High Performance Computing Exercise 1

Luis Fernando Palacios Flores

Introduction

- Collective operations are very important in parallel applications and require efficient and effective algorithms to ensure optimal performance.
- No single algorithm has proven superior in all scenarios.
- Hockney models can take the form:

$$T(P, m, \alpha, \beta) = f(P, m) \times g(m, \alpha, \beta)$$

- Goal: to compare different openMPI algorithms for broadcast and barrier.

Methodology

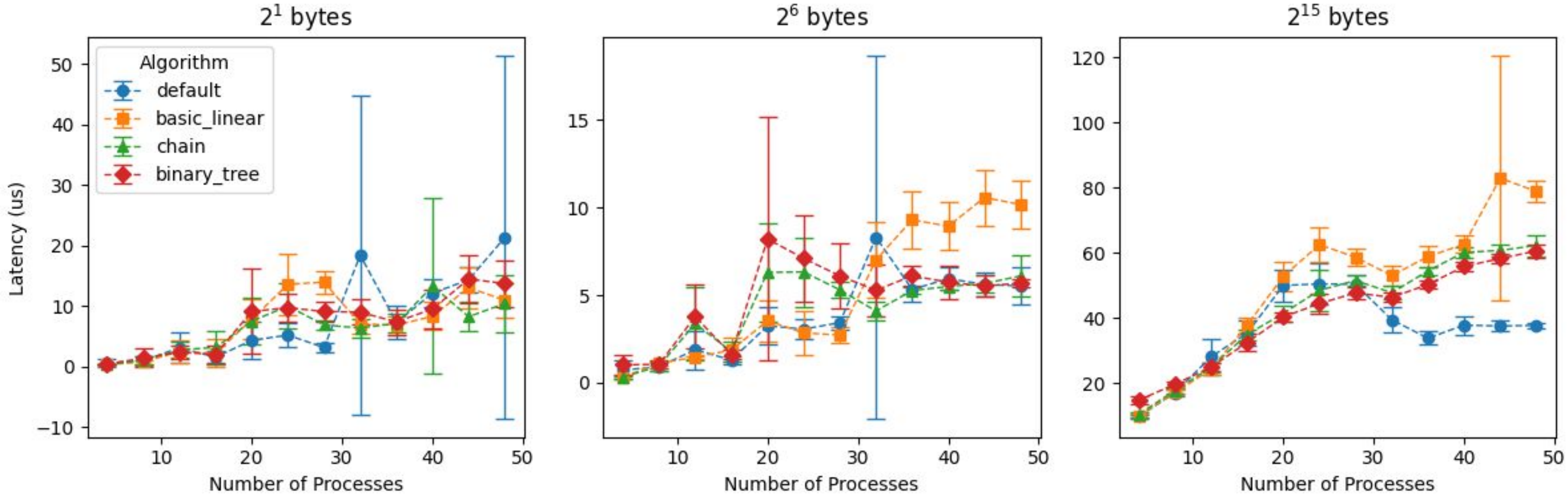
- OSU library to collect measurements of the latency tests.
- ORFEO nodes: two EPYC nodes (epyc002 and epyc004) and two THIN nodes (thin001 and thin002).
- Different numbers of processes and message sizes.
- The procedure was repeated 10 times for broadcast and 20 times for barrier.
- The processes were mapped by core

Selected algorithms:

  - broadcast: basic linear (1), chain (2), binary tree (2)
  - barrier: linear (1), double ring (2), Bruck (4)

