Experience OpenMP4.5

Compute Device Offloading – an overview

Dr. Thilo Maurer, IBM Research & Development Böblingen GmbH
Tuesday, October 18th 2016, PADC Workshop at Jülich Supercomputing Centre

IBM

Contents

- OpenMP – What is it?
- OpenMP 4.5 – What has changed?
- OpenMP 4.5 or OpenACC 2.5 – What is different?
- LLVM C/C++ compiler with OpenMP 4.5 support
- Example: Compute Device Enablement of Gadget4

Open Multi-Processing

- Pragma Language for easy multi-threading
- Shared-memory multiprocessing API for Fortran, C, and C++
- Standardizes task & loop-level parallelism, driven by consortium of large hardware & software vendors
- Supports coarse-grained parallelism by annotations specifying sections of sequential code to be executed by multiple concurrent threads
- Combines serial and parallel code in single expression sequence
- Actively used for > 20 years
  - OpenMP 1.0 (Oct 1997) … OpenMP 4.5 (Nov 2015)

A Brief History of OpenMP

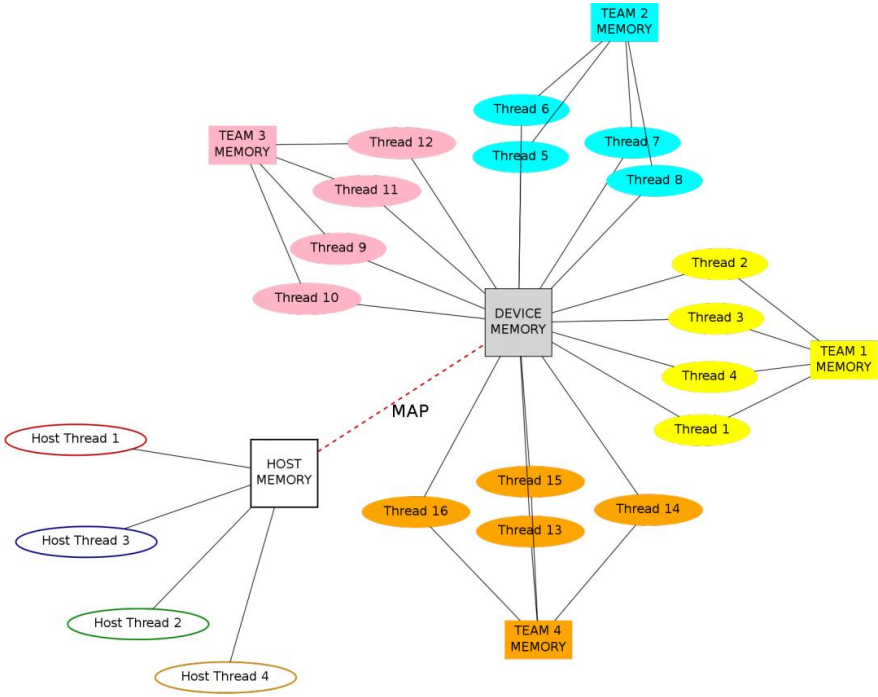
- 1996 - Architecture Review Board (ARB) formed by several vendors implementing
- their own directives for Shared Memory Parallelism (SMP).
- 1997-1998 – Versions 1 – first shot
- 2000-2002 – Versions 2 – gaps filled
- 2005 – Version 2.5 – C/C++ and FORTRAN specs merged
- 2008-2011 – Version 3.x – tasking
- 2013-2015 – Version 4.x – offloading

OpenMP 4.0 – in July 2013

- Significant Update to OpenMP
  - Support for compute devices (aka accelarators)
  - SIMD over loops and functions
    `simd` contruct, `declare simd` contruct
  - Native thread-affinity
    `OMP_PLACES` environment variable, `proc_bind` clause
  - Deep Tasking
    `taskgroup` contruct, `depend` clause
  - User-defined reductions
    `declare reduction` contruct, `depend` clause
  - Enhanced support for atomics
    `capture` clause, `seq_cst` clause
  - Support for Fortran 2003

OpenMP 4.0 – Compute Devices

- Compute device usually has ist own
  - Shared memory
  - Compute resources (e.g. threads blocks)
- Memory Model
  - Host and device memory usually distinct
    - Necessity for device data environment
  - Host and device memory may be shared
- Processing
  - Usual story on host
  - Threads on device bundeled into teams



