MPI Communication Optimization of Massively Parallel Applications

1 INTRODUCTION

In reservoir simulation, high resolution models have become the norm to model detailed characteristics of fluids flow in hydrocarbon bearing reservoir. This leads to enhanced understanding of the physics and accuracy of the simulation results. To run these models efficiently, they require huge compute capacity. Saudi Aramco has a parallel reservoir simulator which has the capability of simulating huge reservoir simulation models. Massively parallel scientific and engineering applications exhibit MPI communication overhead that can be reduced for the applications runtime optimization and scalability. The work presented will provide the process used in identifying communication hotspots by profiling and analyzing various simulation models, and how the optimization was undertaken to overcome communication bottlenecks.

2 METHODS

Software hotspot profiling and analysis was performed over selected cases using Intel® MPS profiler which was used to conduct in depth profiling of the MPI communication. It was observed that "MPI Barrier" calls is one of the major bottlenecks. Table 1 depicts the impact of the MPI Barrier calls on the communication; the overhead of the calls varies from 14% to 44% of the communication time.

Table 1. Illustration of MPI Barrier Calls for Four (4) Simulation Models

Case Name	No. of Cores	Time (sec)	Time (%)	Calls
CASE 1	500	303303.55	14.80	421,693,313
CASE 2	500	260468.63	29.19	761,836,748
CASE 3	490	1125990.74	44.60	286,070,005
CASE 4	500	1315744.11	35.18	1,514,059,234

2.1 First Optimization Technique:

The main objective is to reduce the MPI Barrier calls as much as possible without changing the simulation results. The methodology of removing the MPI synchronization calls was by identifying unnecessary calls within the code. Unnecessary calls are usually those calls that were added for debugging purposes where developers tend to add to debug their work.

2.2 Second Optimization Technique:

Another optimization technique was addressed to see if MPI communication can be further improved by using MPI Intel collective algorithms. Intel has different number of built in algorithms for each MPI collective operation. So, four highly used collective operations within the reservoir simulator were benchmarked by using Intel balance benchmark. Below figures show the best algorithm for each MPI collective operation with respect to the number of Bytes.

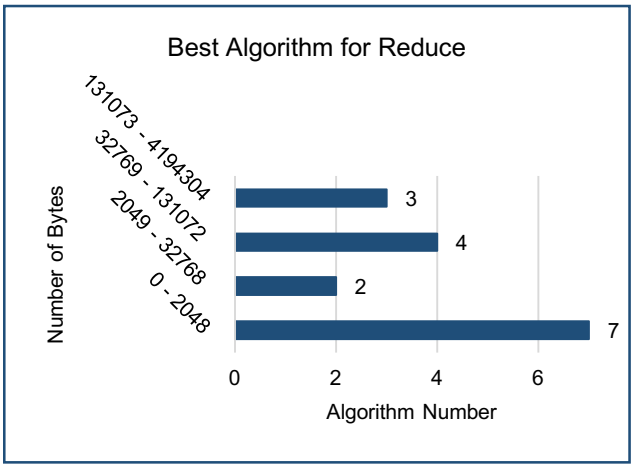

Fig 1. Algorithm Selection for Reduce Operation out of 7 Existing Algorithms.

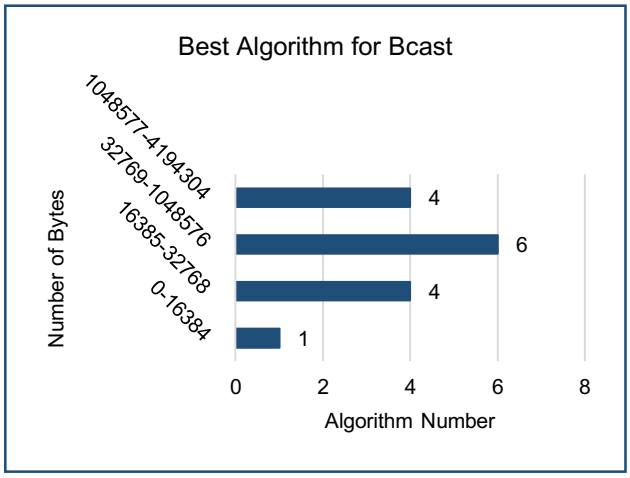

Fig 2. Algorithm Selection for Bcast Operation out of 8 Existing Algorithms.

