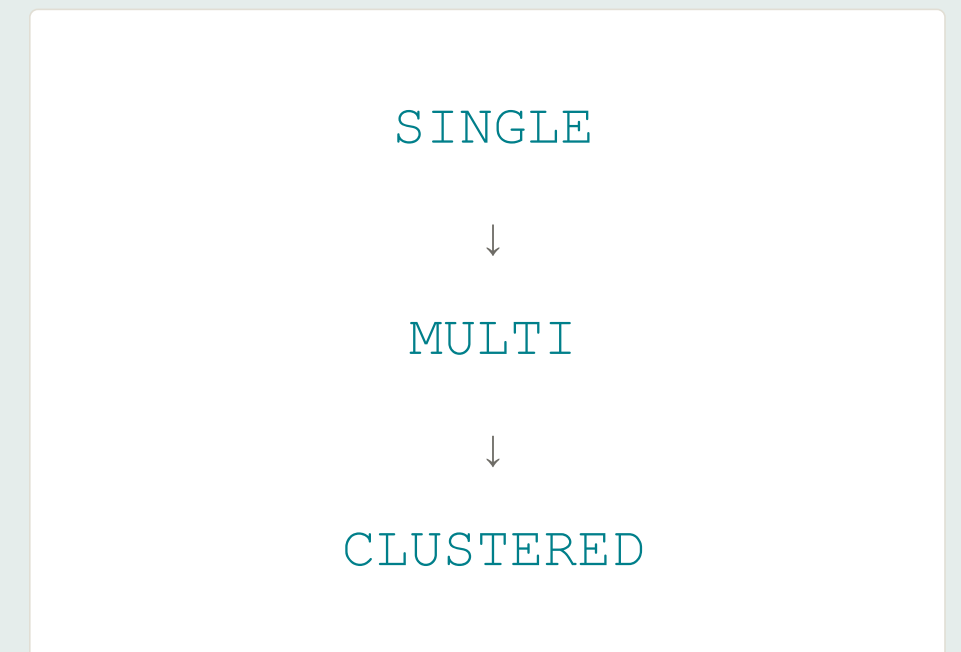


Computer-System Architecture

Section 1.3 — The Processor: single, multi, and clustered



Prof. Mostafa Elhosseini

Professor of Smart Systems Engineering

Organizing a Computer System by Processor Count

1.3.1

Single-Processor Systems

One general-purpose CPU with a single core, plus **special-purpose processors**.

1.3.2

Multiprocessor Systems

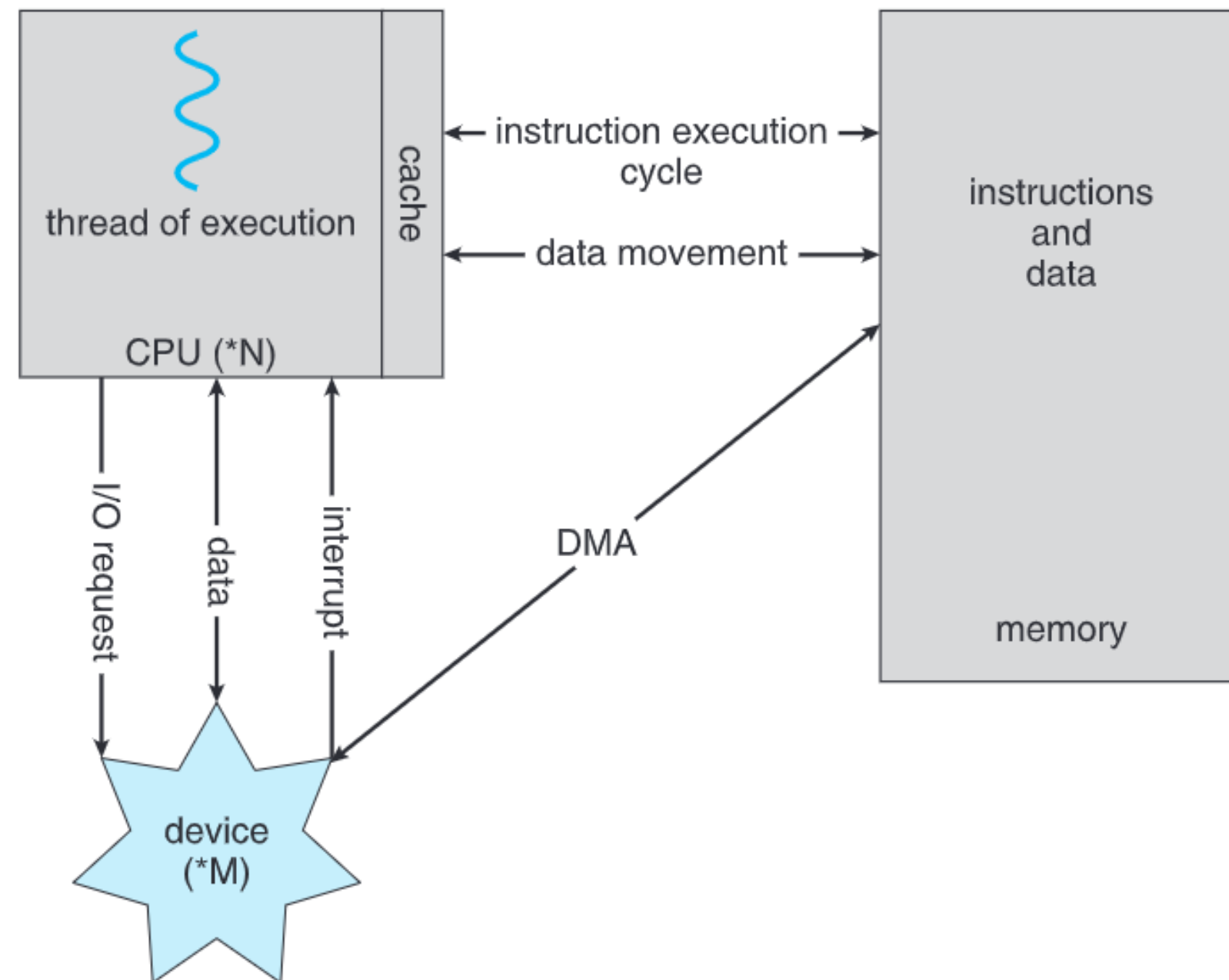
SMP, multicore chips, NUMA, and blade servers — **tightly coupled**.

