

Improving the I/O Performance and Memory Usage of the Xolotl Cluster Dynamics Application*

Philip C. Roth

Oak Ridge National Laboratory
Oak Ridge, Tennessee
rothpc@ornl.gov

David E. Bernholdt

Oak Ridge National Laboratory
Oak Ridge, Tennessee
bernholdtde@ornl.gov

Sophie Blondel

University of Tennessee-Knoxville
Knoxville, Tennessee
sblondel@utk.edu

Brian D. Wirth[†]

University of Tennessee-Knoxville
Knoxville, Tennessee
bdwirth@utk.edu

ABSTRACT

Xolotl is a cluster dynamics simulator used to predict gas bubble evolution in solids. It is currently being used to simulate bubble formation in the plasma-facing surface within fusion reactors and the nuclear fuel used in fission reactors. After observing performance problems in coupled-code simulations of fusion reactors, we used Xolotl’s built-in performance data collection infrastructure and an external profiling tool to identify inefficiencies when writing Xolotl’s two types of checkpoint files. We changed the code to use true parallel writes via the HDF5 data management library, resulting in a code that is approximately 57× faster when writing the program’s main checkpoint file at the scale used in the coupled-code simulations, and that exhibits less performance variability due to external system activity. We also identified and addressed a memory usage problem that reduced Xolotl peak memory usage by approximately 88% per compute node.

KEYWORDS

Performance engineering, I/O, memory usage, Xolotl

ACM Reference Format:

Philip C. Roth, Sophie Blondel, David E. Bernholdt, and Brian D. Wirth. 2018. Improving the I/O Performance and Memory Usage of the Xolotl Cluster Dynamics Application. In *Proceedings of The International Conference for High Performance Computing, Networking, Storage and Analysis (SC’18)*. ACM, New York, NY, USA, Article 4, 3 pages. https://doi.org/10.475/123_4

*This manuscript has been co-authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

[†]Also with Oak Ridge National Laboratory.

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or affiliate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.

SC’18, November 2018, Dallas, Texas USA

© 2018 Association for Computing Machinery.

ACM ISBN 123-4567-24-567/08/06...\$15.00

https://doi.org/10.475/123_4

1 INTRODUCTION

Xolotl [6, 12] is a cluster dynamics simulator used to predict gas bubble evolution in solids. It solves the Advection-Diffusion-Reaction cluster dynamics equations with an incident flux.

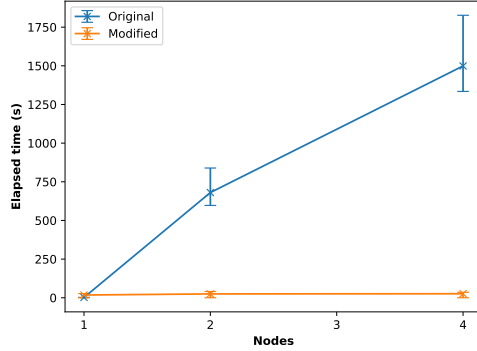
$$\delta_t \bar{C} = \phi \cdot \rho + D \nabla^2 \bar{C} - \nabla \bar{v} C - \bar{Q}(\bar{C})$$

Xolotl was initially developed within a project funded by the United States Department of Energy (DOE) Office of Science’s Scientific Discovery through Advanced Computing (SciDAC) program to study the impact of simulate He/H bubble formation in the plasma-facing surface of the diverter of a fusion reactor [8], and is currently used and extended within that project’s successor project [9]. It is used in another SciDAC project to simulate the formation of Xe bubbles in fission reactor fuel [11], and as part of coupled-code simulations of the PISCES linear plasma device [4] and the ITER fusion reactor [5] for the DOE Office of Fusion Energy Sciences’ fiscal year 2018 Plasma Materials Interactions (PMI) Theory Milestone experiments.

