The Heuristic Static Load-Balancing Algorithm Applied to CESM

Yuri Alexeev*, Sheri Mickelson*, Sven Leyffer*, Robert Jacob
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439, USA
yuri@alcf.anl.gov
{mickelso,leyffer,jacob}@mcs.anl.gov

Anthony Craig
CESM software engineer
Seattle, USA
tcraig@ucar.edu

EXTENDED ABSTRACT

Achieving an even load balance is a key issue of parallel computing. Moreover, the impact of load balancing on the overall algorithm efficiency stands to increase dramatically as we enter the peta-scale supercomputing era. By Amdahl's law, the scalable component of the total wall time shrinks as the numbers of processors increases while the load imbalance, together with the constant sequential component, acts to reduce the speedup below that of linearity. While parallelization of sequential code often requires rewriting the code which can be very challenging, adopting an efficient load balancing scheme can be a simple and effective way to boost scalability and performance of the code.

CESM (Community Earth System Model)[3] falls in the special category, as many other applications like FMO (Fragment Molecular Orbitals)[1] method in quantum chemistry, where there are just a few large tasks of diverse size. Dynamic load balancing (DLB) algorithms would not be appropriate because the number of tasks is much smaller than the number of processors. So static load balancing (SLB) would be the only realistic choice. But finding the optimal allocation is not trivial because, except for simple cases, it is an NP-hard problem. To solve load balancing for this special case we developed the Heuristic Static Load-Balancing (HSLB) algorithm as an alternative to currently used "human optimization" where a scientist takes a guess what is optimal node allocation per task (or component for CESM or fragment in FMO). For the decision making step we formulated a load balancing problem as a mixed-integer nonlinear optimization (MINLP) problem[5-12]. The MINLP approach provides a great flexibility in modeling the allocation problem realistically. Using nonlinear functions, we can capture complex relationships between the run time 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[2], a freely available MINLP toolkit. MINOTAUR offers several algorithms for solving general MINLPs, and can be easily called from different interfaces, for example AMPL[13] scripts or C++ code.

In this poster, we applied HSLB to the climate modeling code called the Community Earth System Model (CESM) - on the Blue Gene/P supercomputer installed in Argonne National Laboratory (ANL) called Intrepid[4]. CESM consists of a few distinct components such as an atmosphere model and an ocean model. Each component has different scaling requirements and performance characteristics. This combination of a few components (tasks) of various sizes makes CESM a perfect candidate for the HSLB application. HSLB for CESM was heavily modified compared to HSLB for FMO[1]. The main difference is in constraining task layouts to reflect sequencing and partitioning of components across processors. These constraints are science requirements in CESM and defined by code organization. As a result, the HSLB mathematical models are more sophisticated for CESM because of the use of multiple constraints.

In the poster, we have shown that the HSLB algorithm we developed is a viable alternative to the currently used "manual" optimization approach. By using a linear optimization technique, HSLB predicts an optimal allocation of nodes to the components, given accurate benchmarking data and a correct mathematical model. It is our intention to develop a "black box" from HSLB which would allow anyone, especially scientists without experience at "manual" optimization, to run CESM efficiently on supercomputers or clusters. There is work under way with NCAR scientists to make HSLB a part of the automated pipeline. To this end, we will be converting the AMPL code used to formulate the MINLP problem to C++ code and make it freely available for anyone to use. This work, although initially targeted at the CESM community, can benefit other climate modeling codes as well.

By using HSLB, we improved the speed of CESM on 32768 nodes for 1/8°-degree resolution simulations by 24% compared to a baseline guess on intrepid. HSLB predicted the optimal node allocation while LBE data for the ocean model was used to test some new ocean node counts. This is a good example of how HSLB predicted that using new node counts beyond those hardcoded could significantly improve code performance. There are other possible HSLB applications to CESM For example, HSLB could be used for finding the optimal cost-effective number of nodes to run CESM for a particular task.

