NERSC Introduction

Parallelware Tools Training -- Motif Guided Parallelization of ZPIC with OpenMP and OpenACC

October 27, 2020

Yun (Helen) He
User Engagement Group
NERSC

Introduction

- Prepare users programming for upcoming Perlmutter
- Presenters from Appentra (in Spain)
  - Manuel Arenaz (CEO and Co-Founder)
  - Javier Novo Rodríguez (Head of Engineering and Product)
- Organizers
  - Appentra: Manuel Arenaz, Javier Novo Rodríguez, Fani Garcia
  - NERSC: Rebecca Hartman-Baker, Helen He, Woo-Sun Yang, Chris Daley, Brandon Cook

Parallelware Tools

- Parallelware tools (pwtrainer and pwanalyzer) are novel tools
  - for debugging and development of C/C++/Fortran parallel codes for multicore CPUs and GPUs using OpenMP and OpenACC
- This is the 3rd training at NERSC provided by Appentra
  - Two held in 2019, focused more on understanding parallel patterns (compute, memory, and flow) and parallelize using pwtrainer
  - This training builds up the tools knowledge first, then show a real code example for parallelization guided with common motifs in NESAP applications

Some Logistics (1)

- Users are muted upon joining Zoom (can unmute to speak)
- Please change your name in Zoom session
  - to: first_name last_name (nersc_user_name)
  - Click “Participants”, then “More” next to your name to rename
- Join Slack for Q&A and discussions
  - “general”, join “cpu”, and “gpu” channels
  - More channels will be created based on need
- Slides are uploaded. Videos will be available after post-processing
  - https://www.nersc.gov/users/training/events/parallelware-training-series-oct-nov-2020/

Some Logistics (2)

- Users are added to m3502 (or ntrain) project for Cori GPU access
  - Valid through Nov 10
- Some KNL and GPU nodes are reserved during the training
  - Oct 27: 10 am - 12 pm, pwtools1_knl, pwtools1_gpu
  - Oct 29: 9 am - 12 pm, pwtools2_knl, pwtools2_gpu
  - Nov 4: 8 am - 1 pm, pwtools3_knl, pwtools3_gpu
- ssh -Y login_name@cori.nersc.gov
- Recommend to use NX to expedite X-forwarding
  - Necessary for using pwtrainer (GUI) from remote
  - Instructions: https://docs.nersc.gov/connect/nx
- Materials (slides, homework, study materials) available on Cori
  - % cd $SCRATCH
  - % cp -r /global/cfs/cdirs/training/2020/parallelware . (notice the dot)

Agenda

Part, Date & Time	Topic and Format
Part 1, Tuesday, October 27 8:30 am - 12:00 pm PDT	**Introduction to Parallelware tools: Ensuring parallel programming best practices** • Introduction • NESAP Applications & Motifs • ZPIC code: Particle-in-Cell (PIC) method • Development of parallel codes with best practice recommendations • Parallelware tools: Trainer & Analyzer **Format**: Remote lectures, demos, and take-home exercises
Part 2, Thursday, October 29 9:00 am - 12:00 pm PDT	**Office hours** • Homework exercises demo (first hour) • Support, Questions, FAQs for using Parallelware tools **Format**: Remote office hours
Part 3, Wednesday, November 4 8:00 am - 1:00 pm PST	**Guided parallelization of ZPIC: Ensuring best practices with Parallelware tools** • Case study: Guided parallelization of ZPIC with Parallelware tools • Performance evaluation of ZPIC • Bring your own applications **Format**: Remote demos and hands-on

Intro lectures (with several 5-min breaks)

homework assignment with step-by-step guides

homework demo and office hour

case study and bring own application

OpenMP on Cori GPU -- LLVM Compiler

C/C++ code only

From a Cori login node via SSH or NX

```
% module purge
% module load cgpu
```

Get on a gpu node via salloc with reservation

```
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 --reservation=pwtools1_gpu -q shared
```

or without reservation

```
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 -q interactive
```

