Numerical Simulation and Scientific Computing I
360.242 -- Exam

Jan 28, 2021

Written Online Exam

General Information:

- *(**x**p) indicates the achievable points for the respective question.*
- *You must answer the questions via hand-written notes on empty sheets of paper.*
- *Show the stack of empty papers at the beginning when asked.*
- *Clearly indicate the respective question number for each answer.*
- *Add page numbers to every sheet of paper.*
- *Note the total number of submitted pages on the top of the first page.*
- *Write clearly using large hand-writing and generously use the available space.*
- *When finished notify a supervisor that you are ready to submit. You will be moved to a breakout room where you will be instructed to scan all the sheets of paper into a single PDF document. Double-check that the PDF contains all pages and is of reasonable quality and submit to: nssc@iue.tuwien.ac.at*

Inform an exam supervisor when

- *You are finished and want to scan and submit.*
- *You need more paper then initially approved by the supervisor.*
- *Something unforeseen happened.*

1. (2p) Consider the schematic of the AMD EPYC 7002 processor (Figure 1), the corresponding detailed schematic of one of the 8 green-colored so-called "core-complex dies (CCDs)" each composed of 8 black-colored "Zen2 cores" (Figure 2), and the information in Figure 3 below.
   (a) Which caches (L2-L3) are "exclusive" to a "Zen2 core" and which are shared among them?
   (b) How many "Zen2 cores" have access to how much shared cache?
   (c) Does this processor support SMT?
   (d) Would you consider this a "UMA" or a "NUMA" architecture?
   <u>*Justify each answer!*</u>

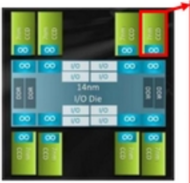


Figure 1

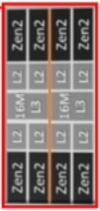
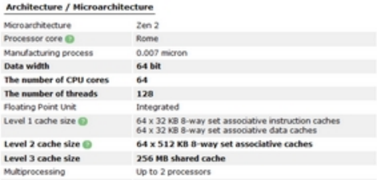


Figure 2



Architecture / Microarchitecture

Microarchitecture	Zen 2
Processor core	Rome
Manufacturing process	0.007 micron
Data width	**64 bit**
The number of CPU cores	**64**
The number of threads	**128**
Floating Point Unit	Integrated
Level 1 cache size	64 x 32 KB 8-way set associative instruction caches 64 x 32 KB 8-way set associative data caches
Level 2 cache size	**64 x 512 KB 8-way set associative caches**
Level 3 cache size	**256 MB shared cache**
Multiprocessing	Up to 2 processors

Figure 3

2. (3p) Explain cache coherence generally and schematically draw the MESI protocol and explain the procedure.

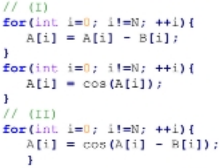
3. (4p) The main memory bandwidth of a cache-based stored-program x86 microprocessor is commonly available from its datasheet. Explain conceptually and step-by-step how to proceed if bandwidth-based modeling is used to approximate the effective bandwidths for different cache levels (which are typically not available from datasheets).

Written Online Exam

4. (3p) Cache-line:
   (a) Explain what a cache-line is and provide the typical size of a cache-line.
   (b) In view of this, discuss the influence of the data layout in memory on the performance of a benchmark adding two matrices based on the pseudocode below.

```
do i=1,N
  do j=1,N
    A[i,j] = B[i,j] + C[i,j]
  enddo
enddo
```

5. (3p) Consider the code snippets (I) and (II) below which perform the same two operations in two distinct loops in (I) and combined in a single loop in (II). Discuss if and why you expect a difference in performance considering large values of N.



```
// (I)
for(int i=0; i!=N; ++i){
    A[i] = A[i] - B[i];
}
for(int i=0; i!=N; ++i){
    A[i] = cos(A[i]);
}
// (II)
for(int i=0; i!=N; ++i){
    A[i] = cos(A[i] - B[i]);
    }
```

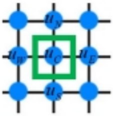
6. (2p) List three vector norms (with formulas) which lead to induced matrix norms. Comment on the computational demands of these induced matrix norms.

