Compilation and Usage of the WAM6-GPU

The code package WAM6-GPU is accelerated by OpenACC based on the WAM Cycle 6 (refer to as WAM6 hereafter), a third-generation spectral ocean wave model extensively used across the EU states. The original source code is maintained by Dr. Arno Behrens (Helmholtz-Zentrum Hereon). Please refer to https://github.com/mywave/WAM for technical manual, usage and compiling instructions.

1. Before the compilation

The WAM6-GPU is an OpenACC version of the WAM6. Though considerable refactoring of the code has been made, the whole structure and algorithms of wave physics are the same as the original code package. Thus, before your compilation, make sure you are familiar with the WAM model, including its physics and setup parameters. The best practice is to download the original WAM code from https://github.com/mywave/WAM, read the relevant documents under the subfolder /documentation, compile and run the CPU code first.

the WAM6-GPU only support NVIDIA's GPU. To compile the WAM6-GPU, you need the following software packages:

- NVIDIA HPC SDK version 22 and above (with OpenMPI)
- NetCDF-C and NetCDF-fortran compiled with NVIDIA HPC SDK
- makedepf90: install with command 'conda install -c conda-forge makedepf90'
- NCVIEW (Optional, for visualization)
- NCCMP (Optional, for comparison of NetCDF files)

Download the WAM6-GPU code package, and uncompress, the wam6-gpu folder includes:

- /src: contains all source files and makefiles
- /src_cpu: the original CPU source files and makefiles
- /exe: compiled executables (preproc, wam, and pnetcdf
- /global_1p10: a global 1/10 degree hindcast case with topography file and one-day wind field provided (WAMgrid_coast.txt and wam_wind.dat)
- /documentation: a copy of the model descriptions from https://github.com/mywave/WAM.

Important note: Be familiar with the makefiles before doing anything! Makefiles of the WAM6-GPU is straightforward and easy to follow for normal user. 'Makefile' under the /src folder is the main makefile for the source code, which executes 3 sub-makefiles individually, including preprocessing grid and tables (Makefile_preproc), main program for WAM (Makefile_wam), and wave output in NETCDF (Makefile_pnetcdf). 'Makefile_userset' is an user-defined file for defining path of compiler and NetCDF library, and other compiling options. If you are familiar with the original WAM code from https://github.com/mywave/WAM, you may notice that the compilation process has been simplified. However, if you prefer that way of compilation, feel free to contact the corresponding author for it.

2. Compilation

The commands for compilation:

- in /src, type ‘make’: execuate the 3 sub-makefiles individually and generate the executables to folder /exe (to activate the OpenACC directives, you must include -acc option in ‘Makefile_userset’);
- in /src, tpye ‘mv ../exe/* ../global_1p10/’ : move the executables to the user’s case;

3. Model setup

There are 3 model setup files, including: Preproc_User, WAM_User, and NETCDF_User. The modification of the setup files refers to the documents provided by https://github.com/mywave/WAM. A copy of them are also available under /documentation.

There are 2 namelist files, including GPU_RESOURCE_SINGLENODE.NML and NAMELIST.NC. The former is to define available GPU resource in the current GPU server. This file is just valid for a single-node GPU server with multiple cards. If some of the GPU cards in the server are used by other users (for example, 4 cards with number from 0 to 3), you can specify the available card number (card number from 4-7) in this namelist. If GPU_RESOURCE_SINGLENODE.NML is not available, the program will not terminate, and bind the CPU processors (ranks) to the GPU cards with serial number from 0 in default.

The latter is to define monthly ERA5 wind forcing files. We provided a fortran code (read_wind_era5.f90) for reading the ERA5 reanalysis wind fields. Please note that this is not a part of the original WAM code. If you want to activate the code, please revise the code in terms of your requirement, and locate the commented line ‘CALL READ_WIND_ERA5’ in wam_wind_module.f90, then uncomment it. Please also remember to comment the line above ‘CALL READ_WIND_INPUT’.

4. Running the model

- in /global_1p10, type ‘nvidia-smi’: check how many GPU cards are installed, how many are available, and the serial number of the available cards;
- in /global_1p10, change GPU_RESOURCE_SINGLENODE.NML according to the information obtained by the previous command;
- in /global_1p10, type ‘mpirun -np 1 ./preproc’ to preprocess grid and tables;
- in /global_1p10, type ‘mpirun -np 4 ./wam’ to run the model (4 is number of NVIDIA card available)
- in /global_1p10, type ‘mpirun -np 1 ./pnetcdf’ to print wave output in NETCDF format.
- use ncview to check the output.

5. Something to know

- The initialization of the model has not ported to the GPU, as it is only executed one time. However, there is a work plan to do so in the near future by us. We don't encourage the user to do so except they are extremely familiar with the WAM6-GPU. When the main program ends successfully, 'TOTAL USER TIME IN SECONDS' shown in the screen or logfiles includes the time for initialization. You may find that the initialization time is not negligible compared to the model integration time.
- The WAM6-GPU has not been evaluated on an multi-node GPU cluster. As stated in the paper, the model now runs on an multi-GPU (8 cards), single-node GPU server. How inter-node communication will affect the model performance is not clear now.
- This GPU version has not included the current/flow refraction. However, it is not complicated to do so.
- Speedup ratios shown in the paper is based on a global case. We encourage using GPU for computing-demanding cases, which can exhibit the advantage of the GPU sufficiently.