Scalability of Global 0.25° Ocean Simulations Using MOM

Marshall Ward[1] and Yuanyuan Zhang[2]

[1] National Computational Infrastructure, Canberra, Australia
marshall.ward@anu.edu.au
[2] Fujitsu Australia Limited, Canberra, Australia
yuanyuan.zhang@au.fujitsu.com

Abstract. We investigate the scalability of global 0.25° resolution ocean-sea ice simulations using the Modular Ocean Model (MOM). We focus on two major platforms, hosted at the National Computational Infrastructure (NCI) National Facility: an x86-based PRIMERGY cluster with InfiniBand interconnects, and a SPARC-based FX10 system using the Tofu interconnect. We show that such models produce efficient, scalable results on both platforms up to 960 CPUs. Speeds are notably faster on Raijin when either hyperthreading or fewer cores per node are used. We also show that the ocean submodel scales up to 1920 CPUs with negligible loss of efficiency, but the sea ice and coupler components quickly become inefficient and represent substantial bottlenecks in future scalability. Our results show that both platforms offer sufficient performance for future scientific research, and highlight to the challenges for future scalability and optimization.

Keywords: ocean modeling, performance profiling, high performance computing, parallel computing.

1 Introduction

Current global climate simulations typically rely on coarse ocean models of approximately 1° resolution, but there is growing demand for a greater resolution of the ocean eddy fields, with corresponding resolutions on the order of 0.25° or 0.1° [1]. The very strong stratification of the ocean causes its most turbulent currents to emerge at these smaller resolutions [2], which are absent from coarse-resolution climate models. Such turbulent processes are expected to have a major role in the maintenance of the ocean's strongest currents, and in the vertical mixing and stratification of the ocean [3].

Along with the many scientific challenges of high-resolution climate modelling, an additional impediment in the adoption of greater resolutions in the ocean is the significant computational cost. A typical climate simulation requires decades, if not centuries, of simulation time to achieve a scientifically valuable result, and can require hundreds of such runs to perfect the tuning of their numerous parameters [1]. In such an environment, one needs the capacity to simulate many years per day. To achieve such results will require proven scalability into hundreds, if not thousands, of CPUs.

R. Denzer et al. (Eds.): ISESS 2015, IFIP AICT 448, pp. 542–551, 2015.

In this paper, we assess the ability to run high-resolution simulations on computing platforms of Australia's National Computational Infrastructure (NCI). We focus on scalability up to 1920 CPUs on the Raijin and Fujin computing platforms. We consider a selection of configurations on each platform and assess their efficiencies for use in future scientific research.

2 Methods

2.1 Numerical Model

The experiment used in this study is the global ocean-sea ice model of [4], which was based on the Geophysical Fluid Dynamics Laboratory (GFDL) CM2.5 model [5]. The numerical submodels are the Modular Ocean Model (MOM) and the Sea Ice Simulator (SIS), built from the Flexible Modeling System (FMS) framework, which also includes the ocean-ice coupler [6]. The simulations presented here use the MOM 5.1 source code release, which includes the SIS and FMS components.

The numerical grid of the ocean and sea ice models is a curvilinear grid with a nominal resolution of 0.25° and contains 1440 × 1080 horizontal grid points. The ocean and sea ice models use 50 and 6 vertical levels, respectively.

Parallelisation is achieved by decomposing the horizontal grid into tiles of equal size, with additional halo grid points surrounding each tile. For each timestep, the halos of adjacent and diagonally adjacent tiles are updated, requiring 8 messages per field.

Standard experiments are run for 31 days, using 1488 timesteps of 1800 second resolution. Each ocean timestep requires 72 additional sea ice timesteps. A longer simulation time was chosen to represent a typical integration time for scientific analysis, and to reduce any statistical variability of the observed walltimes. Simulations using 960 CPUs were run three times to confirm the reproducibility of the results, and the median walltimes was used to select the run in this study. For all other runs, a single simulation was used for each configuration.

Diagnostic output consists of 4 three-dimensional fields, which are saved to disk after every 5 days. This output rate was chosen to reflect a comparable rate of scientific value while also not overshadowing the model calculations.

