

THE AUSTRALIAN NATIONAL UNIVERSITY
First Semester Examination – June 2006

COMP3320/COMP6464/Honours
High Performance Scientific Computing

Study Period: 15 minutes

Time Allowed: 3 hours

Permitted Materials: Calculator

Exam questions total 70 marks.

COMP3320 students answer questions 1-6

COMP6464 students answer questions 1-5 and 7

Honours students answer questions 1-5 and 8

Clarity and conciseness in answers will be highly valued
Marks may be lost for supplying irrelevant information

Question 1 [8 marks]

Fundamentals of Numerical Computing

(a) Floating point computational science applications invariably have to deal with the effects of both truncation and rounding errors. Usually, however, one is dominant.

(i) Define precisely what is meant by truncation and rounding errors.

(ii) Explain this statement, providing a simple illustrative example.

[4 marks]

(b) In a floating-point number system having an underflow level of $\text{UFL} = 10^{-38}$, which of the following computations will incur an underflow? Explain your answer.

i) $a = \sqrt{(b^2 + c^2)}$, with $b = 1$ and $c = 10^{-25}$

ii) $a = \sqrt{(b^2 + c^2)}$, with $b = 10^{-25}$ and $c = 10^{-25}$

iii) $u = (v \times w)/(y \times z)$, with $v = 10^{-15}$, $w = 10^{-30}$, $y = 10^{-20}$ and $z = 10^{-25}$.

In cases where underflow occurs, is it reasonable to simply set to zero the quantity that underflows?

[4 marks]

Question 2 [10 marks]

Performance Modeling and Measurement

(a) An application contains the following C routine:

```
double parallel_tasks(double a[], double b[], double c[], int n){
    double xtmp, sum;
    int k;
    sum = 0.0;
    for (k=0; k<n; k++){
        xtmp = do_task(a[k], b[k], c[k]);
        sum = sum + xtmp;
    }
    return sum;
}
```

You are told that function `do_task` is a pure function, returning a double precision result that depends only on the values of the arguments passed to it. Thus the above code can be correctly parallelized by inserting the following OpenMP directives:

```
    sum = 0.0;
#pragma omp parallel for private(xtmp) shared(a,b,c,sum)
    for (k=0; k<n; k++){
        xtmp = do_task(a[k], b[k], c[k]);
#pragma omp critical
    {
        sum = sum + xtmp;
    }
}
```

Discuss how this piece of OpenMP enabled code might perform. You are expected to consider a variety of possible scenarios that might arise depending on, for example, the value of `n`.

[6 marks]

(b) Modern processors usually include hardware performance counters.

- (i) The number of cycles executed and instructions issued are examples of two quantities that might be counted using hardware performance counters. Define two further events that you might reasonably expect to count using hardware performance counters.
- (ii) You attempt to measure cycles executed and instructions issued for a section of your code using two 32-bit hardware performance counters on a 3GHz Intel Pentium IV processor. You find that for small problem sizes your results are reasonable, but that as you increase the problem size you begin to get obviously wrong results for the instruction count. Expanding the problem size even further gives wrong results for both the instruction and cycle counts. Explain what is happening.

[4 marks]

Question 3 [16 marks]

High Performance Computer Architecture

- (a) The Itanium processor uses explicitly parallel instruction computing (EPIC).
- (i) Explain what EPIC is.
 - (ii) Why is the compiler so critical to achieving good performance on EPIC architectures?

[2 marks]

- (b) Modern processors include a special cache called the Translation Lookaside Buffer (TLB).
- (i) What information is cached by the TLB?
 - (ii) For some applications, using a larger memory page size can give rise to better TLB operation. Why is this the case, and what are the typical characteristics of an application that is likely to benefit from large memory pages?
 - (iii) Give one disadvantage associated with using large memory page sizes.
 - (iv) What is a typical memory page size anyway?

