A ARTIFACT DESCRIPTION APPENDIX: KERNEL-BASED AND TOTAL PERFORMANCE ANALYSIS OF CGYRO ON 4 LEADERSHIP SYSTEMS

A.1 Abstract

This appendix summarizes the steps required to obtain the CGYRO sources, build the executable, create the timing template case, run the code, and obtain the code timing. We also provide the git hash required to obtain the version of the code used to generate the performance data in the paper, and where to find an archive of that data.

A.2 Description

A.2.1 Check-list (artifact meta information).

- **Algorithm:** 5D gyrokinetic simulation, spectral (2 dimensions), pseudospectral (2 dimensions), finite-difference (1 dimension)
- **Program:** CGYRO Fortran binary
- **Compilation:** Fortran 2008, FFTW/cuFFT, BLAS, LAPACK, make, with compiler settings defined in "platform files"
- **Data set:** CGYRO test cases nl03 and nl04
- **Execution:** MPI run
- **Output:** Time to solution in `out.cgyro.timing`
- **Experiment workflow:** MPI run
- **Experiment customization:** number of MPI tasks, OMP threads, set small number of timesteps
- **Publicly available?:** Available on GitHub to registered users

A.2.2 How software can be obtained. Sources are maintained in the GitHub repository https://github.com/gafusion. The checksum for the version used in this paper (in a performance testing branch) appears below

```
$ git clone git@github.com:gafusion/gacode.git
$ git checkout performance
$ git checkout f7c73b
```

A.2.3 Hardware dependencies. GPUs are required for use with accelerated OpenACC/cuFFT kernels. Otherwise, there is no significant hardware dependency.

A.2.4 Software dependencies. Fortran, FFTW/cuFFT, BLAS and LAPACK are requirements.

A.2.5 Datasets. Code performance was measured using the built-in `nl03` and `nl04`. These can be generated from the command line via:

```
$ cgyro -g nl03
$ cgyro -g nl04
```

Timing results presented in this paper are stored in the https://github.com/scidac repository.

A.3 Installation

Clone repository

```
$ cd /mydir
$ git clone git@github.com:gafusion/gacode.git
```

Set platform (from available platforms) in `.bashrc`.

```
export GACODE_ROOT=/mydir/gacode
export GACODE_PLATFORM=TITAN_PGI
. ${GACODE_ROOT}/shared/bin/gacode_setup
```

Build executable

```
$ cd $GACODE_ROOT/cgyro
$ make
```

A.4 Experiment workflow

Create simulation directory and submit job

```
$ cgyro -g nl03
$ cd nl03
$ gacode_qsub -code cgyro -n 4096 -nomp 8 -s
```

Job file `batch.src` is created and submitted to queue. Other options can be added to `gacode_qsub`.

A.5 Evaluation and expected result

Job runs to completion in a few minutes, with output in `out.cgyro.timing` including the 8 kernel timings summarized in present paper.

```
$ cat out.cgyro.timing

 str        str_comm   nl         nl_comm
 3.956E+00  1.702E+00  3.236E+01  2.534E+01
 3.945E+00  1.675E+00  3.248E+01  2.468E+01
 3.942E+00  1.696E+00  3.243E+01  2.411E+01
 3.941E+00  1.673E+00  3.248E+01  2.409E+01

 field      field_comm coll       coll_comm
 4.462E+00  5.218E+00  5.503E-01  8.069E-01
 4.462E+00  5.108E+00  5.359E-01  8.003E-01
 4.460E+00  5.124E+00  5.358E-01  8.004E-01
 4.461E+00  5.147E+00  5.366E-01  8.159E-01
```

Results can be checked and plotted; for example Fig. 11:

```
$ cgyro_plot -plot kxky_phi
```

A.6 Experiment customization

Reduced number of timesteps to limit amount of compute time used on HPC systems. This does not meaningfully affect average timing results.