# A Performance-Portable Kilometer-Scale Global Ocean Model on ORISE and New Sunway Heterogeneous Supercomputers

Junlin Wei[a,c,e] Xiang Han[a,c] Jiangfeng Yu[b,c] Jinrong Jiang[a,c,g,†] Hailong Liu[d,†] Pengfei Lin[b,c,†] Maoxue Yu[d] Kai Xu[d]
Lian Zhao[a] Pengfei Wang[b] Weipeng Zheng[b,c] Jingwei Xie[d] Yanzhi Zhou[b,c] Tao Zhang[b,c] Feng Zhang[a,c] Yehong Zhang[e]
Yue Yu[e] Yuzhu Wang[f] Yidi Bai[a] Chen Li[a] Zipeng Yu[b] Haoyu Deng[a,c] Yaxin Li[a,c] Xuebin Chi[a,c,†]

*a. Computer Network Information Center, Chinese Academy of Sciences, Beijing, China*
*b. Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, China*
*c.University of Chinese Academy of Sciences, Beijing, China*
*d. Laoshan Laboratory, Qingdao, China*
*e. Pengcheng Laboratory, Shenzhen, China*
*f. School of Information Engineering, China University of Geosciences, Beijing, China*
*g. Hangzhou Institute for Advanced Study, UCAS, Hangzhou, China*

***Abstract*—Ocean general circulation models (OGCMs) are indispensable for studying the multi-scale oceanic processes and climate change. High-resolution ocean simulations require immense computational power and thus become a challenge in climate science. We present LICOMK++, a performance-portable OGCM using Kokkos, to facilitate global kilometer-scale ocean simulations. The breakthroughs include: (1) we enhance cutting-edge Kokkos with the Sunway architecture, enabling LICOMK++ to become the first performance-portable OGCM on diversified architectures, i.e., Sunway processors, CUDA/HIP-based GPUs, and ARM CPUs. (2) LICOMK++ overcomes the one simulated-years-per-day (SYPD) performance challenge for global realistic OGCM at 1-km resolution. It records 1.05 and 1.70 SYPD with a parallel efficiency of 54.8% and 55.6% scaling on almost the entire new Sunway supercomputer and two-thirds of the ORISE supercomputer. (3) LICOMK++ is the first global 1-km-resolution realistic OGCM to generate scientific results. It successfully reproduces mesoscale and submesoscale structures that have considerable climate effects.**

***Index Terms*—High-Performance Computing, Ocean General Circulation Model, Performance-Portable, Sunway Architecture**

## I. Justification for ACM Gordon Bell Prize for Climate Modelling

We present an efficient and performance-portable implementation of a kilometer-scale global ocean general circulation model LICOMK++ suitable for various architectures. Using 38366250 Sunway cores and 16000 HIP-based GPUs, we achieve the performances of 1.05 and 1.70 simulated-years-per-day for global realistic simulation at 1-km resolution that successfully reproduces submesoscale processes.

## II. Performance Attributes

| Attribute title | Attribute value |
|---|---|
| Category achievement | Time-to-solution |
| Type of method used | Semi-implicit |
| Results reported on the basis of | Whole application excluding I/O and initialization |
| Precision reported | Double precision |
| System scale | Results measured on near fullsystem scale |
| Measurement mechanism | Timers |

## III. OVERVIEW OF THE PROBLEM

Oceans directly affect the climate by absorbing and redistributing heat. Approximately 90% of the increased anthropogenic warming of the climate system is stored in the ocean [1], which substantially slows the rapid warming of the Earth's surface owing to the large oceanic thermal inertia [2]. The efficiency of the ocean heat uptake is affected considerably by widespread and energetic oceanic eddies at the meso- (10∼100 km) and submesoscale (0.1∼10 km) [3]–[5]. However, these eddying effects are difficult to resolve and sometimes need to be treated by physical parameterization schemes in climate models [6], [7]. In addition to cloud processes in the atmosphere, these oceanic eddying processes are also considered a large source of uncertainty for climate simulations and projections [6].

Improving the horizontal resolution of ocean models is an effective way to enhance the description of the eddying effects and significantly reduce model uncertainties. However, this forms a major computational challenge because of the requirements of kilometer-scale (horizontal) and meter-scale (vertical) resolution measured based on the Rossby deformation radius and baroclinic modes [8], [9]. Although this method presents resolution that are several times finer than the resolution of the latest intergovernmental panel on climate change models

[†] Corresponding Authors: Jinrong Jiang (jjr@sccas.cn), Hailong Liu (hlliu2@qnlm.ac), Pengfei Lin (linpf@mail.iap.ac.cn), and Xuebin Chi (chi@sccas.cn)

SC24, November 17-22, 2024, Atlanta, Georgia, USA
979-8-3503-5291-7/24/$31.00 ©2024 IEEE

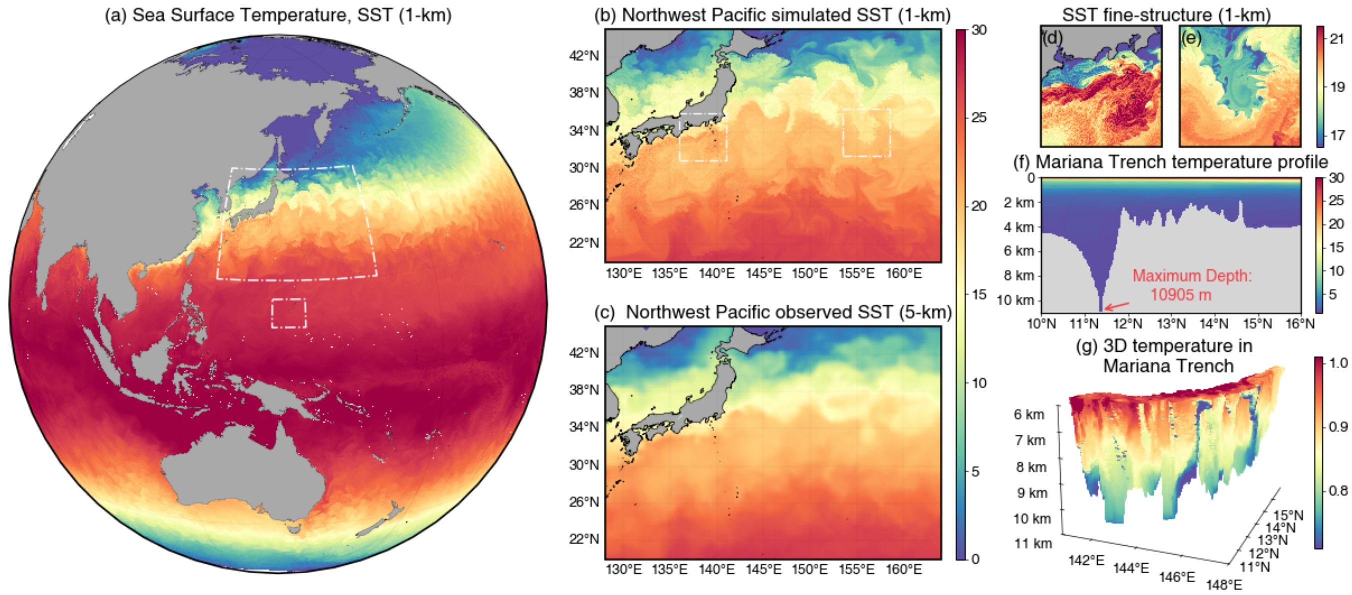


Fig. 1: (a) The sea surface temperature (SST) simulated by LICOMK++ at 1-km resolution. (b) and (c) are SST in Northwest Pacific from 1-km LICOMK++ and 5-km OSTIA (Operational Sea Surface Temperature and Ice Analysis) observation datasets, respectively. (d) and (e) zoom in on the regions of white boxes in (b) to show fine-scale structures in SST. (f) temperature profile along 142.5°E section simulated at 2-km resolution, in which the maximum depth of the model topography (10905 m, red arrow) is close to the bottom of the Mariana Trench. (g) Three-dimensional structure of temperature field below 6000 m in the Mariana Trench.

(approximately 50 km on average), it leads to almost $10^5$ times higher computational workload. The growth rate of the computational demand for ocean general circulation models (OGCMs) is much higher than the development rate of high-performance computing (HPC) systems involving supercomputers (from the petascale to the exascale).

The achievement of a kilometer-scale simulation depends on the performance of the ocean model executed by cutting-edge supercomputers. Porting and optimizing model code on a specific machine is a common challenge for ocean modelling communities. The challenge is threefold. First, the architecture of the supercomputer changes from a homogeneous central processing unit (CPU) cluster to a heterogeneous system. Second, the ocean model codes, which usually span decades to develop and are iterated by the scientific community, are too complicated to port efficiently onto a new system. Third, the workloads of the ocean models have been distributed among many modules with no substantial hotspots.

