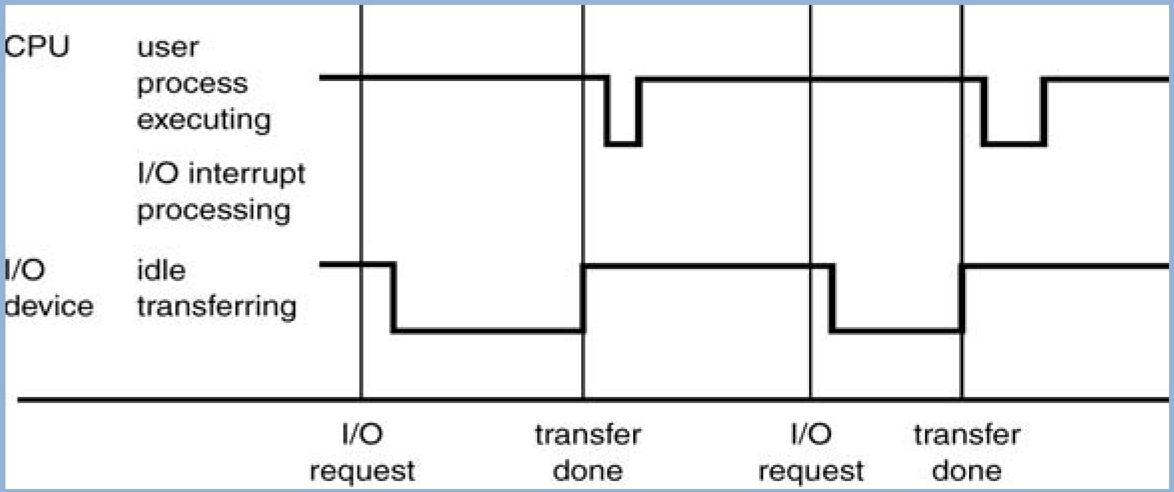
Computer-System Organization (cont.)



Interrupt time line for a single process doing output.

Interrupts are an important part of a computer architecture. Each computer design has its own interrupt mechanism, but several functions are common.

Interrupts must be handled quickly. Since a predefined number of interrupts is possible, a table of pointers (interrupt vectors), which holds the addresses of the interrupt service routine.

Storage Structure

The CPU load instructions only from memory, so for any programs to run they must be first stored there. General-purpose computers run most of their programs from rewriteable memory, called main memory (random access memory-RAM).

Other memory:

DRAM – dynamic random access memory is memory implemented in a semiconductor technology.

ROM – read only memory is memory that cannot be change.

EEPROM - Electrically Erasable Programmable Read-Only Memory is memory that is able to store small amounts of data that must be saved when power is removed.

Von Neumann Architecture

- Fetches an instruction from memory and stores that instruction in the instruction register.
- Instruction is decoded and operand instructions executed (*may cause operands to be fetched from memory and stored in some internal register*)
- Results stored back to memory

Storage Structure (cont.)

Ideally, we want programs and data to reside in main memory permanently. (*But normally not possible*)

