Heidelberg University
Institute of Computer Engineering
Computer Engineering Group

Holger Fröning
Bálint Soproni

Introduction to High-Performance and Distributed Computing
Winter Term 2023/2024

Exercise 7

- **Hand in via Moodle until 09:00 on Monday 8 January, 2024**
- **Include all names on the top sheet. Hand in a single PDF.**
- **Please compress your results into a single archive (`.zip` or `.tar.gz`).**
- **Please employ the following naming convention `hpc<XX>_ex<N>.{zip,tar.gz}`, where `XX` is your group number, and `N` is the number of the current exercise.**
- **A maximum of three students is allowed to collaborate on the exercises.**
- **In case an exercise requires programming:**
  - **include clean and documented code**
  - **include a Makefile for compiling**

7.1 Multi-stage interconnection networks

You are the well-paid engineer responsible for interconnection networks (IN) at a hypothetical company. Assume that due to technology and manufacturing constraints, there is only one networks element of a certain configuration available: a router consisting of a crossbar with 16 inputs and 16 outputs (ports). Consider a bidirectional multi-stage interconnection network (BMIN) following the CLOS architecture with 3 stages based on this router configuration.

- How many ports does the resulting BMIN have?
- Now many network elements (routers) are required in total?
- Can you save network elements? In this case, which key property of the IN changes? Can you quantize this change using a suitable metric? (Do not calculate, just explain)

(16 points)

7.2 n-Body Problem – Implementation

Implement an MPI version of the n-Body problem using the partitioning and communication model developed in the last exercise. All constraints from the previous exercise are still valid. Your program should work with any process count from 1 to 24.

(20 points)

7.3 n-Body Problem – Experiments

- Measure the average time for object counts of 256, 512, 1024, $2k$, $4k$, $8k$, $16k$ $32k$. Report the average time of one iteration by performing for instance 100 iterations, measuring the time with a suitable function (e.g., *gettimeofday()* in Linux) and dividing by the number of iterations. Do not include time for initialization or output. Use compiler-specific optimizations to minimize the runtime.
- Increase the number of processes from 2 to at least 64 for these object counts.
- Fill out the following table (or use plots) for time/iteration, speedup, and efficiency and interpret the results!
- **(Bonus)** Use score−p (see Exercise 6.3) to further analyze your program's runtime behavior.

(20 points + 10 bonus points)

Object count	NP = 2	NP = 4	NP = 8	NP = 12	...	NP = 64
128						
512						
1024						
2048						
4096						
8192						
16384						
32768						

7.4 willingness to present

Please declare willingness to present.

- Multi-stage interconnection networks
- n-Body Problem – Implementation
- n-Body Problem – Experiments

(27 points + 5 bonus points)

81 points (+ 15 bonus points)