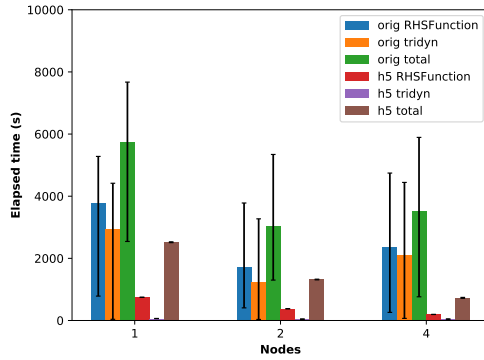
In this poster, we describe how we used Xolotl’s built-in performance data collection infrastructure and external tools to identify and address the source of several I/O and memory usage problems exposed in recent PMI Theory Milestone experiments. Our intent in presenting this poster is not only to describe the benefit of our Xolotl performance engineering activities, but also to document an example of a highly productive collaboration between the SciDAC Institute for Computer Science and Data (RAPIDS) [10] and two SciDAC applications projects.

2 IMPROVING I/O PERFORMANCE

Our PMI Theory Milestone experiments involve a workflow that runs many instances of Xolotl, each simulating 1D problems representing a “pencil” extending into the surface being simulated. To investigate why our recent experiments did not achieve our desired simulation time target, we extracted program inputs from one of these Xolotl instances to act as a benchmark problem for performance engineering on the Eos Cray XC30 deployed in the Oak Ridge Leadership Computing Facility (OLCF). We used the timers provided by Xolotl’s lightweight, “always-on” performance data collection infrastructure to conduct a scalability study. This study showed that a function `startStop1D` that is called every time step by the PETSc solver [2] (a so-called “monitor function”) was not



(a) startStop1D



(b) tridyn

Figure 1: Elapsed time for Xolotl’s startStop1D and tridyn monitor functions on the OLCF Eos Cray XC30 system, 32 processes per node. Data labeled ‘master’ or ‘orig’ indicate the original implementation, ‘opt’ and ‘h5’ indicate our modified versions. Each data point represents the maximum observed time from any process, averaged over at least three runs. Bars indicate the maximum and minimum of the maximum observed times.

scaling well and was highly sensitive to other system activity. One purpose of this function is to write periodically the program’s current state to an HDF5 checkpoint file. Profiling with HPCToolkit [1] and code inspection suggested that MPI_Allreduce calls on small data for each grid point were the culprit, and that these calls were being made so that each Xolotl process could obtain all of the data being written. Combining performance engineering expertise of RAPIDS project personnel with guidance from the primary Xolotl developer, we modified the code to use true parallel I/O using the HDF5 interface, improving the performance of startStop1D by approximately 57× for our benchmark problem while also reducing its performance variability as shown in Figure 1a.

Upon re-running the scalability study using our modified version of Xolotl, we determined that another monitor function called

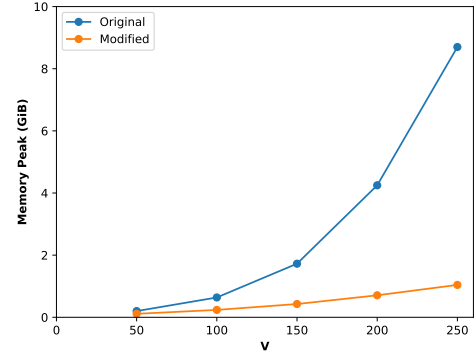


Figure 2: Xolotl peak memory usage vs. input problem size as measured by the maximum number of vacancies (V) in a cluster.

tridyn was also exhibiting a high degree of performance variability on the Eos system. This function is also called by PETSc at every time step, and is used to write a file whose contents may be used as input to Fractal TRIDYN [3] within the larger PMI Theory Milestone coupled-code simulation. This function hadn’t been flagged as a problem in our initial scalability study, both because it was overshadowed by the startStop1D problem and because it was not quite as sensitive to other system activity as startStop1D. Through code inspection, we determined that tridyn had a similar implementation approach to startStop1D including use of MPI collective operations for small data at each grid point in preparation for writing the entire dataset from each Xolotl process. By converting this function to use true parallel I/O via the HDF5 interface, we observed a dramatic reduction in performance variability (see Figure 1b).

