04-Introduction to Distributed Database Management System

School of Computer Science
University of Windsor

Submission Deadlines

- **Lab 2**: Next week
- **Assignment 2 (Hadoop certificate):**
  - Already published in Brightspace
  - Certificate submission: Sec 2: Feb 13; Sec 3: Feb 14; Sec 4: Feb 15

Agenda

- Lecture
  - System Architectures
  - Introduction to Distributed Databases
  - Advantages and Disadvantages of DDBMSs
  - Architectures of a DDBMS
- Assignment 2 Quiz
- Project Proposal Presentations

Introductory Questions

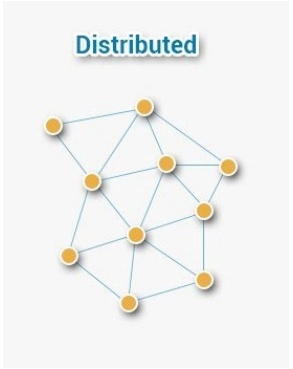
Why Do You Want Distributed Databases?

How DDBMS's architecture differ from Centralized DBMS's?

What are the different System Architectures?

What are the challenges with DDBMS?

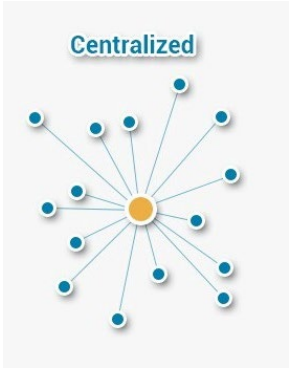
Distributed Databases



Consists of two or more files distributed across different computers/nodes/servers, at different sites on a network. They need to be synched.

- ✓ Faster data access
- ✓ No interference with each others' data when accessing or manipulating data
- ✓ Fault-tolerance
- x Time is required in synchronization of multiple databases

V S



Centralized Databases

A single database located at one site on a network. .

- ✓ A complete view of data
- ✓ Easier to manage, update data
- x Can cause bottlenecks when multiple users are accessing simultaneously

https://medium.com/@srk7official1998/what-is-memcache-3833a1d54256

Why You Want Distributed Databases?

- **Scalability**:
  If data volume, read load, or write load grows bigger than a single machine can handle, you can potentially spread the load across multiple machines.

- **Fault tolerance/high availability:**
  You can use multiple machines to give you redundancy. When one fails, another one can take over.

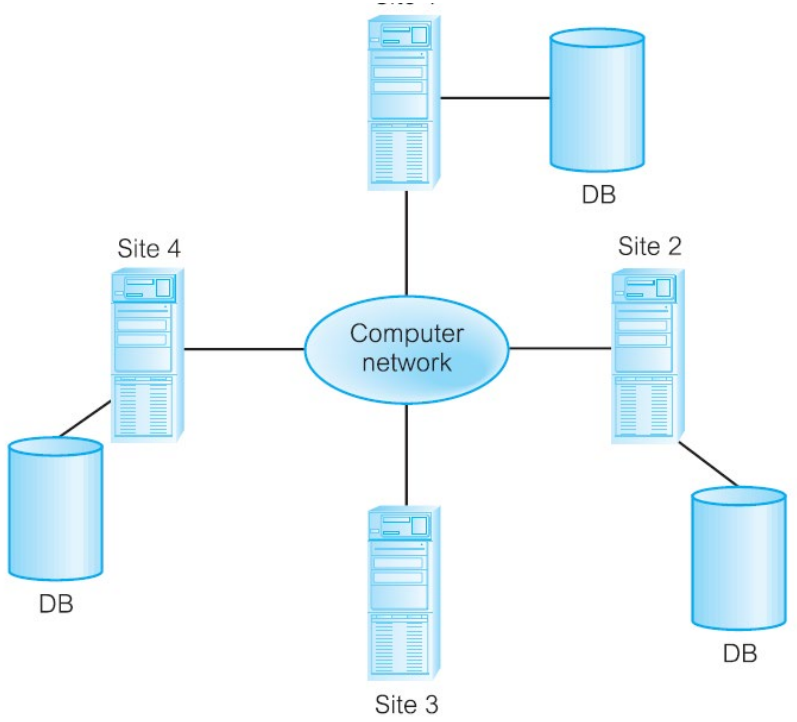
- **Latency**:
  Each user can be served from a datacenter that is geographically close to them.

Distributed database management system.

A distributed database management system (DDBMS) consists of a single logical database that is split into a number of fragments. Each fragment is stored on one or more computers (replicas) under the control of a separate DBMS, with the computers connected by a communications network.

A DDBMS therefore has the following characteristics:

- a collection of logically related shared data;
- data split into a number of fragments;
- fragments may be replicated;
- fragments/replicas are allocated to sites;
- sites are linked by a communications network;
- data at each site is under the control of a DBMS;
- DBMS at each site can handle local applications, autonomously;
- each DBMS participates in at least one global application.



Functions of a DDBMS

Use the same concept in single node DBMSs to support transaction processing and query execution in distributed environments.

Query optimization & Execution, Concurrency Control, Logging & Recovery, Communication services & Security control

In addition, we expect a DDBMS to have the following functionality:

- ✓ Extend communication services to provide access to remote sites and allow to transfer of queries and data among the sites using a network;
- ✓ Extend system catalog to store data distribution details;
- ✓ Distribute query processing, including query optimization and remote data access;
- ✓ Extend security control to maintain appropriate authorization/access privileges to the distributed data;
- ✓ Extend concurrency control to maintain consistency of distributed and possibly replicated data;
- ✓ Extend recovery services to take account of failures of individual sites and the failures of communication links.

Advantages and Disadvantages of DDBMSs

ADVANTAGES

- ✓ Reflects organizational structure
- ✓ Improved shareability and local autonomy
- ✓ Improved availability
- ✓ Improved reliability
- ✓ Improved performance
- ✓ Economics
- ✓ Modular growth
- ✓ Integration
- ✓ Remaining competitive

DISADVANTAGES

- ✓ Complexity
- ✓ Maintenance Cost
- ✓ Security
- ✓ Integrity control more difficult
- ✓ Lack of standards
- ✓ Lack of experience
- ✓ Database design more complex

Homogeneous and Heterogeneous DDBMSs

Homogenous DDBMS: Those database systems which execute on the same operating system and use the same application process and carry the same hardware devices

- ✓ Much easier to design, manage and use.
- ✓ Appears to users as a single system
- x Difficult to force a homogenous environment

Heterogenous DDBMS: Those database systems which execute on different operating systems under different application procedures, and carries different hardware devices. Data may be required from another site that may have:

- ✓ different hardware;
- ✓ different DBMS products;.
- x Difficult to design and manage
- x Translations are required to allow communication between different DBMSs.

