**Transaction Manager**

Lists

Open script files

Currently executing transactions, processes

1. Reads commands from different program files concurrently

Open and parse the script files (round robin or random)

B – initialize new transaction or process (add to list, get TID)

C – pass command on to scheduler/DM (remove from list)

A – pass command on to scheduler/DM (remove from list)

R/M/W/D – pass command on to scheduler/DM

\*Which module actually tracks effects so can undo them?

**Scheduler**

a. Lock manager – strict 2PL

If T wants to read/write an object, first obtains S/X lock

Holds all locks until end of transaction

Guarantees serializability and recoverable schedule too

Thus ensures avoids cascading aborts

2PL ensures that the precedence graph will be acyclic

Lock table – hash of lock entries

OID, Mode, List (Xacts holding lock or a count), List Wait queue

\*Number of transactions currently holding a lock

\*Type of lock held (shared or exclusive)

\*Pointer to queue of lock requests

Locking and unlocking have to be atomic operations

b. Deadlock detector

WFG – nodes are transactions, node from i to j if i waiting for j

1. Receives command from transaction manager along with TID

a. B emode

Anything to do?

b. Request R/M/W/D

If doesn’t have locks -> have to request them

If can’t get -> add to WFG

c. A/C - release locks and remove from WFG

Approve a pending request