3 REDUCING PEAK MEMORY USAGE

In addition to our recent I/O performance engineering work with Xolotl, we have also addressed a memory usage problem that caused the software to exhibit out-of-memory errors with smaller-than-expected problem sizes when running on high performance computing (HPC) systems like OLCF Eos. To diagnose this problem, we used the Valgrind Massif memory profiler [7] running on a large memory Linux system to identify that Xolotl’s peak memory consumption occurred when initializing the PETSc solver. This initialization required construction of two 2D matrices of boolean values, and each dimension can have size larger than 10,000 items for relatively small Xolotl problems. Because there are few non-zeros in these matrices, we proposed a modified application programming interface function to the PETSc developers that takes these matrices in a sparse representation, and they have accepted the new function into their code repository for future release. Using this alternative interface, Xolotl’s peak memory footprint per compute node was reduced by 88% (see Figure 2).

REFERENCES

- [1] Laksono Adhianto, Sinchan Banerjee, Mike Fagan, Mark Krentel, Gabriel Marin, John Mellor-Crummey, and Nathan R. Tallent. 2010. HPCToolkit: tools for performance analysis of optimized parallel programs. *Concurrency and Computation: Practice and Experience* 22, 6 (2010), 685–701. <https://doi.org/10.1002/cpe.1553>
- [2] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. 2016. *PETSc Users Manual*. Technical Report ANL-95/11 - Revision 3.7. Argonne National Laboratory. <http://www.mcs.anl.gov/petsc>
- [3] Jon Drobny, Alyssa Hayes, Davide Curreli, and David N. Ruzic. 2017. F-TRIDYN: A Binary Collision Approximation Code for Simulating Ion Interactions with Rough Surfaces. *Journal of Nuclear Materials* 494 (October 2017), 278–283.
- [4] Y. Hirooka, R.W. Conn, T. Sketchley, W.K. Leung, G. Chevalier, R. Doerner, J. Elverum, D.M. Goebel, G. Gunner, M. Khandagle, B. Labombard, R. Lehmer, P. Luong, Y. Ra, L. Schmitz, and G. Tynan. 1990. A New Plasma-Surface Interactions Research Facility: PISCES-B and First Materials Erosion Experiments on Bulk-Boronized Graphite. *Journal of Vacuum Science & Technology A* 8, 1790 (1990). <https://doi.org/10.1116/1.576805>
- [5] ITER: The Way to New Energy 2018. <https://www.iter.org>.
- [6] Dimitrios Maroudas, Sophie Blondel, Lin Hu, Karl D Hammond, and Brian D Wirth. 2016. Helium segregation on surfaces of plasma-exposed tungsten. *Journal of Physics: Condensed Matter* 28, 6 (2016), 064004. <http://stacks.iop.org/0953-8984/28/i=6/a=064004>
- [7] N. Nethercote and J. Seward. 2007. Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation. In *Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'07)*. San Diego, California, USA.
- [8] Plasma Surface Interactions: Bridging from the Surface to the Micron Frontier through Leadership Computing 2017. <https://collab.cels.anl.gov/display/PSIsdac/Plasma+Surface+Interactions>.
- [9] Plasma Surface Interactions: Predicting the Performance and Impact of Dynamic PFC Surfaces 2018. <https://collab.cels.anl.gov/display/PSIsdac2/Plasma+Surface+Interactions+2>.
- [10] SciDAC Institute for Computer Science and Data (RAPIDS) 2018. <https://rapids.lbl.gov>.
- [11] Simulation of Fission Gas in Uranium Oxide Nuclear Fuel 2018. <https://collab.cels.anl.gov/display/FissionGasSciDAC2>.
- [12] Xolotl: Spatially dependent cluster dynamics simulator 2018. <https://github.com/ORNLFusion/xolotl>.