https://doi.org/10.5194/gmd-2020-323
Preprint. Discussion started: 10 December 2020

The GPU version of LICOM3 under 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 what the same number of CPU cores dose, when 384 AMD GPUs and CPU cores are used. The LICOM3-HIP has excellent scalability; it can still obtain 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 computation processes inside GPUs, and thus greatly reduces the time cost of data transfer between CPUs and GPUs. At the same time, a 14-year spin-up integration following the phase 2 of 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 differences from the CPU version. Further comparison with observations and lower-resolution LICOM3 results suggests that the 1/20° LICOM3-HIP can not only reproduce the observations, but also produce much smaller scale activities, such as submesoscale eddies and frontal scales structures.

1 Introduction

Numerical models are a powerful tool for weather forecast and for 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 of 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). The performance of such models is determined mainly by model resolution and sub-grid parameterization, as well as by surface forcing. Now, the horizontal resolution of global OGCMs has increased to about 5-10 km, which is also called eddy-resolving models. The increasing resolution will greatly 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 the performance of ocean models in terms of both 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 not only crucial to the vertical heat transport in the upper-ocean mixed layer, but also significant to biological processes (Su et al., 2018). This 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 June 2020, Central Processing Unit (CPU) and Graphics Processing Unit (GPU) heterogeneous machines account for six of the top 10. After the super computing technics on GPU were provided by the NVIDIA Corporation, 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 of energy reduction. Yashiro et al. (2016) deployed NICAM model on the TSUBAME supercomputer, and the model sustained a double-precision performance of 60T Flops/s 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/s. 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).

At the same time, the AMD company also provides their own 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 learning resources of CUDA, AMD developers have to study two kinds of GPU programing 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 develop environments, i.e., CUDA and HCC, simultaneously. The grammar of the HIP code 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. For the viewpoint of coding, the HCC/OpenACC is convenient for AMD GPU developers before the HIP is popular. 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 purpose of the study is to develop a high performance OGCM based on the LICOM3, which can be run on AMD GPU architecture using HIP framework. Here, we will not only focus on the best computing performance of the model, but also on its practical usage for research 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 is in Section 6.

2 LICOM3 model and experiments

2.1 LICOM3 model

In this study, the targeting model is LASG/IAP Climate system Ocean Model version 3 (LICOM3), which started to develop since 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 finite-volume atmospheric model (FGOALS-f3; He et al., 2020) and Flexible Global Ocean-Atmosphere-Land System model version 3 with 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 Earth System Model (CAS-ESM, Zhang et al., 2020). The paper to fully describe the new features and baseline performances of LICOM3 is 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 greatly reduced (Craig et al., 2012). That makes the coupler is suitable to apply to high resolution modelling.

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 completely eliminated, which greatly improve the scalability and efficiency of the parallel algorithm of LICOM3 model. In addition, the dynamic core of the model is also 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 bottom (St. Laurent et al., 2002). The coefficient of both isopycnal and thickness diffusivity set to 300 m^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.

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 performances of the two versions of LICOM3 comparing 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 forecast (Liu et al., 2020, under review).

2.2 Configurations of models

To investigate the GPU version, three configurations are employed in the present study. They are 1°, 0.1°, and 1/20°. Details of these models are listed in Table 1. The number 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 times. The original CPU version of 1/20° with MPI parallel on Tianhe-1A only reached 0.31 SYPD using 9216 CPU cores. This will slow down the 10-year spin-up simulation of LICOM3 to more than one month, which is not practical for climate research. Therefore, such simulations are suitable for extreme-scale high-performance computers by applying the GPU version.

Besides the differences in grid points, there are three main aspects that are different among 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 both 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 $-1.0\times10^{8} m^4/s$ for 1/20°. Second, although the forcing including dataset (JRA55-do; Tsujino et al., 2018) and the bulk formula for the three experiments is 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 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), while the low-resolution experiment does not, which was the CPU version of LICOM3 and also 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.

The spin-up experiments for two high resolution versions are both conducted for 14 years forced by 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 evolution of the kinetic energy, the models reach to a quasi-equilibrium state after more than 10 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 totally 7000 nodes, with a 1.9 Ghz X64 CPU of 32 cores on each node. Also, each node is equipped with 4 gfx906 AMD GPU cards with 16 GB memory. The GPU has 64 cores, totally 2560 threads on each card. The nodes of both partitions are interconnected through the high-performance InfiniBand (IB) networks. 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.

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 as well as on 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 place the character "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).

Now, some of the supercomputers install NVIDIA GPU cards, such as P100 and V100, and others install AMD GPU cards, for example 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. For coding on AMD GPU, our experience indicates that the HIP is a good choice for high-performance model development. The developed model ported to HIP can be used on both the AMD and NVIDIA GPU platforms. Meanwhile, the model version is easy to keep consistent and maintain. 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, majority 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 MPI to do direct MPI communication between different nodes in the GPU space, 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. 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 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 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 in the model integrating step include solving baroclinic, barotropic and thermohaline equations; and are solved by the leapfrog or Euler forward scheme. There are seven individual 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 develop language is more efficient. In the CUDA develop forum, both CUDA-C and CUDA-Fortran are provided; for the HIP, however, the support for Fortran is not as good as that for C++. We plan to push all the core process codes into GPUs; hence, the Fortran codes of the seven major subroutines must be converted to HIP/C++. Due to the complexity and large number of lines in these subroutines (approximately 12000 lines Fortran code), also for making sure the converted C/C++ codes be correct, we rewrote them to C first, before finally converting them to HIP codes.

A bit-reproducible climate model produces the exact same numerical results for a given precision, regardless of its execution setup, which includes the choice of domain decomposition, the type of simulation (continuous or restart), compilers, and the architectures executing the model (i.e., the same hardware and software conduct the same result). The C transitional version is bit-reproducible with F90 version of the LICOM3 (the binary output data are the same under Linux with "diff" command). This C transitional version becomes the starting point of HIP/C++ codes, and reduces the complexity of developing HIP version of the LICOM3.

