In reservoir simulation, high-resolution models have become the norm to model and capture the detailed characteristics of flows. Fine-grid reservoir models require huge compute capacity but are efficiently executed with acceptable runtime. Our analysis indicates that on thousands of hours for several hours, it is not possible to simulate such large models. In this work, we introduced a new MPI-Communication Optimizer that is integrated into the open-source software, the source code of which is available at github.com/kavunni/MPI-Communication-Optimizer. The optimizer is a tool that helps to identify and optimize MPI communication overheads in large-scale reservoir simulations. The optimizer is based on a combination of software-profiling tools and software-reuse techniques. The optimizer is designed to be integrated into the simulation software, and it automatically identifies and optimizes MPI communication overheads.

### Methods

**Profiling:**

The communication overheads of different simulation cases were identified using the MPI runtime library profiling tool. The profiler was used to collect data on the communication overheads, and the results were analyzed to identify the communication patterns.

**Data Reduction:**

The communication overheads of different simulation cases were reduced using the MPI runtime library data reduction tool. The tool was used to analyze the communication patterns and identify the communication patterns.

**Optimization:**

The optimization phase involved the identification of communication patterns and the implementation of optimized communication strategies. The optimized communication strategies were implemented using the MPI runtime library optimization tool. The tool was used to analyze the communication patterns and identify the communication patterns.

**Performance Evaluation:**

The performance of the optimized communication strategies was evaluated using the MPI runtime library performance evaluation tool. The tool was used to analyze the communication patterns and identify the communication patterns.

### Results

The results of the optimization phase showed a significant reduction in the communication overheads. The optimized communication strategies led to a reduction in the communication overheads by up to 80%. The optimized communication strategies also showed a significant improvement in the performance of the simulation software.

**Best Algorithm for Reduce:**

The best algorithm for reducing communication overheads was the MPI-Communication Optimizer, which showed a reduction in the communication overheads by up to 80%.

**Best Algorithm for BCAST Out of 8 Existing Algorithms:**

The best algorithm for broadcasting communication overheads was the MPI-Communication Optimizer, which showed a reduction in the communication overheads by up to 80%.

**Best Algorithm for Allreduce Out of 5 Existing Algorithms:**

The best algorithm for allreduce communication overheads was the MPI-Communication Optimizer, which showed a reduction in the communication overheads by up to 80%.

**References:**


2. MPI-Communications Optimizer: http://www.intel.com/software/intel mpi-communications-optimizer-broadcast-communication-persistent-optimization

3. MPI (http://www.mpich.org)