1.3.3

Clustered Systems

Two or more independent systems joined together — **loosely coupled**.

How a Modern Computer System Works



All devices exchange data over a **common bus**. Interrupt-driven I/O suits small transfers, but bulk movement makes the overhead expensive.

Direct memory access (DMA) lets the device controller transfer a whole block to or from main memory with **no CPU intervention** — one interrupt per block, not one per byte.

Some high-end systems use a **switch** instead of a bus, so components can talk **concurrently** — making DMA more effective still.

Definitions of Computer System Components

CPU

The hardware that **executes instructions**.

Processor

A **physical chip** that contains one or more CPUs.

Core

The **basic computation unit** of the CPU.

Multicore

Including **multiple computing cores** on the same CPU.

Multiprocessor

Including **multiple processors**.

Single-Processor Systems

Many years ago, most systems used a **single processor containing one CPU with a single processing core** .

The core **executes instructions** and holds **registers** for storing data locally. That one main CPU can execute a **general-purpose instruction set** , including instructions from processes.

Such systems still contain other **special-purpose processors** — disk, keyboard, and graphics controllers among them.

The chef = general-purpose CPU.
Coffee machine / oven / mixer = special-purpose processors/controllers.

if there is **only one general-purpose CPU with a single processing core** , the system is a single-processor system. By that definition, **very few contemporary systems qualify**.

Special-Purpose Processors

Limited instruction set

They run a **restricted set of instructions** and **do not run processes** .

Designed only for a restricted task

Sometimes OS-managed

The OS may send them their **next task** and monitor status — a disk-controller

Sometimes autonomous

Low-level components built into the hardware that the OS **cannot communicate with at all** .

Embedded processors responsible for: power management, thermal control, SSD flash management,

A **keyboard microcontroller** may scan keys, detect presses/releases, debounce signals, and generate scan codes. It is a processor, but it is not normally running Chrome, Word, or operating-system processes.—It **does not turn a single-processor system into a multiprocessor**.

Multiprocessor Systems

On modern computers — from mobile devices to servers — multiprocessor systems **dominate the landscape of computing** .

- Traditionally such systems have **two or more processors (Tightly coupled - مقترن بإحكام closely connected)** , each with a single-core CPU.
- The processors **share the computer bus** , and sometimes the **clock, memory, and peripheral devices** .

The primary advantage is **increased throughput** : with more processors, we expect **more work done in less time**.

Several chefs working inside the same kitchen and sharing the same refrigerator, ingredients, and equipment.

CPU 1 

CPU 2 → Shared Memory + Shared I/O

CPU 3 

- This gives true parallelism.
- CPU/Core 1 → Chrome CPU/Core 2 → Antivirus
CPU/Core 3 → Video encoding CPU/Core 4 → OS/background task

The speed-up ratio with N processors is **not** N — it is **less than** N .

