DATABASE

CSE3103 : Database
FALL 2020

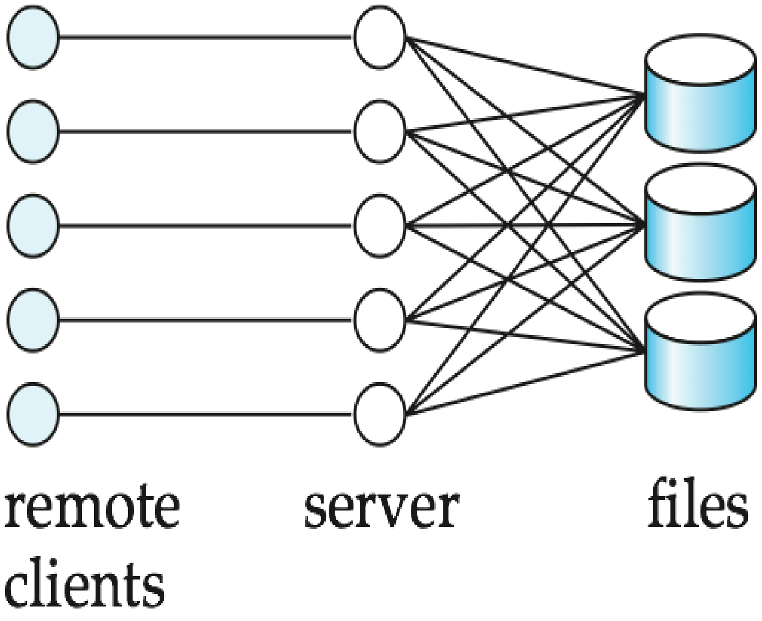
Nazmus Sakib
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Transaction Processing Monitors

- **Transaction Processing (TP) monitors** initially developed as multithreaded servers to support large numbers of terminals from a single process.
- Provide infrastructure for building and administering complex transaction processing systems with a large number of clients and multiple servers.
- Provide services such as:
  - Presentation facilities to simplify creating user interfaces
  - Persistent queuing of client requests and server responses
  - Routing of client messages to servers
  - Coordination of two-phase commit when transactions access multiple servers.

TP Monitor Architectures

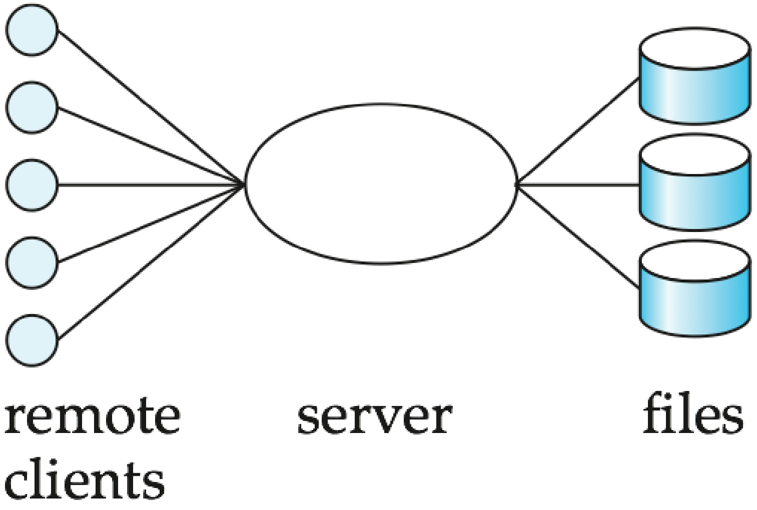
- **Process per client model** - instead of individual login session per terminal, server process communicates with the terminal, handles authentication, and executes actions.
  - Memory requirements are high
  - Multitasking- high CPU overhead for context switching between processes



