Multi-GPU Optimization of a Non-hydrostatic Numerical Ocean Model with Multigrid Preconditioned Conjugate Gradient Method

Takateru Yamagishi(1) Yoshimasa Matsumura(2) Hiroyasu Hasumi(3)

1: Research Organization for Information Science and Technology, yamagishi@rist.jp 2: Atmosphere and Ocean Research Institute, The University of Tokyo, ymatsu@aori.u-tokyo.ac.jp 3: Atmosphere and Ocean Research Institute, The University of Tokyo, hasumi@aori.u-tokyo.ac.jp

Introduction

- The objective is to accelerate high-spatial-resolution numerical ocean simulations with multiple NVIDIA GPUs.
- We implemented and optimized our “kinaco” numerical ocean model on multiple NVIDIA P100 GPUs.
- We optimized the inter-GPU communications of MGCG (multigrid preconditioned conjugate gradient method) solver by overlapping of communication with computation and modifying aggregation level of information in a multigrid method.
- We achieved 3.9 times speedup compared to CPUs with good weak scaling up to 64 GPUs.
- We learned how communications depended on the number of GPUs and the aggregation level of information in a multigrid method.

“Kinaco” non-hydrostatic ocean model

- Kinaco [1] resolves vertical convection and eddy mixing with non-hydrostatic approximation on ~1 km scale.
- Pressure field is solved using MGCG (multigrid preconditioned conjugate gradient method) solver, which can be scaled up to 8k processors [2].
- We implemented and optimized our “kinaco” numerical ocean model on multiple NVIDIA P100 GPUs.
- We optimized the inter-GPU communications of MGCG (multigrid preconditioned conjugate gradient method) solver by overlapping of communication with computation and modifying aggregation level of information in a multigrid method.
- We achieved 3.9 times speedup compared to CPUs with good weak scaling up to 64 GPUs.
- We learned how communications depended on the number of GPUs and the aggregation level of information in a multigrid method.

Multi-GPU implementation

- We implemented domains on GPUs with CUDA Fortran [3], using GPUDirect RDMA transfer.

Performance Evaluation

- CPU vs GPU: elapsed time [s] in our SC18 study [3]

Communication optimization on a fine grid: Overlapping with computation

- Overlapping with computation

- Timeline of overlapping

- Communication other

- Coarse grid aggregation

- Coarse Grid Aggregation

- Structure of MGCG: parallelized with 4 GPUs

- On original configuration, especially, communication of a few grids can be more inefficient compared to communication on CPUs owing to the overhead of CUDA-Aware MPI resulting from the intrinsic structure of a GPU-CPU system.

Communication optimization on coarse grid: Coarse Grid Aggregation of MGCG solver

- We implemented and optimized our “kinaco” numerical ocean model with a MGCG solver on multiple NVIDIA P100 GPUs.
- We used two optimization techniques, overlapping of communication with computation and CGA: for modification of multigrid aggregation levels.
- The speedup of MGCG solver is 16.5 times, which is equivalent to 28% reduction of communication of the MGCG solver.
- We discussed inter-GPU communications on a coarse grid on which GPUs could be intrinsically problematic.
- We learned how inter-GPU communications depended on the number of GPUs and the aggregation level of information in a multigrid method.
- In future work, we will adopt the following optimization techniques for detailed experiments with a large number of cells with thousands of GPUs:
  - Hierarchical coarse grid aggregation (CGA) proposed for CPUs [6] would be effective. In CGA, the number of MPI processes are replicated at an intermediate level before the final coarse grid solver on a single MPI process.
  - To reduce the MPI communication cost, we will attempt to apply NCGA to hundreds of GPUs and evaluate and analyze in detail.

Summary and future work

- We implemented and optimized our “kinaco” numerical ocean model with a MGCG solver on multiple NVIDIA P100 GPUs.
- We used two optimization techniques, overlapping of communication with computation and CGA: for modification of multigrid aggregation levels.
- The speedup of MGCG solver is 16.5 times, which is equivalent to 28% reduction of communication of the MGCG solver.
- We achieved speedup of 3.9 times compared to CPUs with good weak scaling up to 64 GPUs.
- We discussed inter-GPU communications on a coarse grid on which GPUs could be intrinsically problematic.
- We learned how inter-GPU communications depended on the number of GPUs and the aggregation level of information in a multigrid method.
- In future work, we will adopt the following optimization techniques for detailed experiments with a large number of cells with thousands of GPUs:
  - Hierarchical coarse grid aggregation (CGA) proposed for CPUs [6] would be effective. In CGA, the number of MPI processes are replicated at an intermediate level before the final coarse grid solver on a single MPI process.
  - To reduce the MPI communication cost, we will attempt to apply NCGA to hundreds of GPUs and evaluate and analyze in detail.

References