Transaction Processing

An Introduction

Dr Stewart Blakeway
Lecturer in Computer Science

Intro

Fundamentals of Database Systems
Chapter 20
Publisher: Pearson; 7 edition (19 Aug. 2016)
Language: English
ISBN-10: 1292097612
ISBN-13: 978-1292097619

Learning Objectives

- To understand what a Transaction is
- To understand what a Database Item is
- To understand Concurrency and Interleaving
- To understand how Database Items are managed by the DBMS
- To understand why we need concurrency control
- The lost update and dirty read problem

Transaction Processing

Any database system that requires high availability and fast response time for hundreds of concurrent users

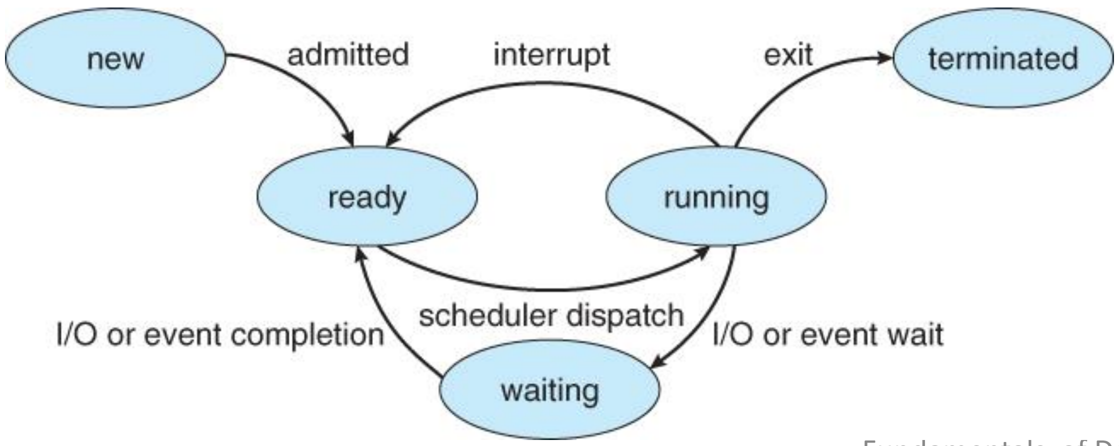
- A transaction is a mechanism for describing logical units of database processing
- Transaction processing systems are systems with large databases and hundreds of concurrent users
  - Airline Reservations
  - Banking Systems
  - Credit Card Processing
  - Online retail Purchases
  - Stock Markets
  - Supermarket Checkouts
  - and many more

Single User vs Multiuser Systems

- One criteria for classifying a database system is the number of users that can access the database concurrently
- A DBMS is single user if only one user at a time can use the system
- A DBMS is multiuser if the system is access by multiple users at the same time (concurrently)
- Concurrency is achieved using the concept of multiprogramming
  - Allows the operating system to execute multiple processes at the same time
  - But… A single core CPU can only execute at most one process at the same time, so how can concurrency be achieved?

Interleaving

- Process A runs, at some time later is switched out (or on I/O)
- Process B runs, at some point later is switched out (or on I/O)
- Process A is switched in and continues, at some point later is switched out (or on I/O)
- The process repeats for the number of processes
- So although we think these processes are running at the same time, they are actually interleaving



Transactions

- executing programs that forms a logical unit of database processing
- include one or more database access operations
- can be embedded within the application program or within the SQL server
- two classification of transactions

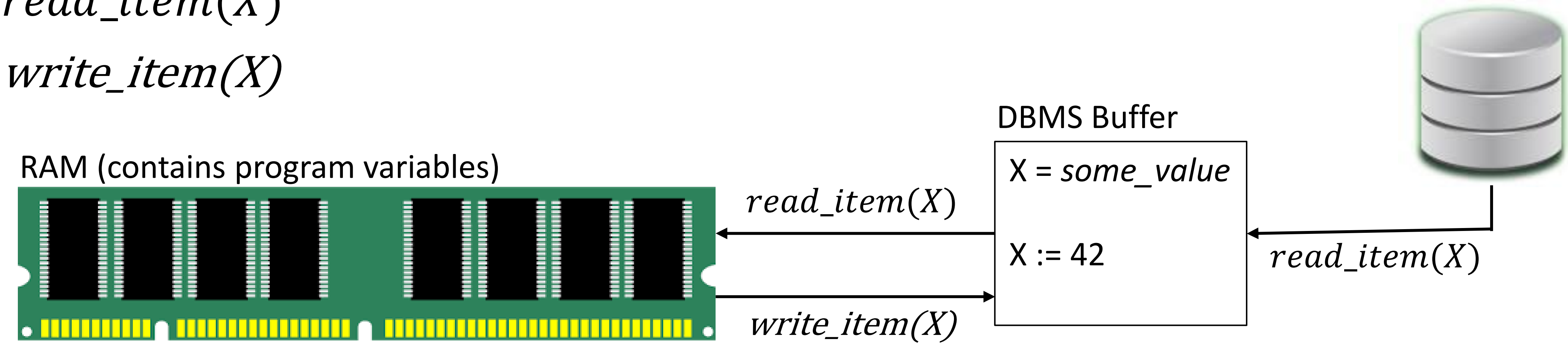
Database Items

- The database model that is used to present transaction processing is relatively simple
- A database is seen as a collection of named data items
- There are various levels of granularity that could be represented