(a) Process-per-client model

TP Monitor Architectures

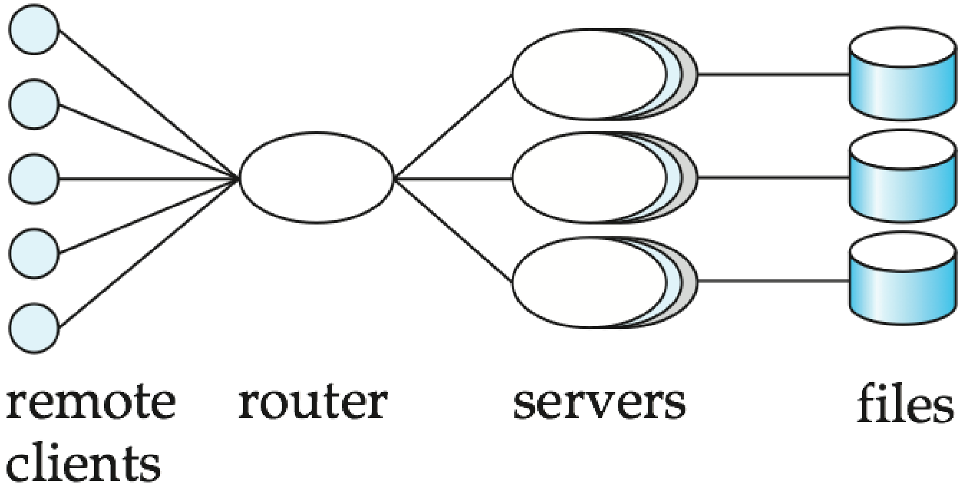
- **Single process model** - all remote terminals connect to a single server process.
  - Used in client-server environments
  - Server process is multi-threaded; low cost for thread switching
  - No protection between applications
  - Not suited for parallel or distributed databases



(b) Single-server model

TP Monitor Architectures

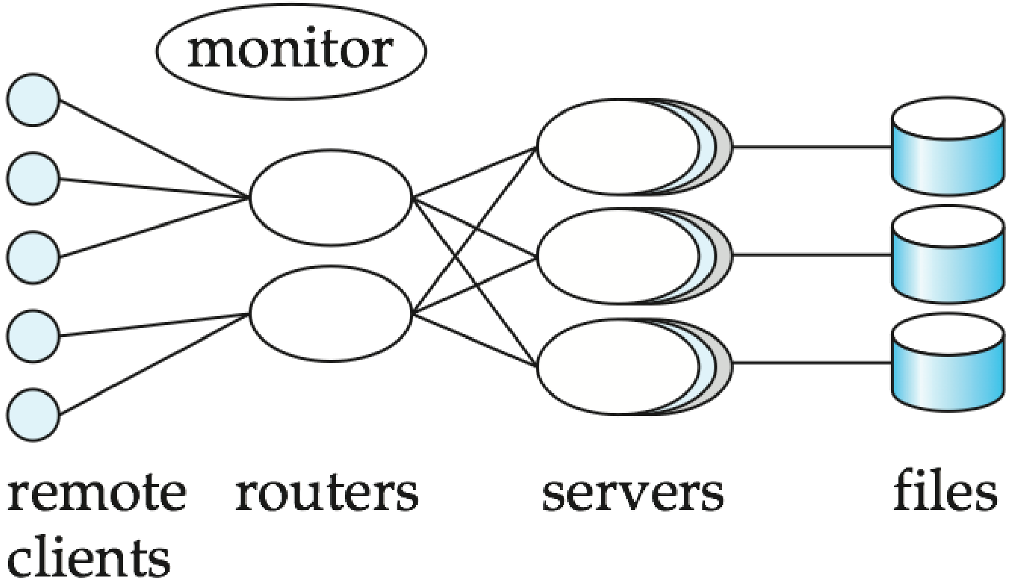
- **Many-server single-router model** - multiple application server processes access a common database; clients communicate with the application through a single communication process that routes requests.
  - Independent server processes for multiple applications
  - Multithread server process
  - Run on parallel or distributed database