Two projects granted computational time on ALCF's Blue Gene/P and Q supercomputers under the DOE INCITE[15-16] program will directly benefit from this work. These are *Climate-Science Computational End Station Development* (PI Warren Washington) and *Attributing Changes in the Risk of Extreme Weather and Climate* (PI Michael Wehner)[17]. By using HSLB, the load balancing optimization can be done quicker and more efficiently and will likely enable them to carry out additional simulation years for 1/8°-degree resolution simulations for a given allocation.

The presented HSLB algorithm is not limited to FMO, CESM, or other climate modeling codes. In fact, any coarse-grained application with large tasks of diverse size can benefit from the present approach. As the number of cores increases in modern supercomputers, the issue of minimizing synchronization time while retaining high efficiency will put load balancing schemes to a highly stressful test. We believe that for coarse-grained applications our HSLB algorithm is a promising and general approach.

Categories and Subject Descriptors
G.1.6 [**Numerical analysis**]: Optimization – *constrained optimization, convex programming, global optimization, integer programming, nonlinear programming.*

General Terms
Algorithms, Performance.

Keywords
Static load balancing, heuristic algorithm, climate modeling, CESM, optimization, MINATOUR, MINLP.

1. ACKNOWLEDGMENTS
We thank Dr. Raymond Loy and ALCF team members for discussions and help related to the poster. We especially thank Jim Edwards and Mariana Vertenstein from NCAR for encouraging this work and helpful discussions.

This submission 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.

2. REFERENCES
[1] Alexeev, Y., Mahajan, A., Leyffer, S., Fletcher, G., and Fedorov, D.G., 2012. Heuristic static load-balancing algorithm applied to the fragment molecular orbital method. In *High Performance Computing, Networking, Storage and Analysis, Supercomputing 2012 International Conference for* IEEE, 1-13.

[2] Mahajan, A., Leyffer, S., Linderoth, J., Luedtke, J., and Munson, T. MINOTAUR wiki. http://www.mcs.anl.gov/minotaur (Accessed January 16, 2013).

[3] The Community Earth System Model. http://www.cesm.ucar.edu/.

[4] Argonne National Laboratory: Argonne Leadership Computing Facility. http://www.alcf.anl.gov/.

[5] Dakin, R.J., 1965. A tree-search algorithm for mixed integer programming problems. *The Computer Journal 8*, 3, 250-255.

[6] Quesada, I. and Grossmann, I.E., 1992. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers & chemical engineering 16*, 10-11, 937-947.

[7] Duran, M.A. and Grossmann, I.E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming 36*, 3, 307-339.

[8] Fletcher, R. and Leyffer, S., 1994. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming 66*, 1, 327-349.

[9] Westerlund, T. and Pettersson, F., 1995. An extended cutting plane method for solving convex MINLP problems. *Computers & chemical engineering 19*, 131-136.

[10] Mahajan, A., Leyffer, S., and Kirches, C., 2012 *Solving mixed-integer nonlinear programs by QP-diving. Technical report ANL/MCS-P2071-0312.* Argonne National Laboratory.

[11] Horst, R. and Hoang, T., 1996. *Global Optimization: Deterministic Approaches*. Springer-Verlag, Berlin.

[12] Tawarmalani, M. and Sahinidis, N.V., 2002. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, Dordrecht.

[13] Fourer, R., Gay, D.M., and Kernighan, B., 2002. *AMPL: A Modeling Language for Mathematical Programming, 2nd Edition* Cengage Learning, Independence, KY.

[14] Balaprakash, P., Alexeev, Y., Mickelson, S., Leyffer, S., Jacob, R., and Craig, A.P., 2013. Machine learning based load-balancing for the CESM climate modeling package. In *Supercomputing 2013 International Conference for* IEEE, submitted.

[15] Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program http://www.doeleadershipcomputing.org/incite-program/.

[16] Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program at ALCF. https://www.alcf.anl.gov/incite-program.

[17] ALCF 2013 Projects. https://www.alcf.anl.gov/projects.