Section 3

Computational studies including new techniques, parallel processing, GPUs. Effects of model resolution.

Optimization of Meteorological Research Institute Community Ocean Model for Japan Meteorological Agency/Meteorological Research Institute Coupled Prediction System version 3 and beyond

Takafumi Kanehama[1,*], Toshinari Takakura[1], Hiroyuki Sugimoto[2], Shogo Urakawa[3], Kei Sakamoto[3], Hideyuki Nakano[3]
1: Numerical Prediction Division, Japan Meteorological Agency, 2: Climate Prediction Division, Japan Meteorological Agency, 3: Meteorological Research Institute, Japan Meteorological Agency
*: tkanehama@met.kishou.go.jp

1 Introduction

JMA's new seasonal prediction system (Japan Meteorological Agency/Meteorological Research Institute Coupled Prediction Model version 3 (JMA/MRI-CPS3 [1])) has been in operation since Feb. 2022. Its ocean model part is Meteorological Research Institute Community Ocean Model (MRI.COM) [2]. To complete time integration within the required timeframe (ideally shorter than 18 minutes (1,080 seconds) for 31-day integration) using available computational resources for CPS3, the computational performance of CPS3 needed to be improved. Code profiling revealed that bottlenecks were mainly found in the ocean model part rather than in the atmosphere model part.

Code optimization was therefore applied to MRI.COM, and this paper presents the optimization methods and computational performance achieved. In addition to the results of CPS3, performance with possible future configurations is also presented for future CPS development.

2 Overview of MRI.COM in CPS3

MRI.COM version 4.6 is used in CPS3[2]. The governing equations of MRI.COM are the primitive equations with Boussinesq approximation, and are discretized using the finite difference method in global tripolar coordinates at a horizontal resolution of 1/4° × 1/4° (an eddy permitting resolution). In the vertical direction, the equations are discretized into 60 layers using re-scaled height coordinates. A dynamical sea ice model is included. For time integration schemes in CPS3, the Leapfrog and Matsuno schemes are used.

3 Advanced code optimization

Advanced code optimization includes a) the hybrid Message Passing Interface (MPI) with OpenMP parallelization, b) use of MPI group communications, c) hiding of lateral MPI communications over computation, and d) reconsideration of compiler options. Code optimization was conducted on Cray XC50 powered by two Intel Xeon Platinum 8160 processors per node with Aries interconnect, dragonfly network topology and with the Intel Fortran compiler version 17.0.4.

While MPI is used for process-level parallelization, thread-level parallelization is implemented as compiler auto-parallelization for most parts and OpenMP for a small fraction of the relevant codes. Although compiler auto-parallelization is a simple approach, its effects are found to be very limited with Intel Fortran compiler. Instead of auto-parallelization, OpenMP directives are inserted into all thread-parallelizable parts, which also improves the performance portability of MRI.COM.

Scattering of data from one MPI process to others or gathering from others to one are typical MPI communication approaches to reading and writing data. MRI.COM supports various methods for this purpose (MPI-IO, blocking, non-blocking and group communications). The non-blocking method was selected for CPS3 here. After comparison of these methods in various combinations of MPI processes and nodes, the fastest option - MPI group communications - was chosen. A new I/O option that involves writing/reading of data for individual processes without MPI communications was added for restarting of forecasts as an option.

MPI is used to exchange variables between decomposed domains and lateral domains in MRI.COM. Blocking communications were replaced with non-blocking methods to overlap communication with packing/unpacking of halo regions into or from arrays for communications, enabling partial hiding of communication time.

Appropriate compiler options were finally applied. For XC50, the Intel compiler optimizes MRI.COM better than the Cray compiler. By using the option `fp-model=strict` which does not allow optimization that may reduce floating point accuracy, the general optimization option `O2` can be replaced with `O3`, shortening time-to-solution without losing bit-reproducible results obtained using `O2` without that option.

4 Results

4.1 CPS3

Table 1 shows computational performance based on comparison of the elapsed time of 31-day integration in the current CPS3 and CPS3 with advanced optimization. Simulation was conducted in MPMD mode in XC50 with 20 nodes for the atmosphere (TL319L100) at 120 mpi with 8 threads/process and 28 nodes for the ocean in 672 mpi with 2 threads/process. Several possible combinations of the number of threads per MPI process

	TOT	PRE	INTEG	POST	EXCH	OTHER
current [sec]	1,140	29	973	24	91	19
advanced [sec]	1,010	21	898	14	63	12
reduction [%]	11.4	28.1	7.7	39.8	30.3	37.3

Table 1: Summary of elapsed times [sec] for 31-day integration at CPS3 (TL319L100 atmosphere model coupled with eddy-permitting ocean model) with current and advanced optimization. Computational resources include 20 nodes with 160 mpi and 8 threads/process for atmosphere and 28 nodes with 672 mpi and 2 threads/process for ocean.