(c) Many-server, single-router model

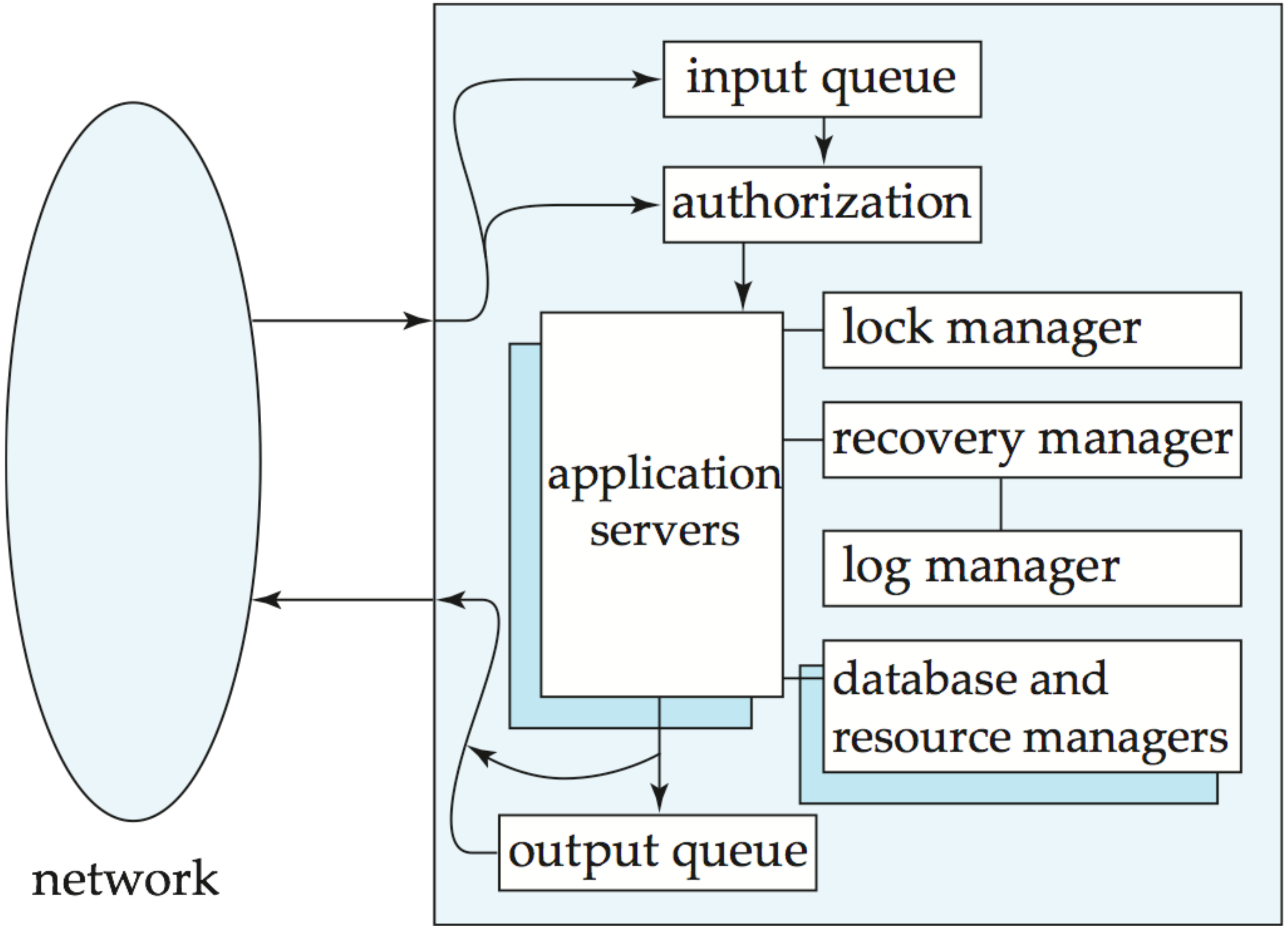
TP Monitor Architectures

- **Many server many-router model** - multiple processes communicate with clients.
  - Client communication processes interact with router processes that route their requests to the appropriate server.
  - Controller process starts up and supervises other processes.