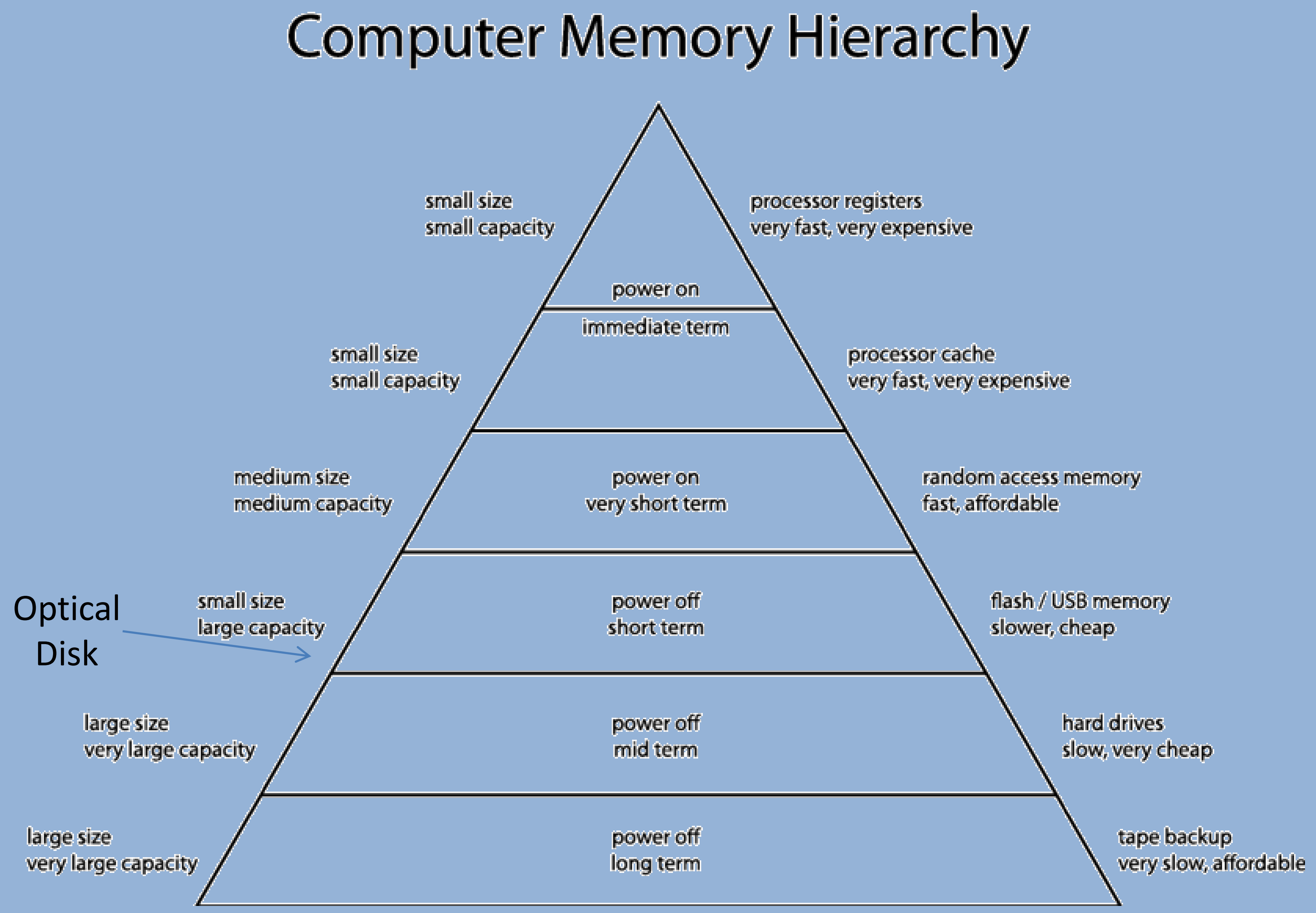
1. Main memory too small to store all needed programs and data.
2. Main memory is a volatile storage device (*loses its contents when power is turned off*)

Other than main memory, computers provide for secondary memory as an extension of main memory.

Secondary memory may consists of:

1. Magnetic disk or tape (floppy, tape and hard disk);
2. Optical disk (CD, DVD)
3. Electronic disk, volatile and nonvolatile

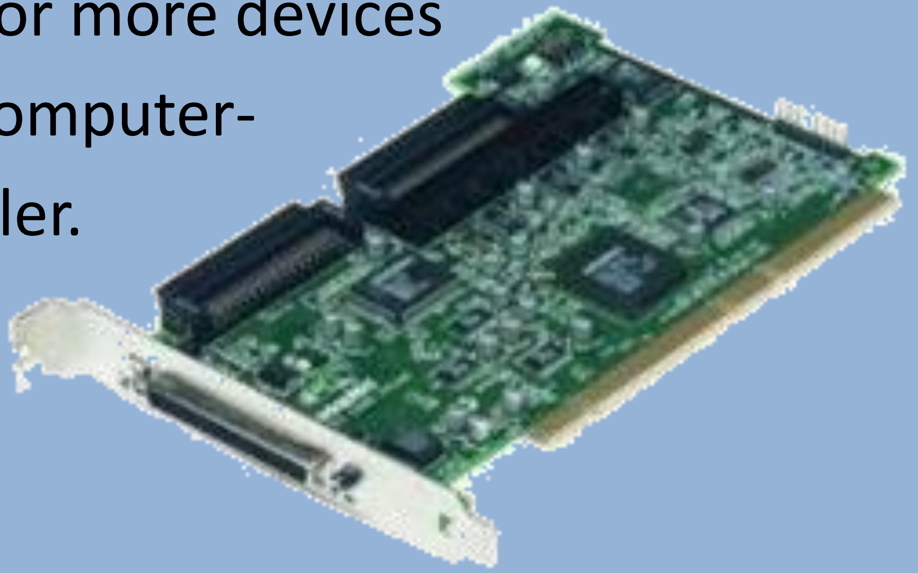
Storage Structure (cont.)



