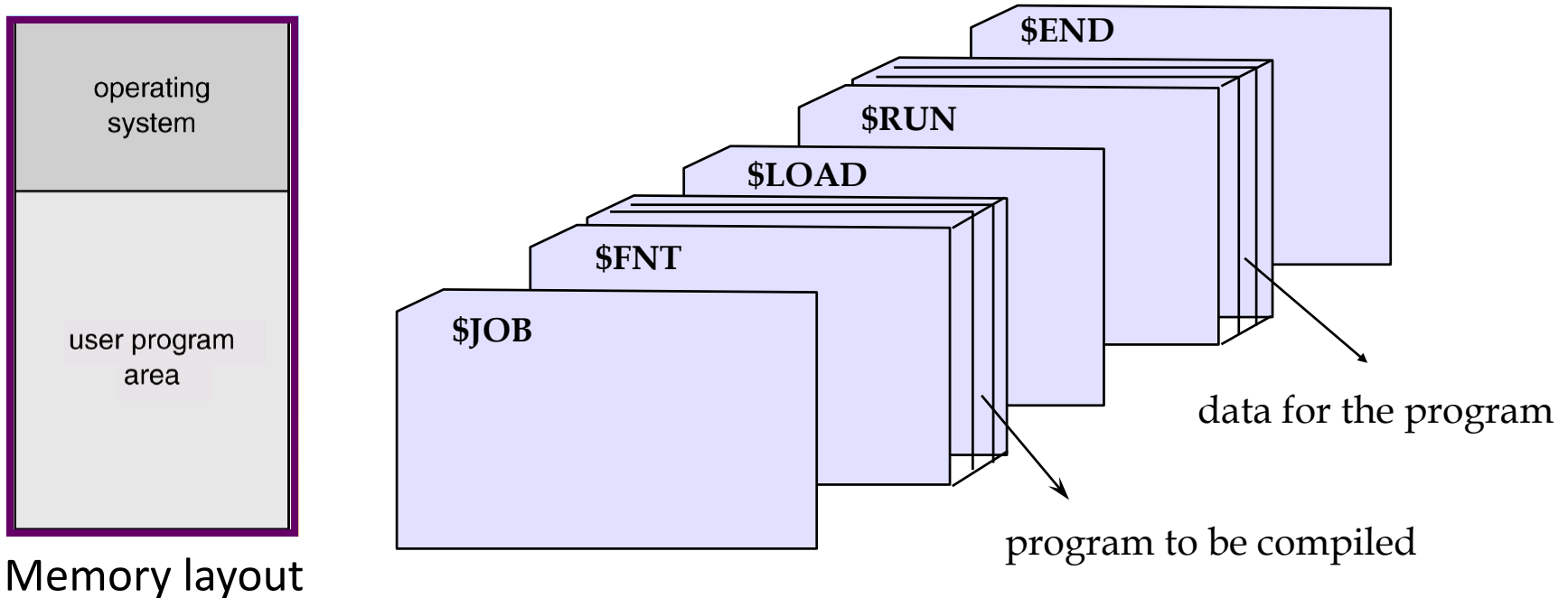
Mainframe Systems

- One of the earliest computers
 - Slow I/O devices: card reader/printer, tape drivers
- Evolution:
 - Batch → Multi-programming → Time-shared
- Still exists in today's world...
 - For critical application with better **reliability & security**
 - Bulk data processing
 - Widely used in hospitals, banks



IBM 704 mainframe in 1954

Mainframe: Batch Systems



■ Processing steps:

- Users submit jobs (program, data, control card)
- Operator sort jobs with similar requirements
- OS simply transfer control from one job to the next

Mainframe: Batch Systems

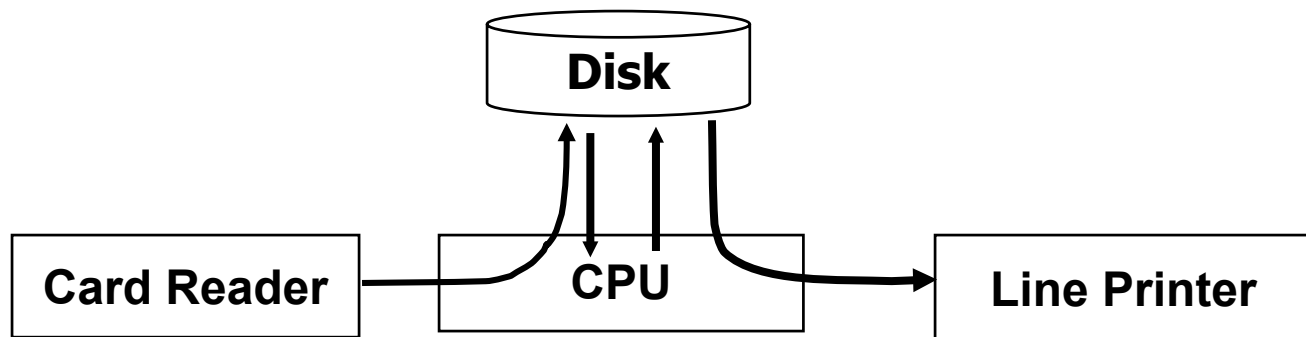
■ Drawbacks:

- One job at a time
- No interaction between users and jobs
- CPU is often idle
 - ◆ I/O speed \ll CPU speed (at least 1:1000)