Read and Write Operations and DBMS Buffers

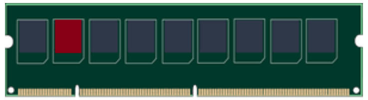
$read\_item(X)$

write_item(X)



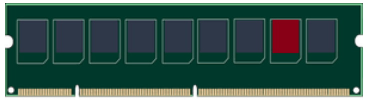
If the buffers are full and we need another buffer. The contents of a buffer will be swapped out to disk to make room. Normally using LRU.

A Simple Example with Context



$Program\ Memory$

T_1
N:=100
read_item(X)
X:=X – N
write_item(X)
read_item(Y)
Y:=Y + N
write_item(Y)



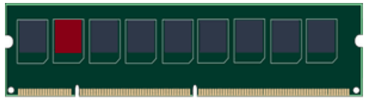
$DBMS\ Buffer$

	Customer Accounts	
X	Current (main)	£100
Y	Savings	£1500
Z	Credit Card	-£1200

Concurrency

- The Lloyds Bank Group has 27,000,000 customers
- It is likely that several customers will access their account at the same time
- It would not be ideal to wait for one transaction to finish before another began
  - They would not have 27 million customers if we did
- So we allow concurrent access to the DBMS
- The control of this concurrent access are mainly concerned with Transactions
  - And recovery mechanisms when something goes wrong

A Simple Example with Context



Program Memory	T_1
	…

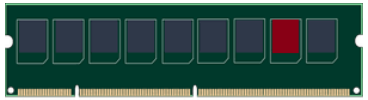
Program Memory	T_2
	…

Program Memory	T_3
	…

Program Memory	T_4
	…

…

Program Memory	T_n
	…



DBMS Buffer

	Supermarket Stock	
X	Beans	1065
Y	Milk	1887
Z	Apples	899

Concurrency Control

- Several Problems can occur when concurrent transactions execute in an uncontrolled manner

- The Lost Update
- The Temporary Update (Dirty Read)
- The Incorrect Summary
- The Unrepeatable Read

Airline Reservations Database

- A record is stored for each airline flight
  - Number of reserved seats
- When a program is written to interact with the database is uses flight number, flight date and the number of seats to be booked as parameters
  - The same program can be used to execute many different transactions each with a different
    - Flight number
    - Date
    - Number of seats to be booked

Airline Reservations Database

- T_1 shows a transfer of N reservations from one flight whose number of reserved seats is stored in X, to another flight whose number of reserved seats is stored in Y
- T_2 shows the reservation of M seats for the first flight
- These transactions are running concurrently

T_1
read_item(X)
X:=X – N
write_item(X)
read_item(Y)
Y:=Y + N
write_item(Y)

T_2
read_item(X)
X:=X + M
write_item(X)

The Lost Update Problem

T_1	T_2
read_item(X)	
X:=X – N	
	read_item(X)
	X:=X + M
write_item(X)	
read_item(Y)	
	write_item(X)
Y:=Y + N	
write_item(Y)	

time

- Two transactions run at approximately the same time
- They are interleaved to allow concurrency
- Our values, `X = 80, N = 5, M = 4, Y=50` where
  - X and Y are the number of reservations for a given flight
  - N is the number of seats to transfer to another flight
  - M is the number of seats to reserve for a given flight

Question, what is the final value of X? 84

Question, should be the final value of X? 79

COMMITTED and ABORTED

- Whenever a transaction is submitted to a DBMS for execution, the system is responsible for making sure that all the operations in the transaction are completed successfully
  - COMMITTED
- If something goes wrong the database must not be updated
  - ABORTED

- A transaction is a logical unit of instructions, if the transaction fails, all changes for that transaction must be rolled back

The Temporary Update Problem

T_1	T_2
read_item(X) X:=X – N write_item(X)	
	read_item(X) X:=X + M write_item(X)
read_item(Y) *Transaction fails*	

time

- Two transactions run at approximately the same time
- They are interleaved to allow concurrency
- Our values, `X = 80, N = 5, M = 4, Y = 50` where
  - X is the number of reservations for a given flight
  - N is the number of seats to transfer to another flight
  - M is the number of seats to reserve for a given flight