Example of a Distributed DBMSs

Examples: Google uses BigTable, Spanner, Google Cloud SQL, MySQL, Dremel, Millwheel, Firestore, Memorystore Firebase, Cloud Dataflow, BigQuery & many more.

"**Google** runs on hundreds of thousands of servers—by one estimate, in excess of 450,000—racked up in thousands of clusters in dozens of data centers around the world. It has data centers in Dublin, Ireland; in Virginia; and in California, where it just acquired the million-square-foot headquarters it had been leasing. Google runs more than 2,000,000 servers in 36 data centers around the globe today"

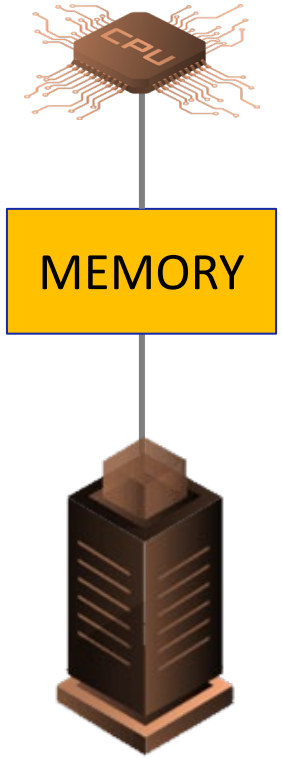
History of Distributed DBMS

INGRES –Early 1970s (UC Berkeley)
MUFFIN – 1979 (University of California-Berkeley)
SDD (System for Distributed Databases)-1 1979 (Computer Corporation of America (CCA))
System R*- 1984 (IBM Research)
Gamma – 1986 (University of Wisconsin)
Client/server – Early 1980s
Cloud computing – Early 2000
- Infrastructure-as-a-service (Iaas)
- Platform-as-a-service (Paas)
- Software as- a-service (Saas)

System Architectures

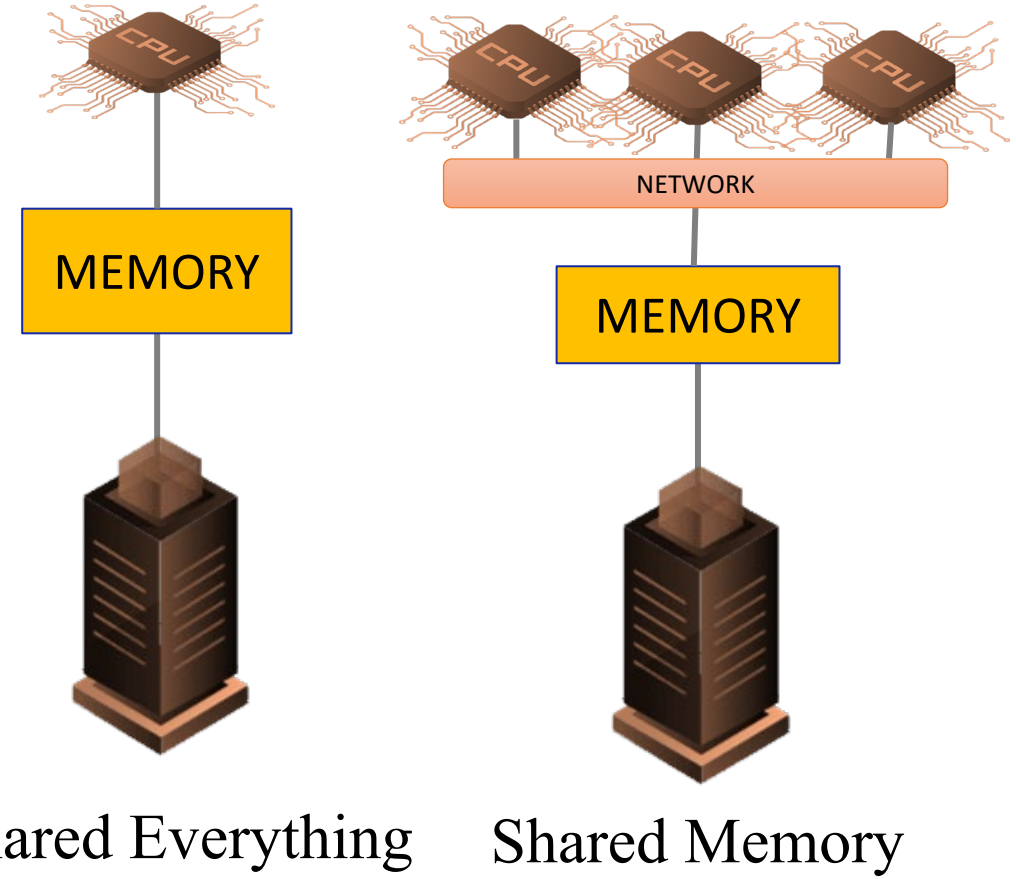
A DBMS's system architecture specifies what shared resources are directly accessible to CPUs.