Land on a compute node to build and run

```
% module load PrgEnv-llvm/11.0.0-git_20200409
% clang -fopenmp -fopenmp-targets=nvptx64-nvidia-cuda -Ofast -o mycode.exe mycode.c -lm
% srun -n 1 ./mycode.exe
```

OpenMP on Cori GPU -- GCC Compiler

From a Cori login node via SSH or NX
% module purge
% module load cgpu

Get on a gpu node via salloc with reservation
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 --reservation=pwtools1_gpu -q shared
or without reservation
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 -q interactive

Land on a compute node to build and run
% module load cuda gcc/8.1.1-openacc-gcc-8-branch-20190215
% gcc -fopenmp -foffload=nvptx-none="-Ofast -lm -misa=sm_35" -Ofast -o mycode.exe mycode.c -lm
% gfortran -fopenmp -foffload=nvptx-none="-Ofast -lm -misa=sm_35" -Ofast -o mycode.exe mycode.F90 -lm
% srun -n 1 ./mycode.exe

OpenACC on Cori GPU -- HPCSDK Compiler

From a Cori login node via SSH or NX:

```
% module purge
% module load cgpu

Get on a gpu node via salloc with reservation
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 --reservation=pwtools1_gpu -q shared
or without reservation
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 -q interactive

Land on a compute node to build and run
% module load hpcsdk/20.7
% nvc -gpu=cc70,cuda11.0 -acc -fast -o mycode.exe mycode.c
% nvfortran -gpu=cc70,cuda11.0 -acc -fast -o mycode.exe mycode.F90
% srun -n 1 ./mycode.exe
```

OpenACC on Cori GPU -- GCC Compiler

From a Cori login node via SSH or NX:

```
% module purge
% module load cgpu
```

Get on a gpu node via salloc with reservation

```
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 --reservation=pwtools1_gpu -q shared
```

or without reservation

```
% salloc -N 1 -C gpu -c 10 -G 1 -t 1:00:00 -A m3502 -q interactive
```

Land on a compute node to build and run

```
% module load cuda gcc/8.1.1-openacc-gcc-8-branch-20190215
% gcc -fopenacc -foffload=nvptx-none="-Ofast -lm -misa=sm_35" -O3 -o mycode.exe mycode.c -lm
% gfortran -fopenacc -foffload=nvptx-none="-Ofast -lm -misa=sm_35" -O3 -o mycode.exe mycode.F90 -lm
% srun -n 1 ./mycode.exe
```

OpenMP on Cori CPU -- Intel Compiler

From a Cori login node via SSH or NX:
<do not run module purge and module load cgpu>

Build on a login node
% module swap craype-haswell craype-mic-knl (if optimize for KNL)
% cc -qopenmp -Ofast -o mycode.exe mycode.c
% CC -qopenmp -Ofast -o mycode.exe mycode.cc
% ftn -qopenmp -Ofast -o mycode.exe mycode.F90

Get on a KNL node via salloc with reservation
% salloc -N 1 -C knl -t 1:00:00 -A m3502 --reservation=pwtools1_knl
or without reservation
% salloc -N 1 -C knl -t 1:00:00 -A m3502 -q interactive

Run on a compute node
% export OMP_NUM_THREADS=4 (or other values)
% ./mycode.exe (pure OpenMP code)
% srun -n xx -c xx --cpu-bind=cores ./mycode.exe (MPI/OpenMP code)

OpenMP on Cori CPU -- GCC Compiler

From a Cori login node via SSH or NX:
<do not run module purge and module load cgpu>

Build on a login node
% module swap PrgEnv-intel PrgEnv-gnu
% module swap craype-haswell craype-mic-knl (optimize for KNL)
% cc -fopenmp -Ofast -o mycode.exe mycode.c
% CC -fopenmp -Ofast -o mycode.exe mycode.cc
% ftn -fopenmp -Ofast -o mycode.exe mycode.F90

