Chapter 5
Transaction Processing Monitors: An Overview

Mijin An
meeeeejin@gmail.com

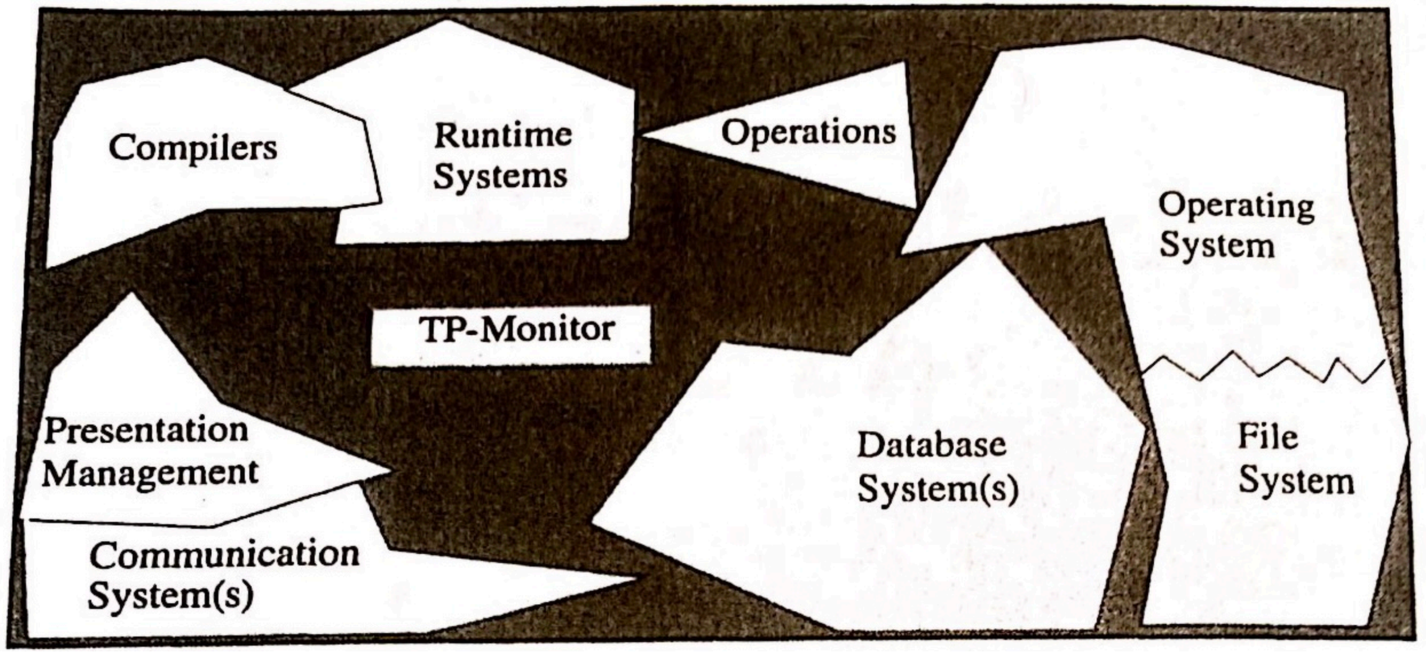
VLDB Lab.

Introduction

- The book says: There is no commonly accepted definition of
  - What a TP monitor is
  - How it interfaces to other system components
  - Whether it is really needed

- Today's **TP monitor coordinates the flow of transaction** between the client processes that issue requests and the back-end servers that process them

The Role of TP Monitors in Transaction Systems



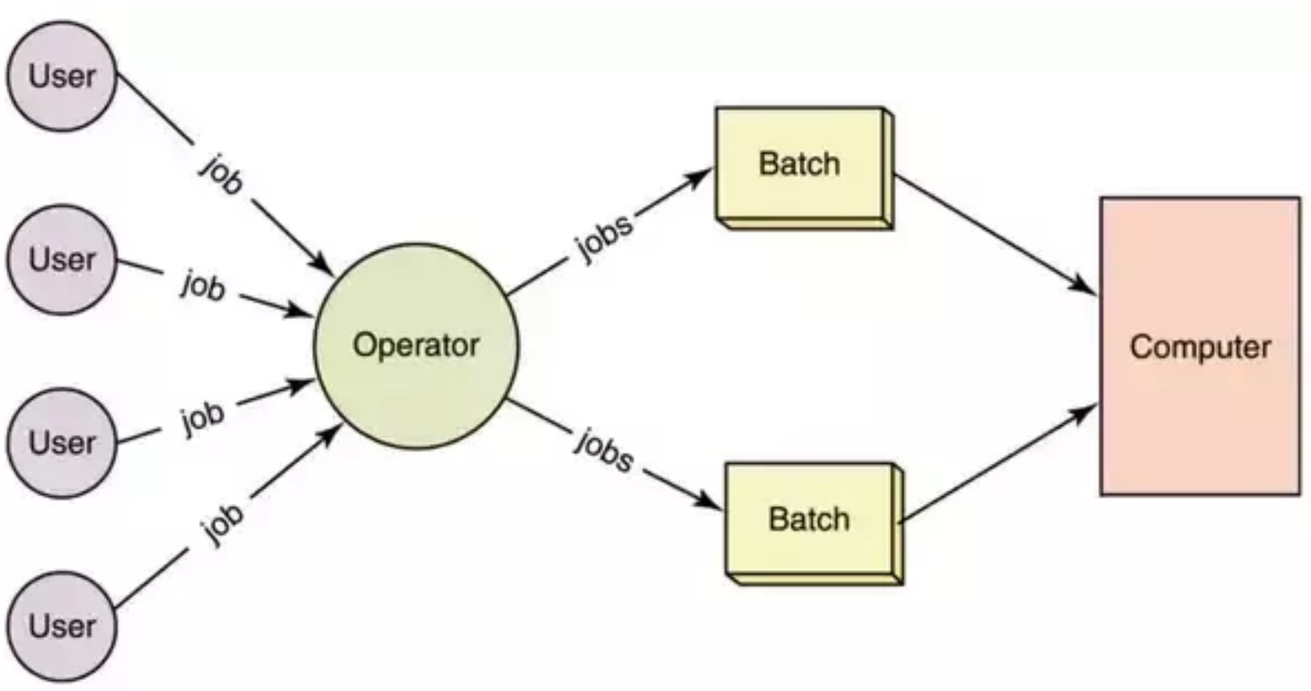
- To **integrate other system components and manage resources**
- To make the components work together in a special way, known as **transaction-oriented processing**

Five Computing Styles

- Batch processing
- Time-sharing
- Real-time processing
- Client-server processing
- Transaction oriented processing

