2014 IEEE International Conference on High Performance Computing and Communications (HPCC), 2014 IEEE 6th International Symposium on Cyberspace Safety and Security (CSS) and 2014 IEEE 11th International Conference on Embedded Software and Systems (ICESS)

CESMTuner: An Auto-Tuning Framework for the Community Earth System Model

Ding Nan*†‡, Xue Wei*†‡, Ji Xu*†‡, Xu Haoyu*† Song Zhenya§
*Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China
†Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China
‡Ministry of Education Key Laboratory for Earth System Modeling,
and Center for Earth System Science, Tsinghua University, Beijing 100084, China
§First Institute of Oceanography, SOA , Qingdao 266061, China

***Abstract*—The growing scientific demands of climate predication and climate projection have promoted to manage the computational resources of climate model rationally. The Community Earth System Model (CESM) is one of the state-of-the-art and the most widely-used coupled models for simulating the earth system. Although considerable effort has been put to improve the scalability of single component, CESM is still struggling with the poor performance due to load balance across components. To solve this problem, an easy-used and easy-ported auto-tuning framework named CESMTuner is proposed in this paper. It targets to reduce the time consumed of CESM as much as possible by looking for the optimal process configuration. In which, a novel process layout searching algorithm is presented that can look for the optimal process count of each component as well as the best process layout across components simultaneously. Moreover, a lightweight and accurate performance model is built to reduce searching overhead effectively. With the evaluation over TianHe-1A, CESMTuner can achieve 58.49% performance improvement compared to the widely-used sequential process layout and achieve 38.23% performance improvement compared to the heuristic branch and bound algorithm based on the performance model of simply fitting each component's runtime.**

Keywords: auto-tuning; CESM; load balance; processor allocation; performance prediction

I. Introduction

As the climate change has become a pivotal topic in both industry and science, coupled models have been developed into an indispensable tool for climate and projection of anthropogenic activities on the earth climate. The challenge of climate simulation is due to that it involves a large number of physical processes interacting over a large range of space and time scales. The finer grid mesh and larger number of physical processes with more complicated theories, which leads to significantly increased demand for computing power. Especially it always runs for long-term simulation, typically decades, centuries, and sometimes even millennium.

The Community Earth System Model (CESM)[1] is one of the state-of-the-art and the most widely-used climate models, which is developed and maintained for over nearly 30 years by the National Center for Atmospheric Research (NCAR). It has been regarded as one of the killer applications of the modern supercomputers, such as TITAN, TianHe-1A[2]. Moreover, it is one of the most popular high performance applications ported to both highend system and local clusters all over the world. Now the latest version of CESM is 1.2.0, it normally contains seven components on potentially different grids of atmospheric general circulation model, ocean general circulation model, land surface model, sea ice model, land ice model, river transport model and ocean wave model, which exchange boundary data with each other via a coupler component. According to the report from National Energy Research Scientific Computing Center(NERSC) [3], CESM has been awarded 30M hours for anticipating the climate change research in 2013, and the estimated need will increase to 800-1000M hours with the scientific simulations for the resolution changing from 1 to 0.25 degree in 2017.

So far, a lot of works have been done targeting the scalability of stand alone component. Worley[4] and Alexeev[5] presented a comprehensive theory analysis of the performance optimization of each component from a practical view. Tony Craig *et al.* analyzed the factors of computational performance of CICE [6]. Dennis *et al.* studied the scalability of the atmosphere component with high resolution[7] and demonstrate the performance of the physical processes in different parameter configurations under different load balancing strategies[8]. They also conducted the scalability researches on the scalability of different dynamical cores in the atmosphere model[9][10]. As to POP, there were quite a lot of work on analyzing the impacts of grid partitioning and the corresponding scalability[11][12][13][14].

The poor performance of CESM lies in the complex interaction between components and their diverse computational characteristics. Only increasing the process count can not satisfy the performance requirements. The experiments show 36.8% performance improvement was achieved by rearranging process layout across components from a sequential run to make the atmosphere, land and ice components shared the same process set (as shown in Figure 1). Tuning CESM by manpower in such case is unrealistic. According to the widely-used of CESM and a variety of potential applications with CESM, it is better to design and implement an easy-used and effective tool, which can help the user community improving the performance of CESM. However, it is not easy to get the optimum process configuration effectively in practice. In summary, the main challenges of tuning CESM include:

- ***Huge searching space***. The diverse combinations of both process layout across components and process count

978-1-4799-6123-8/14 $31.00 © 2014 IEEE
DOI 10.1109/HPCC.2014.51

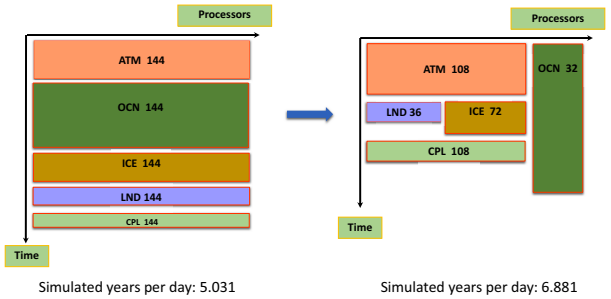


Fig. 1: Performance improvement by rearranging the component layout.

contribute to the huge searching space, which challenges the design of efficient optimization algorithm.