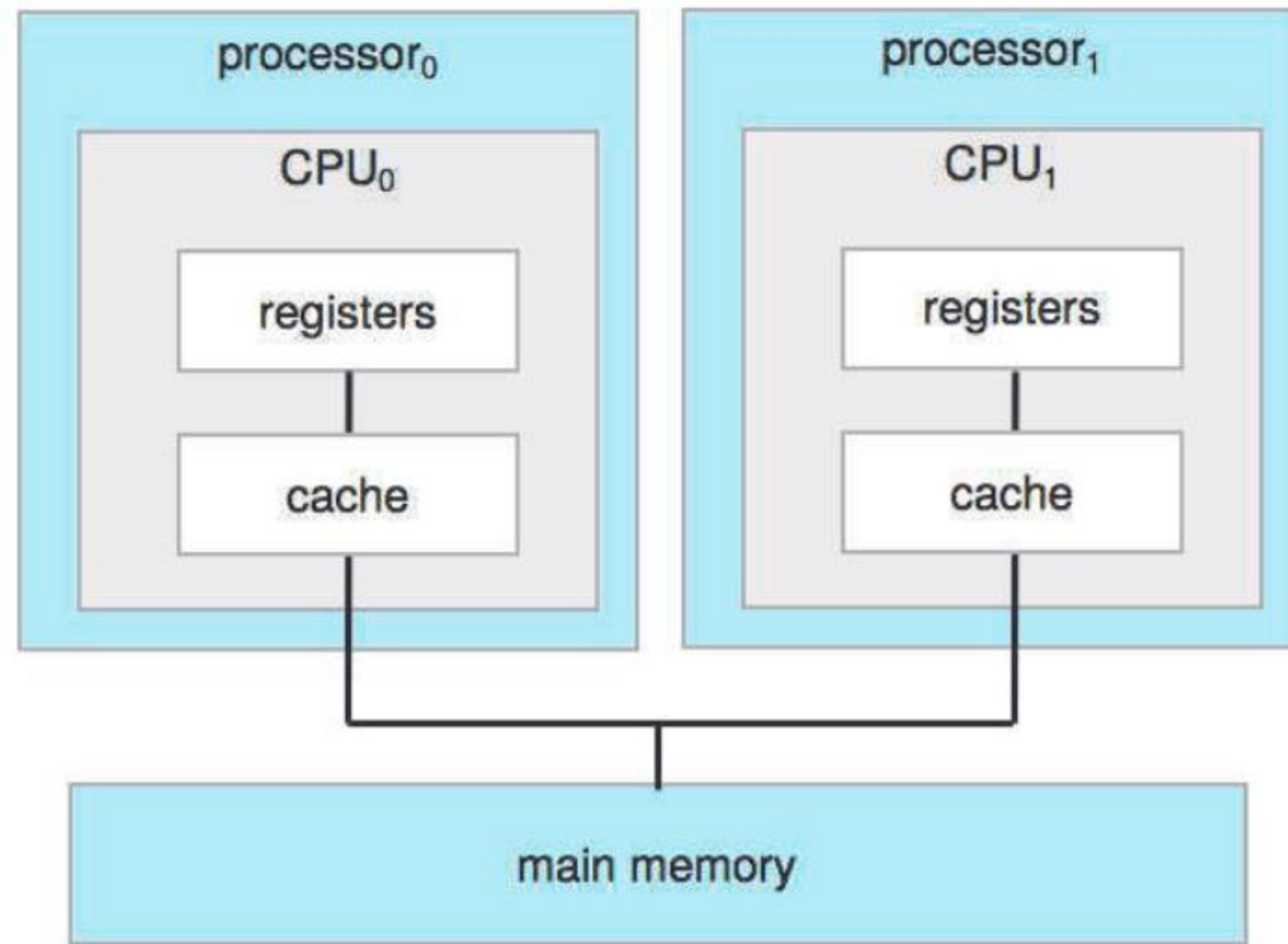
If you have N processors, you usually do not get N -times speedup.

Suppose one processor finishes a task in 100 seconds. If you use 4 processors, the ideal dream would be: $100 / 4 = 25$ seconds
which would mean: **Speedup = 4×**

When multiple processors cooperate on a task, **overhead is incurred in keeping all the parts working correctly**. That overhead, plus **contention for shared resources**, lowers the expected gain from additional processors.

More processors increase performance, but **coordination** overhead, shared-resource **contention**, and **sequential parts** prevent perfect N -fold speedup.

Symmetric Multiprocessing (SMP)