Get on a KNL node via salloc with reservation
% salloc -N 1 -C knl -t 1:00:00 -A m3502 --reservation=pwtools1_knl
or without reservation
% salloc -N 1 -C knl -t 1:00:00 -A m3502 -q interactive

Run on a compute node
% export OMP_NUM_THREADS=4 (or other values)
% ./mycode.exe (pure OpenMP code)
% srun -n xx -c xx --cpu-bind=cores ./mycode.exe (MPI/OpenMP code)

OpenMP on Cori CPU -- Cray Compiler

From a Cori login node via SSH or NX:
<do not run module purge and module load cgpu>

Build on a login node
% module swap PrgEnv-intel PrgEnv-cray
% module swap craype-haswell craype-mic-knl (optimize for KNL)
% cc -fopenmp -Ofast -o mycode.exe mycode.c
% CC -fopenmp -Ofast -o mycode.exe mycode.cc
% ftn -homp -O3 -o mycode.exe mycode.F90 (notice usage is different for Fortran)

Get on a KNL node via salloc with reservation
% salloc -N 1 -C knl -t 1:00:00 -A m3502 --reservation=pwtools1_knl
or without reservation
% salloc -N 1 -C knl -t 1:00:00 -A m3502 -q interactive

Run on a compute node
% export OMP_NUM_THREADS=4 (or other values)
% ./mycode.exe (pure OpenMP code)
% srun -n xx -c xx --cpu-bind=cores ./mycode.exe (MPI/OpenMP code)

Using pwtrainer and pwanalyzer at NERSC

From within an salloc session on CPU or GPU (as shown in the previous slides)

- % module load pwtrainer
- % pwtrainer &

- % module load pwanalyzer
- % pwreport <options> or pwcheck, pwloops, pwdirectives <...>

- Can use CPU for most of preparation work for GPU
- Can use any compiler, such as gcc for homework, and other compilers (recommended for better performance) for own applications
  - OpenMP on GPU: LLVM, gcc
  - OpenACC on GPU: hpcsdk, gcc
  - OpenMP on CPU: Intel, gcc, cray

Study Materials - helpful for homework

- See /global/cfs/cdirs/training/2020/parallelware/study-materials/README
- Pwanalyzer Quick Start Guide using NPB
  - https://www.appentra.com/parallelware-analyzer-npb-quickstart/
  - Full Quick Start sheet and NPB codes available on Cori
- Four 5-min videos to quickly learn how to use pwtrainer
  - https://www.appentra.com/training/courses/using-parallelware-trainer/
- 4 tips to_avoid_race_conditions_on_GPUs
- Help menu for tools commands
  - such as “pwanalyzer --help”, “pwdirectives --help”
- docs and examples in pwtrainer and pwanalyzer installation directory on Cori

pwtrainer “docs” and “examples”

```
yunhe@cori11:~> module show pwtrainer
-------------------------------------------------------------------Intro lectures (with several 5-min breaks)

/global/common/software/nersc/cle7/extra_modulefiles/pwtrainer/1.5.0:

conflict  pwtrainer
prepend-path    PATH /global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64
prepend-path    LD_LIBRARY_PATH /global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64/lib
setenv          PWT_LICENSE_PATH
/global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64/../licensing/pwtrainer.lic
-------------------------------------------------------------------

yunhe@cori11:~> cd /global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64
yunhe@cori11:/global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64> ls
ChangeLog.txt  EULA.txt  Third_Party_Licenses.txt  data  docs  lib  plugins  pwtrainer  qt.conf
yunhe@cori11:/global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64> cd docs
yunhe@cori11:/global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64/docs> ls
UserManual.pdf  examples
yunhe@cori11:/global/common/cori_cle7/software/pwtrainer/pwtrainer-1.5.0_linux-x64/docs>
```

can find docs and examples similarly for pwanalyzer

Thank You!