Performance Optimization and Evaluation of a Global Climate Application using a 440m Horizontal Mesh on the K computer

Masaaki Terai, Hisashi Yashiro, Kiyotaka Sakamoto, Shin-ichi Iga
Hirofumi Tomita, Masaki Satoh and Kazuo Minami

I. Abstract

Using almost all the compute nodes on the K computer, we evaluated the performance of the Nonhydrostatic Icosahedral Atmospheric Model (NICAM), a weather/climate application for a high-resolution global atmospheric simulation study. First, based on profiling results, we improved single node performance using six dynamical and two physical kernel programs extracted from the application, resulting in increases in the peak performance ratio of the kernel programs. Especially, the peak performance of the divergence and radiation improved from 3.5% to 16.8%, and 5.5% to 9.3%, respectively. Thereafter, we applied the optimization results to the application, and evaluated the performance by weak scaling the compute nodes from 5 to 81,920 using hybrid MPI and thread parallelization. In the evaluation, to retain the same problem size per node, we reduced the horizontal mesh size as the number of nodes increased, decreasing it to 0.44 km with 81,920 nodes. Although we achieved good scalability of the dynamical step, the elapsed time of the physical step increased with the number of nodes owing to inter-node load imbalances in the cloud microphysics. Finally, the peak performance ratio of the main loop of NICAM reached 8.3% with 81,920 nodes.

II. Introduction

NICAM [1] is one of the weather/climate applications for high-resolution global atmospheric simulation studies with explicit cloud expression. It uses a nonhydrostatic and fully compressive equation system that is discretized on an icosahedral grid arrangement. Development of NICAM began in 2000, and subsequently, the first global 3.5 km-mesh simulation was implemented on the Earth Simulator.

High-resolution simulation studies provide more precise cloud behavior and an insight into extreme weather events, such as typhoons and heavy rains associated with climate change. Such simulations require massively parallel computation, and are often implemented on state-of-the-art supercomputers. To reduce communication between compute nodes, NICAM uses a finite volume method, requiring only point-to-point communication instead of global communication, which is an advantage over the spectral transform method. Localized communication is preferable for massively parallel computation in the peta-scale computing era.

The K computer [2] is the first ten peta-flops massively parallel computer, providing the opportunity to conduct high-resolution simulations. The first global sub-km atmospheric simulation on the K computer was recently reported by Miyamoto et al [3]. To perform more effective computations in a production run, we evaluated the scalability of NICAM on the K computer.

III. Performance Evaluation

Originally, NICAM employed domain decomposition parallelization using MPI. First, we extracted kernel programs from the application, and improved single node performance with the kernels. In addition, for intra-node parallelization, we leveraged compile-time automatic thread parallelization mainly for loops in the vertical direction in the 3D atmosphere grids. Then, we applied the optimization results to the application, and evaluated weak scaling using hybrid parallelization on the K computer.

A. Single Node Performance

There are two main components in the main loop of NICAM: one is the fluid dynamics solver (referred to as the dynamics), which is the set of stencil operators with communication, while the other is the set of physical components (referred to as the physics) including the cloud microphysics, boundary layer turbulence, and atmospheric radiation, etc. We careated kernel programs from hot-spot subroutines based on the profiling results. From the dynamics, we extracted kernels for horizontal divergence operators (divergence), horizontal diffusion operators (diffusion), horizontal gradient operators (gradient), horizontal divergence damping (divdamp), 3D divergence damping (divdamp3D), and tracer advection with limiters (advlim). From the physics, we extracted a 1-moment bulk scheme of the cloud microphysics and a broadband longwave/shortwave radiative transfer scheme (radiation).

Fig. 1 shows the results of single node (a octa-core chip) optimization of the kernel programs. We attempted various optimizations such as array merging, array dimension interchange, and compiler option optimization, etc. Array dimension interchange in particular, has a major impact on performance improvement of the stencil kernels in the dynamics. Peak performances of the divergence and diffusion improved from 3.5% to 16.8% and 6.3% to 11.9%, respectively. In the physics, the peak performance ratios of the cloud microphysics and radiation improved from 4.6% to 6.4% and 5.5% to 9.3%, respectively.

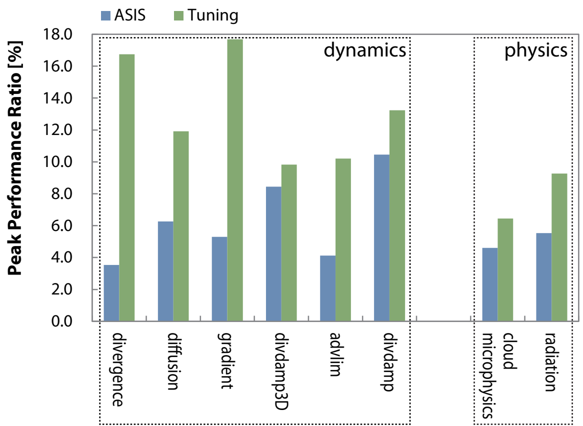

Fig. 1. Peak performance ratio in a single node.

