

Status of OpenACC and OpenMP

Meifeng Lin

Brookhaven National Laboratory

USQCD Software Meeting

April 27, 2019, BNL

OpenACC

- OpenACC = Open Accelerators
- Compiler directives; supports both multithreading on CPUs, and accelerator offloading
- Descriptive: Compilers analyze the code and generate parallelization strategies at compile time
 - Examples: `#pragma acc kernels`, `#pragma acc parallel loop`
- Developers can also be specific:
 - Example: `#pragma acc parallel loop gang (number_of_gangs) worker (number_of_workers) vector (length_of_vector)`
- Current version: OpenACC 2.7 (Nov 2018)
 - Manual deep copy
 - Host as device, serial construct, ...
- **Main compiler: NVIDIA PGI**

OpenMP Recap

- OpenMP 3.x: Popular multithreading programming model for CPUs
 - Compiler directives; supported by most C/C++ and Fortran compilers
 - Easy to use: `#pragma omp parallel`
 - Portable across different CPU architectures
- New in OpenMP 4.x: accelerator target offloading
- Comparison of OpenACC and OpenMP offloading

OpenACC	<pre>#pragma acc parallel loop independent copyin(expr[0:1]) for(int ss=0;ss<_osites;ss++){ _odata[ss] = eval(ss,expr); }</pre>
OpenMP	<pre>#pragma omp target device(0) map(to: expr) map(tofrom:_odata[0:_osites]) { #pragma omp teams distribute parallel for { for (int i=0; i<_osites; i++) _odata[ss] = eval(ss,expr); } }</pre>

“OPENMP 5.0 IS A MAJOR LEAP FORWARD”

- **“Full support for accelerator devices.** OpenMP now has full support for accelerator devices, including **mechanisms to require unified shared memory** between the host system and coprocessor devices, the ability to use device-specific function implementations, better control of implicit data mappings, and the ability to override device offload at runtime. In addition, it supports reverse offload, implicit function generation, and **the ability to copy object-oriented data structures easily.”**
- **“Support for the latest versions of C, C++, and Fortran.** OpenMP now supports important features of Fortran 2008, C11, and C++17.”
- **Of course, compilers will need time to catch up**

Main OpenMP 4.5+ Compilers

- **IBM XL Compilers:**

- <https://www.ibm.com/us-en/marketplace/xl-cpp-linux-compiler-power>

- XL C/C++ for Linux V16.1.1 and XL Fortran for Linux V16.1.1 fully support OpenMP 4.5 features including the target constructs.
 - Compile with `-qsmp=omp` to enable OpenMP directives and with `-qoffload` for offloading the target regions to GPUs.

- Available on Summit

- **Clang 9:** <https://clang.llvm.org/docs/OpenMPsupport.html> (should have full OpenMP 4.5 support, and already partial OpenMP 5 support)

- Support OpenMP 4.5
 - Supports offloading to X86_64, AArch64, PPC64, basic support for NVIDIA GPUs

Other OpenMP Compilers

- Clang 4.0: <https://github.com/clang-ykt/clang/wiki>
 - Supports OpenMP 4.5 offload for NVIDIA GPUs
 - Need to build from source
 - Got merged into the main clang (See Clang 9)
- Cray CCE 8.7 (April 2019):
<https://pubs.cray.com/content/S-2179/8.7/cray-c-and-c++-reference-manual/openmp-overview>
 - Support OpenMP 4.5 with a few exceptions
 - Will switch to clang as well
- Perlmutter scheduled to be installed at NERSC in 2020
 - AMD CPUs + NVIDIA next-gen GPUs
 - **NVIDIA PGI to support OpenMP offloading as part of the Perlmutter contract**
- [GCC 9](#) to support OpenMP 5

OpenMP Offloading in Grid?

- Grid is challenging for OpenMP/OpenACC offloading
- Nested/abstracted data structures not easy for accelerator data management
- Requires true **deep copy** or **unified virtual memory** (UVM) support
- OpenACC+PGI UVM support worked to some degree (Boyle et al., [arXiv:1710.09409](https://arxiv.org/abs/1710.09409))
 - `pgc++ -acc -ta=tesla:managed --c++11 -O3 main.cc -o gpu.x`
 - Also possible with manual deep copy
- OpenMP 5 adds UVM and deep copy support. **Compiler support is not complete yet.**
- **Lambda function offloading**
 - Under investigation/consideration by OpenACC
 - Already supported in OpenMP 5.

Summary

- OpenACC and OpenMP are adding support for more C++-friendly accelerator offloading
- OpenACC features likely to be merged into future OpenMP spec
- When compilers catch up, OpenMP 5.x may be promising for QCD accelerator offloading, especially for simpler C/C++ codes

- OpenMP for Grid: work in progress
 - Task at the OpenMP Hackathon next week at BNL