The diversified architectures have led the HPC and climate communities to develop performance portability technology that continuously adapts large legacy codes to new architectures [10]. Kokkos is one of the outstanding tools that provide a solution for the performance portability challenge [11]–[13]. It has been successfully employed by the climate model community and has achieved remarkable milestones [14], [15]. The new-generation, heterogeneous Sunway supercomputer, a potential exascale system, offers the opportunity to develop a kilometer-scale OGCM. The precursor of the new system won first place on the TOP500 list [16] four times from 2016 to 2017 and is still 13th on the latest list (June 2024). This new system facilitates advancements in extreme-scale simulations [17]–[19]. However, this system uses a complex programming model, and its numerous nodes make the acceleration of the models extremely difficult. Regrettably, Kokkos currently lacks support for Sunway's many-core heterogeneous systems, a significant architecture limitation. **This paper addresses this gap by advancing Kokkos to support the Athread programming model for Sunway processors, as indicated in Table I.**

TABLE I: Programming models and Kokkos have supported the major modern architectures in the TOP500 supercomputers since 2010.

| Architecture | Programming model | Supported by Kokkos |
|---|---|---|
| Intel coprocessors | OpenMP | Yes |
| ARM CPUs | OpenMP | Yes |
| NVIDIA GPUs | CUDA | Yes |
| AMD GPUs | HIP | Yes |
| **Sunway many-cores** | **Athread** | **Yes (This work)** |

The State Key Laboratory of Numerical Modelling for Atmospheric Sciences and Geophysical Fluid Dynamics (LASG) at the Institute of Atmospheric Physics (IAP) has been developing the OGCM LICOM (LASG IAP Climate System Ocean Model) since the 1980s [20]–[23]. This study was based on the well-established LICOM version employed in the Coupled

Model Intercomparison Project and Ocean Model Intercomparison Project experiments [22]–[27]. Figure 1 shows the snapshot of sea surface temperature (SST) on 19th January 2016. This is simulated by the ORISE supercomputer using 4000 HIP-based graphics processing units (GPUs). Figure 1a shows the SST field in the global 1-km-resolution simulation. The model well captures the high-temperature warm pool, the temperature gradient between the tropical and polar regions, and a large gradient in the Kuroshio and its extension region. Figures 1b and 1c show the Northwest Pacific SST fields of the global 1-km-resolution simulation and the global 5-km-resolution OSTIA observation dataset, respectively. The global 1-km-resolution model output shows high consistency on large-scale flow and mesoscale eddies with the observation dataset, which validates the model run. The global 1-km-resolution simulation also demonstrates richer submesoscale structures (Figures 1d and 1e). Moreover, the high resolution allows the model to precisely express the ocean bottom topography (e.g., seamounts and trenches). We develop a full-depth version of 2-km horizontal resolution and 244 vertical levels to resolve the Challenger Deep in the Mariana Trench below the depth of 10000 m and obtain the three-dimensional temperature structure deep in the ocean (Figures 1f and 1g).

## IV. CURRENT STATE OF THE ART

Before 2020, ocean modelling efforts primarily focused on coarser resolution (typically of the order of 100 km), leveraging the capabilities of small-scale computing platforms [28]–[32]. However, the critical role of ocean modelling in refining weather and climate predictions, particularly in polar and oceanic regions [33], has driven the pursuit of simulations at higher resolution.

To understand the research landscape on accelerating ocean models for improved execution in large-scale systems, Fig. 2 provides a comprehensive overview. In 2020, Zeng et al. leveraged the Athread programming model to port high-resolution, parallel ocean program (POP2) for execution on the Sunway TaihuLight system. Utilizing 1189500 cores, this optimization effort yielded a performance enhancement of POP2 within the Community Earth System Model (CESM) G-compset. The simulation speed increased from 1.43 SYPD to 5.5 SYPD [34]. Veros, a global ocean model written in Python, will be introduced in 2021. This model leverages the message passing interface (MPI) for communication and the JAX library for computation. Veros operates at a 0.1° eddy-resolving resolution and achieves a performance of 0.8 SYPD when executed on 16 NVIDIA A100 GPUs [35]. By leveraging the power of the new Sunway system, Ye et al. (2022) enhanced the performance of the Nucleus for the European Modelling of the Ocean model (NEMO4). The optimized model achieved a peak performance of 1.97 Pflops, translating to a simulation speed of 0.42 SYPD. This improvement was achieved on a massive scale using 27988480 cores and a 500-m resolution [36]. These studies showcase the utilization of diverse intranode programming models. This highlights the need for advancements in portability to leverage the full potential of modern computing architectures.

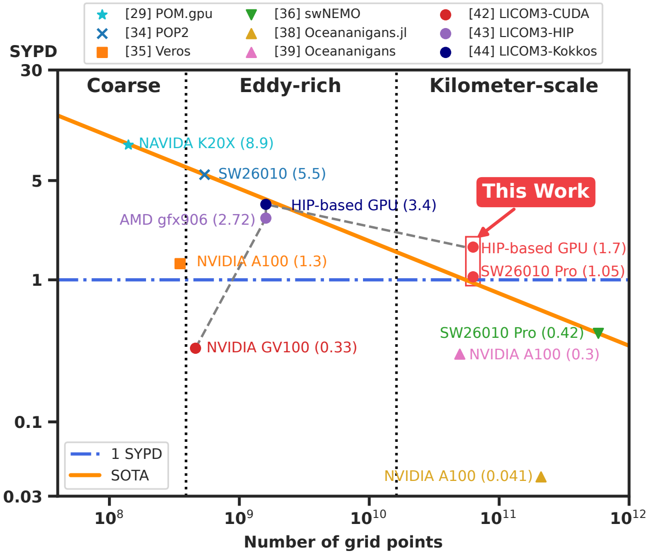


Fig. 2: Summary of recent high-resolution ocean modelling efforts on large-scale systems. These represent the continuous efforts expended toward the completion of simulations at a global kilometer-scale resolution. The results of this study have been marked with red points.

A new ocean model, called Oceananigans, was developed from scratch using the Julia programming language [37]. An Oceanographic model based on a Julia library has a key advantage because it runs efficiently across various computing architectures. Furthermore, performance testing has thus far focused on NVIDIA GPUs. Oceananigans achieved 0.3 SYPD at a $1/96°$ resolution in realistic ocean setups. Additionally, it demonstrated the speed of idealized bathymetry and surface-forcing simulations, reaching 15 simulated days per wall clock day (SDPD, equivalent to 0.041 SYPD) at 488-m resolution on a Perlmutter system of 768 NVIDIA A100 GPUs [38], [39]. In climate and atmospheric models, the nonhydrostatic dynamical core of the Energy Exascale Earth System Model (E3SM) was based on Kokkos, as of 2019, to facilitate performance portability across CPUs and GPUs [40]. The model's performance has shown remarkable improvements. In 2020, it achieved 0.97 SYPD on the Summit system [41] when executed at the realistic cloud resolution of 3 km. In 2023, the E3SM Atmosphere Model surpassed performance records by reaching 1.26 SYPD on the Frontier system at the simple cloud-resolving resolution of 3.25 km [14].

The LICOM team pioneered the development of high-resolution ocean models for heterogeneous systems. The recent efforts of the LICOM team have focused on achieving performance portability for LICOM3. Previous studies have successfully ported LICOM to specific architectures (NVIDIA GPUs with OpenACC and CUDA, and AMD GPUs with HIP) [32], [42], [43], and Wei et al. (2024) introduced LICOM3-Kokkos.

This version utilized Kokkos; it is associated with a key advancement, allowing the model to run efficiently on various computing systems without code modifications. This approach is crucial to leverage the full potential of modern heterogeneous architectures. LICOM3-Kokkos demonstrates this benefit by achieving an impressive 3.4 SYPD at 5-km resolution on a system with 4096 HIP-based GPUs [44].

This study enhances Kokkos with the Sunway many-core processor. We leverage the updated Kokkos library to develop an efficient, performance-portable, full-depth, global OGCM implementation with kilometer-scale resolution, resulting in a new model named LICOMK++. The breakthroughs of this study are the following:

- Performance portability is achieved on almost all types of supercomputers listed in the TOP500 list of supercomputers, including Sunway many-core processors, NVIDIA GPUs, AMD/HIP-based GPUs, and ARM CPUs.
- LICOMK++ is the first realistic OGCM to achieve global 1-km (0.01°) horizontal resolution and successfully reproduce scientific results (e.g., oceanic mesoscale and submesoscale processes).
- The global 1-km-resolution LICOMK++ exhibits strong performance scaling responses across heterogeneous architectures. It achieves 1.05 and 1.70 SYPD with a parallel efficiency of 54.8% and 55.6% scaling on the Sunway supercomputer (38366250 cores) and the ORISE supercomputer (16000 HIP-based GPUs), respectively.

## V. Innovations Realized

### A. *Overview of LICOMK++ model*

LICOMK++ employs tripolar and Arakawa-B grids, a two-step shape-preserving advection scheme, and a split-explicit leapfrog time-stepping scheme with Asselin filtering. More details can be found in the work [43]. Fig. 3 illustrates the overview of the LICOMK++ model, demonstrating how primitive equations are mapped to diverse hardware architectures for parallel computing.

Appropriate expression of vertical mixing is essential for simulating kilometer-scale oceanic dynamics. Building upon the LICOM3-Kokkos [44], LICOMK++ introduces a new *canuto* parameterization scheme [45], [46] for enhancing vertical mixing processes.

### B. *Enhancing Kokkos with Athread*

In the code 1, we present a new Kokkos syntax as an example using a basic linear algebra operation with the AXPY functor that performs the "$Y = \alpha X + Y$" basic linear algebra operation. The new Kokkos functor differs from the standard definition in that it requires an additional line of code, **KOKKOS_REGISTER_FOR_1D (Arg1, Arg2)**, which is a macro for implementing a preset function. The function reinterprets template parameters and executes the functor on the Sunway processor's computing processing elements (CPEs). The **FOR** construct represents a parallel loop modifiable to **REDUCE** for reduction operations, and **1D** signifies a one-dimensional loop adaptable to any dimension. The **Arg1** parameter specifies a user-defined function name, and **Arg2** indicates the functor class.

```
template<typename T>
class FunctorAXPY {
 public:
  using View1D = Kokkos::View<T *>;
  FunctorAXPY (const T &alpha,
      const View1D &v_x, const View1D &v_y)
    : a_(a), v_x_(v_x), v_y_(v_y) {}
  KOKKOS_INLINE_FUNCTION
  void operator () (const int &i) const {
    v_y_(i) = a_ * v_x_(i) + v_y_(i);
    return ;
  }
 private:
  const T a_;
  const View1D v_x_, v_y_;
};
KOKKOS_REGISTER_FOR_1D (my_axpy,
 ↪  FunctorAXPY<double>)
```

Code 1: An example AXPY functor.