- ***High overhead of CESM runs***. There usually take days and even months for one CESM run. Sampling-based optimization algorithm will suffer from the high overhead of CESM runs. We need to build a lightweight and effective performance model, emulating the performance behavior of CESM on certain platform.
- ***Difficulty of tool development***: Effectively profiling and analyzing complex computation and communication behaviors bring the challenges of designing and implementing an easy-used and easy-ported auto-tuning tool.

In this paper, we propose an easy-used and easy-ported auto-tuning framework named CESMTuner, which has been integrated into the script system of CESM. In CESMTuner, a novel process layout search algorithm is presented to solve the mixed integer nonlinear programming problem arises from searching the best process configuration. Moreover, a lightweight performance model is proposed to quickly and accurately predict the running time of each CESM component by quantifying the computation and communication performance behaviors of the important kernels in each component. In summary, the main contributions of this work are:

- Building an easy-used and easy-ported auto-tuning framework for searching the best process configuration of CESM. The auto-tuning tool has been integrated into the existing script system and can be used in different applications of CESM. Even an inexperienced CESM user can easily use this tool to improve the computational performance of CESM. We get over 58% performance improvement compared to the widely-used sequential process layout and achieve 38.23% performance improvement compared to the heuristic branch and bound algorithm [15] on TianHe-1A.
- An accurate and lightweight performance model for CESM components. It avoids the long-term CESM simulations while keeping high predicting accuracy. Compared to the performance model used in[15], our performance model focuses on predicting the running times of different components and takes the computation and communication of the important kernels in each component into account separately which keeps the model simplicity while getting better accuracy of prediction. With the evaluations using TianHe-1A and local cluster, the model error can be reduced to 10% compared to the performance model of simply fitting each component's runtime [15] with the profiling overhead is only of 8% in total.
- A novel process layout search algorithm which reduces the searching overhead largely. We take the optimal process count of each component as well as the best process layout across components into account simultaneously and use the improved branch and bound ideology method to further enhance the searching efficiency. The overhead of the searching strategy is less than 1% on TianHe-1A.

The rest of this paper is organized as follows. Section II provides an overview of CESM and the corresponding computation and communication characteristics of each component. In Section III, we introduce the process layout searching algorithm, the performance modeling of CESM components and the implementation of CESMTuner. Following a brief introduction of TianHe-1A and TH-HPCA, and the experiment results are presented as well as the analysis of the performance results in Section IV. Section V is a summary of the related work in performance tuning and performance modeling. Section VI concludes the paper and presents the future work.

II. The analysis of CESM

CESM is a complex software comprised of a system of multi-geophysical components, which periodically exchange two-dimensional boundary in the coupler. It consists of the atmosphere,ocean, land surface, sea ice, river transport, ocean wave and land ice and the coupler. Except the land ice and ocean wave are stub components, they all support to execute in dynamic model or data model which can satisfy with the different research purposes.The components and the coupler run simultaneously as a multi-program-multi-data (MPMD) message passing system, using MPI to exchange data which always exhibits irregular computation loads and communication patterns. Achieving maximum possible performance on CESM is more challenging than ever due to the increasing complexity of MPMD parallel scientific program and its interaction with the underlying hardware. We conducted the experiments using CESM1.2.0 coupling with the Parallel Ocean Program version 2 (POP2.0) [16], the Community Atmosphere Model Version 5(CAM5.0) [17], the Community Land Model Version 4 (CLM4.0) [18], the Los Alamos sea ice model version 4 (CICE4.0) [19] and the river transport model (RTM) [20]. A new WAV model is introduced to CESM1.2 with initial support as the stub and dead versions.

The atmosphere component is characterized by two phases: the dynamics, which expresses the evolutionary equations for the atmospheric flow, and the physics which approximates subgrid-scale phenomena. These two phases are executed in turn during each simulation timestep. In the dynamics, update halo and computation of integration are the main cost. To

balance the computational load as well as minimizing update halo cost, space-filling curve is used to map the basic elements from finite-volume grid to the processor. In the physics, the computation cost occupies the entire cost. Each timestemp of POP is divided into two phases: baroclinic and barotropic. The baroclinic phase utilizes an explicit time integration method for the three-dimensional fluid equations which is a typical CPU-bound program. The barotropic phase uses preconditioned conjugate gradient method to solve the two-dimensional surface pressure which contains a large amount of update-halo operation and global communication MPI_Allreduce. CLM is a single column model with nested subgrid hierarchy whose grid cells are composed of multiple landunits, each has multiple snow/soil with multiple plant functional types. The grid columns are grouped into blocks of nearly equal computational cost and these blocks are subsequently assigned to MPI processes. Each process has only one block with MPI-only parallelism. A two-dimensional grid in horizontal is the representation to CICE, and the vertical direction is on behalf of sea ice thickness. Its computational cost is mainly manifested in the constantly grid changing both spatially and temporally over a climate simulation which also bring the load imbalance. While CICE4.0 applies a rake algorithm to improved load balance across processors, and redistribution based on space-filling curves.