[6 marks]

- (c) Discuss the performance of the following C routine on a typical modern day processor:

```
void matvec(int m, int n, double x[], double a[], double b[], double c){
    int i,j;
    for (i=0; i<m; i++){
        for (j=0; j<n; j++){
            x[i] += (a[i*n+j]-c*b[j])*(a[i*n+j]+c*b[j])/2
        }
    }
}
```

Your discussion should include estimates of the likely cycles per loop iteration, and how this might change as a function of m and n . Any architectural details that you assume in order to derive your estimates must be clearly identified, as should any assumptions you make concerning data layout. If there are modifications to the code as written that are likely to arise from optimizations undertaken by the compiler you must clearly identify these.

[8 marks]

Question 4 [12 marks]

Shared Memory Parallelism

- (a) UNIX creates processes using the `fork` system call. OpenMP creates threads using the `#pragma omp parallel` directive.
- (i) Explain the difference between a process and a thread.
 - (ii) What does it mean to say that a system or library routine is thread safe?
 - (iii) Recently Intel introduced a range of processors that included hyperthreading. What is a “hyperthreaded processor”? Do you require a hyperthreaded processor to run OpenMP code? If so, why? If not, do hyperthreaded processors offer advantages to OpenMP programs?

[4 marks]

- (b) In the following C code the arrays `f`, `d` and `val` correspond to distinct memory regions.

```
void mystery(double f[], double d[], double val[], int n){
    int i,j,k,l,ij,kl;
    ij=0;
    for (i=0; i<n; i++){
        for (j=0; j<i; j++){
            ij++;
            kl=0;
            for (k=0; k<n; k++){
                for (l=0; l<k; l++){
                    kl++;
                    f[i*(i-1)/2+j] += d[k*(k-1)/2+l]*val[ij*(ij-1)/2+kl];
                }
            }
        }
    }
    return;
}
```

- (i) How does the computational cost of the above code scale as a function of n ?
- (ii) Outline how you would parallelise this code on a shared memory parallel computer using OpenMP. You are not required to reproduce the exact OpenMP syntax, but you are required to make the intent of your directives clear. 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).

[8 marks]

Question 5 [17 marks]

Distributed Memory Parallelism

- (a) Beowulf clusters are popular distributed memory computing platforms.
- Describe two characteristics of Beowulf cluster style computers.
 - Torque* is a widely used resource manager for cluster computers. Give one example of the sort of functionality a resource manager might provide.
 - How does a cluster computer resource manager differ from a cluster computer resource monitor?

[3 marks]

- (b) The MPI message passing model is widely used on cluster computers.
- MPI provides blocking send and receive message passing communications. How do blocking communications differ from synchronous and asynchronous communications.
 - Compared to MPI-1, MPI-2 introduced three major new features. Describe any one of these new features.

[4 marks]

- (c) The following code fragment solves the heat distribution problem on a square grid:

```
for (iter=0; iter<max_iter; iter++){
  for (i=1; i<n-1; i++){
    for (j=1; j<n-1; j++){
      Agrid[i][j]=(Bgrid[i-1][j]+Bgrid[i+1][j]+
                  Bgrid[i][j-1]+Bgrid[i][j+1])/4.0;
    }
  }
  for (i=1; i<n-1; i++){
    for (j=1; j<n-1; j++){
      Bgrid[i][j]=(Agrid[i-1][j]+Agrid[i+1][j]+
                  Agrid[i][j-1]+Agrid[i][j+1])/4.0;
    }
  }
}
```

where *Agrid* and *Bgrid* are distinct $n \times n$ arrays.

- Outline how you would parallelise this code on a distributed memory parallel computer using MPI. You are not required to reproduce the exact MPI calling syntax, but you are required to make the intent of your MPI calls clear. 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).
- Discuss the performance of your parallel code as a function of input data size, number of MPI processes, network performance etc.

[10 marks]

COMP3320 STUDENTS ONLY

Question 6 [7 marks]

(a) 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.”

(i) Outline one scenario for which grid computing might be useful.