2.2 Raijin

Raijin is the principal supercomputer of the NCI National Facility. It is a Fujitsu PRIMERGY cluster comprised of 3592 computing nodes, with each containing two 2.6 GHz 8-core Intel Xeon Sandy Bridge (E5-2670) CPUs, with a total core count of 57472. Turbo boost is enabled for these runs, increasing the maximum clock speed of 16 cores to 3.0 GHz. Operational nodes have approximately 32 GiB of memory, with more memory available on selected nodes.

The cluster network uses InfiniBand (IB) FDR-14 interconnects, which provide peak (4x aggregate) transfer speeds up to 56 Gb/s between nodes. There are 18 nodes connected to each top-of-rack (ToR) IB switch. The 72 nodes in each rack use 4 ToR switches. Each ToR switch connects to a top-level director switch, with 3 connections

per ToR to each director. Because of the high usage rate of Raijin, it is prohibitive to arrange jobs within the network, and we assume that all experiments in this study depend on the director switches.

The operating system platform is CentOS 6.5, which uses the Linux 2.6.32 kernel. Jobs are managed using the PBSPro scheduler, and files are kept under a Lustre 2.5 filesystem.

We use the Intel Compiler Suite 14.3.172 for model compilation. MPI communication uses Open MPI 1.8.2 and data IO uses NetCDF 4.3.0, which was built using HDF5 1.8.10. The model is built with the -O2 and -xhost optimisation flags.

We consider three experiment configurations on Raijin: a standard configuration using 16 model processes per node (herein "default"), a similar configuration using hyperthreaded cores, and a third configuration running 12 model processes per node (herein "12 PPN"). For the 12 PPN configuration, turbo-boosted clock speeds are increased to 3.1 GHz.

2.3 Fujin

Fujin is NCI's Fujitsu FX10 cluster consisting of 96 compute nodes, each containing a 1.848 GHz 16-core SPARC64 IXfx CPU. For our investigation, numerical experiments were limited to a maximum of 84 nodes, or 1344 cores.

The interconnect is a streamlined version of the K-Computer's Tofu interconnect system, and consists of 8 serially connected Tofu units. Each unit contains 12 compute nodes arranged as a 3D torus of shape $2\times 3\times 2$. Nodes are connected by 10 bidirectional links with 5 Gb/s transfer rates, yielding a peak transfer rate of 100 Gb/s between nodes. Tofu interconnects also include the Tofu barrier hardware module for optimised MPI reductions and broadcasts, but we do not use it in this study.

When compared to Raijin, the lower clock speed of Fujin's CPUs will necessarily result in lower optimal performance for computationally-bound software. However, the fixed Tofu interconnect offers the potential for optimized node layout and efficient scalability for higher CPU counts.

The FX10 operating system is Fujitsu XTC OS 1.3.6, with a Linux 2.6.25 kernel. Jobs are managed by the scheduler of the Parallelnavi suite. The filesystem is the Fujitsu Exabyte File System (FEFS) 1.3.1, which is based on the Lustre 1.8.5 filesystem.

The model is compiled using the Fujitsu compiler suite 1.2.1 and includes Fujitsu's MPI library, which is based on Open MPI 1.4.3. Data IO uses NetCDF 4.3.2, built with HDF5 1.8.12 and NetCDF-Fortran 4.2. The model is compiled using the -Kfast, -Kocl, and -O2 optimization flags.

We consider two configurations on Fujin, denoted here as snake and ladder layouts. Node placement for a single Tofu unit for each layout is illustrated in Figure 1. The snake layout guarantees that all communications is exclusively between neighbour nodes, but also requires that each node span an entire latitude line. The ladder layout divides each latitude line across two nodes, allowing for smaller tile widths, but it also requires diagonally adjacent tiles to send data through non-neighbour nodes.

