NTNU
Norwegian University of
Science and Technology

Department of Mathematical Sciences

Examination paper for
TMA4280 Introduction to Supercomputing

Academic contact during examination: Aurélien Larcher

Phone: 45674644

Examination date: May 16. 2018

Examination time (from–to): 09:00–13:00

Permitted examination support material: B: All printed and handwritten aids permitted. Specific, simple calculator permitted.

Other information:
The examination is divided into three problems and evaluated on a total of 60 points. All the answers should be motivated:

1. Calculatory results should always be supported by a proof with intermediate steps.
2. Whenever specified in the question, interpretation of the results counts as part of the answer.
3. Specify any assumption made to support the answer.

Incomplete answers will not receive full points.

Language: English

Number of pages: 7

Number of pages enclosed: 0

Checked by:

Informasjon om trykking av eksamensoppgave	
Originalen er:	
1-sidig ☐	**2-sidig** ☒
sort/hvit ☐	**farger** ☒
skal ha flervalgskjema ☐	

Date Signature

Problem 1 Sparse Matrix-Vector Multiplication (SpMV) *(24 points).* The following problem focuses on the development of a linear algebra package for sparse systems. The development platform is a cluster of compute nodes with a topology described in Figure 1: they are based on two 64-bit Intel E5-2630v4 processors with a clock frequency of 2.2GHz, for which characteristics are given in Table 1.

The computation of matrix-vector products $\boldsymbol{y} = \mathrm{A}\boldsymbol{x}$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and $\mathrm{A} \in \mathbb{R}^{n \times n}$ in double precision is first considered in serial.

a) *(2 points)* Explain how real numbers are stored using the IEEE 754 double-precision binary floating-point format. What is the maximum number of fractional digits that can be represented with this format?

b) *(4 points)* Suggest two algorithms to perform the matrix-vector multiplication. How can the memory layout influence the choice of algorithm? How should the implementation of the matrix be modified for the sparse case?

c) *(2 points)* Estimate the computational complexity when A is a dense matrix. What is the ratio between computational cost and data movement? How is it affected when the matrix is sparse?

The new implementation should be parallelized to take advantage of modern architectures.

d) *(6 points)* Choose one of the algorithms for computing a sparse matrix-vector product and sketch a hybrid MPI-OpenMP parallelization: highlight the communication pattern required and cost estimate.

e) *(2 points)* Describe two characteristics of the NUMA architecture of Figure 1 that are important for parallelism. Discuss how to choose the parameters for running hybrid MPI-OpenMP computations.

For an application in molecular dynamics a single-precision version of the library is prepared. To improve the performance of the implementation is modified to take advantage of the AVX SIMD unit.

f) *(2 points)* Explain the purpose of Flynn's taxonomy and which type of computer architecture is implemented by AVX? How is it relevant for parallelism?

g) *(2 points)* The AVX unit uses sixteen YMM registers of 256-bits each. How many single-precision and double-precision numbers can be stored per register?

The following matrix-vector benchmark is run on one compute node:

```
 [aurelila@lille-login2  ]$ ./avxtest
Time taken:  26680 seconds
46.6457, 54.6938, 52.5235, 46.8823, 47.7327, 53.5059, 44.7574, 53.7893,
50.923, 47.9733, 50.9802, 51.735, 57.4111, 50.2515, 48.7359, 50.6785,
54.7976, 52.3944, 49.6013, 48.1748, 51.2003, 43.9086, 49.5595, 48.0124,
Time taken:  3040 seconds
46.6457, 54.6938, 52.5235, 46.8823, 47.7327, 53.5059, 44.7574, 53.7893,
50.923, 47.9733, 50.9802, 51.735, 57.4111, 50.2515, 48.7359, 50.6785,
54.7976, 52.3944, 49.6013, 48.1748, 51.2003, 43.9086, 49.5595, 48.0124,
```

The first result is the walltime without AVX optimizations, while the second result has AVX optimizations enabled.

h) *(2 points)* How do you interpret these benchmark results?

i) *(2 points)* A strong scaling analysis is performed up to 8 nodes with the non-AVX implementation, then with the AVX implementation. Which one should exhibit the best speed-up? Support your argument.

Architecture	4-issue pipelined superscalar processor Out-of-Order execution, Speculative execution
L1 Data Cache	32 KB per core, 64 B/line, 8-WAY, Write-back policy latency 4-5 cycles
L1 Instruction Cache	32 KB per core, 64 B/line, 8-WAY
L2 Cache	256 KB per core, 64 B/line, 8-WAY, Write-back policy latency 12 cycles
L3 Cache	25 MB, 64 B/line, 16-WAY, Write-back policy latency 59 cycles
RAM	latency 59 cycles + 46 ns
SIMD Extensions	SSE SSE2 SSE3 SSSE3 SSE4 SSE4.1 SSE4.2 AVX AVX2

Table 1: Specifications for the Broadwell processor architecture.