	TOT	PRE	INTEG	POST	EXCH	OTHER
current [sec]	2,880	398	1,547	507	198	234
advanced [sec]	1,580	166	1,078	141	136	56
reduction [%]	45.1	58.3	30.3	72.1	31.2	76.2

Table 2: As per Table 1, but for the TL479L128 atmosphere model coupled with the eddy-resolving ocean model. Computational resources include 160 nodes with 640 mpi and 12 threads/process for atmosphere. For the ocean model, 330 nodes with 3,960 mpi and 4 threads/process are used for advanced optimization and with 7,920 mpi and 2 threads/process for current optimization.

were tested, and the fastest one was selected. No MPI process was assigned to sub-domains where all grids were identified as land grids. Coupling intervals were every hour. The elapsed times were categorized into `TOT`, `PRE`, `INTEG`, `EXCH` and `OTHER`. `PRE` and `POST` include initial setup and post-process after time integration, respectively. The main time integration was counted as `INTEG`. The term `EXCH` represents time consumed to variable exchange between atmosphere and ocean models. The remaining parts and the sum of all are shown by `OTHER` and `TOT`, respectively.

The elapsed time for the current CPS3 was 1,140 seconds, which exceeds the limit by 60 seconds. The time was shortened by more than 120 seconds via advanced optimization accounting for an 11% reduction in total. The use of MPI group communication explains the speed-up in `PRE` (28%) and `POST` (40%), and exploitation of OpenMP is responsible for the shortening of `INTEG` by 8% together with improvements in lateral MPI communication. The time of array packing/unpacking in `EXCH` was also greatly shortened by up to 30% via OpenMP parallelization. Compiler optimization with `O3` also helped to reduce elapsed time in all sections.

4.2 Eddy-resolving CPS

A potential configuration for future CPS versions involves the use of an eddy-resolving resolution of 1/11° × 1/10° for an ocean model with TL479L128 for the atmosphere (hereafter, an "eddy-resolving CPS"). However, as an eddy-resolving CPS consumes far more computational resources than CPS3, it is important to evaluate related feasibility in advance. Against such a background, computational performance was measured using available resources (160 nodes for the atmosphere (640mpi, 12 threads/process) and 330 nodes (3,960mpi, 4 threads/process) for the ocean). An eddy-resolving CPS without advanced optimization runs at 7,920 mpi with two threads per process. In contrast to CPS3, the combination of Leapfrog and Matsuno schemes is replaced by the Leapfrog-Adams-Moulton scheme, which also greatly helps to shorten elapsed time but needs to be tested more for operational use. At this point, MPI processes are assigned to all sub-domains even if the domain is entirely occupied by land grids. The results of 31-day integration are presented in Table 2.

A 45% of reduction is obtained in total (`TOT`) with advanced optimization with values ranging from 30% (`INTEG`) to over 70% (`POST`) by category. Significant reductions are observed in `PRE` (58%) and `POST` (72%), where MPI group communication works effectively. OpenMP parallelization also reduced the elapsed time in `INTEG` by 30%. However, the time-to-solution of eddy-resolving CPS with advanced optimization is still far from the required time, meaning that more resources or further optimization are needed.

5 Conclusions

The computational performance of MRI.COM was extensively optimized for CPS3 and beyond. CPS3 elapsed time was reduced by up to 10%, enabling operation within the maximum time allowed. While a 45% reduction was determined for potential future configurations, further optimization is required for eddy-resolving CPS operation. As the optimization applied here was restricted, the results were bit-identical. However, use of mixed-precision computation may lead to further increase in speed. In addition, code-adaptation for GPUs (graphics processing units) is likely to be necessary for further speed-up in eddy-resolving simulations for future high-performance computing architecture.

References

[1] Hirahara Shoji, et al. Japan Meteorological Agency/Meteorological Research Institute-Coupled Prediction System version 3 (JMA/MRI-CPS3). *in preparation*, 2022.

[2] Tsujino Hiroyuki, et al. *Reference manual for the Meteorological Research Institute Community Ocean Model version 4 (MRI.COMv4)*, Vol. 80. Tech. Rep. Meteor. Res. Inst., 2017.