■ OS doesn't need to make any decision

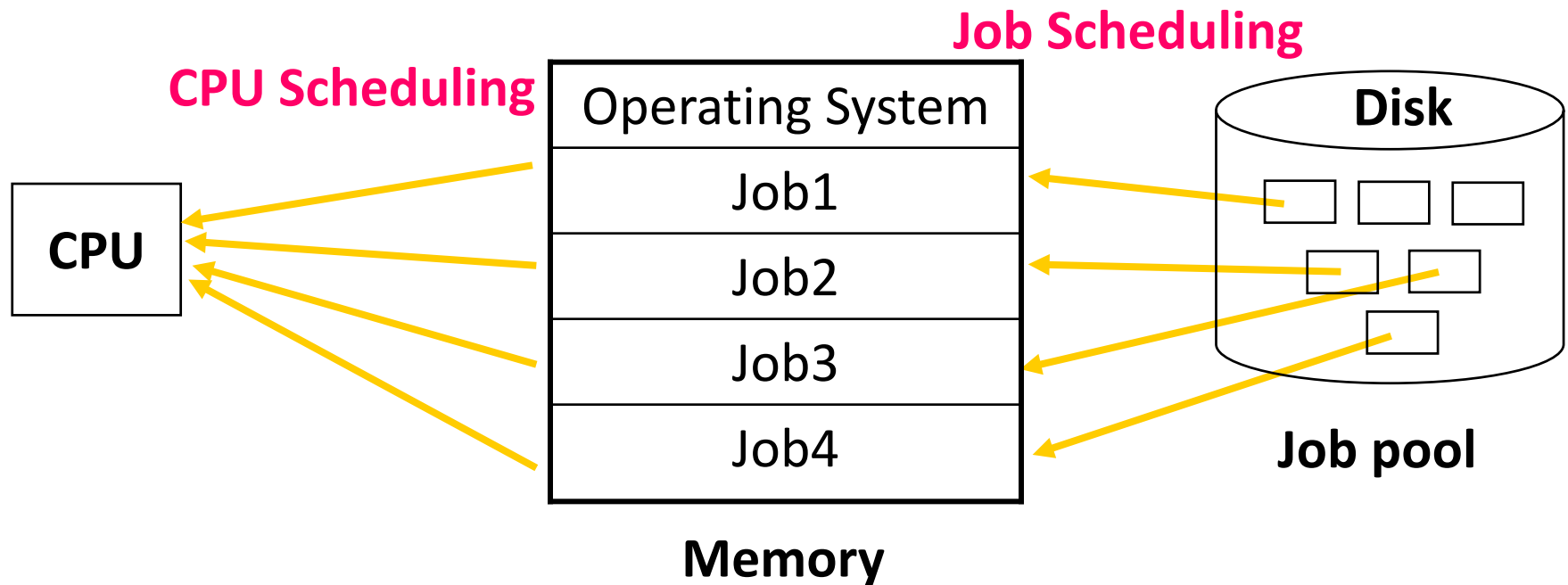
Mainframe: Multi-programming System

- **Overlaps the I/O and computation of jobs**
 - Keeps both CPU and I/O devices working at higher rates
- **Spooling** (Simultaneous Peripheral Operation On-Line)
 - I/O is done with no CPU intervention
 - CPU just needs to be notified when I/O is done



Mainframe: Multi-programming System

- Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them



Mainframe: Multi-programming System

■ OS tasks

- Memory management (chap 9) – the system must allocate the memory to several jobs
- CPU scheduling (chap6) – the system must choose among several jobs ready to run.
- I/O system (chap13) – I/O routine supplied by the system, allocation of devices

Mainframe: Time-sharing System (Multi-tasking System)

- An **interactive** system provides direct communication between the users and the system
 - CPU switches among jobs so frequently that users may interact with programs
 - Users can see results immediately (response time < 1s)
 - Usually, **keyboard/screen** are used
- Multiple **users** can share the computer simultaneously
- Switch job when
 - finish
 - waiting I/O
 - **a short period of time**

Mainframe: Time-sharing System (Multi-tasking System)

■ OS tasks

- Virtual memory (chap 10) – jobs swap in and out of memory to obtain reasonable response time
- File system and disk management (chap11,12) – manage files and disk storage for user data
- Process synchronization and deadlock (chap7,8) – support concurrent execution of programs

Mainframe System Summary

	Batch	Multi-programming	Time-sharing (Multi-tasking)
System Model	Single user Single job	Multiple prog.	Multiple users Multiple prog.
Purpose	Simple	Resource utilization	Interactive Response time
OS features	N.A	CPU scheduling Memory Mgt. I/O system	File system Virtual memory Synchronization Deadlock

System Category

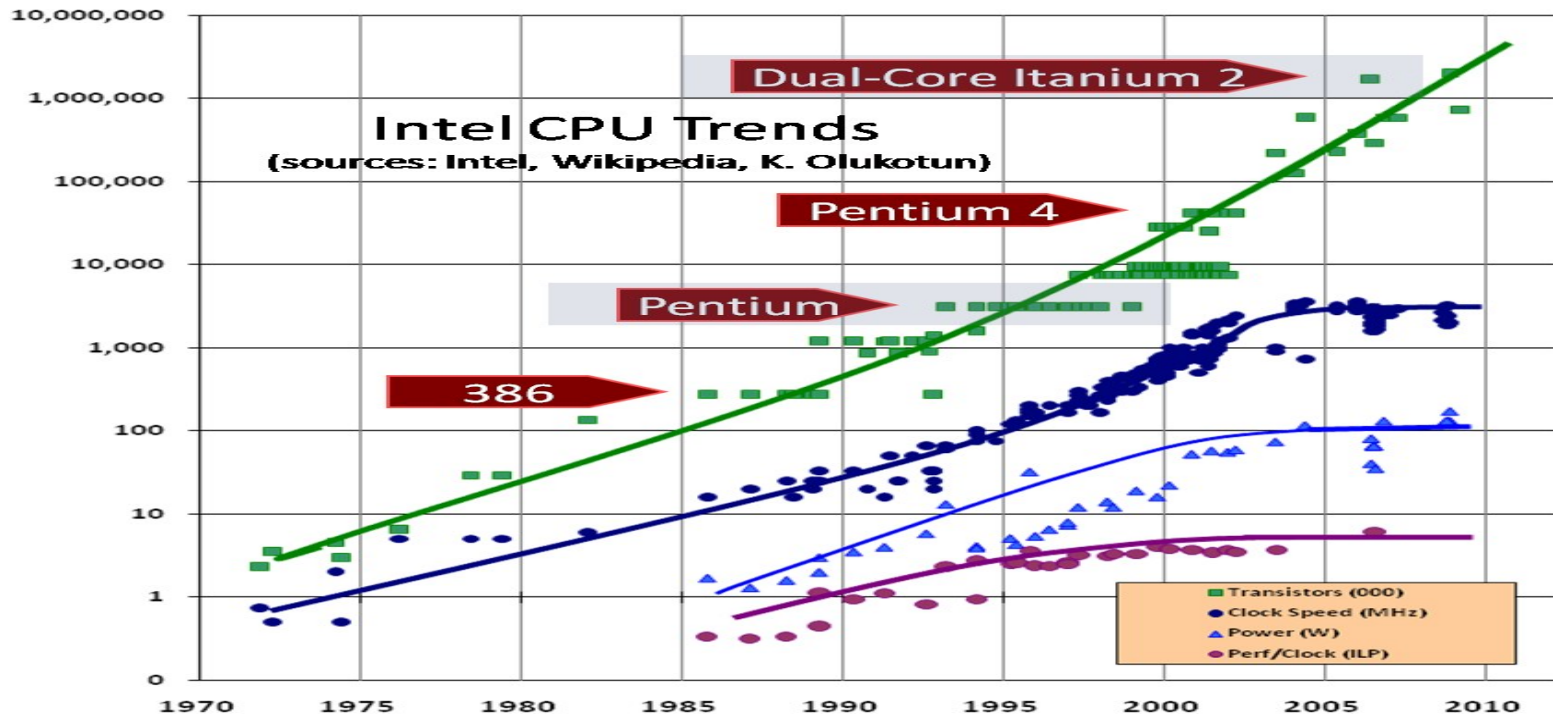
- Mainframe Systems
- Computer-system architecture
 - Desktop Systems: single processor
 - Parallel Systems: tightly coupled
 - Distributed Systems: loosely coupled
- Special-purpose Systems