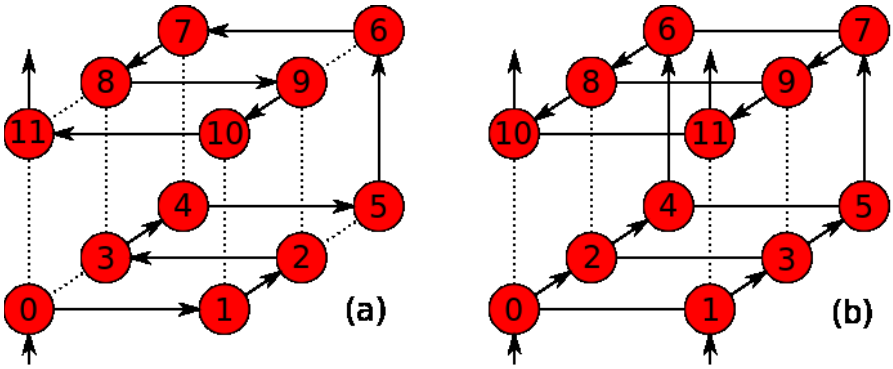

Fig. 1. Node layouts for each Tofu unit. Each node number corresponds to 16 domain tiles. Figure (a) shows the snake layout and figure (b) shows the ladder layout.

2.4 Timing

On Raijin, we use the IPM 2.0.2 profiler to construct basic runtime statistics. Model runtime and MPI usage times are based on IPM output logs. On Fujin, runtime and MPI usage times are from the `fipp` profiler output logs.

Submodel runtimes are provided by the internal timers of the MOM ocean model, which are based on the `system_clock` subroutine of the Intel Fortran compiler and the system clock of the Linux kernel.

3 Results

3.1 Platform Configuration

Figure 2a shows the total runtime for a 480-CPU experiment across different hardware platforms and configurations. The configuration choice on Raijin had a major impact on performance, with observed speedups of 28% and 60% for hyperthreaded and 12 PPN experiments, respectively. Process layout on Fujin had an observable but less notable impact, with the snake layout achieving a 15% speedup in comparison to the ladder layout.

Experiment runtimes on Raijin are faster than on Fujin, with the fastest runs on Raijin outperforming the Fujin jobs by more than a factor of two. Much of this difference in performance can be attributed to the clock speeds of the two platforms. However, the clock speed ratio of the Raijin and Fujin CPUs is only 1.62, and cannot alone account for the differences in performance.

Figure 2b attempts to clarify this performance difference across the platforms by estimating the mean number of cycles per CPU with the following formula:

