The GPU version of LICOM3 under the HIP framework and its large-scale application

Pengfei Wang[1,3], Jinrong Jiang[2, 4*], Pengfei Lin[1,4*], Mengrong Ding[1], Junlin Wei[2],Feng Zhang[2],Lian Zhao[2], Yiwen Li[1], Zipeng Yu[1], Weipeng Zheng[1,4], Yongqiang Yu[1,4], Xuebin Chi[2, 4] and Hailong Liu[1,4*]

[1]State Key Laboratory of Numerical Modeling for Atmospheric Sciences and Geophysical Fluid Dynamics (LASG), Institute of Atmospheric Physics (IAP), Chinese Academy of Sciences (CAS), Beijing 100029, China
[2]Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China
[3]Center for Monsoon System Research (CMSR), Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100190, China
[4]University of Chinese Academy of Sciences, Beijing 100049, China

Correspondence to: Drs. Jinrong Jiang [jjr@sccas.cn], Pengfei Lin [linpf@mail.iap.ac.cn] and Hailong Liu [lhl@lasg.iap.ac.cn]

Abstract. A high-resolution (1/20°) global ocean general circulation model with Graphics processing units (GPUs) code implementations is developed based on the LASG/IAP Climate system Ocean Model version 3 (LICOM3) under Heterogeneous-compute Interface for Portability (HIP) framework. The dynamic core and physics package of LICOM3 are both ported to the GPU, and 3-dimensional parallelization is applied. The HIP version of the LICOM3 (LICOM3-HIP) is 42 times faster than the same number of CPU cores when 384 AMD GPUs and CPU cores are used. The LICOM3-HIP has excellent scalability; it can still obtain a speedup of more than four on 9216 GPUs comparing to 384 GPUs. In this phase, we successfully performed a test of 1/20° LICOM3-HIP using 6550 nodes and 26200 GPUs, and at the grand scale, the model's time to solution can still obtain an increasing, about 2.72 simulated years per day (SYPD). The high performance was due to putting almost all of the computation processes inside GPUs, reducing the time cost of data transfer between CPUs and GPUs. Simultaneously, a 14-year spin-up integration following phase 2 of the Ocean Model Intercomparison Project (OMIP-2) protocol of surface forcing has been conducted, and the preliminary results have been evaluated. We found that the model results have little difference from the CPU version. Further comparison with observations and lower-resolution LICOM3 results suggests that the 1/20° LICOM3-HIP can reproduce the observations and produce many smaller-scale activities, such as submesoscale eddies and frontal scales structures.

1 Introduction

Numerical models are a powerful tool for weather forecasts and climate prediction and projection. High-resolution atmospheric, ocean and climate models remain significant scientific and engineering challenges because of the enormous computing, communication, and input/output (IO) involved. Kilometer-scale weather and climate simulation start to emerge recently (Schär et al., 2020). Due to the tremendous increase in computation cost, such models will only work with extreme-scale high-performance computers and new technologies.

The global ocean general circulation models (OGCMs) are a fundamental tool for oceanography research, ocean forecast, and climate change research (Chassignet et al., 2019). Such models' performance is determined mainly by model resolution and sub-grid parameterization, and surface forcing. The horizontal resolution of global OGCMs has increased to about 5-10 km, which is also called eddy-resolving models. The increasing resolution will significantly improve the simulation of the western boundary currents, mesoscale eddies, fronts and jets, and the currents in narrow passages (Hewitt et al., 2017). Meanwhile, the ability of an ocean model in simulating the energy cascade (Wang et al., 2019), the air-sea interaction (Hewitt et al., 2017), and the ocean heat uptake (Griffies et al., 2015) will be improved with the increasing resolution. All these will effectively improve ocean models' performance in simulation and prediction of the ocean circulation. Additionally, the latest numerical and observational results show that the much smaller eddies (sub-mesoscale eddies with a spatial scale of about 5-10 km) are crucial to the vertical heat transport in the upper-ocean mixed layer and also significant to biological processes (Su et al., 2018). To resolve the smaller-scale processes raises a new challenge for the horizontal resolution of OGCMs, which also demands much more computing resources.

Heterogeneous computing has become a development trend of high-performance computers. In the latest TOP500 supercomputer list released in Nov. 2020, Central Processing Unit (CPU) and Graphics Processing Unit (GPU) heterogeneous machines account for six of the top 10. After the NVIDIA Corporation provided the supercomputing technics on GPU, more and more ocean models applied these high-performance acceleration ways to conduct weather or climate simulations. Xu et al. (2015) developed POM.gpu, a full GPU solution based on the mpiPOM on a cluster, and gained 6.8 times energy reduction. Yashiro et al. (2016) deployed the NICAM model on the TSUBAME supercomputer, and the model sustained a double-precision performance of 60T Flops on 2560 GPUs. Yuan et al. (2020) developed a GPU version of a wave model with 2 V100 cards and obtained 10-12 times speedup than the 36 cores of CPU. Yang et al. (2016) implemented a fully implicit β-plane dynamic model with 488m grid spacing on the TaihuLight system and achieving 7.95P Flops. Fuhrer et al. (2018) reported a 2-km regional Atmospheric General Circulation Model (AGCM) test using 4888 GPU cards and obtained simulation performance for 0.043 simulated years per wall clock day (SYPD). Zhang et al. (2020) successfully ported a high-resolution (25 km atmosphere and 10 km ocean) Community Earth System Model in TaihuLight supercomputer, and obtained 1-3.4 SYPD.

At the same time, the AMD company also provides its GPU solutions. In general, AMD GPU uses Heterogeneous Compute Compiler (HCC) tools to compile codes; and they cannot use the Compute Unified Device Architecture (CUDA) development environments, which is supported by the NVIDIA GPU only. Therefore, due to the wildly used and numerous CUDA learning resources, AMD developers have to study two kinds of GPU programming skills. AMD's Heterogeneous-compute Interface for Portability (HIP) is an open-source solution to deal with this problem. It provides a higher-level framework to contain these two types of lower-level development environments, i.e., CUDA and HCC, simultaneously. The HIP code's grammar is like that of the CUDA code, and with a simple convert tool, the code can be compiled and run at CUDA and AMD architects, respectively. The HCC/OpenACC is convenient for AMD GPU developers before the HIP is popular from the coding

viewpoint. Another reason is that CUDA GPU has more market share in the present. It is believed that more and more codes will be ported to the HIP in the future. However, almost no ocean models use the HIP framework so far.

The study aims to develop a high-performance OGCM based on the LICOM3, which can be run on AMD GPU architecture using the HIP framework. Here, we will focus on the model's best computing performance and its practical usage for research

70 and operation purposes. Section 2 is the introduction of the LICOM3 model. Section 3 contains the main optimizing of the LICOM3 under the HIP. Section 4 covers performance analysis and model verification. Section 5 is for discussion, and the conclusions are in Section 6.

2 LICOM3 model and experiments

2.1 LICOM3 model

75 In this study, the targeting model is LASG/IAP Climate system Ocean Model version 3 (LICOM3), which has been developed in the late 1980s (Zhang and Liang, 1989). Now the LICOM3 is the ocean model for two air-sea coupled models of the CMIP6, the Flexible Global Ocean-Atmosphere-Land System model version 3 with a finite-volume atmospheric model (FGOALS-f3; He et al., 2020) and Flexible Global Ocean-Atmosphere-Land System model version 3 with a grid-point atmospheric model (CAS FGOALS-g3; Li et al., 2020). The LICOM version 2 (LICOM2.0, Liu et al., 2012) is also the ocean model of the CAS

80 Earth System Model (CAS-ESM, Zhang, et al., 2020). The paper to fully describe the new features and baseline performances of LICOM3 are preparing.

In recent years, the LICOM model was substantially improved based on the LICOM2.0 (Liu et al., 2012). There are three main aspects: First, the coupling interface of LICOM has been upgraded. Now the NCAR flux coupler version 7 is employed (Lin et al., 2016), in which the memory usage has been dramatically reduced (Craig et al., 2012). It makes the coupler suitable to

85 be applied to high-resolution modeling.

Second, both the orthogonal curvilinear coordinate (Murray, 1996; Madec & Imbard, 1996) and the tripolar grid have been introduced in the LICOM. Now, the two poles are at (65°E, 60.8°N) and (115°W, 60.8°N) for the 1° model, at (65°E, 65°N) and (115°W, 65°N) for the 0.1° model, and at (65°E,60.4°N) and (115°W, 60.4°N) for the 1/20° model of the LICOM. After that, the zonal filter in the high latitude, particularly in the northern hemisphere, has been eliminated, which significantly

90 improves the scalability and efficiency of the parallel algorithm of the LICOM3 model. In addition, the dynamic core of the model has also been updated accordingly (Yu et al., 2018), including applying a new advection scheme for the tracer formulation (Xiao, 2006) and a vertical viscosity for the momentum formulation (Yu et al., 2018).

Third, the physical package has been updated, including introducing an isopycnal and thickness diffusivity scheme (Ferreira et al., 2005) and the vertical mixing due to internal tides breaking at the bottom (St. Laurent et al., 2002). The coefficient of

95 both isopycnal and thickness diffusivity is set to 300m^2 s^{-1} as the depth is within the mixed layer or the water depth is shallower than 60 m. The upper and lower boundary values of the coefficient are 2000 and 300 m^2 s^{-1}, respectively. Additionally, the

chlorophyll-depended solar shortwave radiation penetration scheme of Ohlmann (2003), the isopycnal mixing scheme (Redi, 1982; Gent & McWilliams, 1990), and the vertical viscosity and diffusivity schemes (Canuto et al. 2001; 2002) are employed in LICOM3.

100 Both the low-resolution (1°) (Lin et al., 2020) and high-resolution (1/10°) (Li Y. et al., 2020) stand-alone LICOM3 are also involved in the OMIP-1 and OMIP-2; their outputs can be downloaded from websites. The two versions of LICOM3's performances compared with other CMIP6 ocean models are shown in Tsujino et al. (2020) and Chassignet et al. (2020), respectively. The 1/10° version has also been applied to do short-term ocean forecasts (Liu et al., 2020, under review).

2.2 Configurations of models

105 To investigate the GPU version, we have employed three configurations in the present study. They are 1°, 0.1°, and 1/20°. Details of these models are listed in Table 1. The numbers of horizontal grid points for the three configurations are 360×218, 3600×2302, and 7200×3920, respectively. The vertical levels for the low-resolution are 30, while they are 55 for the other two high-resolution models. From 1° to 1/20°, it increases the computational effort by about 8000 (20^3) times (considering the 20 times for decreasing the time step), plus the vertical resolution increase from 30 to 55, totally approximately 15000

110 times. The original CPU version of 1/20° with MPI parallel on Tianhe-1A only reached 0.31 SYPD using 9216 CPU cores. This speed will slow down the 10-year spin-up simulation of LICOM3 to more than one month, not practical for climate research. Therefore, such simulations are suitable for extreme-scale high-performance computers by applying the GPU version. Besides the different grid points, three main aspects are different among the three experiments, particularly between 1° version and the other two versions. First, the horizontal viscosity schemes are different: using Laplacian for 1° and biharmonic for

115 1/10° and 1/20°. The viscosity coefficient is one order smaller for the 1/20° version than for the 1/10° version, namely, -1.0×10^9 m^4/s for 1/10° vs -1.0×10^8 m^4/s for 1/20°. Second, although the forcing including dataset (JRA55-do; Tsujino et al., 2018) and the bulk formula for the three experiments are all standard of the OMIP-2, the periods and temporal resolutions of the forcing fields are different: 6-hour data from 1958 to 2018 for the 1° version, and daily mean data in 2016 only for both the 1/10° and 1/20° versions. Third, the 1° version is coupled with a sea ice model of the CICE4, via NCAR's flux coupler

120 version 7, while the two higher-resolution models are stand-alone, without a coupler or sea ice model. Additionally, the two higher-resolution experiments employ the new HIP version of LICOM3 (i.e., LICOM3-HIP). The low-resolution experiment does not, which was the CPU version of LICOM3 and the same as the version submitted to OMIP (Lin et al., 2020). We also listed all the important information in Table 1, such as the bathymetry data and the bulk formula, etc., though these items are similar in the three configurations.

125 The spin-up experiments for two high-resolution versions are conducted for 14 years, forced by the daily JRA55-do dataset in 2016. The atmospheric variables include the wind vectors at 10-m, air temperature at 10-m, relative humidity at 10-m, total precipitation, the downward shortwave radiation flux, the downward longwave radiation flux, and the river runoff. According

to the kinetic energy evolution, the models reach a quasi-equilibrium state after more than ten years of spin-up. The daily mean data are output for store and analysis.