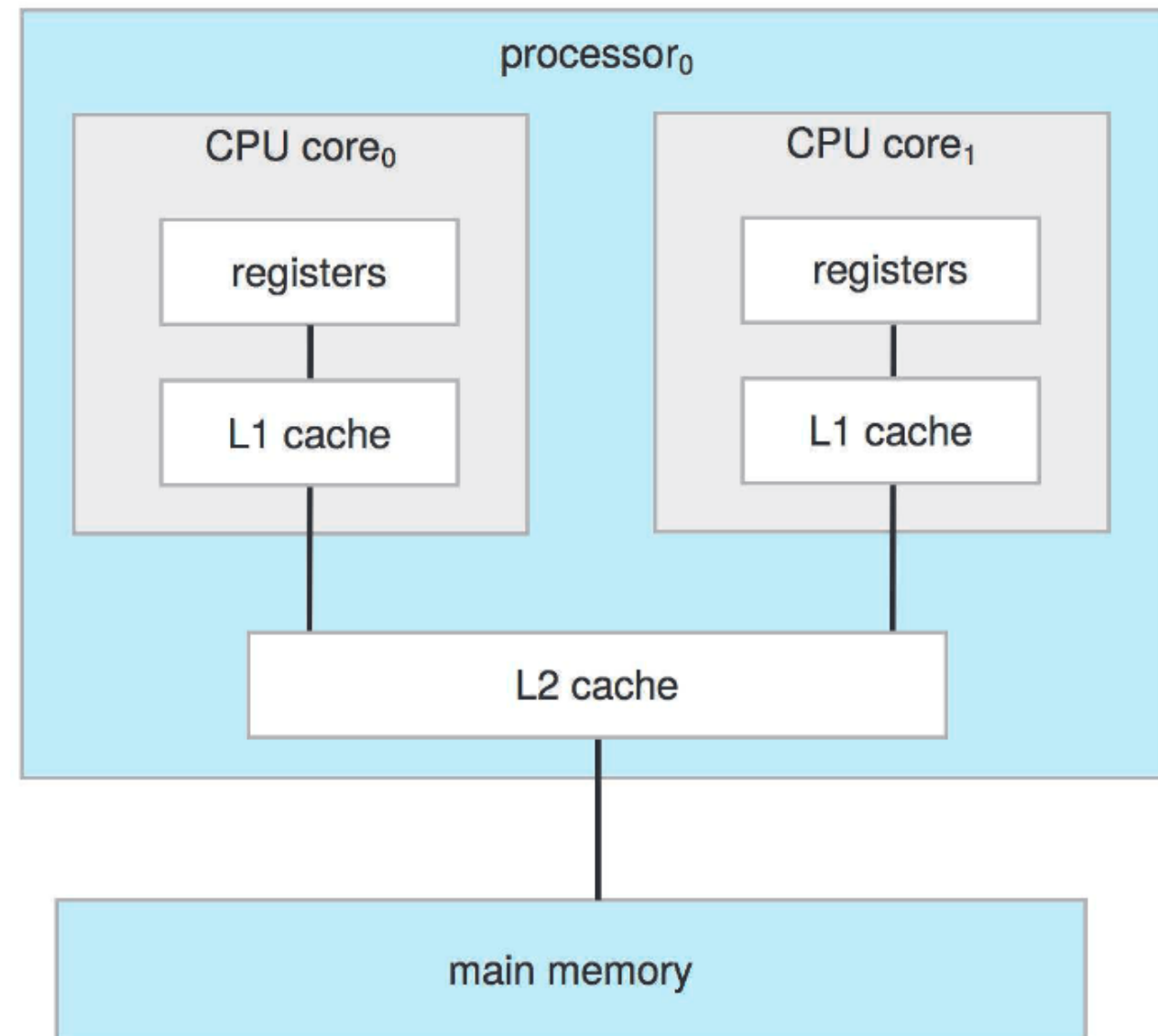
“**Symmetric**” means the processors are essentially **peers** (الأقران).

In **SMP**, each **peer CPU performs all tasks** — operating-system functions and user processes alike.

Each CPU has its **own registers** and a **private local cache**; all processors **share physical memory** over the system bus.

- **N processes can run on N CPUs** without significant deterioration — but **one CPU may sit idle while another is overloaded**.
- The **OS scheduler** can decide: “This process runs on Core 1; that process runs on Core 3”

Multicore Design: Two Cores on One Chip



Each core has its own **register set** and **local L1 cache** ; the **L2 cache** is local to the chip but **shared by both cores**.