I/O Structure

Storage is only one of many types of I/O devices within a computer. A large portion of operating system code is dedicated to managing I/O, both because of its importance to the reliability and performance of a system and because of the varying nature of the devices.

Depending on the controller, more than one device may be attached. For example, seven or more devices
Can be attached to the small computer-
System interface (SCSI) controller.
SCSI pronounced (scuzzy).



I/O Structure (cont.)

Operating system need device drivers for each device controller.

When starting an I/O operation:

1. Device Control Registers (DCR) is loaded
2. DCR examines the content and determine actions to be taken (i.e. read character from keyboard)
3. Controller transferred data from its device to local buffer.
4. Controller transfers and verifies that operation is complete via an interrupt.

I/O Structure (cont.)

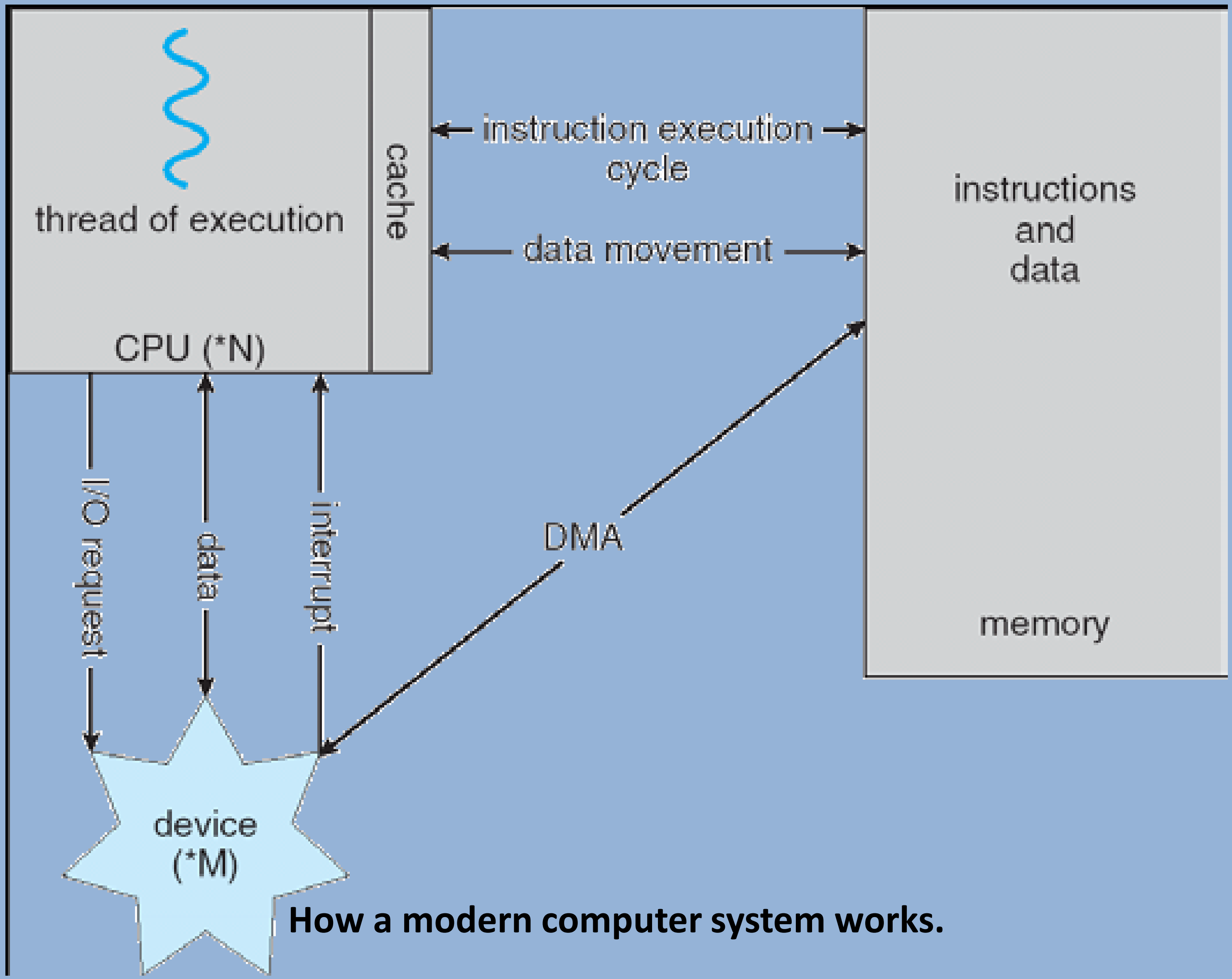
In other operations, the device driver returns status information. (*Good for moving small amount of data but can have a high overhead for bulk data movement.*)

