Collective operation in *Message Passing Interface (MPI)* [**?**] are foundamental building blocks for parallel applications. They are used to exchange data among processes and are often used to implement higher level parallel algorithms. Efficient implementations of these operations that aim to minimize latency and maximize throughput are therefore a must for high performance computing.

This report studies the case of the broadcast and reduce collective implementations of the MPI standard by presenting two models for latency prediction. In particular, since the *OpenMPI* library offers the possibility to choose among different algorithms for these operations, this study, together with the default MPI implementation, also considers the *linear*, *chain* and *binary tree* algorithms for the selected operations.

The main problem covered in this report is the task of predicting the overall latency of these operations for various message sizes and number of processes in multi-core systems. The measurement of latency times in different conditions has therefore been carried out by running the well-known *OSU Micro-Benchmarks* suite [**?**] on the *ORFEO* cluster of the AREA Science Park in Trieste [**?**] by testing different processes allocations and message sizes.

With the collected data, two different models for latency estimation have been built. The first one is based on point-to-point communications latencies and builds on top of the model presented by *Nuriyev et al.* [**?**]. The second model is based on a linear regression approach and uses the collected data to train a model that can predict the latency of the collective operations.

The underlying goal of this project has been to compare the two approaches and attempt to find the best compromise between accuracy and affidability in predictions and the overall explainability of the model, taking into account the architecture on which the benchmark has been conducted.

The following sections are organized as follows. In Section **??**, the studied algorithms for the collective operations will be presented and analyzed from a theoretical point of view. Section **??** describes the architecture and the precise methodology used for the measurements. Sections **??** and **??** will then present the implemented models for the latency predictions, followed by the obtained results and some final considerations in the last two sections of this report.

With the aim of maintaining a clear and concise report of the work, the following sections will only present the most noticeable results of the study, while further analysis and the detailed model implementations will be available on the author's GitHub repository [**?**].