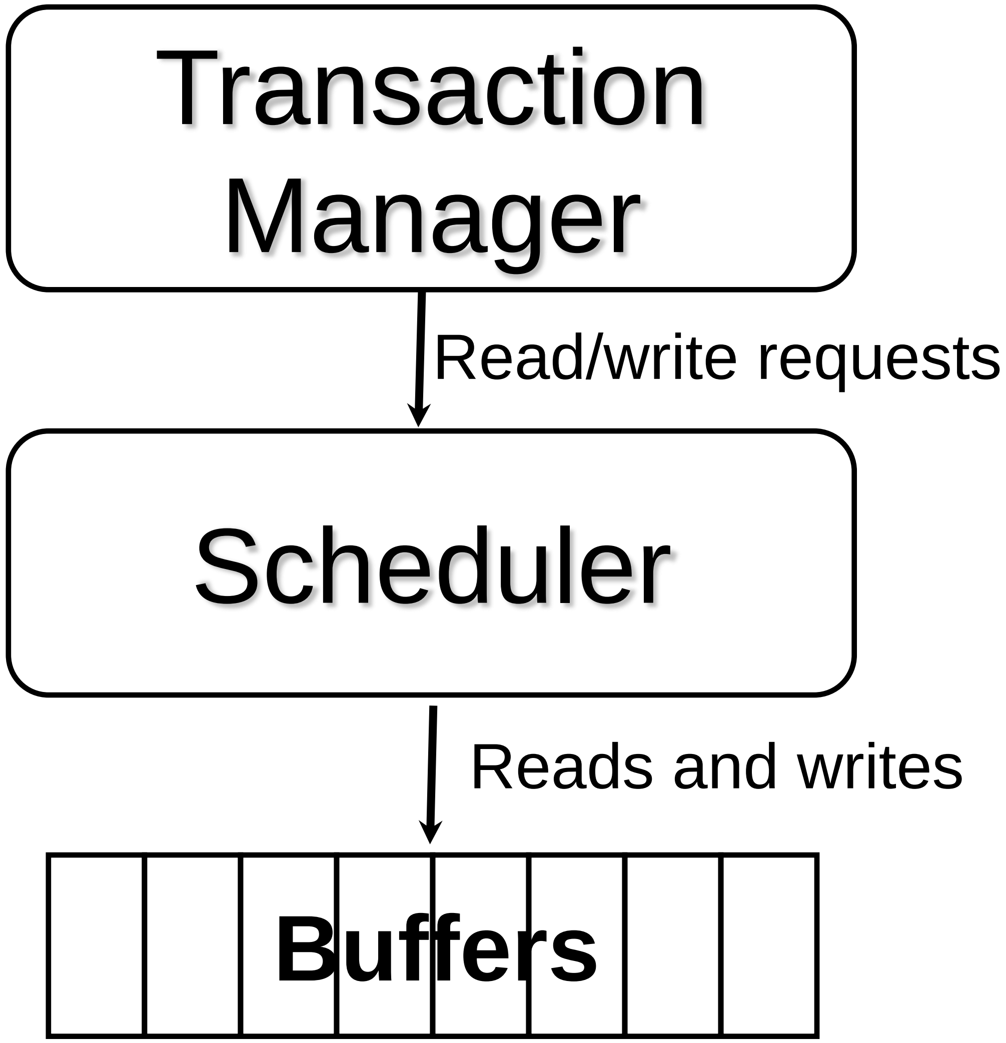
Section 5: Concurrency Control

Thursday, April 30 2009

Reminder

- Homework 1 was due yesterday (10/28/09)
- Project 2 due next Wednesday (11/4/09)
- Pickup Concurrency Control Worksheet for today

Concurrency Control



- What is the purpose of the scheduler?

Schedules

- Serial
- Serializable
- Conflict Serializable

Optimistic vs Pessimistic

- What is the difference?
- When is it preferable to have optimistic concurrency control?
- When is it preferable to have pessimistic concurrency control?

Pessimistic Concurrency Control: Locks

- Simple?
- 2PL?

Optimistic Concurrency Control

- Timestamps
- Validation

Concurrency Control: Timestamps

- Key idea: The timestamp order defines the serialization order.
- Scheduler maintains:
  - **TS(T)** for all transactions T
  - **RT(X)**, **WT(X)**, and **C(X)** for all data elements X

Scheduler receives request from transaction T ...

- grant request
- rollback T
- delay T

Scheduler receives request from transaction T ...

1. If read request $r_T(X)$:

2. If write request $w_T(X)$:

3. Commit request:

4. Abort request:

Exercises

1.st1; st2; st3; r1(A); r2(B);r2(C); r3(B); com2; w3(B);w3(C)

2.st1; st2; r1(A), r2(B);w2(A); com2; w1(B)

3.st1; st3; st2; r1(A); r2(B); r3(B);w3(A);w2(B); com3; w1(A)

4.st1; r1(A); w1(A); st2; r2(C); w2(B); r2(A); w1(B)

Multiversion Timestamps

- Keep multiple version of each data element along with the write timestamp.
- Will reduce number of aborts due to read-too-late problem.