

Heuristic Static Load-Balancing Algorithm Applied to CESM

Yuri Alexeev¹, Sheri Mickelson¹, Sven Leyffer¹, Robert Jacob¹, Anthony Craig²

¹Argonne National Laboratory, Argonne, IL; ²National Center for Atmospheric Research, Boulder, CO

Overview

We propose to use the heuristic static load-balancing algorithm for solving load-balancing problems in CESM (Community Earth System Model), using fitted benchmark data as an alternative to the current manual approach. The problem of allocating the optimal number of CPU cores to CESM components is formulated as a mixed-integer nonlinear optimization problem which is solved by using an optimization solver implemented in the MINLP package MINOTAUR. Our algorithm was tested for the 1° and 1/8° resolution simulations on 163,840 cores of IBM Blue Gene/P where we consistently achieved well load-balanced results. This work is a part of a broader effort by NCAR and ANL scientists to eliminate the need for manual tuning of the code for each platform and simulation type, improve the performance and scalability of CESM, and develop automated tools to achieve these goals.

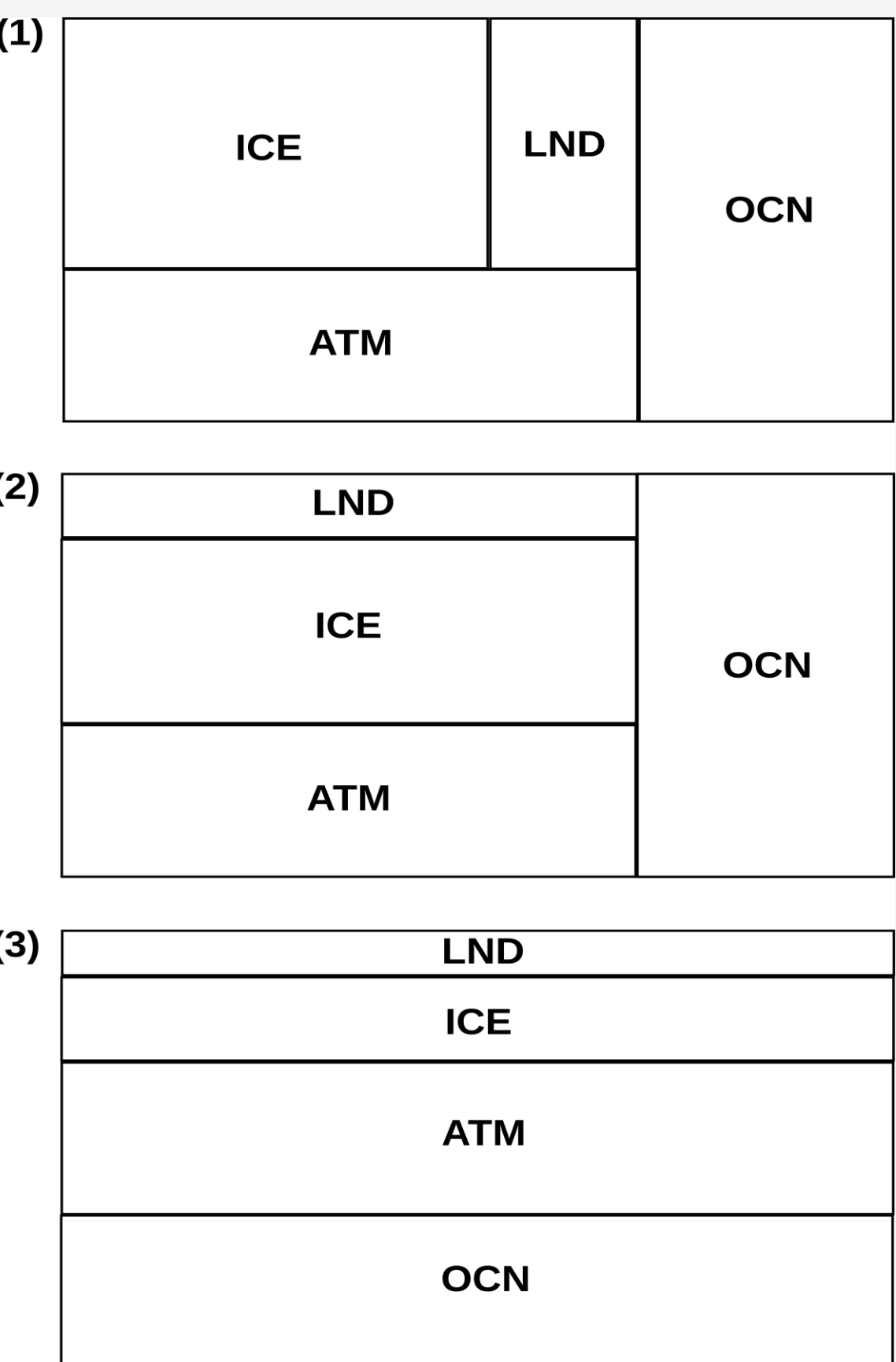


Figure 1. Popular layouts of CESM components. Width of each component represents the number of nodes allocated to it while height represents the time to run it.

CESM, fully coupled runs are usually set up to run with mixed sequential/concurrent layouts (Figure 1). The typical setup runs the atmosphere and ocean on separate processors. The atmospheric model then shares processors with the land and ice models. Other layouts include running the atmosphere, land, and ice models sequentially on a group of processors and running the ocean model on the remaining processors (see Figure 1 (2)). You can also run all of the models sequentially across all processors as shown in Figure 1 (3).

Method