3.3. Optimization and tuning methods in LICOM3-HIP

The unit of computation in LICOM3-HIP is horizontal grid point. For example, 1/20° corresponds to 7200×3920 grids. For convenience of MPI parallelism, the grids were united as blocks, that is, if $\mathrm{Proc}_x \times \mathrm{Proc}_y$ MPI processes are used in x and y directions, then each block has $B_x \times B_y$ grids, where $\mathrm{Proc}_x \times B_x = 7200$ and $\mathrm{Proc}_y \times B_y = 3920$. 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.

The original LICOM3 was written in F90. To adapt it to GPU, the Fortran/C hybrid programing is applied. As shown in Figure 2, the codes are kept using 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 reference to them. In the GPU space, newly allocated GPU global memories to hold the arrives correspond to those in the CPU space, and the HipMemcpy is called to copy them in and out.

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

The LICOM3-HIP is a 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. To reduce the data copy time, all read-only GPU variables are allocated and copied at the initial stage. Some data copy is still needed in the stepping loop, e.g., MPI call in barotr.cpp. In fact, four subroutines call halo in each step, but the number of halo calls in "barotr" is more than 95% of all (Table 3); so we only use "barotr" subroutine to represent the halo communications.

The computation block in MPI (corresponding to 1 GPU) is a 3-D grid; in HIP revision, the 3-D parallelism is implemented. This change adds extra parallel inside one block than the MPI solo parallelism (only 2-D). In order to adapt to this change, some optimizations are needed, such as increasing the global arrays to avoid data dependency. A demo of using temporary array to parallel the computation inside a block can be found in Figure 3. Figure 3a represents a loop of the original code in the *k* direction. Since the variable *v(i,j,k)* has dependence on *v(i,j,k+1)*, it will cause 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 3b, a temporary array *vt* is used to hold the result of *f1()*, and it can be GPU threads parallel in the *k* direction. Then, in the bottom of Figure 3b, 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 shared memory program; the memory write conflicts occur in the advection computation of subroutine "tracer". We change the if-else tree in this subroutine; hence, the data conflicts between neighboring grids are 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 it will not increase the total computation time, the reduction of memory cache improves the final performance of this subroutine.

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 data are referenced by GPU process, the system call in HipMemcpy will occur compile errors. In this situation, we should change the data structure of the original Fortran arrays from "static" to "allocatable" type. 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. 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. As shown in Fig. 4, we only pack the necessary data for halo operation from $imt \times jmt$ to $4(imt+jmt)$. This change reduces the data buffer size to $(4/imt+4/jmt)$ of the original one. The larger *imt* and *jmt* are, the less the transfer 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).

3.4. Model I/O optimization

The time cost for reading daily forcing 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 the time of one step when 1536 GPUs are applied; hence, we must optimize it for more total speedup. The cause of low performance is due to the daily data reading and scattering to all nodes every model day; we then rewrite the data reading strategy and do parallel scattering for 10 different forcing variables. Finally, the time cost of input is reduced to about 20 s, which is 1/10 of the original one (shown below).

As indicated, the core-process time cost is about 200s using 1536 GPUs. The output (65 GB data in binary format) of one model day 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 is an challenging task (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 need 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 throughput standard is 8 minutes to finish one model day, which is equivalent to 0.5 SYPD.

Figure 5 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 5a). While the scale increases to 9216 GPUs, the I/O time increases, but is still smaller than the GPU's step time (Figure 5b). The improved LICOM3 I/O totally costs about 50-90 s (depends on scales), especially the input remains stable (Figure 5c) while scale increases. This optimization of I/O maintains that the LICOM3-HIP 1/20° runs well at all practice scales for a realistic climate study. To analyze the purely parallel performance, the I/O time has been cut off from the total simulation time in the results of the follow-up tests.

Figure 6 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, which is 384 GPUs in 96 nodes versus the total 3072 CPU cores in 96 nodes. We can get the overall performance speedup of 384 being about 6-7 times. The figure also indicates that for all scales, the SYPD keeps increasing. At the scale of 9216 GPUs, the SYPD first goes beyond 2, and that is seven times of 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.

Figure 7 depicts the real times and speedups of difference GPUs computation. The green line in Figure 7a is the function of stepon time cost; it decreases while the GPU number increases. The blue curve of Figure 7a shows the increasing of speedup with the increasing 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, 384-1536 scale is a better choice for realistic modeling studies.

Figure 8 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 totally cost only about 1% of the whole computation time. When 384 GPUs are applied, the “barotr” costs about 50% of the total time (Figure 8a), which solves the barotropic equations. When GPUs are increased to 9216, time cost for each of the subroutines is decreased, but the percentage of subroutine “barotr” is increased to 62% (Figure 8b). As mentioned above, the phenomenon can be interpreted by the calling of halo 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

To test if the HIP version of the LICOM meets the requirement of numerical precision for scientific usage, the daily mean sea surface height (SSH) fields of CPU and HIP versions’ simulations are compared using the results from 1/20° experiments on a particular day, March 1st of the 4th model year (Figures 9a, 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 areas (Figure 9c); in most places, the difference values fall into the range between -0.1 and 0.1 cm. Because the hardware is different and the mathematical operation sequence of the HIP codes 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.

To evaluate the preliminary results of the global 1/20° simulation from LICOM3-HIP, the sea surface temperature (SST) of the GPU version is compared with the observed SST (Figure 10). 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 are 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 increasing horizontal resolution of the observation, we now know that mesoscale eddies are ubiquitous in the ocean with the spatial scale of 100-300 km. The rigorous eddies usually occur along major ocean currents, such as the Kuroshio and its Extension, the Gulf Stream and the Antarctic Circumpolar Current (Figure 11a). The eddies also capture more than 80% of the kinetic energy of the ocean, estimated using the satellite data (e.g., Chelton et al., 2011). Therefore, these mesoscale eddies must be solved in the ocean model. To resolve eddies of the global ocean, the horizontal resolution of a numerical model must be higher than 1/10°, but that cannot resolve the eddies in the high latitude and shallow waters (Hallberg, 2013). Therefore, higher resolution is required to resolve the eddies globally. It is clear that 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 summaries 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 the development of supercomputing, the horizontal resolution of ocean circulation models will keep on increasing, and much 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 terms of 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). Actually, 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 well and are in their best performances. 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}. 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 $(1\text{-}384\times10^{-5})^3$=98.85%, and the failure rate is then $1\text{-}(1\text{-}384\times10^{-5})^3$=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 middle 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 10 continues 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 $(1\text{-}26200\times10^{-6})^3$=92.34%, and its failure rate is $1\text{-}(1\text{-}26200\times10^{-6})^3$=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:

$$\mathrm{E} = \mathrm{TDP} \times \mathrm{N} \times 24/\mathrm{SYPD}$$

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 limited power supply. To calculate the value of E, 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 of INTEL (2 INTEL 16-core CPUs). We only include the TDP of CPUs and GPUs here.

