CGYRO performance on Slingshot-connected Perlmutter GPU nodes

Igor Sfiligoi
University of California San Diego
La Jolla, CA, USA
isfiligoi@sdsc.edu

Emily Belli
General Atomics
La Jolla, CA, USA
bellie@fusion.gat.com

Jeff Candy
General Atomics
La Jolla, CA, USA
candy@fusion.gat.com

***Abstract*— The NERSC Perlmutter HPC system is the most recent large-scale US system that is publicly available. NERSC chose to deploy a first phase of its GPU-based nodes in late 2021 using 2x Slingshot10 connections and has been upgrading them to 4x Slinghot11 connections starting in summer 2021. In this paper we provide benchmark numbers for using CGYRO, a popular fusion turbulence simulation tool, comparing the original and the upgraded network setup. CGYRO has been previously shown to be communication-bound in many recent HPC systems and we show that the upgraded networking provides a significant boost for fusion science.**

Keywords—benchmarking, HPC, MPI, Slingshot, fusion

I. Introduction

The Perlmutter system [1] is the latest HPC system being deployed at National Energy Research Scientific Computing (NERSC) Center. Perlmutter is a Hewlett Packard Enterprise (HPE) Cray EX supercomputer based on the HPE Cray Shasta platform, featuring both CPU-only and GPU-accelerated compute nodes. Note that for the purpose of this poster, we will only consider the GPU-accelerated ones. All nodes are connected using HPE Cray Slingshot interconnect [2].

Each node [3] is composed of a single AMD EYPC 7763 CPU and four NVIDIA A100 SXM4 40GB GPUs. The GPUs are connected in a full NVLINK-3 mesh between them and while communicating with the CPU and the networking over PCIe 4.0. In the original setup, each node was configured with two Slingshot 10 network interface cards (NICs), while the upgraded nodes have four Slingshot 11 NICs each.

In this poster we focus on the CGYRO [4] fusion turbulence simulation tool. CGYRO is an Eulerian gyrokinetic solver designed and optimized for collisional, electromagnetic, multiscale simulation, and is widely used in the fusion research community. While fusion energy research has made significant progress over the years, the complexity of the turbulence in toroidal plasmas makes it difficult to accurately predict fusion reactor performance. While experimental methods are essential for gathering new operational modes, simulations are used to validate basic theory, plan experiments, interpret results on present devices, and ultimately to design future devices.

CGYRO operates on a 6-dimensional grid (3D space + 2 D velocity + 1 D species). The compute is naturally parallelizable, so most of the compute happens on GPUs. Memory access is however not partitioned, so in order to concurrently utilize multiple GPUs, the problem is split in many smaller sub-problems using the Message Passing Interface (MPI) paradigm, splitting the grid using two orthogonal MPI communicators [5]. At each time step there are several interleaved MPI_AllToAll and MPI_AllReduce collective operations needed on the two MPI communicators. The amount of data exchanged between processes is thus significant, making the simulation communication-bound on most recent systems [6].

Since the fusion community is heavily relying on NERSC systems, we are very happy to see the networking of the GPU-accelerated nodes being upgraded. The next section provides the benchmark results of a cutting-edge benchmark simulation, followed by an analysis of the results.

II. Benchmark Results

In this poster we showcase the benchmark results for the sh04 [7] benchmark simulation case. This benchmark case represents the current cutting-edge multi-scale fusion turbulence simulations, having a 5+1-dimensional grid of (128 x 1152 x 24 x 18 x 8) x 3 species. Note that this use-case was pushing the limits of what the previous-generation Cori KNL system at NERSC could deliver, so the new Perlmutter system is a very welcome addition for these kinds of studies.

TABLE I. RUN TIMES FOR THE SH04 CGYRO SIMULATION

Machine Name	NICs	Nodes	Runtime for 10 a/c_s, in minutes		
			Total	*Compute*	*Comm.*
Perlmutter GPU	Sligshot 11 x4	96	**13.7**	4.5	7.4
Perlmutter GPU	Slingshot 10 x2	96	**23.8**	5.1	16.9
Cori KNL	Aries	1152	**82.1**	40.1	23.5

We picked the smallest number of Perlmutter GPU nodes that could ideally fit the simulation problem, i.e. 384 A100 40GB GPUs distributed over 96 nodes. We initially ran the simulation on the nodes that had the Slingshot 10 interconnect, and more recently on nodes that were upgraded with the Slingshot 11 interconnect. The measured run time for the initial 10 a/c_s simulation time is available in Table I. CGYRO is instrumented to categorize the time spent in various tasks, so in addition to the total time, we also provide the time spent in

compute-focused and communication portions of the code. Moreover, we also include the timing for a 1152 node Cori KNL simulation run, to showcase how the new system helps with fusion research. All numbers should be considered correct within a 10% error margin.