System Architectures



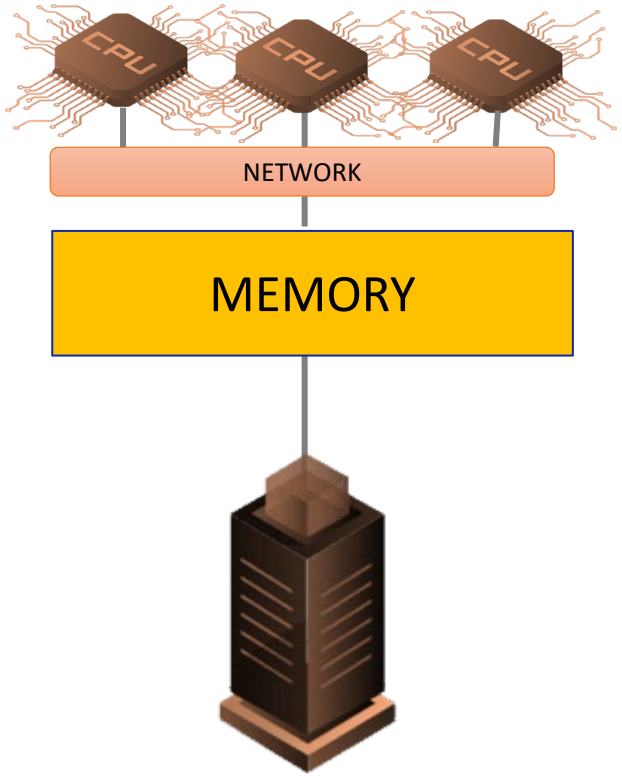
Shared Everything

System Architectures: Scaling to Higher Load



Shared Everything Shared Memory

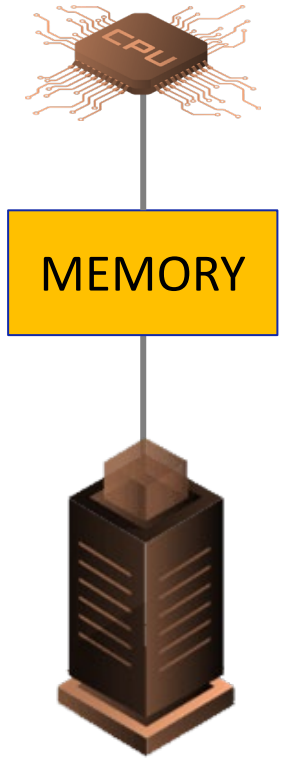
Shared Memory



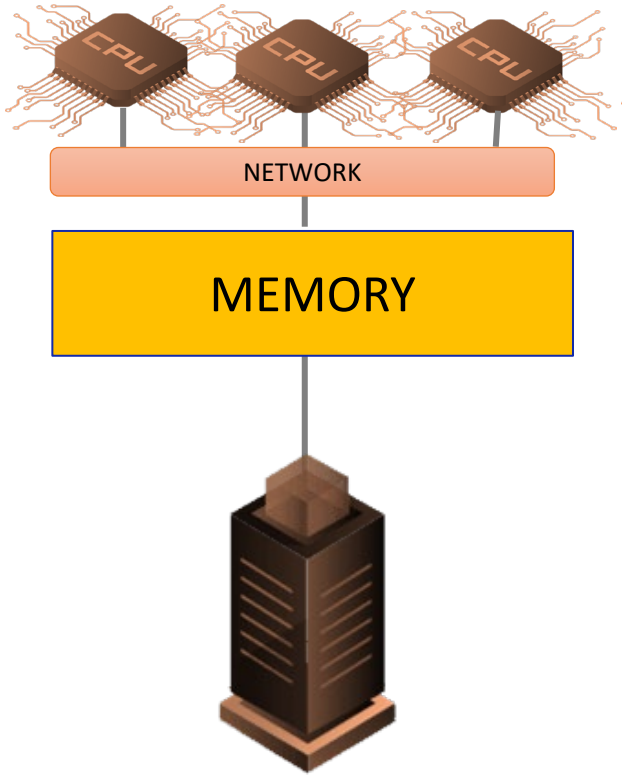
Shared Memory

- All the CPUs can access the same memory and are all controlled by a single operating system.
- A fast interconnection network allows any processor to access any part of the memory in parallel.
- Advantages: simplicity and load balancing.
- Problems: cost, limited extensibility and low availability.

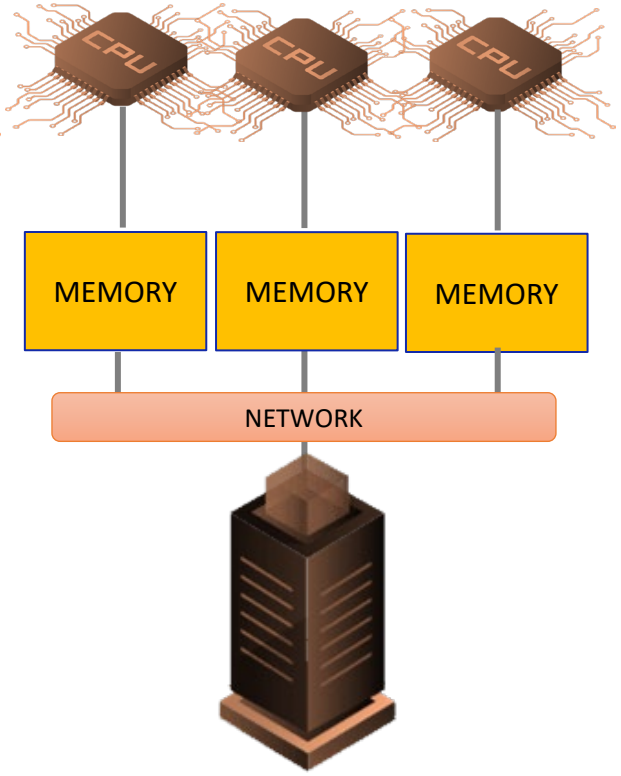
System Architectures: Scaling to Higher Load