Desktop Systems: Personal Computers

- Personal computers (PC) – computer system dedicated to a **single user**
- User **convenience and responsiveness** – GUI
- I/O devices – keyboards, **mice**, screens, printers
- Several different types of operating systems
 - Windows, MacOS, Unix, Linux
- Lack of file and OS protection from users
 - Worm, Virus

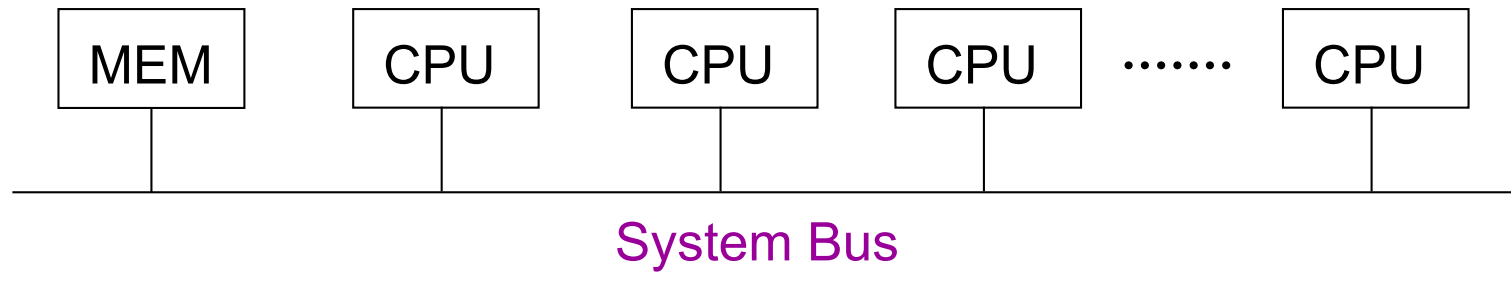
The Death of CPU Scaling

- Increase of transistor density \neq performance
 - The power consumption and clock speed improvements collapsed
 - Non-CPU bottleneck: memory and disk access speed



Parallel Systems

- A.k.a **multiprocessor** or *tightly coupled system*
 - More than one CPU/core in close communication
 - Usually communicate through **shared memory**
- Purposes
 - Throughput, Economical

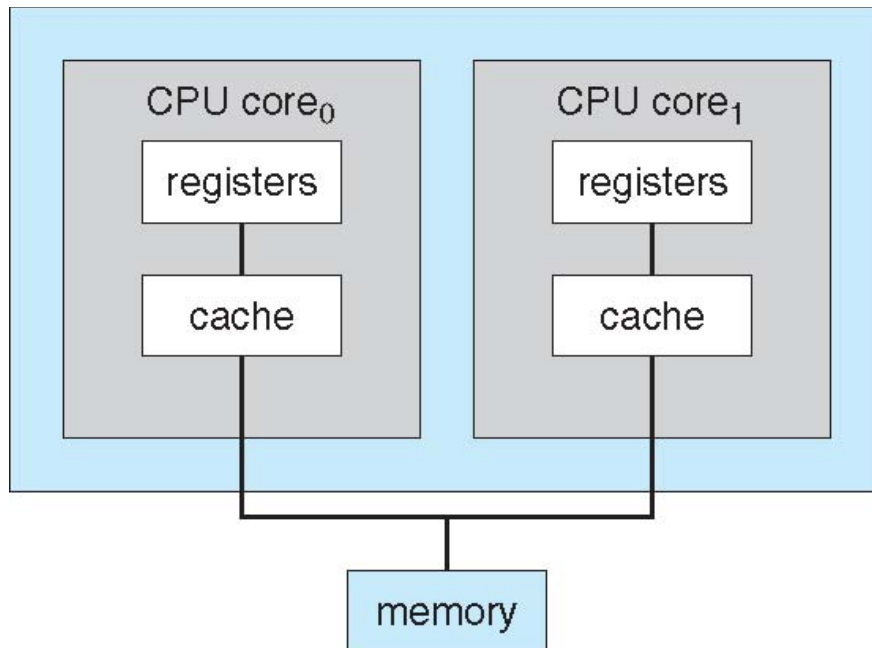


Parallel Systems

- Symmetric multiprocessor system (SMP)
 - Each processor runs the same OS
 - Most popular multiple-processor architecture
 - Require **extensive synchronization** to protect data integrity
- Asymmetric multiprocessor system
 - Each processor is assigned a specific task
 - One Master CPU & multiple slave CPUs
 - More common in extremely large systems

Multi-Core Processor

- A CPU with **multiple cores on the same die (chip)**
- On-chip communication is **faster** than between-chip communication
- One chip with multiple cores uses significantly **less power** than multiple single-core chips



blade servers:

Each blade-processor board boots independently and runs its own OS



Many-Core Processor

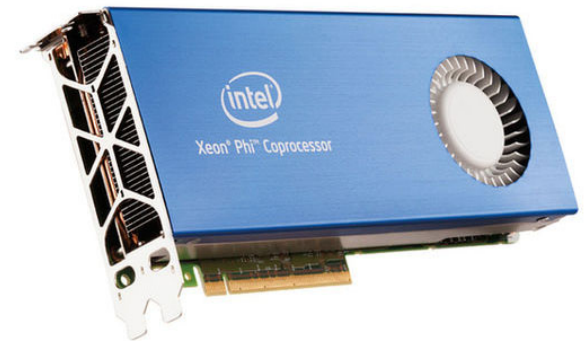
■ Nvidia General-Purpose GPU

- First release in Apr. 2008
- High degree of parallelism
 - ◆ 2,880 thread processor
 - ◆ Single Instruction Multiple Data (SIMD)
- Higher Performance per Watt
 - ◆ 1.43TFlops (x200 faster than a single Intel Core i7)
 - ◆ 245 WATTS
- Lower clock frequency: 600~750 MHz
- \$3000 USD



■ Intel Xeon Phi

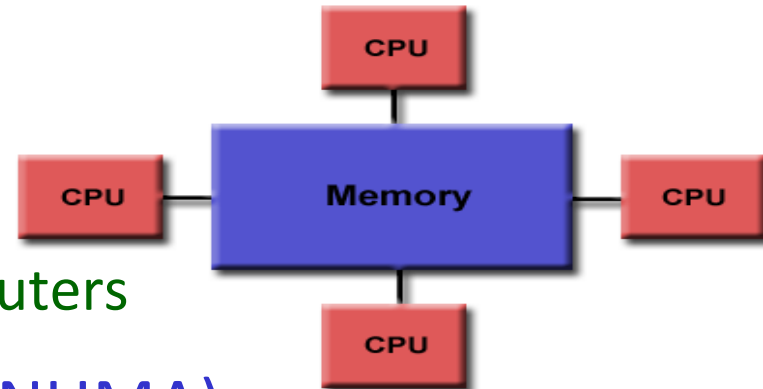
- First release in Nov. 2012
- A coprocessor computer architecture based on Intel Many Integrated Core (MIC)
- 61 cores , 1.2TFlops, 300WATTS
- It runs **Intel assembly code** just like the main CPU in your computer
- Can be used as a **standalone CPU**



Memory Access Architecture

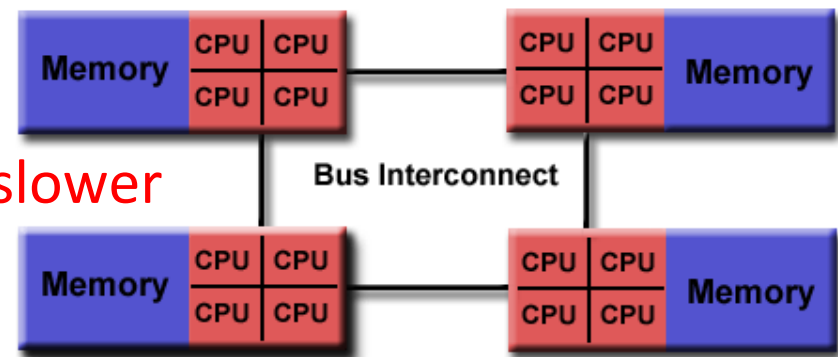
■ Uniform Memory Access (UMA):

- Most commonly represented today by Symmetric Multiprocessor (**SMP**) machines
- **Identical processors**
- **Equal access times** to memory
- Example: **most commodity computers**



■ Non-Uniform Memory Access (NUMA):

- Often made by physically **linking two or more SMPs**
- One SMP can directly access memory of another SMP
- **Memory access across link is slower**
- Example: **IBM Blade server**