$$N_c = f(1-p)\tau_o$$

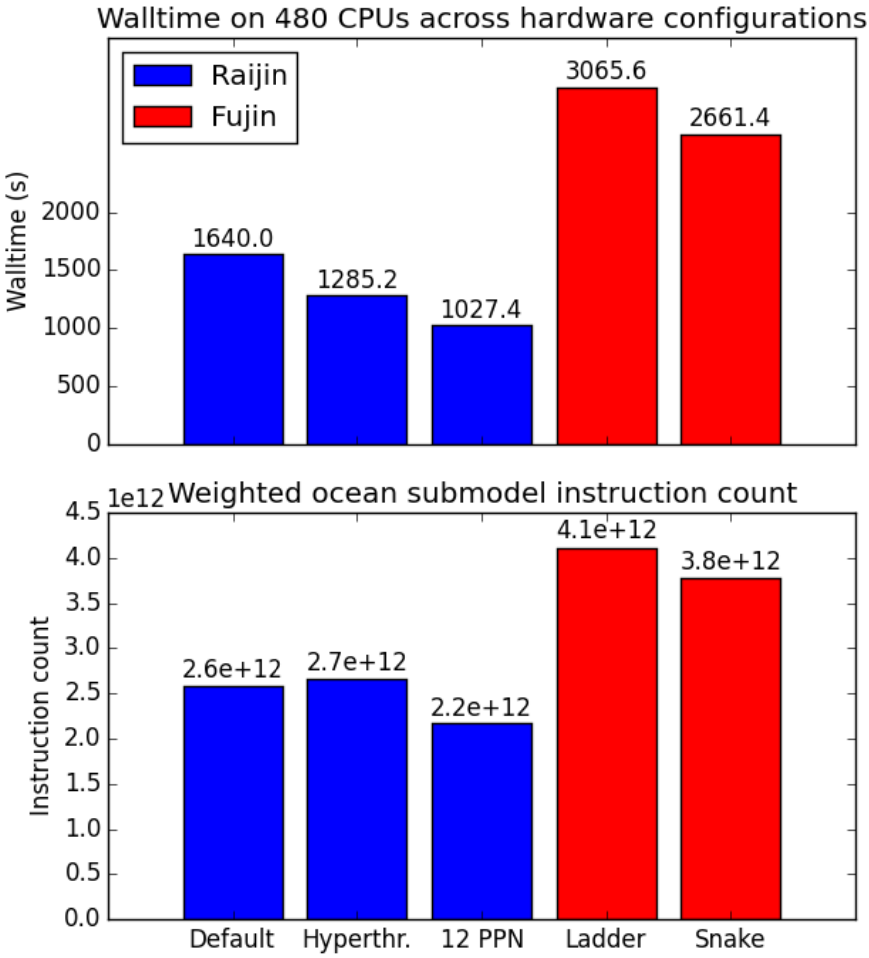

Fig. 2. Model performance across platforms and configurations. Figure (a) compares walltimes in seconds, and figure (b) compares cycle counts in the ocean submodel.

where f is the CPU clock speed, p is the mean fraction of MPI instructions across all ranks, and τ_o is the ocean submodel runtime. We focus on the ocean submodel because it is strongly dominated by floating point arithmetic, and is shown in the next section to demonstrate high scalability up to 1920 CPUs.

N_c is an imperfect estimate of cycle count, since total communication time may not accurately represent communication time in the ocean model. There may also be additional communication costs outside of the MPI library. But it can provide an approximation of the number of computational cycles, as well as the relative efficiency of vector arithmetic on each platform.

After correcting for clock speed and focusing on the ocean model, our estimate indicates that the performance of the two platforms is comparable, with SPARC operations requiring only 15% more cycles than the x86 operations for its simulation.