Batch Processing

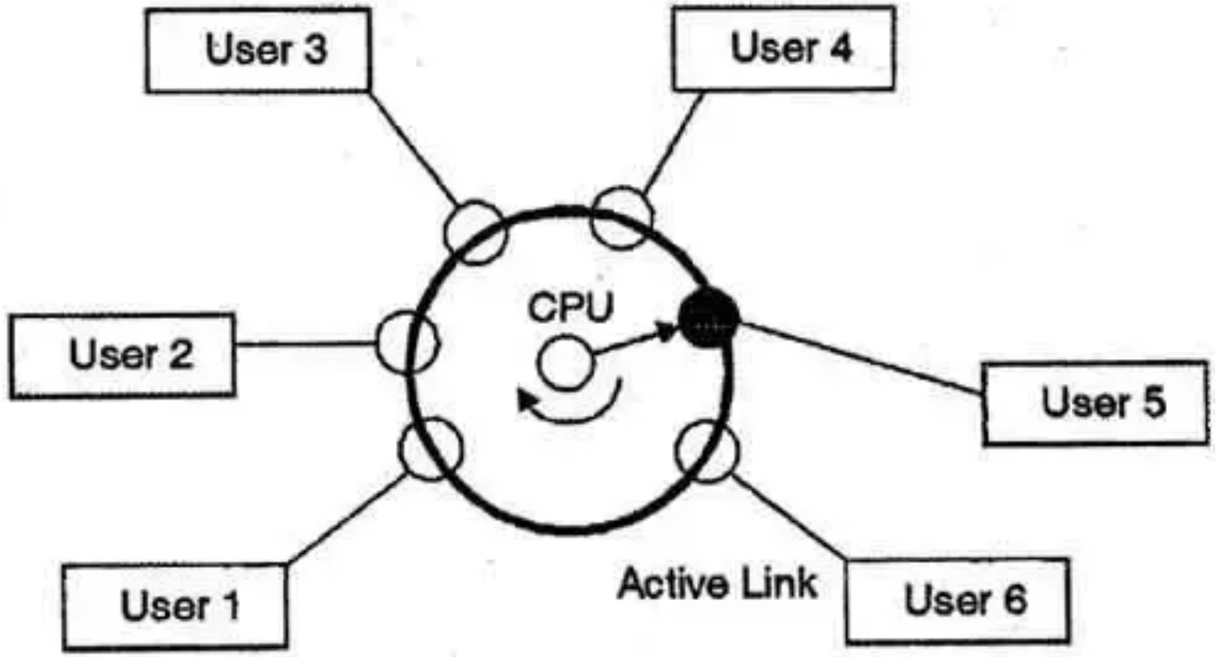
- Large units of work
- Coarse-grained resource allocation
- Sequential access patterns
- Application does recovery
- Few (tens of) concurrent jobs
- Isolated execution



Time-Sharing

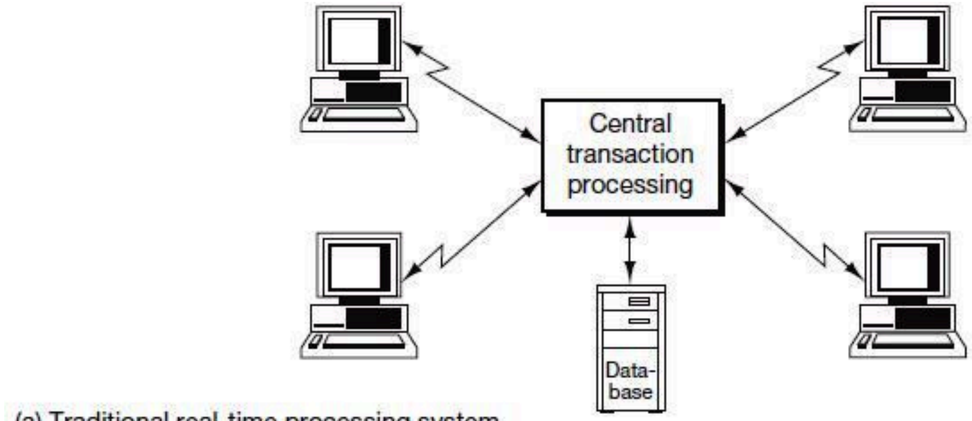
- **The terminal-oriented version of batch**

- Process per terminal
- Coarse-grained resource allocation
- Unpredictable demands
- Sequential access patterns
- Application does recovery
- Hundreds of concurrent users

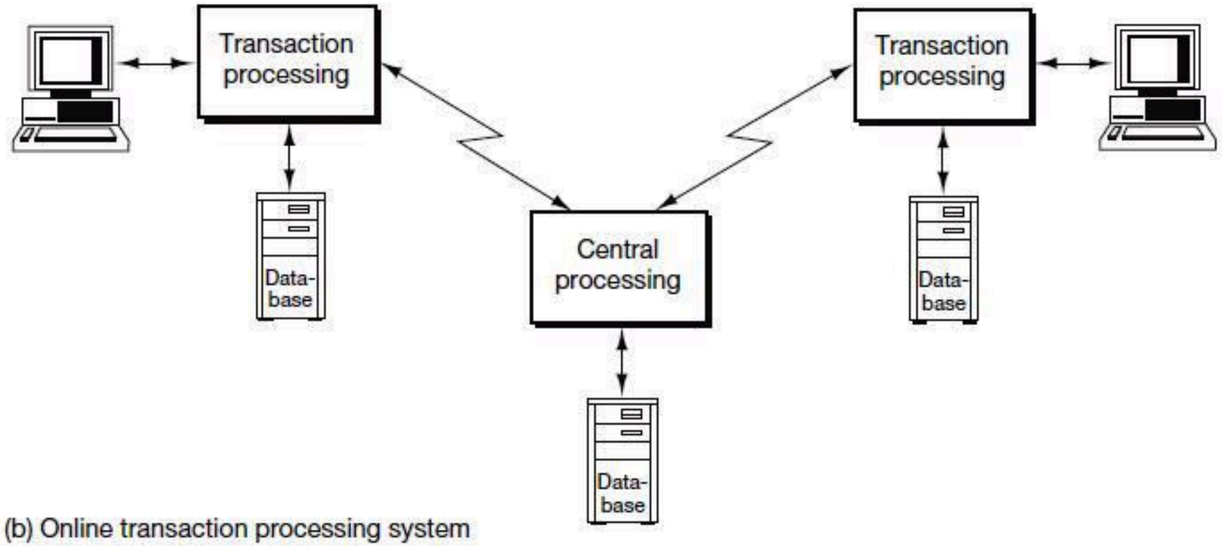


Real-Time Processing

- Event-driven operation
- Repetitive workload
- Dynamic binding of devices to tasks
- Isolated execution
- High availability
- High performance