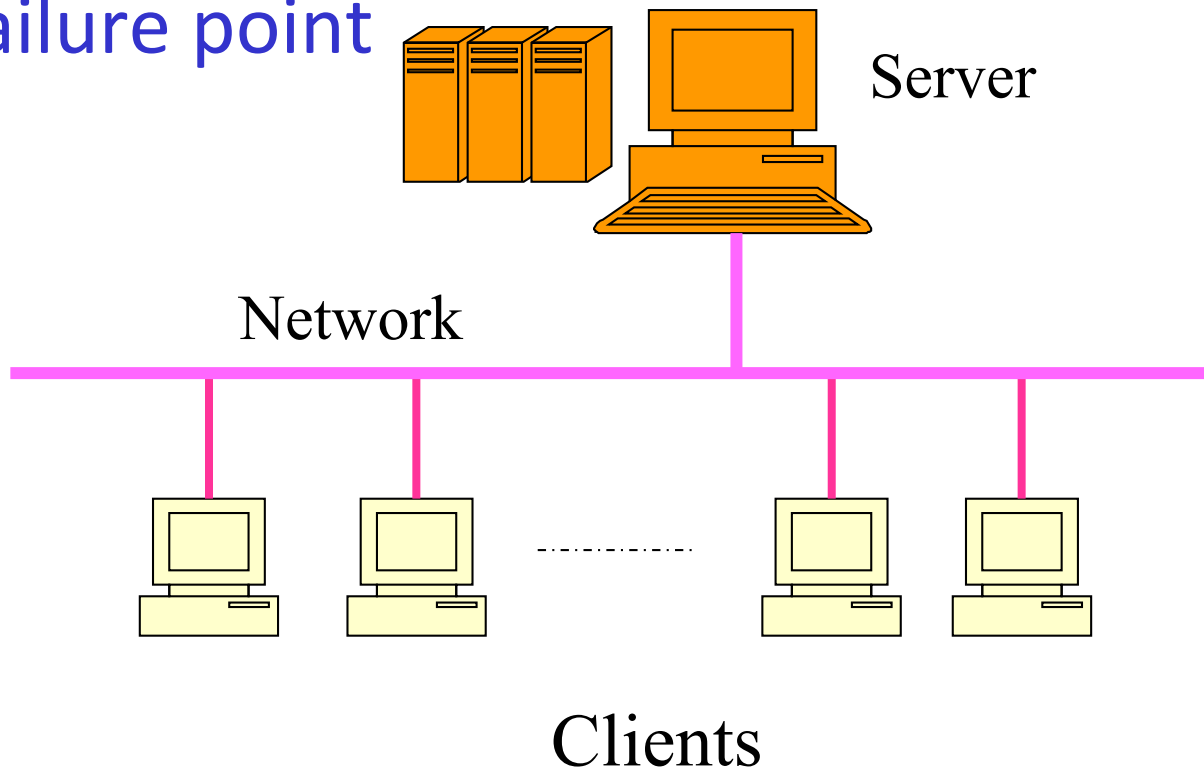


Distributed Systems

- Also known as *loosely coupled system*
 - Each processor has its own **local memory**
 - processors communicate with one another through various communication lines (I/O bus or **network**)
 - Easy to **scale to large number of nodes** (hundreds of thousands, e.g. Internet)
- Purposes
 - Resource sharing
 - Load sharing
 - Reliability
- Architecture: **peer-to-peer** or **client-server**

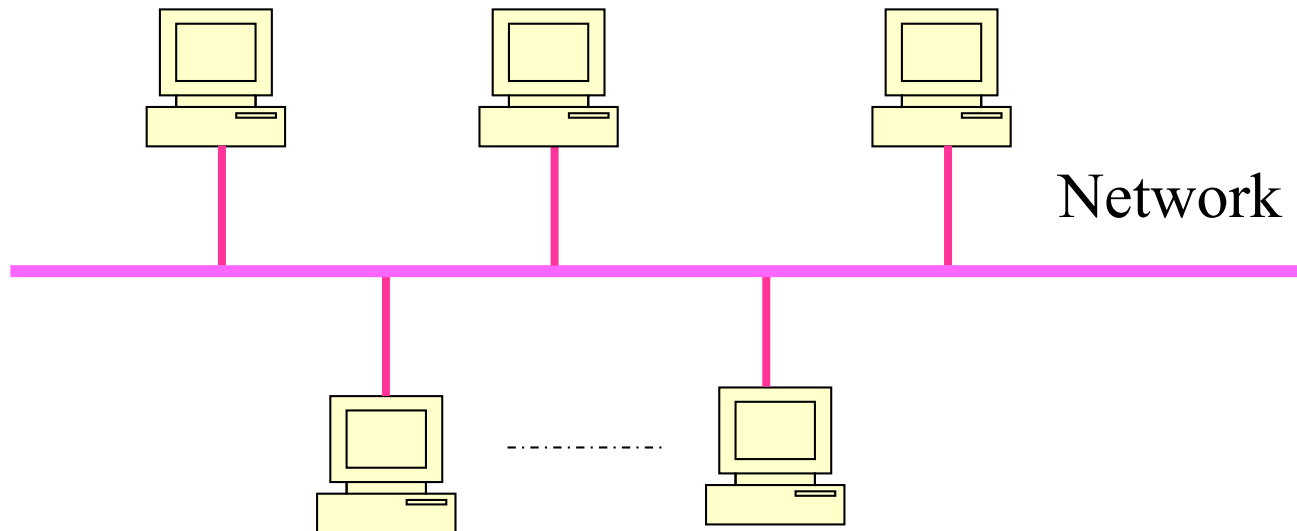
Client-Server Distributed System

- Easier to manage and control resources
- But, server becomes the bottleneck and single failure point



Peer-to-Peer Distributed System

- Every machine is identical in its role in the distributed system – decentralized
- Example: ppStream, bitTorrent, Internet



Clustered Systems

■ Definition:

- Cluster computers share storage and are closely linked via a local area network (LAN) or a faster interconnect, such as InfiniBand (up to 300Gb/s).

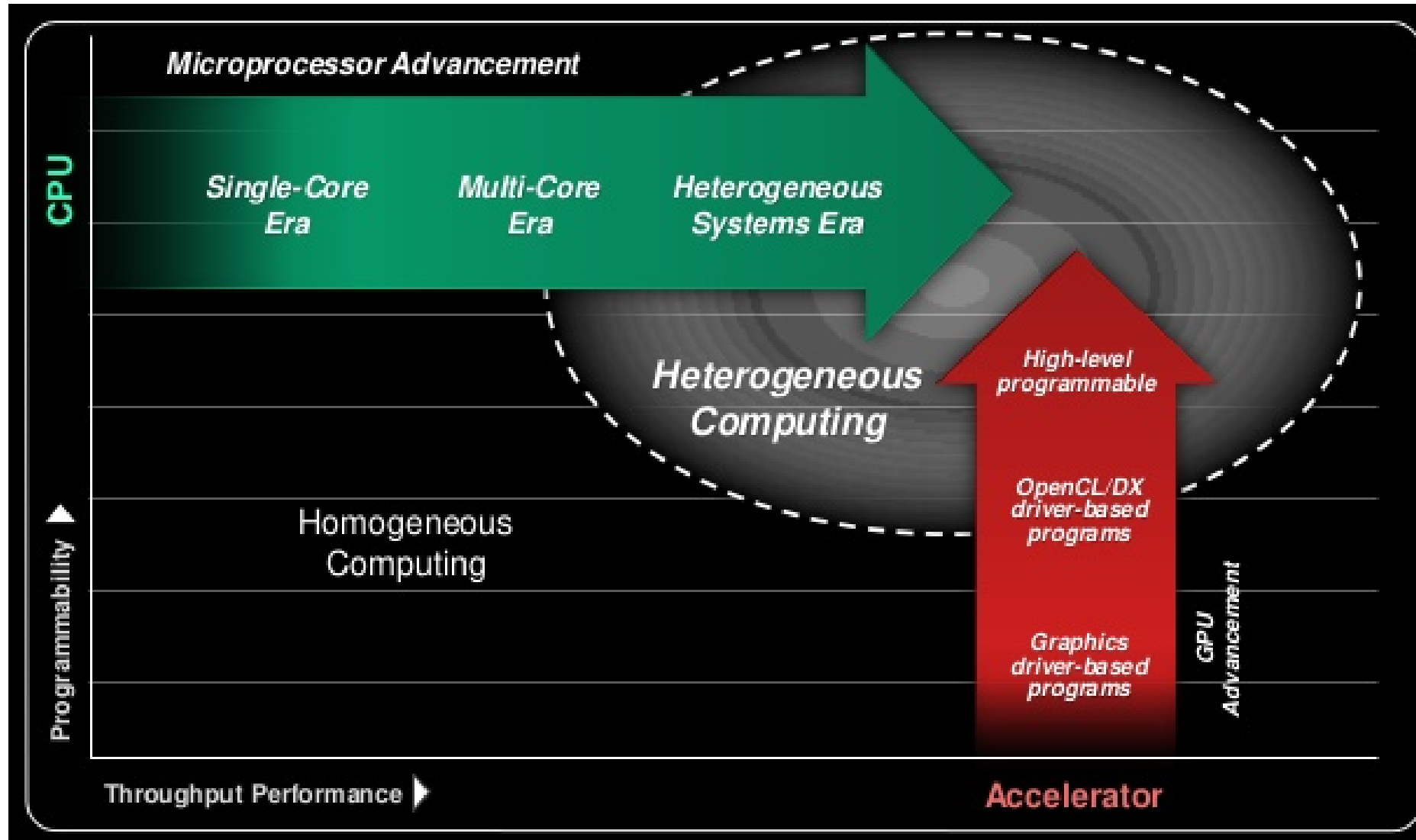
■ *Asymmetric clustering*: one server runs the application while other servers standby