Based on the above power measurements, the energy cost of simulations are 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 the 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 indicates the GPU is not fully loaded from this scale.

6 Conclusions

The GPU version of LICOM3 under 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 good 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.

Based on our test using the 1/20° configuration, the LICOM3-HIP is 42 times fast than the CPU dose, when 384 AMD GPUs and CPU cores are used. The LICOM3-HIP has good scalability, it can still obtain speedup of more than four on 9216 GPUs comparing to 384 GPUs. The SYPD, which is equilibrium to the speedup, keeps on 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 the performances of 10 and 100 km, although we did not analyze their performances in this article.

The efficiency of model decreases with the increasing number of GPUs. At 9216 GPUs, the model efficient 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 test of kernel functions, the decreasing efficiency was mainly caused by the latency of data copy in and out to the GPU memory in solving the barotropic equations, particular for the number of GPUs larger than 10000.

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

The eddy-resolving ocean circulation model, which is a fundamental platform for oceanography research, ocean forecast, and climate prediction and projection, can simulate the variations of the circulations, temperature, salinity, and sea level with 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 model used for climate research. 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 definitely used for short-term ocean prediction in the future.

Besides, the block size 36×30×55 (1/20° setup, 26200 GPUs) is not a large computation task for one GPU. Since one GPU has 64 cores totally 2560 threads, if a subroutine computation is 2-D, the operation in each thread 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.

There are still potential to further increase the speedup of LICOM3-HIP. Now, 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, the latency of data transfer between CPU and GPU cannot be totally overcome. So far, we can only try to reduce the time consumed through 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 been already tested by using an atmospheric model, and an average gain in computational efficiency by approximately 40% (Váňa et al., 2017). We would like to test 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/4302811#.X8mGYssvNb8 with doi: 10.5281/zenodo.4292796.

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.

Acknowledgements

The study is founded by National Key Research and Development Program (2018YFA0605904 and 2018YFA0605703), National Natural Sciences Foundation (41931183) and Strategic Priority Research Program of 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., et al. (2015). A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1. 0). *Geoscientific Model Development* (Online), 8 (9), 2829-2840. https://doi.org/10.5194/gmd-8-2829-2015

Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U., Aloisio, G., Benson, R., et al. (2017). CPMIP: measurements of real computational performance of Earth system models in CMIP6. *Geoscientific Model Develoment*, 10 (1), 19–34. https://doi.org/10.5194/gmd-10-19-2017.

Canuto, V., Howard, A., Cheng, Y., & Dubovikov, M. (2001). Ocean turbulence. Part I: One-point closure model—Momentum and heat vertical diffusivities. *Journal of Physical Oceanography*, 31(6), 1413-1426. https://doi.org/10.1175/1520-0485(2001)031<1413:OTPIOP>2.0.CO;2

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

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

Chassignet, E. P., Yeager, S. G., Fox-Kemper, B., Bozec, A., Castruccio, F., Danabasoglu, G., Kim, W. M., Koldunov, N., et al. (2020). 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). *Geoscientific Model Development*, 1-58. https://doi.org/10.5194/gmd-2019-374

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

Craig, A. P., Vertenstein, M. & Jacob, R. (2012). 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

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

Fox-Kemper, B., & Menemenlis D . Can Large Eddy Simulation Techniques Improve Mesoscale-Rich Ocean Models? Ocean Modeling in an Eddying Regime, 177, 319-338. https://doi.org/ 10.1029/177GM19.

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

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

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

Griffies, S., Danabasoglu, G., Durack, P., Adcroft, A., Balaji, V., Böning, C., Chassignet, E., Curchitser, E., et al. (2016). OMIP contribution to CMIP6: experimental and diagnostic protocol for the physical component of the Ocean Model Intercomparison Project. *Geoscientific Model Developmen*t, 9(9). https://dx.doi.org/10.5194/gmd-9-3231-2016

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

He, B., Yu, Y., Bao, Q., Lin, P. F., Liu, H. L., Li, J. X., Wang, L., Liu, Y. M, et al. (2020). CAS FGOALS-f3-L model dataset descriptions for CMIP6 DECK experiments. *Atmospheric and Oceanic Science Letters*, 1-7. https://dx.doi.org/10.10801/16742834.2020.1778419

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

Jiang, J., Lin, P., Wang, J., Liu, H., Chi, X., Hao, H., Wang, Y., Wang, W., et al (2019). 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

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

Li, L., Yu, Y., Tang, Y., Lin, P., Xie, J., Song, M., Dong, L., Zhou, T., et al. (2020). The Flexible Global Ocean Atmosphere Land System Model Grid Point Version 3 (FGOALS-g3): Description and Evaluation. *Journal of Advances in Modeling Earth Systems*. https://dx.doi.org/10.1029/2019MS002012.

Li, Y. W., Liu, H. L., Ding, M. R., Lin, P. F., Yu, Z. P., Meng, Y., Li, Y. L., et al. (2020). Eddy-resolving Simulation of CAS-LICOM3 for Phase 2 of the Ocean Model Intercomparison Project. *Advances in Atmospheric Sciences*, 37(10), 1067-1080. https://doi.org/ 10.1007/s00376-020-0057-z.