(a) Traditional real-time processing system

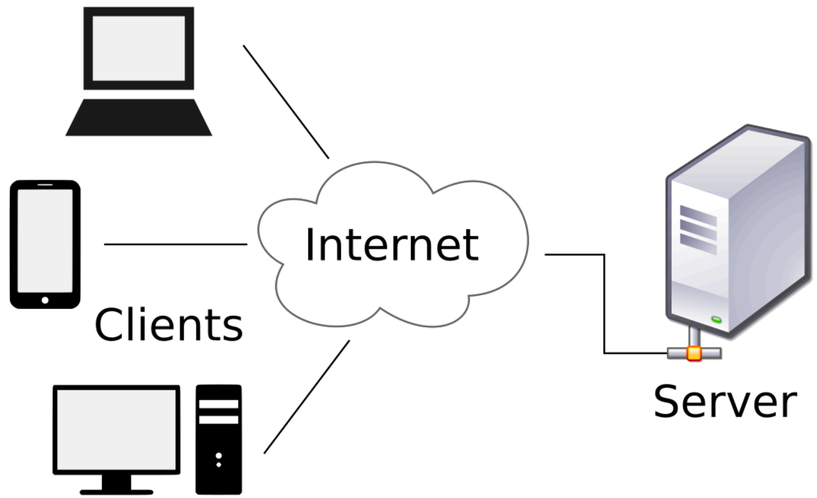


(b) Online transaction processing system

Client-Server Processing

- **The modern version of time-sharing**
  - Not everything is resolved within one process, but between different processes

- Each server provides a well-defined service
- All persistent data are encapsulated in database servers



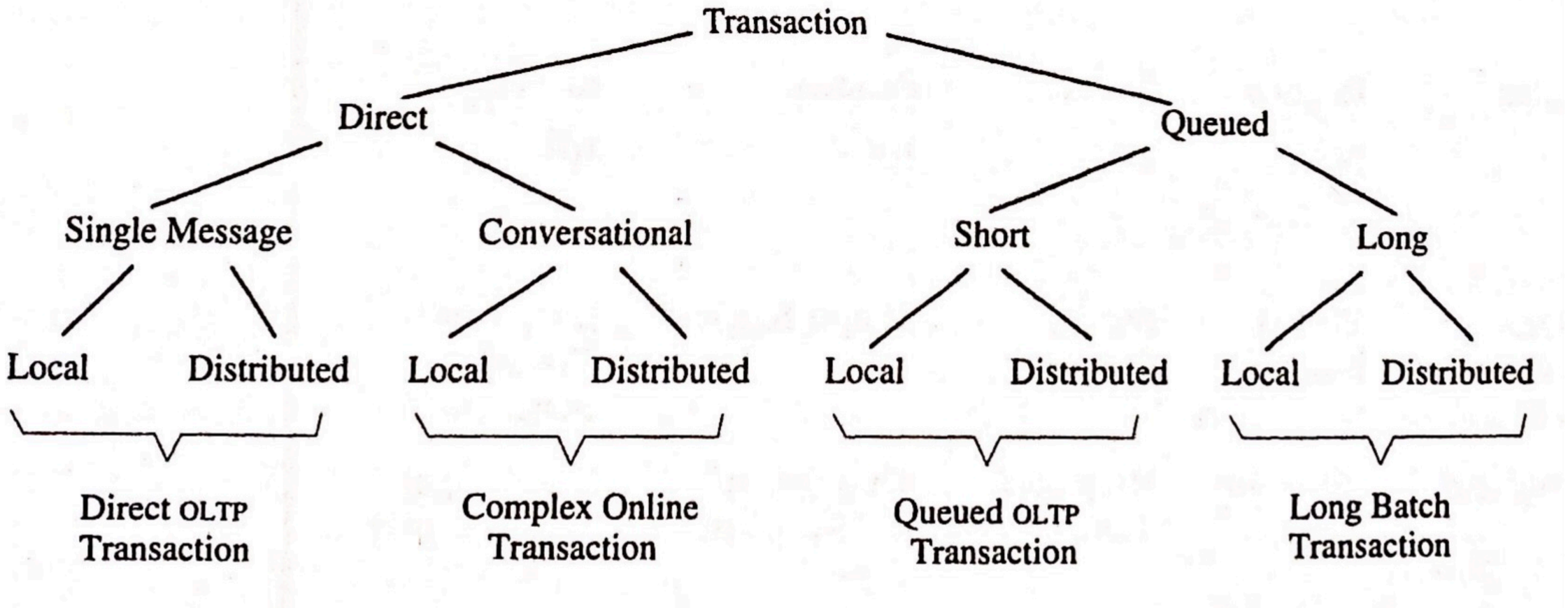
Transaction-Oriented Processing (OLTP)

- Sharing
- Variable requests
- Repetitive workload
- Mostly simple functions
- Some batch transactions
- Many terminals and intelligent clients
- High availability
- System does recovery (ACID)
- Automatic load balancing

Comparison of The Five Styles of Computation

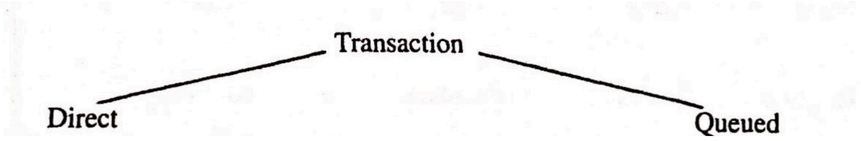
Styles	Batch	Time-Sharing	Real-Time	Client-Server	Transaction-Oriented
Data	Private	Private	Private	Shared	Shared
Duration	Long	Long	Very Short	Long	Short
Guarantees of Reliability	Normal	Normal	Very High	Normal	Very High
Guarantees of Consistency	None	None	None	-	ACID
Work Pattern	Regular	Regular	Random	Random	Random
# of Work Src/Dest	10^1	10^2	10^3	10^2	10^5
Services Provided	Virtual Processor	Virtual Processor	Simple Function	Simple Request	Simple or Complex Function
Performance Criteria	Throughput	Response Time	Response Time	Throughput & Response Time	Throughput & Response Time
Availability	Normal	Normal	High	High	High
Unit of Auth	Job	User	-	Request	Request

A Taxonomy of Transaction Execution



Direct vs. Queued Transactions

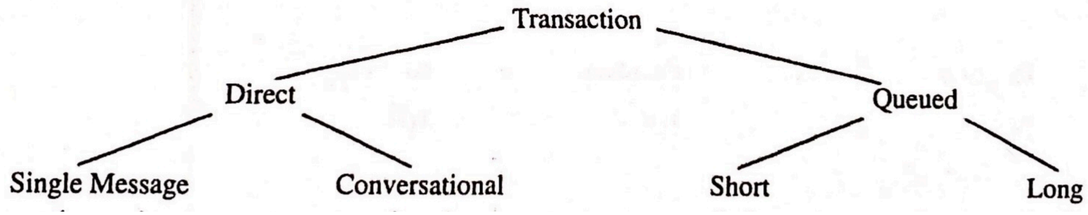
- **Direct transaction**
  - The client interacts directly with the server
  - e.g., TPC-A benchmark



- **Queued transaction**
  - The client's request is queued in the server and scheduled for processing according to the queueing discipline

Simple vs. Complex Transactions

- **Simple transaction**
  - It is short and touches only a few objects
  - Direct + Simple = Single Message