Direct Memory Access is used to solve this problem.

1. Device controller transfers an entire block from it buffer storage to memory with no CPU intervention. (*Only one interrupt per block is sent to tell the device driver the operation is complete; rather than one interrupt per byte*)

Some high-end systems use switch rather than bus architecture. On these systems, multiple components can talk to other components concurrently, rather than competing for cycles on a shared bus, where DMA is more effective.

I/O Structure (cont.)



How a modern computer system works.

Computer-System Architecture

- **Single-Processor Systems**

  Range from PDAs to mainframes. They consists of one main CPU

  general purpose processors (*move data rapidly among components*)

  special purpose processors (*run limited instructions*)

- **Multi-Processor Systems** also known as parallel systems or tightly coupled systems.

  Has two or more processors in close communication sharing computer bus and sometimes the clock, memory and peripheral devices.

Computer-System Architecture (cont.)

Three advantages of multiprocessor systems

1. Increased throughput – when multiple processors cooperate on a task, a certain amount of overhead is incurred. (*i.e. dual processors does not give you double the speed.*)
2. Economy of scale – can cost less than equivalent multiple single-processor systems, because they share peripherals, mass storage, and power supplies. (*i.e. one system have all resources, data; rather than have multiple system with data on each and resources split amount systems.*)
3. Increased reliability – one processor failure will not halt he system, only slow it down. (*i.e. if we have ten processors and one fails, the remaining nine pick up the load of the failed processor.*)

Computer-System Architecture (cont.)

Multiprocessor Systems

Graceful degradation – the ability to continue providing service proportional to the level of surviving hardware in the event of a failure.

Fault tolerance – systems that can suffer a failure of any single component and still continue to operate. The problem is detected, diagnosed and repaired if possible.

Two type of multiple-processor systems used today

1. Asymmetric multiprocessing
2. Symmetric multiprocessing (SMP)

Computer-System Architecture (cont.)