Lin, P., Liu, H., Xue, W., Li, H., Jiang, J., Song, M., Song, Y., Wang, F., et al. (2016). A Coupled Experiment with LICOM2 as the Ocean Component of CESM1. *Journal of Meteorological Research*, 30(1), 76-92. https://dx.doi.org/10.1007/s13351-015-5045-3

Lin, P., Yu, Z., Liu, H., Yu, Y., Li, Y., Jiang, J., Xue, W., Chen, K., Yang, Q., et al. (2020). LICOM Model Datasets for the CMIP6 Ocean Model Intercomparison Project. *Advances in Atmospheric Sciences*, 37(3), 239-249. https://dx.doi.org/10.1007/s00376-019-9208-5

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

Liu, H. L., Lin, P., Zheng, W., Luan, Y., Ma, J., Mo, H., Wan L., and Tiejun Ling (2019). A global eddy-resolving ocean forecast system – LICOM Forecast System (LFS). *Journal of Operational Oceanography*, under review

Madec, G., & Imbard M. (1996). A global ocean mesh to overcome the North Pole singularity. *Climate Dynamics*, 12(6), 381-388. https://doi.org/10.1007/s003820050115

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

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

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

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

Schär, C., Fuhrer, O., Arteaga, A., Ban, N., Charpilloz, C., Di Girolamo, S., Hentgen, L., Hoefler, T., et al. (2020). Kilometer-scale climate models: Prospects and challenges. *Bulletin of the American Meteorological Society*, 101(5), E567-E587. https://doi.org/ 10.1175/BAMS-D-18-0167.1

St. Laurent, L., Simmons, H., & Jayne, S. (2002). Estimating tidally driven mixing in the deep ocean. *Geophysical research letters*, 29(23). https://doi.org/ 10.1029/2002GL015633.

Su, Z., Wang, J., Klein, P., Thompson, A. F., & Menemenlis, D. (2018). 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

Tsujino, H., Urakawa, S., Nakano, H., Small, R., Kim, W., Yeager, S., Danabasoglu, G., Suzuki, T., et al. (2018). JRA-55 based surface dataset for driving ocean - sea-ice models (JRA55-do). *Ocean Modelling*, 130, 79-139. https://dx.doi.org/10.1016/j.ocemod.2018.07.002

Tsujino, H., Urakawa, L. S., Griffies, S. M., Danabasoglu, G., Adcroft, A. J., Amaral, A. E., Arsouze, T., Bentsen, M., et al. (2020). Evaluation of global ocean–sea-ice model simulations based on the experimental protocols of the Ocean Model Intercomparison Project phase 2 (OMIP-2). *Geoscientific Model Development Discussion*, 1-86. https://doi.org/ 10.5194/gmd-2019-363

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

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

Xiao, C. (2006). Adoption of a two-step shape-preserving advection scheme in an OGCM and its coupled experiment, (master's thesis). Institute of Atmospheric Physics, Chinese Academy of Sciences.

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

Yang, C., Xue, W., Fu, H., You, H., Wang, X., Ao, Y., Liu, F., Gan, L., et al. (2016), 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.

Yashiro, H., M. Terai, R. Yoshida, S.-i. Iga, K. Minami, & H. Tomita (2016). 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.

Yu, Y., Tang, S., Liu, H., Lin, P., & Li X. (2018). Development and evaluation of the dynamic framework of an ocean general circulation model with arbitrary orthogonal curvilinear coordinate. *Chinese Journal of Atmospheric Sciences*, 42(4), 877-889. https://doi.org/ 1006-9895(2018)42:4<877:RYZJQX>2.0.TX;2-B

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

Zhang, X., & Liang X. (1989). A numerical world ocean general circulation model, *Advances in atmospheric sciences*, 6(1), 44-61. https://doi.org/ 0256-1530(1989)6:1<44:ANWOGC>2.0.TX;2-S

Zhang, H., M., Jin, J., Fei, K., Ji, D., Wu, C., Zhu, J., He, J., et al. (2020). CAS-ESM2: Description and Climate Simulation Performance of the Chinese Academy of Sciences (CAS) Earth System Model (ESM) Version 2. *Journal of Advances in Modeling Earth Systems*, under review

https://doi.org/10.5194/gmd-2020-323
Preprint. Discussion started: 10 December 2020

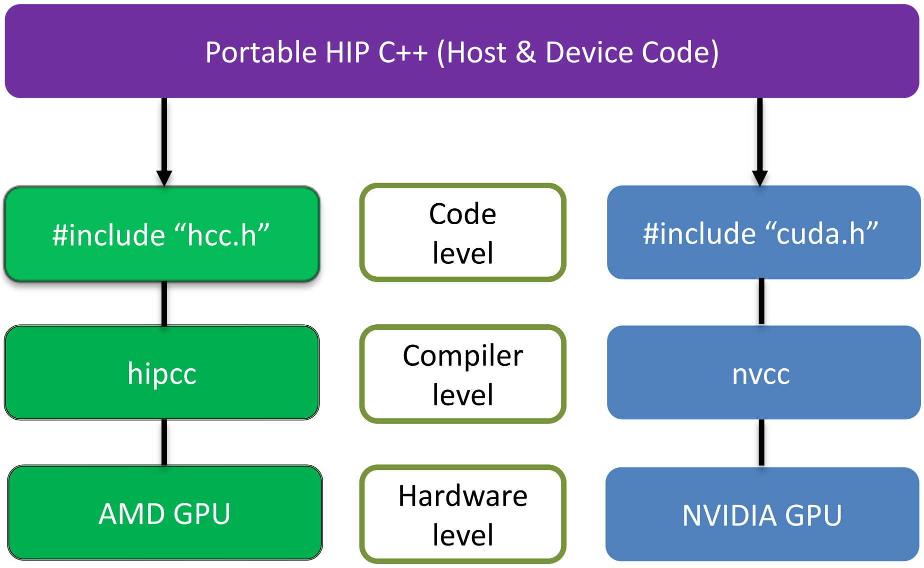


Figure 1: The schematic diagram of the comparison of coding on AMD and NVIDIA GPU in three levels.