■ *Symmetric clustering*: two or more hosts are running application and are monitoring each other

Heterogeneous Computing

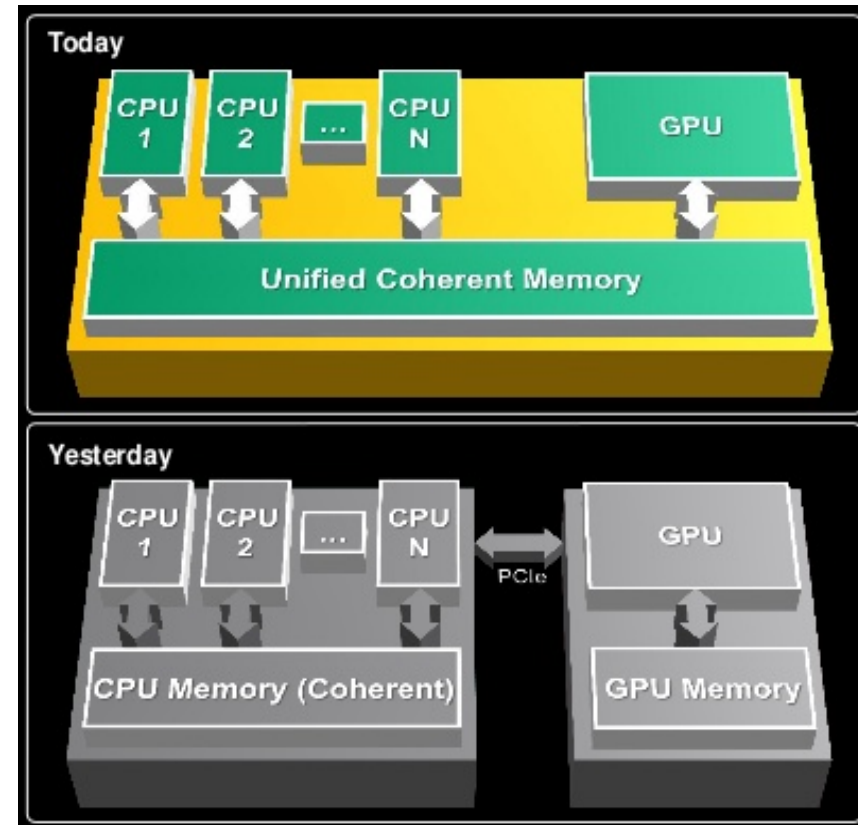
- Heterogeneous computing is an integrated system that consists of different types of (programmable) computing units.
 - DSP (digital signal processor)
 - FPGA (field-programmable gate array)
 - ASIC (application-specific integrated circuit)
 - GPU (graphics processing unit)
 - Co-processor (Intel Xeon Phi)
- A system can be a cell phone or a supercomputer

Shift of Computing Paradigm



Heterogeneous System Architecture (HSA)

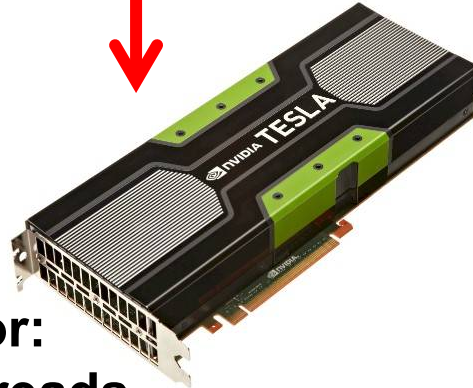
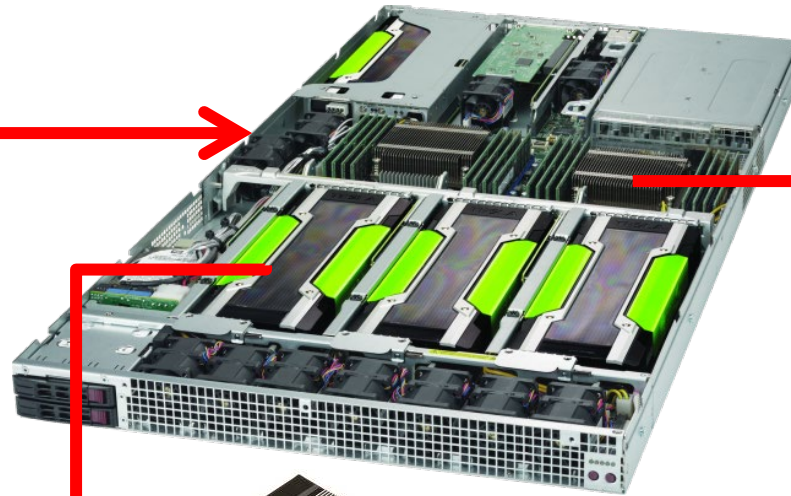
- Aim to provide a common system architecture for designing higher-level programming models for all devices
- Unified coherent memory
 - Single virtual memory address space
 - Prevent memory copy