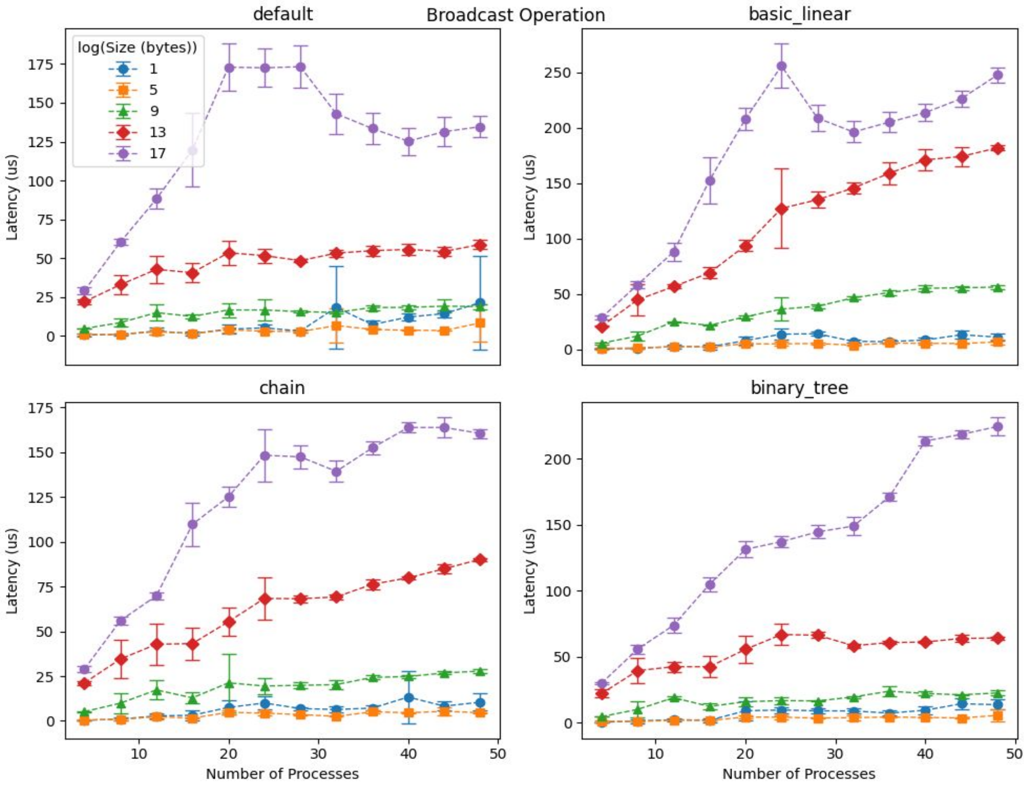
Default algorithms (0) were selected as a baseline for benchmarking.

Results and Discussion



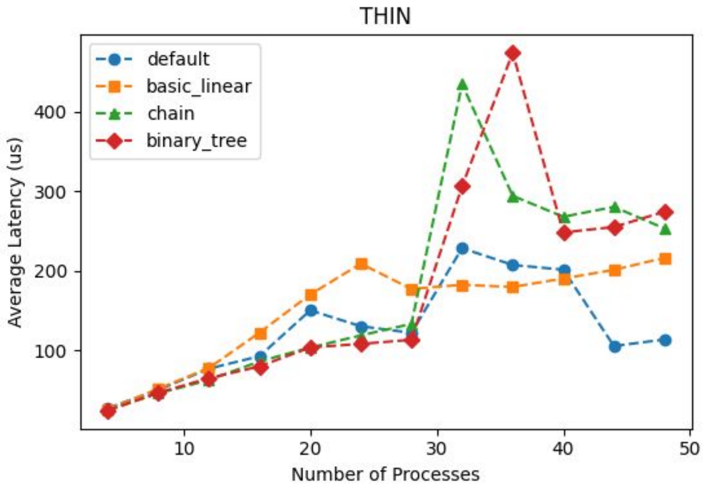
- The latency **tends** to increase with the number of processors though **not necessarily in a linear fashion.**

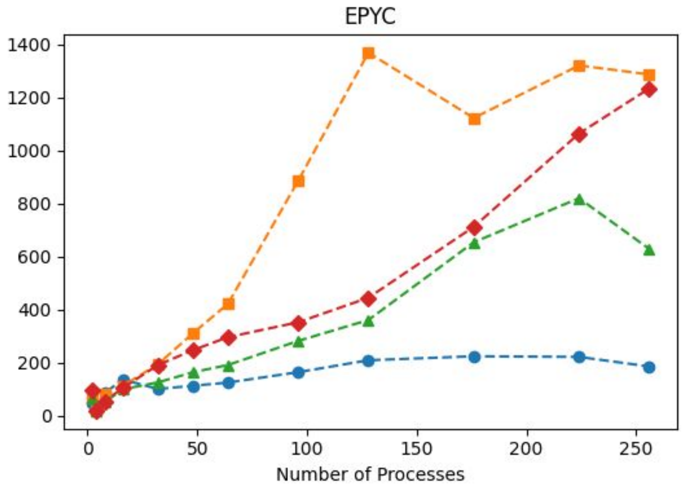
Results and Discussion



- The latency had large variability for processes that exceeded the number of cores in one node.
- It suggests that the performance can be influenced by factors related to the computer architecture.

Results and Discussion





The algorithms have to be carefully selected depending on the application, cluster partitions and more.