We applied a Heuristic Static Load-Balancing (HSLB) algorithm to CESM. For the decision making step, we formulated a load-balancing problem as a mixed-integer nonlinear optimization (MINLP) problem. The MINLP approach provides great flexibility in modeling the allocation problem realistically. Using nonlinear functions, we can capture complex relationships between the runtime and the number of processors. At the same time, we can impose integer restrictions on certain variables (e.g., number of processors or time requirements). To solve the MINLP for load balancing, we use MINOTAUR, a freely available MINLP toolkit. MINOTAUR offers several algorithms for solving general MINLPs, and can be easily called from different interfaces, for example AMPL scripts or C++ code.

Our HSLB method consists of four steps. First, we collect benchmarking data for each component. Second, we solve for the optimal parameters by a least squares method based on our chosen scalability model. Third, we solve an integer optimization problem in order to obtain an optimal allocation of nodes. Fourth, we allocate the optimal number of nodes obtained from the optimization to run CESM in static load-balancing mode. We will outline each of these steps in greater detail in the next section.

Algorithm

(1) **Gather Data:** Perform CESM simulation for the intended layout D times using a different total numbers of nodes. Collect the running times y_{ji} for each fragment j . It is important to run the model at the fewest and greatest number of cores in order to obtain an accurate representation of the curve.

(2) **Fit:** Next, solve 4 (number of components to fit) different least squares problems outlined in Figure 3 to determine the coefficients a_j , b_j , c_j , and d_j for each fragment j . The results of fitting are shown in Figure 2.

(3) **Solve:** Determine the best node allocation to each component by solving the MINLP outlined in Figure 4, and obtain the optimal values of size n_j for each fragment j .