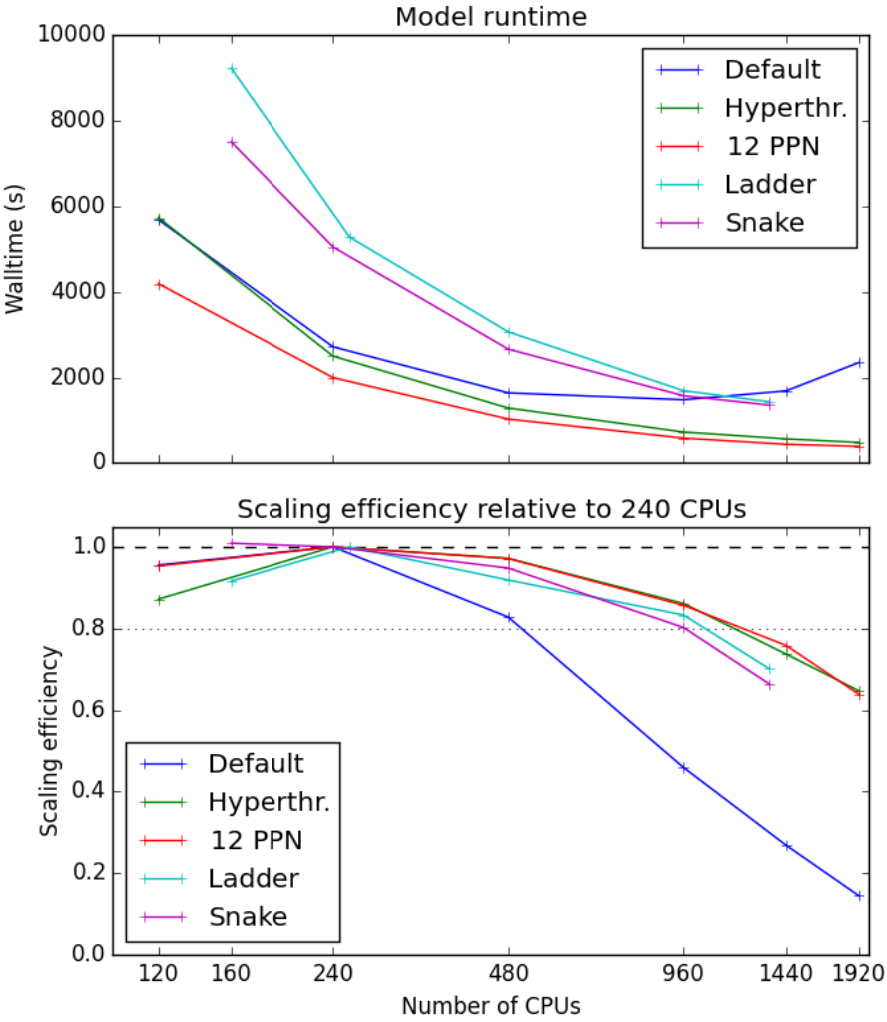

Fig. 3. Model walltime (a) and scaling efficiency (b)

3.2 Scaling

The general performance and scalability of each experiment across different CPUs is shown in Figure 3. The general trends shown in the previous section are also observed in Figure 3a, with simulations on Raijin generally outperforming those on Fujin. Hyperthreaded and 12 PPN jobs continue to outperform the default runs on Raijin, and snake layouts outperform ladder layouts on Fujin.

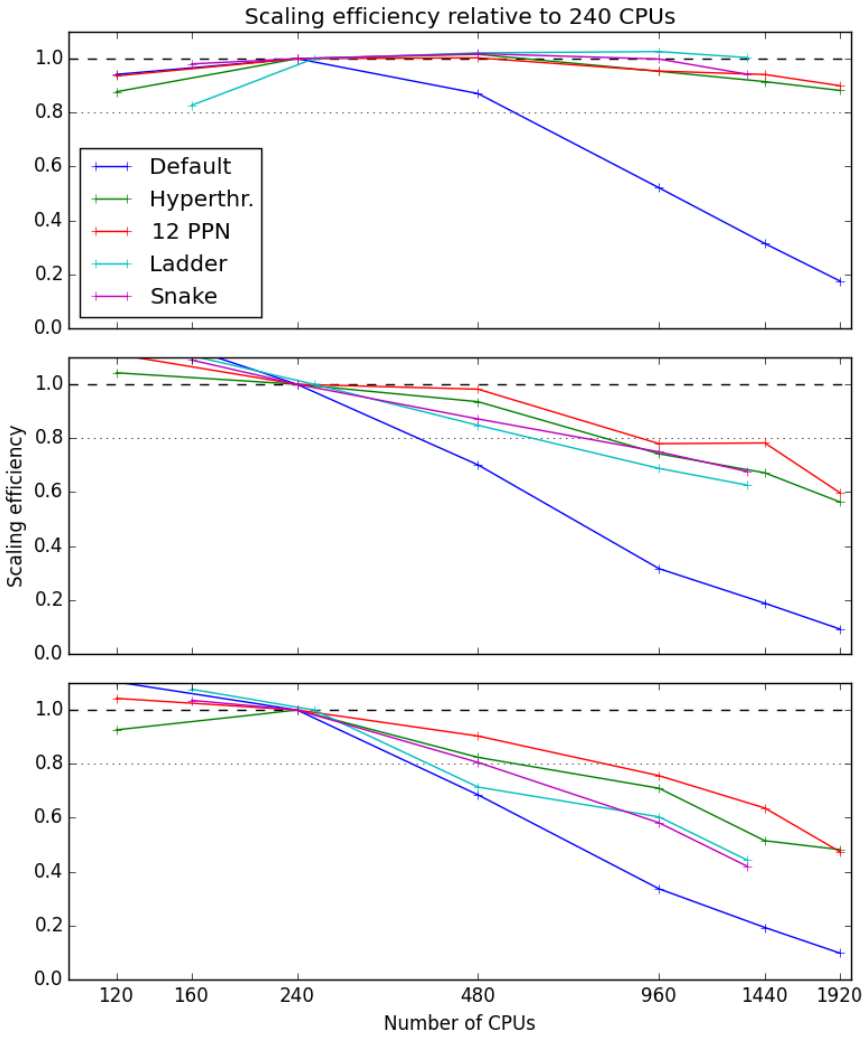

Fig. 4. Scaling efficiency for the ocean (a), sea ice (b), and coupler (c) submodels