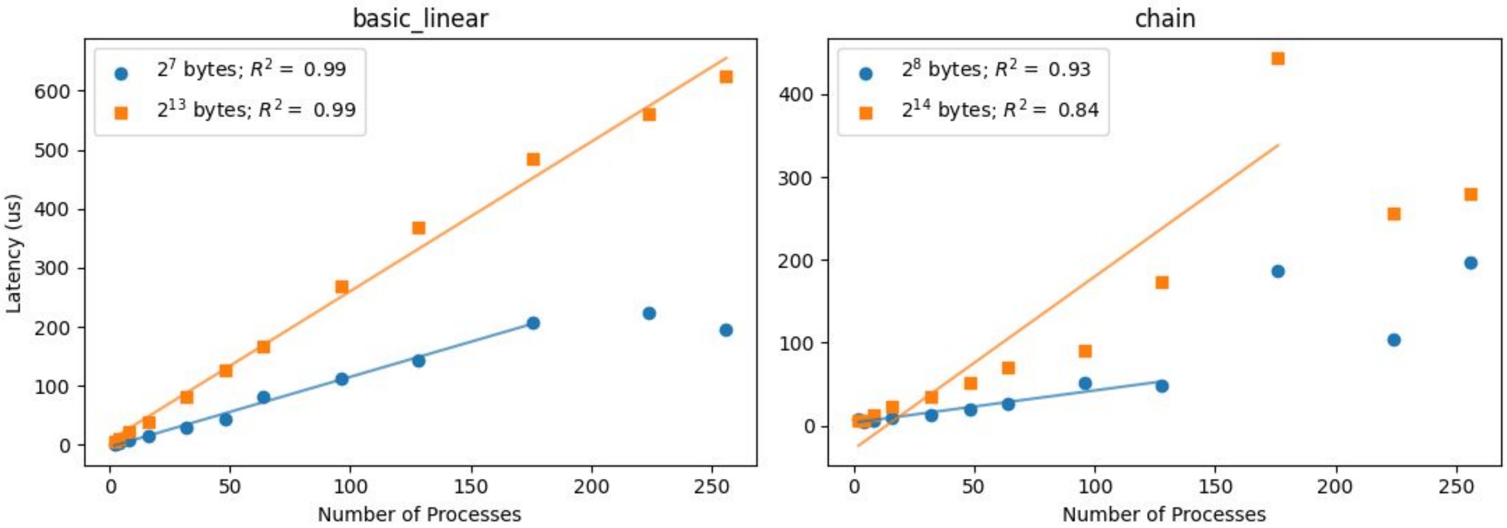
THIN nodes

- The default algorithm outperformed the basic linear, binary tree, and chain algorithms.
- For processes in one node the binary tree algorithm proved to be better.

EPYC nodes

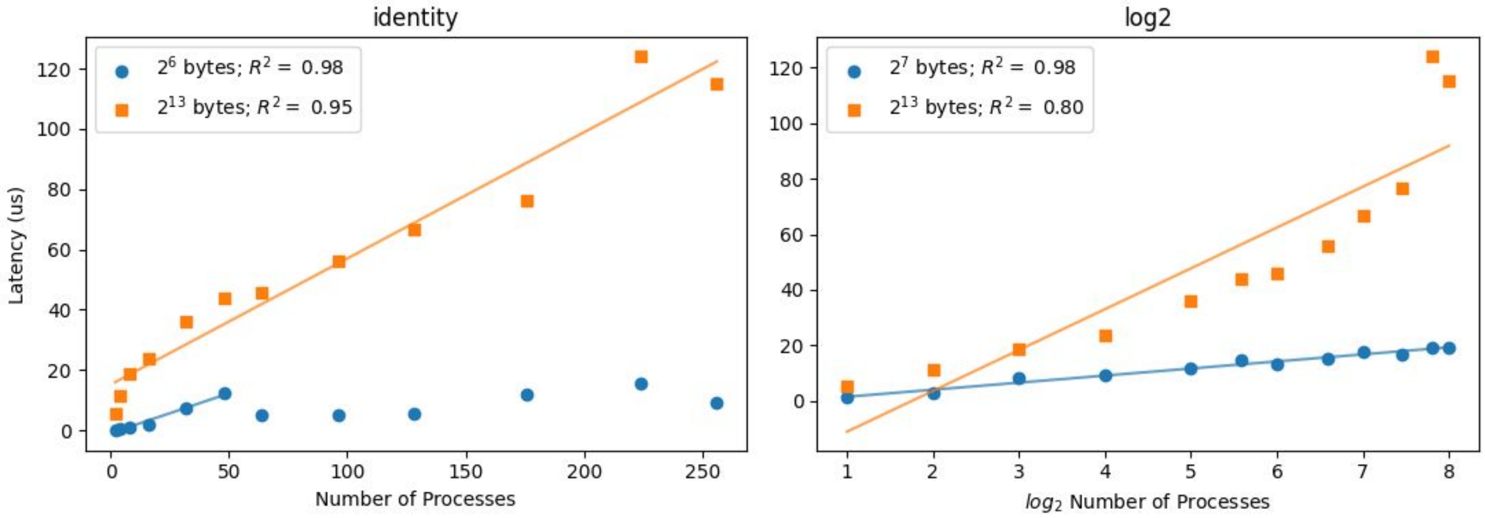
- The ranking of the algorithms was: default, chain, binary tree, and basic linear.
- The binary tree performed well for small and middle-size messages.