(d) Many-server, many-router model

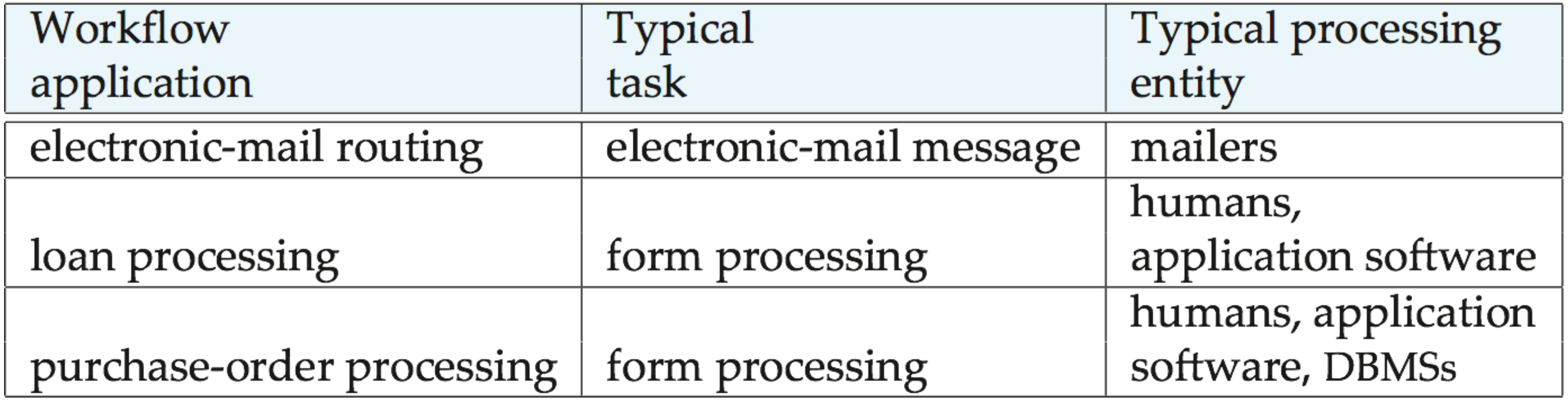
Detailed Structure of a TP Monitor



Transactional Workflows

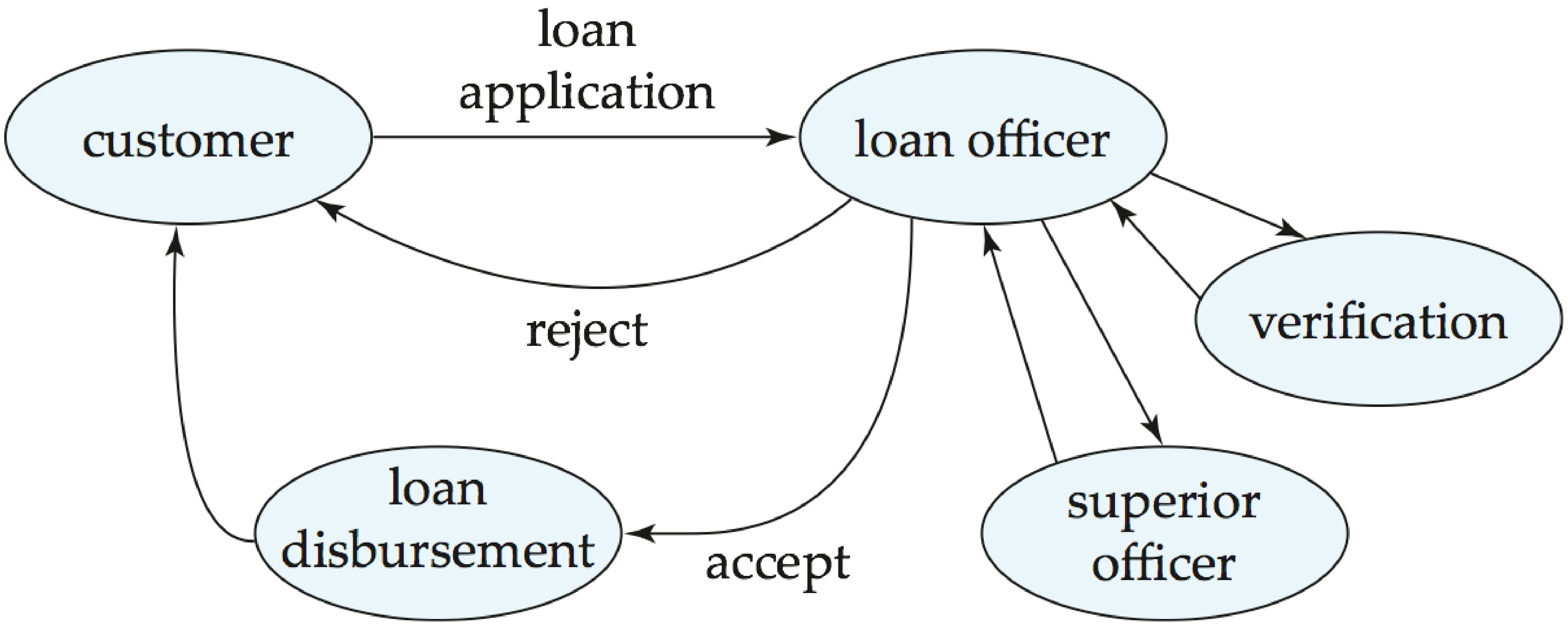
- **Workflows** are activities that involve the coordinated execution of multiple tasks performed by different processing entities.
- With the growth of networks, and the existence of multiple autonomous database systems, workflows provide a convenient way of carrying out tasks that involve multiple systems.
- Example of a workflow delivery of an email message, which goes through several mails systems to reach destination.
  - Each mailer performs a tasks: forwarding of the mail to the next mailer.
  - If a mailer cannot deliver mail, failure must be handled semantically (delivery failure message).
- Workflows usually involve humans: e.g. loan processing, or purchase order processing.

Examples of Workflows



Workflow application	Typical task	Typical processing entity
electronic-mail routing	electronic-mail message	mailers
loan processing	form processing	humans, application software
purchase-order processing	form processing	humans, application software, DBMSs

Loan Processing Workflow



- In the past, workflows were handled by creating and forwarding paper forms
- Computerized workflows aim to automate many of the tasks. But the humans still play role e.g., in approving loans.

Transactional Workflows

- Must address following issues to computerize a workflow.
  - Specification of workflows - detailing the tasks that must be carried out and defining the execution requirements.
