ORNL/LTR-2022/415

The Importance of Scientific Visualization on Novel Hardware

Kenneth Moreland
David Pugmire
Berk Geveci
Li-Ta Lo
Hank Childs
Mark Bolstad
Ruchi Shah
Panruo Wu

September 2022

Approved for public release.
Distribution is unlimited.

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via OSTI.GOV.

Website: www.osti.gov/

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: 703-605-6000 (1-800-553-6847)
TDD: 703-487-4639
Fax: 703-605-6900
E-mail: info@ntis.gov
Website: http://classic.ntis.gov/

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone: 865-576-8401
Fax: 865-576-5728
E-mail: report@osti.gov
Website: https://www.osti.gov/

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/LTR-2022/415

Computer Science and Mathematics Division

The Importance of Scientific Visualization on Novel Hardware

Submitted to the ASCR Workshop on Visualization for Scientific Discovery, Decision-Making, & Communication

Kenneth Moreland, David Pugmire, Berk Geveci, Li-Ta Lo, Hank Childs, Mark Bolstad, Ruchi Shah, Panruo Wu

September 2022

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-Battelle LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

Topics 3b – Novel technology for visualization (primary), 3a – Visualization for data and technology at the edge (secondary)

Challenge

Innovation in HPC hardware and adoption of heterogeneous systems has led to a variety of unique programming models. This has led to a challenge for scientific visualization software (and indeed all HPC software) to take full advantage of recent generations of supercomputing. Edge computing, where hardware is specialized for the needs of the particular application, exacerbates the problem. VTK-m has had many successes on this front by providing device-agnostic algorithms that compare favorably to implementations written to specific devices as shown in Table 1. However, VTK-m has focused mostly on GPU and traditional CPU multicore technology. There are numerous processor technologies, both existing and potential future, that are not being addressed by current R&D efforts.

Table 1. Aggregate speedup of VTK-m compared to equivalent algorithms written for a specific device. A speedup greater than 1 means that VTK-m runs faster. From Moreland, et al. [3].

Algorithm	CPUs	GPUs	X. Phi	Serial	Total
External facelist	-	-	-	0.34	0.34
PGM 18	3.32	-	0.87	-	1.69
PGM 20	2.39	-	0.25	-	0.78
Particle advection	0.38	1.53	-	-	0.76
Point merge	1.82	-	-	3.10	2.38
Ray tracing	0.47	0.55	-	-	0.51
Volume rendering	1.13	0.83	-	3.10	1.43
Wavelet compression	1.13	0.75	-	-	0.92
Hashing	5.97	1.45	-	-	2.94
Total	1.45	0.95	0.47	1.48	**1.14**

SIMD Many modern processors contain special vector instructions that perform SIMD parallel processing. SIMD is similar to other threading technology like GPU warps but far more restrictive. VTK-m currently relies on compilers to generate SIMD instructions. However, the performance of Xeon Phi, which relies heavily on deep SIMD instructions, shown in Table 1 as well as other research [1] suggest VTK-m's abstractions are working against SIMD generation.

Mobile Devices Visualization on mobile devices is useful in its own right. But processors originally designed for the mobile market are having an impact on HPC. For example, the current top supercomputer, Fugaku, achieves its record-breaking performance with ARM processors inspired by the mobile electronics market. It is unclear how optimizations for low-powered mobile devices differ from top-end number crunchers.

Half-Precision Floats Some modern processors support 16-bit floating point numbers. The advantage of using these less precise numbers is less demand on memory and more instructions that can be performed. In fact, half-precision enabled Summit to perform better than 1 ExaFLOP as early as 2018 [2]. Although

the uncertainty introduced by half precision in visualization is not well studied, it is likely that computation could be possible with little to no artifacts in many cases. After all, 8-bit color channels are still normal.

Tensor Cores Driven by the AI market, many processor manufacturers are including tensor cores, which are exceptionally fast for small matrix computation. These instructions may be useful for visualization operations such as affine transformations. As a key technology for machine learning, we expect tensor cores to become an essential component of processors.

FPGAs A field programmable gate array (FPGA) are essentially circuits that can be reconfigured in place to perform a custom operation. Once burned, an FPGA can perform computations faster and with less power than its instruction-based processor counterpart. Although not yet leveraged by DOE LCFs today, it is conceivable they would be considered for performance improvements in future generations. In contrast, FPGAs are a very attractive option for edge computing, which typically has dedicated hardware for a specific task.

Reversible Computing The von Neumann-Landauer limit establishes the minimum amount of energy physically required to change a bit's state. This is a fundamental physical limit to computer scaling, and thus one of the limits to Moore's law. Getting around this limit requires the computation to be reversible. A proposed technique to implement reversible logic is the use of adiabatic circuits, which reclaim the energy released in one part of the circuit in another part. This is a hardware-level optimization, but what does it mean for memory systems? Can energy be reclaimed in a DRAM circuit? This has the potential to seriously disrupt the ratio of computing to storage.

Quantum Computing A qubit in a quantum computer can simultaneously represent a 0 or 1 with assigned probabilities. Qubits can be entangled so that n bits can represent 2n states, which makes it possible for quantum computers to solve problems with exponential requirements on classical computers. Quantum computers are in their infancy and there has been very little in related visualization work. It is still an open question on what if any open needs of visualization are there for quantum computers or the data they generated. A separate question is how can we apply quantum computing to visualization.

Opportunity

Each of the aforementioned challenges is simultaneously an opportunity. In each situation, it is likely that a classical visualization algorithm could be used, albeit inefficiently (with the possible exception for quantum computers). Just as each of these processor technologies have been introduced to provide better computation features, visualization software is given an opportunity to take advantage of these new features. Similarly, innovations continue to happen with software like C++ standards, OpenCL, and Kokkos.

The interesting questions are what features can be leveraged, how they might be leveraged, what benefits do we stand to gain, how do we manage the interaction between these features, and how do we deliver the solutions to the masses? Using these new hardware features requires extensive research in new algorithms. Any new computing model will require new programming models and new algorithms that fit within. If the scientific visualization community is not vigilant, we will once again be challenged to provide data discovery on emerging processors.

Timeliness

The introduction of general GPU processors into HPC systems was a wake up call to the scientific visualization community, which responded with several research projects to address visualization computations on these new accelerators [4]. Today, the VTK-m project [3] provides a software framework for developing accelerated visualization algorithms as well as a collection of written algorithms. With the basics of GPU computation well in hand, it is time to focus our new research activities to other processor hardware features. That said, several of the aforementioned hardware features are available at DOE LCFs today but remain underutilized. And although GPU technology remains the predominant acceleration technology, there are numerous other technologies available today that are likely to soon be integrated into HPC. Furthermore, instances of edge computing are more quickly adopting many of these solutions to satisfy specialized needs. In all, further R&D is needed to satisfy DOE's needs.

References

[1] E. Wes Bethel, Colleen Heinemann, and Talita Perciano. Performance tradeoffs in shared-memory platform portable implementations of a stencil kernel. In *EGPGV*, 2021.

[2] Thorsten Kurth et al. Exascale deep learning for climate analytics. In *SC '18*, November 2018.

[3] Kenneth Moreland et al. Minimizing development costs for efficient many-core visualization using MCD^3. *Parallel Computing*, 108(102834), 2021.

[4] Christopher Sewell et al. The SDAV software frameworks for visualization and analysis on next-generation multi-core and many-core architectures. In *Proc. of UltraVis*, pages 206–214, 2012.