Directive-based, Fortran/C++ interoperable approach to GPU offloading of the high performance gyrokinetic turbulence code GENE-X

Jordy Trilaksono¹, Philipp Ulbi¹, Andreas Stegmeir¹, Frank Jenko¹,²

¹Max Planck Institute for Plasma Physics, Boltzmannstraße 2, 85748 Garching, Germany, ²University of Texas at Austin, Austin, TX 78712, USA

ABSTRACT
The achievement of high plasma confinement is the key to realize commercially attractive energy production by magnetic confinement fusion (MCF) devices. Turbulence plays a significant role in maintaining plasma confinement within MCF devices. The GENE-X code is based on an Eulerian (continuum) approach to the discretization of the five dimensional gyrokinetic equation that describes plasma turbulence. Our discretization is specialized to simulate plasma turbulence anywhere within MCF devices, from the hot plasma core to the cold wall. GENE-X is written in object-oriented modern Fortran 2008 leveraging MPI+OpenMP parallelization to facilitate large scale turbulence simulations. Here, we present our development efforts to further extend the parallelization scheme to GPUs, which is essential for scalability support towards simulations of larger, reactor-relevant fusion devices. The current implementation in GENE-X provides a proof of concept of our native Fortran/C++ interoperability approach by successfully supporting several GPU backends such as OpenACC, OpenMP offload and CUDA. We present first benchmarks of our directive-based OpenACC implementation of the most computationally expensive part of GENE-X. A significant performance increase was achieved on the GPU, compared to equivalent CPU benchmarks.

INTRODUCTION TO GENE-X

GENE-X [1] is a full gyrokinetic turbulence code written in Fortran 2008 with object-oriented design. GENE-X features electromagnetic field equations [2], collision models [5], and a flux-coordinate independent (FCi) [4] coordinate system. GENE-X shares an FCi-based mesh equilibrium and field solver library called PARALLAX with the Braginskii fluid code GEM [3].

GENE-X solves 5D arrays of distribution function consisting $R, Z, \varphi, \psi, \phi$ and species axes. GENE-X uses a new parallelization hierarchy: OpenMP for intra-node parallelization on CPUs MPI for inter-node parallelization across CPUs GPU: low latencies, best for wide-range of tasks GPU: more cores, high throughput for more repetitive tasks

GPU OFFLOADING STATUS

The arithmetic operators are experimented on to show the compatibility of the C++ layer to various GPU backend combinations, i.e. OpenACC, OpenMP offload and CUDA. The directive-based backend, i.e. OpenACC, is prioritized due to the multidimensional array of the distribution function. The following are the compiler combinations used on MPCDF machines (Cobra and Raven):

- Fortran: GNU compiler (gfortran)
- C++: NVIDIA HPC SDK compiler (nvcc++)

GPU offloading status of GENE-X

CONCLUSION AND OUTLOOK

GENE-X build configuration now supports Fortran/C++ hybrid model with mainly OpenACC and OpenMP offload as GPU backend. The directive-based approach is chosen due to maintainability factor and good affinity with the numerics of GENE-X. Here, the preliminary performance monitoring of the static and dynamic operators of gyrokinetic Vlasov equation are presented and showing promising speedups. Next milestones are:

- Multi-GPU implementation
- Latency analysis
- Kernel optimization

REFERENCES

Corresponding author: Jordy Trilaksono (jtrilaksono@iap.mpg.de); PASC 2023, Davos, Switzerland, June 26-28, 2023