

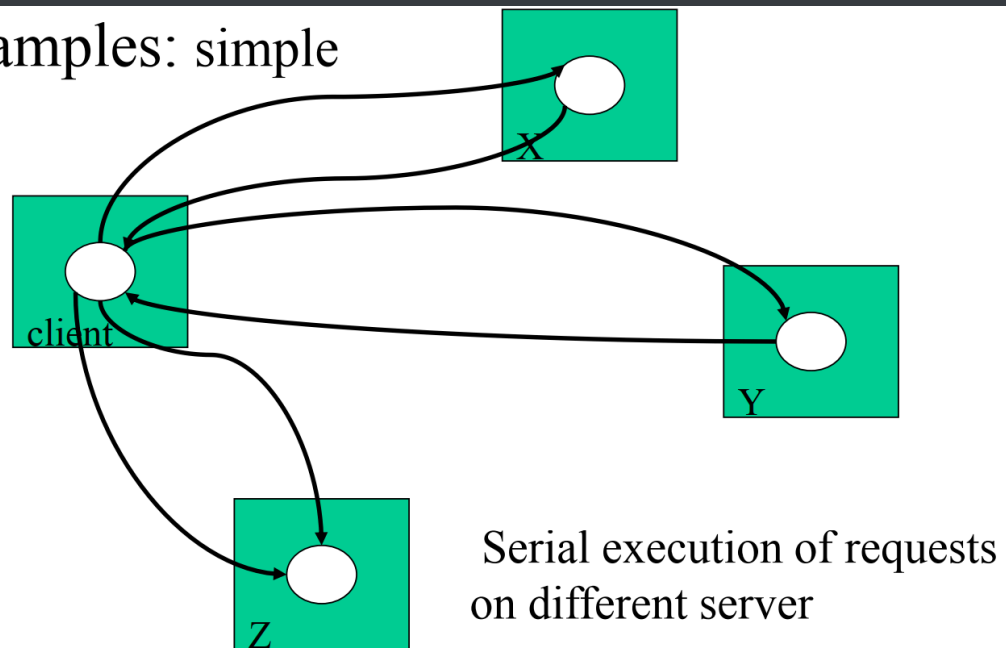
transaction – 2

flat and nested distributed transactions

simple distributed transaction

client accesses several servers

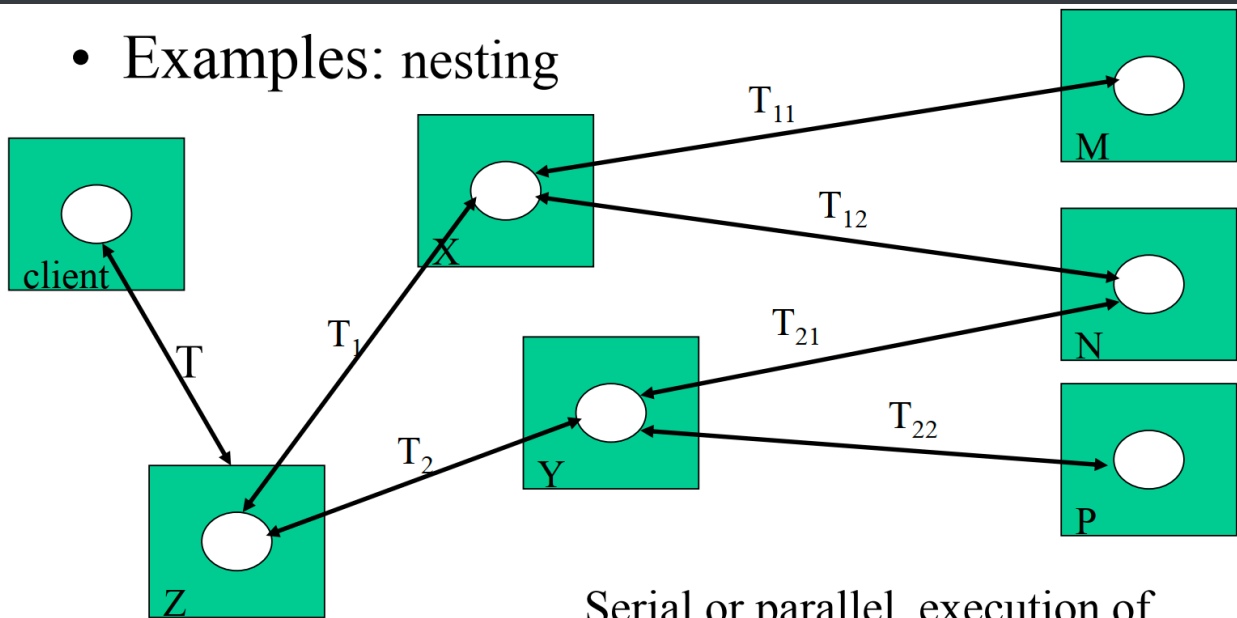
- Examples: simple



nested distributed transaction

server accesses several other servers

- Examples: nesting



Serial or parallel execution of requests on different servers

coordinator, workers

- new service operations:

