THE AUSTRALIAN NATIONAL UNIVERSITY
First Semester Examination – June 2008

COMP3320/6464/HONS
High Performance Scientific Computing

Study Period: 15 minutes
Time Allowed: 3 hours
Permitted Materials: Calculator
Exam questions total 100 marks.

COMP3320 students answer questions 1-5
COMP6464/Honours students answer questions 1-4 and 6
Clarity and conciseness in answers will be highly valued
Marks may be lost for supplying irrelevant information
Question 1 [30 marks]

Short Questions

(a) (i) Explain why the divergent infinite series:

\[ \sum_{n=1}^{\infty} \frac{1}{n} \]

can have a finite sum in floating-point arithmetic.

(ii) At what point would the partial sum cease to change? [4 marks]

(b) Most compilers translate code into an intermediate representation (IR) that is then subjected to optimisation and eventual code generation. Usually the IR incorporates the core features of the RISC architecture.

(i) Give two core RISC features that the IR might try to model.

(ii) Illustrate your answer with small bits of C code and their corresponding IR. [4 marks]

(c) State two fundamental problems that a compiler has in exposing parallelism from an ordinary serial program. [2 marks]

(d) Modern compute systems make widespread use of caches for storing data and instructions. The translation lookaside buffer (TLB) is also a type of cache.

(i) For each of the following cache types explain the circumstances under which you would expect to observe poor cache usage.
   - Instruction cache
   - Data cache
   - TLB

(ii) For any one of the above cache types provide a few lines of C code that will exhibit poor cache performance. Include in your answer some explanation as to why your code performs poorly for the given cache type, and as necessary define relevant parameters for the cache in question. [9 marks]

(e) In their paper, *The Anatomy of the Grid*, Ian Forster, Carl Kesselman and Steven Tuecke state: “Grid computing has emerged as an important new field, distinguished from conventional computing by its focus on large-scale resource sharing, innovative applications, and in some cases, high-performance orientation”. Outline one scenario for which grid computing might be useful. [2 marks]
(f) To evaluate the expression:

\[ x = 2.785 \times \prod_{i=1}^{N} \frac{3i - 1}{3i + 1} \]

you have written the following OpenMP code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