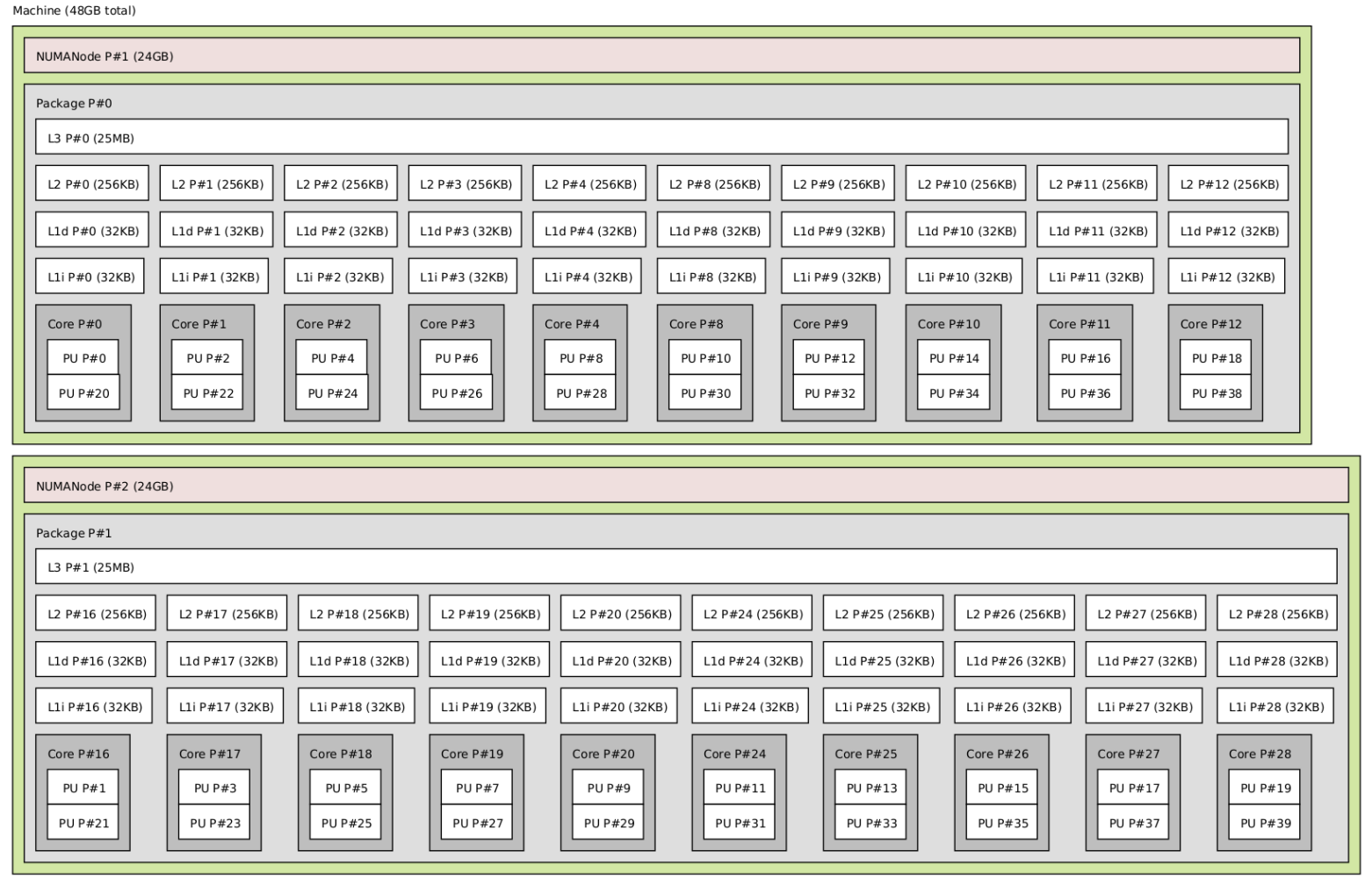


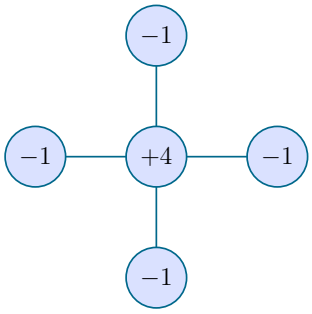
Figure 1: Topology of a dual-socket E5-2630v4 (Broadwell) compute node.

Problem 2 **Solving the Poisson equation** *(24 points).* Consider the solution u to the two-dimensional Poisson problem with homogeneous Dirichlet boundary conditions,

$$-\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in \Omega = (0,1) \times (0,1), \tag{1}$$

$$u(\boldsymbol{x}) = 0 \qquad \boldsymbol{x} \in \partial\Omega. \tag{2}$$

with f a given right-hand side. Problem (1)–(2) is approximated by a finite difference method on a regular grid with $(n+1)$ points in each spatial direction, with the grid size $h = 1/n$. The Laplace operator is discretized by the standard 5-point stencil,



a) *(2 points)* Discuss the nature of the linear system and how the implementation should be adapted in consequence to reach the best performance.