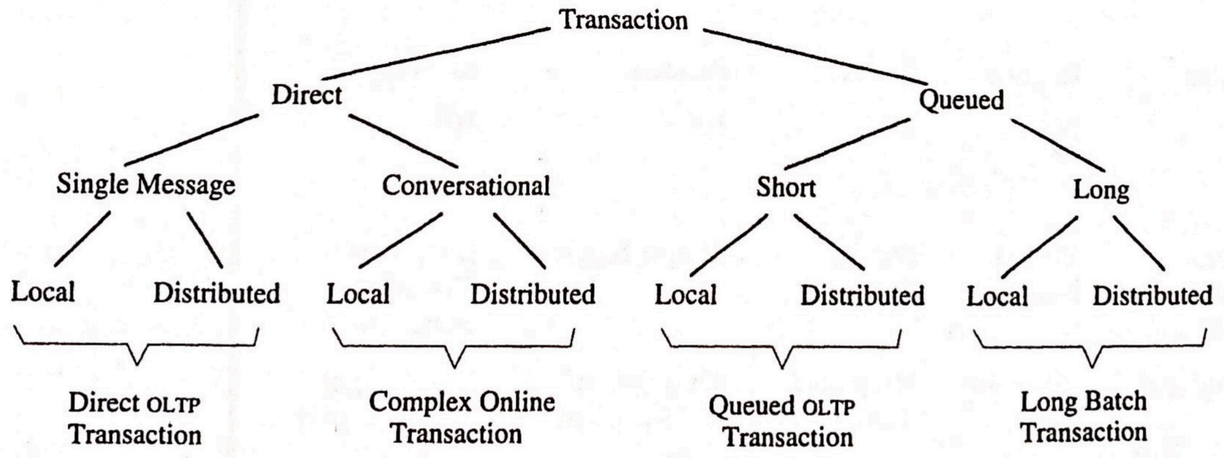
- **Complex transaction**
  - It is has a rich control flow and touches many objects
  - Direct + Complex = Conversational

- In queued transactions, there is nobody the transaction program could talk to at the other end → They are only classified as short or long transactions (based on the duration of the transaction & the amount of work it does)

Local vs. Distributed Transactions

- **Local transaction**
  - Transactions run entirely on the network node where the request originated

- **Distributed transaction**
  - Transactions invoke services from other nodes



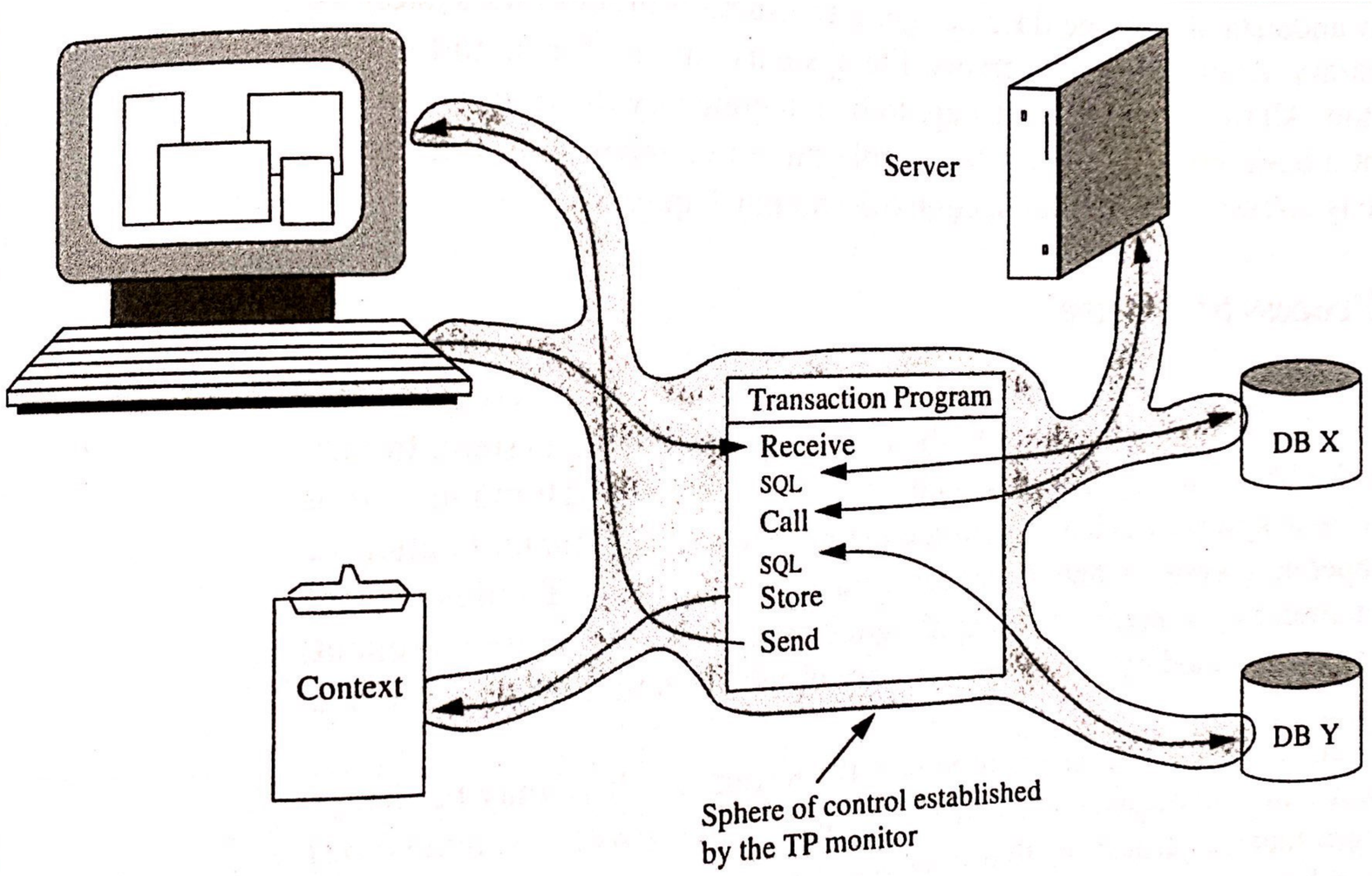
Transaction Processing Services (1)

- **Manage heterogeneity**
  - Transaction services must combine all operations on autonomous object into one transaction (i.e., resource manages (RMs))
- **Control communication**
  - The status of the remote/local communication must be subject to transaction control by the transaction services (i.e., transactional remote procedure call (TRPC))
- **Terminal management**
  - Since the ACID properties of a transaction must be perceived by the user, sending and receiving the messages must be part of the transaction