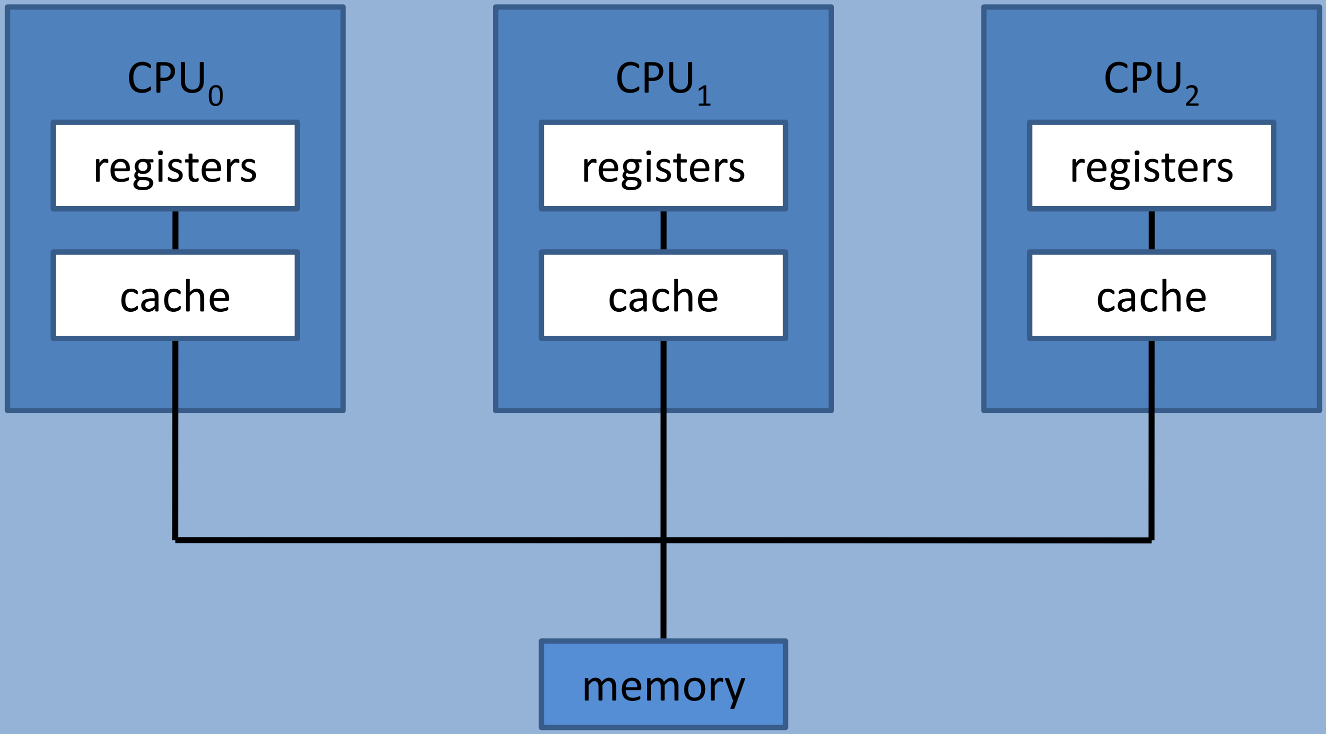
Asymmetric multiprocessing - each processor is assigned a specific task.

1. One processor is designated the master processor controls the system.
2. The other processors either look to the master for instruction or have predefined tasks.
3. The master processor schedules and allocates work to the slave processors.

Symmetric multiprocessing (SMP) – each processor performs all tasks within the operating system.

1. All processors are peers; there are no slave relationship between processors.
2. Each processor has its own set of registers, as well as private and local cache. (*all share physical memory*)

Computer-System Architecture (cont.)



Symmetric multiprocessing architecture.

The benefit of this model is that many processes can run simultaneously –N processes can run if there are N CPUs – without causing a significant deterioration of performance.

Computer-System Architecture (cont.)

Symmetric multiprocessing involves:

- Ensuring that data reach the appropriate processor
- One process may be idle while another is overloaded. (can avoid this by allowing processor to share certain data structures)
- Allow processes and resources – such as memory – to be shared dynamically among the various processors which can lower the variance among the processors.

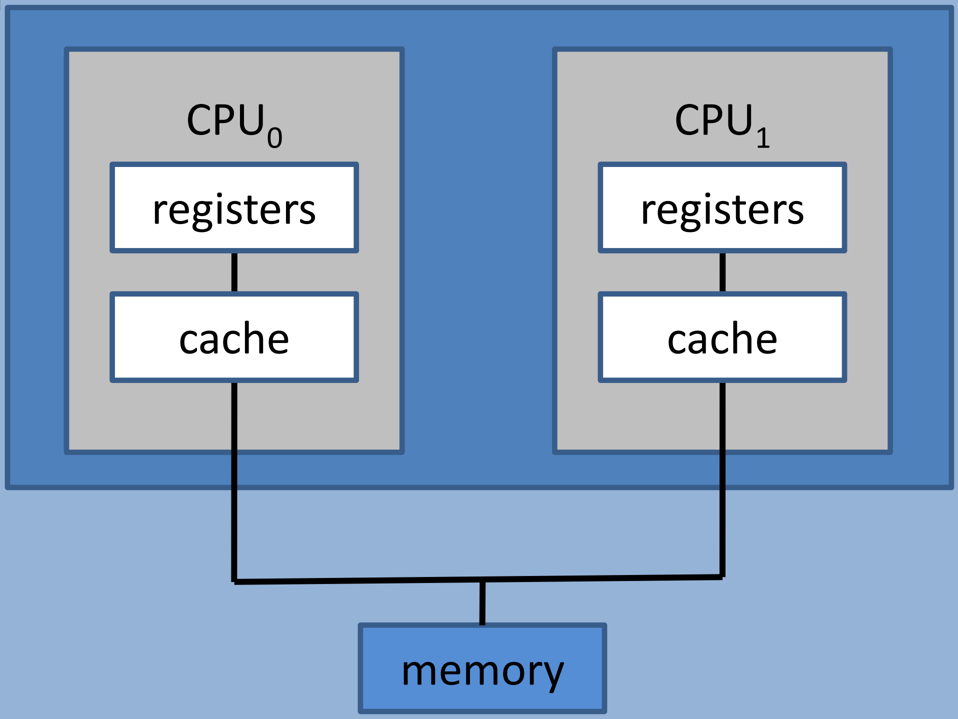
Multiprocessing adds CPUs to increase computing power. If the CPU has an integrated memory controller, then adding CPUs can also increase the amount of memory addressable in the system. Either way, multiprocessing can cause a system to change it memory access model from uniform memory access to non Uniform memory.

