

# Message Passing Programming coursework: optimization of `percolate` v0.1.0 using a 2d domain decomposition with MPI

## Abstract

**Keywords:** Scientific programming, benchmark, parallelization, performance optimization, MPI

## 1. Introduction

`percolate` v0.1.0 is a scientific program written in the Fortran programming language. It generates a random matrix with two kinds of cells: empty and filled. Empty cells build clusters with their neighboring empty cells. `percolate` computes all clusters in the matrix and searches for a cluster that percolates. A cluster percolates, if it begins in the left most column of the matrix and ends in the right most column.

The computing of the clusters—the clustering—is done iteratively and is the main cause of computation in `percolate`. If the clustering is done in a serial way, it does not scale well and clustering bigger matrices can consume a lot of time and power.

This paper presents a parallelized version of `percolate`. The parallel version decomposes the matrix into smaller chunks and distributes them among worker instances. Every worker performs the clustering of its chunk and communicates with the neighboring worker instances through halo swapping, which ensures that the whole matrix is clustered. Chunks are generated by splitting the matrix on both axes. This makes it a 2d domain decomposition of the matrix. The parallel version is based on MPI and the worker instances are MPI processes (see Message Passing Interface Forum, 2015).

First, this paper describes the parallel version of `percolate`. The parallel version's correctness is tested by a regression test suite, which is briefly outlined. Afterwards a benchmark is presented, which analyzes the scaling behavior of the parallel version over multiple amounts of MPI processes and with different sized matrices. The results of the benchmark are discussed and a conclusion is drawn.

**2. Method**

**3. Results**

**4. Discussion**

**5. Conclusion**

**References**

Message Passing Interface Forum. MPI: A Message-Passing Interface Standard. Technical report, 2015. URL <https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf>.