2.3 Hardware and software environments of the testing system

The two higher-resolution experiments were performed on a heterogeneous Linux cluster supercomputer, located at the Computer Network Information Center (CNIC) of the CAS, China. This supercomputer consists of 7200 nodes (6 partitions or rings, each partition has 1200 nodes), with a 1.9 GHz X64 CPU of 32 cores on each node. Also, each node is equipped with four gfx906 AMD GPU cards with 16 GB memory. The GPU has 64 cores, total of 2560 threads on each card. The nodes are interconnected through the high-performance InfiniBand (IB) networks (3-level fat-tree architecture using Mellanox 200Gb/s HDR InfiniBand, whose measured point-to-point communication performance is about 23GB/s). The OpenMPI version 4.02 is employed for compiling, and the AMD GPU driver and libraries are rocm-2.9 integrated with HIP version 2.8. The storage file system of the supercomputer is ParaStor300S with a 'parastor' file system, whose measured write and read performance is about 520GB/s and 540 GB/s.

3 LICOM3 GPU code structure and optimization

3.1 Introduction to HIP on an AMD hardware platform

AMD's HIP is a C++ runtime API and kernel language. It allows developers to create portable applications that can be run on AMD's accelerators and CUDA devices. The HIP provides an API for an application to leverage GPU acceleration for both AMD and CUDA devices. It is syntactically similar to CUDA, and most CUDA API calls can be converted in placing of the character "cuda" by "hip" (or "Cuda" by "Hip"). The HIP supports a strong subset of CUDA runtime functionality, and its open-source software is currently available on GitHub (https://rocmdocs.amd.com/en/latest/Programming_Guides/HIP-GUIDE.html).

Some supercomputers install NVIDIA GPU cards, such as P100 and V100, some install AMD GPU cards, like AMD VERG20, etc.. Hence, our HIP version LICOM3 can adapt and gain very high performance at different supercomputer centers, such as Tianhe-2 and AMD clusters. Our coding experience on AMD GPU indicates that the HIP is a good choice for high-performance model development. Meanwhile, the model version is easy to keep consistent in these two commonly used platforms. In the following, the successful simulation of LICOM3-HIP is confirmed to be adequate to employ HIP.

Figure 1 demonstrates the HIP implementations to support different types of GPUs. Besides the differences in naming and including libraries, there are other differences between HIP and CUDA: 1) AMD Graphics Core Next (GCN) hardware "warp" size = 64; 2) device and host pointers allocated by HIP API use flat addressing (unified virtual addressing is enabled by default); 3) dynamic parallelism not currently supported; 4) some CUDA library functions do not have AMD equivalents, and 5) shared memory and registers per thread may differ between AMD and NVIDIA hardware. Despite these differences, most of the CUDA codes in applications can be easily translated to the HIP and vice versa.

Technical supports of CUDA and HIP also have some differences. For example, CUDA applications have some CUDA-aware
160 MPI to direct MPI communication between different GPU space nodes, but HIP applications have no such functions so far. We have to transfer data from GPU memory to CPU memory for exchanging data with other nodes and then transfer them back to the GPU memory.

3.2. Core computation process of LICOM3 and C transitional version

We tried to apply the LICOM on a heterogeneous computer about five years earlier, cooperating with the NVIDIA Corporation.
165 The LICOM2 was adapted to NVIDIA P80 by OpenACC technical (Jiang et al., 2019). That was a convenient implementation of LICOM2-gpu using 4 NVIDIA GPUs to achieve a 6.6 speedup compared to 4 Intel CPUs, but its speedup was not so good when further increasing the GPU number.

This time we started from the CPU version of LICOM3. The code structure of LICOM3 includes four steps. The first step is the model setup; it involves MPI partition and ocean block distribution. The second stage is model initialization, which includes
170 reading the input data and initialize the variables. The third stage is integration loops, the core computation of the model. Three explicit time loops, which are for tracer, baroclinic and barotropic steps, are in one model day. The outputs and final processes are included in the fourth step.

Figure 2 shows the flowchart of LICOM3. The major processes within the model time integration include baroclinic, barotropic, and thermohaline equations, which are solved by the leapfrog or Euler forward scheme. There are seven individual
175 subroutines, such as "readyt", "readyc", "barotr", "bclinc", "tracer", "icesnow", and "convadj". When the model finishes one day's computation, the diagnostics and output subroutine will write out the predicted variables to files. The output files contain all the necessary variables to restart the model and for analysis.

To obtain high performance, using the native GPU development language is more efficient. In the CUDA development forum, both CUDA-C and CUDA-Fortran are provided; however, Fortran's support is not as good as that for C++. We plan to push
180 all the core process codes into GPUs; hence, the seven significant subroutines' Fortran codes must be converted to HIP/C++. Due to the complexity and many lines in these subroutines (approximately 12000 lines Fortran code) and making sure the converted C/C++ codes be correct, we rewrote them to C before finally converting them to HIP codes.

A bit-reproducible climate model produces the same numerical results for a given precision, regardless of the choice of domain decomposition, the type of simulation (continuous or restart), compilers, and the architectures executing the model (i.e., the
185 same hardware and software conduct the same result). The C transitional version (not fully C code, but the seven cores subroutine) is bit-reproducible with the F90 version of the LICOM3 (the binary output data are the same under Linux with the "diff" command). We have also tested the execution time. The Fortran and C hybrid version's speed is slightly faster than the original Fortran code, less than 10%. Figure 3 shows a speed benchmark by the LICOM3 for 100km and 10km running on an Intel platform. The results are the time running one model month for a low-resolution test and one model day for a high-
190 resolution test. The details of the platform are in the caption of Figure 3. The results indicate that we have successfully ported these kernels from Fortran to C.

This C transitional version becomes the starting point of HIP/C++ codes and reduces the complexity of developing the HIP version of the LICOM3.

3.3. Optimization and tuning methods in LICOM3-HIP

195   The unit of computation in LICOM3-HIP is a horizontal grid point. For example, 1/20° corresponds to 7200×3920 grids. For the convenience of MPI parallelism, the grids were united as blocks, that is, if $Proc_x \times Proc_y$ MPI processes are used in x and y directions, then each block has $B_x \times B_y$ grids, where $Proc_x \times B_x = 7200$ and $Proc_y \times B_y = 3920$. Each GPU process does 2-D or 3-D computation in these $B_x \times B_y$ grids as the MPI process does. In practice, four lateral columns are added to B_x and B_y (two on each side, $imt=B_x+4$, $jmt=B_y+4$) for halo. Table 2 lists the frequently used block definitions of LICOM3.

200   The original LICOM3 was written in F90. To adapt it to GPU, we applied the Fortran/C hybrid programming. As shown in Figure 2, the codes are kept using the F90 language before entering device-stepon and after stepon-out. The core computation processes within the stepons are rewritten by using HIP/C. Data structures in the CPU space remain the same as the original Fortran structures. The data commonly used by F90 and C are then defined by extra C, including files and defined by "extern" type pointers in C syntax to refer to them. In the GPU space, newly allocated GPU global memories hold the arrives

205   correspondence to those in the CPU space, and the HipMemcpy is called to copy them in and out.

Seven major subroutines (including their sub-recurrent calls) are converted from Fortran to HIP. The seven subroutine calls sequences are maintained, but each subroutine is deeply re-coded in the HIP to obtain the best performance. The CPU space data are 2-D or 3-D arrays; in the GPU space, we change them to 1-D arrays, improving the data transfer speed between different GPU subroutines.

210   The LICOM3-HIP is two-level parallelism, each MPI process corresponding to an ocean block. The computation within one MPI process is then pushed into GPU. The latency of data copy between GPU and CPU is one of the bottlenecks for daily computation loops. All read-only GPU variables are allocated and copied at the initial stage to reduce the data copy time. Some data copy is still needed in the stepping loop, e.g., MPI call in barotr.cpp.

The computation block in MPI (corresponding to 1 GPU) is a 3-D grid; in HIP revision, the 3-D parallelism is implemented.

215   This change adds extra parallel inside one block than the MPI solo parallelism (only 2-D). Some optimizations are needed to adapt to this change, such as increasing the global arrays to avoid data dependency. A demo of using a temporary array to parallel the computation inside a block can be found in Figure 4. Figure 4a represents a loop of the original code in the k direction. Since the variable $v(i,j,k)$ has a dependence on $v(i,j,k+1)$, it will cause an error when the GPU threads are paralleled in the k direction. We then separate the variable into two HIP kernel computations. In the upper of Figure 4b, a temporary

220   array vt is used to hold the result of $f1()$, and it can be GPU threads parallel in the k direction. Then, at the bottom of Figure 4b, we use vt to do the computations of $f2()$ and $f3()$; it can still be GPU threads parallel in the k direction. Finally, this loop of codes is parallelized.

The parallel in GPU is more like a shared-memory program; the memory write conflicts occur in the subroutine "tracer" advection computation. We change the if-else tree in this subroutine; hence, the data conflicts between neighboring grids are

225 avoided, making the 3-D parallelism successful. Moreover, in this subroutine, we use more operations to alternate the data movement to reduce the cache usage. Since the operation can be GPU threads parallelized and will not increase the total computation time, reducing the memory cache improves this subroutine's final performance.

A notable problem when the resolution is increased to 1/20° is that the total size of Fortran common blocks will be bigger beyond 2 GB. This change will not cause abnormal for C in the GPU space. But if the GPU process references the data, the

230 system call in HipMemcpy will occur compilation errors (It is perhaps due to the compiler limitation of the GPU compilation tool). We should change the original Fortran arrays' data structure from "static" to "allocatable" type in this situation. Since a GPU is limited to 16 GB GPU memory, the ocean block size in one block should not be too large. In practice, the 1/20° version starts from 384 GPUs (and it is regarded as the baseline for speedup here); if the partition is smaller than that value, sometimes the GPU memory insufficient errors will occur.

235 We found that the "tracer" is the most time-consuming subroutine for the CPU version (Figure 5). With the increase of CPU cores from 384 to 9216, the ratio of cost time for "tracer" is also increasing from 38% to 49%. "readyt" and "readyc" are computing-intensive subroutines. "tracer" is both computing-intensive and communication-intensive subroutine. "barotr" is a communication-intensive subroutine. The communication of "barotr" is 45 times more than that of "tracer" (Table 3). The computing-intensive subroutines can achieve good GPU speed, but the communication-intensive subroutine will achieve poor

240 performance.

We have done a set of experiments to measure the time cost of both halo update and memory copy in the HIP version (Figure 6). These two processes in the time integration are conducted in three subroutines: "barotr", "bclinc," and "tracer". The figure shows that "barotr" is the most time-consuming subroutine, and the memory copy dominates, which takes about 40% of the total time cost.

245 Variable operations in CPU and GPU memory are at least one magnitude faster than the data transfer between GPU and CPU through 16X PCI-e. The halo exchange in the MPI level is similar to POP have (Jiang et al. 2019). We did not change these codes in the HIP version. The four blue rows and columns in Figure.7 demonstrate the data that need to be exchanged with the neighbors. As shown in Figure 7, in GPU space, we pack the necessary lateral data for halo operation from $imt \times jmt$ to $4(imt+jmt)$. This change reduces the HipMemcpy data size to $(4/imt+4/jmt)$ of the original one. The larger imt and jmt are, the

250 less the transferred data is. At 384 GPUs, this change saves about 10% of the total computation time. The change is valuable for the HIP since the platform has no CUDA-aware MPI installed; otherwise, the halo operation can be done in the GPU space directly as POM.gpu does (Xu et al., 2015). The test indicates that the method can decrease by about 30% of the total wall clock time of "barotr" when 384 GPUs are used. But we have not optimized other kernels so far because their performance not as good as 384GPUs' when GPUs scale beyond 10000. We just keep it here as an option to improve the performance of

255 'barotr' at operational scales (i.e., GPU scales under 1536).

3.4. Model I/O optimization

Approximately 3GB forcing data are read from disk every model year, while about 60GB daily mean predicted variables are stored to disk every model day. The time cost for reading daily forcing data from the disk is increased to 200 s in one model day after the model resolution is updated from 1° to 1/20°. This time is equivalent to one step when 1536 GPUs are applied; hence, we must optimize it for total speedup. The cause of low performance is daily data reading and scattering to all nodes every model day; we then rewrite the data reading strategy and do parallel scattering for ten different forcing variables. Finally, the time cost of input is reduced to about 20 s, 1/10 of the original one (shown below).

