Code Optimization and Stabilization for a High-Resolution Terrain Generation Application*

Judith D. Gardiner
Ohio Supercomputer Center,
Columbus, OH 43212,
judithg@osc.edu

Karen A. Tomko
Ohio Supercomputer Center,
Columbus, OH 43212,
ktomko@osc.edu

Myoung-Jong Noh
Byrd Polar and Climate Research
Center, Ohio State University,
Columbus, OH 43210,
ngnmj77@gmail.com

Ian M. Howat
School of Earth Sciences, Ohio State University, Columbus, OH 43210, ihowat@gmail.com

ABSTRACT

Code improvements have been made to the Surface Extraction from TIN based Search-space Minimization (SETSM) software, used to produce digital elevation maps from satellite imagery. The improvements include increased code portability and stability, along with enhanced performance. The code has been made available on a public repository.

CCS CONCEPTS

• General and reference → Performance • Applied computing → Earth and atmospheric sciences

KEYWORDS

High performance computing (HPC), digital elevation models, open-source software

ACM Reference format:

Judith D. Gardiner, Karen A. Tomko, Myoung-Jong Noh and Ian M. Howat. 2018. Code Optimization and Stabilization for a High-Resolution Terrain Generation Application*. In *Proceedings of Practice and Experience in Advanced Research Computing (PEARC '18)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3219104.3229241>

INTRODUCTION

Access to submeter resolution commercial satellite imagery through NSF-sponsored research centers such as the Polar Geospatial Center (PGC) has the potential to revolutionize Earth observation. To make use of this imagery there must be an efficient system for producing and distributing digital elevation models (DEMs). The Surface Extraction from TIN-based Searchspace Minimization (SETSM) software from The Ohio

State University (OSU) provides fully-automated DEM extraction to produce DEM-based terrain maps (see Fig. 1).

The ArcticDEM Production project on Blue Waters is using SETSM to construct the first high-resolution (2m), high-quality (<5m uncertainty), and openly distributed DEM set of all Arctic land masses. This project involves generations of DEMs from thousands of satellite images, utilizing millions of node-hours.

Optimization and parallelism of the SETSM code are essential for large-scale terrain generation. This poster describes the software engineering efforts that contributed to the success of this project.

Figure 1: Hillshade image derived from an 8m DEM computed by SETSM. 19km x 23km area in northwestern Alaska, generated with SETSM on OSC's Owens cluster.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PEARC '18, July 22–26 2018, Pittsburgh, PA, USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6446-1/18/07.

<https://doi.org/10.1145/3219104.3229241>

SOFTWARE ENGINEERING GOALS

Our task was to turn an innovative research code into a high-quality, high-performance, open-source production code. We identified several software engineering goals motivated by our targets of providing an open-source release of SETSM and providing the software on XSEDE resources as “Software as a Service” (SAAS). These goals are outlined below

- Robustness – Reduce problems including crashes and erroneous output to just a few out of millions of runs.
- Portability – Work with the most common C/C++ compilers on common Linux/Unix-based HPC systems.
- Maintainability – Perform code cleanup and restructuring to improve readability.
- Performance – See “Code Optimization Approaches”.

SOFTWARE ENGINEERING CHALLENGES

SETSM code optimization proved more challenging than we anticipated. The algorithm was evolving as we were working on the code. Some algorithm updates had significant performance impacts to both execution time and memory requirements. SETSM originally worked with just one compiler, and several bugs were found when trying different compilers. Portions of the code had floating point sensitivities which impacted the reproducibility of results and slowed correctness testing and debugging efforts. Testing was further complicated by the lack of clear criteria for correctness. A correct output DEM should be “close” to a reference output, and close was obvious to the geoscience researchers but was not defined in a way that was amenable to automated comparison by computer scientists. Lastly, existing OpenMP regions introduced nondeterministic results due to multithreading, but the resulting outputs were considered correct within the variability this introduced.

CODE OPTIMIZATION APPROACHES

SETSM is a young software product that is still under active development. Development is primarily focused on new capabilities and improving the quality of the resulting DEM images. At the start of the effort described here, SETSM employed OpenMP-based multithreading for parallelism, but was otherwise unoptimized. We took an iterative approach to optimizing SETSM, which occurred alongside algorithm updates and software engineering improvements. Throughout the process, profiling tools were applied to focus our optimization efforts. The following techniques were investigated, some with success and others with little impact.

- Improved OpenMP loop scheduling and switched to version 3.1, which supports min/max reductions.
- Moved to a highly optimizing compiler for 25% performance improvement (see Fig. 2).
- Streamlined code to reduce redundant operations.
- Refactored loop nests for memory locality to improve cache utilization.

- Restructured loop bodies for vectorization with limited success. The current algorithm is not amenable to vectorization.
- Explored use of OpenACC with experts in the Blue Waters PAID program to utilize GPUs for SETSM runs, with limited success due to the complexity of the loop nests required to run on a GPU in order to make significant performance gains.
- Introduced MPI-level parallelism resulting in a Hybrid MPI/OpenMP implementation using ad hoc static load balancing (see Fig. 3).

Figure 2: Impact of compiler on SETSM performance; walltimes for different compilers for generating Alaska benchmark to 2m on OSC’s Owens cluster (Intel Xeon E5-2680 v4 CPUs, with 28 cores and 128 GB of memory per node).

Figure 3: MPI/OpenMP hybrid parallel processing; walltimes for various numbers of nodes for generating Alaska benchmark to 2m on OSC’s Owens cluster (Intel Xeon E5-2680 v4 CPUs with 28 cores and 128 GB of memory per node).

RESULTS

We are approaching the end of a 3-year involvement with SETSM and have achieved most of our goals. Our robustness and portability goals have been largely achieved. Maintainability has improved, but more can be done. We estimate a 34% performance improvement; however, this is difficult to measure because of algorithm changes. The performance improvements along with algorithm changes have allowed Arctic and Antarctic DEMs to be computed within the allocated time on Blue Waters. The hybrid OpenMP/MPI parallel SETSM implementation, available in a recent release, allows users to make trade-offs between single-image turnaround time vs. throughput for processing of large-scale geographic regions. SETSM is available as an open source project on github, and a “Software as a Service” deployment is in progress on SDSC’s Comet system.

ACKNOWLEDGEMENTS

This research is supported in part by National Science Foundation awards PLR-1542736 and OCI-0725070 and by NNX10AN61G from the US National Aeronautics and Space Administration.

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

This research utilized the resources of the Ohio Supercomputer Center.

REFERENCES

- [1] Noh, M. J., I.M. Howat, 2017. Surface Extraction from TIN based Search-space Minimization (SETSM) algorithm, International Society for Photogrammetry and Remote Sensing (ISPRS) Journal of Photogrammetry and Remote Sensing, Vol. 129, pp. 55-76.
- [2] Morin, Paul. Mapping the Poles with Petascale, PEARC17 keynote talk.
- [3] <https://mjremotesensing.wordpress.com/setsm/>
- [4] <https://github.com/setsmdeveloper/SETSM>