**Challenge:** Details of the Sunway architecture are presented in Section VI-A. Athread is a vendor-provided lightweight parallel-computing library tailored for direct CPE drives. The implementation of an Athread backend for Kokkos presents two primary challenges.

- Sunway architecture provides limited support for advanced C++ features, which are essential for Kokkos
- Athread diverges from common parallel-computing paradigms. Athread is integrated into a unified programming model while it utilizes CPEs efficiently

Kokkos utilizes complex template specializations to launch its functions. However, the Athread API for initiating kernels on CPEs supports only C syntax, which does not allow the passage of template parameters to CPE-run kernels. Consequently, template functors cannot be specialized.

**Innovations:** Enhancing Kokkos with Athread using our innovative approach relies heavily on functional registration and callback techniques. Kokkos leverages template metaprogramming to achieve kernel abstraction by developers. Its kernel launch process specializes in incorporating template parameters into concrete functional classes. This enables the enumeration of these specialized template parameters. We created preset functions that execute kernel statements by explicitly invoking the overloaded "*operator* ()" method within the class functor. To link these functions with their corresponding functors at runtime, we leveraged a combination of registration functions and a callback strategy. During the initialization of Kokkos, each preset function is registered, enabling Kokkos to identify and invoke the functor when a specific kernel needs to be launched on the CPEs.

A linked-list data structure was selected to implement the registration and lookup. This choice prioritizes a trade-off between the temporal and spatial complexities while maintaining robustness. Linked lists, classified as linear data structures, exhibit average temporal and spatial complexities of $\mathcal{O}(n)$,

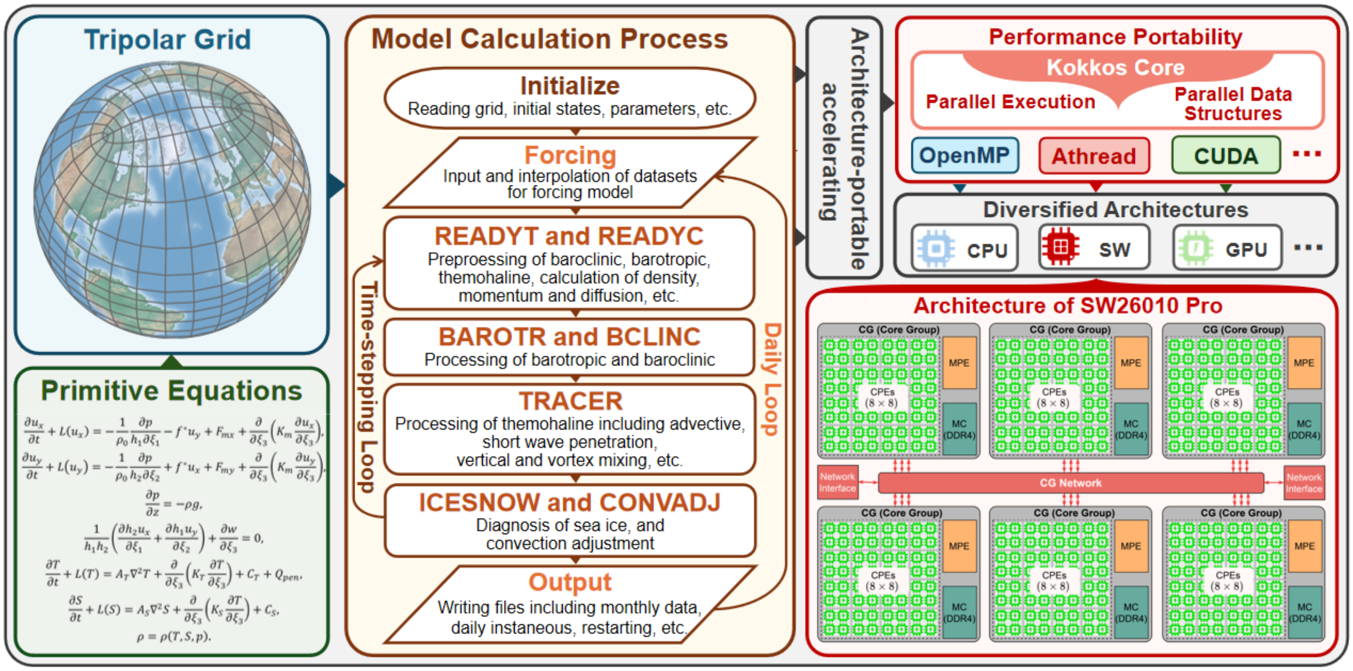

Tripolar Grid
Primitive Equations
$\frac{\partial u_x}{\partial t} + L(u_x) = -\frac{1}{\rho_0}\frac{\partial p}{h_1 \partial \xi_1} - f^* u_y + F_{mx} + \frac{\partial}{\partial \xi_3}\left(K_m \frac{\partial u_x}{\partial \xi_3}\right),$
$\frac{\partial u_y}{\partial t} + L(u_y) = -\frac{1}{\rho_0}\frac{\partial p}{h_2 \partial \xi_2} + f^* u_x + F_{my} + \frac{\partial}{\partial \xi_3}\left(K_m \frac{\partial u_y}{\partial \xi_3}\right),$
$\frac{\partial p}{\partial z} = -\rho g,$
$\frac{1}{h_1 h_2}\left(\frac{\partial h_2 u_x}{\partial \xi_1} + \frac{\partial h_1 u_y}{\partial \xi_2}\right) + \frac{\partial w}{\partial \xi_3} = 0,$
$\frac{\partial T}{\partial t} + L(T) = A_T \nabla^2 T + \frac{\partial}{\partial \xi_3}\left(K_T \frac{\partial T}{\partial \xi_3}\right) + C_T + Q_{pen},$
$\frac{\partial S}{\partial t} + L(S) = A_S \nabla^2 S + \frac{\partial}{\partial \xi_3}\left(K_S \frac{\partial T}{\partial \xi_3}\right) + C_S,$
$\rho = \rho(T, S, p).$
Model Calculation Process
Initialize
Reading grid, initial states, parameters, etc.
Forcing
Input and interpolation of datasets for forcing model
READYT and READYC
Preproessing of baroclinic, barotropic, themohaline, calculation of density, momentum and diffusion, etc.
BAROTR and BCLINC
Processing of barotropic and baroclinic
TRACER
Processing of themohaline including advective, short wave penetration, vertical and vortex mixing, etc.
ICESNOW and CONVADJ
Diagnosis of sea ice, and convection adjustment
Output
Writing files including monthly data, daily instaneous, restarting, etc.
Time-stepping Loop
Daily Loop
Architecture-portable accelerating
Performance Portability
Kokkos Core
Parallel Execution
Parallel Data Structures
OpenMP
Athread
CUDA
···
Diversified Architectures
CPU
SW
GPU
···
Architecture of SW26010 Pro
CG (Core Group)
MPE
CPEs (8 × 8)
MC (DDR4)
Network Interface
CG Network

Fig. 3: Schematic of a high-level overview of LICOMK++, the architecture of SW26010 Pro, and their relationship.

where $n$ denotes the number of functors. To further enhance the performance of the preset function-matching process, we leveraged Sunway architecture features such as Local Data Memory (LDM) to reduce memory latency and single-instruction, multiple-data (SIMD) vectorization, for accelerated kernel matching.

**Parallel Execution:** Our proposed mapping scheme for distributing parallel executions across CPEs is as follows: Equation 1 calculates the total number of tiles for each loop dimension based on the loop range and tile length.

$$total\_tile = \prod_{n=1}^{num\_dim} \lceil len\_range_n / len\_tile_n \rceil \qquad (1)$$

Where $len\_range\_n$ and $len\_tile\_n$ denote the loop body range and tile length in the $n$-dimensional loop, respectively; these denote the inputs via the unified interface of Kokkos. Equation 2 computes the number of tiles required by each CPE for an ergodic sweep, aiming to achieve a balanced workload distribution across all CPEs as evenly as possible.

$$num\_tile\_per\_cpe = \lceil total\_tile / num\_cpe \rceil \qquad (2)$$

where $num\_cpe$ is the number of CPEs, typically equal to 64, the number of CPEs per core group (CG).

**Memory Management:** The management processing element (MPE) and CPEs of Sunway processors share the same memory space, similar to the unified memory used in CUDA-capable GPUs. In addition, the Sunway processor's memory hierarchy is standard. Consequently, we can apply the Kokkos memory model from the host space without needing to implement a separate device memory space. A unique feature of the Sunway architecture is the direct manipulation of LDM. Although this capability has not been integrated into Kokkos, developers can optimize memory latency by using LDM in two ways: 1) by defining and using local arrays within the functor, and 2) by accessing *View* data pointers via the *View.data* interface and utilizing direct memory access (DMA) functions provided by Athread.

### C. Hotspot Kernel Optimization

*1) Load balancing optimization:* As ocean models incorporate increasingly realistic topography features at higher resolution and larger computational scale, MPI ranks assigned at the sea-land boundary become more susceptible to load imbalances. The *canuto* parameterization calculation is the second most computationally expensive kernel. This kernel is oriented vertically in the downward direction when the Earth's surface is oceanic. As shown in Fig. 4, each MPI rank gathers information regarding the ocean points requiring *canuto* parameterization calculations to address the load imbalance. Subsequently, these MPI ranks partition the workload evenly and distribute the calculations to computing resources.