As indicated, the core-process time cost is about 200s using 1536 GPUs. One model day's output needs about 250 s; it is also beyond the GPU computation time for one step. We modify the subroutine to a parallel version, and it decreases the data write time to 70 s on the test platform (this also depends on system I/O performance).

4 Model performance

4.1. Model performance in computing

Performing kilometer-scale and global climatic simulation are challenging (Palmer, 2014; Schär et al., 2020). As Fuhrer et al. (2018) pointed out, the SYPD is a useful metric to evaluate model performance for a parallel model (Balaji et al., 2017). Because a climate model often needs to be run at least 30-50 years for each simulation, 0.2-0.3 SYPD will require too much time to finish the experiment. The common view is that at least 1-2 SYPD is an adequate entrance for a realistic climate study. It also depends on the time scale in a climate study. For example, for 10-20-year simulation, 1-2 SYPD seems acceptable, and for 50-100 year simulation, 5-10 SYPD is better. NCEP weather prediction system's throughput standard is 8 minutes to finish one model day, equivalent to 0.5 SYPD.

Figure 8 illustrates the I/O performance of LICOM3-HIP, comparing the performances of computation processes. When the model applies 384 GPUs, the I/O costs 1/10 of the total simulation time (Figure 8a). While the scale increases to 9216 GPUs, the I/O time increases but is still smaller than the GPU's step time (Figure 8b). The improved LICOM3 I/O totally costs about 50-90 s (depends on scales), especially the input remains stable (Figure 8c) while scaling increases. This optimization of I/O maintains that the LICOM3-HIP 1/20° runs well at all practice scales for a realistic climate study. The I/O time has been cut off from the total simulation time in the follow-up test results to analyze the purely parallel performance.

Figure 9 shows the roof-line model using the Stream-GPU and LICOM program's measured behavioral data on a single computation node bound to one GPU card depicting the relationship between arithmetic intensity and performance floating points operations. The 100km resolution case is employed for the test. The blue and grey oblique line is the fitting line related to the Stream-GPU program's behavioral data using 5.12e8 and 1e6 threads, respectively, both with a blocksize of 256, which attain the best configuration. For details, the former is approximately the maximum threads number restricted by GPU card memory achieving the bandwidth limit to 696.52 GB/s. In comparison, the latter is close to the average number of threads in

GPU parallel calculations used by LICOM, reaching the bandwidth of 344.87 GB/s on average. Here we use the oblique grey line as a benchmark to verify the rationality of LICOM's performance, accomplishing the bandwidth of 313.95 GB/s averagely. Due to the large calculation scale of the whole LICOM program, the divided calculation grid bound to a single GPU card is limited by video memory; most kernel functions issue no more than 1.2e6 threads. As a result, the floating-point operations performance is a little far from the oblique roof-line shown in Figure 9. In particular, the subroutine bclinc apparently stray off the whole trend for including frequent 3d-array Halo MPI communications and a lot of data transmission between CPU and GPU.

Figure 10 shows the SYPD at various parallel scales. The baseline (384) of GPUs could achieve 42 times speedup than that of the same number of CPU cores. Sometimes, we also count the overall speedup, 384 GPUs in 96 nodes versus the total 3072 CPU cores in 96 nodes. We can get the overall performance speedup of 384 is about 6-7 times. The figure also indicates that for all scales, the SYPD keeps increasing. On the scale of 9216 GPUs, the SYPD first goes beyond 2, seven times the same CPUs result. A quasi-whole machine (26200 GPUs, 26200×65=1703000 cores totally, one process corresponds to one CPU core plus 64 GPU cores) result indicates it can still obtain an increasing SYPD to 2.72.

Since each node has 32 CPU cores and 4 GPUs, each GPU is managed by one CPU thread in the present cases. We can also quantify GPUs' speedup vs. all CPU cores' on the same number of nodes. For example, the 384 (768) GPUs correspond to 96 (192) nodes, which have 3072 (6144) CPU cores. Therefore, the overall speedup is about 6.375 (0.51/0.08) for 384 GPUs and 4.15 (0.83/0.2) for 768 GPUs (Figure 10). The speedups are comparable with our previous work porting LICOM2 to GPU using OpenACC (Jiang et al., 2019), which is about 1.8-4.6 times speedup using one GPU card vs. two 8-core Intel GPU in small-scale experiments for specific kernels. Our results are also slightly better than Xu et al. (2015), which has ported another ocean model to GPU using Cuda C. But due to the limitation of the number of intel CPUs (maximal 9216 cores), we didn't obtain the overall speedup for 1536 and more GPUs.

Figure 11 depicts the actual times and speedups of different GPUs computation. The green line in Figure 11a is the function of stepon time cost; it decreases while the GPU number increases. The blue curve of Figure 11a shows the increase of speedup with the rise of the GPU scale. Despite the speedup increase, the efficiency of the model decreases. At 9216 GPUs, the model efficiency starts under 20%; and for more GPUs (19600 and 26200), the efficiency is flattened to about 10%. The efficiency decreasing is mainly caused by the latency of data copy in and out to the GPU memory. For economical consideration, the 384-1536 scale is a better choice for realistic modeling studies.

Figure 12 depicts the time cost of seven core subroutines of LICOM3-HIP. We find that the top four most time cost subroutines are "barotr," "tracer," "bclinc," and "readyc"; and the other subroutines cost only about 1% of the whole computation time. When 384 GPUs are applied, the "barotr" costs about 50% of the total time (Figure 12a), which solves the barotropic equations. When GPUs are increased to 9216, each subroutine's time cost decreases, but the percentage of subroutine "barotr" is increased to 62% (Figure 12b). As mentioned above, the phenomenon can be interpreted by haloing in "barotr" being more than the other subroutines; hence, the memory data copy and communication latency make it slower.

4.2. Model performance in climate research

The daily mean sea surface height (SSH) fields of CPU and HIP versions' simulations are compared to test the usefulness of the HIP version of the LICOM for the numerical precision of scientific usage. Here, the results from 1/20° experiments on a particular day, March 1ˢᵗ of the 4th model year, are used (Figures 13a, b). The general SSH spatial patterns of the two are very similar visually. The significant differences are only found in very limited areas, such as in the eddy rich regions near strong currents or high-latitude regions (Figure 13c); in most places, the difference values fall into the range -0.1 and 0.1 cm. Because the hardware is different and the HIP codes' mathematical operation sequence is not always the same as that for the Fortran version, the HIP and CPU versions are not identical byte-by-byte. Therefore, it is hard to verify the correctness of the results from the HIP version. Usually, the ensemble method is employed to evaluate the consistency of two model runs (Baker et al., 2015). Considering the unacceptable computing and storage resources, besides the differences between the two versions, we here simply compute Root Mean Square Errors (RMSEs) between the two versions, which is only 0.18 cm, much smaller than the spatial variation of the system, which is 92 cm (about 0.2%). That indicates the results of LICO3-HIP are generally acceptable for research.

The GPU version's sea surface temperature (SST) is compared with the observed SST to evaluate the global 1/20° simulation's preliminary results from LICOM3-HIP (Figure 14). Because the LICOM3-HIP experiments are forced by the daily mean atmospheric variables in 2016, we also compare the outputs with the observation data of 2016. Here, the 1/4° Optimum Interpolation Sea Surface Temperature (OISST) is employed for comparison, and the simulated SST is interpolated to the same resolution as the OISST's. We find that the global mean values of SST are close to each other, but with a slight warming bias, 18.49°C for observations vs. 18.96°C for the model. The spatial pattern of SST in 2016 is well reproduced by LICOM3-HIP. The spatial Standard Deviation (STDs) of SST is 11.55°C for OISST and 10.98°C for LICOM3-HIP. The RMSE of LICOM3-HIP against the observation is only 0.84°C.

With an increasing horizontal resolution of the observation, we now know that mesoscale eddies are ubiquitous in the ocean with a 100-300 km spatial scale. The rigorous eddies usually occur along significant ocean currents, such as the Kuroshio and its Extension, the Gulf Stream, and the Antarctic Circumpolar Current (Figure 15a). The eddies also capture more than 80% of the ocean's kinetic energy, estimated using satellite data (e.g., Chelton et al., 2011). Therefore, these mesoscale eddies must be solved in the ocean model. A numerical model's horizontal resolution must be higher than 1/10° to resolve the global ocean's eddies, but that cannot resolve the eddies in the high latitude and shallow waters (Hallberg, 2013). Therefore, a higher resolution is required to determine the eddies globally. The EKE for the 1° version is low, even in the areas with strong currents, while the 1/10° version can reproduce most of the eddy-rich regions in the observation. The EKE is increased when the resolution is further enhanced to 1/20°, indicating much more eddy activities are resolved.

5 Discussion

5.1. Application of ocean climate model beyond 10000 GPUs

Table 4 summarizes detailed features of some published GPU version models. We can find that various programing methods have been implemented for different models. A near-kilometer atmospheric model using 4888 GPUs was reported as a large-scale example of weather/climate studies. With supercomputing development, the horizontal resolution of ocean circulation models will keep increasing, and more sophisticated physical processes will also be developed. The LICOM3-HIP has a larger scale, not only in terms of grid size but also in final GPU numbers.

We successfully performed a quasi-whole machine (26200 GPUs) test, and the results indicate the model obtained an increasing SYPD (2.72). The application of an ocean climate model beyond 10000 GPUs is not easy because the multi-nodes plus multi-GPUs running requires that the network connection, PCI-e and memory speed, and input/output storage systems all work in their best performances. Gupta (Gupta et al., 2017) have investigated the 23 types of system failures to improve HPC systems' reliability. Unlike Gupta's study, only three types of failures we have mostly met are discussed here. The three most occur errors when running LICOM3-HIP are MPI hardware errors, CPU memory access errors, and GPU hardware errors. Let's suppose that the probability of an individual hardware (or software) error to occur is 10^{-5} (which means 1 failure in 100000 hours). Along with the MPI (GPUs) scale increasing, the total error rate is increased; and once a hardware error occurs, the model simulation will fail.

When 384 GPUs are applied, the success rate within one hour can be expressed as $(1-384\times10^{-5})^3$=98.85%, and the failure rate is then $1-(1-384\times10^{-5})^3$=1.15%. Applying this formula, we can obtain the failure rate corresponding to 1000, 10000, and 26200 GPUs. The results are listed in Table 5. As shown in Table 5, in the medium-scale (i.e., 1000 GPUs are used), three failures will happen through 100 runs; when the scale increases to 10000 GPUs, 1/4 of them will fail. The 10^{-5} error probability also indicates that 10000 GPUs task cannot run ten continuous hours on average. If the success time restriction decreases, the model success rate will increase. For example, within 6 minutes, the 26200 GPUs task success rate is $(1-26200\times10^{-6})^3$=92.34%, and its failure rate is $1-(1-26200\times10^{-6})^3$=7.66%.

5.2. Energy to solution

We also measured energy to solution here. A simulation normalized energy (E) has been employed here as a metric. The formula is as follows:

$$E = TDP \times N \times 24/SYPD$$

where TDP is the Thermal Design Power, N is the computer nodes used, and SYPD/24 equals the simulated years per hour. So, the smaller the E value, the better, which means that we can get more simulated years within a limited power supply. To calculate E's value, we estimated the TDP of 1380 W for a node on the present platform (1 AMD CPU and 4 GPUs) and 290 W for a reference node (2 Intel 16-core CPUs). We only include the TDP of CPUs and GPUs here.

Based on the above power measurements, simulations' energy cost is shown in Table 6 in MWh per Simulation Year (MWh/SY). The energy costs for the 1/20° LICOM3 simulations running on CPUs and GPUs are comparable when the numbers of MPI processors are within 1000. The energy costs of LICOM3 at 1/20° running on 384 (768) GPUs and CPUs are about 6.234 (7.661) MWh/SY and 6.845 (6.280) MWh/SY, respectively. But the simulation speed of LICOM3 on GPU is much faster than that on CPU, about 42 times for 384 processors and 31 times for 768 processors. When the number of MPI processors is beyond 1000, the value of E for GPU becomes much larger than that for CPU. This result indicates the GPU is not fully loaded from this scale.