int main(int argc, char* argv[]) {
    int nterms, ncpus, istart, iend, nthrd, iam;
    nterms = atoi(argv[1]);
    ncpus = atoi(argv[2]);
    if (nterms <= 0 || ncpus <= 0)return -1;
    omp_set_num_threads(ncpus);
    double x;
    nterms, ncpus, istart, iend, nthrd, iam;
    nterms = atoi(argv[1]);
    ncpus = atoi(argv[2]);
    if (nterms <= 0 || ncpus <= 0) return -1;
    x = 2.785;
    #pragma omp parallel default(none) firstprivate(nterms) \
    reduction(*:x) private(iam, nthrd) shared(istart, iend)
    {
        nthrd = omp_get_num_threads();
        iam = omp_get_thread_num();
        istart = iam * (nterms / nthrd) + 1;
        iend = (iam + 1) * (nterms / nthrd);
        do {
            x *= (3.0 * istart - 1.0) / (3.0 * istart + 1.0);
            #pragma omp atomic
            istart++;
        } while (istart <= nterms);
    }
    printf(" Number of Terms = %d Value = %20.12e\n", nterms, x);
    return 0;
}
```

The code compiles and runs correctly with `ncpus = 1`, but gives wrong results when `ncpus > 1`.

(i) Detail as many errors as possible with the above code. (You will get negative marks for giving false errors).

(ii) Modify the code so that it works correctly with `ncpus > 1`, while also partitioning work across the available CPUs. Your revised code MUST maintain the use of the `do/while` construct. You will be marked initially on the correctness of your code, and then if your code is correct, its likely scaling.

[9 marks]
Question 2 [20 marks]

Performance Modelling and Measurement

(a) You have built a computer system that contains a special purpose processor for performing floating-point operations. You have determined that 60% of your computation can use this special purpose processor. When this processor is used, it is 1.5 times faster for floating-point operations compared to the standard processor.

(i) What is the overall speedup obtained by using the special purpose floating point processor in the above computation.

(ii) In order to improve speedup further you are considering two options:

- Option 1: modifying the code/compiler so that 90% of the computation can use the special floating-point processor. The cost of this option is estimated as $50K.
- Option 2: modifying the special floating-point processor so that it is twice as fast as the standard processor. The cost for this option is estimated as $60K.

Which option would you recommend? Justify your answer quantitatively by comparing speedups for $ spent.

[8 marks]

(b) Suppose you have a load/store computer that can issue one instruction per cycle, but cannot issue the next instruction until the previous one is complete. The computer runs the following instruction mix with that have the following times to complete:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Frequency</th>
<th>No of Cycles to Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic (ALU)</td>
<td>35%</td>
<td>1</td>
</tr>
<tr>
<td>Loads</td>
<td>25%</td>
<td>3</td>
</tr>
<tr>
<td>Stores</td>
<td>15%</td>
<td>3</td>
</tr>
<tr>
<td>Branches</td>
<td>25%</td>
<td>4</td>
</tr>
</tbody>
</table>

(i) Compute the overall number of cycles per instruction (CPI). Show exactly how you derive your result.

(ii) You observe that 40% of the ALU operations are paired with a load, so you propose to replace these ALU operations and their associated loads with a single new instruction. The new instruction takes 2 clock cycle to complete, but in the revised architecture branches now take 6 cycles to complete. What is the new CPI value? Show exactly how you derive your new CPI value.

(iii) If the ratio of clock speeds is $Clock_{old}/Clock_{new} = 1.2$, which processor has the faster execution time and by what percent? Show exactly how you derive your result.

[12 marks]
Question 3 [20 marks]

High Performance Computer Architecture

(a) (i) In the context of modern microprocessors what is pipelining?
(ii) Under what circumstances, if any, is it better to have a 5 stage pipeline with each stage taking 2ns, compared to a 15 stage pipeline with each stage running at 1ns? Under what circumstances is the opposite true?
(iii) What is software pipelining? Include in your answer an example.

[10 marks]

(b) Consider the following memory system:

- The cache line size is 4 words
- The cache uses a write allocate policy, i.e. on a write miss the relevant cache line worth of data must first be read from memory into the cache before the write can proceed.
- The cache is write through, i.e. the store of a word propagates that change through to main memory. (Only the word is propagated, not the entire cache line).
- The bus can transfer data to or from memory at a rate of $10^6$ words/sec. (Only in one direction at any given time.)

The following behaviour is observed when running a certain application.

- The CPU sends memory requests (loads or stores of one word) to the cache at a rate of $10^6$ requests/sec, with 25% of the requests being writes.
- 90% of all memory requests issued by the CPU are resolved by data already in the cache (i.e. the cache hit ratio is 90%).

(i) Given the above compute the bus traffic generated (in words/sec) by the following events:
- Read hits.
- Read misses.
- Write hits.
- Write misses.