III. APPROACH AND IMPLEMENTATION

To construct the auto-tuning framework CESMTuner, we have to solve the mixed integer nonlinear programming(MINLP) problem as below.

$$\min_{p \le P} T(component, layout, process\_count) \qquad (1)$$

where P represents the maximum number of computing nodes for searching, function T(.) is the running time of the whole CESM along with the critical path, which can be determined by both process layouts across components and process count for each component. The *component*, *layout* and *process count* are the decision variables, where *process count* is required to be integer valued. Due to complex interaction between component code and the underlying hardware, the running time of each component is a nonlinear function of the process count. Furthermore, there are a large number of possible process layouts across components, which make the problem more difficult to solve than ever. Such problem is not feasible to solve with the traditional optimization method for huge searching space. Different from the traditional optimization method enumerates all possible solutions and then determines which one is the best, we use depth first search (DFS) method combining with branch and bound ideology method to solve the MINLP. Moreover, we have two choices to get the function between running time and the process count of each component. One is to perform a component run with certain process count and to record the corresponding computational time, which is unaffordable because of taking days even months for one run. The other one is to build an efficient and lightweight performance model to guide the load balancing decisions. In this paper, we mainly focus on modeling the key kernels of each component separating the running time into computation part and communication part. Thus it can ensure the accuracy of performance model while can prevent the large overhead of fine-grained performance modeling.

A. The novel process layout search algorithm

Branch and bound ideology algorithm is a typical method to solve the MINLP problem as well as to find optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. In CESMTuner, it is an especially complicated situation that we have to consider both process count of component and process layout at the same time. we propose a novel process layout search algorithm for the best process configuration of CESM, and we use branch and bound algorithm to ensure the efficiency of the optimization process.

As shown in Figure 2, the DFS algorithm begins with the state in which no component has been started, then we try to fill any possible component starting at the beginning time point with all possible number of processes. In the searching node of next level (for example, the left node of level-2 in Figure 2), the end time points of all placed components have been recorded (including the end time point of LND component is recorded as well as the beginning time point in Figure 2) and we try to perform the same placement process with any other component starting from all possible end time points already recorded later than the starting point of last level (for example, the OCN component can be started from the end time point of LND component or the beginning time point in Figure 2). It is noted that the sequence of component placement should be recorded for avoid searching the same process configuration multiple times. By using this method, we can search all the possible process counts and process layouts and get an optimal one.

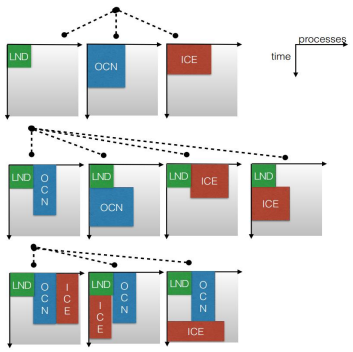


Fig. 2: The overview of the DFS tree. The x-axis is the parallelism and the y-axis is time. Each node represents a series of nodes with the same layout but with the different process count. It is noted that we only take one node as an example to expand in each depth.

Several useful pruning strategies strongly reduce the running

time of our algorithm. Both theoretical pruning method and experimental ways are used:

- ***Time pruning***. Pruning the branches where the time cost in the current state is larger than the shortest time so far.
- ***Searching order***. Adjusting the search sequence of process count in descending order. This can avoid searching the process counts that are smaller ones can meet the conditions in the first pruning strategy.
- ***Scientific dependency between components***. Avoiding the scenarios that the atmosphere model runs parallel with the ICE and LND models for the computational dependency between those components.
- ***Stopping searching when larger process count than the best parallelism***. Preventing the ICE and LND components from using their best performance because the performance of the ICE and LND will get decreased when they have more processes than the best parallelism.
- ***Large stride processor unit for searching***. Stipulating the unit of processes allocated to each component equals to multiple of the processor number in one node when using large parallelism, which can reduce the searching time, as well as can reduce the communication overhead across nodes.

These strategies mentioned above reduce the time consuming greatly. But the novel process layout search algorithm still costs a lot of time when the number of available processors is increased to hundreds of thousand. We further propose a novel process layout searching algorithm, which first uses large stride of process count to get coarse-grain solution and then refine the solution with local optimization.