6 Conclusions

The GPU version of LICOM3 under the HIP framework has been developed in the present study. The dynamic core and physic packages are both ported to the GPU, and 3-D parallelization is applied. The new model has been implemented and gained an excellent accelerating rate on a Linux cluster with AMD GPU cards. This is also the first time an ocean general circulation model is fully applied on a heterogeneous supercomputer using the HIP framework. It totally took nineteen months, and five Ph.D. students and five part-time staff to finish the porting and testing work.

Based on our test using the 1/20° configuration, the LICOM3-HIP is 42 times fast than the CPU does, when 384 AMD GPUs and CPU cores are used. The LICOM3-HIP has good scalability, and it can still obtain a speedup of more than four on 9216 GPUs comparing to 384 GPUs. The SYPD, which is equilibrium to the speedup, keeps increasing as the number of GPUs increases. We successfully performed a quasi-whole machine test, which was 6550 nodes and 26200 GPUs, using 1/20° LICOM3-HIP on the supercomputer, and at the grand scale, the model can still obtain an increasing SYPD of 2.72. The modification or optimization of the model also improves 10 and 100 km performances, although we did not analyze their performances in this article.

The efficiency of the model decreases with the increasing number of GPUs. At 9216 GPUs, the model efficiency starts under 20% against 384 GPUs; and when the number of GPUs reaches or exceeds 20000, the efficiency is only about 10%. Based on our kernel functions test, the decreasing efficiency was mainly caused by the latency of data copy in and out to the GPU memory in solving the barotropic equations, particularly for the number of GPUs larger than 10000.

Using the 1/20° configuration of LICOM3-HIP, we have conducted a 14-year spin-up integration. Because the hardware is different and the GPU codes' mathematical operation sequence is not always the same as that of the Fortran version, the GPU and CPU versions cannot be identical byte by byte. The comparison between GPU and CPU versions of LICOM3 shows that the differences in most places are minimal, indicating that the results from LICOM3-HIP can be used for practical research. Further comparison with the observation and the lower-resolution results suggests that the 1/20° configuration of LICOM3-HIP can reproduce the observed large-scale features and produce much more smaller-scale activities than that of lower-resolution results.

The eddy-resolving ocean circulation model, which is a virtual platform for oceanography research, ocean forecast, and climate prediction and projection, can simulate the variations of the circulations, temperature, salinity, and sea level with a spatial scale larger than 15 km and temporal scale from diurnal cycle to decadal variability. As mentioned above, 1-2 SYPD is a good entrance for a realistic climate research model. The more practical GPU scale range for realistic simulation is around 384-1536 GPUs. At these scales, the model still has 0.5-1.22 SYPD. Even if we decrease the loops in "barotr" procedure to 1/3 of the original in the spin-up simulation, the performance will achieve 1-2.5 SYPD for 384-1536 GPUs. This performance will satisfy 10-50-year scale climate studies. Besides, this version can be used for short-term ocean prediction in the future.

Besides, the block size 36×30×55 (1/20° setup, 26200 GPUs) is not an enormous computation task for one GPU. Since one GPU has 64 cores total of 2560 threads, if a subroutine computation is 2-D, each thread's operation is too small. Even for the 3-D loops, it is still not big enough to load the whole GPU. This indicates that it will gain more speedup when the LICOM resolution is increased to the kilometer level. The LICOM3-HIP codes are now written for 1/20°, but they are kilometer-ready GPU codes.

The optimizing strategies here are mostly at the program level, not treat the dynamic or physics parts separately. We only ported all seven core subroutines within the time integration loops to GPU, including both the dynamic and physics parts. Unlike the atmospheric models, there are no many time-consuming physical processes in the ocean model, such as the radiative transportation, cloud, precipitation, and convection processes. Therefore, the two kinds of parts are usually not separated in the ocean model, particular in the early stage of model development. This is also the case of LICOM. Further optimization to explicitly separate the dynamic core and the physical package should be done in the future.

There is still potential to further increase the speedup of LICOM3-HIP. The bottleneck is in the high-frequency data copy in and out to the GPU memory in the barotropic part of the LICOM3. Unless the HIP-aware MPI is supported, data transfer latency between CPU and GPU cannot be overcome. So far, we can only reduce the time consumed by decreasing the frequency or magnitude of the data copy and even modifying the method to solve the barotropic equations. Additionally, using the single-precision within the time integration of LICOM3 might be another solution. The mixing precision method has already been tested using an atmospheric model, and an average gain in computational efficiency by approximately 40% (Váňa et al., 2017). We would like to try these methods in the future.

Code availability

The model code (LICOM3-HIP V1.0) along with the dataset and a 100km case can be downloaded from the website https://zenodo.org/record/4302813#.X8mGWcsvNb8 with the Digital Object Identifier (doi): 10.5281/zenodo.4302813.

Data availability

The data for figures in this paper can be downloaded from https://zenodo.org/record/4542544#.YCs24c8vPII with doi: 10.5281/zenodo.4542544.

Author contribution

Pengfei Wang: Software, Visualization, Formal analysis, and writing-original draft

Jinrong Jiang: Software and Writing – review & editing

Pengfei Lin: Software and Writing – review & editing

Mengrong Ding: Visualization and Data curation

Junlin Wei: Software

Zhang Feng: Software

Lian Zhao: Software

Yiwen Li: Software and Visualization

Zipeng Yu: Software and Data curation

Weipeng Zheng: Formal analysis

Yongqiang Yu: Conceptualization

Xuebin Chi: Conceptualization

Hailong Liu: Supervision, Formal analysis, and writing-original draft

Competing interests:

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The study is funded by National Natural Sciences Foundation (41931183), the National Key Research and Development Program (2018YFA0605904 and 2018YFA0605703), and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDC01040100). Dr.s H.L.L. and P.F.L. were also supported by the "Earth System Science Numerical Simulator Facility" (EarthLab).

References

Baker, A. H., Hammerling, D. M., Levy, M. N., Xu, H., Dennis, J. M., Eaton, B. E., Edwards, J., Hannay, C., Mickelson, S. A., Neale, R. B., Nychka, D., Shollenberger, J., Tribbia, J., Vertenstein, M., and Williamson, D.: A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1. 0), *Geosci. Model Dev.*, 8, 2829-2840, https://doi.org/10.5194/gmd-8-2829-2015, 2015.

Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U., Aloisio, G., Benson, R., Caubel, A., Durachta, J., Foujols, M.-A., Lister, G., Mocavero, S., Underwood, S., and Wright, G.: CPMIP: measurements of real computational performance of Earth system models in CMIP6, *Geosci. Model Dev.*, 10, 19–34, https://doi.org/10.5194/gmd-10-19-2017, 2017.

Canuto, V., Howard, A., Cheng, Y., and Dubovikov, M.: Ocean turbulence. Part I: One-point closure model—Momentum and heat vertical diffusivities, *J. Phys. Oceanogr.*, 31, 1413-1426, https://doi.org/10.1175/1520-0485(2001)031<1413:OTPIOP>2.0.CO;2, 2001.

Canuto, V., Howard, A., Cheng, Y., and Dubovikov M.: Ocean turbulence, Part II: Vertical diffusivities of momentum, heat, salt, mass, and passive scalars, *J. Phys. Oceanogr.*, 32(1), 240-264, https://doi.org/10.1175/1520-0485(2002)032<0240:OTPIVD>2.0.CO;2, 2002.

Chassignet, E. P., Sommer, J. L., and Wallcraft A. J: General Circulation Models In "Encyclopedia of Ocean Sciences (3rd edition)", Cochran, K. J., Bokuniewicz, H. J., and Yager P. L. (Eds.), 5, 486-490, https://doi.org/10.1016/B978-0-12-409548-9.11410-1, 2019.

Chassignet, E. P., Yeager, S. G., Fox-Kemper, B., Bozec, A., Castruccio, F., Danabasoglu, G., Kim, W. M., Koldunov, N., Li, Y., Lin, P., Liu, H., Sein, D. V., Sidorenko, D., Wang, Q., and Xu, X.: Impact of horizontal resolution on global ocean-sea-ice model simulations based on the experimental protocols of the Ocean Model Intercomparison Project phase 2 (OMIP-2), *Geosci. Model Dev.*, 1-58, https://doi.org/10.5194/gmd-2019-374, 2020.

Chelton, D. B., Schlax, M. G., & Samelson R. M.: Global observations of nonlinear mesoscale eddies, *Prog. Oceanogr.*, 91(2), 167-216, https://doi.org/10.1016/j.pocean.2011.01.002, 2011.

Craig, A. P., Vertenstein, M. and Jacob, R.: A new flexible coupler for earth system modeling developed for CCSM4 and CESM1, *The International Journal of High Performance Computing Applications*, 26(1), 31-42, https://doi.org/10.1177/1094342011428141, 2012.

Ferreira, D., Marshall, J. and Heimbach, P.: Estimating eddy stresses by fitting dynamics to observations using a residual-mean ocean circulation model and its adjoint, *J. Phys. Oceanogr.*, 35(10), 1891-1910, https://doi.org/10.1175/JPO2785.1, 2005.

Fox-Kemper, B., and Menemenlis D.: Can Large Eddy Simulation Techniques Improve Mesoscale-Rich Ocean Models? In: Ocean Modeling in an Eddying Regime, edited by: Hecht, M.W., and Hasumi H., 177, 319-338, https://doi.org/ 10.1029/177GM19, 2008.

Fuhrer, O., Chadha, T., Hoefler, T., Kwasniewski, G., Lapillonne, X., Leutwyler, D., Lüthi, D., Osuna, C., C., Schär, C.,
500 Schultess, T. C., and Vogt, H., Near-global climate simulation at 1 km resolution: establishing a performance baseline on
4888 GPUs with COSMO 5.0, *Geosci. Model Dev.*, 11(4), 1665-1681, https://doi.org/10.5194/gmd-11-1665-2018, 2018.

Gent, P. R., and J. C. Mcwilliams: Isopycnal mixing in ocean circulation models, *J. Phys. Oceanogr.*, 20(1), 150-155,
https://doi.org/10.1175/1520-0485(1990)020<0150:IMIOCM>2.0.CO;2, 1990.

Griffies, S. M., Winton, M., Anderson, W. G., Benson, R., Delworth, T. L., Dufour, C. O., Dunne, J. P., Goddard, P., Morrison,
505 A. K., Rosati, A., Wittenberg, A. T., Yin, J., and Zhang, R.: Impacts on ocean heat from transient mesoscale eddies in a
hierarchy of climate models, *J. Climate*, 28(3), 952-977, https://doi.org/10.1175/JCLI-D-14-00353.1, 2015.

Griffies, S., Danabasoglu, G., Durack, P., Adcroft, A., Balaji, V., Böning, C., Chassignet, E., Curchitser, E., Deshayes, J.,
Drange, H., Fox-Kemper, B., Gleckler, P. J., Gregory, J. M., Haak, H., Hallberg, R. W., Heimbach, P., Hewitt, H. T., Holland,
D. M., Ilyina, T., Jungclaus, J. H., Komuro, Y., Krasting, J. P., Large, W. G., Marsland, S. J., Masina, S., McDougall, T. J.,
510 Nurser, A. J. G., Orr, J. C., Pirani, A., Qiao, F., Stouffer, R. J., Taylor, K. E., Treguier, A. M., Tsujino, H., Uotila, P., Valdivieso,
M., Wang, Q., Winton, M., and Yeager, S. G.: OMIP contribution to CMIP6: experimental and diagnostic protocol for the
physical component of the Ocean Model Intercomparison Project, *Geosci. Model Dev.*, 9(9), https://dx.doi.org/10.5194/gmd-
9-3231-2016, 2016.

Gupta, S., Patel, T., Engelmann, C., and Tiwari, D.: Failures in Large Scale Systems: Long-term Measurement, Analysis, and
515 Implications. In *Proceedings of SC17*, Denver, CO, USA, November 12-17, https://dx.doi.org/10.1145/3126908.3126937,
2017.

Hallberg, R.: Using a resolution function to regulate parameterizations of oceanic mesoscale eddy effects, *Ocean Model.*, 72,
92-103, https://dx.doi.org/10.1016/j.ocemod.2013.08.007, 2013.

He, B., Yu, Y., Bao, Q., Lin, P. F., Liu, H. L., Li, J. X., Wang, L., Liu, Y. M., Wu, G., Chen, K., Guo, Y. Zhao, S., Zhang, X.,
520 Song, M., and Xie, J.: CAS FGOALS-f3-L model dataset descriptions for CMIP6 DECK experiments, *Atmos. Oceanic Sci.
Lett.*, 1-7, https://dx.doi.org/10.10801/16742834.2020.1778419, 2020.