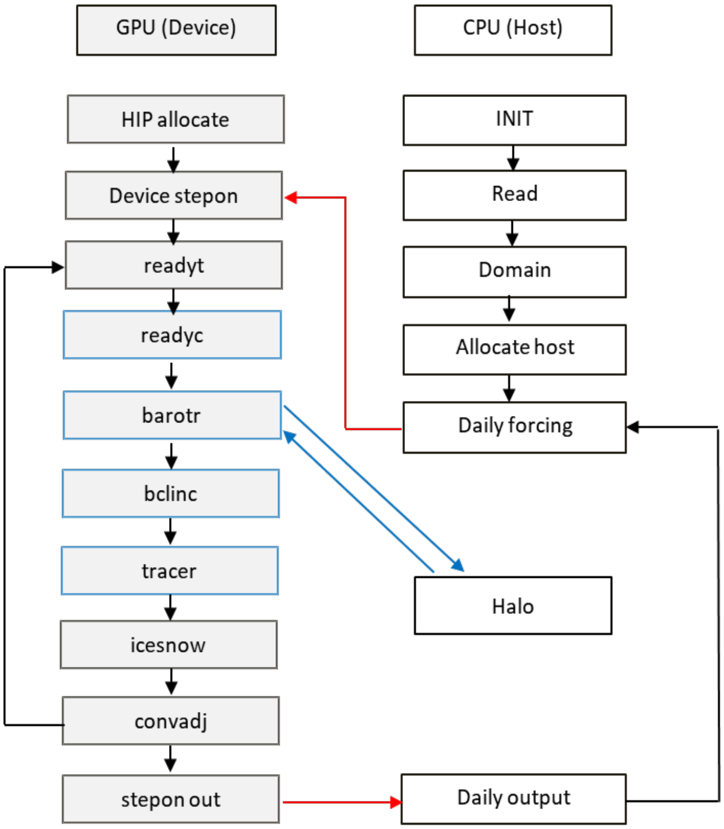
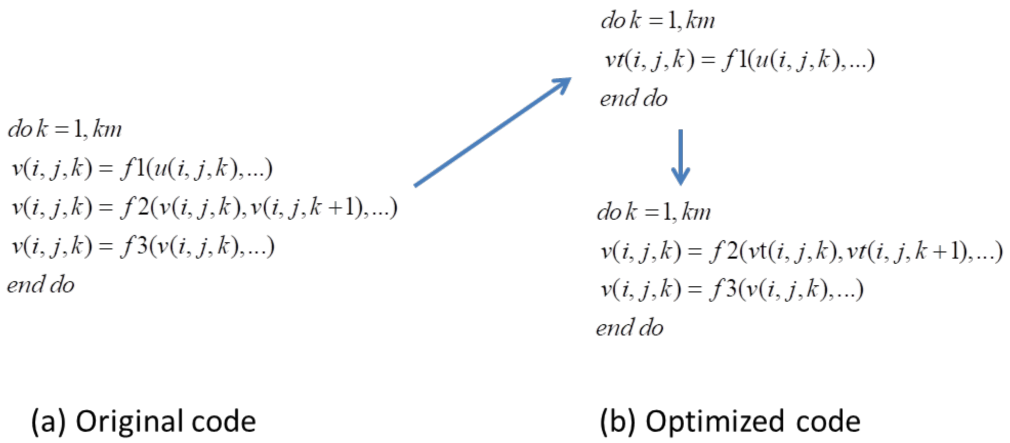


Figure 2: LICOM3 computation flowchart with GPU (HIP device). The red line indicates whole block data transfer between host and GPU, while blue line means only transferring lateral data of a block.



```
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
```

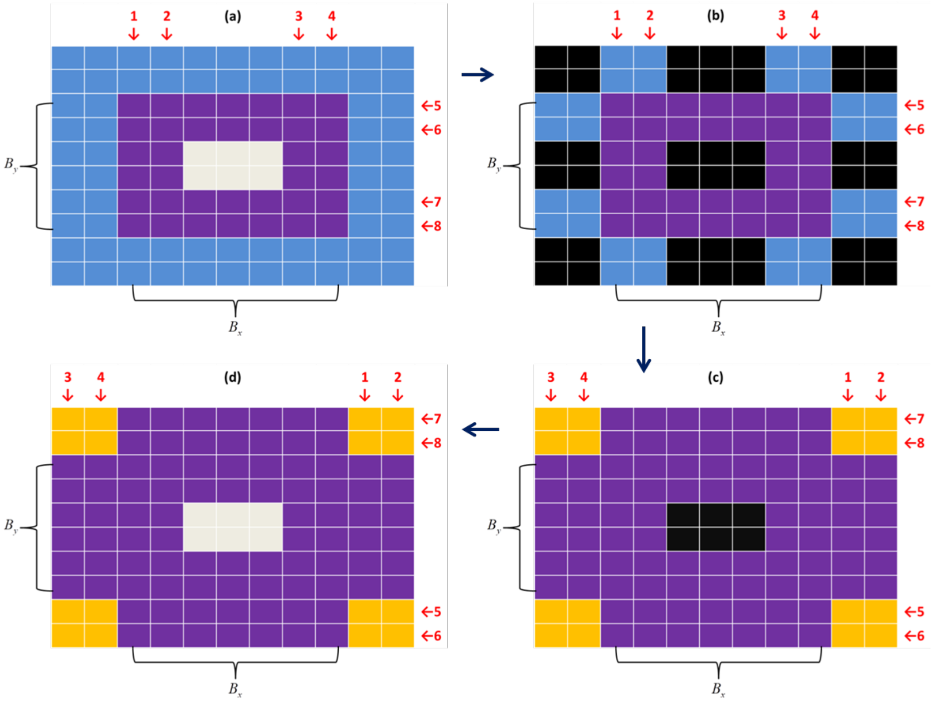
```
do $k = 1, km$
    $v(i,j,k) = f2(vt(i,j,k), vt(i,j,k+1),...)$
    $v(i,j,k) = f3(v(i,j,k),...)$
end do
```