B. Performance modeling of CESM

The most time-consuming components are the OCN, ATM and ICE components in Figure 3. The LND component and the coupler have a relatively small effect on the overall performance. The RTM nearly has no effect on the CESM timing. Without loss of generality we focus on the relatively detailed performance models of the POP, CAM and ICE components and use a simple modeling method for CLM component. The time of CPL is mainly used for synchronization due to the load imbalance and the computation dependency.

The performance model of curve-fitting presented in [15] got impressive accuracy while keeping the simplicity for FMO computation. However, as shown in Figure 4, the model error can be up to 46% with this method. To solve the problem, we try to build a more accurate and lightweight performance model for CESM components based on the fact that there only are a few computation and communication patterns dominated the performance behaviors of climate models like CESM. We focus on predicting the running time of each components and takes the computation and communication kernels in each component into account separately which keeps the simplicity of the models while getting better accuracy of prediction. For the computation part of POP, CAM and ICE, we use the fitting approach to predicted the computation time, because the

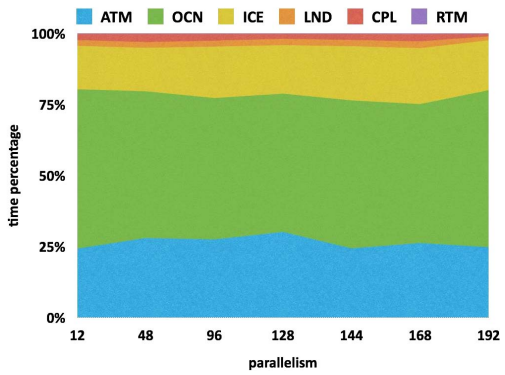


Fig. 3: The time proportion of CESM components(B1850, f19_g16)for one month simulation on TH-HPCA.

computation part have to take charge of the initialization, finalization and the load imbalance, and it can fit the Amdahl's law perfectly. For the communication pattern is mainly composed of update halo which we associated the total communication size with the time cost. Here we don't model the performance of individual messages due to the large overhead and the difficulty on the accurate concurrency analysis.

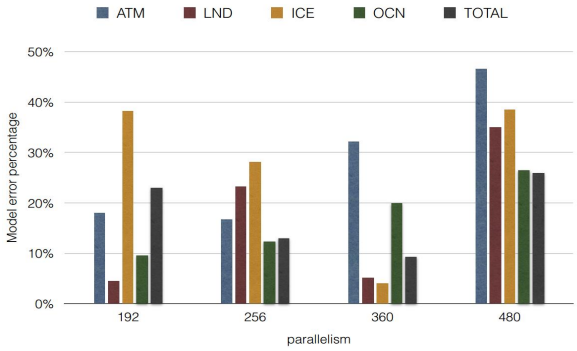


Fig. 4: The error of simple performance model on TianHe-1A. The x-axis is the parallelism and the y-axis is performance model error percentage. The component performance model error is 46% and the total prediction error is 26%.

In POP, the explicit three-dimensional baroclinic solver is the most computationally intensive while the preconditioned conjugate gradient solver for the barotropic process consists of a two-dimensional nine-point stencil operator with many small message update-halo and relatively few floating point operations and global reductions. In CAM, the dynamics contains computation and the irregular neighborhood communication. The main job for physics process is to compute in each independent cell column. By the diversity of regions, each column may assign to different amount of workloads. The ICE acts as a barrier between the polar atmosphere and the ocean to hinder flux exchange including heat, greenhouse gas and so on. The main cost of ICE is the dynamics run. Update-

halo of ICE is the critical part with major irregularity and the computation of ICE only lies in where there is sea ice.

The performance model is generally established by depicting the performance of each computational and communication kernels in Figure 5. Note that the ICE communication has a different tendency with others, it is due to the serious load imbalance.

	Kernel	Computational Characteristic	Communicational Characteristic	Model Description	
				computation	communication
POP	baroclinic	Simply fitting according the Amdahl's law	Update Halo	T= a/p+b *p^c+ d	$T_{halo}=(a*S_{total}+b)/P$ $T_{glo}=a*logP+b$
	barotropic		Update Halo Global communication		
CAM	dynamic		Update Halo		$T=(a*S_{total}+b)/P$
	physics				
CICE	dynamic		Update Halo		$T=a*S_{total}\hat{}b$
CLM			Global communication	T=a/p+b *p^c+ d	

Fig. 5: The performance model of CESM.

The performance model of computation part can be built based on the basis of Amdahl's law. In our performance model, the computation time of different components can be associated with parallelism suggested by [15].

$$T(p) = a/p + b \times p^c + d \tag{2}$$

Where a,b,c and d is the model parameters, p is the process count. As for communication, update halo and the global communication are the two important patterns in CESM. Update halo is the communication pattern widely-use in every component. Global communication is the main part in barotropic of POP which is given priority to MPI_AllReduce. MPI_AllReduce is implemented by a kind of binary tree algorithm to gather the data and to perform the calculation and then to broadcast the final result to each process, whose timing can be fit into the log curve. With the increasing parallelism of strong scaling experiment, message size of each process decreases while the total message count is increasing, it is quite a hard rock for performance modeling because the complexity of both processing messages in queue system and exploiting the concurrency of messages cannot be accurately modeled using the timing of individual message in one process. We try to use an overall modeling approach for update-halo operation to reduce the negative effect of the performance deviation aggregating individual message deliveries. It is worth noting that launching vary processor number in one node shares the similar trend.