Hewitt, H. T., Bell, M. J., Chassignet, E. P., Czaja, A., Ferreira, D., Griffies, S. M., Hyder, P., McClean, J. L., New, A.L., and
Roberts, H. J.: Will high-resolution global ocean models benefit coupled predictions on short-range to climate timescales?
Ocean Model., 120, 120-136, https://dx.doi.org/10.1016/j.ocemod.2017.11.002, 2017.

525 Jiang, J., Lin, P., Wang, J., Liu, H., Chi, X., Hao, H., Wang, Y., Wang, W., and Zhang, L. H.: Porting LASG/IAP Climate
System Ocean Model to GPUs Using OpenAcc, *IEEE Access*, 7, 154490-154501,
https://dx.doi.org/10.1109/ACCESS.2019.2932443, 2019.

Large, W. G., and Yeager, S. G.: The global climatology of an interannually varying air-sea flux data set, *Clim. Dyn.*, 33,
341−364, https://doi.org/10.1007/s00382-008-0441-3, 2009.

530 Li, L., Yu, Y., Tang, Y., Lin, P., Xie, J., Song, M., Dong, L., Zhou, T., Liu, L., Wang, L., Pu, Y., Chen, X. L., Chen, L., Xie,
Z. H., Liu, H. B., Zhang, L. X., Huang, X., Feng, T., Zheng, W. P., Xia, K., Liu, H. L., Liu, J. P., Wang, Y., Wang, L. H., Jia,
B. H., Xie, F., Wang, B., Zhao, S. W., Yu, Z. P., Zhao, B. W., and Wei, J. L.: The Flexible Global Ocean Atmosphere Land

System Model Grid Point Version 3 (FGOALS-g3): Description and Evaluation, *J. Adv. Model Earth Sy.*, https://dx.doi.org/10.1029/2019MS002012, 2020.

535 Li, Y. W., Liu, H. L., Ding, M. R., Lin, P. F., Yu, Z. P., Meng, Y., Li, Y. L., Jian, X. D., Jiang, J. R., Chen, K. J., Yang, Q., Wang, Y. Q., Zhao, B. W., Wei, J. L., Ma, J. F., Zheng, W. P., and Wang, P. F.: Eddy-resolving Simulation of CAS-LICOM3 for Phase 2 of the Ocean Model Intercomparison Project, *Adv. Atmos. Sci.*, 37(10), 1067-1080, https://doi.org/ 10.1007/s00376-020-0057-z., 2020.

Lin, P., Liu, H., Xue, W., Li, H., Jiang, J., Song, M., Song, Y., Wang, F., and Zhang, M. H.: A Coupled Experiment with

540 LICOM2 as the Ocean Component of CESM1, *J. Meteorol. Res.*, 30(1), 76-92, https://dx.doi.org/10.1007/s13351-015-5045-3, 2016.

Lin, P., Yu, Z., Liu, H., Yu, Y., Li, Y., Jiang, J., Xue, W., Chen, K., Yang, Q., Zhao, B. W., Wei, J. L., Ding, M. R., Sun, Z. K., Wang, Y. Q., Meng, Y., Zheng, W. P., and Ma, J. F.: LICOM Model Datasets for the CMIP6 Ocean Model Intercomparison Project, *Adv. Atmos. Sci.*, 37(3), 239-249, https://dx.doi.org/10.1007/s00376-019-9208-5, 2020.

545 Liu, H., Lin, P., Yu, Y., and Zhang X.: The baseline evaluation of LASG/IAP climate system ocean model (LICOM) version 2, *Acta Meteorol. Sin.*, 26(3), 318-329, https://doi.org/10.1007/s13351-012-0305-y, 2012.

Liu, H. L., Lin, P., Zheng, W., Luan, Y., Ma, J., Mo, H., Wan L., and Tiejun Ling: A global eddy-resolving ocean forecast system – LICOM Forecast System (LFS), *J. Oper. Oceanogr.*, under review, 2020.

Madec, G., and Imbard, M.: A global ocean mesh to overcome the North Pole singularity, *Clim. Dyn.*, 12(6), 381-388,

550 https://doi.org/10.1007/s003820050115, 1996.

Murray, R. J.: Explicit generation of orthogonal grids for ocean models, *J. Comput. Phys.*, 126(2), 251-273, https://doi.org/10.1006/jcph.1996.0136, 1996.

Ohlmann, J. C.: Ocean radiant heating in climate models, *J. Climate*, 16(9), 1337-1351, https://doi.org/10.1175/1520-0442-16.9.1337, 2003

555 Palmer, T.: Climate forecasting: Build high-resolution global climate models, *Nature News*, 515(7527), 338-339, https://doi.org/10.1038/515338a, 2014.

Redi, M. H.: Oceanic isopycnal mixing by coordinate rotation, *J. Phys. Oceanogr.*, 12(10), 1154-1158, https://doi.org/ 10.1175/1520-0485(1982)012<1154:OIMBCR>2.0.CO;2, 1982.

Schär, C., Fuhrer, O., Arteaga, A., Ban, N., Charpilloz, C., Di Girolamo, S., Hentgen, L., Hoefler, T., Lapillonne, X.,

560 Leutwyler, D., Osterried, K., Panosetti, D., Rüdisühli, S., Schlemmer, L., Schulthess, T. C., Sprenger, M., Ubbiali, S., and Wernli, H.: Kilometer-scale climate models: Prospects and challenges, *Bull. Am. Meteorol. Soc.*, 101(5), E567-E587, https://doi.org/ 10.1175/BAMS-D-18-0167.1, 2020.

St. Laurent, L., Simmons, H., and Jayne, S.: Estimating tidally driven mixing in the deep ocean, *Geophys. Res. Lett.*, 29(23), https://doi.org/ 10.1029/2002GL015633, 2002.

565 Su, Z., Wang, J., Klein, P., Thompson, A. F., and Menemenlis, D.: Ocean submesoscales as a key component of the global heat budget, *Nature communications*, 9(1), 1-8, https://doi.org/ 10.1038/s41467-018-02983-w, 2018.

Tsujino, H., Urakawa, S., Nakano, H., Small, R., Kim, W., Yeager, S., Danabasoglu, G., Suzuki, T., Bamber, J. L., Bentsen, M., Böning, C. W., Bozec, A., Chassignet, E. P., Curchitser, E., Dias, F. B., Durack, P. J., Griffies, S. M., Harada, Y., Ilicak, M., Josey, S. A., Kobayashi, C., Kobayashi, S., Komuro, Y., Large, W. G., Le Sommer, J., Marsland, S. J., Masina, S., Scheinert, M., Tomita, H., Valdivieso, M., Yamazaki, D.: JRA-55 based surface dataset for driving ocean - sea-ice models (JRA55-do), *Ocean Model.*, 130, 79-139, https://dx.doi.org/10.1016/j.ocemod.2018.07.002, 2018.

Tsujino, H., Urakawa, L. S., Griffies, S. M., Danabasoglu, G., Adcroft, A. J., Amaral, A. E., Arsouze, T., Bentsen, M., Bernardello, R., Böning, C., Bozec, A., Chassignet, E., Danilov, S., Dussin, R., Exarchou, E., Fogli, P., Fox-Kemper, B., Guo, C., Ilicak, M., Iovino, D., Kim, W., Koldunov, N., Lapin, V., Li, Y. W., Lin, P. F., Lindsay, K., Liu, H. L., Long, M., Komuro, Y., Marsland, S., Masina, S., Nummelin, A., Rieck, J., Ruprich-Robert, Y., Scheinert, M., Sicardi, V., Sidorenko, D., Suzuki, T., Tatebe, H., Wang, Q., Yeager, S., and Yu, Z. P.: Evaluation of global ocean–sea-ice model simulations based on the experimental protocols of the Ocean Model Intercomparison Project phase 2 (OMIP-2), *Geosci. Model Dev.*, 13(8), 3643-3708, https://dx.doi.org/10.5194/gmd-13-3643-2020, 2020.

Váňa, F., Düben, P., Lang, S., Palmer, T., Leutbecher, M., Salmond, D., and Carver, G.: Single Precision in Weather Forecasting Models: An Evaluation with the IFS, *Mon. Weather Rev.*, 145(2), 495-502, https://dx.doi.org/10.1175/mwr-d-16-0228.1, 2017.

Wang, S., Jing, Z., Zhang, Q., Chang, P., Chen, Z., Liu, H. and Wu L.: Ocean Eddy Energetics in the Spectral Space as Revealed by High-Resolution General Circulation Models, *J. Phys. Oceanogr.*, 49(11), 2815-2827, https://doi.org/10.1175/JPO-D-19-0034.1, 2019.

Xiao, C.: Adoption of a two-step shape-preserving advection scheme in an OGCM and its coupled experiment, Master thesis, Institute of Atmospheric Physics, Chinese Academy of Sciences, 78pp, 2006.

Xu, S., Huang, X., Oey, L. Y., Xu, F., Fu, H., Zhang, Y., and Yang, G.: POM. gpu-v1. 0: a GPU-based Princeton Ocean Model, *Geosci. Model Dev.*, 8(9), 2815-2827, https://doi.org/10.5194/gmd-8-2815-2015, 2015.

Yang, C., Xue, W., Fu, H., You, H., Wang, X., Ao, Y., Liu, F., Gan, L., Xiu, P., and Wang, L.: 10M-core scalable fully-implicit solver for nonhydrostatic atmospheric dynamics. Paper presented at SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2016.

Yashiro, H., Terai, M., Yoshida, R., Iga, S., Minami, K., and Tomita, H.: Performance analysis and optimization of nonhydrostatic icosahedral atmospheric model (NICAM) on the K computer and TSUBAME2, 5, Paper presented at Proceedings of the Platform for Advanced Scientific Computing Conference, 2016.

Yu, Y., Tang, S., Liu, H., Lin, P., and Li, X.: Development and evaluation of the dynamic framework of an ocean general circulation model with arbitrary orthogonal curvilinear coordinate, *Chinese J. Atmospheric Sci.*, 42(4), 877-889, https://doi.org/10.1006-9895(2018)42:4<877:RYZJQX>2.0.TX;2-B, 2018.

Yuan, Y., Shi, F., Kirby, J. T., and Yu, F.: FUNWAVE-GPU: Multiple-GPU Acceleration of a Boussinesq-Type Wave Model, *J. Adv. Model Earth Sy.*, 12(5), https://doi.org/ 10.1029/2019MS001957, 2020.

600   Zhang, X., and Liang, X.: A numerical world ocean general circulation model, *Adv. Atmos. Sci.*, 6(1), 44-61, https://doi.org/ 0256-1530(1989)6:1<44:ANWOGC>2.0.TX;2-S, 1989.

Zhang, H., M., Jin, J., Fei, K., Ji, D., Wu, C., Zhu, J., He, J., Chai, Z. Y., Xie, J. B., Dong, X., Zhang, D. L., Bi, X. Q., Cao, H., Chen, H. S., Chen, K. J., Chen, X. S., Gao, X., Hao, H. Q., Jiang, J. R., Kong, X. H., Li, S. G., Li, Y. C., Lin, P. F., Lin, Z. H., Liu, H. L., Liu, X. H., Shi, Y., Song, M. R., Wang, H. J., Wang, T. Y., Wang, X. C., Wang, Z. F., Wei, Y., Wu, B. D., Xie,

605   Z. H., Xu, Y. F., Yu, Y. Q., Yuan, L., Zeng, Q. C., Zeng, X. D., Zhao, S. W., Zhou, G. Q., and Zhu, J.: CAS-ESM2: Description and Climate Simulation Performance of the Chinese Academy of Sciences (CAS) Earth System Model (ESM) Version 2, *J. Adv. Model Earth Sy.*, 12, e2020MS002210. https://doi.org/10.1029/2020MS002210, 2020.

Shaoqing Zhang, Haohuan Fu, LixinWu, Yuxuan Li, Hong Wang, Yunhui Zeng, Xiaohui Duan,WubingWan, Li Wang, Yuan Zhuang, Hongsong Meng, Kai Xu, Ping Xu, Lin Gan, Zhao Liu,SihaiWu, Yuhu Chen, Haining Yu, Shupeng Shi, Lanning

610   Wang, Shiming Xu, Wei Xue, Weiguo Liu,Qiang Guo, Jie Zhang, Guanghui Zhu, Yang Tu, Jim Edwards, Allison Baker, Jianlin Yong, Man Yuan,Yangyang Yu, Qiuying Zhang, Zedong Liu, Mingkui Li, Dongning Jia, Guangwen Yang, Zhiqiang Wei, Jingshan Pan, Ping Chang, Gokhan Danabasoglu, Stephen Yeager, Nan Rosenbloom, and Ying Guo.: Optimizing high-resolution Community Earth System Model on a heterogeneous many-core supercomputing platform, *Geosci. Model Dev.*, 13, 4809-4829, https://doi.org/10.5194/gmd-13-4809-2020, 2020.