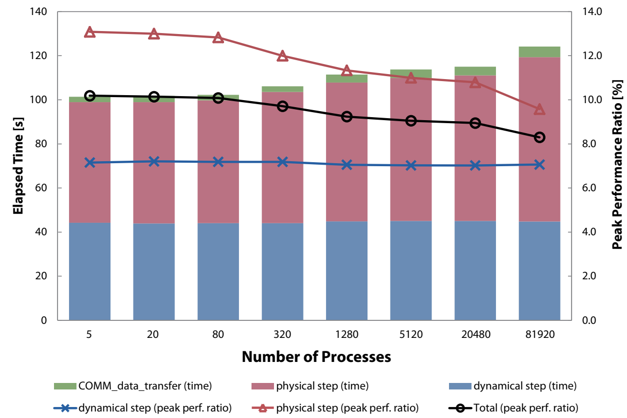

Fig. 2. Elapsed time and peak performance ratio in weak scaling.

TABLE I
EXPERIMENTAL CONDITIONS.

glevel	# of horiz. grids	# of procs.	Δx[km]
7	163840	5	56
8	655360	20	28
9	2621440	80	14
10	10485760	320	7.0
11	41943040	1280	3.5
12	167772160	5120	1.7
13	671088640	20480	0.87
14	2684354560	81920	0.44

B. Weak Scaling and Performance

We evaluated weak scaling from 5 to 81,920 nodes to analyze application behavior depending on the number of processes. To maintain the same problem size per process, the experiment was carried out under the conditions listed in Table I, where $glevel$ is the grid decomposition level relevant to the horizontal mesh size Δx. Each node was assigned one MPI process. All experiments were performed over ten time steps including one radiation procedure call.

Fig. 2 shows the elapsed time vs. peak performance ratio. To separate the elapsed time for the dynamical, physical, and communication steps, we added an MPI_barrier to the boundary of each step. We also ignored the setup time because this is relatively short in a production run, and excluded disk I/O time from the evaluation. COMM_data_transfer is the communication kernel, consisting of data packing/unpacking and MPI communication. The elapsed time for the COMM_data_transfer increases with the number of processes, and accounts for about 3.9% of the total elapsed time with 81,920 nodes.

In the dynamical step, the peak ratio is 7.2% with 5 nodes and 7.1% with 81,920 nodes. This good scalability of the dynamical step means that the floating-point operations of the stencil kernels are mostly equal. However, in the physics, the peak ratio decreased from 13.1% with 5 nodes to 9.6% with 81,920 nodes. The elapsed time increased with the number of processes despite there being no communication. Finally, the total peak performance ratio of the main loop reached 10.2% and 8.3% with 5 and 81,920 nodes, respectively.

Additional measurements in the physical step show that the average number of floating-point operations per node is constant for all weak scaling conditions. However, we found that load imbalances between the processes occurred mainly in the cloud microphysics. Fig. 3 shows a global map of the normalized numbers of floating-point operations in the cloud microphysics scheme. Regions with a large number of floating-point operations correspond to the distribution of the ice water. Regionally and seasonally uneven distributions of cloud ice, snow, and graupel cause heavy loads from floating-point operations in a small number of processes. This localization becomes more remarkable as the total number of processes increases. Our results suggest that the physical step has an intrinsic bottleneck that cannot be ignored in future high-resolution calculations.

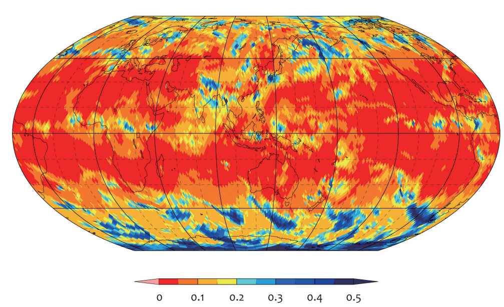

Fig. 3. Contour of normalized numbers of floating-point operations on a global map in the cloud microphysics (glevel-14).

REFERENCES

[1] M. Satoh et al. Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations. *J. of Comp. Phys.*, 227:3486–3514, 2008.

[2] H. Miyazaki et al. K computer: 8.162 petaflops massively parallel scalar supercomputer built with over 548k cores. In *ISSCC 2012*, pages 192–194, 2012.

[3] Y. Miyamoto et al. Deep moist atmospheric convection in a subkilometer global simulation. *Geophys. Res. Lett.*, 40(18):4922–4926, 2013.