(ii) Foster *et al.* propose a grid architecture that has five components: application, collective, resource, connectivity and fabric. Pick any two of these and outline what is involved in each.

[4 marks]

(b) In this course it was stated that “Physics drove computing capacity in the last 20 years, biology will do the same in the next 20 years”. Outline how physics drove computing developments in the last 20 years, and argue either for or against the statement that biology will do the same in the coming 20 years.

[3 marks]

(c) Optional feedback — but only if you want to and have time:

(i) I would rate the exam as:

1. Very hard, with over 80% of the questions being very challenging.
2. Hard, with over 50% of the questions being very challenging.
3. Moderate, with about 30% of the questions being very challenging, 40% moderate, and the rest relatively easy.
4. Fairly easy, with over 50% of the questions being relatively straight forward.
5. Too easy, with very few challenging questions.
6. Other, please elaborate.

(ii) In this exam I estimate that I will score:

1. $\geq 80\%$
2. $\geq 70\%$
3. $\geq 60\%$
4. $\geq 50\%$
5. $< 50\%$ but $\geq 45\%$ so I'll be back for a supplementary (please make it easy!).
6. Too low to estimate!

[0 marks]

COMP6464 STUDENTS ONLY

Question 7 [7 marks]

(a) Give two advantages and two disadvantages of OpenMP over MPI.

[2 marks]

(b) In the following loop, `a`, `b` and `c` are all of type `double`:

```
for (i=0; i<n; i++){
    a[i] = a[i] + b[i]*c;
}
```

There are no dependencies and the loop can in principle be run in parallel on a shared memory parallel computer by inserting the following OpenMP directive:

```
#pragma omp for schedule(type[,chunk])
```

Discuss how you would go about measuring the overhead of the above OpenMP directive, and how you would then use this information to determine whether or not it was worth while parallelising the above code.

[5 marks]

(c) Optional feedback — but only if you want to and have time:

(i) I would rate the exam as:

1. Very hard, with over 80% of the questions being very challenging.
2. Hard, with over 50% of the questions being very challenging.
3. Moderate, with about 30% of the questions being very challenging, 40% moderate, and the rest relatively easy.
4. Fairly easy, with over 50% of the questions being relatively straight forward.
5. Too easy, with very few challenging questions.
6. Other, please elaborate.

(ii) In this exam I estimate that I will score:

1. $\geq 80\%$
2. $\geq 70\%$
3. $\geq 60\%$
4. $\geq 50\%$
5. $< 50\%$ but $\geq 45\%$ so I'll be back for a supplementary (please make it easy!).
6. Too low to estimate!

[0 marks]

HONOURS STUDENTS ONLY

Question 8 [7 marks]

- (a) Molecular dynamics simulations employing pairwise interaction potentials often use cutoffs to give better performance.
- (i) Give one example of an interaction potential that is well suited for use with a cutoff. Give another example of an interaction potential that is poorly suited for use with a cutoff. Explain your choices.
 - (ii) Outline exactly how use of a cutoff can reduce the computational cost of a molecular dynamics simulation.
 - (iii) A colleague plans to write their own molecular dynamics simulation code in Java. What advice can you offer him/her on using Java for numerical computation.

[7 marks]

- (b) Optional feedback — but only if you want to and have time:

- (i) I would rate the exam as:
 1. Very hard, with over 80% of the questions being very challenging.
 2. Hard, with over 50% of the questions being very challenging.
 3. Moderate, with about 30% of the questions being very challenging, 40% moderate, and the rest relatively easy.
 4. Fairly easy, with over 50% of the questions being relatively straight forward.
 5. Too easy, with very few challenging questions.
 6. Other, please elaborate.
- (ii) In this exam I estimate that I will score:
 1. $\geq 80\%$
 2. $\geq 70\%$
 3. $\geq 60\%$
 4. $\geq 50\%$
 5. $< 50\%$ but $\geq 45\%$ so I'll be back for a supplementary (please make it easy!).
 6. Too low to estimate!

[0 marks]