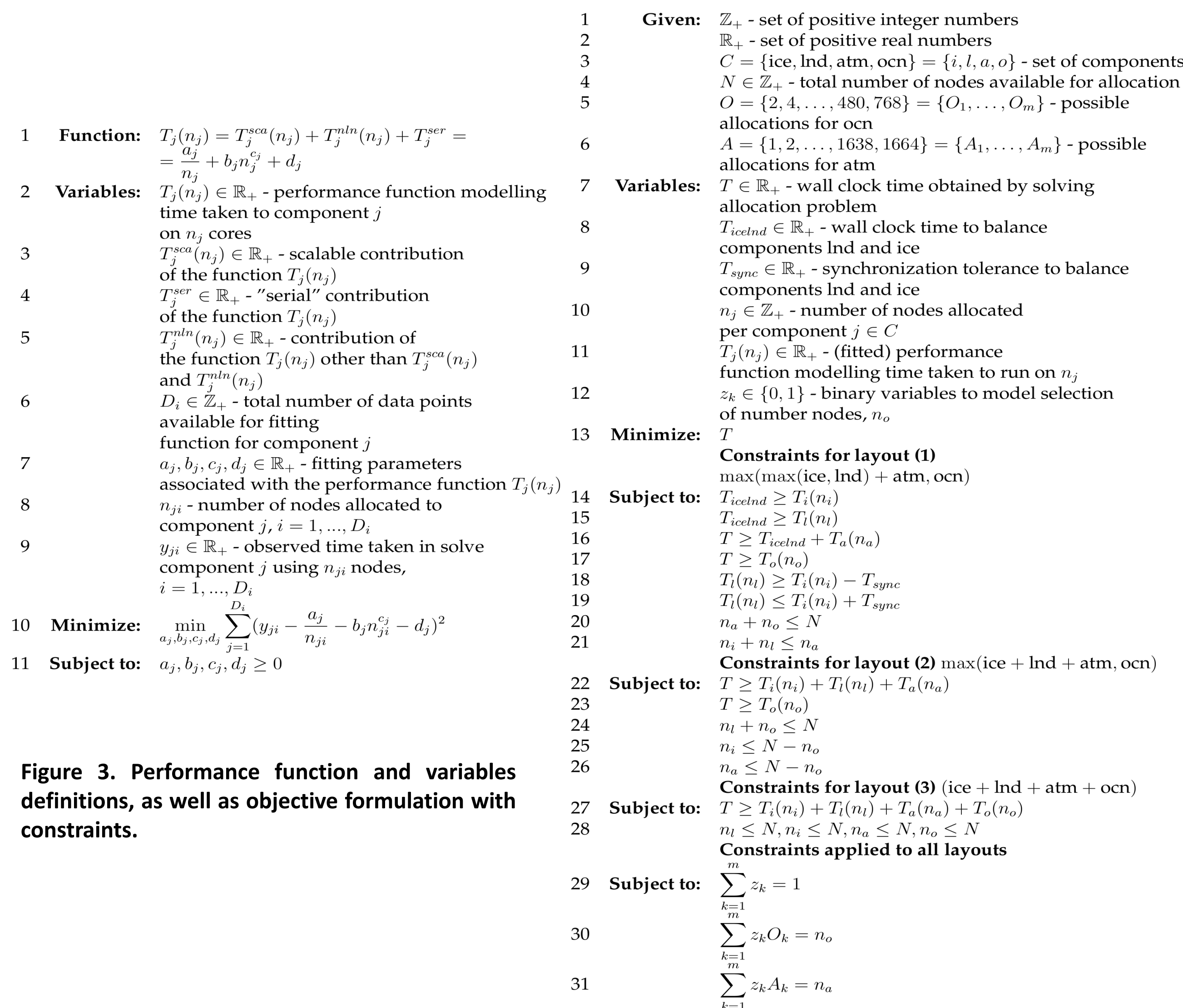


Figure 3. Performance function and variables definitions, as well as objective formulation with constraints.