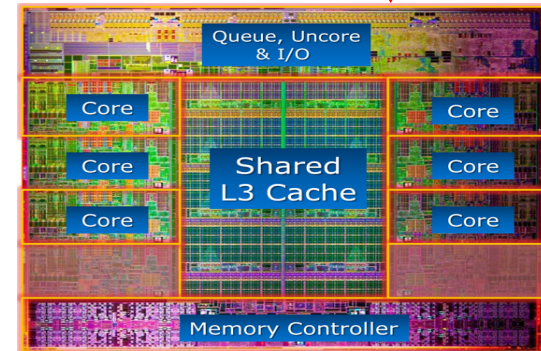
Today's Typical Parallel Computers

Racks: 16~42U

Node/Server: 1~4U



**Multi-core Processor:
100x cores/1000xthreads**



Co-Processor: 4~12 cores

Supercomputers

- Definition: A computer with a **high-level computational capacity** compared to a general-purpose computer
- Its **performance** is measured in floating-point operations per second (**FLOPS**) instead of million instructions per second (MIPS)
- Ranked by the **TOP500** list since 1993
 - According to the **HPL benchmark** results
 - Announced twice a year at ISC and SC conferences

TOP500 List (2021 June)

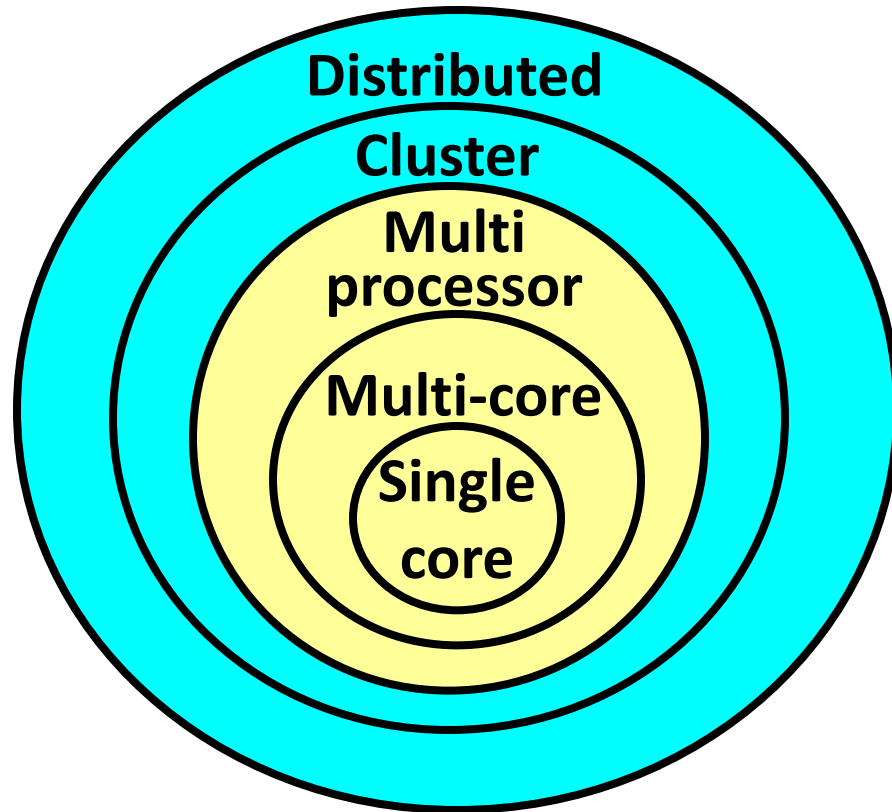
	Country	System	Vendor	Power (kW)	#cores	Accelerator	Rmax	Rpeak (PFLOPS)
1	Japan	Fugaku	Fujitsu	29,899	7.6M		442.0	537.2
2	US	Summit	IBM	10,096	2.4M	Volta V100	148.6	200.8
3	US	Sierra	IBM	7,438	1.5M	Volta V100	94.6	125.7
4	China	TaihuLight	NRCPC	15,371	10.6M		93.0	125.4
5	US	Perlmutter	Cray	2,528	706K	Volta A100	64.6	90.0
15	Swiss	Piz Daint	Cray	2,384	387K	Volta P100	21.2	27.1

2016 June

8	Swiss	Piz Daint	Cray	2,325	116K	Tesla K20X	6.2	7.8
---	-------	-----------	------	-------	------	------------	-----	-----

- Accelerators/Many-core processors provide huge computing power with better energy efficiency
 - **Fugaku** is the first processor to use the **ARMv8.2-A Scalable Vector Extension SIMD** instruction set
 - **TaihuLight** uses **manycore 64-bit RISC** processors
 - **Perlmutter** uses the latest **A100 GPUs** to have highest power efficiency

System Architecture Summary



Tightly coupled 

Loosely coupled 

Trend of Parallel Computers/Computing

Single-Core Era

Enabled by:
Moore's Law
Voltage Scaling

Constraint by:
Power
Complexity

Assembly → C/C++ → Java ...

Heterogeneous Systems Era

Enabled by:
Abundant data
parallelism
Power efficient GPUs

Constraint by:
Programming
models
Comm. overhead

Shader → CUDA → OpenCL ...

Muti-Core Era

Enabled by:
Moore's Law
SMP

Constraint by:
Power
Parallel SW
Scalability

Pthread → OpenMP ...

Distributed System Era

Enabled by:
Networking

Constraint by:
Synchronization
Comm. overhead

MPI → MapReduce ...