*2) Architecture-specific strategies:* Kernels exhibiting the highest overhead with very low computation-to-memory access ratio and severely scattered memory access patterns must be optimized using a more flexible, architecture-specific code. For example, the *advection_tracer* function [47] emerged as a critical performance bottleneck in this model. This function implements three-dimensional (3D) stencil computations, operates on numerous data arrays, and exhibits enhanced logical complexity. For the *advection_tracer* executed on a GPU, we leveraged the optimization strategy proposed by Wei et al. [42] in 2023. Whenever the Sunway system is

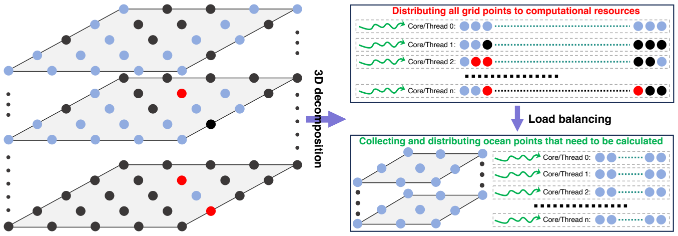


Fig. 4: Load balancing for the *canuto* parameterization calculation on the realistic topography. The ocean points are depicted in blue, the land points are in black, and the ocean points excluded *canuto* parameterization calculations are in red.

used, we adopt a double-buffered technique that leverages the asynchronous mechanism of the Sunway architecture between the CPE workload execution and DMA transfers.

Because there are fundamental differences in memory access among architectures, some algorithms can achieve coalesced memory access on GPUs and discrete memory access on CPU and Sunway systems. We used an architecture-dependent code for optimization. One approach involves leveraging the architecture-specific, low-latency memory, to implement data rearrangement and prefetching.

### D. Communication Optimization

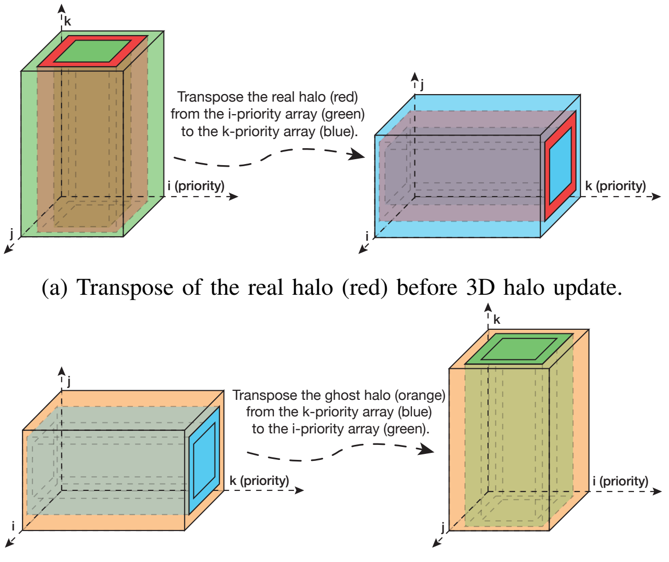
LICOM divides the Earth into horizontal two-dimensional (2D) grid blocks, with each MPI rank handling one block. Each grid block includes the outermost two layers of the ghost halo, a second layer with two layers of the real halo, and internal data. Ghost halo overlaps between neighboring grid blocks. The halo update within the model is facilitated by two components: 1) pack/unpack, and 2) halo exchange.

While most data packing/unpacking operations involve a minimal amount of boundary data, which is associated with a negligible overall cost, specific scenarios arise in polar regions. Herein, packing/unpacking a large volume of data leads to a time complexity of $\mathcal{O}(n)$, where $n$ represents the product of the horizontal resolution and vertical levels of the model. Notably, the cost of pack/unpack operations remains constant and does not benefit from parallelization as the computational scale increases. In addition, the overhead of the halo exchange may tend to increase with growing computational scale. Overall, the halo update process within the model acts as a serial bottleneck according to Amdahl's law [48]. This characteristic, in which the halo update cost grows significantly at increasing model resolution and computational scale, limits the overall scalability of the model.

To address the halo update bottlenecks associated with pack/unpack, we implemented three optimization strategies: 1) we leveraged C++ to rewrite the halo update process. This transition facilitated the introduction of advanced HPC technologies to improve efficiency. 2) We analyzed and optimized the redundant packing/unpacking operations. 3) The Kokkos was employed to accelerate the optimized packing/unpacking routines. Furthermore, we implemented an overlap of the computation-communication strategy to mask the overhead of the halo exchange. Since our heterogeneous systems lack support for GPU-aware MPI technology, we minimize data copying between the host and devices to reduce data movement overhead, adhering to the communication algorithm.

In addition, LICOM employs both 2D and 3D halo updates, with 3D halo updates extending 2D halo updates point-wise in the vertical direction. In 3D halo update, placing the outermost vertical direction as the innermost loop results in substantial data access discontinuity. 3D halo update is a significant performance bottleneck as the vertical levels increase.

We have developed three optimized algorithms for 3D halo update: 1) An efficient halo transpose algorithm that converts the real halo from a horizontal major order array to a vertical major order array, as shown in Fig. 5a; 2) A redesigned approach for 3D communication that priorities the vertical direction; 3) A high-performance transpose operator that transforms the ghost halo from a vertical major order array to a horizontal major order array, illustrated in Fig. 5b. For GPU implementations, we utilize shared memory, while for Sunway processors, we use CPEs along with LDM and SIMD instructions to further optimize the transpose algorithms.


k
i (priority)
j
Transpose the real halo (red)
from the i-priority array (green)
to the k-priority array (blue).
j
k (priority)
i
j
k (priority)
i
Transpose the ghost halo (orange)
from the k-priority array (blue)
to the i-priority array (green).
k
i (priority)
j

(a) Transpose of the real halo (red) before 3D halo update.

(b) Transpose of the ghost halo (orange) after 3D halo update.

Fig. 5: Transpose of the real (a) and ghost (b) halos to optimize 3D halo update. Green and blue denote arrays with horizontal and vertical major ordering, respectively. Red and orange symbolize the real halo and the ghost halo, respectively.

## VI. How Performance Was Measured

### A. Architectural Details

The results discussed in this paper were obtained through experiments conducted on three heterogeneous systems and one ARM architecture-based system. The heterogeneous systems include a single-node NVIDIA GPU workstation, ORISE, and the new-generation Sunway supercomputers, while the ARM-based system is a Huawei Taishan 2280 server.

Key hardware specifications for these computing platforms are detailed in Table II.

TABLE II: Critical hardware configurations of nodes in four computing systems.

| Systems | GPU workstation | ORISE | New Sunway | Taishan server |
|---|---|---|---|---|
| CPUs | 2× Intel Xeon Gold 6240R CPUs | 4-way 8-core x86 CPU | SW26010 Pro processor | 2× Taishan 2280 processors |
| Cores | 48 | 32 | 390 | 128 |
| Host memory | 754 GB | 128 GB | 96 GB | 256 GB |
| Accelerators | 4× Tesla V100 GPUs | 4× HIP-based GPUs | \ | \ |
| Device memory | 128 GB | 64 GB | \ | \ |
| Back-end | CUDA | HIP | Athread | OpenMP |

ORISE is a heterogeneous supercomputer installed at the Computer Network Information Center at the Chinese Academy of Sciences, China. Each computing node is equipped with one CPU and four GPGPU-like accelerators connected by a high-speed network with a bandwidth of 25 GB/s. The CPU features a 4-way, 8-core architecture with an ×86 instruction set, and runs at 2.0 GHz with a memory of 128 GB. The GPGPU-like accelerator, also known as the HIP-based GPU, utilizes HIP as the back-end programming model and can achieve a performance comparable to that of AMD MI60. The CPU and GPUs are interconnected through 32-bit PCIe (Peripheral Component Interconnect Express) buses featuring a DMA with a bandwidth of 16 GB/s.

The new-generation Sunway system is among the largest supercomputers in China. The computational capability of the system is derived from a massive SW26010 Pro processor, which is the latest generation Sunway many-core processor, as shown in Fig. 3 (lower right panel). The computational power of the SW26010 Pro processor is provided by 64 CPEs, also known as slave cores. Each CPE has a 32 kB instruction cache and 256 kB low-latency data memory shared by the LDM and local data cache. Data transfer between the main memory and LDM is achieved using DMA functions. A CPE cluster (8 × 8 CPEs), an MPE (also called the host core), and a memory controller constitute a CG. Each CG contains a 16 GB DDR4 memory and has a 51.2 GB/s bandwidth. The 6 interconnected CGs constitute one SW26010 Pro processor with 390 cores (6 MPEs and 384 CPEs).

### *B. Model Configurations*

Table III presents the four model configurations employed herein; these correspond to the coarse, eddy-resolving, and two kilometer-scale resolution used for performance portability (Section VII-B) and strong scaling evaluations (Section VII-C).

A coarse resolution (approximately 100 km) assesses the performance portability across various architectures using a single node. This resolution is associated with 360 × 218 × 30 grid points, with time steps set to 120, 1440, and 1440 s for the barotropic, baroclinic, and tracer components, respectively. The eddy-resolving case is approximately 10-km and has been extensively adopted for scientific research in ocean modelling. There are 3600 × 2302 × 55 grid points at this resolution, and the time steps for the barotropic, baroclinic, and tracer processes are 9, 180, and 180 s, respectively.