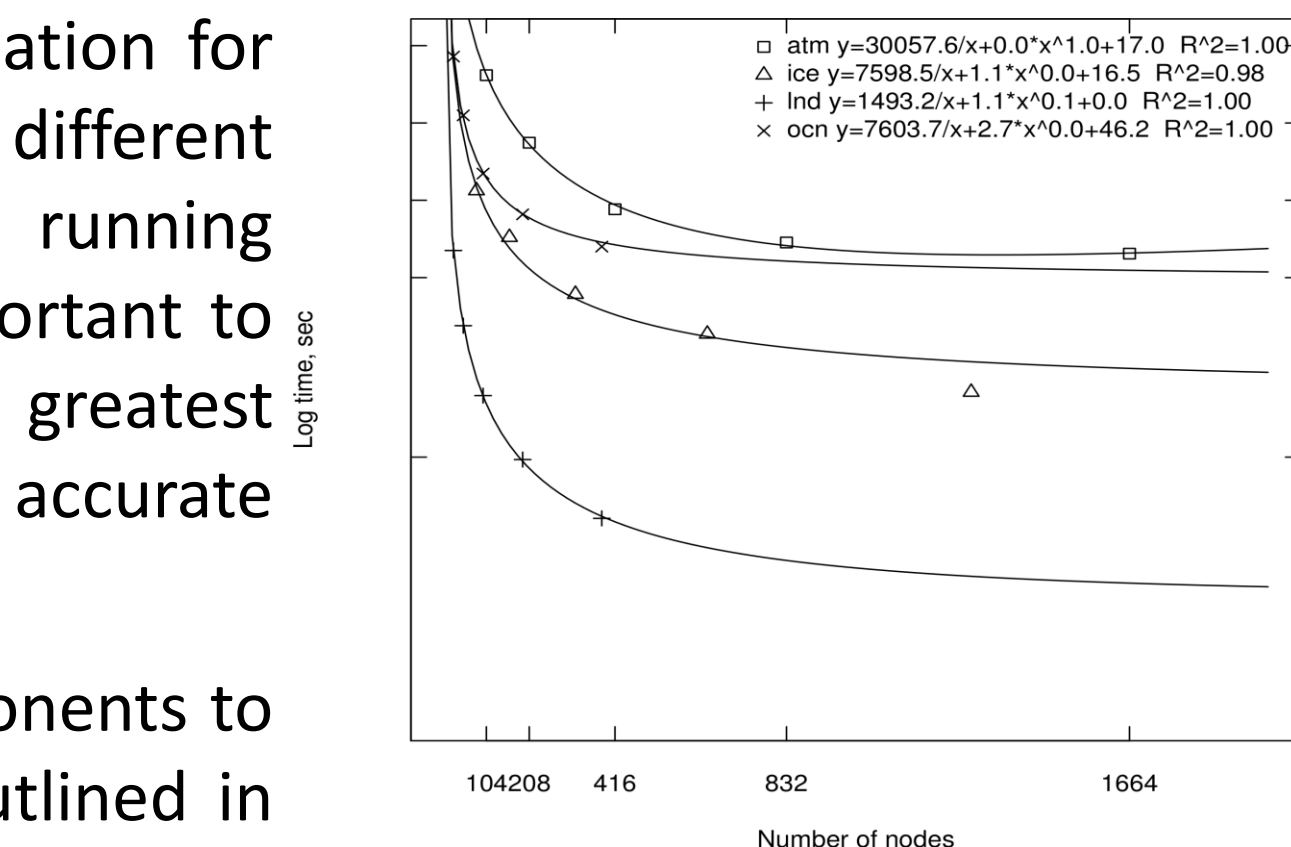


Figure 2. 1 degree FV fully coupled model scaling curves. These were ran on the BGP at Argonne National Laboratory.

Results Continued

We also looked into load balancing 1/8° degree fully coupled model runs. As shown in Figure 5, for 32768 nodes, the model speedup using HSLB compared to the constrained node counts from 2.3 to 2.7 (computed by dividing T(32768) by T(8192) which in case of ideal speedup would be equal to 4). More importantly, the computational time was decreased by 24%.