(a) Original code

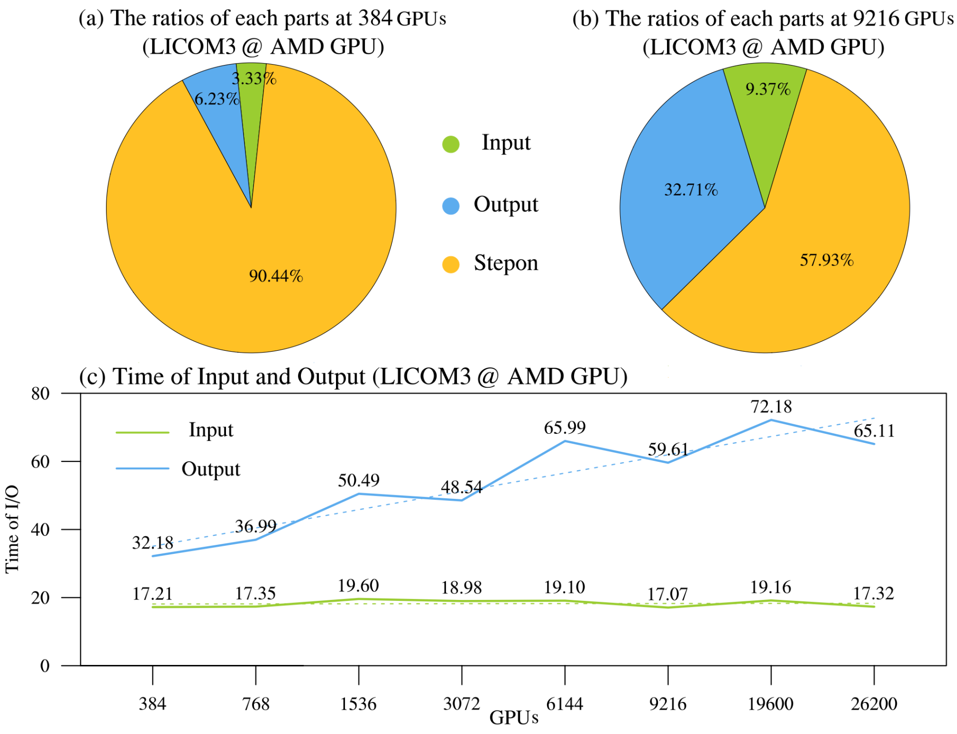
(b) Optimized code

Figure 3: The code using temporary arrays to avoid data dependency.

https://doi.org/10.5194/gmd-2020-323
Preprint. Discussion started: 10 December 2020



545 **Figure 4: The lateral packing (only transfer 4 rows and 4 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 neighbours; and (d) transfer back to the GPU space.**



550 **Figure 5: (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.**

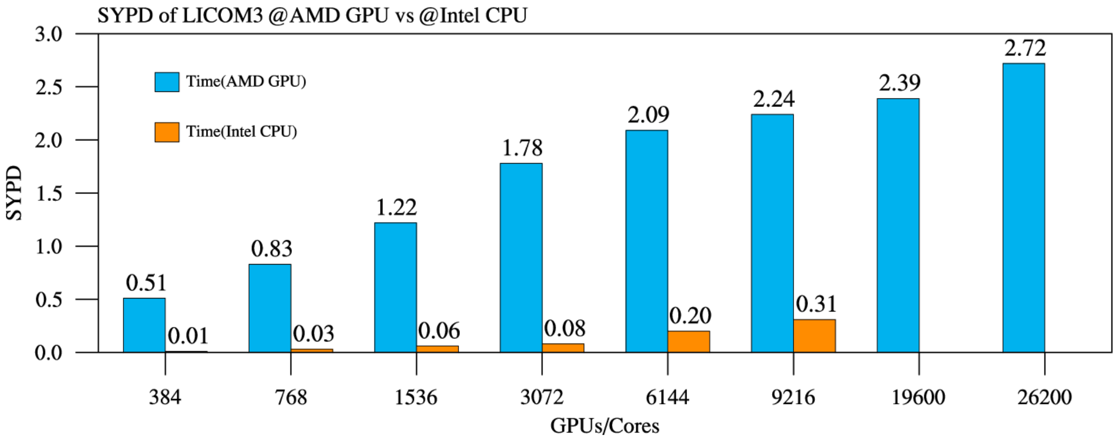


Figure 6: Simulation performances of AMD GPU versus Intel CPU core for LICOM3 (1/20°). Unit: SYPD.

555

https://doi.org/10.5194/gmd-2020-323
Preprint. Discussion started: 10 December 2020

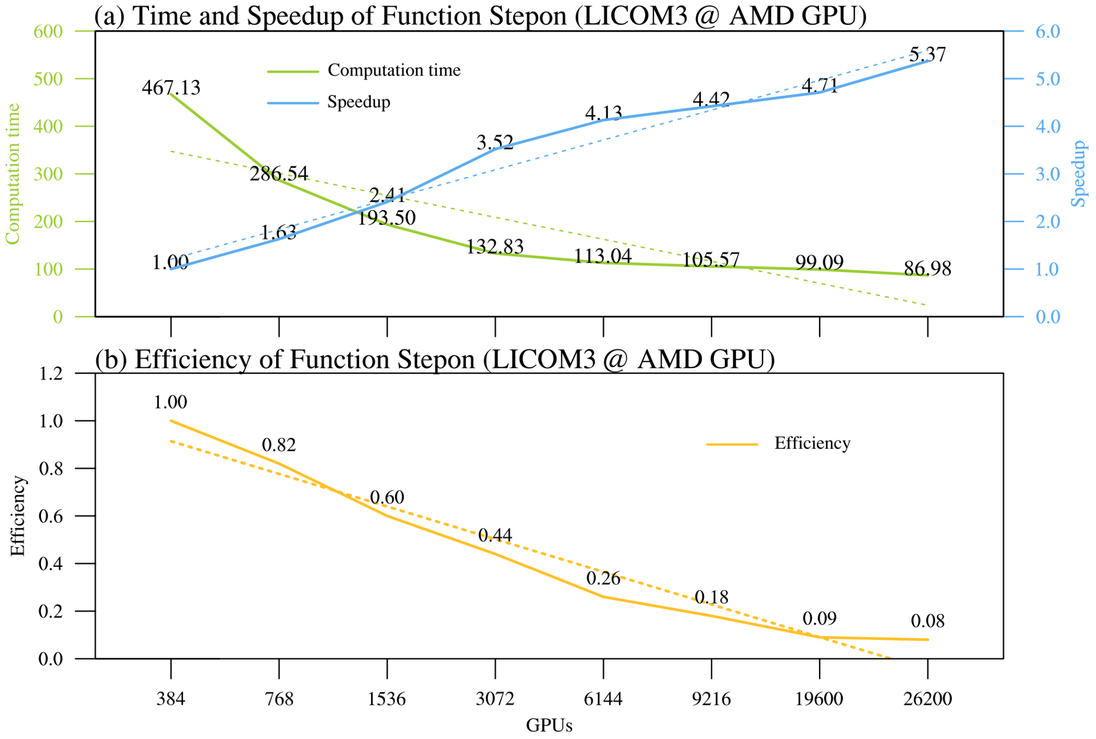


Figure 7: (a) Computation time (green) and speedup (blue), and (b) parallel efficiency (orange) at different scales for stepon of LICOM3-HIP (1/20°).