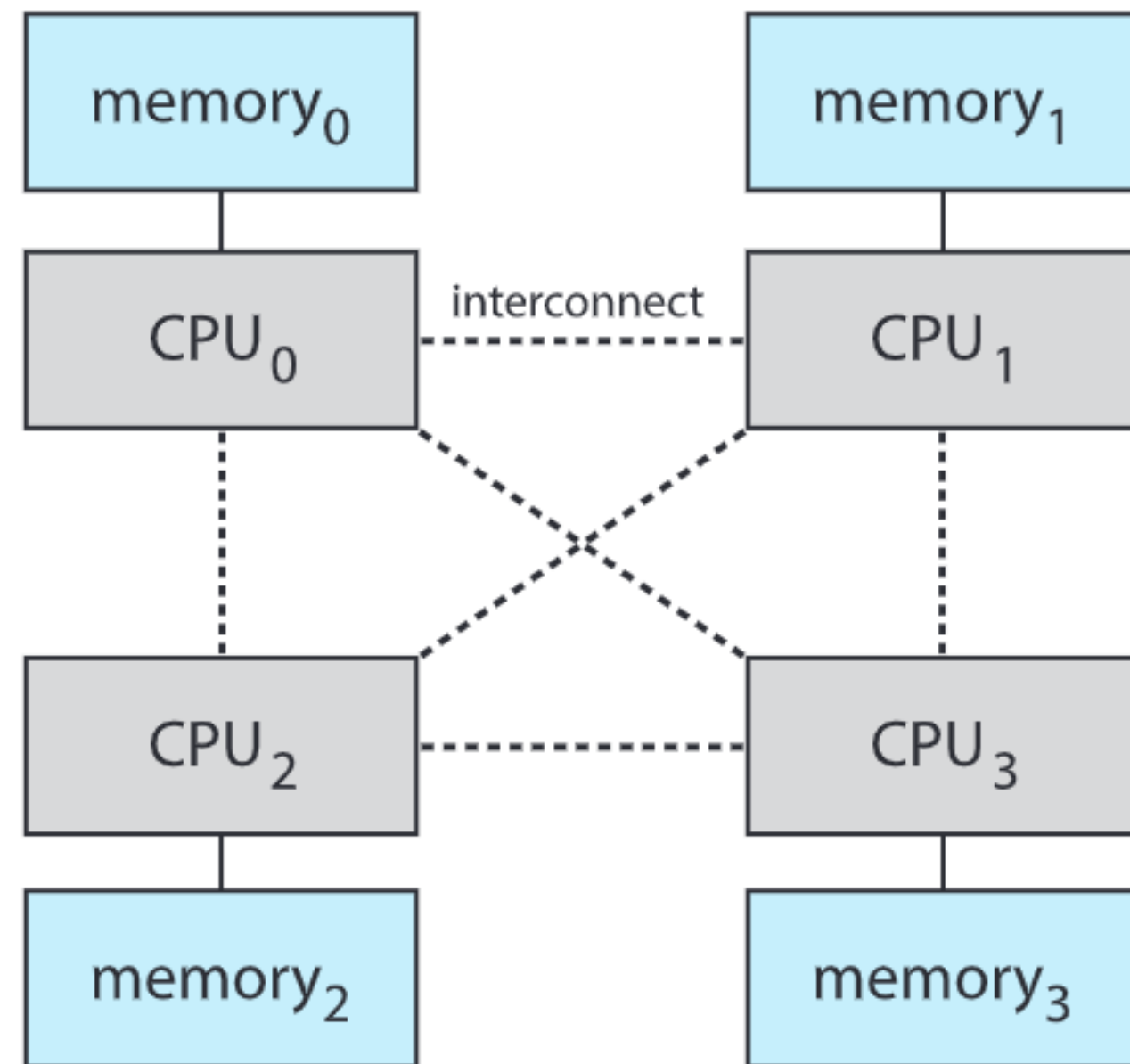
Local lower-level caches are **smaller and faster** than higher-level shared caches — most architectures **combine both**.

On-chip communication is faster than between chips, and one multicore chip uses **significantly less power** than multiple single-core chips.

A core is an **independent processing unit** inside a CPU that can fetch, decode, and execute instructions. **CPU chip = kitchen Core = chef**

Aside from cache, memory, and bus contention, an **N-core processor appears to the OS as N standard CPUs**.

Non-Uniform Memory Access



Adding CPUs to an SMP system eventually makes the **system bus a bottleneck**. Instead, give each CPU its **own local memory** on a small, fast local bus.

Advantage: local access is fast and creates **no contention** on the system interconnect, so NUMA **scales to large processor counts**.

Drawback: CPU 0 cannot reach CPU 3's memory as quickly as its own — a **latency penalty** the OS minimizes through careful **CPU scheduling and memory management**.

Memory access time is **non-uniform**. The OS tries to place a process and its memory close together to improve performance.

Blade Servers

Multiple **processor boards, I/O boards, and networking boards** placed in the **same chassis**.