C. Implementation of CESMTuner

CESMTuner consists of four modules: *Performance Model Builder*, *Layout Optimization*, *Layout Configuration* and *Time Parser*, which can be integrated into the existing script system of CESM, as shown in Figure 6.

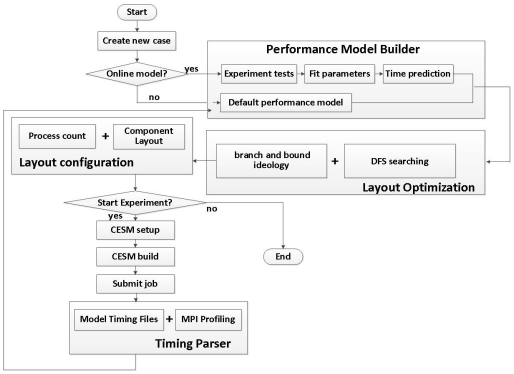


Fig. 6: Overview of CESMTuner

A lightweight performance model is built in *Performance Model Builder* module to predict the running time of each component which supports on-line and off-line use to determine the parameters of the performance model. For the on-line use, CESMTuner will perform at least 12 five-day simulations including four sampling parallelism and each having three repeated runs. The *Layout Optimization* module take charge of auto-tuning CESM process layout which combines the depth first search algorithm and the branch and bound algorithm to ensure the effectiveness of the tool. *Time Parser* module uses mpiP[21] to capture the performance behaviour of communication. At the same time, we use the model timing outputs produced by CESM itself for getting time costs of each component. *Layout Configuration* module reformulates the existing env_mach_pes.xml script using the optimal process configuration we get.To build the performance model, sampling and analysis procedure is conducted automatically in CESMTuner. We use mpiP to trace the communication time as well as the message size. To reduce the overhead of trace collection, we use MPI_Pcontrol to limit the scope of profiling measurements to specific regions of CESM. It will reduce the overhead from 13% to 6.6%. In which, we first locate the code positions need to be measured. Then MPI_Pcontrol(1) and MPI_Pcontrol(0) are put at the beginning and the end of these code segments. To collect profiling from different kernels with the same CESM run, we can classify the communication traces from different kernels using the identifications of File/Address, Line Parent_Funct and MPI_Call.

IV. EXPERIMENTS AND RESULTS

We evaluate CESMTuner over two platforms as shown in Table 1. TH-HPCA is a dedicated-use cluster which only has 16 compute nodes for total 192 cores. As a petascale supercomputer, TianHe-1A features an MPP architecture of hybrid CPU-GPU computing. Note in this work we only use CPU to run CESM. A proprietary high-speed interconnection network, the TH-net, is designed and implemented to enhance the communication capabilities of the system. The topology of the TH-net is an optoelectronic hybrid, hierarchical fat tree. The MPI implementation on the TianHe-1A is customized

	TianHe-1A	TH-HPCA
CPU	2× Intel Xeon X5670 (6 cores)	2× Intel Xeon X5650 (6 cores)
Frequence	2.93GHz	2.67GHz
Complier	icc 11.1	icc 11.0.069
MPI	MPICH2 Version 1.4.1p1	Intel MPI Version 3.2
File System	Lustre	NFS
Network	TH-net	InfiniBand QDR
Node Number	7168	16

TABLE I: Information of the two platforms

to achieve high-bandwidth and low latency data transfers. It is worth noting that TianHe-1A is shared-use supercomputer during CESMTuner evaluation.

We selected B1850_f19_g16 as the performance test case, which is the control run experiment of the IPCC AR5 experiments[22]. B1850 represents all active components for pre-industrial simulation. It is used to get the climate model into a stable state before any historical experiments and projection experiments. The running time of this experiment is expected to be as short as possible.It is also the optimization objective of CESMTuner. f19_g16 represents the atmosphere and land surface models with 1.9x2.5 degree horizontal resolution (144x96 computational gird) are coupled with the ocean and sea ice models using a normal 1 degree horizontal resolution (384x320 computational grid). During the performance experiments on these 2 systems, the output of CESM is turned off due to large disturbance of I/O .

