20F-COMSCIM151B-1 Homework 7

NEVIN LIANG

TOTAL POINTS

46.25 / 60

QUESTION 1

RAID 15 pts

1.1 Increase in needed storage 5 / 5

✓ + **5 pts** **Correct**

+ **3 pts** Correct reasoning

+ **2 pts** Correct RAID 1 storage

+ **0 pts** Incorrect

1.2 Write latency 2.5 / 5

+ **5 pts** Correct

+ **2.5 pts** One incorrect

✓ + **2.5 pts** **Assumed sequential reads and writes**

+ **0 pts** Both incorrect

1.3 Reliability 2 / 5

+ **5 pts** Good

+ **3 pts** Incomplete justification

✓ + **2 pts** **Partially wrong**

+ **1 pts** Mostly wrong

+ **0 pts** Wrong

QUESTION 2

I/O 10 pts

2.1 Transaction per second A 5 / 5

✓ + **5 pts** **Correct**

+ **0 pts** Incorrect

2.2 Transaction per second B 5 / 5

✓ + **5 pts** **Correct**

+ **0 pts** Incorrect

QUESTION 3

Memory bus 10 pts

3.1 Disk transfers 1 5 / 5

✓ + **5 pts** **OK**

+ **0 pts** Incorrect

3.2 Disk transfers 2 5 / 5

✓ + **5 pts** **OK**

+ **0 pts** Incorrect

QUESTION 4

Transpose and add 15 pts

4.1 MIMD 4 / 5

✓ - **0.5 pts** **Invalid RISC-V assembly**

✓ - **0.5 pts** **Wrong accessed addresses**

4.2 False sharing 2 / 5

+ **5 pts** Correct

+ **4 pts** Solution that requires modifying the implemented function

+ **3 pts** Unclear but good solution

+ **3.75 pts** Split then merge

+ **3.75 pts** Two-pass solution

✓ + **2 pts** **Bad solution**

+ **0 pts** Wrong

4.3 SIMD 3.25 / 5

✓ - **1.75 pts** **Impossible vector load/store instruction**

QUESTION 5

5 NUMA 7.5 / 10

+ **10 pts** Correct

✓ + **7.5 pts** **Assumed that processing stops during memory accesses**

+ **0 pts** Incorrect

QUESTION 6

6 Late Submission Penalty 0 / 0

✓ - 0 pts On time

Homework Set - 7

Session: Fall 2020
Total Points: 60

Instructor: Prof. P. Gupta
Due Date: December 3

Instructions

1. **There are two sections in this homework set. The questions under Section – I are for your practice. Solutions to these problems will be provided soon. Students are encouraged to solve the problems before looking at the solution provided.**

2. **Problems under Section – II are to be submitted.**

Section – I

1. Suppose we want to use a laptop to send 100 files of approximately 40 MB each to another computer over a 5 Mbit/sec wireless connection. The laptop battery currently holds 100,000 Joules of energy. The wireless networking card alone consumes 5 watts while transmitting, while the rest of the laptop always consumes 35 watts. Before each file transfer we need 10 seconds to choose which file to send. How many complete files can we transfer before the laptop's battery runs down to zero?

2. Consider the following portions of two different programs running at the same time on four processors in a symmetric multicore processor (SMP). Assume that before this code is run, both x and y are 0.

   Core 1: x = 2;
   Core 2: y = 2;
   Core 3: w = x + y + 1
   Core 4: z = x + y

   What are the possible resulting values of w, x, y, and z? For each possible outcome, explain how we might arrive at those values. You will need to examine all possible interleaving of instructions.
   How could you make the execution more deterministic so that only one set of values is possible?

3. Consider the following specifications of a disk. If the average queuing delay is 20ms, then estimate how much time will it take to read a file of size 100 MB (There is no RAID here). Ignore controller delay.

	Seagate Cheetah Ultra160 SCSI
Diameter	3.5"
Formatted capacity	73.4 GB
Cylinders	14,100
Disks	12
Heads	24
Bytes / sector	512-4096
Sectors / track	~424
RPM	10,033
Seek time	4.0 ms
Data transfer rate	80 MB/s

4. Matrix multiplication plays an important role in a number of applications. Two matrices can only be multiplied if the number of columns of the first matrix is equal to the number of rows in the second. Let's assume we have an m x n matrix A and we want to multiply it by an n x p matrix B. We can express their product as an m x p matrix denoted by AB (or A.B). If we assign C = AB, and c(i,j) denotes the entry in C at position (i,j), then c(i,j) = a(i,1)b(1,j) + a(i,2)b(2,j) + ... + a(i,n)b(n,j) for each element i and j with 1 <= i <= m and 1 <= j <= p. Now we want to see if we can parallelize the computation of C. Assume that matrices are laid out in memory sequentially as follows: a(1,1), a(2,1), a(3,1), a(4,1),..., etc.

a. Assume that we are going to compute C on both a single core shared memory machine and a 4-core shared memory machine. Compute the speedup we would expect to obtain on the 4-core machine, ignoring any memory issues.
b. Repeat (a) assuming that updates to C incur a cache miss due to false sharing when consecutive elements are in a row (i.e., index i) are updated.
c. How would you fix the false sharing issue that can occur?

Section – II

1. Consider two RAID disk systems that are meant to store 15 terabytes of data (not counting any redundancy). System A uses RAID 1 technology and System B uses RAID 5 technology with four disks in a "protection group." **[15 Points]**

   a. How many more terabytes of storage are needed in System A than in System B?

   $15 \cdot 2 = 30$ $\quad 15/4 = 3.75 \quad 15 + 3.75 = 18.75$

   $30 - 18.75 = 11.25$

   11.25

   b. Suppose an application writes one block of data to the disk. If reading or writing a block takes 20 ms, how much time will the write take on System A in the worst case? How about on System B in the worst case?