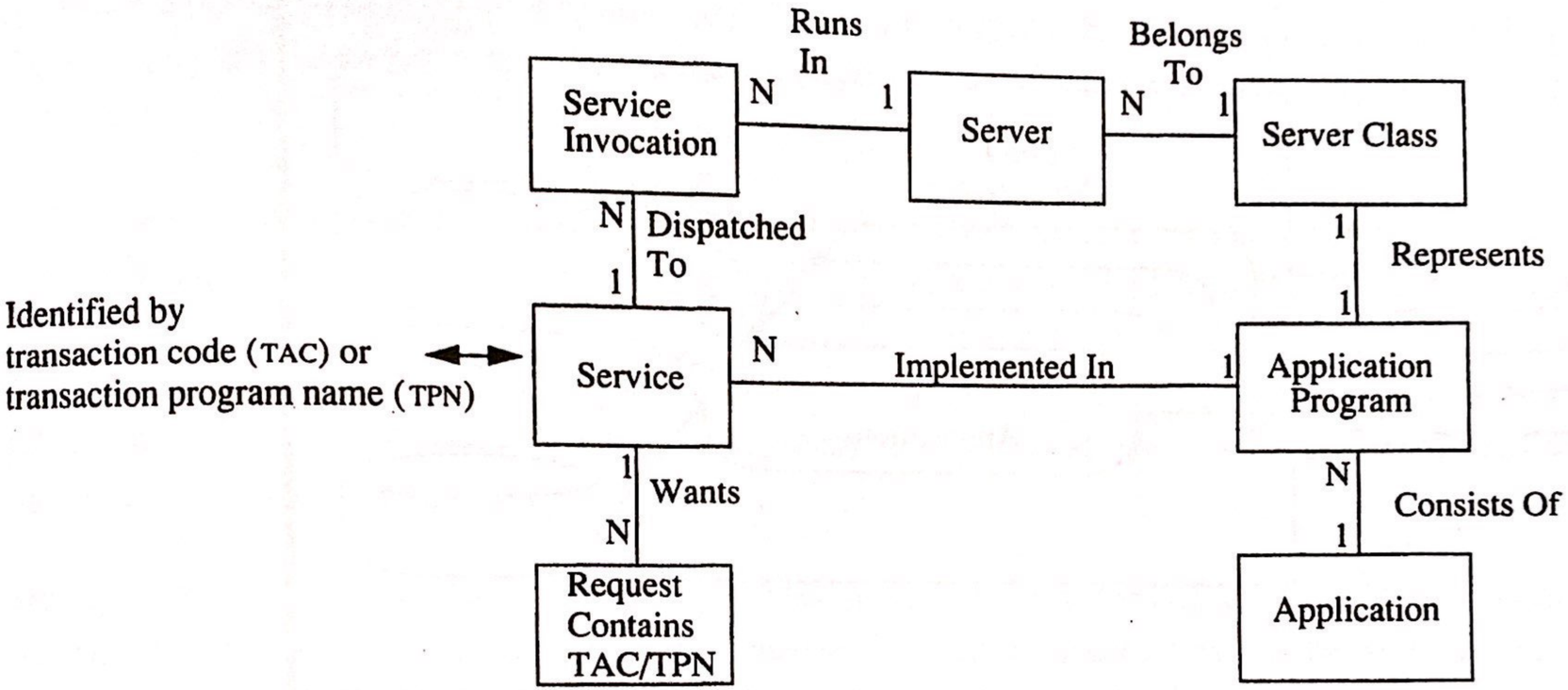
Transaction Processing Services (2)

- **Presentation services**
  - If the terminal uses presentation services, then reestablishing the window environment after a crash is also part of the transaction guarantee
- **Context management**
  - Storing and recovering context is bound to the sphere of control of the transaction that created or last modified the context
- **Start/restart**
  - Since the transaction services are responsible for virtually all the components the application needs to run, the TP monitor must handle the restart after any failure