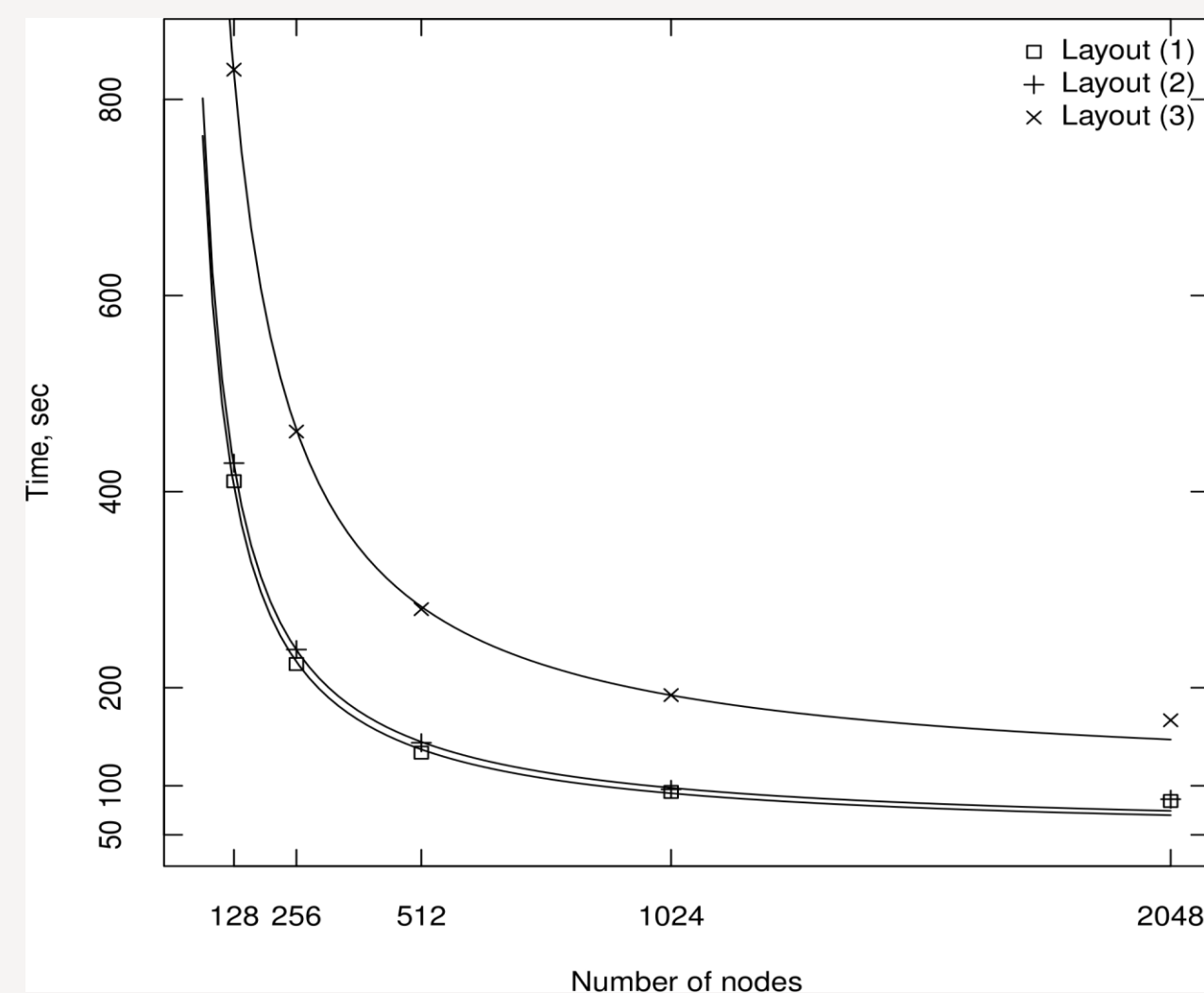


Figure 6. Scaling curves for layouts 1-3 (see Figure 1) for 1° resolution. All data is predicted except for layout (1) for which the experimental data is shown as layout (1exp). The R² between predicted is experimental data for layout (1) is equal to 1.0.

Another important HSLB application may be the prediction of the optimal nodes to run a job. The definition of optimal depends on the goal; it could be the most cost-effective way to run simulations or the shortest time to solution. Since we developed HSLB in part to support DOE INCITE program projects, it is only appropriate to use DOE guidelines to define the maximum size of a job which would have a minimum of 50% efficiency. The efficiency is defined as where and is time to run 5 model day simulations on cores. We modeled efficiency for the 1° resolution layout (1) which is shown in Figure 7. It was found that the maximum allowed cores for this particular job on Intrepid would be 2780. Other metrics could also be easily implemented.

Acknowledgements

We thank Dr. Raymond Loy and ALCF team members for discussions and help related to the paper. We especially thank Jim Edwards and Mariana Vertenstein from NCAR for encouraging this work and helpful discussions. The poster has been created by the UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”) under Contract No. DE-AC02-06CH11357 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. This work was also supported by the U.S. Department of Energy through grant DE-FG02-05ER25694.