A. Performance Model Validation

The parameters of our performance model can be fitted by profiling the runs using 12, 64, 108 and 128 cores on TH-HPCA, as listed in Figure 7. And the total communication cost of the experiment can be obtained by profiling tool mpiP and the computation cost of the experiment can be got by CESM timing outputs. Figure 8 shows the time predictions according to the performance model for fully sequential one-month simulation of CESM on TH-HPCA. And Figure 9 gives the time predictions for sequential one-month simulation of CESM on TianHe-1A.

			computation				communication	
			a	b	c	d	a	b
OCN	Baroclinic		0.046	-1	-8.312	0	0.274	0.052
	Barotropic	Update halo	-0.003	47.68	0	0	2.807e-010	0.5737
		global					62.61	-1406
ATM	dynamic		0.104	24.45	0	0	1.16e-005	38.98
	physics		180	-1.404	3.667	0		
ICE			1.064e+004	-1.878	0	0	8.982e+014	-2.152
LND			a=85.94	b=-0.69		c=778.7	d=0	

Fig. 7: Parameters of the performance model over TH-HPCA

As shown in Figure 8 and 9, the relative error of the performance model is mostly less than 10% while most of the cases in performance model presented in [15] is larger than 20%, which proves that the proposed performance model can get more accurate performance results than curve-fitting model over whole component perspective. Due to well-reduced profiling mechanism, the overhead of our performance model is less than 7% compared with no-profiling run, which is feasible to use during tuning CESM. We consider the errors of our performance model might lie in two folds. The first one is the ignorance of non-critical parts of CESM component in our performance model which give rise to 5% the model error. The second one is the interaction between software and hardware such as memory and communication contention in the system that is hard to capture accurately in performance model. It is noted that the errors over TianHe-1A is less than those on the dedicated-use TH-HPCA is still under investigation.

We conduct a 500-years experiment on TH-HPCA which takes 45.7 days, while the predicted time using our performance model and profiling strategy is 42.8 days with 6.26% prediction error. This suggests shorter-simulation samples can satisfy the predicition precision. With a lot of experiments, we carry out the performance model by five-day simulations along with mpiP profiling.

B. Performance Tuning of CESM

The performance tuning of CESM over TH-HPCA and TianHe-1A can be found in Figure 10(a) and 10(b). On TH-HPCA, the simulated years per day can increase to 12.19 years with the increase of parallelism while the fully sequential run can only get 10 years with 144 cores and get decrease to 6.3 years with 192 cores. The detailed analysis shows that the optimal process configuration has a reduce of communication time by 62.85%. The similar results can be found on TianHe-1A. When process count equals to 360, the optimum process configuration is different than the other parallelism. This is because LND and ICE components both achieve the best performance with 168 cores, CESMTuner can figure out the feature and make these two components use the maximum process count simultaneously to get better performance.

Figure 10(c) shows the performance improvement by using CESMTuner compared to using the curve-fitting performance model in [15]. Note that a remarkable performance improvement by 38.23% can be achieved with CESMTuner compared to curve-fitting one. Although the process layouts across the components are pretty the same with our results, the process count are different between two methods. It is obviously to note that the accuracy of performance model has a considerable influence on performance tuning of CESM.

V. RELATED WORK

Performance modeling and auto-tuning have been and continue to be of great practical and theoretical importance. A heuristic static load-balancing algorithm applied to the Fragment Molecular Orbital (FMO) Method in reference[15] along with a curve-fitting based performance model. Although

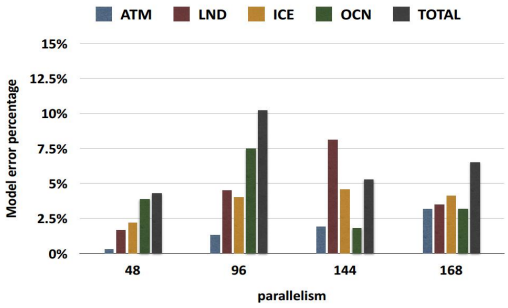


Fig. 8: The model error on TH-HPCA is around 10%.

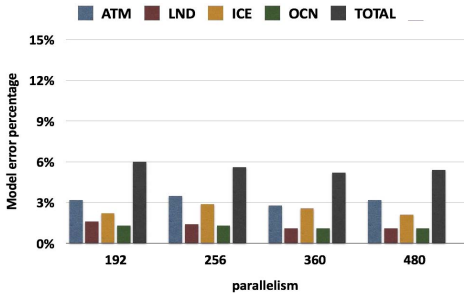
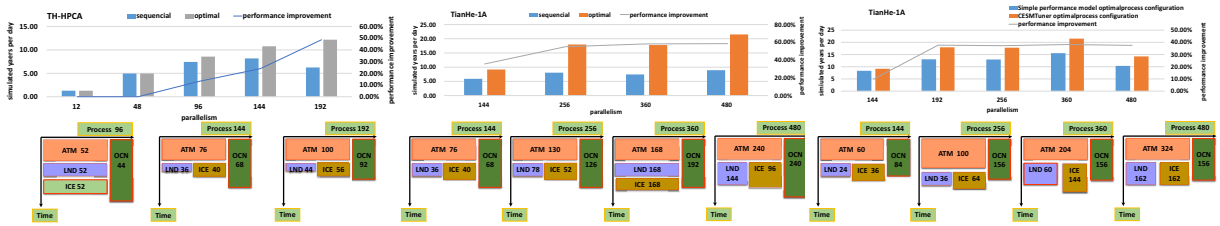


Fig. 9: The model error on TianHe-1A is around 6%.