   $\times 2 \quad \times 4$

System A	40 ms
System B	80 ms

c. Is System A more reliable that System B? Why or why not?

System A is more reliable because it has a copy and less time/ more storage.

2. Here are two different I/O systems intended for use in transaction processing: **[10 Points]**
   a. System A can support 1500 I/O operations per second
   b. System B can support 2000 I/O operations per second

   The systems use the same processor that executes 100 million instructions per second. Assume that each transaction requires 5 I/O operations, and that each I/O operation requires 20,000 instructions. Ignoring response time and assuming that transactions may be arbitrarily overlapped, what is the maximum transaction-per second rate that each machine can sustain?

$1500/5 = 300$

$2000/5 = 400$

System A	300
System B	400

3. Suppose we have a system with the following characteristics:

   a. A memory and bus system supporting block access of 4 or 6 32-bit words.
   b. A 64-bit synchronous bus clocked at 300 MHZ, with each 64-bit transfer taking 1 clock cycle, and 1 clock cycle required to send an address to member.
   c. Two clock cycles needed between each bus operation. (Assume the bus is idle before an access.)
   d. A memory access time for the first four words of 500 ns; each additional set of four words can be read in 20 ns.

   If the I/O is allowed to consume 100% of the bus and memory bandwidth, what is the maximum number of simultaneous disk transfers that can be sustained for the two block sizes? **[10 points]**

300 MHz → mem cycles =

$$\frac{500\cancel{ns} \cdot 3e8}{\cancel{1e9}} = 150 \text{ cycles}$$

155 cycles total

$$\frac{32 \cdot 4 \cdot 3e8}{8 \cdot 155 \cdot 1e6} = \cancel{30}.31 \text{ MB/sec.}$$

$$\frac{4 \cdot 32 \cdot 1e9}{8 \cdot 500e{-9} \cdot 1e6} = 32 \text{ MB} \quad \longrightarrow \boxed{1}$$

157 cycles.

$$\boxed{1}$$

$$\frac{32 \cdot 6 \cdot 3e8}{8 \cdot 157 \cdot 1e6} \Big/ \frac{6 \cdot 32 \cdot 1e9}{8 \cdot 500 \cdot 1e6} = 461 \text{ MB/se.}$$

System1,4-word block: 1

System2,6-word block: 1

4. We would like to execute the loop below as efficiently as possible. We have two different machines, an MIMD machine and a SIMD machine (The arrays are of type int). **[15 points]**

```
for (i = 0; i < 3000; i++)
        for (j = 0; j < 4000; j++)
            X_array[i*4000+j] = Y_array[j*3000+i] +200;
```

a. For a four CPU MIMD machine, what is the speed-up for this MIMD machine against a single core machine? Assume memory latency is not an issue here. Write a RISC-V code for this program that will run on each core and show the bounds of the variables i and j in each core.

4x speed up.

Let x18 contain & X_array and x19 = & Y_array

```
ADDi  x28,x0,x9
ADDi  x30,x28,750
ADDI  X31, x0,4000
L1:
AND   x29,x29,x0
L2:
MULT  x5, x29, 3000
ADD   x5, x5,x28
ADD   x5, x5,x19
LW    x5,x5,0
ADDI  x5, x5, 200
MULTI x6, x28,4000
ADDI  x6,x6, x29
ADD   x6, x6, x18
SW    x6, x5, 0
```

```
ADDI  x29, x29, 1
BLT   x29, x31, L2
ADDI  x28, x28,1
BLT   x28, x30, L1
```

X9 is 0,750,1500, or 2250 depending on core.

b. Would there be an issue with false sharing in this program? Explain a way to avoid it.

w/ false sharing many diff things could invalidate things in memory

can resolve with cache-coherence protocol. (snooping)

c. For an 8-wide SIMD machine, what is the ratio of the number of instructions executed on the SIMD machine to the MIMD machine. For that assume your own SIMD RISC-V instruction and show the code here. You can assume you have access to a separate register file with 8-element vector registers y0-y31.

1:8

x18 : & X_array   x19 : & Y_array.

```
  AND      x28, x28, x0
  ADDi     x30, x0, 3000
  ADDi     x31, x0, 4000
L1
  AND      x29, x29, x0
L2
   MULTI        x5, x29, 3000
   ADDI         x5, x5, x28
   ADDI         x5, x18, x5
   LW-SIMD      y5, x5, 0
   ADDI-SIMD    y5, y5, 200
   MULTI        x6, x28, 4000
   ADDI         x6, x6, x29
   ADDI         y6, x19, x6
   SW-SIMD      y5, y6, 0

   ADDi         x29, x29, 8
   BLT          x29, x31, L2
  ADD           x28, x28, 1

  BLT           x28, x30, L1
```

5. On a CC-NUMA system, the cost of accessing non-local memory can limit our ability to use multiprocessing effectively. The following table shows the costs associated with data access in local memory versus non-local memory and the locality of our application expressed as the proportion of accesses that are local.

Local load/store (cycle)	Non-local load/store (cycles)	% local accesses
20	200	25

Answer the following questions. Assume that memory accesses are evenly distributed through the application, and that we can continue processing when a memory access is active (no true dependencies). Also, assume that only a single memory operation can be active during any cycle. Assume non-memory operations are executed in 1 cycle.

What is the CPI of the machine (single thread) if on average we need to access memory once every 100 cycles? **[10 points]**

$0.99 \cdot 1 + 0.01\,(.25 \cdot 20 + .75 \cdot 200)$

$= 0.99 + 0.01\,(5 + 150)$

$$= \begin{array}{r} 0.99 \\ + 1.55 \\ \hline 2.54 \end{array}$$

$\boxed{2.54}$