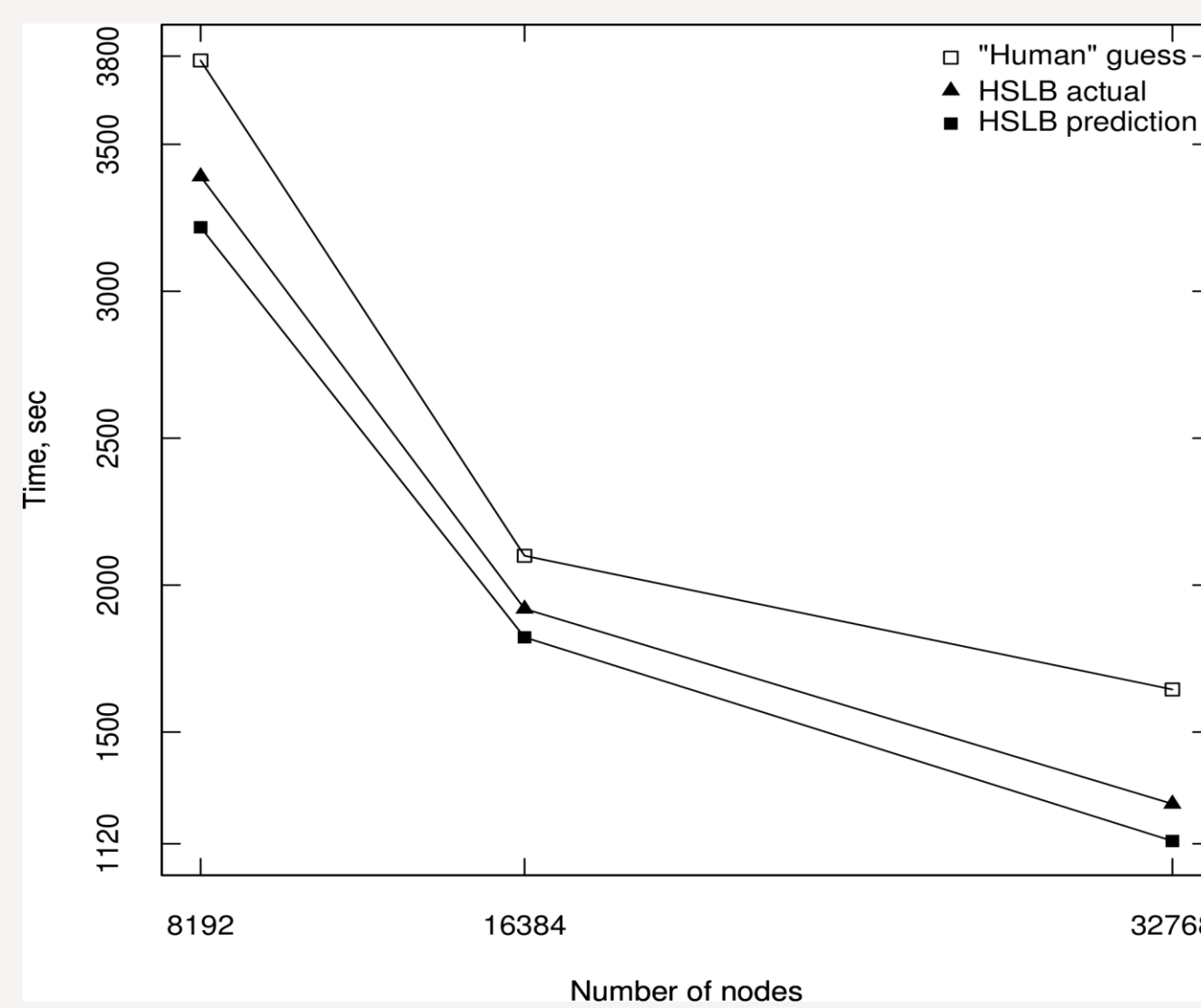


Figure 5. 1/8° resolution scaling curves for layout (1). “Human” guess means optimal allocation was guessed by an expert, HSLB prediction represents timings predicted by HSLB algorithm, and HSLB actual represents actual timings for HSLB prediction.

Since we built mathematical models for three component layouts (see Figure 1), but ran simulations only for most promising layout we decided to predict scaling of the other layouts at the 1° resolution based on the scaling curves shown in Figure 2. The results shown in Figure 6 are somewhat surprising. The model with layouts 1 and 2 performed very similarly, while layout 3, as expected, performs the worst. All layouts scale similarly, but layout 3 is not the most cost effective.