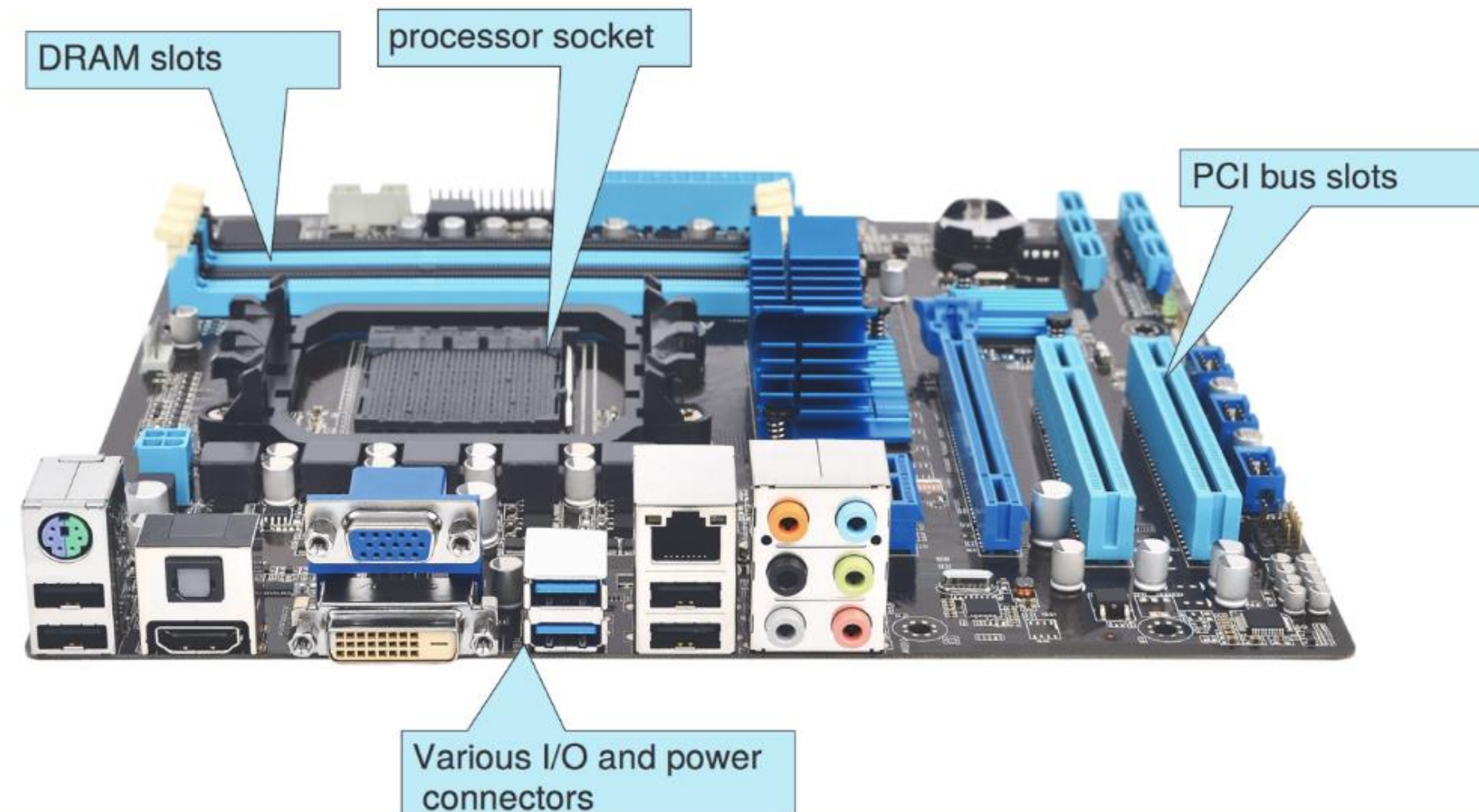
Some blade-server boards are **themselves multiprocessor**, which **blurs the lines between types of computers** . In essence, these servers consist of **multiple independent multiprocessor systems**.

The difference from a traditional multiprocessor system: **each blade-processor board boots independently and runs its own operating system** .



هو خادم مدمج على شكل لوحة رفيعة، يُركَّب مع عدة خوادم مشابهة داخل هيكل واحد مشترك (Chassis)، مع مشاركة الطاقة والتبريد والشبكة.

The PC Motherboard



A **processor socket** containing a CPU, **DRAM sockets**, **PCIe bus slots**, and I/O connectors of various types — a **fully functioning computer** once its slots are populated.

Even the **lowest-cost general-purpose CPU** contains **multiple cores**. Some motherboards contain **multiple processor sockets**.