- AddServer (TransID, CoordinatorID)

called by clients, 当transaction需要访问新的server时

client让新server加入transaction

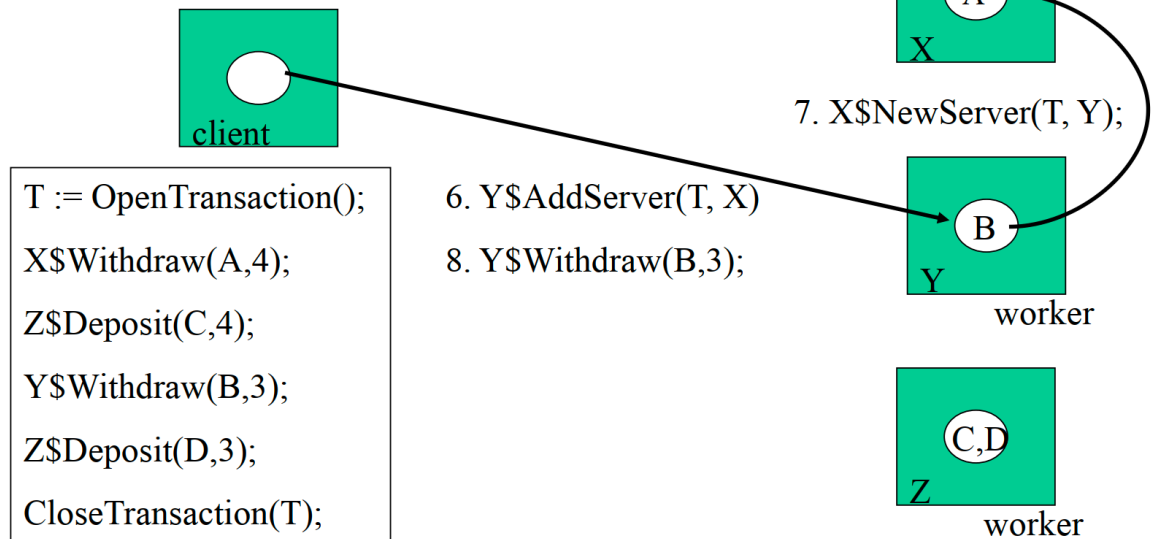
- NewServer (TransID, WorkerID)

called by new server on the coordinator

new server通知coordinator自己已经加入新transaction

- example:

- Examples: simple

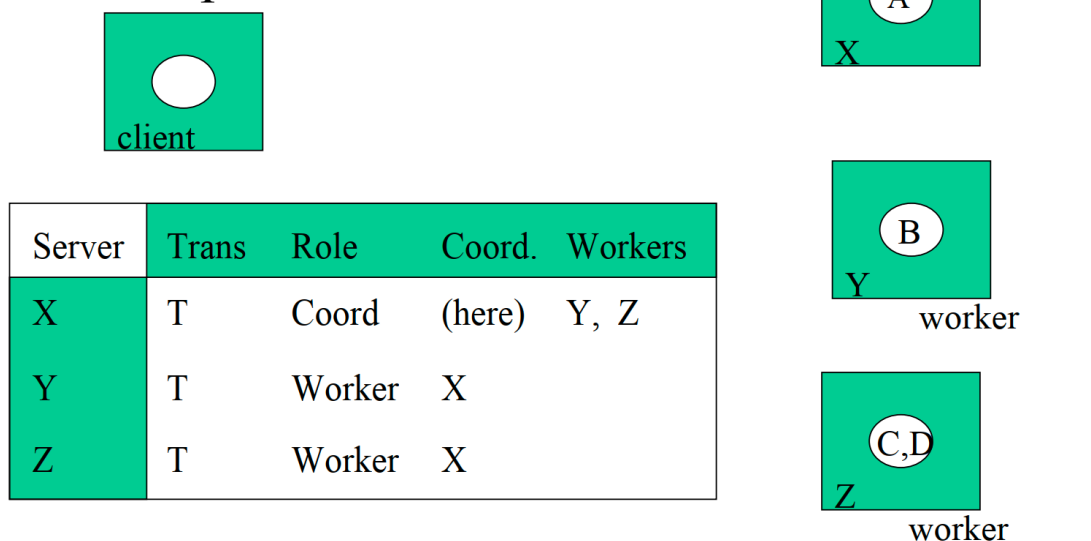


“喂，你的coordinator是X，你要参与的transaction是T”

“喂，我是Y，我要参与的transaction是T”

- server上维护的表：

- Examples: data at servers



atomic commit protocols

protocol

- phase 1: voting_phase
 - coordinator on CloseTransaction
 - send "CanCommit" to each worker

waits for replies from workers

- **worker** on **receiving CanCommit**

- if can commit

- saves data items, sends Yes

- if cannot commit

- sends No, clears data structures, removes locks

- phase 2: completion according to outcome of vote

- Coordinator **collecting votes**

- send DoCommit to workers only **all vote Yes**

- worker **who voted Yes**

- on DoCommit

- makes committed data available, removes locks

- on AbortTransaction

- clears data structure, removes locks

concurrency in distributed transactions

distributed deadlocks

transaction recovery

replication