Shared Everything

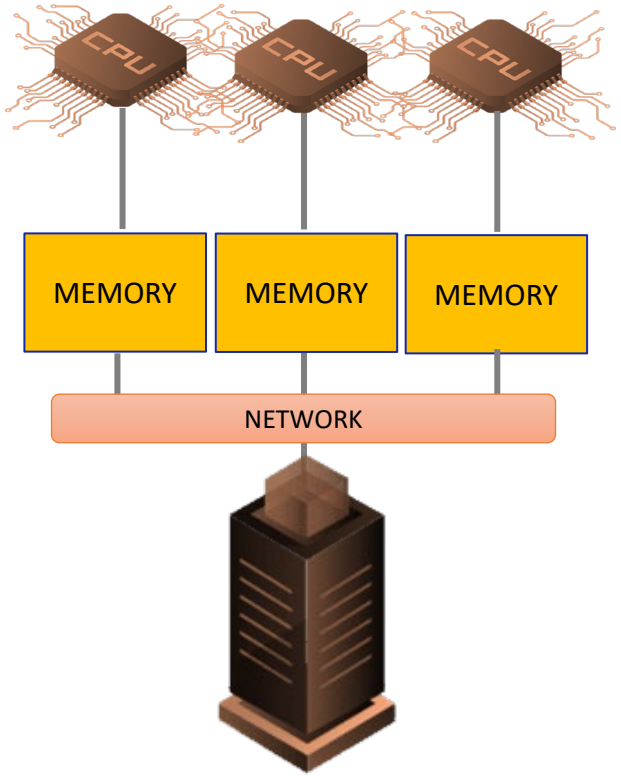


Shared Memory



Shared Disk

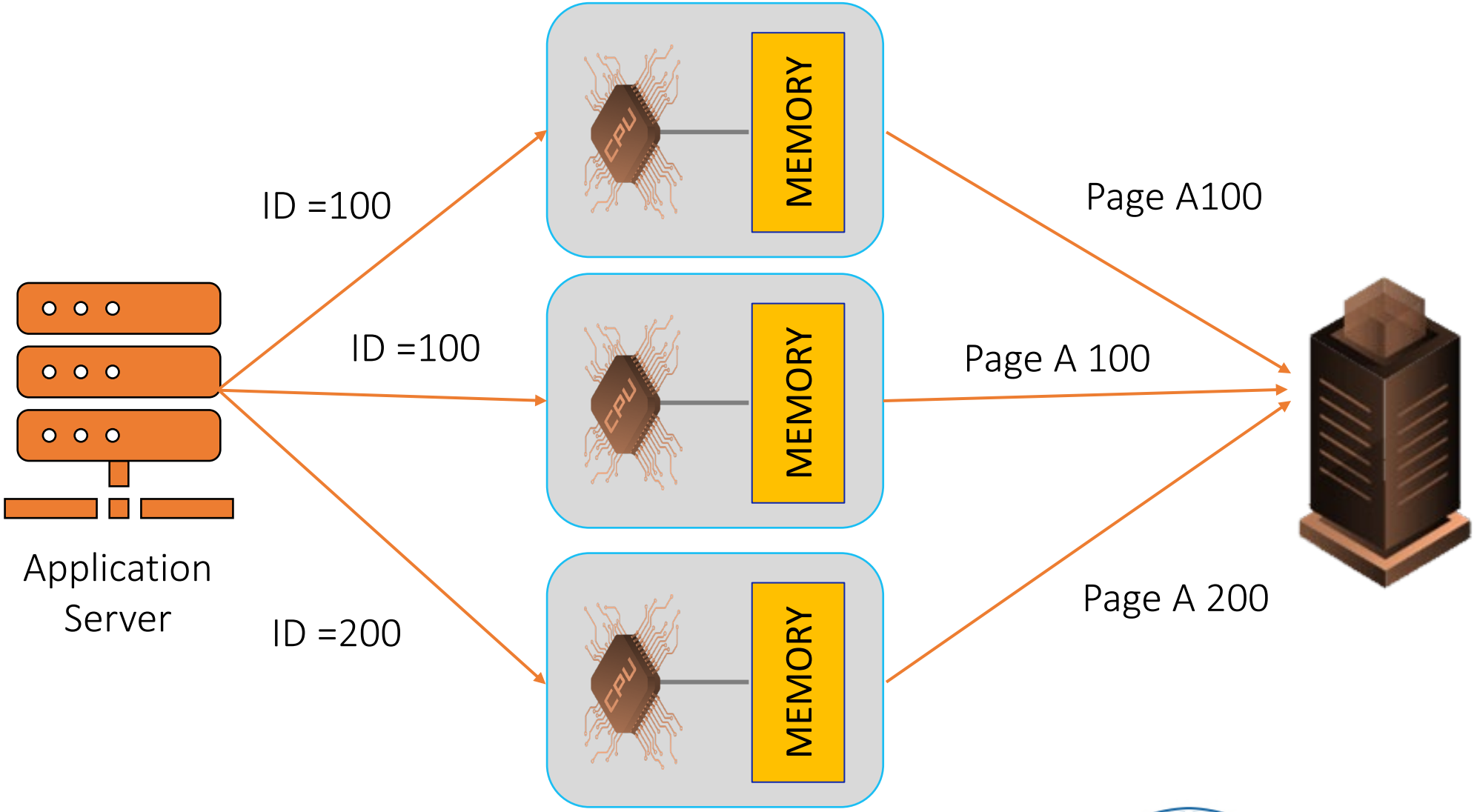
Shared Disk



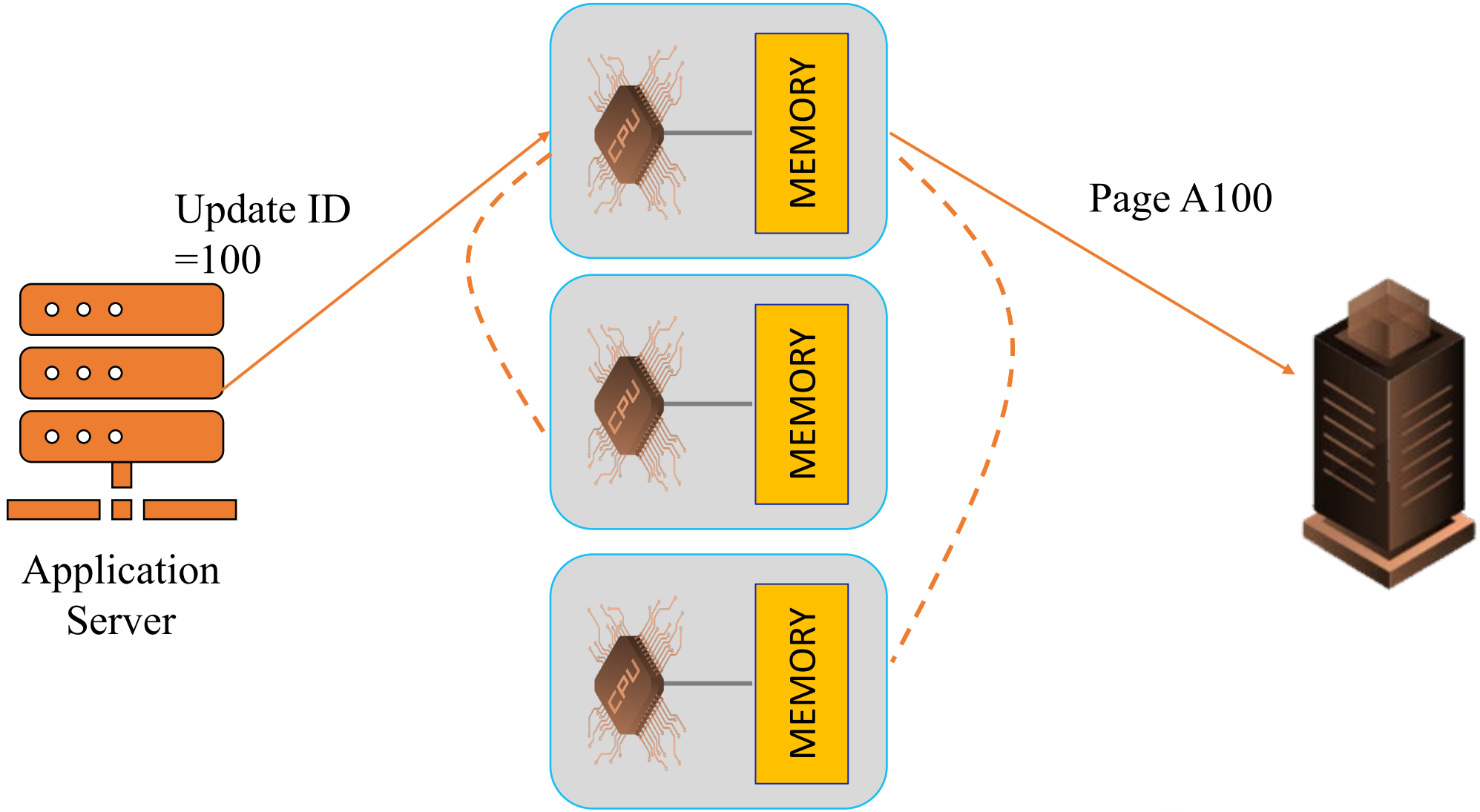
Shared Disk

- ✓ All CPUs can access a single logical disk directly via an interconnect network, but each have their own private memories.
- ✓ Disk sharing architecture requires suitable lock management techniques to control the update concurrency control.