The kilometer-scale resolutions, including an approximate 2-km horizontal resolution with full depth (>10000 m), and an approximate 1-km horizontal resolution, represent a crucial advancement for LICOMK++. LICOMK++ facilitates the exploration of large-scale performance and the analysis of intricate ocean/climate phenomena. The model with 2-km resolution features 18000 × 11511 × 244 grid points, while the model with 1-km resolution has 36000 × 22018 × 80 grid points. In both cases, the time steps are configured to 2, 20, and 20 s for the barotropic, baroclinic, and tracer processes, respectively.

Six problem sizes with horizontal resolution ranging from 10 km to 1 km were utilized to evaluate the weak scalability (Section VII-D) outcomes listed in Table IV. These configurations maintained consistent time steps, vertical layers, and computational processes. The time steps were set to 2, 20, and 20 s for the barotropic, baroclinic, and tracer simulations, respectively. There were 80 vertical depths.

MPI configurations for single-node experiments at 100-km resolution included the following: on the GPU workstation, Fortran employed 48 MPI ranks; simultaneously, Kokkos was allocated 4 MPI ranks per rank bound to 1 GPU. In the ORISE system, Fortran was assigned 32 MPI ranks, with Kokkos utilizing 4 MPI ranks and associating them with 4 HIP-based GPUs. In the Sunway experiment, Fortran and Kokkos operated with 6 MPI ranks. Simultaneously, each rank of the Kokkos code was allocated to 64 CPEs. Regarding the Taishan server, 128 MPI ranks were designated for Fortran without OpenMP threads and 64 MPI ranks for Kokkos, each with 2 OpenMP threads.

Large-scale simulations were performed using the ORISE and new Sunway supercomputers. Regarding the Kokkos code, each node was allocated 4 MPI ranks and 4 HIP-based GPUs on ORISE, whereas for the new Sunway system, the model employed 65 cores (1 MPE and 64 CPEs) per MPI rank.

While several versions of LICOM3 are available in the community [42], [43], none have been adapted to kilometer-scale resolutions, nor has an Athread version been developed. In a separate study [44], we conduct a thorough comparison of Kokkos with raw CUDA, HIP, and OpenMP implementations of LICOM3 at 100-, 10-, and 5-km resolutions. Therefore, the experiments in this study focus exclusively on the latest Kokkos and the existing Fortran version of LICOM3.

### *C. Performance Metrics*

This study used the intrinsic time-to-solution metric SYPD to assess the throughput of the overall application, representing the number of years the model can simulate on one wall-clock day. Thus, a higher SYPD value signifies a higher computational model speed. We calculated the parallel efficiency for strong and weak scaling studies using the wall-clock time of a simulated day. The elapsed wall-clock time was determined by the maximum value across the ranks from the timer for the top-level daily loop, including the simulation and daily memory copies in heterogeneous systems. The times for I/O and initialization tasks, including system configuration, grid

TABLE III: Four configurations of LICOMK++ are employed in the present study.

| | Coarse-resolution | Eddy-resolving | Kilometer-scale | Kilometer-scale |
|---|---|---|---|---|
| Resolution | $\mathcal{O}(100\,km)$ | $\mathcal{O}(10\,km)$ | $\mathcal{O}(2\,km)$ | $\mathcal{O}(1\,km)$ |
| Horizontal grid | 360 × 218 | 3600 × 2302 | 18000 × 11511 | 36000 × 22018 |
| Vertical levels | $30\,\eta$ | $55\,\eta$ | $244\,\eta$ | $80\,\eta$ |
| Time steps | 120/1440/1440 s for barotropic/baroclinic/tracer | 9/180/180 s for barotropic/baroclinic/tracer | 2/20/20 s for barotropic/baroclinic/tracer | 2/20/20 s for barotropic/baroclinic/tracer |

TABLE IV: Six scales used for the weak scalability test. These scales use consistent time steps, vertical layers, and grid point numbers in all tested computational processes.

| Resolution | Grid points | HIP-based GPUs | Sunway cores |
|---|---|---|---|
| 10 km | 3600 × 2302 × 80 | 160 | 404625 |
| 6.66 km | 5400 × 3453 × 80 | 360 | 910780 |
| 5 km | 7200 × 4605 × 80 | 640 | 1608750 |
| 3.33 km | 10800 × 6907 × 80 | 1440 | 3612375 |
| 2 km | 18000 × 11511 × 80 | 4000 | 10042500 |
| 1 km | 36000 × 22018 × 80 | 15360 | 38366250 |

mapping, MPI initialization, and finalization, were excluded. We primarily employed the GPTL and Chrono libraries as timers. GPTL is broadly used in atmospheric, oceanic, and climate models, whereas Chrono provides a high-precision timing library for C++. On the new Sunway system, we use a job-level performance monitoring and analysis toolchain to gather floating-point statistics.

## VII. Performance Results

### A. Model Performance

The oceanic submesoscale processes significantly impact the Earth's system climatology through interaction with oceanic mesoscale motions [3], [5], [49], [50]. The capability for describing submesoscale processes should be a crucial criterion for measuring the performance of a kilometer-scale-resolution OGCM. Herein, we use the Rossby number (Ro), defined as the vertical component of the relative vorticity ($\zeta$) divided by the local Coriolis parameter ($f$), to describe the richness of submesoscale activities [49]. $|Ro| \sim O(1)$ indicates the active submesoscale processes and fully-developed mesoscale motions [51]. From Fig. 6, more submesoscale signals and fine-scale structures emerge at higher horizontal resolution. It is noteworthy that the simulation at 1-km resolution captures a significant part of submesoscale motions (Fig. 6a and Fig. 6d), especially 1) the fine structure of symmetric instability around mesoscale eddies and 2) the "submesoscale soup" that spreads nearly everywhere [52]. The remarkable improvement in submesoscale results at higher resolution demonstrates the scientific significance and application value of developing OGCMs at the kilometer-scale resolution.

### B. Performance Portability Evaluation

The results presented in Fig. 7 indicate that the proposed code can be executed across diverse architectures, achieving efficient computational capabilities in heterogeneous systems.

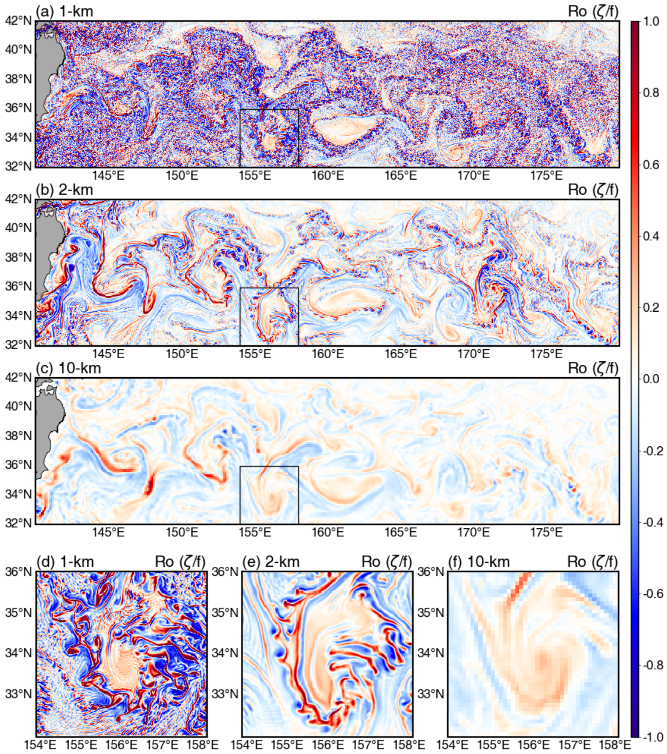


Fig. 6: Snapshots of Rossby number in the Kuroshio Extension region from global (a) 1-, (b) 2-, and (c) 10-km-resolution simulations. (d-f) Respective magnified views of black-box regions in (a-c). The first is from global 1-km-resolution LICOMK++ on the ORISE system. The others are from LICOM3 on the Earth System Numerical Simulation Facility.

Additionally, it maintains performance parity with the original code on the CPUs. LICOMK++ attained 317.73 SYPD by using four V100 GPUs. Operating on four HIP-based GPUs, it achieved a computational speed of 180.56 SYPD. By leveraging 390 cores of the SW26010 Pro, the model reached a throughput of 22.22 SYPD. Moreover, by fully utilizing the computing cores of the Taishan server, a computing speed of 63.01 SYPD was achieved. Using the original LICOM3 as a benchmark, LICOMK++ achieved speed enhancements of up to 7.08 (V100 GPU), 11.42 (HIP-based GPU), 11.45 (SW26010 Pro), and 1.03 (Taishan 2280) times. LICOMK++ offers portability similar to that of its Fortran-based predecessor, LICOM3, which, coupled with significant performance gains, enables LICOMK++ to achieve computational speeds that are faster or considerably faster than LICOM3. Furthermore, using the toolchain provided by Sunway, we achieved 14.12 GFLOPS with LICOMK++ at 100-km resolution on a single SW26010 Pro processor.

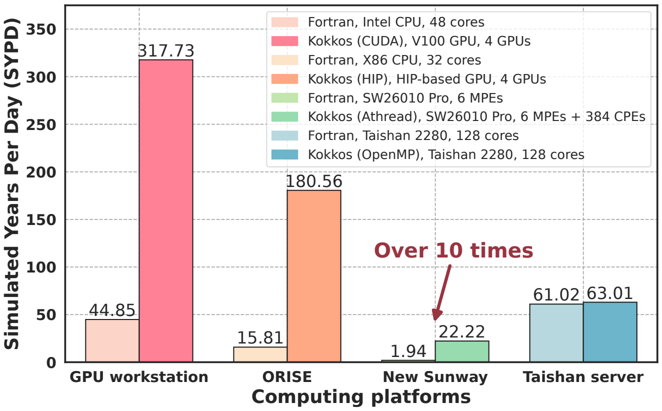


Fig. 7: SYPD metrics computed for different implementations executed on four single-node platforms at the resolution of 100 km.

