Parallel Performance Evaluation of MITgcm

Rin Irie, Helen Stewart, Toru Nakamura, Masaki Hisada
Nippon Telegraph and Telephone Corporation
Musashino-shi, Tokyo, Japan
{rin.irie|helen.stewart|toru.nakamura|masaki.hisada}@ntt.com

1 INTRODUCTION

MITgcm is an ocean-atmosphere general circulation model developed by the Massachusetts Institute of Technology[1]. This model utilizes isomorphism between the atmospheric and ocean fluid equations and uses one hydrodynamical kernel to drive both atmospheric and ocean models. It has a non-hydrostatic capability and can be used to study both small-scale phenomena such as convection and large-scale phenomena such as global general circulation.

MITgcm supports parallel computation using multiple processors by dividing the computation area in the horizontal direction. As a general rule, model computation time does not necessarily increase in proportion to the number of processors, and parallel performance is dependent on the algorithm and implementation. In a previous study, MITgcm computation times were measured for different CPUs and instances on the Google Cloud Platform and optimal cost-performance conditions were determined[5]. However, this study did not cover scaling of the computation time for different numbers of processors. Therefore, in this paper, we evaluate the performance of computation time as the number of processors (i.e., the number of region partitions) is varied.

2 METHODS AND RESULTS

MITgcm supports running multiple processes (distributed memory, MPI[4]) and multiple threads (shared memory, OpenMP[3]) in parallel. The simulation area is horizontally partitioned into grid tiles. For parallel computing, the number of tiles and threads to run in a single process must be set. The number of threads must also be divisible by the number of tiles.

The physical model used for benchmarking is the Barotropic Ocean Gyre, serving as a standard example of a model run. The simulation conditions specified in Table 1, and other parameters such as initial values and boundary conditions follow Section 4.1 of the MITgcm User Manuall[1]. Although the number of the grid points in the z-direction (N_z) is one, partitioning of the simulation area is solely in the horizontal direction, so this does not affect benchmarking of parallel performance. The respective computation times are taken as the average of three measurements. We use two computing nodes for benchmarking interconnected using Infiniband[2], as shown in Table 2. To stabilize CPU performance, Hyper-Threading is disabled, the C-state is fixed at C0, and CPU frequency scaling is fixed at the maximum frequency.

Figure 1 shows the computation time as the number of processes is increased for grid partitions of (1008×1008) and (1344×1344). The lines labeled 'Slope-2' represent optimal strong scaling behavior. Points falling above these lines indicate sub-optimal slowdown of acceleration in the computation time. The benchmark results are almost in line with the 'Slope-2' line, indicating that MITgcm performance scales very well for up to 28 processors. When using 112 processors, the acceleration rate (the ratio of the computation time

Table 1: Simulation Conditions

Simulating Model	Barotropic Ocean Gyre (Manual Sec. 4.1[1])
Grid Spacing	$\Delta x = \Delta y = 2.0$ km, $\Delta z = 5.0$ km
Grid Points	$N_x = N_y = \{1008, 1344\}$, $N_z = 1$
Time step	$\Delta t = 5.0$ minutes
Integration Time	$N_t = 104832$ steps (=1.0 years)

Table 2: Hardware/Software Specifications

CPU	Intel Xeon Gold 6258R (2.7GHz, 28c) x2 sockets
Memory	768GB (DDR4-2933 ECC 64GB x12)
Interconnect	Mellanox ConnectX-6 (200Gb-HCA)
OS	Ubuntu 22.04 LTS
Compiler	gfortran (gcc v11.3.0), mpiexec (OpenRTE v4.1.2)
MITgcm	checkpoint68i (Mid 2022 version)

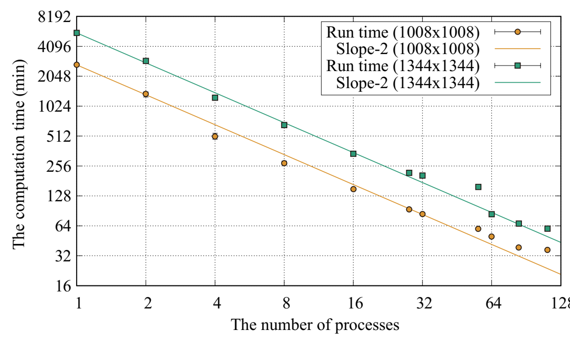

Figure 1: Benchmark of Strong Scaling.

to the computation time when using a single processor) is 72.9 for (1008×1008) grid points but increases to 93.1 for (1344×1344) grid points, close to the theoretical strong scaling value.

3 DISCUSSION

A potential cause for the reduction in acceleration rate for the (1008×1008) grid is an increase in processing time needed to parallelize the computation (i.e., too few grid points computed on a single processor). When using 112 processors, the number of grid points per processor is (192×84) for the (1344×1344) grid, approximately 2.3 times the number of grid points per processor in the (1008×1008) grid. In this poster, results for alternative grid partitions and node allocation will be discussed in more detail.

REFERENCES

[1] A. Adcroft et al. [n.d.]. *MITgcm's User Manual.* Retrieved 2022/12/21 from https://mitgcm.readthedocs.io/en/latest/
[2] Infiniband Trade Association. 2010. *Supplement to InfiniBand Architecture Specification.* Volume 1.2.1.
[3] OpenMP Architecture Review Board. 2018. *OpenMP Application Programming Interface.* Version 5.0.
[4] Message Passing Interface Forum. 1993. MPI: A Message Passing Interface. In *Proceedings of the 1993 ACM/IEEE Conference on Supercomputing*. 878–883.
[5] Fluid Numerics Journal. [n.d.]. *MITgcm Benchmarks on Google Cloud Platform.* Retrieved December 21, 2022 from https://journal.fluidnumerics.com/mitgcm-benchmarks-on-google-cloud-platform