Heidelberg University
Institute of Computer Engineering
Computer Engineering Group

Holger Fröning
Bálint Soprani

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

Exercise 2

- **Hand in via Moodle until 09:00 on Monday 6 November, 2023**
- **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**

Preparation

Refer to the attached document on how to log in and compile on our brook cluster. Use the account you've been assigned in the exercise. If you have no account, contact the exercise lead.

2.1 MPI Ring Communication

Write an MPI program for n processes, that exchanges m messages per process in the following way: process i sends m messages to process ((i+1) % n). Include time measurement functions to report the total execution time and the average time per message. Ensure that your program cannot run into a deadlock.
Execute the program on the available creek nodes with up to 24 processes total. Choose m in a way that the average time per message is stable (several consecutive runs should report similar results). Report latency for 2, 4, 6,. . . , 24 processes. Create a graph with your results.
Minimize the average time per message for 12 processes by optimizing the mapping. Explain why this mapping is optimal for this program.

(25 points)

2.2 Measure Latency

Latency is a very important metric for an interconnection network. There are two kinds of latency:

1. Full round-trip: Time between sending (source) and receiving an appropriate response (source). Because different nodes have no shared clock this is the only possibility to determine the time required for a message exchange.
2. Half round-trip: The full round-trip latency divided by two. This is usually the amount of time referred to as latency.

Write a MPI ping-pong test program to measure the round-trip latency. Execute your program both on one node (with two processes) and on two nodes of the creek cluster. Choose two idle nodes for measurement. Do your experiments with several message sizes e.g. 1KB, 2KB, 4KB, 8KB, . . . , 1MB. Vary the message size within your program. Choose an appropriate number of iterations to yield stable results.
Plot your results for several message sizes in a graph and interpret them. Do the same for execution on one and two nodes.

(25 points)

Holger Fröning
Bálint Soprani

2.3 Measure Bandwidth

Another metric for interconnection networks is bandwidth. Bandwidth describes how much data can be transferred within a given time.
To measure bandwidth you have to write another MPI test program called flood test. Send as much data to another process as possible. You should vary the message size in the same way as described in the previous exercise.
Write two flood test programs by using different MPI send operations:

1. non-blocking send
2. blocking send

Execute your program both on one and on two nodes (choose two idle nodes of the creek cluster). Plot your results for several message sizes in a graph and interpret them. Choose an appropriate number of iterations to yield stable results. Compare the bandwidth of blocking MPI send and non-blocking MPI send. Do the same for execution on one and two nodes. Do you observe a difference between message sizes and if so why?

(25 points)

2.4 Barrier Synchronization

Now we want to measure the performance of an MPI barrier. First implement your own barrier, either a centralized barrier or a tree-based barrier. It's your choice, however a centralized barrier might be easier to implement. Your choice won't have an impact on achievable points.
Write a suitable test program to measure the performance of your own barrier implementation. For this purpose call the barrier within a for-loop and measure the overall execution time of the loop. The number of iterations should be passed to the program as a command line parameter. Use a sufficient number of iterations to get stable results. Report average barrier latency (loop execution time divided by the number of iterations) for 2,4,6,...,24 processes.
Do the same measurement with the MPI built-in barrier. Compare the performance and plot both results in one graph.

(25 points)

Total: 100 points