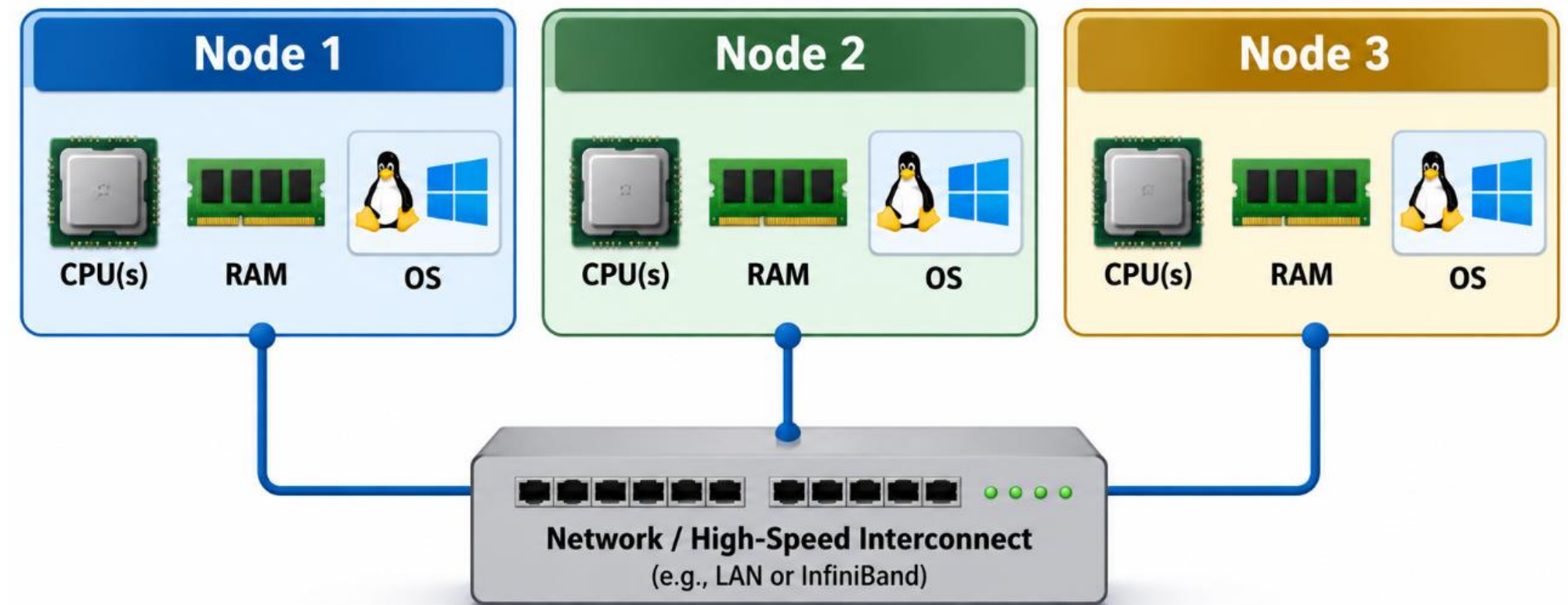
More advanced computers allow **more than one system board**, creating **NUMA systems**.

- **PCI/PCIe expansion slots** for devices such as network or graphics cards
- **Socket** = physical location for one CPU package

Clustered Systems

- A clustered system gathers together multiple CPUs — **nodes** — joined together.
- Each node is typically a **multicore system**.

loosely coupled (ارتباطاً فضفاضاً)
multiprocessor systems



The definition of *clustered* is **not concrete**. The generally accepted one: clustered computers **share storage** and are **closely linked via a local-area network (LAN)** or a faster interconnect such as **InfiniBand**.

Clustered Systems

	Multiprocessor	Cluster
Machines	One computer system	Multiple computers
CPUs	Multiple CPUs/cores	CPUs in each node
Memory	Usually shared or closely integrated	Usually separate per node
Communication	Memory/interconnect	Network/interconnect
Coupling	Tightly coupled	Loosely coupled
OS view	Usually one OS manages processors	Each node may run its own OS
Main uses	Performance, parallelism	Availability, scalability, HPC

High Availability, Graceful Degradation, Fault Tolerance

High availability

- The service remains available even when one or more nodes fail.
- Node A fails → cluster detects failure → Node B takes over → service continues
- High Availability = minimize service downtime.

Graceful degradation

The ability to **continue working after failures, but perhaps with reduced performance or reduced capacity.**

A restaurant has four chefs.
One chef goes home sick.
The restaurant remains open, but customers wait longer.

Fault tolerance

Suffering the **failure of any single component** and still continuing operation — requiring **failure detection, diagnosis, and correction** .

High availability keeps the service running, graceful degradation allows reduced performance after failures, and fault tolerance is the ability to detect, isolate, and survive component failures.

Asymmetric and Symmetric Clustering

ASYMMETRIC CLUSTERING

One machine runs the applications ; the other sits in hot-standby mode.

The hot-standby host **does nothing but monitor** the active server; it **becomes active if that server fails - failover**

Cost: hardware sits **idle** waiting for a failure.

One driver is driving the bus; the second driver sits beside him, ready to take over if needed.

SYMMETRIC CLUSTERING

Two or more hosts run applications and monitor each other.

All available hardware is in use , so the structure is more efficient.

Requirement: more than one application must be available to run.