- ✓ This is the present architecture in today's cloud environment.- The most notable parallel database system which uses shared-disk is Oracle.
- ✓ Advantages: lower cost, good extensibility, availability, load balancing, and easy migration from centralized systems.
- ✓ Problems: complexity and potential performance problems

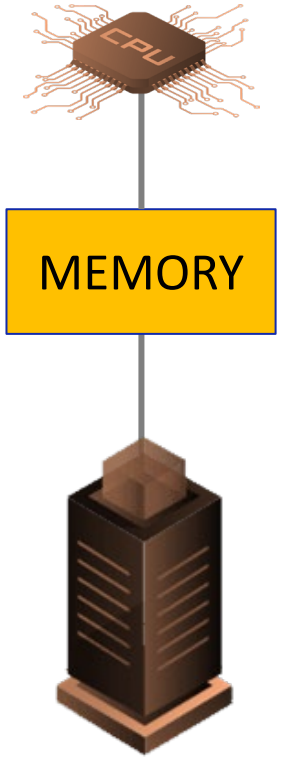
Shared Disk Example



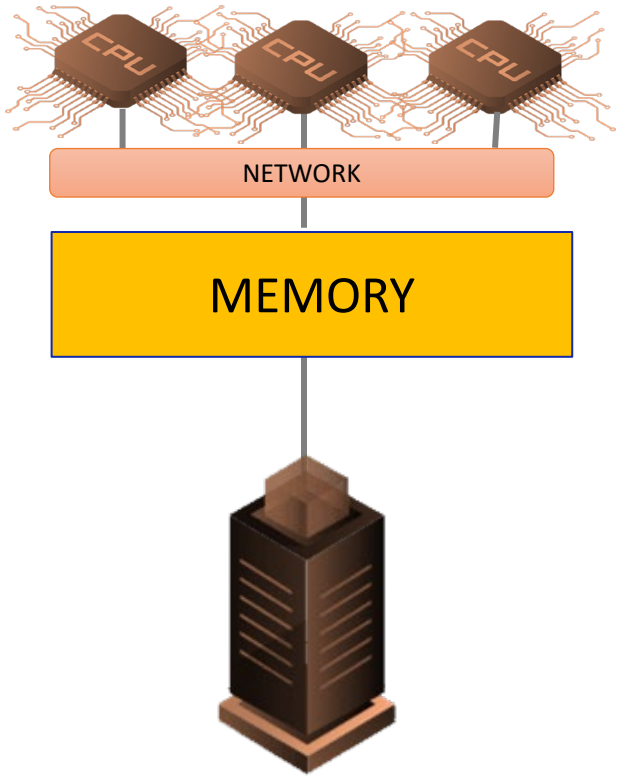
Shared Disk Example



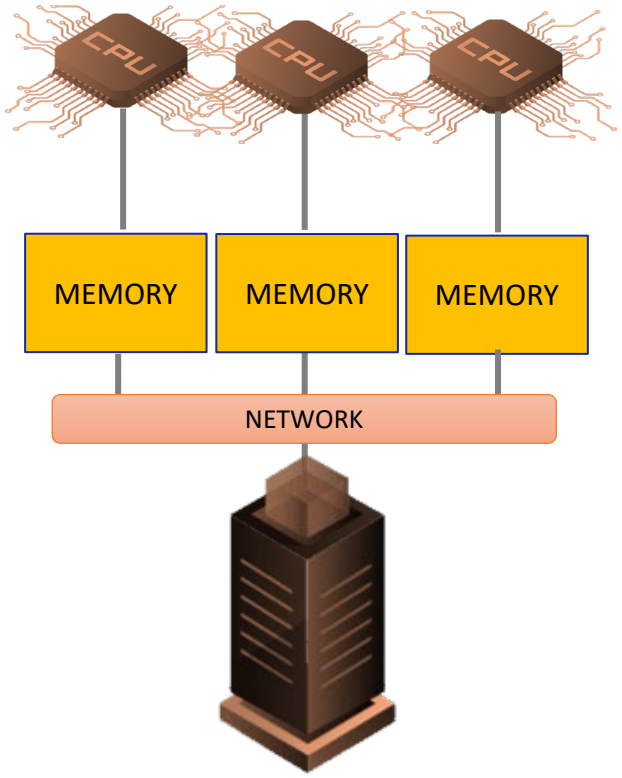
System Architectures: Scaling to Higher Load