(a) The simulated years per day of optimal CESM configuration in parallelism on TH-HPCA.

(b) The simulated years per day of optimal CESM configuration in parallelism on TianHe-1A.

(c) The time prediction of simple performance model on TianHe-1A.

Fig. 10: The performance tuning of CESM. (a) is the result on TH-HPCA.With the parallelism increasing, the component layout varies and the performance continues to improve. Such achievement lies in the layout can well overlap the computation and communication time. (b) is the result on TianHe-1A. With the parallelism increasing, the performance improvement get increased from 144 cores to 480 cores. (c) shows that CESMTuner can achieve 38.23% performance improvement compared to the curve-fitting performance model.

their performance model has low overhead, the model accuracy cannot be satisfying due to the lack of considering the kernel characteristics and the hardware and software interaction, the error of their performance model can achieve even up to 46%, which leads to 38.23% performance decrease when using the model for CESM tuning. Furthermore, their heuristic load balance strategy had the limitation of not taking the process layouts across components into account at all while the process layout issue has been proved to be critical in the coupled climate model. A processor allocation strategy was proposed in [7] for ultra-high-resolution CESM while the performance impact of different layouts did not be addressed. D. Kim*et al.* targeted the dynamical load balancing for CCSM [23], while the approach strongly depends on the Malleable MCT environment. There was some work focusing on the performance prediction framework for undeliverable machines in [24] which gives us the inspiration for designing CESMTuner.

While there is a spectrum of different approaches for performance modeling. Tracing data method [25] is often used to predict performance while it always comes with larger overhead. Construct probabilistic model to predict parallel performances by analyzing the runtime distribution of the sequential runs [26] with the limitation that the expectation of the minimum distribution must be able to compute. Kim et al. predicted the potential speedup of serial code by constructing a memory performance model [27], but the communication contention was not addressed, which is not suitable for CESM contains large amount of communication. Kerbyson*et al.* [28] built a fine-grained sample-based performance model for the POP in the fat-tree InfiniBand network. It is not suitable for CESMTuner with the large overhead as well as not practice to use for the whole CESM package.

In CESMTuner, we build a lightweight and accurate performance model of CESM qualifying the computation and communication of important kernels separately and propose the effective process layout search algorithm taking account of both process counts of each component and process layout across components. The evaluations show our final design of CESMTuner can achieve 58.49% performance improvement compared to the sequential process configuration while only introducing the profiling overhead of 8%.

VI. CONCLUSION

CESM is one of the state-of-the-art and the most widely-used coupled models for simulating the earth. Although considerable effort has been put to improve the scalability of single component, it is still struggling with the poor performance due to load imbalance across components. We pre-

sented an easy-used and easy-ported auto-turning framework, CESMTuner, for automatically configuring and submitting the optimum process count of each component and the process layout across the components which strongly reduce the time consumed of CESM run. A lightweight performance model is built by taking the computation and communication of the important kernels in each component into account separately. Our experimental results demonstrate CESMTuner has achieved 58.49% performance improvement comparing with the easy-used sequential process layout and also has a 38.23% performance improvement compared to the process configuration conducted by the curve-fitting performance model.

REFERENCES

[1] "About cesm." [Online]. Available: http://www.cesm.ucar.edu/about/

[2] W. M.Washington, "Scientific grand challenges: Challenges in climate change science and the role of computing at the extreme scale." [Online]. Available: http://www.science.doe.gov/ascr/ProgramDocuments/Docs/ClimateReport.pdf

[3] R. A.Gerber and H. J.Wasserman, "Large scale computing and storage requirements for biological and environ-mental science: Target 2017," DOE Office of Sciennce, Office of Biological and Environmental Research, Office of Advanced Scientific Computing Research, and National Energy Research Scientific Computing Center, Tech. Rep., 2012.

[4] P. H. Worley, A. P. Craig, J. M. Dennis, A. A. Mirin, M. A. Taylor, and M. Vertenstein, "Performance of the community earth system model," in *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*. IEEE, 2011, pp. 1–11.

[5] "G8 ecs: Toward exascale climate simulation name of the presenter."

[6] "Cice performance in cesm." [Online]. Available: http://www.cesm.ucar.edu/

[7] J. M. Dennis, M. Vertenstein, P. H. Worley, A. A. Mirin, A. P. Craig, R. Jacob, and S. Mickelson, "Computational performance of ultra-high-resolution capability in the community earth system model," *International Journal of High Performance Computing Applications*, vol. 26, no. 1, pp. 5–16, 2012.

[8] P. H. Worley and J. B. Drake, "Performance portability in the physical parameterizations of the community atmospheric model," *International Journal of High Performance Computing Applications*, vol. 19, no. 3, pp. 187–201, 2005.