https://doi.org/10.5194/gmd-2020-323
Preprint. Discussion started: 10 December 2020

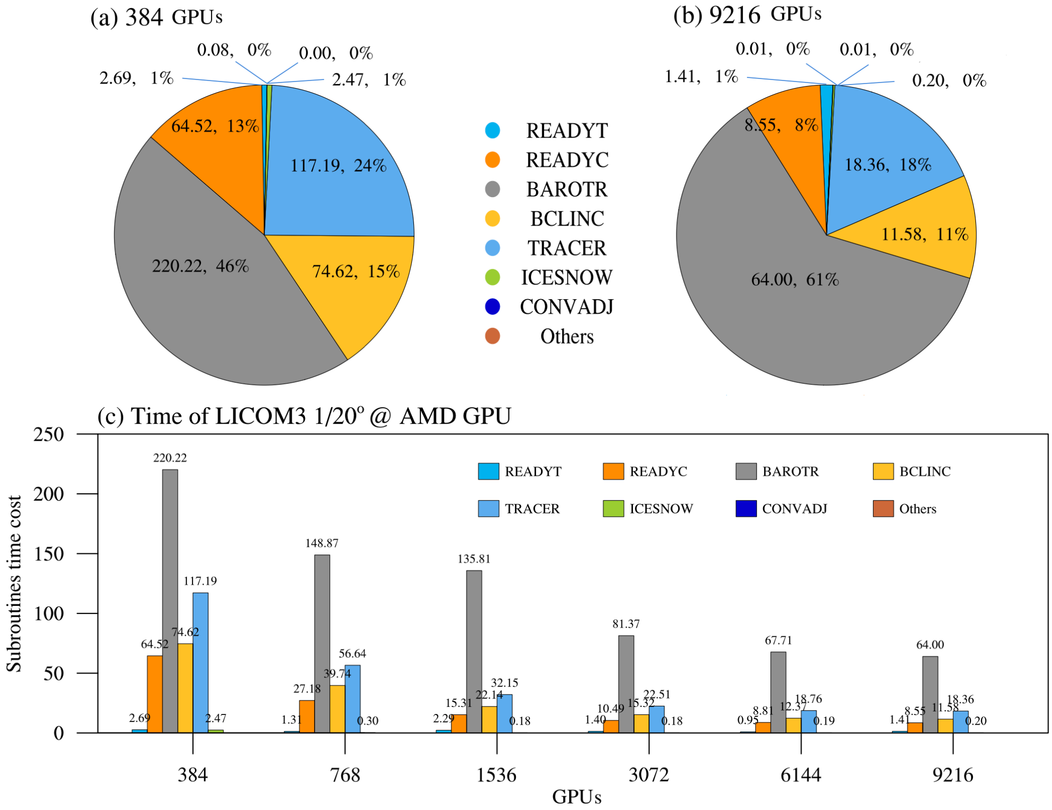
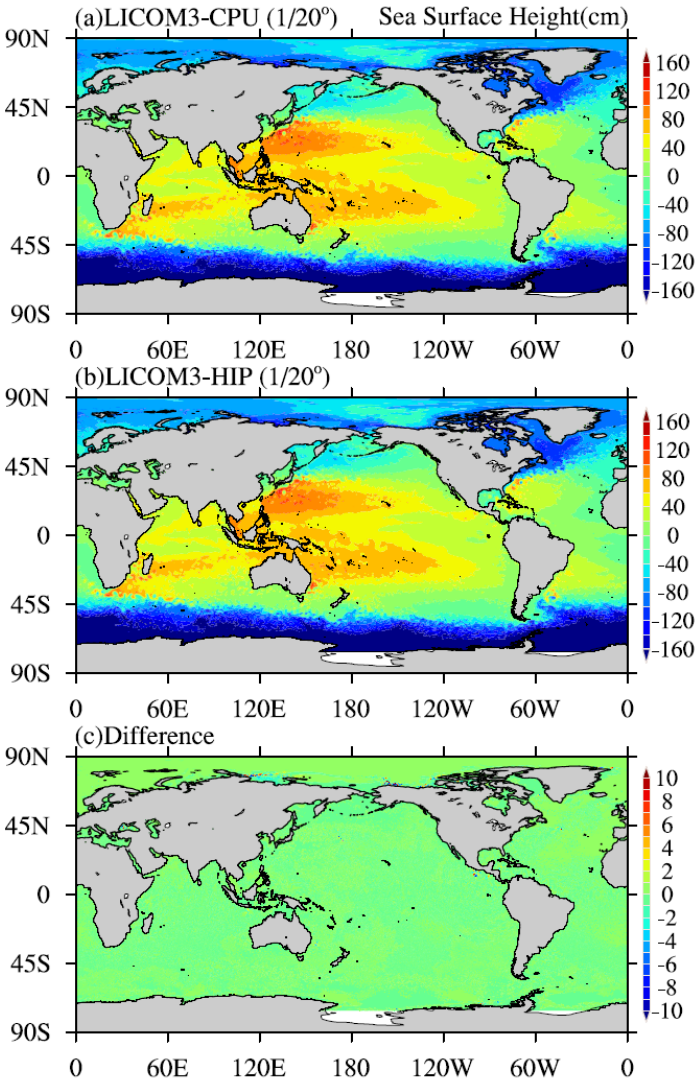
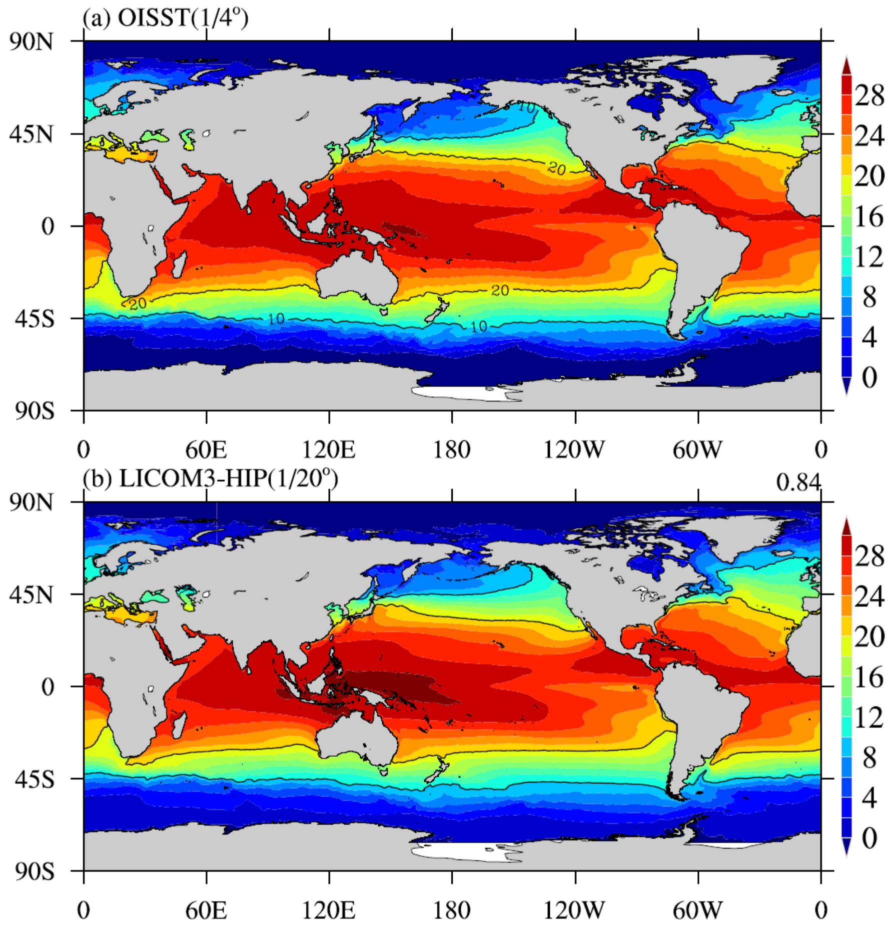