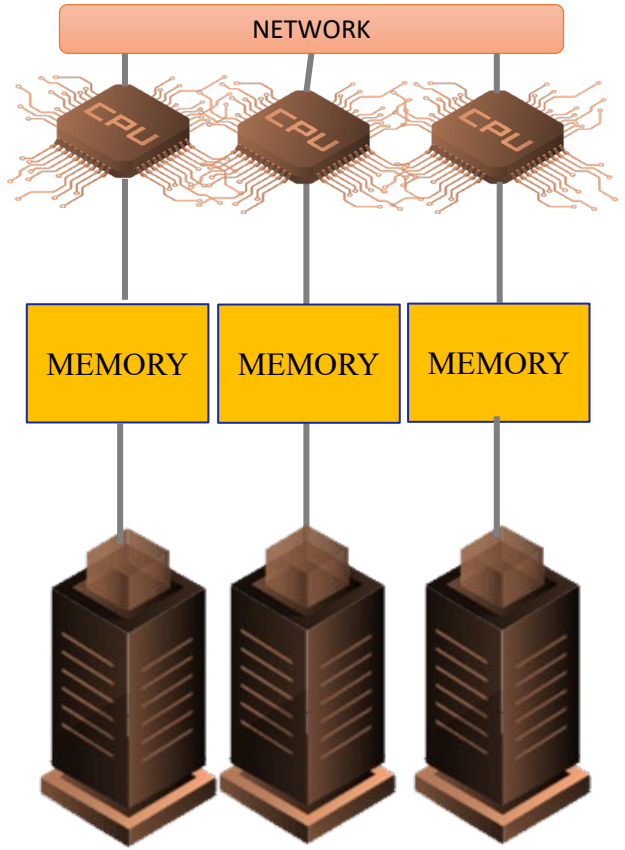
Shared Everything



Shared Memory

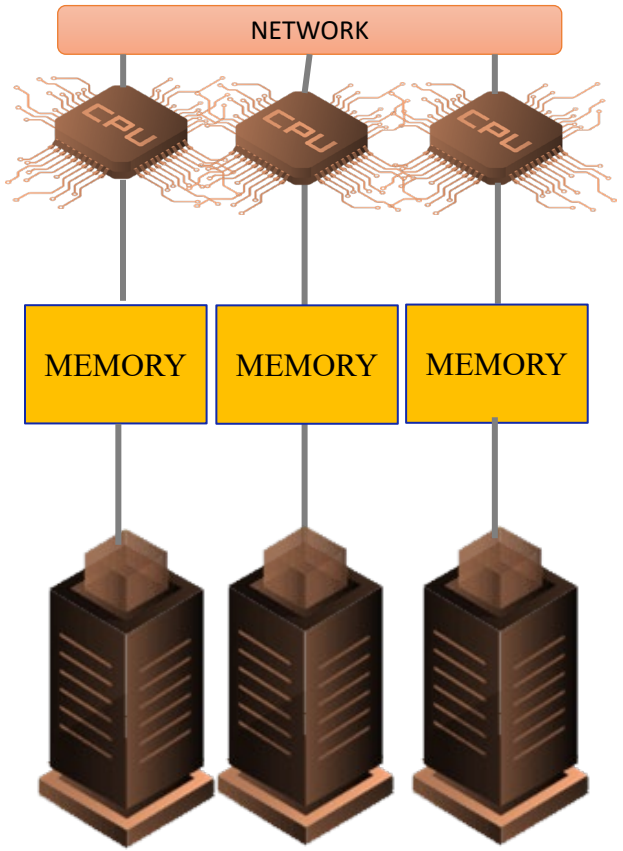


Shared Disk



Shared Nothing

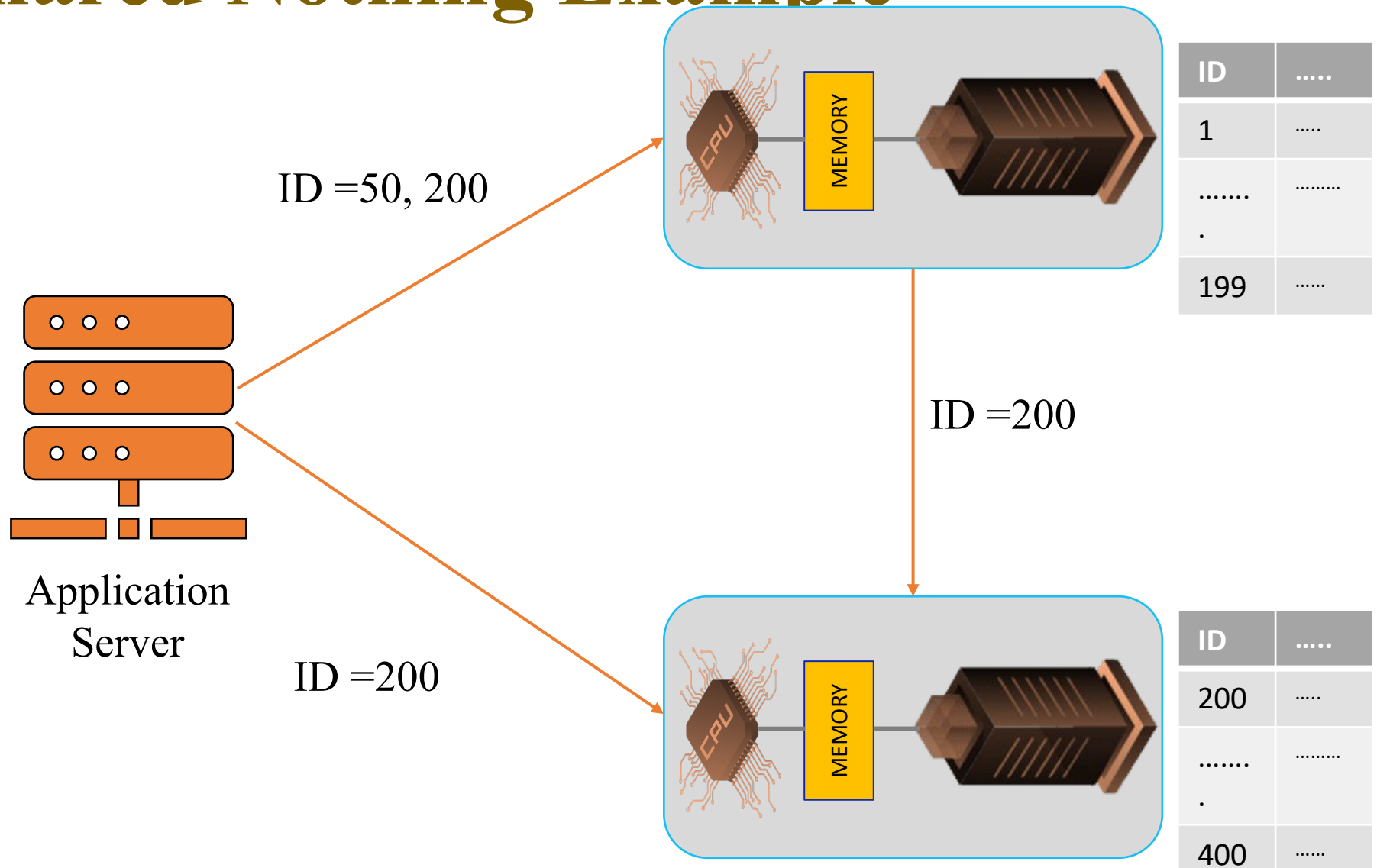
Shared Nothing



Shared Nothing

- ✓ Each node uses its CPUs, RAM, and disks independently.
- ✓ Any coordination between nodes is done at the software level, using a conventional network.
- ✓ No special hardware is required by a shared-nothing system.