The linear system is solved using a Conjugate Gradient (CG) method summarized in Table 2.

b) *(6 points)* Describe the different operations involved in the algorithm, then give an estimate of the computational complexity and the memory requirements for one iteration of CG. Discuss the scalability in terms of operations and memory.

An implementation is now considered for distributed memory architectures and with a double-precision storage. Let us assume first that the computational domain is partitioned into p slices of size $n \times n/p$ and for a number p of processors which is assumed to be a power of two.

The operation rate of the processor is η Flops, and the communication cost through the network interconnect is modelled with a linear relation where the time to send a message of k bytes can be expressed as

$$\tau_c(k) = \tau_s + \gamma k$$

where τ_s is the startup time (latency) and γ is the inverse bandwidth.

c) *(4 points)* For each operation involved, describe how they parallelize in the context of a distributed memory programming model: express the computational cost for a serial run, then the communication cost T_c.

Let us consider that the time T_p for a parallel run on $p > 1$ processes can be divided into two parts:

1. a purely parallel runtime,
2. a given communication cost.

d) *(4 points)* Express the speed-up S_p for the inner-product operation and matrix-vector product (with a one-dimensional (slice) distribution) in terms of p, η, n and parameters introduced by the communication model. Use the previous results to express the speed-up S_p for one Conjugate Gradient iteration.

e) *(2 points)* The algorithm is now extended to the three-dimensional case with a seven-point stencil, and the domain is still sliced in one-dimension. Express the speed-up S_p for one Conjugate Gradient iteration. Compare the result with the two-dimensional case.

f) *(2 points)* How is the speed-up S_p influenced by the way the domain is partitioned?

g) *(2 points)* Looking at the Top500 List of November, the first ranked computer is Sunway TaihuLight which has a measured LINPACK performance of $R_{max} = 93,014.6$ TFLOPS for a theoretical $R_{peak} = 125,435.9$ TFLOPS. What do these numbers mean? Calculate the computational efficiency and interpret the result.

h) *(2 points)* A recent benchmark shows that Sunway TaihuLight delivered 480.84 TFLOPS for the High-Performance Conjugate Gradient benchmark. Calculate the computational efficiency and interpret the result.

Given $\boldsymbol{b} \in \mathbb{R}^N$, $\mathrm{A} \in \mathbb{R}^{N\times N}$ symmetric positive definite and $\epsilon \in \mathbb{R}$ a tolerance, compute an approximate to $\boldsymbol{x} \in \mathbb{R}^N$ satistfying $\mathrm{A}\boldsymbol{x} = \boldsymbol{b}$.

Let $\boldsymbol{r}_0 = \boldsymbol{b} - \mathrm{A}\hat{\boldsymbol{x}}^0$ be the initial residual and let us assume that $\hat{\boldsymbol{x}}^0 = 0$.

While $\sqrt{\varrho_k} = \|\boldsymbol{r}_k\|_2 > \epsilon\|\boldsymbol{b}\|_2$:

1. Compute the projection factor:

$$\beta_0 = 0; \quad \beta_k = \frac{\varrho_k}{\varrho_{k-1}}, \quad \forall k > 0$$

2. Compute the direction:

$$\boldsymbol{e}_1 = \boldsymbol{r}_0; \quad \boldsymbol{e}_{k+1} = \boldsymbol{r}_k + \beta_k \boldsymbol{e}_k, \quad \forall k > 0$$

3. Compute the direction factor:

$$\boldsymbol{w} = \mathrm{A}\boldsymbol{e}_{k+1}; \quad \alpha_{k+1} = \frac{\varrho_k}{\boldsymbol{e}_{k+1}^T \boldsymbol{w}}$$

4. Update the solution:

$$\hat{\boldsymbol{x}}^{k+1} = \hat{\boldsymbol{x}}^k + \alpha_{k+1}\boldsymbol{e}_{k+1}$$

5. Update the residual:

$$\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \alpha_{k+1}\boldsymbol{w}$$

Table 2: Conjugate Gradient algorithm

Problem 3 General knowledge *(12 points)*

Answer by true or false, and provide a justification:

a) *(2 points)* An optimized implementation shows a better strong scaling than a non-optimized implementation.

b) *(2 points)* When using non-blocking communication in MPI, the buffer can be re-used as soon as the function returns.

c) *(2 points)* A global reduction has a binary logarithm scaling with respect to the number of processors.

d) *(2 points)* The computational efficiency is higher for BLAS Level 1 operations than for Level 3 operations.

e) *(2 points)* The number of OpenMP threads per compute node should always be chosen to the maximum of threads possible on the hardware to maximize performance.

f) *(2 points)* In OpenMP the dynamic scheduling policy provides always the best performance for linear algebra operations.

Open question:

g) *(Bonus: 4 points)* Describe in four points each the main characteristics of the two programming models introduced during the course, and their relation to the notion of fine- and coarse-grained parallelism.