### C. Strong Scalability

The performances of the models at eddy-resolving (10-km) and kilometer-scale (2- and 1-km) resolution were experimentally evaluated using ORISE supercomputer and the new Sunway supercomputer at different scales. The experimental results are shown in Fig. 8. The optimized version of LICOMK++ running on the Sunway system at nearly full computational scale achieves a speedup of 2.7× over the original version at a resolution of 2-km. The speedup increases to 3.9× at a resolution of 1-km.

The performance evaluations of the eddy-resolving (10-km) model on the ORISE supercomputer utilized 10–250 nodes, whereas the experiments on the new Sunway supercomputer utilized 27–260 nodes. On the ORISE system, Compared with 40 GPUs, LICOMK++ can scale by 25× on 1000 GPUs, a parallel efficiency of 53.7% was achieved. Similarly, in the case of the new Sunway system, an approximately 10-fold increase in the core number (from 10400 to 101400) yielded a parallel efficiency of 77.6%. In the 2-km resolution case, the GPU count increased from 4000 to 16000 on the ORISE supercomputer and yielded a parallel efficiency of 48.8%. In the new Sunway system, the core count increased from > 5 million (5070000) to > 37 million (37440000) and achieved a parallel efficiency of 50.9%. At the global resolution of 1 km, encompassing > 63 billion grid points, LICOMK++ attained a peak performance of 1.701 SYPD with a parallel efficiency of 55.6% using 16000 HIP-based GPUs. In addition, it demonstrated a performance of 1.047 SYPD and a parallel efficiency of 54.8% by utilizing 38366250 cores on the new Sunway supercomputer. Overall, LICOMK++ at the eddy-resolving and kilometer-scale resolutions demonstrate good scalability outcomes in the ORISE and the new Sunway supercomputers. The optimal LICOMK++ of numerical values presented in Fig. 8 are listed in Table V.

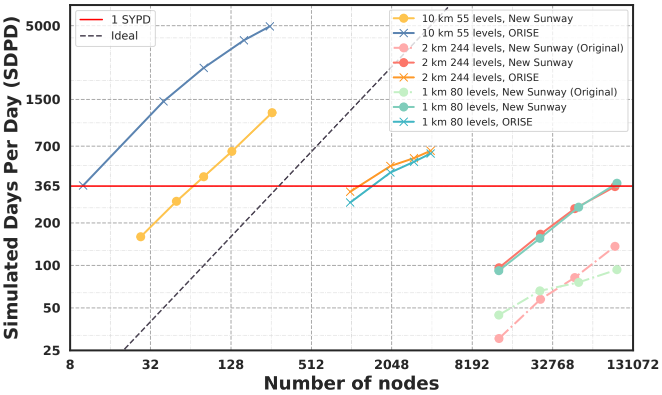


Fig. 8: Strong scaling results of the LICOMK++ at different resolutions without input/output. The optimal LICOMK++ of data are also listed in Table V.

TABLE V: Comparison of the performances of LICOMK++ in different systems and at different resolutions.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10 km | ORISE | Nodes | 10 | 40 | 80 | 160 | 250 |
| | | GPUs | 40 | 160 | 320 | 640 | 1,000 |
| | | SYPD | 1.009 | 3.984 | 6.880 | 10.794 | 13.543 |
| | | Efficiency | 100% | 98.7% | 85.2% | 66.8% | 53.7% |
| 10 km | New Sunway | Nodes | 27 | 50 | 80 | 130 | 260 |
| | | Cores | 10400 | 19500 | 31200 | 50700 | 101400 |
| | | SYPD | 0.437 | 0.780 | 1.165 | 1.761 | 3.312 |
| | | Efficiency | 100% | 95.1% | 88.8% | 82.6% | 77.6% |
| 2 km | ORISE | Nodes | | 1000 | 2000 | 3000 | 4000 |
| | | GPUs | | 4000 | 8000 | 12000 | 16000 |
| | | SYPD | | 0.912 | 1.386 | 1.577 | 1.779 |
| | | Efficiency | | 100% | 76.0% | 57.6% | 48.8% |
| 2 km | New Sunway | Nodes | | 13000 | 26580 | 48000 | 96000 |
| | | Cores | | 5070000 | 10366200 | 18720000 | 37440000 |
| | | SYPD | | 0.264 | 0.456 | 0.692 | 0.992 |
| | | Efficiency | | 100% | 84.5% | 71.1% | 50.9% |
| 1 km | ORISE | Nodes | | 1000 | 2000 | 3000 | 4000 |
| | | GPUs | | 4000 | 8000 | 12000 | 16000 |
| | | SYPD | | 0.765 | 1.248 | 1.486 | **1.701** |
| | | Efficiency | | 100% | 81,6% | 64.8% | **55.6%** |
| 1 km | New Sunway | Nodes | | 12959 | 25920 | 51300 | 98375 |
| | | Cores | | 5053750 | 10108800 | 20007000 | 38366250 |
| | | SYPD | | 0.252 | 0.426 | 0.709 | **1.047** |
| | | Efficiency | | 100% | 84.7% | 71.1% | **54.8%** |

### D. Weak Scalability

Fig. 9 presents the weak scaling results at the horizontal resolution of 10, 6.66, 5, 3.33, 2, and 1 km, which were scaled by more than 95 times. In the ORISE system with 15360 GPUs, a parallel efficiency of 85.6% was achieved. Similarly, the new Sunway system employing 38366250 cores exhibited a parallel efficiency of 91.2%. The results of the weak scalability experiments indicate that LICOMK++ demonstrates good scalability.

The experimental results demonstrate the excellent strong and weak scalability of the model, which is partly attributed to the optimization of the halo update bottleneck. By addressing this critical challenge, we effectively mitigated the performance degradation often associated with an increasing computational scale.

From the perspective of the floating-point performance, the execution of the model on the new Sunway system should be faster than that executed on the ORISE system. However, the opposite was observed in the experimental results. 1) *Memory Access Bottleneck*: The Ocean model required numerous stencil computations involving many arrays, yielding a very low computation-to-memory ratio and severely discontinuous memory access. Additionally, the Sunway architecture's memory access bandwidth was low (SW26010 Pro: Offers a memory bandwidth of 51.2 GB/s for its CPEs and main memory; V100 GPU: Boasts a significantly increased high-bandwidth memory bandwidth of 887.9 GB/s.) and neither the LDM nor the cache could store the vast amounts of data required for stencil computations. 2) *Hotspot dispersion*: Despite the optimization of several time-consuming kernels, the computational load of the ocean model was almost evenly distributed across all kernels, necessitating the fine and complex optimization of each kernel to achieve further performance improvements. 3) *Communication Overhead*: Harnessing greater computational power requires decomposition into large-scale MPI tasks, which substantially increases the communication overhead and results in insufficient workload per core.

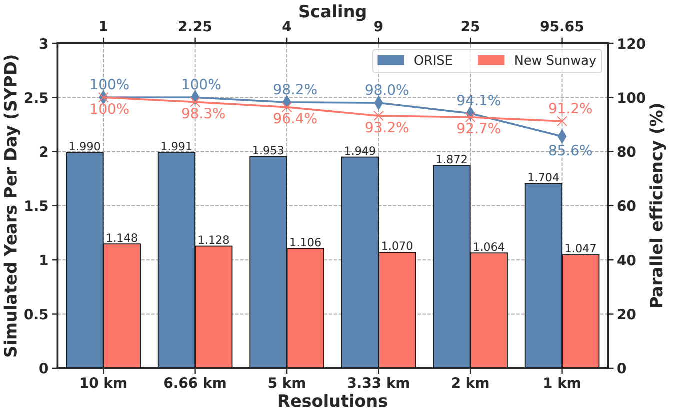


Fig. 9: Weak scaling results for LICOMK++ of six different resolutions from 10 km to 1 km on the ORISE system and the new Sunway system. The configurations of the cases are given in Table IV.

## VIII. IMPLICATIONS

This study enhances Kokkos to support the Sunway architecture, whose supercomputers ranked first in the TOP500 list from 2016 to 2017, and the new-generation Sunway supercomputer is now an exascale system. The public members must familiarize themselves with the Sunway architecture, and it is difficult to port and optimize the code to effectively leverage its powerful computing capacity. This study provides an excellent solution for this challenge. Kokkos's support of the Sunway architecture is a transformative accomplishment for the Kokkos C++ performance portability programming ecosystem. As a result of this study, all TOP500 supercomputers that have held the top rank positions since 2010 have achieved the unification of intranode parallel programming models.

Using Kokkos, LICOMK++ achieved swift deployment on the new Sunway supercomputer and stable scaling of the entire system. A group of optimizations yielded significant performance gains. On the new Sunway system, the model's speed improved from 0.25 SYPD to an impressive 1.05 SYPD. The model exhibited excellent scalability and achieved parallel efficiencies of 54.8% (strong scalability) and 91.2% (weak scalability) while utilizing almost the entire system. Similarly, in the ORISE system, the speed reached 1.70 SYPD with parallel efficiencies of 55.6% (strong scalability) and 85.6% (weak scalability) using 16000 HIP-based GPUs. This exceptional performance underscores the benefits of performance portability technology for the effective use of modern supercomputers.