(ii) What percentage of the total available bus bandwidth ($10^6$ words/sec) is used?

(iii) What additional information do you require in order to estimate the percentage of the total available bus bandwidth used if the cache were write back?

[10 marks]
Question 4 [20 marks]

Distributed Memory Parallelism

(a) In assignment 2, you wrote code to simulate the motion of a piece of cloth. The cloth was modelled as a 2-dimensional grid of point masses, where each mass is linked to neighbouring masses by springs. The initial grid configuration and a sample of the spring connectivity is shown below:

With some simplifications the code required to evaluate the force \((\text{fx}, \text{fy}, \text{fz})\) on each mass, and the total potential energy of the system \((\text{pe})\) is given on the following page. The grid is assumed to be square of size \((n \times n)\), initially located in the \(xy\) plane with a distance of \(\text{sep}\) between adjacent masses in the \(x\) or \(y\) direction. The force constant for each spring is \(\text{fcon}\) with all springs at equilibrium (zero force) for the initial cloth geometry. A given mass is connected via springs to all other masses located within \(\text{delta}\) neighbours in the \(x\) or \(y\) direction (shown above for \(\text{delta} = 1\)).

(i) In the assignment you parallelised this code for a shared memory computer using OpenMP. Outline here how you would parallelise this code for a distributed memory parallel computer using MPI. You must clearly indicate what data you expect to be resident in the memory of each process, and when data is being passed between processes. You are not required to reproduce the exact MPI calling syntax, but you are required to make the intent of your MPI calls clear. Use the line numbers provided to indicate where you would make changes, and for referencing unmodified code. You are free to re-write the code and/or to allocate additional memory for temporary arrays if you believe this would lead to better overall performance (but still give the correct results).

[10 marks]

(ii) Discuss the performance of your parallel code as a function of the various simulation parameters, number of MPI processes, network performance etc.

[10 marks]
pe=0.0;
for (nx=0;nx<n;nx++){
    for (ny=0;ny<n;ny++){
        fx[nx*n+ny]=0.0;
        fy[nx*n+ny]=0.0;
        fz[nx*n+ny]=0.0;
        // loop over displacements
        for (dx=MAX(nx-delta,0);dx<MIN(nx+delta+1,n);dx++){
            for (dy=MAX(ny-delta,0);dy<MIN(ny+delta+1,n);dy++) {
                // exclude self interaction
                if (nx!=dx || ny!=dy){
                    // compute reference distance
                    rlen=sqrt((double)((nx-dx)*(nx-dx)+(ny-dy)*(ny-dy)))*sep;
                    // compute actual distance
                    xdiff = x[dx*n+dy]-x[nx*n+ny];
                    ydiff = y[dx*n+dy]-y[nx*n+ny];
                    zdiff = z[dx*n+dy]-z[nx*n+ny];
                    vmag=sqrt(xdiff*xdiff+ydiff*ydiff+zdiff*zdiff);
                    // potential energy and force
                    pe += fcon*(vmag-rlen)*(vmag-rlen);
                    fx[nx*n+ny]+=fcon*xdiff*(vmag-rlen)/vmag;
                    fy[nx*n+ny]+=fcon*ydiff*(vmag-rlen)/vmag;
                    fz[nx*n+ny]+=fcon*zdiff*(vmag-rlen)/vmag;
                }
            }
        }
    }
}
Question 5 [10 marks]

(a) You have written code in C to perform some type of scientific simulation (the nature of simulation is not important). You have performed some useful computations, but now find that the time required to complete your latest simulations is too long. You are considering the following three options in order to achieve faster turn around times:

(i) Better optimizing your code for your existing hardware.
(ii) Adding one of the latest NVIDIA graphics cards to the existing machine and off-loading some of the computation to the graphics card.
(iii) Purchasing a new computer that has a higher value for the SPEC performance benchmark.
(iv) Using spare cycles on other identical computers at your establishment and parallelising your code to run across the local area network over all these machines.

Discuss each of the above options, giving at least one positive and one negative for each approach. Your reasons should be different in each case, with the positive and negative reasons also being different. As necessary you should qualify your arguments by stating under what circumstances your comments will hold.

[10 marks]
Question 6  [10 marks]

(a) In lab 2 you investigated the Stream benchmark for measuring memory bandwidth. Stream has four components: copy, scale, add, triad. In copy elements of one array are copied into another (a[i] = b[i]).

(i) Sketch C code for the other three components of Stream.  

(ii) How does Stream ensure that the measured bandwidth corresponds to that associated with retrieving data from main memory, and not from the data cache.  

(iii) The author of Stream defines something called the “Machine Balance”, that is stated to be the “relative cost of memory accesses vs arithmetic”. What is this, and how would you compute this for any one of the four Stream components.  

(iv) Sketch how you would design a benchmark to measure memory latency, rather than memory bandwidth.  