Results and Discussion



- In EPYC nodes, the latency is well described by a linear model using almost all processes for small messages for the basic linear algorithm and large messages for the chain algorithm. For other message sizes, a satisfactory description is achievable only up to a certain number of processes.

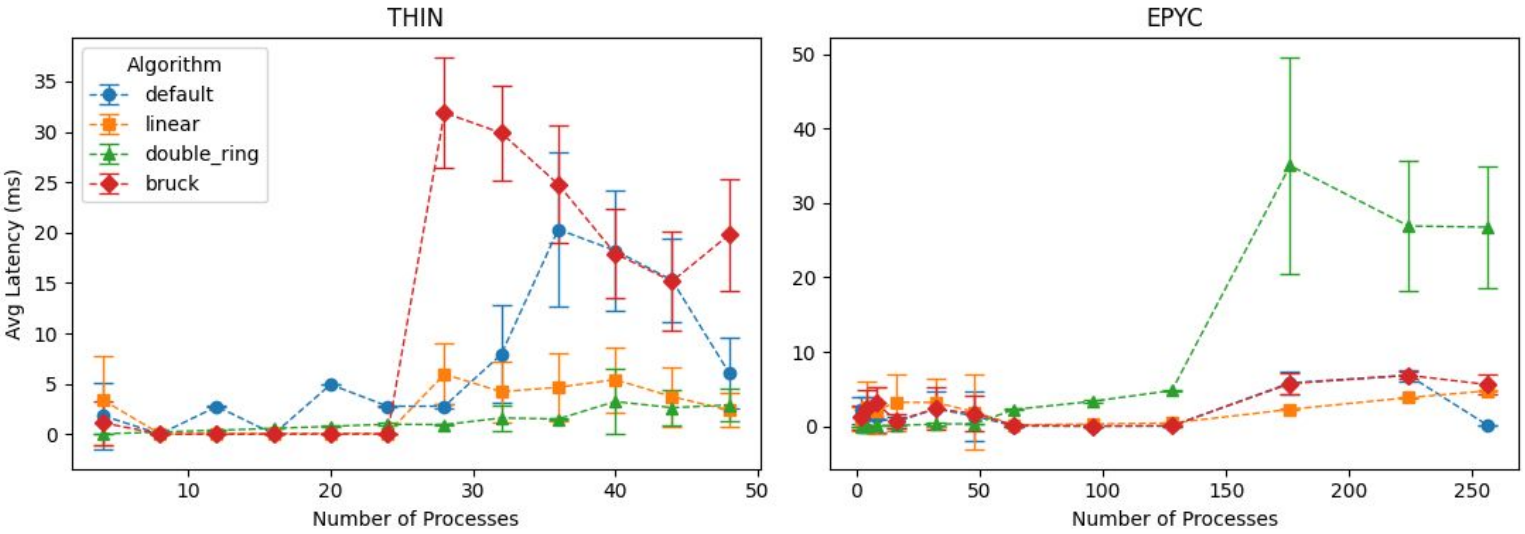
Results and Discussion



- In most cases, a logarithmic relationship best fits the data.
- In contrast to the description provided in the literature, for large message sizes the linear relationship outperforms the logarithmic one.