System Category

- Mainframe Systems
- Computer-system architecture
- Special-purpose Systems
 - Real-Time Systems
 - Multimedia Systems
 - Handheld Systems

Real-Time Operating Systems (Chap19)

- Well-defined **fixed-time constraints**
 - “Real-time” doesn’t mean speed,
but *keeping deadlines*
- Guaranteed response and reaction times
- Often used as a control device in a dedicated application:
 - Scientific experiments, medical imaging systems, industrial control systems, weapon systems, etc
- Real-time requirement: **hard** or **soft**

Soft vs. Hard Real-Time

■ Soft real-time requirements:

- Missing the deadline is unwanted, but is not immediately critical
- A critical real-time task gets **priority** over other tasks, and retains that priority until it completes
- Examples: multimedia streaming

■ Hard real-time requirements:

- Missing the deadline results in *a fundamental failure*
- **Secondary storage limited or absent**, data stored in short term memory, or read-only memory (ROM)
- Examples: nuclear power plant controller

Multimedia Systems (Chap20)

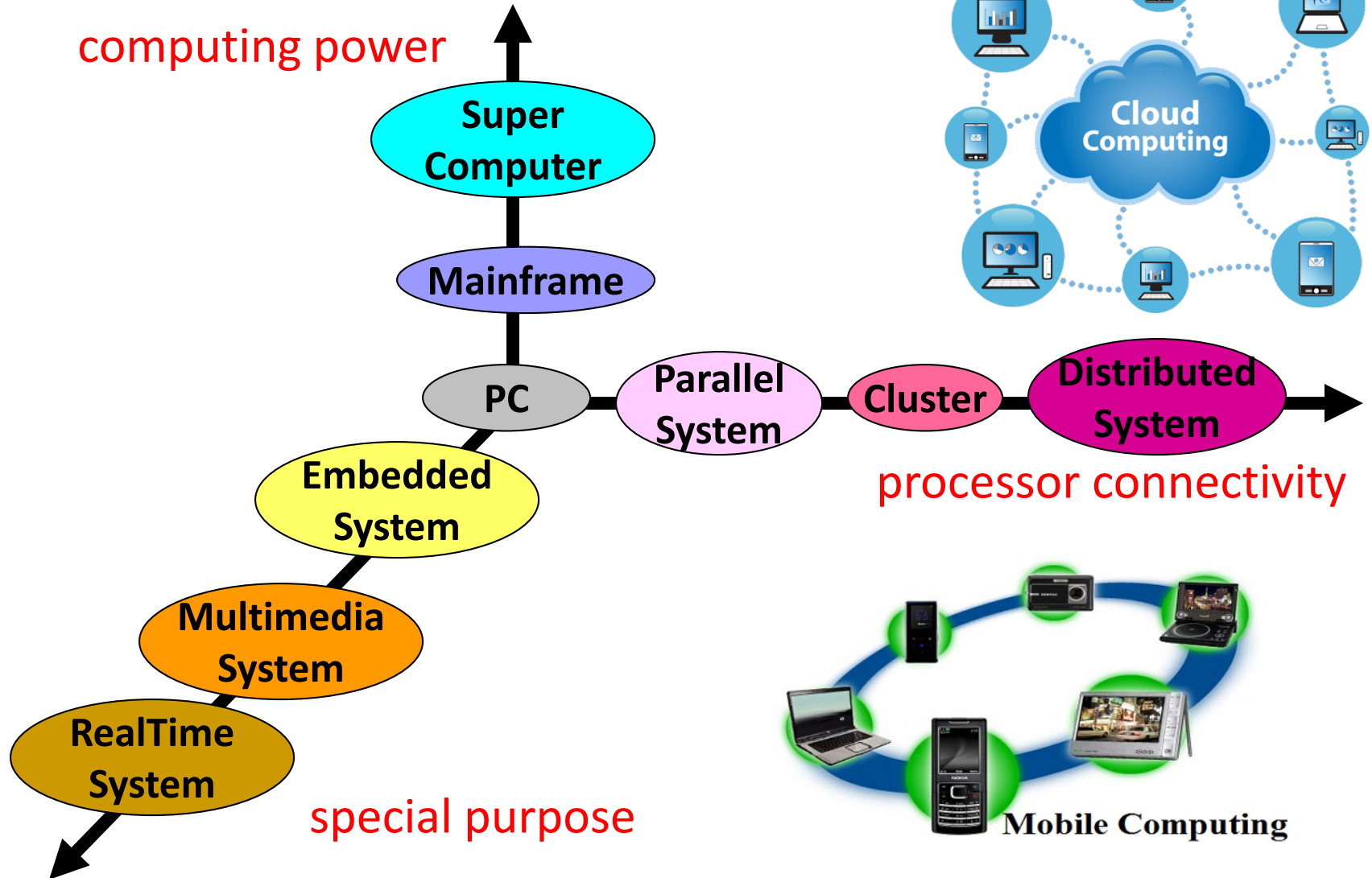
- A wide range of applications including audio and video files (e.g. ppstream, online TV)
- Issues:
 - **Timing constraints:** 24~30 frames per second
 - **On-demand/live streaming:** media file is only played but not stored
 - **Compression:** due to the size and rate of multimedia systems

Handheld/Embedded Systems

- Personal Digital Assistants (PDAs)
- Cellular telephones
- HW specialized OS
- Issues
 - Limited memory
 - Slow processors
 - Battery consumption
 - Small display screens

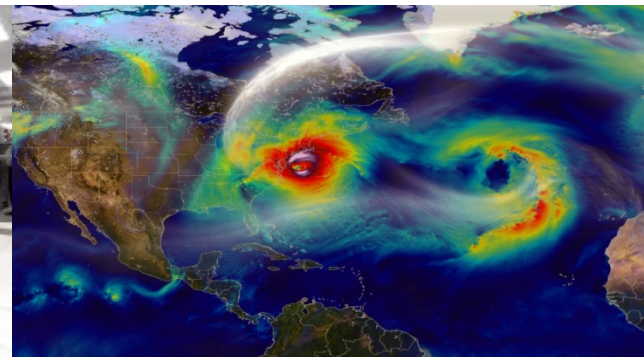


Computer Systems



Computer Systems

- Which system to use? How to use it?
- They have many things in common, but also with different design decisions for their OS.



Review Slides

■ Mainframe system

- Batch, **Multi-programming** and **Time-sharing**

■ Comparison

- Tightly coupled system vs. Loosely coupled system
- Client-server vs. P2P
- Memory architecture: NUMA vs. UMA
- Real-time system: Soft vs. Hard real-time

■ Computer systems

- Multi-core processor
- Supercomputer
- Distributed / Cluster system
- Heterogeneous computing system