Integrated Control by Transaction Services

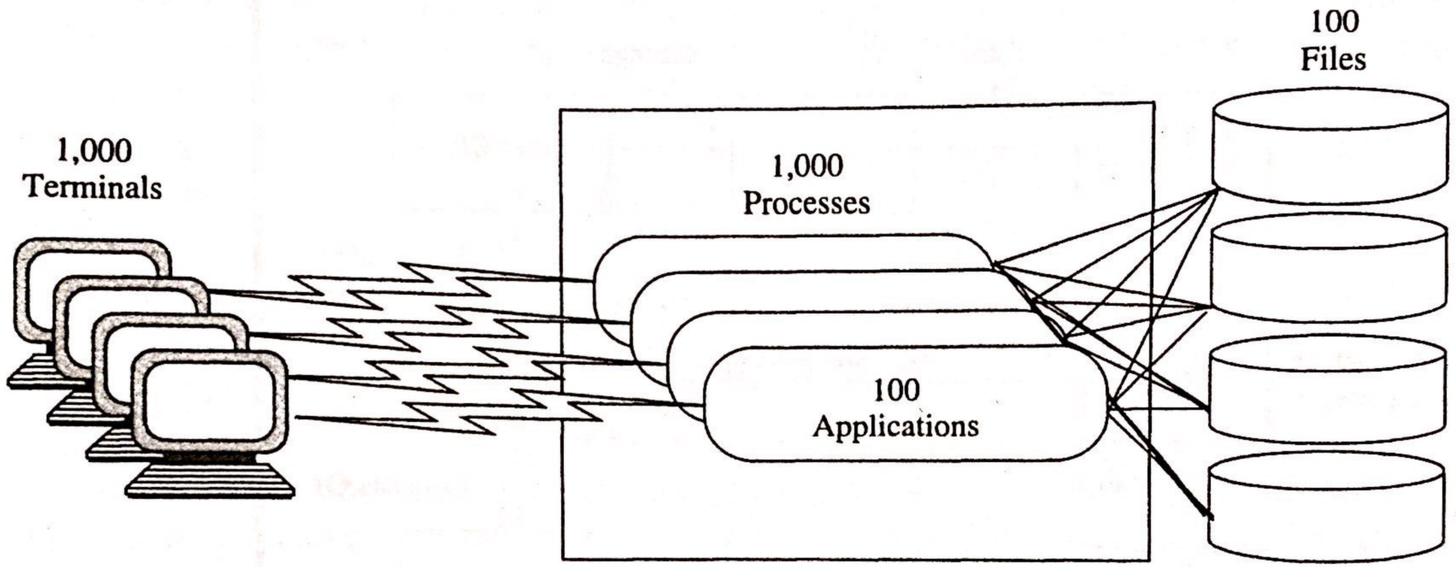


Concepts Involved in TP



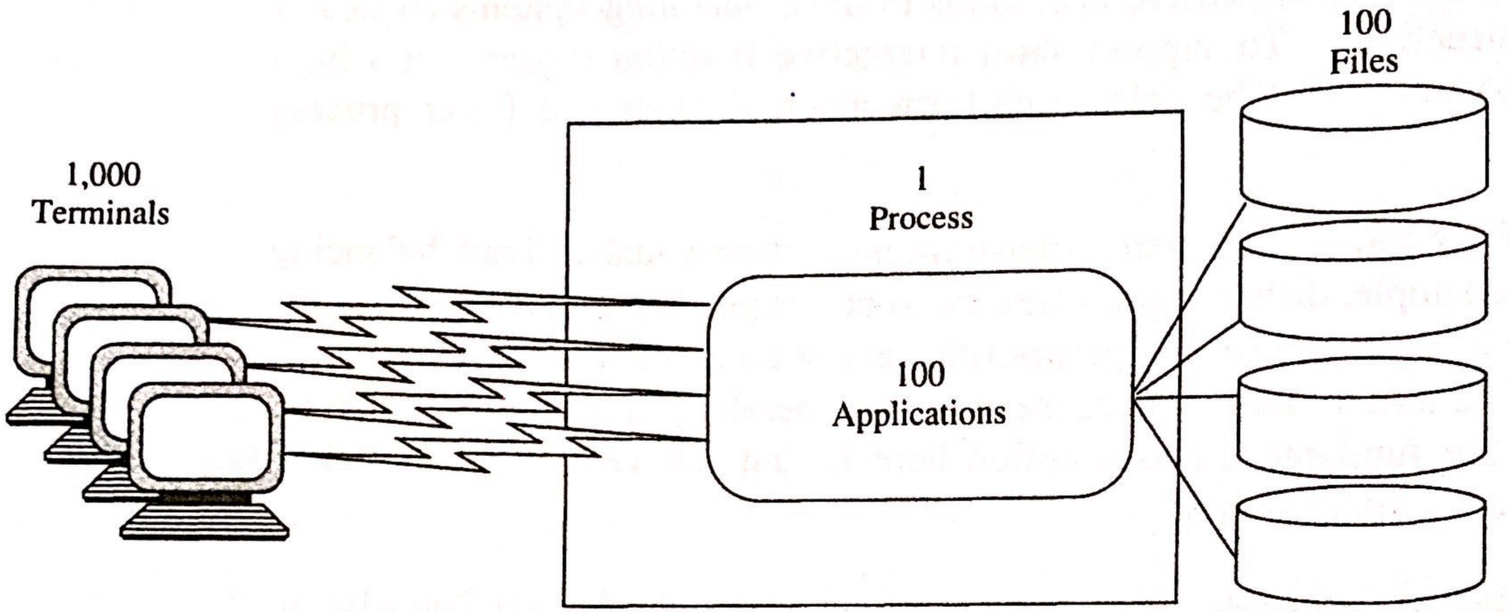
How to map a request to a proper process?

Option 1: One Process per Terminal



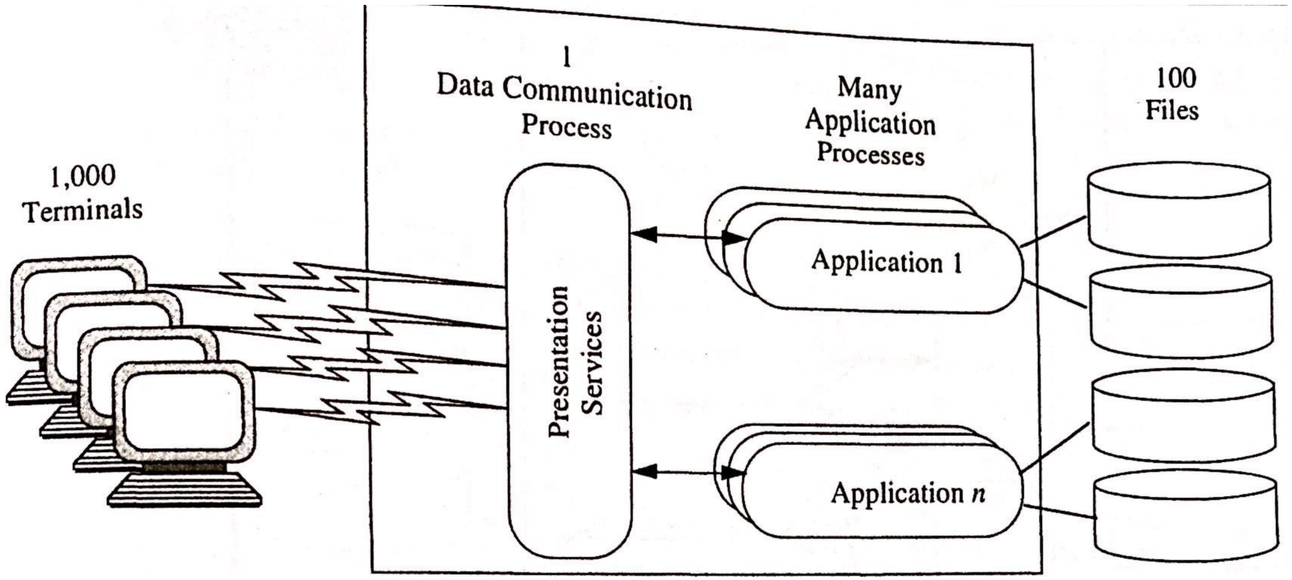
- Too many processes, control blocks, and process switches
- Inflexible load balancing
- Too many capabilities per process

Option 2: Only One Terminal Process



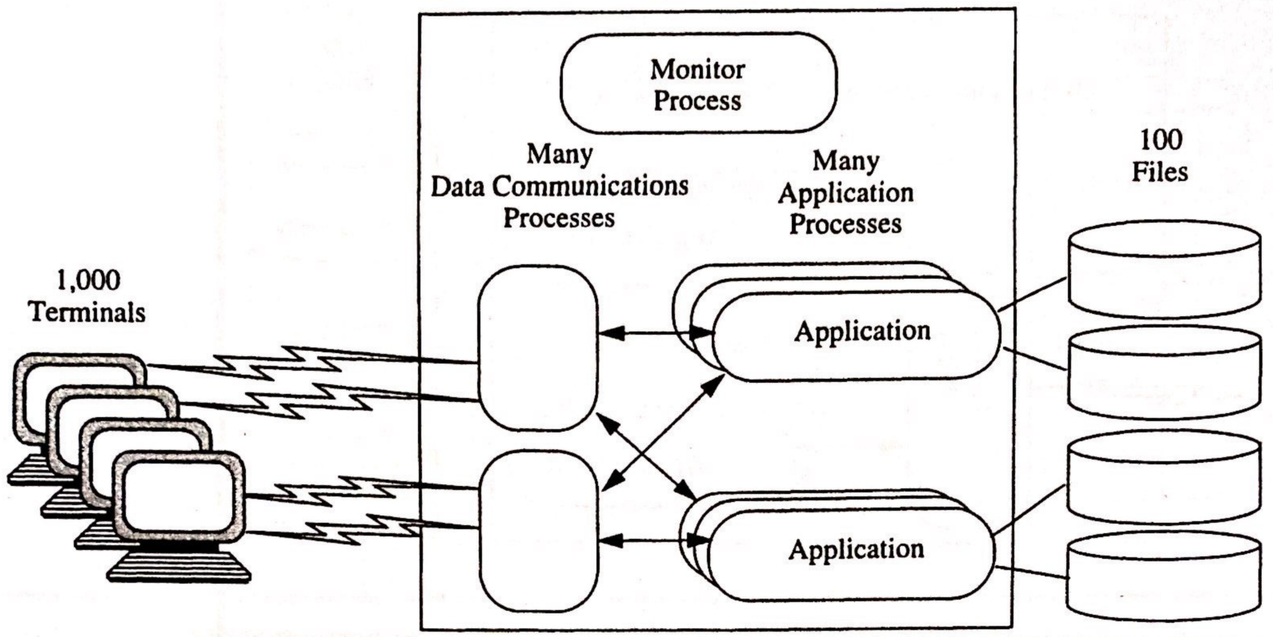
- A single process makes things easy
  - It is ideal to coordinate distributed transaction processing
- But, the TP monitor is confined within one address space and one OS process
  - A serious limitation for large apps and multiprocessor systems