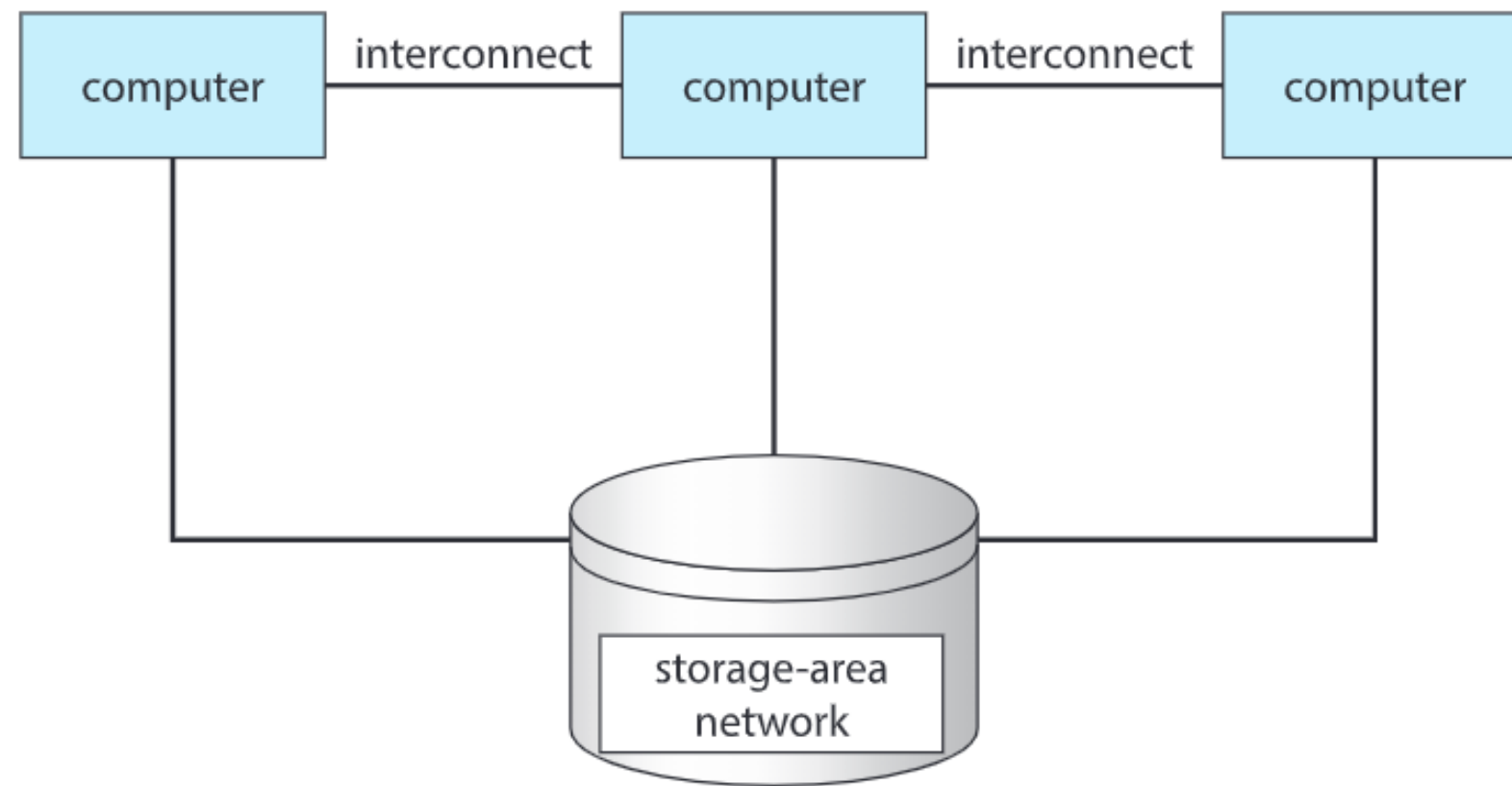
High-Performance Computing and Parallelization

Because a cluster consists of **several computer systems on a network**, it can supply **far greater computational power** than a single-processor or even an SMP system — running **one application concurrently on all computers in the cluster** .

The application **must be written specifically to take advantage of the cluster**. This involves **parallelization**: dividing a program into **separate components that run in parallel** on individual cores in a computer, or on computers in a cluster.

Cluster computing = divide the problem, solve parts in parallel on different nodes, then combine the results.

Parallel Clusters and Storage-Area Networks



- **Parallel clusters** let multiple hosts access the **same data on shared storage** . Because most operating systems lack support for **simultaneous access by multiple hosts**
- If several independent machines directly write to the same storage without cluster-aware coordination, they may have **inconsistent caches or metadata**.
- they require **special versions of software** — Oracle Real Application Cluster, for example.

Storage-area networks (SANs) let many systems attach to a **pool of storage** , so cluster software can assign an application to **any attached host** — and **any other host can take over on failure** .

A **distributed lock manager (DLM)** supplies the **access control and locking** that keeps **conflicting operations** from occurring - DLM prevents conflicting operations from happening at the same time.

What to Take from Section 1.3

- 01 Systems are classified by the **number of general-purpose processors**; special-purpose processors **do not count**.
- 02 Multiprocessor systems raise **throughput**, but the speed-up with **N processors is less than N**.
- 03 **SMP** shares memory over a bus; **multicore** chips add local L1 and shared L2 caches; **NUMA** trades remote-access latency for scalability.
- 04 **Clusters** are **loosely coupled nodes sharing storage** — used for **high availability** and for **high-performance computing** .
- 05 **Next:** Section 1.4, Operating-System Operations.