  - Execution of workflows - execute transactions specified in the workflow while also providing traditional database safeguards related to the correctness of computations, data integrity, and durability.
  - E.g.: Loan application should not get lost even if system fails.
- Extend transaction concepts to the context of workflows.
- State of a workflow - consists of the collection of states of its constituent tasks, and the states (i.e., values) of all variables in the execution plan.

DBMS
data
database
distributed
parallel
shared
system
systems
architecture
Distributed
DDBMS
DBMSs
memory
network
processing
performance
processors
processor
access
centralized
multiple
sites
site
Parallel
may
major
fact
fragments
typically
across
reliability
eliminates
distribution
communication
also
many
replicated
discuss
make
provide
local
sometimes
computing
one
single
split
order
scalable
architectures
referred
operational
decentralized
directly
com
Date
disks
large
islands
resources
remote
defined
controlled
based
often
Management
number
dis
user
per
makes
incompatible
structure
collection
provides
global
way
logically
consists
associated
stored
disk
partitioned
used
computer
least
must
System
technology
capable
nodes
definition
development
bottleneck
administered
mode
organizational
interconnection
Figure
transparent
management
application
nothing
Shared
databases
share
logical
distinction
improve
approach
scalability
control
computers
applications
Database
like
machines
users
within
physically
require
independently
organizations
communications
possibly
requested
solely
maintained
creates
autonomy
invisible
microprocessors
using
promotes
shows
located
electronic
inter
workload
vendors
regarded
projects
consider