Option 3: Many Servers, One Scheduler



- There are multiple processes running the applications
- There is one place for scheduling and load control
- But, under high load, the presentation service process will become a bottleneck

Option 4: Many Servers, Many Scheduler

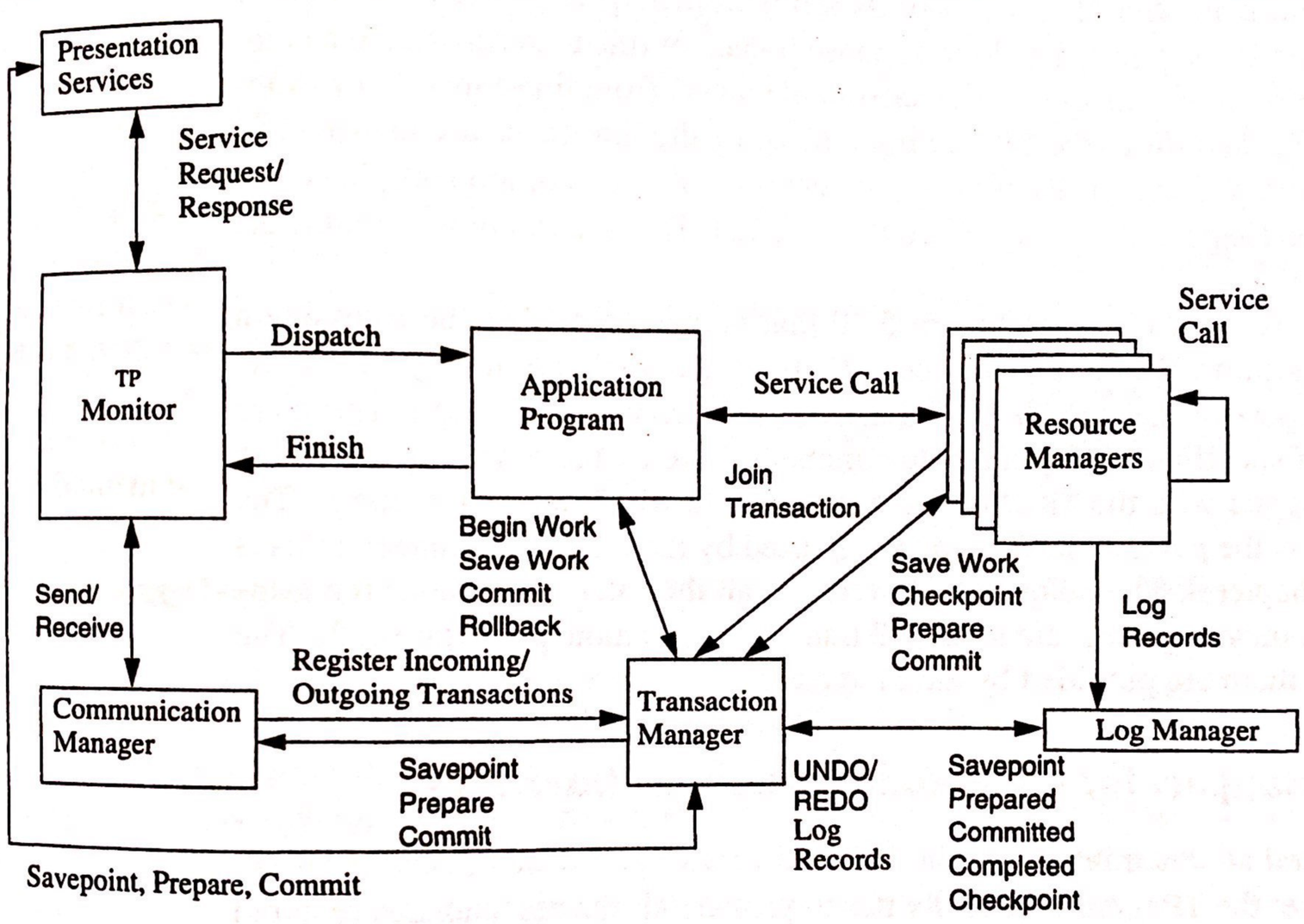


- The presentation service processes serve as a routers

To-Do List for TP Monitors

- Scheduling
- Server class management
- Authentication and authorization
- Resource administration
- System operation
- Recovery

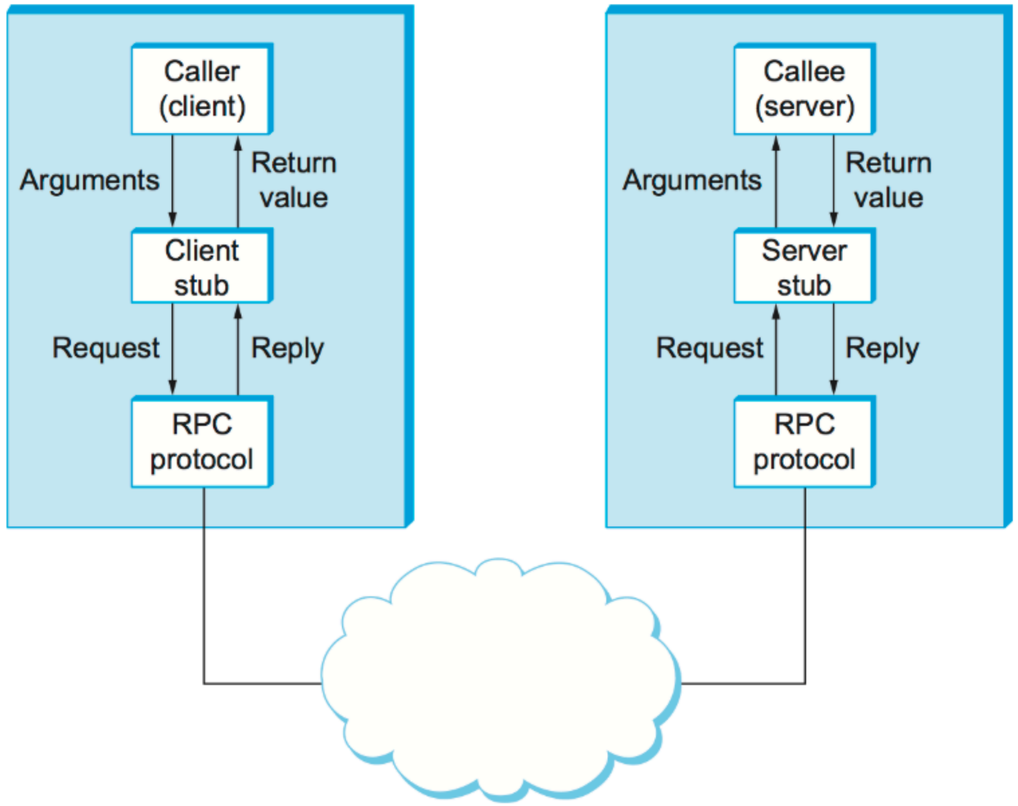
TP Components and Interfaces



The Major Responsibilities of TP Monitors

- With respect to the resource managers:
  - Restart and system startup
  - Definition of a new resource manager
  - Changing the process configuration
  - **Handling transactional remote procedure calls (TRPCs)**
    - A fundamental chore of TP monitors
    - To manage the server class