615

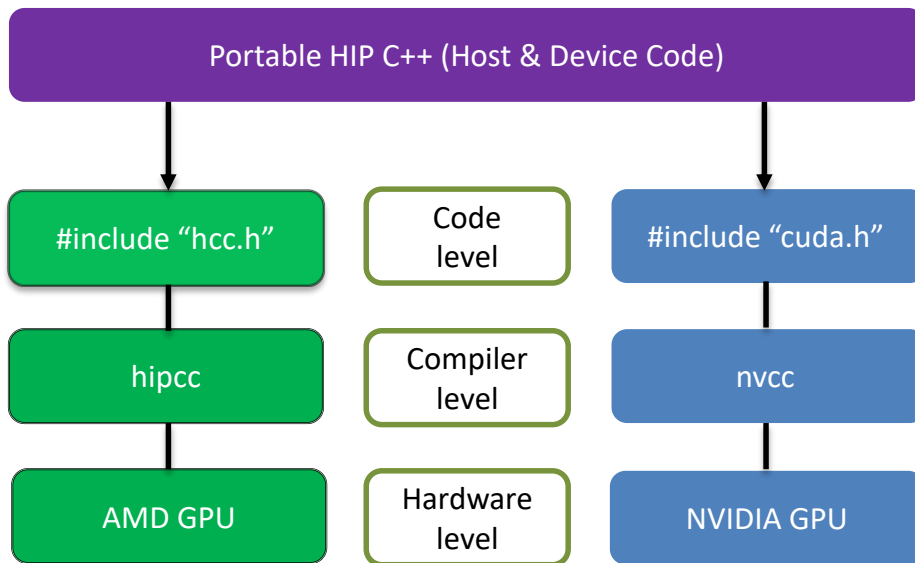



Figure 1: The schematic diagram of the comparison of coding on AMD and NVIDIA GPU in three levels.

620

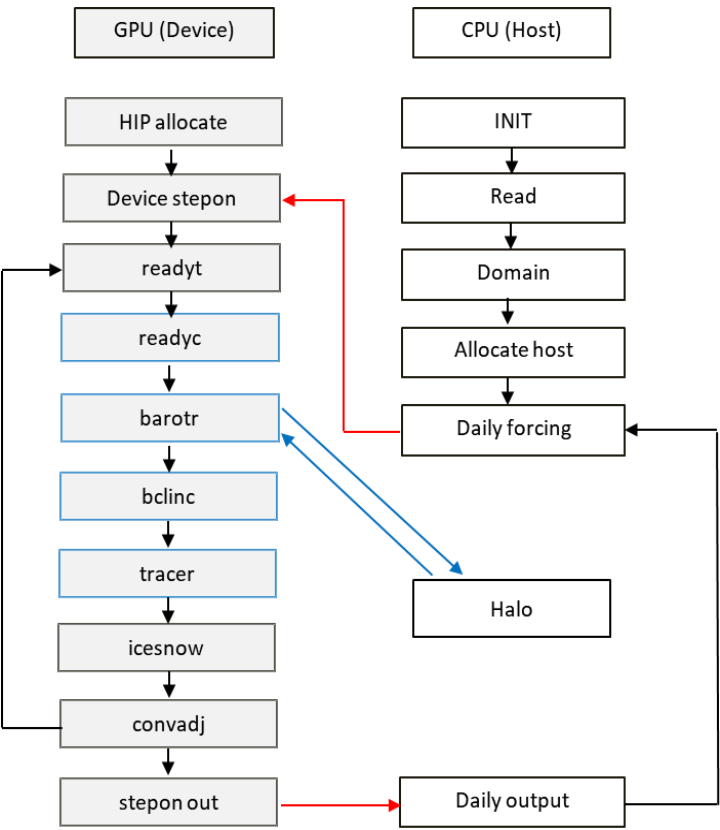
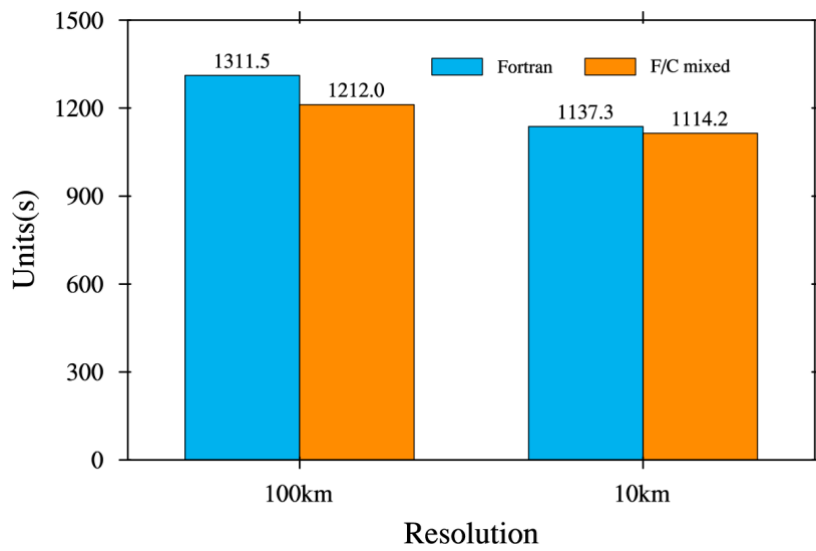



Figure 2: LICOM3 computation flowchart with GPU (HIP device). The red line indicates whole block data transfer between host and GPU, while the blue line means only transferring lateral data of a block.

625   **Figure 3: The wall clock time of a model day for the 10km version and a model month for the 100km version. The blue and orange bars are for the Fortran and the Fortran and C mixed version. These tests were conducted on a platform of Intel Xeon CPU (E5-2697A v4, 2.60GHz). We used 28 and 280 cores for the low and high resolution, respectively.**

$do\,k = 1, km$
$v(i, j, k) = f1(u(i, j, k),...)$
$v(i, j, k) = f2(v(i, j, k), v(i, j, k+1),...)$
$v(i, j, k) = f3(v(i, j, k),...)$
$end\,do$

$do\,k = 1, km$
$vt(i, j, k) = f1(u(i, j, k),...)$
$end\,do$

$do\,k = 1, km$
$v(i, j, k) = f2(vt(i, j, k), vt(i, j, k+1),...)$
$v(i, j, k) = f3(v(i, j, k),...)$
$end\,do$

(a) Original code   (b) Optimized code

630

Figure 4: The code using temporary arrays to avoid data dependency.

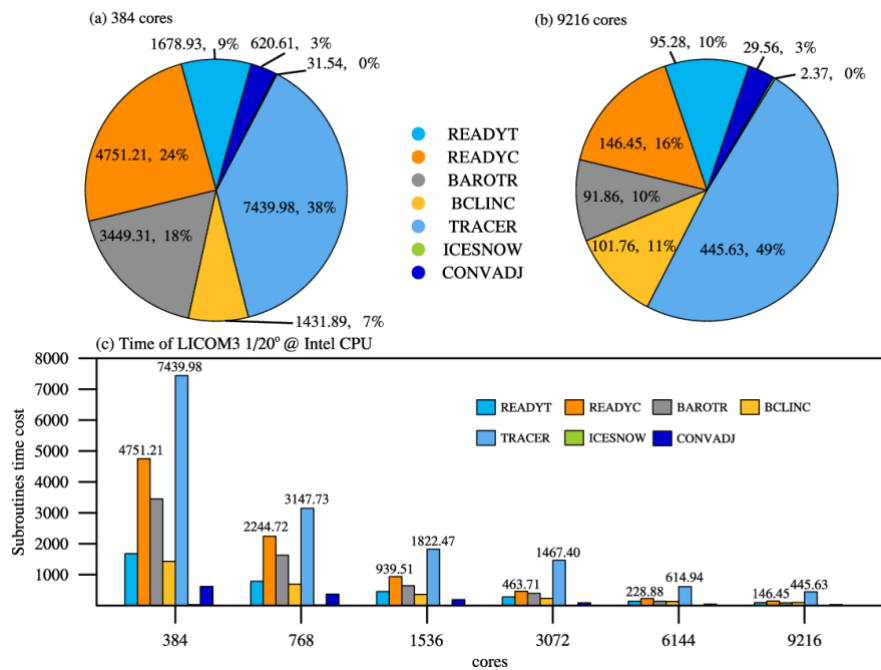



Figure 5: The seven core subroutines' time cost percentage for (a) 384 and (b) 9216 CPU cores. (c) the subroutines' time cost at different scales of LICOM3 (1/20°). These tests were conducted on a platform of Intel Xeon CPU (E5-2697A v4, 2.60GHz).

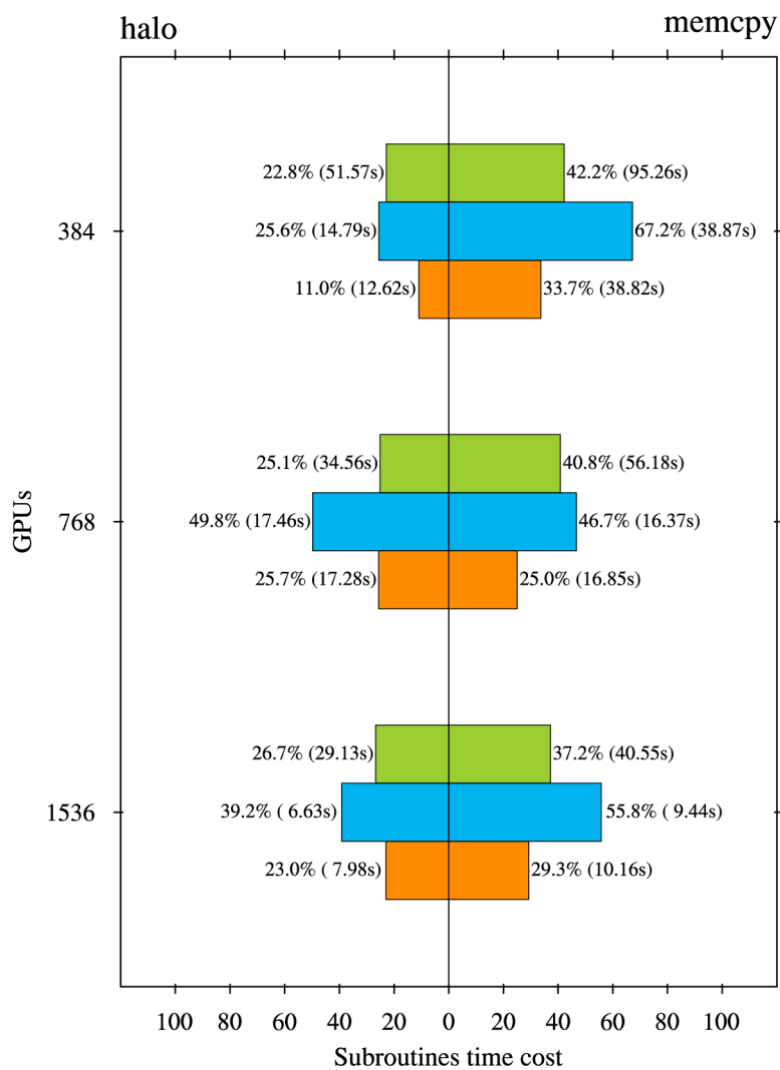



Figure 6: The ratio of the time cost of halo update and memory copy to the total time cost for three subroutines, "barotr" (green), "bclinc" (blue), and "tracer" (orange) in the HIP version LICOM for three scales (Unit: %). The numbers in the blankets are the time cost of the two processes (Unit: second).