7. (4p) Consider the regular finite difference discretization of the 2D Laplace equation on a square domain using second order central differences and a constant grid spacing h:

$$-\Delta u = -(u_{xx} + u_{yy}) = 0$$

$$-\left(u_{xx} + u_{yy}\right) \approx -\frac{1}{h^2}\left(u_N + u_S + u_E + u_W - 4u_C\right) = 0$$

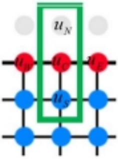
An interior point u_C:



Provide the discretization for a point u_C on the north boundary for a mixed boundary condition (see below) using a central difference to approximate the boundary condition.

$$\frac{\partial u}{\partial n} + \alpha u = g_3$$

A point u_C on the north boundary:



8. (2p) The IEEE floating point standards include the “guarantee” that the result (rounded to finite precision) of the basic binary mathematical operators applied to two finite precision representations is identical to the result when performing the operation with infinite precision and subsequent rounding. Explain why IEEE compatible hardware implementations typically make use of extra bits (guard bit, round bit and sticky bit) when performing these operations.

9. (2p) Which of the following statements are true regarding preconditioning? More than one statement might be true.
   a. Preconditioning will always improve convergence.
   b. The preconditioning algorithm can change the Krylov subspace from which the approximation is constructed.
   c. Preconditioning can only be applied to the CG method.
   d. Preconditioners can improve convergence by reorganizing the spectrum of the matrix.

10. (4p) Given the following A matrix in the CCS format:

$$V = [\,4.5 \quad 1.2 \quad 3.7 \quad 8.3 \quad 5.7 \quad 8.6 \quad 5.9 \quad 7.3]$$

$$IA = [\,1 \quad 4 \quad 2 \quad 0 \quad 3 \quad 1 \quad 4 \quad 5]$$

$$JA = [\,0 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 8]$$

Write A in the conventional, dense square-brackets format including zero entries. Calculate its memory requirement in bytes for both the CCS and dense representations, assuming double precision floating point numbers and 32-bit integers.

11. (2p) In a certain random sequence, the following numbers were drawn in order:

$$[\ldots, 54, 789, 1096, \ldots]$$

Following the same sequence, the number 789 was drawn again. What can you state about the number which will succeed 789 assuming that the random number generator is a:

   a. Linear Congruential Generator?
   b. Mersenne Twister?

12. (3p) Which of the following STL containers are typically implemented using Binary Search Trees (BSTs)? More than one answer is possible.
   a. std::array
   b. std::set
   c. std::unordered_map
   d. std::forward_list
   e. std::map

   Additionally, compare BSTs and hash tables. Give at least one advantage and one disadvantage.

13. (4p) Element insertion:
   a. Describe the algorithm to insert an element in the middle of an array.
   b. Express the asymptotic behavior of the algorithm in subtask (a.) as a function of the array size N in “Big O” notation. How does this compare to inserting an element in a linked list?
   c. Compare arrays and linked lists with respect to memory footprint.

14. (3p) Consider the code snippet below, which contains a bug. Identify and explain the bug and discuss an approach to fix it.

```
long i, N = 10;
long best_cost = 0;
#pragma omp parallel for nowait shared(best_cost)
for(i=0; i<N; i++)
{
  long my_cost;
  my_cost = estimate(i);
  if(best_cost < my_cost)
    best_cost = my_cost;
}
```

15. (2p) Consider the code snippet below. What is the value of x2?

```
long i  = 0;
long n  = 4;
long x  = 0;
long x2 = 0;
std::vector<long> a(n);
#pragma omp parallel for lastprivate(x)
for(i = 1; i <= n; i++)
{
  x = i;
  a[i] = x;
}
x2 = x;
std::cout << "x2: " << x2 << std::endl;
```

16. (3p) Define *conformal mesh* and draw and discuss three examples of mesh conformity errors.