The most notable observation is the dramatic loss of performance of the default experiments on Raijin, where scalability begins to diminish at 480 CPUs and wall-times begin to increase after 960 CPUs. Scalability is restored by either introducing hyperthreading or by using fewer cores, indicating the presence of a computational bottleneck related to the number of processes per node.

The relative efficiency of each configuration is shown in Figure 3b, which is computed as

$$\varepsilon = \frac{r\tau_r}{n\tau_n}$$

where n is the number of CPUs, τ_n is the runtime using n CPUs, and r is the reference experiment, which is 240 CPUs for all configurations except the ladder Fujin runs, which use 255 CPUs.

Figure 3b shows that, with the exception of the default Raijin setup, all runs demonstrate scalable performance up to 960 CPUs and maintain an efficiency greater than 80%. Experiments on Raijin scale slightly better than on Fujin, although the results are comparable over all platforms and configurations. Beyond 960 CPUs, all experiments begin to drop substantially and no longer provide a reasonable level of performance.

Model performance can be further clarified by investigating the scalability of the respective submodels, as shown in Figure 4. Figure 4a shows that, despite a reduction of general scalability around 960 CPUs, the ocean submodel continues to scale efficiently to 1920 CPUs. However, the sea ice and coupling model subcomponents, shown in Figures 4b and 4c, grow progressively worse in performance as the number of CPUs increase, indicating major bottlenecks in the sea ice model and submodel flux exchange components.

There is some evidence of improved efficiency in the ocean model, indicating that the communication bottlenecks in the sea ice and coupler models may also be freeing additional resources for the ocean model. Scaling efficiency on Fujin appears to be better than Raijin for the ocean model, although it is notably lower in the other model components.

4 Discussion

4.1 Configuration

Scalability on Raijin beyond 480 CPUs was only possible after either reducing the number of processes per node, or enabling hyperthreading on the core. In both situations, we reduced the number of processes per computational core. While the underlying cause is not known, one point of consideration is use of subthreads within Open MPI. Since each model instance create four MPI subthreads, there is a competition for resources between these processes on each core. By allowing the kernel to shift some of these jobs to other computational cores, we may be relieving any bottlenecks related to shared memory management or context switching. But further investigations are required to confirm this explanation, and to determine if it is a consequence of x86 architecture, kernel scheduling, or Open MPI implementation.

Scalability on Fujin was not affected by processor layout, although the performance of the snake layout was measurably faster than the ladder layout. From this, we conclude that even a modest dependence on non-neighbor communication, such as in our ladder layouts, can have a detrimental effect on performance, and that snake layouts should be used when possible.

4.2 Platform Comparison

An additional observation in this study is the significant discrepancy in performance between the Raijin and Fujin platforms. Our estimate of cycle count indicates that

most of this difference can be attributed to CPU clock speed. However, the x86 CPUs still demonstrate somewhat better performance per cycle. This can be attributed to various factors, such as differences in CPU cache architecture or compilers, but a more thorough investigation would be required to confirm the level of efficiency and underlying cause.

An unexplored opportunity for improved hardware performance on Fujin is the Tofu barrier acceleration of the MPI reductions and broadcasts. Such operations are a dominant part of most diagnostic calculations, and the Tofu barrier could have an major impact on future simulations with high levels of diagnostic output.

Another point of consideration is the increasing trend of greater numbers of cores on each node, including MIC architectures such as Xeon Phi accelerators. Although Raijin outperformed Fujin in runtime, scalability was not possible without the introduction of redundant computational cores. No such additional resources were required to achieve scalable results on Fujin.