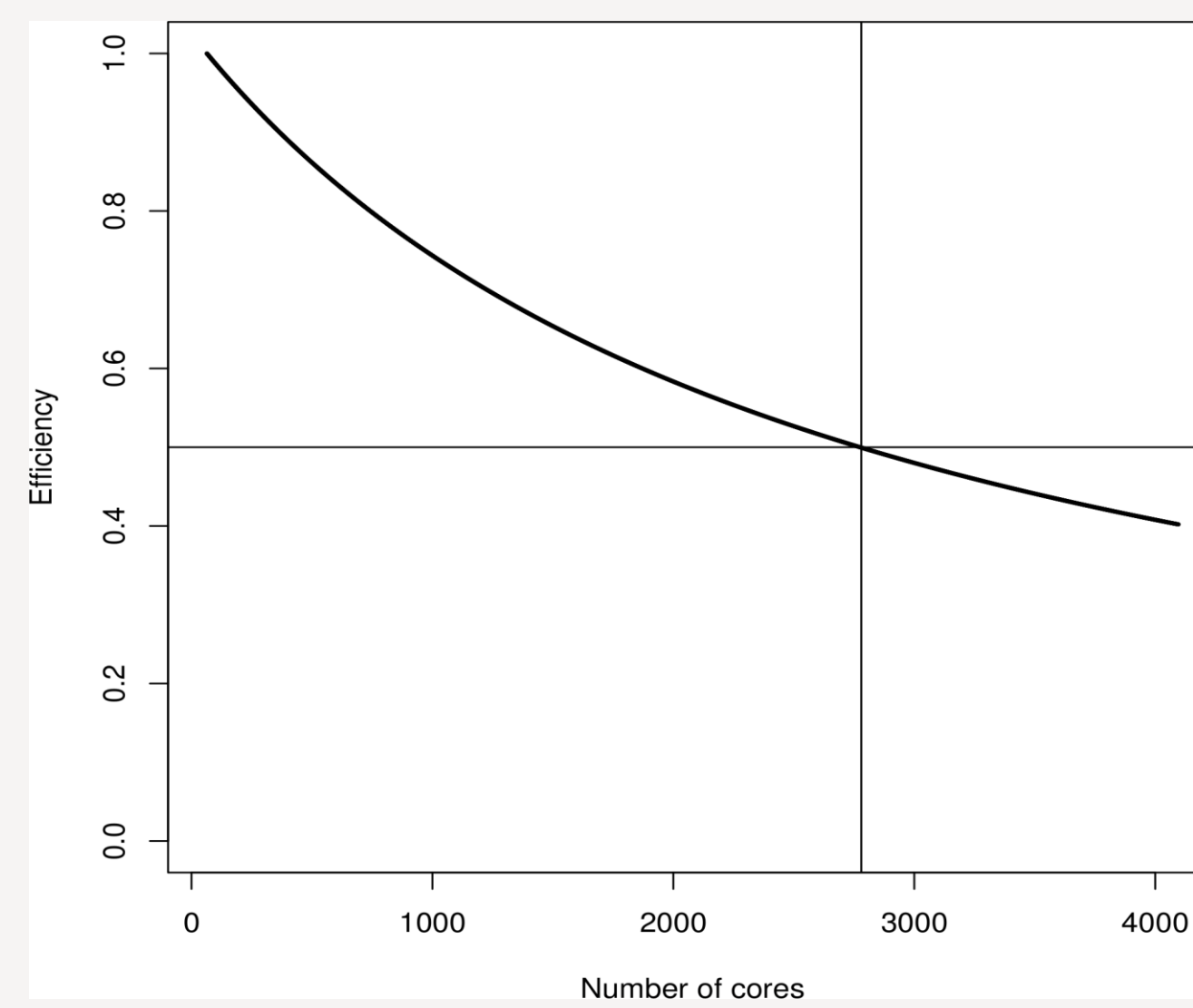


Figure 7. Predicted efficiency curve for layout (1) and 1° resolution.

Results

1° resolution, 128 nodes					
components	Manual		HSLB		
	# nodes	Time, sec	Predicted # nodes	Predicted Time, sec	Actual Time, sec
lnd	24	63.766	15	100.951	100.202
ice	80	109.054	89	102.972	116.472
atm	104	306.952	104	307.651	308.699
ocn	24	362.669	24	365.649	365.853
Total time, sec		416.006		410.623	425.171

1° resolution, 2048 nodes					
components	Manual		HSLB		
	# nodes	Time, sec	Predicted # nodes	Predicted Time, sec	Actual Time, sec
lnd	384	5.777	71	22.693	23.158
ice	1280	17.912	1454	22.822	18.242
atm	1664	61.987	1525	61.662	63.313
ocn	384	61.987	256	78.532	79.139
Total time, sec		79.899		84.484	86.471

Table 1. Detailed timings for each component in layout (1) for 1° resolution. Timings for the lowest and highest node counts are the most interesting ones since some of the data was interpolated. Manual means the best node allocation was predicted by “human optimization” while HSLB means HSLB optimization was used for finding node allocations.

The results of “human” and HSLB optimization for 1 degree FV, fully coupled model runs are shown in Table 1. In column two are the node allocations for each component made by an expert. The corresponding times to run the components are shown in column three; the total time to execute the full CESM run is shown in the row labeled “Total time.” The HSLB node allocations to each component are shown in column four. The AMPL script prints predicted time to run components for comparison with actual time shown in the corresponding columns five and six.