UMA – access to any RAM from any CPU takes the same amount of time.

NUMA – some parts of memory may take longer than other parts

Computer-System Architecture (cont.)

A recent trend in CPU design is to include multiple computing cores on a single chip. In essence, there are multiprocessor chips. They can be more efficient than multiple chips with single cores because on-chip communication is faster than between-chip communication.



A Dual-core design with two cores placed on the same chip

Computer-System Architecture (cont.)

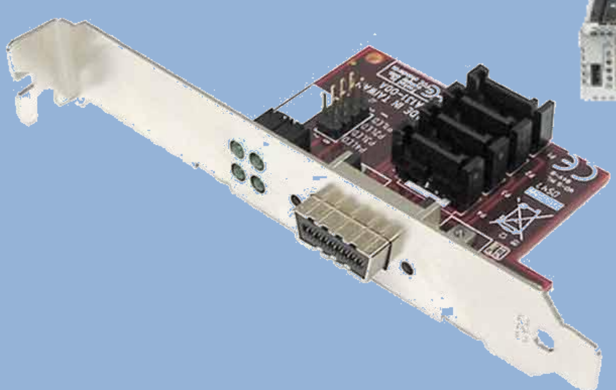
- **Clustered Systems** – another type of multiple CPU system. Like multi processor systems, clustered system gather together multiple CPUs to accomplish computational work.

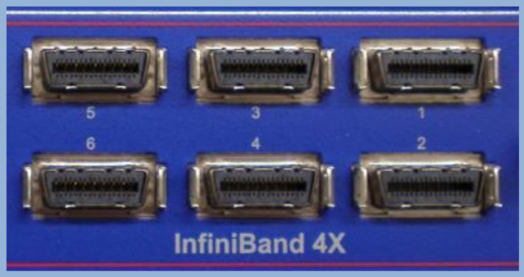
Clusters are different from Multiprocessor Systems in that clusters are composed of two or more individual systems – or nodes – joined together. Clusters are defined as computers that share storage and are closely linked via a local-area network (LAN) or an faster interconnect, such as InfiniBand.

Clustering is used to provide high-availability service (continue to perform if one or more systems in the cluster fails).

InfiniBand

- **InfiniBand** is a switched fabric communications link used in high-performance computing and enterprise data centers. Its features include high throughput, low latency, quality of service and failover, and it is designed to be scalable. The InfiniBand architecture specification defines a connection between processor nodes and high performance I/O nodes such as storage devices.





Computer-System Architecture (cont.)

General Operation of a Clustered System

1. Cluster software runs on each node.
2. Each node has the capability of monitoring each other over the LAN
3. When monitored machine fails the other node takes ownership of its storage the restarts applications that were running on the failed machine. (User and client normally sees only a brief interruption of service.)

A high level of redundancy occurs to accomplish these tasks. Clusters can be constructed asymmetrically or symmetrically.

Computer-System Architecture (cont.)

Asymmetrically Constructed Clusters

1. One machine is in hot-standby mode, while the other is running the applications.
2. The hot-standby host machine does nothing but monitor the active server. If the server fails the hot-standby takes over as the active server.

Symmetrically Constructed Clusters

1. Two or more hosts are running applications and are monitoring each other
2. When monitored machine fails the other node takes ownership of its storage the restarts applications that were running on the failed machine.