4.3 Submodel Performance

The difference in ocean scalability versus the ice and coupler submodel also indicates a potential target for optimization. Running the ocean and sea ice serially on each process is a potential bottleneck that could be addressed by moving the sea ice calculations onto separate core. In such a configuration, load balancing would become a greater concern, and a separation of ocean and ice would put a greater burden on the coupler. But given the inability of SIS to scale beyond 480 CPUs, and the very large number of sea ice timesteps required per ocean timestep, this may become a requirement in the future.

The poor scalability of the coupling subroutines indicate that a parallelisation of of ocean and sea ice timesteps will not be sufficient to achieve future scalability. Efficient field exchange between models must also be a target of future optimisation efforts.

5 Conclusion

We have shown that the global 0.25° resolution ocean-sea ice configurations of the MOM ocean model can run efficiently on both x86 PRIMERGY and SPARC FX10 platforms. On the x86-based Raijin system, we were able to simulate over 10 model years per day at a sufficiently high level of efficiency. On the FX10 Fujin platform, we can simulate over 4 model years per day at a similar level of efficiency.

In order to achieve a high level of performance on the Raijin platform, we were required to either reduce the number of model processes per node to 12, leaving four cores to run idle during the simulation, or to enable hyperthreading on our CPUs, allowing two computational threads per core. Although reducing the number of cores yields a greater level of performance, hyperthreaded runs yield comparable results without any idle cores, thereby achieving much greater efficiency. When all 16 cores on each node were used without hyperthreading, the runtimes increased by a factor of three and were unable to scale beyond 480 CPUs.

Comparison of the submodels within the ice-ocean configuration shows that the ocean model scales up to 1920 cores with only a negligible loss of efficiency, while the sea ice and coupling components cease to scale after 480 CPUs. Future efforts to improve the scalability of MOM should target these subcomponents.

Both platforms have proven capable of running global high-resolution simulations of the ocean, and enable scientists to investigate the impacts of turbulent-scale processes on global climate. Development towards resolutions of 0.1° and beyond will require further investigations of the underlying codes and their resource requirements, and ongoing collaborations between industry and the academic research community.

Acknowledgements. This work was conducted as part of the ACCESS Optimisation Project, a collaboration between NCI, Fujitsu, and the Australian Bureau of Meteorology. Research was undertaken with the assistance of resources from NCI, which is supported by the Australian Government. We are grateful to Mark Cheeseman of NCI, Symon Swann of Fujitsu Australia, and Motohiro Yamada of Fujitsu Japan for constructive reviews on an earlier draft of this paper.

References

1. Taylor, K.E., Stouffer, R.J., Meehl, G.A.: An Overview of CMIP5 and the Experiment Design. Bull. Amer. Meteor. Soc. 93, 485–498 (2012)
2. Chelton, D.B., de Szoeke, R.A., Schlax, M.G., El Naggar, K., Siwertz, N.: Geographical variability of the first baroclinicRossby radius of deformation. J. Phys. Oceanogr. 28, 433–459 (1998)
3. Farneti, R., Delworth, T.L., Rosati, A.J., Griffies, S.M., Zeng, F.: The Role of Mesoscale Eddies in the Rectification of the Southern Ocean Response to Climate Change. J. Phys. Oceanogr. 40, 1539–1557 (2010)
4. Spence, P., Griffies, S.M., England, M.H., Hogg, A., Mc, C., Saenko, O.A., Jourdain, N.C.: Rapid subsurface warming and circulation changes of Antarctic coastal waters by poleward shifting winds. Geophys. Res. Lett. 41, 4601–4610 (2014)
5. Delworth, T.L., Rosati, A., Anderson, W.G., Adcroft, A., Balaji, V., Benson, R., Dixon, K.W., Griffies, S.M., Lee, H.C., Pacanowski, R.C., Vecchi, G.A., Wittenberg, A.T., Zeng, F., Zhang, R.: Simulated climate and climate change in the GFDL CM2.5 high-resolution coupled climate model. Journal of Climate 25(8) (2012)
6. Griffies, S.M.: Elements of the Modular Ocean Model (MOM), GFDL Ocean Group Technical Report No. 7. NOAA/Geophysical Fluid Dynamics Laboratory. 618 + xiii pages (2012 release)