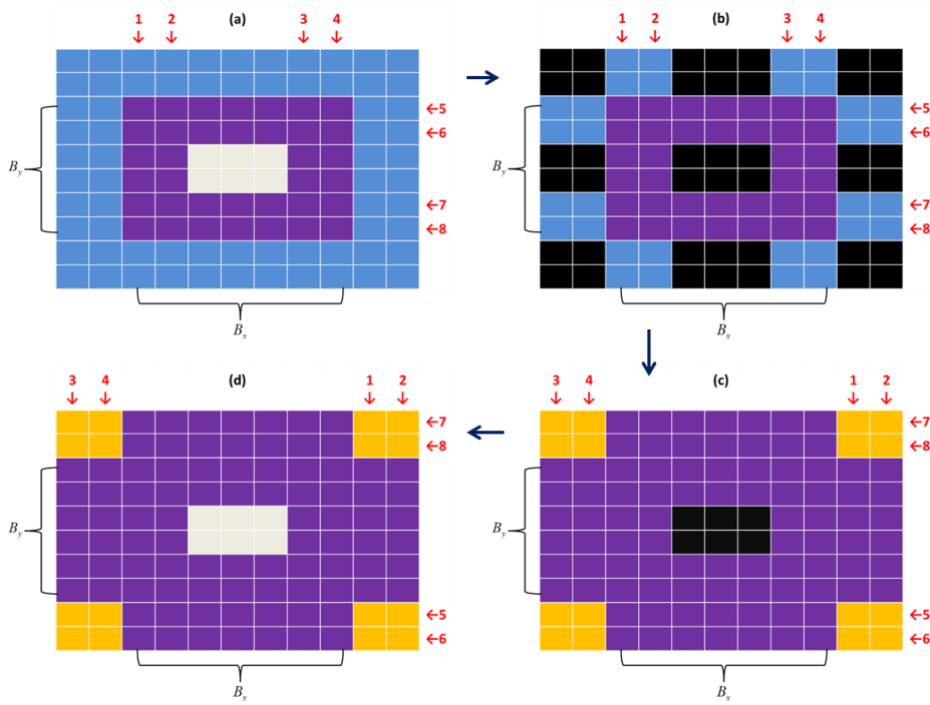



Figure 7: The lateral packing (only transfer four rows and four columns data between GPU and CPU) method to accelerate halo. (a) In the GPU space, where central (gray) grids are unchanged; (b) transferred to the CPU space, where black grids mean no data; (c) after halo with neighbors; and (d) transfer back to the GPU space.

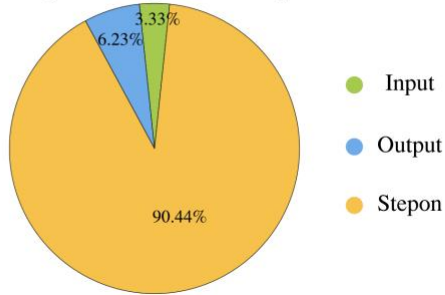
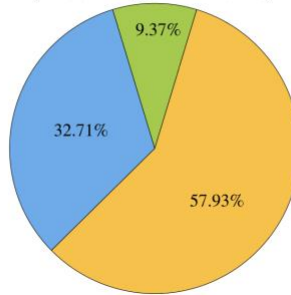
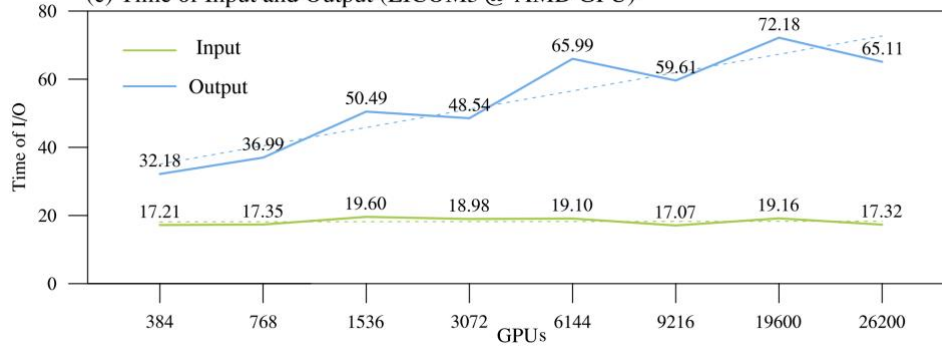



Figure 8: (a) The 384 GPUs, (b) 9216 GPUs, the I/O ratio in total simulation time for 1/20° setup, (c) the changes of I/O times versus different GPUs.

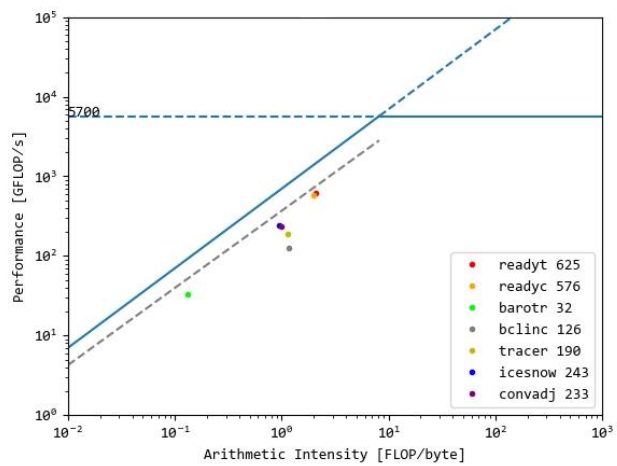



Figure 9: Roofline model for AMD GPU and the performance of LICOM's main subroutines.

655

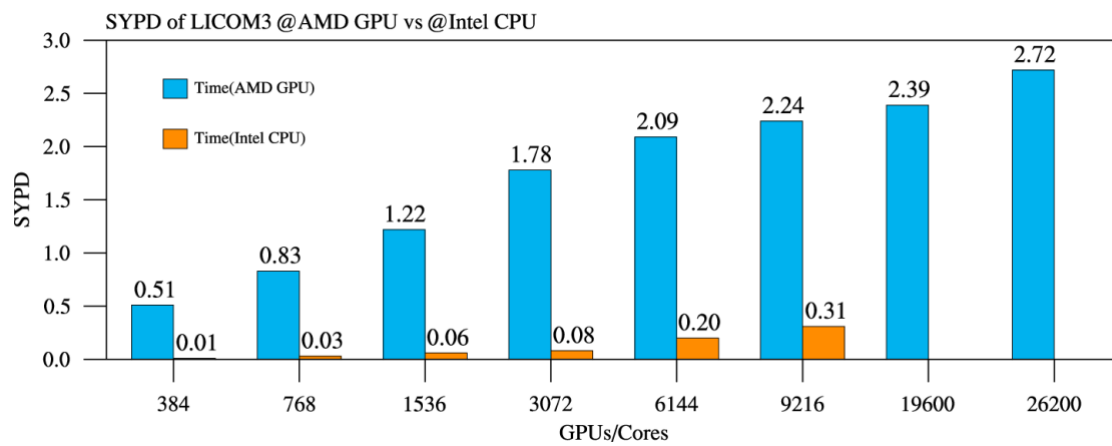
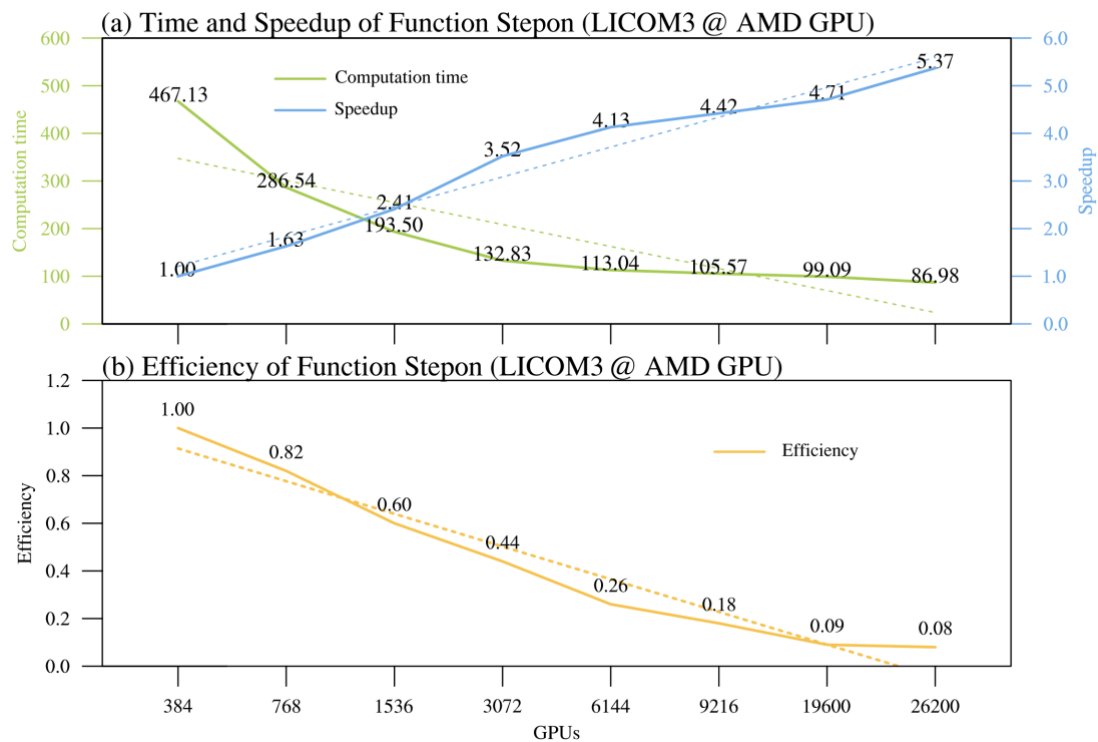



Figure 10: Simulation performances of AMD GPU versus Intel CPU core for LICOM3 (1/20°). Unit: SYPD.

660   **Figure 11: (a) Computation time (green) and speedup (blue), and (b) parallel efficiency (orange) at different scales for stepon of LICOM3-HIP (1/20°).**

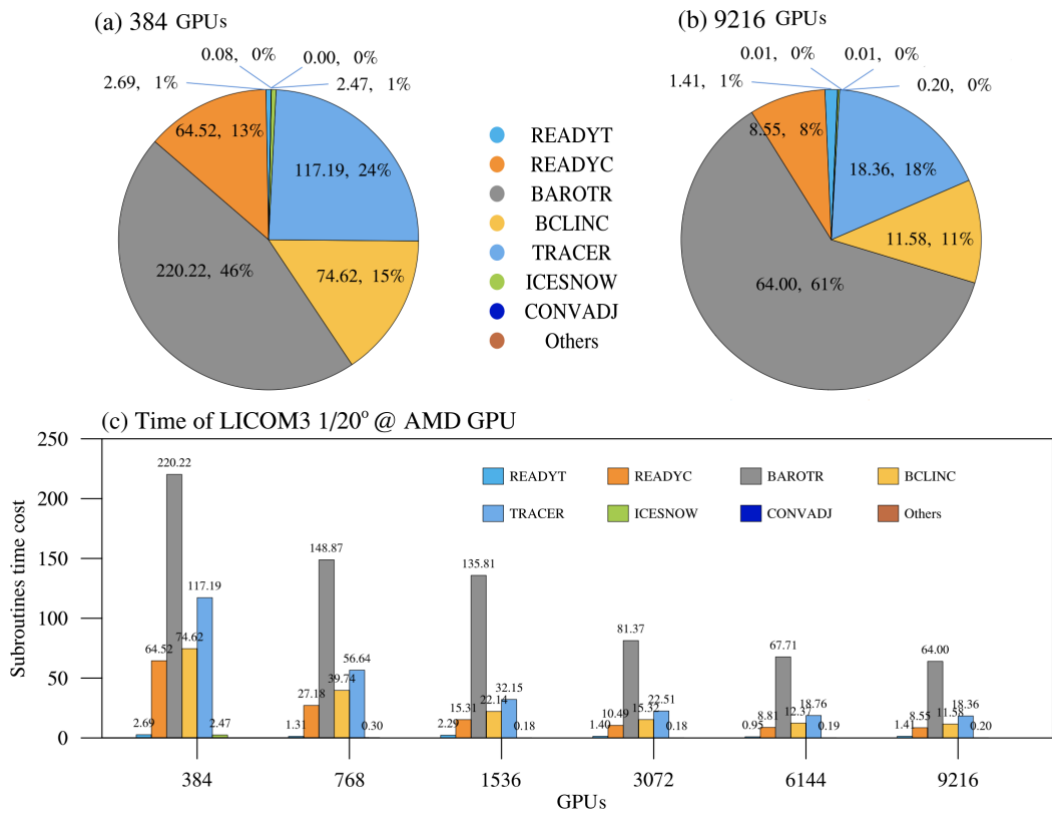



Figure 12: The seven core subroutines' time cost percentage for (a) The 384 GPUs and (b) 9216 GPUs. (c) the subroutines' time cost at different scales of LICOM3-HIP (1/20°).