- ✓ Advantages: Better performance, better efficiency, low cost, high extensibility, and high availability.
- ✓ Problem: Hardest to ensure consistency, add capacity, and Higher complexity

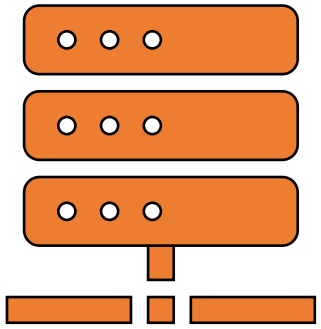
Shared Nothing Example



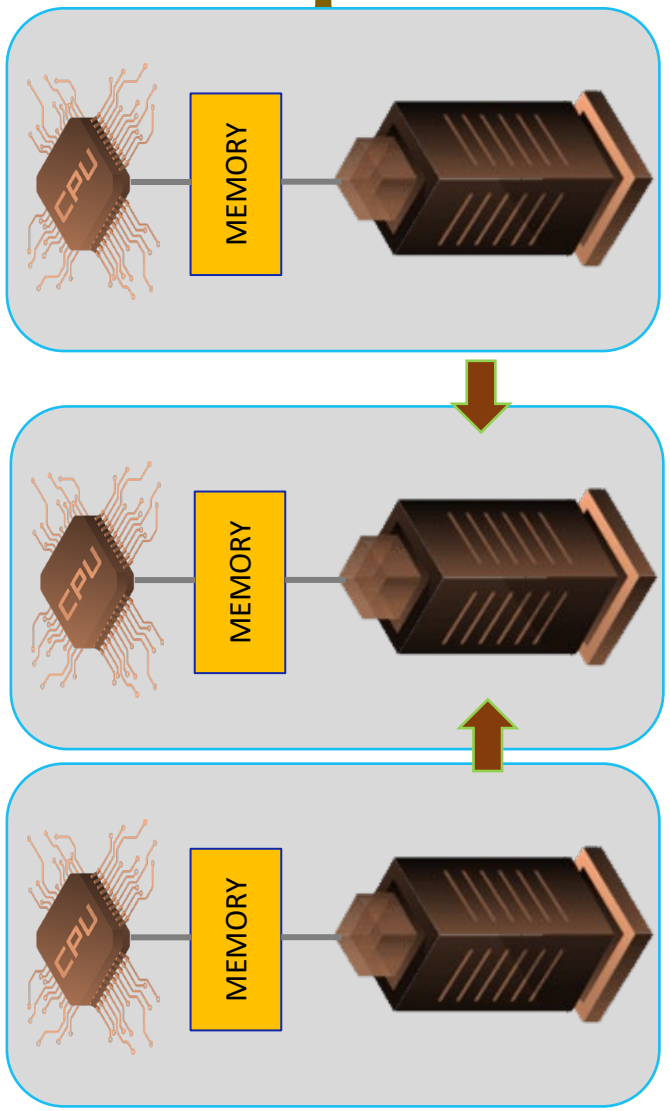
ID	.....
1	.....
..... .	.....
199	.....

ID	.....
200	.....
..... .	.....
400	.....

Shared Nothing Example



Application Server



ID	.....
1	.....
..... .	.....
199	.....

ID	.....
151	.....
..... .	.....
250	.....

