

CSCE 4610/5610: Solutions To Final Exam

May 8, 2008 1:30-3:30pm

1. (15%) Describe in your own words what Sequential Consistency means, Total Store Order means and what a release consistency means. Use example code sequences to illustrate your understanding.

Key.

Sequential consistency is the most widely accepted definition of correct parallel program execution. SC states that any correct parallel program should behave as if the parallel segments of the program are executed sequentially by interleaving the code segments, while maintaining program order within each parallel code segments. In addition to program order, SC requires that all memory operations (both read and write) performed by all processing units (executing parallel code segments) be totally ordered – in other words, all memory operations be seen by all processing units in the same order.

Total Store Order is similar to SC, but relaxes the ordering on some memory operations. TSO requires a total ordering only on the store operations performed by the processing units.

Release consistency is one form of weak memory orderings. Such weak consistency models enforce ordering only at lock boundaries: all pending memory operations must be complete before any locks are acquired. All lock acquisitions are totally ordered. All memory operations performed after lock acquisitions must be completed before locks can be released.

Release consistency maintains memory consistency only on lock releases. In this case all processors get copies of data after acquiring the lock, write back data prior to releasing a lock and flush their data caches.

2. (30%) Let us assume that all caches are initially in cold start state (that is initial access cause misses). Show the cache states using MOSI protocol for the following sequence of memory accesses.

Remember the operation will be represented as P#: <op> <address> [← <value>]

To indicate that processor # is performing a read or write (as <op>) to memory address given by <address> and if the operation is write, the value stored at the address is given by <value>.

P0: read 120
P1: read 120
P1: write 120 ← 80
P0: write 120 ← 60
P1: read 120
P1: read 110
P0: read 110
P0: write 110 ← 90

Key: P0: read	P0: 120 O 110 I	P1: 120 I 110 I
P1: read	P0: 120 O 110 I	P1: 120 S 110 I
P1: write 120 ← 80	P0: 120 I 110 I	P1: 120 M 80 110 I
P0: write 120 ← 60	P0: 120 M 60 110 I	P1: 120 I 110 I
P1: read 120	P0: 120 O 60 110 I	P1: 120 S 60 110 I
P1: read 110	P0: 120 O 60 110 I	P1: 120 S 60 110 O
P0: read 110	P0: 120 O 60 110 S	P1: 120 S 60 110 O
P0: write 110 ← 90	P0: 120 O 60 110 M 90	P1: 120 S 60 110 I

b). We are given that it takes 100ns to get data from memory, 70ns to get data from another cache, 15ns to invalidate data. Calculate the total time needed to complete the above sequence of memory accesses.

Key:

P0: read	100ns	(get data from memory)
P1: read	70ns	(get data from P0 cache)
P1: write 120 ← 80	15ns	(invalidate P0 data)
P0: write 120 ← 60	70ns+15ns	(get data from P1, and invalidate P1 cache)
P1: read 120	70ns	(get data from P0 cache)
P1: read 110	100ns	(get data from memory)
P0: read 110	70ns	(get data from P1)
P0: write 110 ← 90	15ns	(invalidate P1 cache)
Total	525ns	

3. (20%) Given the following 16-bit data, find the values of the parity bits to construct a Hamming Distance- 3 code: 0001 0010 0100 1000

Key:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
P ₁	P ₂	D ₁	P ₃	D ₂	D ₃	D ₄	P ₄	D ₅	D ₆	D ₇	D ₈	D ₉	D ₁₀	D ₁₁	P ₅	D ₁₂	D ₁₃	D ₁₄	D ₁₅	D ₁₆	
0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	0	

(b)

If you received the following two values, assuming that we are using Hamming distance-3 codes, find out if there is an error in the received information and if so locate it and correct it.

1111 0101 1010
0001 0000 1111

Key:

	1	2	3	4	5	6	7	8	9	10	11	12
	P ₁	P ₂	D ₁	P ₃	D ₂	D ₃	D ₄	P ₄	D ₅	D ₆	D ₇	D ₈
received	1	1	1	1	0	1	0	1	1	0	1	0
syndrome	0	0		0				1				

Only parity 4 is in error and the syndrome is 1000 or bit 8 (the P₄ itself) is incorrect.
 The correct value is 1111 01001 1010

	1	2	3	4	5	6	7	8	9	10	11	12
	P ₁	P ₂	D ₁	P ₃	D ₂	D ₃	D ₄	P ₄	D ₅	D ₆	D ₇	D ₈
received	0	0	0	1	0	0	0	0	1	1	1	1
syndrome	0	0		0				0				

There are no errors in parity and the syndrome is 0000 (no correction needed)

4. (15%). This problem is related to the use of queuing theory. Consider that we need to verify user ID and password before allowing wireless access to Eaglenet. We will assume that it takes 10ms to access the database and verify this information. If we assume that we anticipate 100 attempts to connect to Eaglenet per second, how long will it take before a connection is granted?

Key:

Here the service time is 10ms or service rate is $\mu = 1/(10 \times 10^{-3}) = 100$ requests per second
 The arrival rate is $\lambda = 100$

This will lead to saturation. Service rate must be higher than arrival rate.
 Remember expected number of customer in the system is given by

$$E(N) = (\rho / (1 - \rho)) \text{ where } \rho = \lambda / \mu.$$

$$E(N) = (\lambda / (\mu - \lambda))$$

Using Little's law: Response time = (expected number of customers in system) / (arrival rate)
 $= (\lambda / (\mu - \lambda)) * 1 / \lambda = 1 / (\mu - \lambda)$

If you plug in the numbers you will find that the response time is infinite

If change the service time to 1ms then the service rate $\mu = 1000$

Now, with arrival rate of 100, the response time = $1 / (1000 - 100) = 1 / 900 = 0.0011$ seconds
 $= 1.1$ ms

5. (20%) Let us assume that we are using RAID 6 with 4 data disks, one horizontal and one diagonal parity disk (for a total of 6 disks). We use RAID 4 to recover from a single failure and RAID 6 to recover from 2 failures.

Let us assume that the mean time to fail (MTTF) fro any disk is 900,000 hours. Let us assume that it takes 100 hours to recover from one failure (using RAID 4) but 1000 hours recover from 2 failures (using RAID 6).

a). What is the probability that a second failure occurs while recovering from one failure?

Key:

When one failure occurred we 3 working disks (out of 4). Each one can failure with a MTTF of 900,000 hours. The probability that one of the 3 fail in 100 hours is

$$3 * 100 * (1/900,000) = 3.3 * 10^{-4}$$

b). What is the probability that the system fails when a 3rd failure occurs while recovering from two failures?

Key:

With 2 disks already failed we only have 2 working disks and each one can fail with a MTTF of 900,000. The probability that one of them fail in the 1000 hours needed to repair 2 failures is given by

$$2 * 1000 * (1/900,000) = 2.2 * 10^{-3}$$