As can be seen, the 96 Perlmutter GPU nodes deliver almost a 8x reduction in compute times compared to the 1152 Cori KNL nodes. The original Slingshot 10 interconnect is still faster than the Aries networking, even with a drastic reduction in node count. Nevertheless, the communication speedup is significantly lower than the compute speedup, resulting in a more modest 3.4x speedup in overall simulation progress.

The upgraded Slingshot 11 interconnect on the Perlmutter GPU nodes provides an additional 2.3x speedup for CGYRO communication routines, resulting in an impressive 1.7x speedup in overall simulation progress compared to the same nodes before the upgrade. Compared to the Cori KNL 1152 nodes, the overall simulation progress speedup is now about 6x.

As a corollary, we want to point out that the fraction of time spent on communication is higher on Perlmutter GPU nodes compared to Cori KNL nodes, even with the upgraded interconnect.

III. Synthetic AllToAll benchmark results

Since CGYRO is so communication heavy, with the bulk of communication costs coming from MPI_AllToAll, we also created and ran a simple synthetic test program [8]. The synthetic test is completely communication-bound on MPI_AllToall, with just some token compute in between to avoid eventual compiler code elimination. Given the minimal compute requirements, we opted for presenting the results using the same identical setup on Cori and Perlmutter, namely 64 MPI ranks spread over 16 nodes. As can be seen from Table II, at the time of writing the latest Perlmutter's Slingshot 11 interconnect provides an impressive 15x throughput speedup compared to Cori's Aries.

TABLE II. Synthetic AllToAll runtime of 64 MPI ranks on 16 nodes

Machine Type	CPU	Runtime per step
Perlmutter GPU	Sligshot 11 x4	**35**
Perlmutter GPU	Slingshot 10 x2	**87**
Cori KNL	Aries	**510**

IV. Summary and conclusions

The fusion research community is heavily relying on NERSC HPC systems for its simulation needs. The new Perlmutter system undoubtedly provides a major compute capability upgrade compared to the previous ones, and the users of the popular fusion turbulence simulation tool CGYRO are eager to start using it.

The CGYRO problem decomposition makes it very communication-heavy, so we measured the run times of the new system using a representative benchmark simulation case. We observe that the simulation indeed progresses on Perlmutter orders of magnitude faster than on the previous-generation Cori system, with a major additional boost coming from the recent upgrade of its interconnect from 2x Slingshot 10 to 4x Slingshot 11 NICs per node.

The new NERSC Perlmutter system is a well-balanced HPC system that is very effective for CGYRO fusion simulation users. The NVIDIA GPUs are very effective at providing the compute needed by the fusion community, and pairing one HPE Cray Slingshot 11 NIC with every GPU provides the bandwidth that is needed when performing cutting-edge simulations.

Acknowledgment

This work was partially supported by the U.S. Department of Energy under awards DE-FG02-95ER54309, DE-FC02-06ER54873 (Edge Simulation Laboratory) and DE-SC0017992 (AToM SciDAC-4 project). An award of computer time was provided by the ALCC program. This research used resources of the National Energy Research Scientific Computing Center, which is an Office of Science User Facility supported under Contract DE-AC02-05CH11231.

References

[1] NERSC Documentation, "Using Perlmutter," online, accessed Aug. 5th 2022. https://docs.nersc.gov/systems/perlmutter/

[2] NERSC Documentation, "Perlmutter Interconnect," online, accessed Aug 5th 2022. https://docs.nersc.gov/systems/perlmutter/interconnect/

[3] NERSC Documentation, "GPU-Accelerated Compute Nodes," online, accessed Aug. 5th 2022. https://docs.nersc.gov/systems/perlmutter/system_details/#gpu-accelerated-compute-nodes

[4] J. Candy, E.A. Belli, and R.V. Bravenec, "A high-accuracy Eulerian gyrokinetic solver for collisional plasmas," Journal of Computational Physics, Vol. 324, pp. 73-9. (2016) https://doi.org/10.1016/j.jcp.2016.07.039

[5] J. Candy et al., "Multiscale-optimized plasma turbulence simulation on petascale architectures," Computers & Fluids, Vol. 188, pp. 125-135. (2019) https://doi.org/10.1016/j.compfluid.2019.04.016

[6] E. A. Belli, J. Candy, I. Sfiligoi, and F. Würthwein, "Comparing single-node and multi-node performance of an important fusion HPC code benchmark," In Practice and Experience in Advanced Research Computing (PEARC '22). Association for Computing Machinery, New York, NY, USA, Article 10, 1–4. (2022) https://doi.org/10.1145/3491418.3535130

[7] CGYRO_inputs, "sh04 input.cgyro," GitHub, online. https://github.com/scidac/atom-open-doc/blob/master/CGYRO_inputs/sh04/input.cgyro

[8] I. Sfiligoi, "cgyro_test_alltoall.F90," GitHub, online. (2022) https://github.com/scidac/atom-open-doc/blob/master/2022.11-SC22/cgyro_test_alltoall.F90