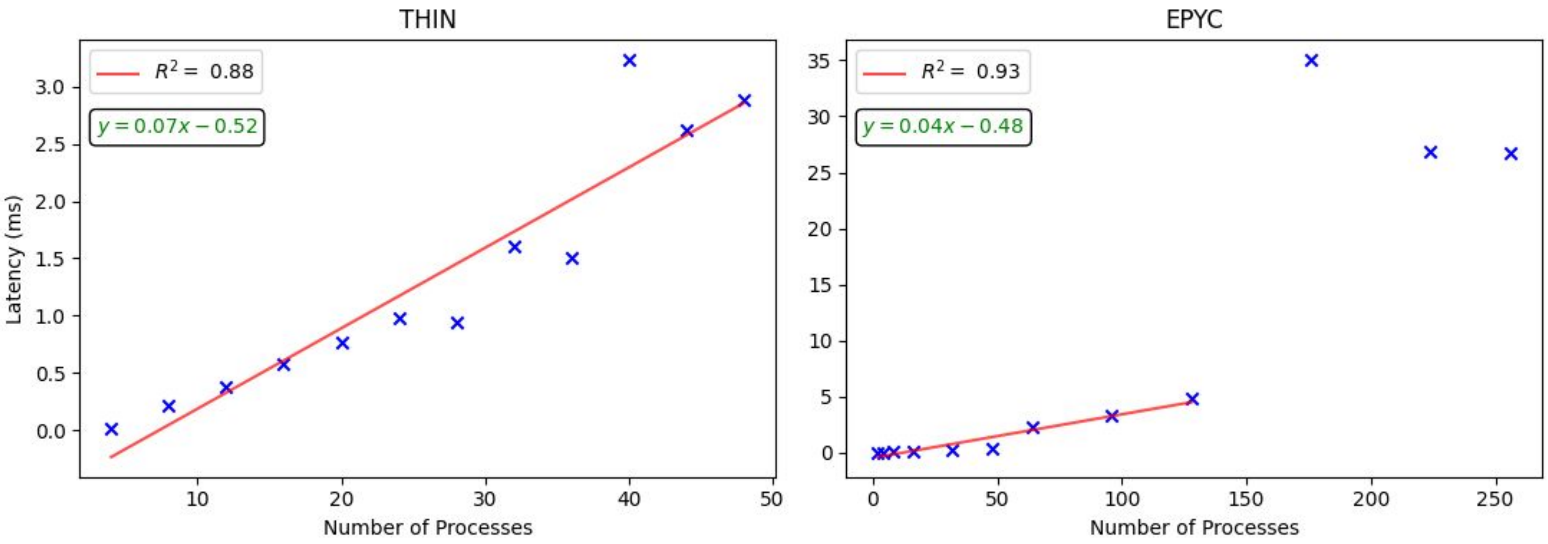
- For the THIN nodes, similar observations were made for the binary tree algorithm.
- Surprisingly, for the other algorithms a logarithmic relationship also provides a good description of the data **or even better than the linear one**.

Results and Discussion



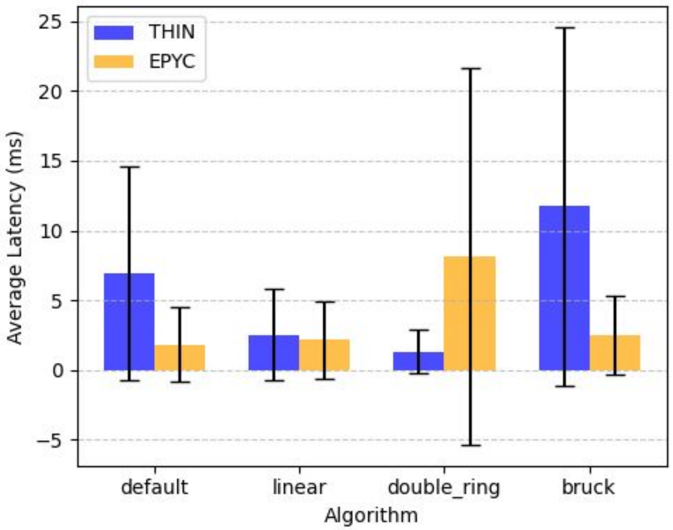
- The relationship between latency and the number of processes for the barrier operation in both THIN and EPYC nodes is not as clear as for the broadcast operation.

Results and Discussion



- Only the double-ring algorithm is accurately described by the model found in the literature.
- THIN nodes: total description; EPYC nodes: good model for one (out of two) node.
- Estimated latency of point-to-point communication is 35.12 ± 4.14 μs (THIN nodes) and 19.43 ± 2.03 μs (EPYC node).

Results and Discussion



- THIN nodes: the double-ring algorithm performs best with an average latency of 1.31 ± 1.59 ms.
- Most algorithms perform better in the EPYC nodes than in the THIN nodes.
- EPYC nodes: the default algorithm performs slightly better than the linear one with an average latency of 1.83 ± 2.70 ms.

Conclusions

- The comparison of the algorithms revealed that there is no single algorithm that consistently performs best in all scenarios.
- The default algorithm performed best on both THIN and EPYC nodes for the broadcast operation and only for the EPYC partition in the barrier operation.
- The models found in the literature not always allowed to have a good description of the relationship between latency and the number of processes.
- Only the broadcast operation was accurately modeled for all models.
- The barrier operation was modeled according to the literature only for the double-ring algorithm.
- The models had limitations regarding the number of processes used to represent the latency.
- Algorithms for parallel applications have to be carefully selected based on the application, cluster partitions, and other relevant parameters.