Achieving fast (beyond 1 SYPD) global realistic OGCMs with kilometric horizontal resolution is vital for simulating and predicting climate changes [53], [54]. LICOMK++ is the first 1-km-resolution global realistic OGCM, achieving $> 1$ SYPD. LICOMK++ sets a new precedent for simulating oceanic meso- and submesoscale phenomena and capturing ocean changes in the abyss, including those in the Challenger Deep. These fast and realistic simulations are urgently needed to predict real-time and accurate oceanic changes that depend highly on mesoscale and submesoscale processes conventionally parameterized in the coarse-resolution OGCMs.

Adapting an OGCM to portable technologies is a key step toward an extreme-scale ESM. The original F90 version LICOM3 cannot easily keep up with the rapid development of new HPC systems. As Kokkos continues to evolve, LICOMK++ is expected to extend its support to more sophisticated architectures. This feature can save considerable time (i.e., 1–2 years) and human resources for adapting models to newer HPC systems. Other similar models that can be borrowed from LICOMK++ include the following: 1) Kokkos enables developers to use almost one-suite code to build, run, and evaluate models on different platforms while maintaining high performance; 2) it makes it easier for the model teams to control the code tree while hiding underlying differences in different computation acceleration hardware. This endeavor holds considerable value for models related to the ocean, atmosphere, climate, and Earth system.

The portability of LICOMK++ provides a fundamental framework support for achieving interactions between coarse- and fine-grained simulations executed on heterogeneous supercomputing clusters at different scales. Given the existing HPC machines with different structures and scales, the appropriate selection of the platform according to the simulation requirements and scale can significantly improve resource utilization and reduce the cost of the experiment. Multiscale oceanic experiments on various platforms provide an encouraging paradigm for improving computing power utilization through collaborations across geographically distributed heterogeneous HPC systems. In addition, performance portability technologies, similar to cross-platform mobile app development, are also important for computing power network because they can support the flexible scheduling of applications across regions, architectures, and operational entities.

**The LICOMK++ introduced in this study is the first performance-portable global OGCM on diversified supercomputer architectures and the first global 1-km-resolution realistic OGCM that overcomes the 1-SYPD performance**

**challenge and generates scientific results with superb skills.** It represents a significant advancement over existing OGCMs. The leap in model resolution offers potential opportunities to deepen our understanding of ocean dynamics, improve the accuracy of weather forecasts, and refine the estimates of future climate change trajectories.

Some methods can be used to improve the speed of LICOMK++, such as the introduction of mixed precision and artificial intelligence. In addition, considering the realistic ocean simulation at the resolution of 1 km necessitates the improvement of the I/O capability, storage capacity, and computing power for the development of future supercomputers.

## Acknowledgment

Supported by the Strategic Priority Research Program of Chinese Academy of Sciences (XDB0500303, XDB0500101), National Natural Science Foundation of China (92358302, 41931183), the Major Key Project of PCL (PCL2023A09), the National Key R&D Program for Developing Basic Sciences (2022YFC3104802), the Tai Shan Scholar Program (tstp20231237), Laoshan Laboratory (LSKJ202300301) and Huawei Technologies Co., Ltd. The numerical calculations in this study were carried out on the ORISE Supercomputer, the new Sunway Supercomputer, and the machine of the National Key Scientific and Technological Infrastructure project "Earth System Science Numerical Simulator Facility" (EarthLab).

## References

[1] V. Masson-Delmotte, P. Zhai, A. Pirani, S. L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis *et al.*, "Climate change 2021: the physical science basis," *Contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change*, vol. 2, no. 1, p. 2391, 2021.

[2] Z. Li, M. H. England, and S. Groeskamp, "Recent acceleration in global ocean heat accumulation by mode and intermediate waters," *Nature Communications*, vol. 14, no. 1, p. 6888, 2023.

[3] Z. Su, J. Wang, P. Klein, A. F. Thompson, and D. Menemenlis, "Ocean submesoscales as a key component of the global heat budget," *Nature communications*, vol. 9, no. 1, p. 775, 2018.

[4] N. Beech, T. Rackow, T. Semmler, S. Danilov, Q. Wang, and T. Jung, "Long-term evolution of ocean eddy activity in a warming world," *Nature climate change*, vol. 12, no. 10, pp. 910–917, 2022.

[5] S. Wang, Z. Jing, L. Wu, W. Cai, P. Chang, H. Wang, T. Geng, G. Danabasoglu, Z. Chen, X. Ma *et al.*, "El Niño/Southern Oscillation inhibited by submesoscale ocean eddies," *Nature Geoscience*, vol. 15, no. 2, pp. 112–117, 2022.

[6] H. T. Hewitt, M. Roberts, P. Mathiot, A. Biastoch, E. Blockley, E. P. Chassignet, B. Fox-Kemper, P. Hyder, D. P. Marshall, E. Popova *et al.*, "Resolving and parameterising the ocean mesoscale in earth system models," *Current Climate Change Reports*, vol. 6, pp. 137–152, 2020.

[7] D. B. Chelton, M. G. Schlax, and R. M. Samelson, "Global observations of nonlinear mesoscale eddies," *Progress in oceanography*, vol. 91, no. 2, pp. 167–216, 2011.

[8] K. Stewart, A. M. Hogg, S. Griffies, A. Heerdegen, M. Ward, P. Spence, and M. H. England, "Vertical resolution of baroclinic modes in global ocean models," *Ocean Modelling*, vol. 113, pp. 50–65, 2017.

[9] R. Hallberg, "Using a resolution function to regulate parameterizations of oceanic mesoscale eddy effects," *Ocean Modelling*, vol. 72, pp. 92–103, 2013.

[10] P. Bauer, P. D. Dueben, T. Hoefler, T. Quintino, T. C. Schulthess, and N. P. Wedi, "The digital revolution of Earth-system science," *Nature Computational Science*, vol. 1, no. 2, pp. 104–113, 2021.

[11] C. R. Trott, D. Lebrun-Grandié, D. Arndt, J. Ciesko, V. Dang, N. Ellingwood, R. Gayatri, E. Harvey, D. S. Hollman, D. Ibanez *et al.*, "Kokkos 3: Programming model extensions for the exascale era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 4, pp. 805–817, 2021.

[12] H. C. Edwards, C. R. Trott, and D. Sunderland, "Kokkos: Enabling manycore performance portability through polymorphic memory access patterns," *Journal of parallel and distributed computing*, vol. 74, no. 12, pp. 3202–3216, 2014.

[13] T. M. Evans, A. Siegel, E. W. Draeger, J. Deslippe, M. M. Francois, T. C. Germann, W. E. Hart, and D. F. Martin, "A survey of software implementations used by application codes in the Exascale Computing Project," *The International Journal of High Performance Computing Applications*, vol. 36, no. 1, pp. 5–12, 2022.

[14] M. Taylor, P. M. Caldwell, L. Bertagna, C. Clevenger, A. Donahue, J. Foucar, O. Guba, B. Hillman, N. Keen, J. Krishna *et al.*, "The simple cloud-resolving E3SM atmosphere model running on the Frontier exascale system," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023, pp. 1–11.

[15] A. S. Donahue, P. M. Caldwell, L. Bertagna, H. Beydoun, P. A. Bogenschutz, A. Bradley, T. C. Clevenger, J. G. Foucar, J.-C. Golaz, O. Guba *et al.*, "To exascale and beyond–the simple cloud-resolving E3SM atmosphere model (SCREAM), a performance portable global atmosphere model for cloud-resolving scales," *Authorea Preprints*, 2024.

[16] A. Khan, H. Sim, S. S. Vazhkudai, A. R. Butt, and Y. Kim, "An analysis of system balance and architectural trends based on top500 supercomputers," in *The International Conference on High Performance Computing in Asia-Pacific Region*, 2021, pp. 11–22.

[17] W. Wan, L. Gan, W. Wang, Z. Yin, H. Tian, Z. Zhang, Y. Wang, M. Hua, X. Liu, S. Xiang *et al.*, "7-pflops extreme scale earthquake simulation with crossing multi-faults and topography on Sunway," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023, pp. 1–15.

[18] S. Huang, J. Chen, Z. Zhang, X. Hao, J. Gu, H. An, C. Zhao, Y. Hu, Z. Wang, L. Chen *et al.*, "Establishing a modeling system in 3-km horizontal resolution for global atmospheric circulation triggered by submarine volcanic eruptions with 400 billion smoothed particle hydrodynamics," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023, pp. 1–12.

[19] W. Hu, H. An, Z. Guo, Q. Jiang, X. Qin, J. Chen, W. Jia, C. Yang, Z. Luo, J. Li *et al.*, "2.5 million-atom ab initio electronic-structure simulation of complex metallic heterostructures with DGDFT," in *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2022, pp. 1–13.

[20] Z. Xuehong and L. Xinzhong, "A numerical world ocean general circulation model," *Advances in atmospheric sciences*, vol. 6, no. 1, pp. 44–61, 1989.

[21] Y. Yu, S. Tang, H. Liu, P. Lin, and X. Li, "Development and evaluation of the dynamic framework of an ocean general circulation model with arbitrary orthogonal curvilinear coordinate," *Chinese Journal of Atmospheric Sciences*, vol. 42, no. 4, pp. 877–889, 2018.

[22] Y. Li, H. Liu, M. Ding, P. Lin, Z. Yu, Y. Yu, Y. Meng, Y. Li, X. Jian, J. Jiang *et al.*, "Eddy-resolving simulation of CAS-LICOM3 for phase 2 of the ocean model intercomparison project," *Advances in Atmospheric Sciences*, vol. 37, pp. 1067–1080, 2020.