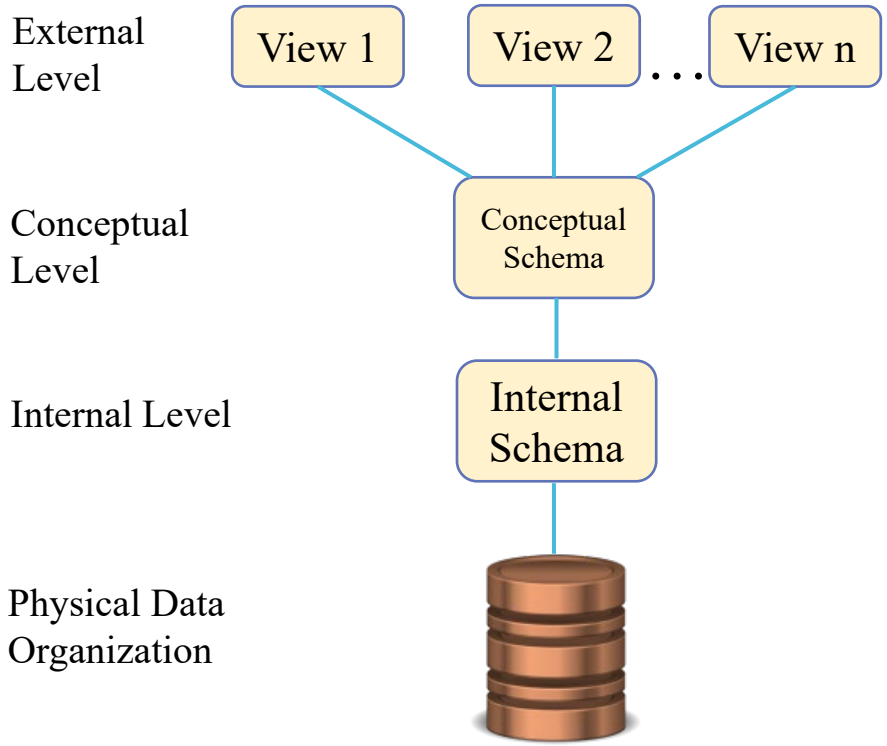
ID	.....
200	.....
..... .	.....
400	.....

Design Issues

- How to find data?
  - Tradeoff between availability and consistency
- How to execute queries:
  - Push and Pull mechanism

Architectures of a DDBMS

Due to diversity of distributed DBMSs, there is no accepted architecture equivalent to ANSI/SPARC 3-level architecture.
However, it may be useful to present one possible reference architecture that addresses data distribution.

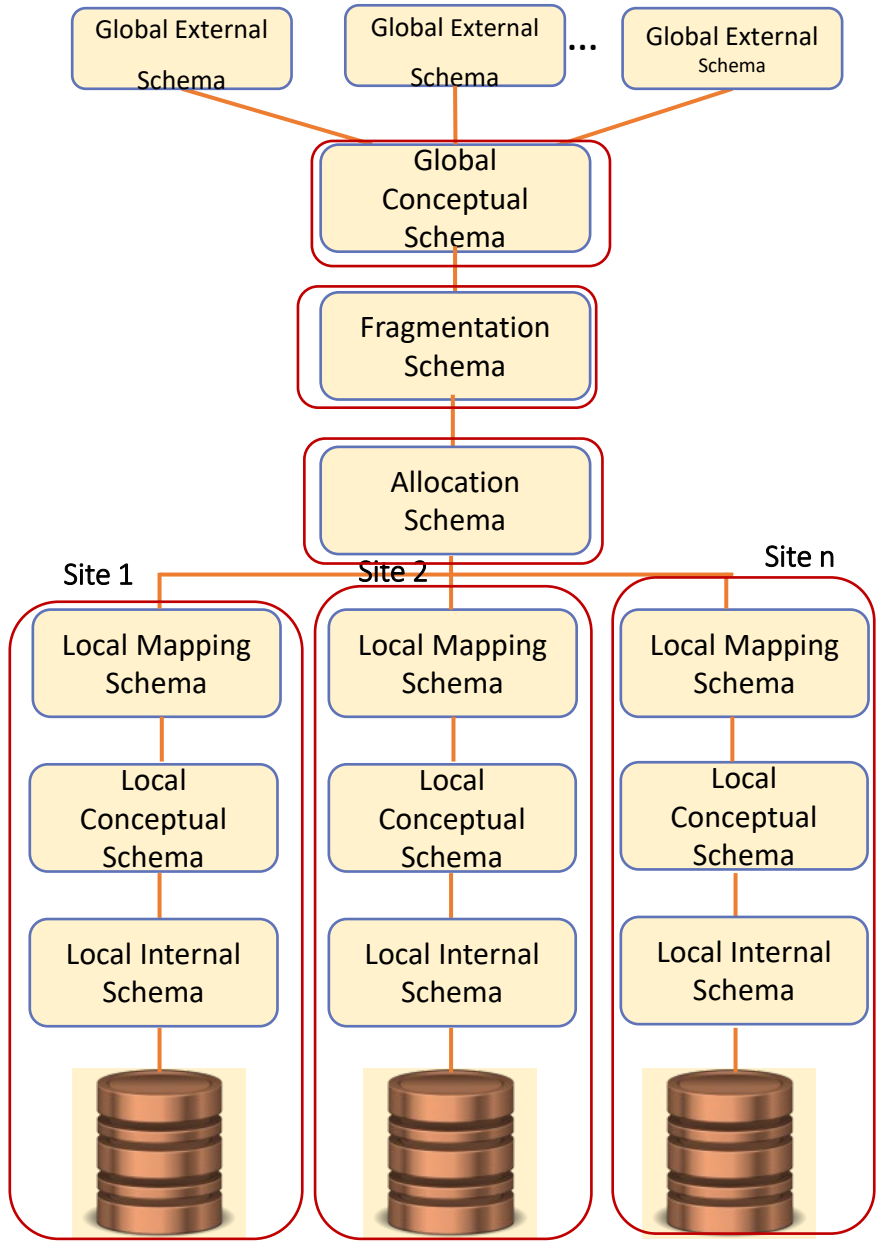


ANSI/SPARC 3-level architecture

Architectures of a DDBMS

Due to diversity of distributed DBMSs, there is no accepted architecture equivalent to ANSI/SPARC 3-level architecture. However, it may be useful to present one possible reference architecture that addresses data distribution.
Reference Architecture for a DDBMS consists of the following schemas:

- ✓ Set of global external schemas.
- ✓ Global conceptual schema (GCS).
- ✓ Fragmentation schema and allocation schema.
- ✓ Set of schemas for each local DBMS conforming to 3-level ANSI/SPARC.



Summary

Major different between DDBMS and Centralized DBMS.

Various type of system architectures: Shared Everything, Shared Memory, Shared Disk and Shared Nothing

Define DDBMS: the software that manages a collection of multiple, logically interrelated databases located at the nodes of a distributed system.

Major Advantages and Disadvantages of DDBMSs.

Homogeneous and Heterogeneous DDBMSs.

Architectures of a DDBMS

Any Questions