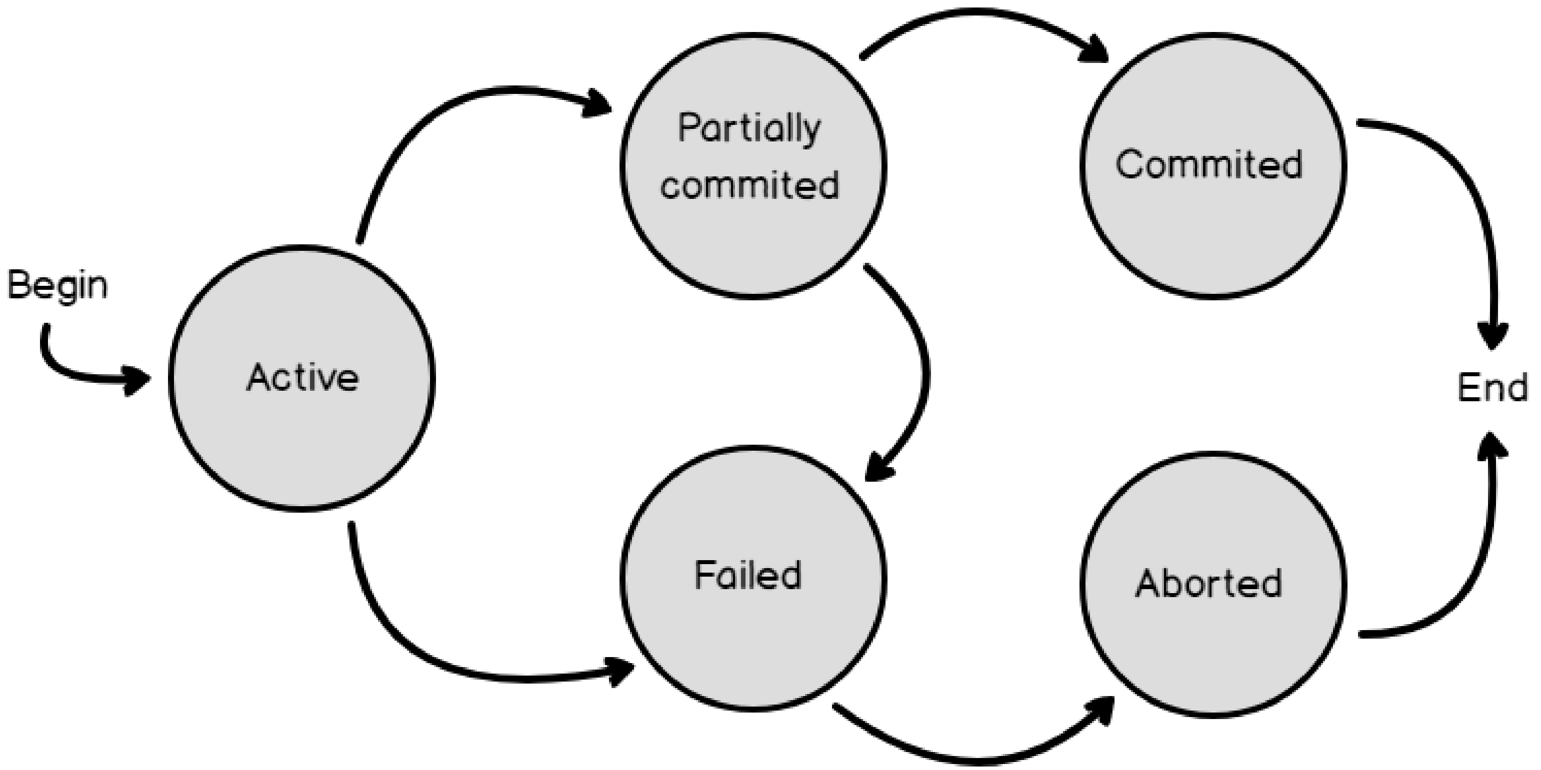
Question, what is the final value of X? 80

Question, should be the final value of X? 79

What could possibly go wrong?

- A Computer Failure (System Crash)
- Transaction Error
- Logical Errors or Exception Conditions detected by the Transaction
- Disk Failure
- Physical Problems and Catastrophes

Transaction States and Additional Operations



We have learnt

- what a Transaction is
- what a Database Item is
- what meant by Concurrency and Interleaving
- an abstract understand how Database Items are managed
- why we need concurrency control
- The lost update and dirty read problem

Transaction Processing

An Introduction

Dr Stewart Blakeway
Lecturer in Computer Science