[23] P. Lin, Z. Yu, H. Liu, Y. Yu, Y. Li, J. Jiang, W. Xue, K. Chen, Q. Yang, B. Zhao *et al.*, "LICOM model datasets for the CMIP6 ocean model intercomparison project," *Advances in Atmospheric Sciences*, vol. 37, pp. 239–249, 2020.

[24] H. Tsujino, L. S. Urakawa, S. M. Griffies, G. Danabasoglu, A. J. Adcroft, A. E. Amaral, T. Arsouze, M. Bentsen, R. Bernardello, C. W. Böning *et al.*, "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*, vol. 13, no. 8, pp. 3643–3708, 2020.

[25] E. P. Chassignet, S. G. Yeager, B. Fox-Kemper, A. Bozec, F. Castruccio, G. Danabasoglu, W. M. Kim, N. Koldunov, Y. Li, P. Lin *et al.*, "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 Discussions*, vol. 2020, pp. 1–58, 2020.

[26] A. M. Treguier, C. de Boyer Montégut, A. Bozec, E. P. Chassignet, B. Fox-Kemper, A. McC. Hogg, D. Iovino, A. E. Kiss, J. Le Sommer, Y. Li *et al.*, "The mixed-layer depth in the ocean model intercomparison project (OMIP): impact of resolving mesoscale eddies," *Geoscientific Model Development*, vol. 16, no. 13, pp. 3849–3872, 2023.

[27] Q. Wang, Q. Shu, A. Bozec, E. P. Chassignet, P. G. Fogli, B. Fox-Kemper, A. M. Hogg, D. Iovino, A. E. Kiss, N. Koldunov *et al.*, "Impact of high resolution on arctic ocean simulations in ocean model intercomparison project phase 2 (OMIP-2)," *Geoscientific Model Development Discussions*, vol. 2023, pp. 1–46, 2023.

[28] J. Mak, P. Choboter, and C. Lupo, "Numerical ocean modeling and simulation with CUDA," in *OCEANS'11 MTS/IEEE KONA*, 2011, pp. 1–6.

[29] S. Xu, X. Huang, L.-Y. Oey, F. Xu, H. Fu, Y. Zhang, and G. Yang, "POM. gpu-v1. 0: a GPU-based princeton ocean model," *Geoscientific Model Development*, vol. 8, no. 9, pp. 2815–2827, 2015.

[30] T. Aketh, S. Vadhiyar, P. Vinayachandran, and R. Nanjundiah, "High performance horizontal diffusion calculations in ocean models on Intel® Xeon Phi™ coprocessor systems," in *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*. IEEE, 2016, pp. 203–211.

[31] T. Yamagishi and Y. Matsumura, "GPU acceleration of a non-hydrostatic ocean model with a multigrid Poisson/Helmholtz solver," *Procedia Computer Science*, vol. 80, pp. 1658–1669, 2016.

[32] J. Jiang, P. Lin, J. Wang, H. Liu, X. Chi, H. Hao, Y. Wang, W. Wang, and L. Zhang, "Porting LASG/IAP climate system ocean model to GPUs using OpenAcc," *IEEE Access*, vol. 7, pp. 154 490–154 501, 2019.

[33] B. Fox-Kemper, A. Adcroft, C. W. Böning, E. P. Chassignet, E. Curchitser, G. Danabasoglu, C. Eden, M. H. England, R. Gerdes, R. J. Greatbatch *et al.*, "Challenges and prospects in ocean circulation models," *Frontiers in Marine Science*, vol. 6, p. 65, 2019.

[34] Y. Zeng, L. Wang, J. Zhang, G. Zhu, Y. Zhuang, and Q. Guo, "Redistributing and optimizing high-resolution ocean model POP2 to million Sunway cores," in *Algorithms and Architectures for Parallel Processing: 20th International Conference, ICA3PP 2020, New York City, NY, USA, October 2–4, 2020, Proceedings, Part I 20*. Springer, 2020, pp. 275–289.

[35] D. Häfner, R. Nuterman, and M. Jochum, "Fast, cheap, and turbulent—global ocean modeling with GPU acceleration in python," *Journal of Advances in Modeling Earth Systems*, vol. 13, no. 12, p. e2021MS002717, 2021.

[36] Y. Ye, Z. Song, S. Zhou, Y. Liu, Q. Shu, B. Wang, W. Liu, F. Qiao, and L. Wang, "swNEMO_v4.0: an ocean model NEMO for the next generation Sunway supercomputer," *Geoscientific Model Development Discussions*, vol. 2022, pp. 1–29, 2022.

[37] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, Sep. 2017.

[38] S. Silvestri, G. Wagner, C. Hill, M. R. Ardakani, J. Blaschke, J.-M. Campin, V. Churavy, N. Constantinou, A. Edelman, J. Marshall, A. Ramadhan, A. Souza, and R. Ferrari, "Oceananigans.jl: A model that achieves breakthrough resolution, memory and energy efficiency in global ocean simulations," 2023.

[39] S. Silvestri, G. L. Wagner, N. C. Constantinou, C. N. Hill, J.-M. Campin, A. N. Souza, S. Bishnu, V. Churavy, J. C. Marshall, and R. Ferrari, "A GPU-based ocean dynamical core for routine mesoscale-resolving climate simulations," *Authorea Preprints*, 2024.

[40] L. Bertagna, M. Deakin, O. Guba, D. Sunderland, A. M. Bradley, I. K. Tezaur, M. A. Taylor, and A. G. Salinger, "HOMMEXX 1.0: a performance-portable atmospheric dynamical core for the Energy Exascale Earth System Model," *Geoscientific Model Development*, vol. 12, no. 4, pp. 1423–1441, 2019.

[41] L. Bertagna, O. Guba, M. A. Taylor, J. G. Foucar, J. Larkin, A. M. Bradley, S. Rajamanickam, and A. G. Salinger, "A performance-portable nonhydrostatic atmospheric dycore for the energy exascale earth system model running at cloud-resolving resolutions," in *Sc20: International conference for high performance computing, networking, storage and analysis*. IEEE, 2020, pp. 1–14.

[42] J. Wei, J. Jiang, H. Liu, F. Zhang, P. Lin, P. Wang, Y. Yu, X. Chi, L. Zhao, M. Ding *et al.*, "LICOM3-CUDA: A GPU version of LASG/IAP climate system ocean model version 3 based on CUDA," *The Journal of Supercomputing*, pp. 1–31, 2023.

[43] P. Wang, J. Jiang, P. Lin, M. Ding, J. Wei, F. Zhang, L. Zhao, Y. Li, Z. Yu, W. Zheng *et al.*, "The GPU version of LASG/IAP climate system ocean model version 3 (LICOM3) under the heterogeneous-compute interface for portability (HIP) framework and its large-scale application," *Geoscientific Model Development*, vol. 14, no. 5, pp. 2781–2799, 2021.

[44] J. Wei, P. Lin, J. Jiang, H. Liu, L. Zhao, Y. Zhang, X. Han, F. Zhang, J. Huang, Y. Wang *et al.*, "Accelerating LASG/IAP climate system ocean model version 3 for performance portability using Kokkos," *Future Generation Computer Systems*, 2024.

[45] W.-Y. Huang, B. Wang, Y.-Q. Yu, and L.-J. Li, "Improvements in licom2. part i: Vertical mixing," *Journal of Atmospheric and Oceanic Technology*, vol. 31, no. 2, pp. 531–544, 2014.

[46] V. Canuto, A. Howard, Y. Cheng, C. J. Muller, A. Leboissetier, and S. Jayne, "Ocean turbulence, iii: New giss vertical mixing scheme," *Ocean Modelling*, vol. 34, no. 3-4, pp. 70–91, 2010.

[47] Y. Rucong, "A two—step shape—preserving advection scheme," *Advances in Atmospheric Sciences*, vol. 11, no. 4, pp. 479–490, 1994.

[48] J. L. Gustafson, "Reevaluating amdahl's law," *Communications of the ACM*, vol. 31, no. 5, pp. 532–533, 1988.

[49] R. Schubert, J. Gula, R. J. Greatbatch, B. Baschek, and A. Biastoch, "The submesoscale kinetic energy cascade: Mesoscale absorption of submesoscale mixed layer eddies and frontal downscale fluxes," *Journal of Physical Oceanography*, vol. 50, no. 9, pp. 2573–2589, 2020.

[50] J. R. Taylor and A. F. Thompson, "Submesoscale dynamics in the upper ocean," *Annual Review of Fluid Mechanics*, vol. 55, no. 1, pp. 103–127, 2023.

[51] J. Xie, X. Wang, H. Liu, P. Lin, J. Yu, and Z. Yu, "Isom 1.0: A fully mesoscale-resolving idealized southern ocean model and the diversity of multiscale eddy interactions," *Geoscientific Model Development Discussions*, vol. 2024, pp. 1–31, 2024.

[52] J. McWilliams, "Submesoscale currents in the ocean," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 472, no. 2189, p. 0117, 2016.

[53] P. D. Dueben, N. Wedi, S. Saarinen, and C. Zeman, "Global simulations of the atmosphere at 1.45 km grid-spacing with the integrated forecasting system," *Journal of the Meteorological Society of Japan. Ser. II*, vol. 98, no. 3, pp. 551–572, 2020.

[54] P. Neumann, P. Düben, P. Adamidis, P. Bauer, M. Brück, L. Kornblueh, D. Klocke, B. Stevens, N. Wedi, and J. Biercamp, "Assessing the scales in numerical weather and climate predictions: will exascale be the rescue?" *Philosophical Transactions of the Royal Society A*, vol. 377, no. 2142, p. 20180148, 2019.