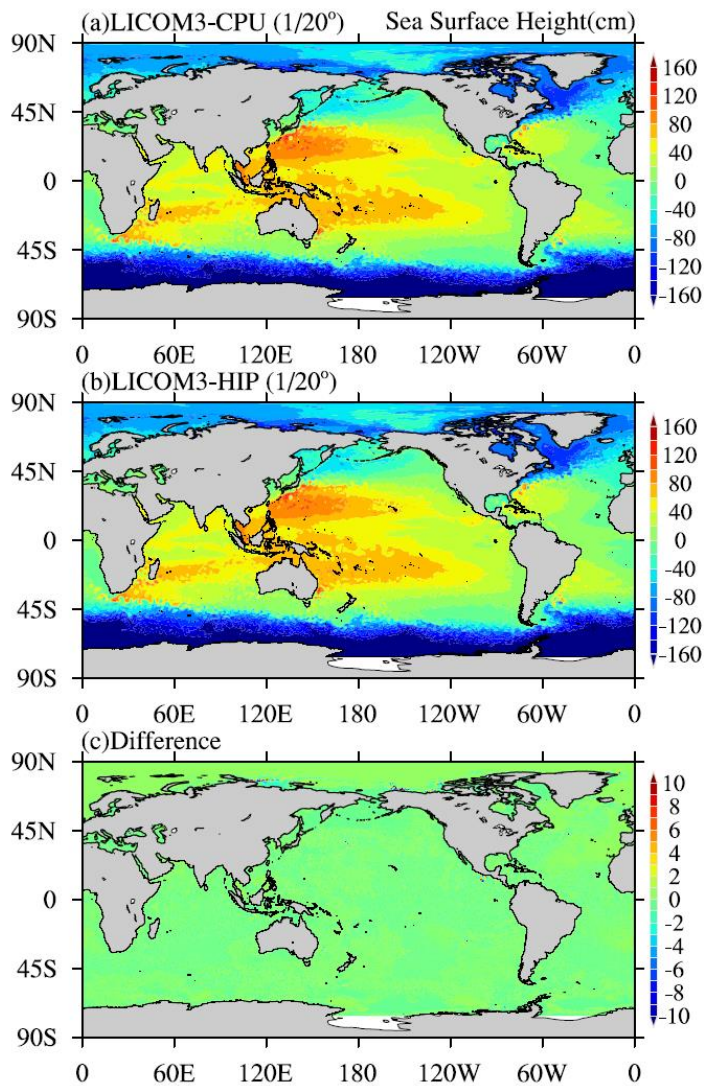



Figure 13: Daily mean simulated sea surface height for (a) CPU and (b) HIP versions of LICOM3 at 1/20° on March 1st of the 4th model year. (c) The difference between the two versions (HIP minus CPU). Units: cm.

670

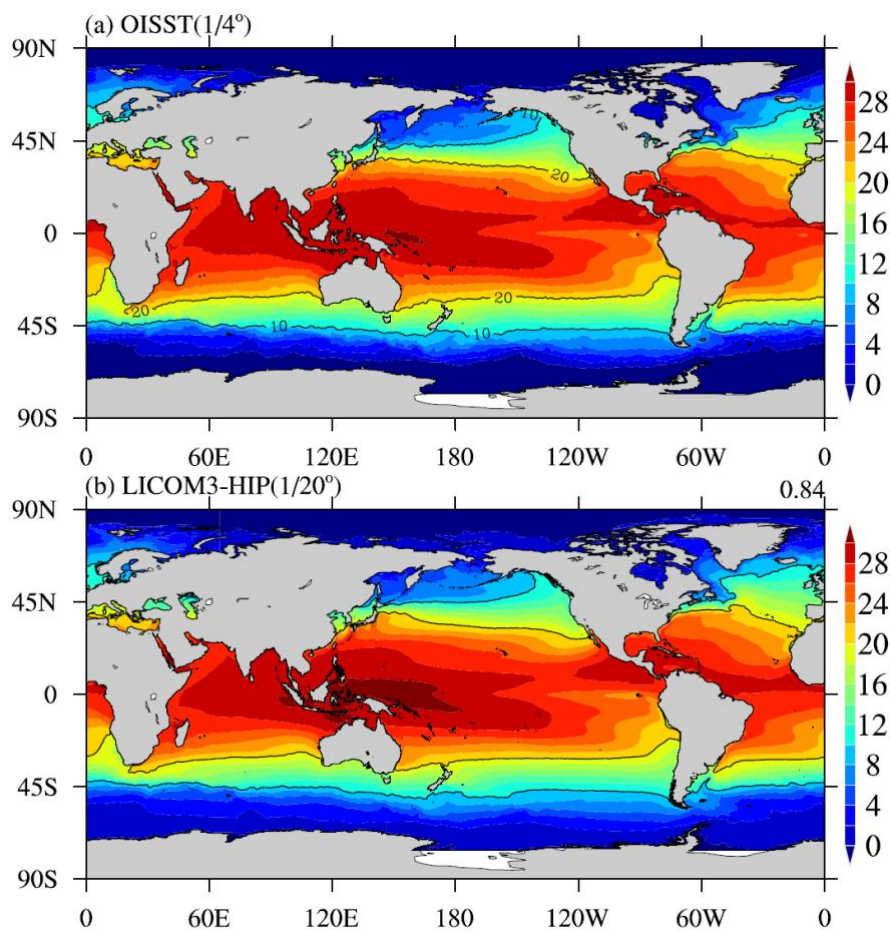



Figure 14: (a) Observed annual mean sea surface temperature in 2016 from Optimum Interpolation Sea Surface Temperature (OISST); (b) simulated annual mean SST for the LICOM3-HIP at 1/20° during the model years 0005-0014. Units: °C.

675

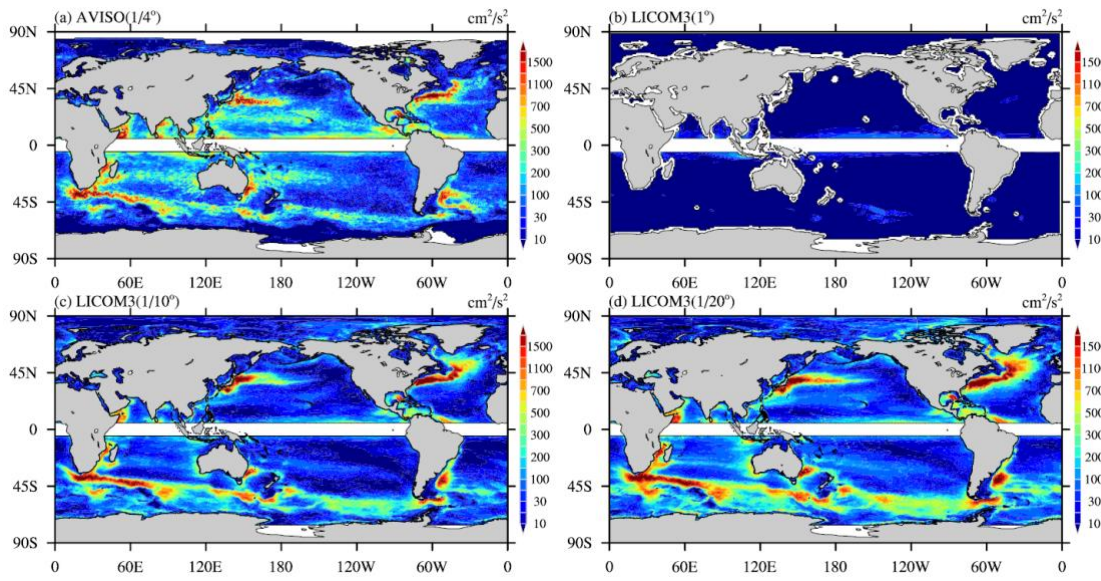



Figure 15: (a) Observed annual mean eddy kinetic energy (EKE) in 2016 from AVISO. Simulated annual mean SST in 2016 for the LICOM3-HIP at (b) 1°, (c) 1/10°, and (d) 1/20°. Units: cm^2/s^2.

Table 1: Configurations of the LICOM3 model used in the present study.

Experiment	LICOM3-CPU (1°)	LICOM3-HIP(1/10°)	LICOM3-HIP(1/20°)
Horizontal grid spacing	1° (110 km in longitude, about 110 km at the equator, and 70 km at mid-latitude)	1/10° (11 km in longitude, about 11 km at the equator, and 7 km at mid-latitude)	1/20° (5.5 km in longitude, about 5.5 km at the equator, and 3 km at mid-latitude)
Gridpoint	360×218	3600×2302	7200×3920
North Pole	(65°E, 60.8°N) and (115°W, 60.8°N)	(65°E, 65°N) and (115°W, 65°N)	(65°E, 60.4°N) and (115°W, 60.4°N)
Bathymetry data	ETOPO2	Same	Same
Vertical coordinates	30 η levels	55 η levels	55 η levels
Horizontal viscosity	Laplacian A_2=3000 m^2/s	Biharmonic (Fox-Kemper & Menemenlis, 2008) A_4=-1.0×10^{9}m^4/s	Biharmonic (Fox-Kemper & Menemenlis, 2008) A_4=-1.0×10^{8}m^4/s
Vertical viscosity	Background viscosity of 2×10^{-6}m^2/s with the upper limit of 2×10^{-2}m^2/s	Background viscosity of 2×10^{-6}m^2/s with the upper limit of 2×10^{-2}m^2/s	Background viscosity of 2×10^{-6}m^2/s with the upper limit of 2×10^{-2}m^2/s
Time steps	120/1440/1440 for barotropic/baroclinic/tracer	6s/120s/120s for barotropic/baroclinic/tracer	3s/60s/60s for barotropic/baroclinic/tracer
Bulk Formula	Large & Yeager (2009)	Same	Same
Forcing data	JRA55_do, 1958-2018, 6 hourly	JRA55_do, 2016, daily	JRA55_do, 2016, daily
Integration period	61 years / 6 cycles	14 years	14 years
Mixed layer scheme	Canuto et al. (2001, 2002)	Same	Same
Isopycnal mixing	Redi (1982); Gent & McWilliams (1990)	Laplacian	Laplacian

Bottom drag	$C_b=2.6\times10^{-3}$	$C_b=2.6\times10^{-3}$	$C_b=2.6\times10^{-3}$
Surface wind-stress	Relative wind stress	Same	Same
SSS restoring	20 m/year; 50 m/30 days for sea ice region	Same	Same
Advection scheme	Leapfrog for momentum; two-step preserved shape advection scheme for tracer	Same	Same
Time stepping scheme	Split-explicit Leapfrog with Asselin filter (0.2 for barotropic; 0.43 for baroclinic; 0.43 for tracer)	Same	Same
Sea ice	Sea ice model of CICE4	Not coupled	Not coupled
Ref.	Lin et al. (2020)	This paper	This paper

Table 2: Block partition for 1/20° setup.

GPUs	$B_x \times B_y$	$imt \times jmt$
384	600×124	604×128
768	600×62	604×66
1536	300×62	304×66
3072	150×62	154×66
6144	100×62	104×66
9216	75×62	79×66
19600	36×40	40×44
26200	36×30	40×34

Table 3: The number calls of halo in LICOM3 subroutines for each step.

Subroutine	Calls	Calls Percentage
barotr	180	96.7%
bclinc	2	1.1%
tracer	4	2.2%

Table 4: Some GPU versions of weather/climate models.

Model	Language	Max. Grids	Max GPUs	Year and references
POM.gpu	CUDA-C	1922×1442×51	4 (K20X)	2015 (Xu et al., 2015)
LICOM2	OpenACC	360×218×30	4 (K80)	2019 (Jiang et al., 2019)
FUNWAVE	CUDA-Fortran	3200×2400	2 (V100)	2020 (Yuan et al., 2020)
NICAM	OpenACC	56×56km×160	2560 (K20X)	2016 (Yashiro et al., 2016)
COSMO	OpenACC	346×340×60	4888 (P100)	2018 (Fuhrer et al., 2018)
LICOM3	HIP	7200×3920×55	26200 (gfx906)	2020 (This paper)

690

Table 5: Success and failure rates of different scales for one wall clock hour simulation.

GPUs	Success	Failure
384	98.85%	1.15%
1000	97.02%	2.98%
10000	72.90%	27.10%
26200	40.19%	59.81%