[9] A. A. Mirin and W. B. Sawyer, "A scalable implementation of a finite-volume dynamical core in the community atmosphere model," *International Journal of High Performance Computing Applications*, vol. 19, no. 3, pp. 203–212, 2005.

[10] J. M. Dennis, J. Edwards, K. J. Evans, O. Guba, P. H. Lauritzen, A. A. Mirin, A. St-Cyr, M. A. Taylor, and P. H. Worley, "Cam-se: A scalable spectral element dynamical core for the community atmosphere model," *International Journal of High Performance Computing Applications*, vol. 26, no. 1, pp. 74–89, 2012.

[11] D. J. Kerbyson and P. W. Jones, "A performance model of the parallel ocean program," *International Journal of High Performance Computing Applications*, vol. 19, no. 3, pp. 261–276, 2005.

[12] P. W. Jones, P. H. Worley, Y. Yoshida, J. White, and J. Levesque, "Practical performance portability in the parallel ocean program (pop)," *Concurrency and Computation: Practice and Experience*, vol. 17, no. 10, pp. 1317–1327, 2005.

[13] D. E. Dietrich, "Application of a modified arakawa agrid ocean model having reduced numerical dispersion to the gulf of mexico circulation," *Dynamics of Atmospheres and Oceans*, vol. 27, no. 1, pp. 201–217, 1998.

[14] J.-C. Dutay, J. Bullister, S. Doney, J. Orr, R. Najjar, K. Caldeira, J.-M. Campin, H. Drange, M. Follows, Y. Gao *et al.*, "Evaluation of ocean model ventilation with cfc-11: Comparison of 13 global ocean models," *Ocean Modelling*, vol. 4, no. 2, pp. 89–120, 2002.

[15] Y. Alexeev, A. Mahajan, S. Leyffer, G. Fletcher, and D. G. Fedorov, "Heuristic static load-balancing algorithm applied to the fragment molecular orbital method," in *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for*. IEEE, 2012, pp. 1–13.

[16] R. Smith and P. Gent, "Reference manual for the parallel ocean program (pop), ocean component of the community climate system model (ccsm2. 0 and 3.0)," Technical Report LA-UR-02-2484, Los Alamos National Laboratory, Los Alamos, NM, http://www. ccsm. ucar. edu/models/ccsm3.0/pop, Tech. Rep., 2002.

[17] R. B. Neale, C. Chen, A. Gettelman, P. Lauritzen, S. Park, D. Williamson, A. Conley, R. Garcia, D. Kinnison, J. Lamarque *et al.*, "Description of the ncar community atmosphere model (cam 5.0)," *NCAR Tech. Note NCAR/TN-486+ STR*, 2010.

[18] K. W. Oleson, D. M. Lawrence, B. Gordon, M. G. Flanner, E. Kluzek, J. Peter, S. Levis, S. C. Swenson, E. Thornton, J. Feddema *et al.*, "Technical description of version 4.0 of the community land model (clm)," 2010.

[19] E. Hunke and W. Lipscomb, "Cice: the los alamos sea ice model, documentation and software users manual, version 4.1," 2010.

[20] C. S. E. Group, "Cesm users guide (cesm1.2 release series user's guide)," the National Science Foundation, the Department of Energy, the National Aeronautics and Space Administration, and the University Corporation for Atmospheric Research National Center for Atmospheric Research, http://www.cesm.ucar.edu/models/cesm1.2, Tech. Rep., 2013.

[21] "mpip: Lightweight, scalable mpi profiling." [Online]. Available: http://mpip.sourceforge.net/

[22] "Intergovernmental panel on climate change." [Online]. Available: http://www.ipcc.ch/index.htm

[23] D. Kim, J. W. Larson, and K. Chiu, "Automatic performance prediction for load-balancing coupled models," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013, pp. 410–417.

[24] J. Zhai, W. Chen, and W. Zheng, "Phantom: predicting performance of parallel applications on large-scale parallel machines using a single node," in *ACM Sigplan Notices*, vol. 45, no. 5. ACM, 2010, pp. 305–314.

[25] G. Llort, H. Servat, J. González, J. Giménez, and J. Labarta, "On the usefulness of object tracking techniques in performance analysis," in *Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, p. 29.

[26] C. Truchet, F. Richoux, and P. Codognet, "Prediction of parallel speed-ups for las vegas algorithms," in *Parallel Processing (ICPP), 2013 42nd International Conference on*. IEEE, 2013, pp. 160–169.

[27] M. Kim, P. Kumar, H. Kim, and B. Brett, "Predicting potential speedup of serial code via lightweight profiling and emulations with memory performance model," in *Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*. IEEE, 2012, pp. 1318–1329.

[28] D. J. Kerbyson and P. W. Jones, "A performance model of the parallel ocean program," *International Journal of High Performance Computing Applications*, vol. 19, no. 3, pp. 261–276, 2005.