Remote Procedure Calls (RPCs)



- RPCs are invented for distributed systems to make the invocation of services at remote nodes look like local subroutine calls

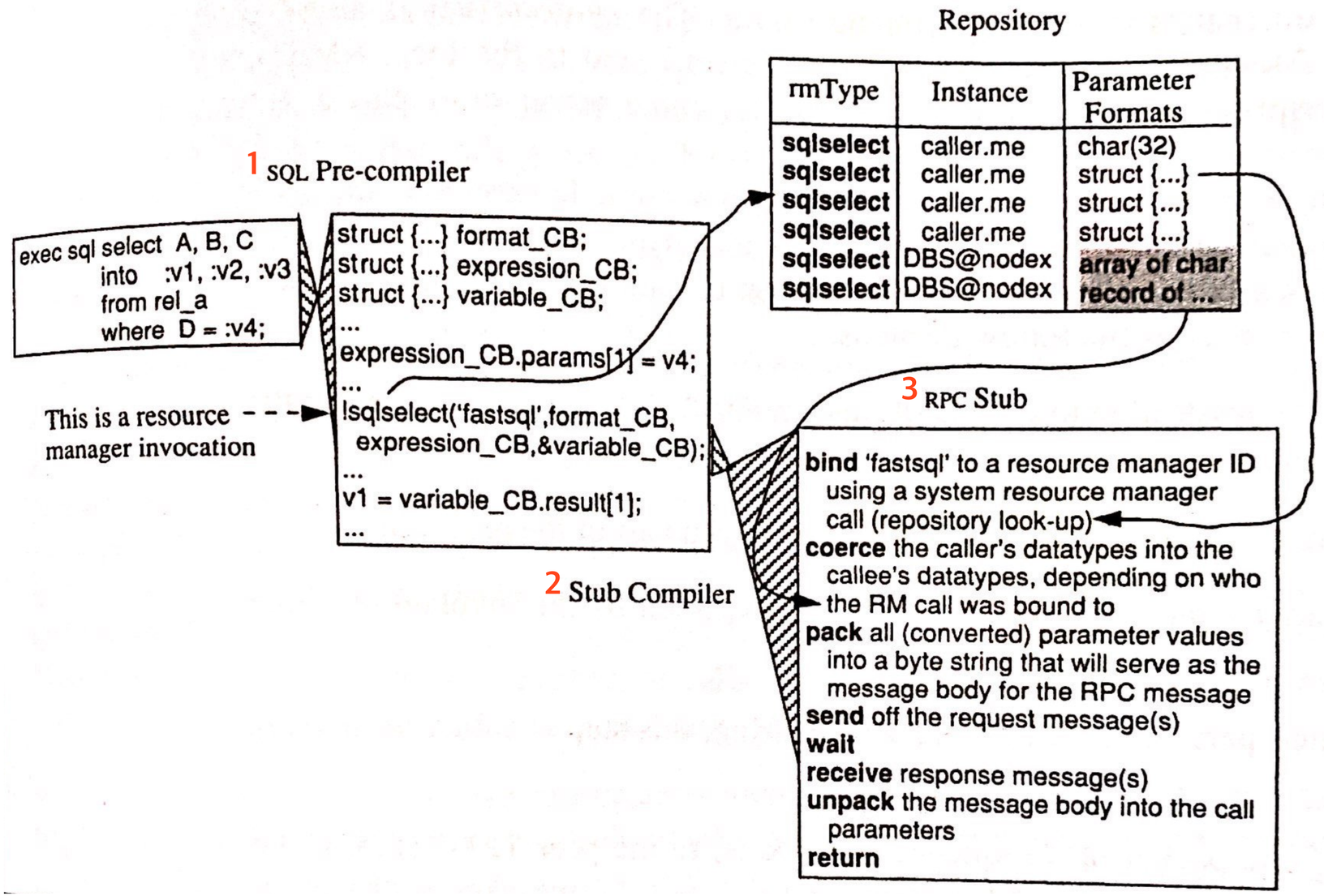
Transactional Remote Procedure Calls (TRPCs)

- The TRPC is a RPC protocol that implements the necessary plumbing to cope with a caller and/or callee that are running a transaction

- Resource managers issue RPCs to other resource managers
  - With `RMNAME` and `RMID`

- Main Features:
  - Bind RPCs to transactions
  - Inform the transaction manager

SQL Example of TRPCs



Terminological Wrap-Up

1. A user issues a ***service request*** at a terminal
2. The ***presentation services*** translate the terminal inputs into a message with a standard format according to a ***forms definition*** stored in the ***repository***
3. This message is given to the ***TP monitor***
4. The TP monitor maintains a ***server class*** for that application program and dispatches the request to a ***server process*** of that server class
5. If the service request is not part of an ongoing transaction, the program calls BEGIN_WORK, which is a ***transactional remote procedure call (TPRC)*** to the ***transaction manager***
6. The transaction manager assigns a globally unique ***transaction identifier (TRID)*** and establishes ***transaction context***
    - The transaction program can contain calls to ***resource managers*** (through TRPC)
    - The transaction manager will interact with resource manages to complete the transaction (COMMIT_WORK)

Reference

[1] Jim Gray and Andreas Reuter, “Transaction Processing: Concepts and Techniques”, Morgan Kaufmann, San Mateo, CA (1993)

[2] “Working with Transaction Monitors with Oracle XA”, Oracle9i Application Developer's Guide - Fundamentals, Oracle, https://docs.oracle.com/cd/B10501_01/appdev.920/a96590/adga1_xa.htm#619390

[3] “What is a batch operating system, time sharing operating system, distributed operating system, network operating system and an embedded system?”, Quora, https://www.quora.com/What-is-a-batch-operating-system-time-sharing-operating-system-distributed-operating-system-network-operating-system-and-an-embedded-system

[4] Amos Kingatua, “What is OLTP - Online Transaction Processing?”, Electronics, Information & Communications Technology, https://electronicsandict.com/what-is-oltp-online-transaction-processing/

[5] Stefan Debloch, “Chapter 4 Remote Procedure Calls and Distributed Transactions”, Technische Universität Kaiserslautern, http://wwwlgis.informatik.uni-kl.de/cms/fileadmin/courses/ws1516/middleware/Chapter_4_RPC_and_DTP.pdf