Figure 8: The seven core subroutines' time cost percentage for (a) The 384 GPUs and (b) 9216 GPUs. (c) the subroutines' time cost at different scales of LICOM3-HIP (1/20°).

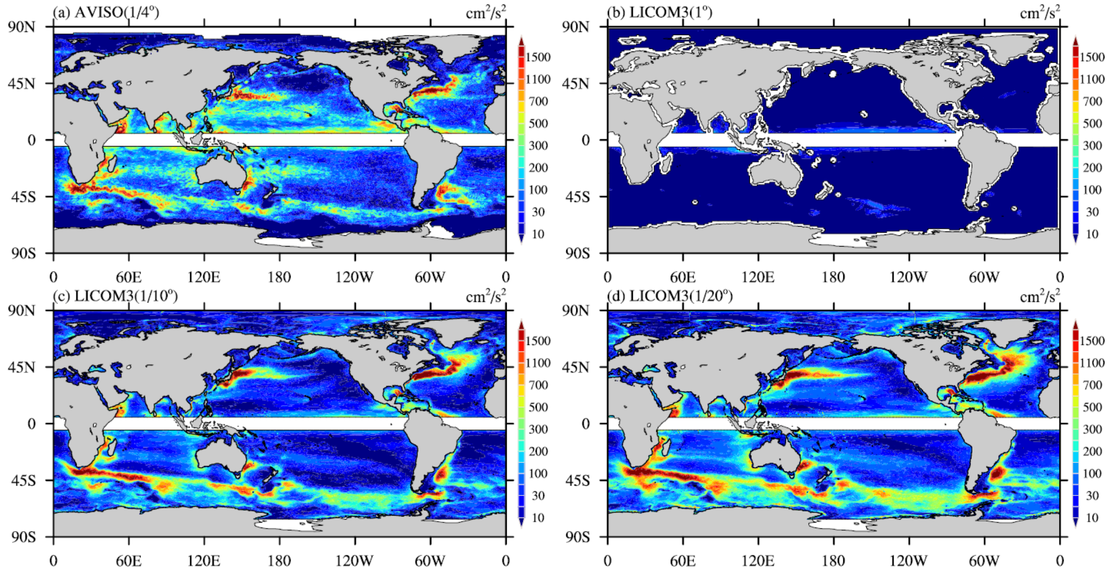


565 **Figure 9: Daily mean simulated sea surface height for (a) CPU and (b) HIP versions of LICOM3 at 1/20° on March 1st of the 4th model year. (c) The difference between the two versions (HIP minus CPU). Units: cm.**



570 **Figure 10: (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.**

https://doi.org/10.5194/gmd-2020-323
Preprint. Discussion started: 10 December 2020



575 **Figure 11: (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 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 equator, and 70 km at mid-latitude)	1/10° (11 km in longitude, about 11 km at equator, and 7 km at mid-latitude)	1/20° (5.5 km in longitude, about 5.5 km at equator, and 3 km at mid-latitude)
Grid point	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=2.6×10^{-3}	C_b=2.6×10^{-3}	C_b=2.6×10^{-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	40×66
26200	36×30	40×34

Table 3: The number and ratio to the total number of calls of halo in LICOM3 subroutines.

Subroutine	Calls	Percentage
barotr	122	95.2%
bclinc	3	2.4%
readyc	1	0.8%
tracer	2	1.6%

585

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)

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%