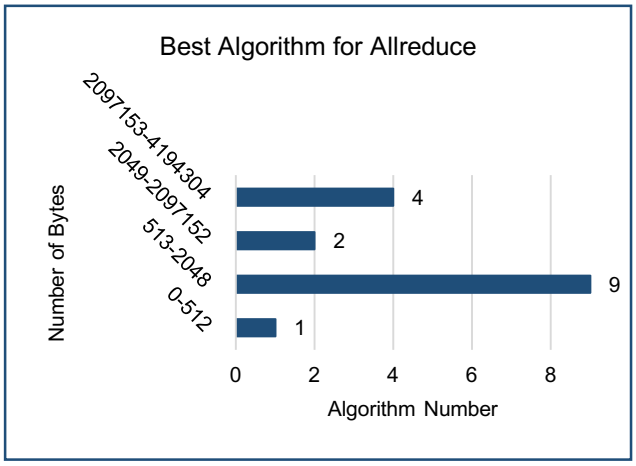

Fig 3. Algorithm Selection for Allreduce Operation out of 9 Existing Algorithms

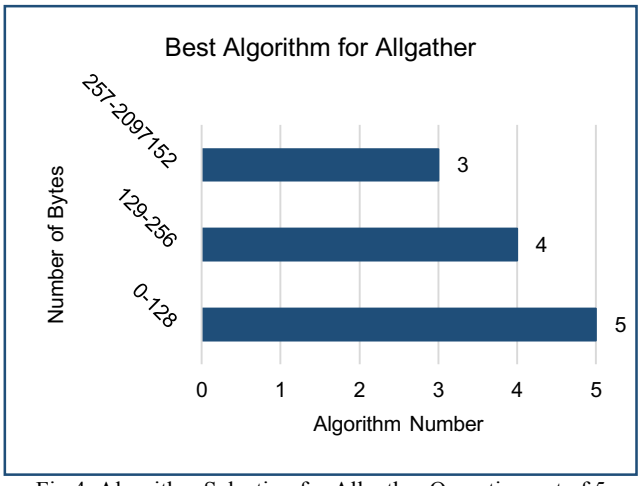

Fig 4. Algorithm Selection for Allgather Operation out of 5 Existing Algorithms

3 RESULTS

3.1 First Optimization Outcomes:

Toward an optimal optimization, unnecessary MPI Barrier calls were targeted for removal. The percentage usage of MPI Barrier calls before optimization varies from 14.80% to 44.60%. After optimization communication overhead coming from MPI Barrier calls was reduce to 0% in all simulation cases. Table 2 illustrates that. This level of optimization has shown up to 4% speedup.

Table 2. Illustration of MPI Barrier Calls in the Original vs. Optimized Cases

Case Name	Executable	Time (sec)	Time (%)	Calls
CASE 1	Original	303303.55	14.80	421693313
	Optimized	0	0	0
CASE 2	Original	260468.63	29.19	761836748
	Optimized	0	0	0
CASE 3	Original	1125990.74	44.60	286070005
	Optimized	0	0	0
CASE 4	Original	1315744.11	35.18	1514059234
	Optimized	0	0	0

3.2 Second Optimization Outcomes:

On top of the first optimization, four highly used collective operations within the simulator application were optimized by selecting the best algorithm for each message size with respect to each MPI collective. The targeted MPI collectives here are MPI Allreduce, Allgather, Reduce, and Bcast. This level of optimization has shown up to 3% speedup.

4 CONCLUSIONS

Simulation models in the order of one billion cells pose a challenge on runtime. MPI communication optimization is one of the main approaches to speed up the simulation runs and prepare the software for Exascale computing. MPI Barrier reduction has shown runtime improvements without any changes in the simulation results. Besides, Intel MPI collectives algorithms has improved the overall runtime. The cumulative speedup out of both optimizations goes up to 6% which saves a lot of reservoir simulation computing resources.

REFRENCES

[1] Dogru, A. H., Fung, L.S., Middya, U., et al. (2011). New Frontiers in Large Scale Reservoir Simulation. SPE Reservoir Simulation Symposium. The Woodlands, Texas: Society of Petroleum Engineers. doi:10.2118/142297-MS.

[2] Dogru, A. H., Fung, L. S., Middya, U., et al. (2009). A Next-Generation Parallel Reservoir Simulator for Giant Reservoirs. SPE Reservoir Simulation Symposium. The Woodlands, Texas: Society of Petroleum Engineers. doi:10.2118/119272-MS.

[3] Dogru, A.H., Fung, L.S., Middya, U., et al. (2008). From Mega Cell to Giga Cell Reservoir Simulation. SPE Annual Technical Conference and Exhibition. Denver, Colorado: Society of Petroleum Engineers. doi:10.2118/116675-MS.

[4] Kini, S.P., Liu, J., Wu, J., Wyckoff, P., and Panda, D.K. (2003). Fast and Scalable Barrier Using RDMA and Multicast Mechanisms for Infiniband0Based Clusters, Proceedings of Euro PVM/MPI Conference, (2003).

2018. In *Proceedings of International Conference on High Performance Computing in Asai-Pacific Region, Tokyo, Japan, January 2018 (HPCAsia), 4 pages.*
DOI: XXXX