Drawing ©Ramses van Zon
https://wiki.scinet.utoronto.ca/wiki/images/9/9b/Ds-openmp.pdf

OpenMP 4.0 – Compute Devices

- Explicit mappings of host memory to device memory
  - Limited lifetime

```
omp target data map(to:params,k,
                    from:output[0:k],
                    tofrom:input[0:k],
                    alloc:temp[0:k]) { … }
```

  - Unlimited lifetime

```
omp declare target
…
omp end declare target
```

  - Data synchronisation during lifetime

```
omp target update to(params) from(k)
```

- Offloading calculations
  - Mark region to execute on device

```
omp target { … }
```

  - Create a league of thread teams

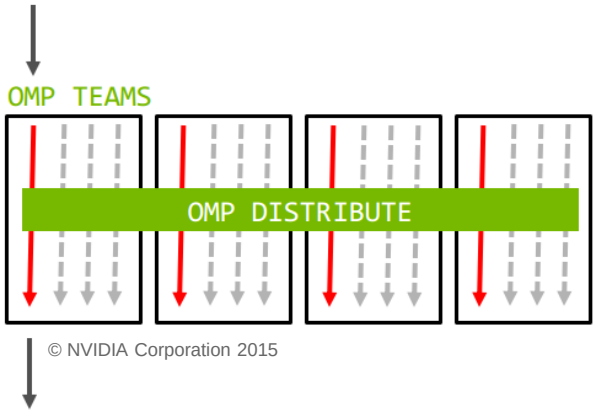
```
omp teams num_teams(…) { … }
```

  - Distribute loop(s) over teams

```
omp distribute { … }
```

  - Assign adjacent threads to adjacent loop interations

```
omp parallel for schedule(static,1) { … }
```



- Offloading to multiple devices supported natively:
  `device` clause available

OpenMP 4.5 – in November 2015

- Taskloop – handy for tasking multiple collapsed loops – `num_tasks` or `grainsize`
- Tasks: `priority` clause
- Doacross parallelism: `ordered` clause improved with explicit source&sink dependencies
- Offloading using `target` regions
  - Scalars default to `firstprivate`
  - Pointers mapped to device pointers (not treated as scalars)
    - Consequencly add `use_device_ptr` clause for CUDA compatibility
  - Asynchronous offloading using `nowait` and waiting for completion using `depend` (already possible with 4.0 using host tasking)
  - Floating target data lifetimes using `target enter data` and `target exit data` constructs
- Other smaller changes and clarifications

OpenMP vs OpenACC

Directives for Accelerators

Philosophical Differences

- More Explicit
- User-directed parallelism
  Host-centric execution model
- Compiler has less performance responsibility
- User adds additional directives to exclude parallelization issues
- Different architectures require different directives

- More Implicit
- User-guided parallelism
  Not necessarily host-centric
- Compiler has more performance responsibility
- Users write code actually free of parallelization issues
- Higher-level directives allow same-code targeting of different architectures

OpenMP vs OpenACC

Directives for Accelerators

Technical Differences

- `#pragma omp parallel`
  - Creates a single team of threads
  - User chooses thread count
  - Some communication between threads
  - Data races are the user‘s responsibility

- `#pragma omp teams distribute …`
  - Creates teams of threads
  - Compiler chooses counts of teams & threads & simd lanes
  - No communication between teams
  - Data races lead to malfunction
  - Only in target regions

- `#pragma acc parallel`
  - Creates gangs of workers
  - Compiler chooses counts of gangs & workers & threads
  - No communicaition between gangs
  - Data races forbidden

- `#pragma acc kernels`
  - Compiler does all the hard work

OpenMP vs OpenACC

Directives for Accelerators

Compiler Availability – as of year-end 2016

- OpenMP 4.5
  - LLVM C/C++ can offload to Nvidia Tesla & Intel XeonPhi
  - GCC 6 can offload to Intel XeonPhi & AMD HSA-IL
  - Intel C/C++ can offload to Intel XeonPhi & Host
- OpenMP 4.0
  - PathScale ENZO can offload to NVIDIA Tesla & AMD FirePro

- OpenACC 2.0a
  - GCC 6 can offload to Nvidia Tesla (conflicts with OpenMP)
  - PathScale ENZO can offload to Nvidia Tesla & AMD FirePro
- OpenACC 2.5
  - PGI can offload to Nvidia Tesla
  - PGI can offload to Host / MultiCore
  - PGI no conflict with host OpenMP

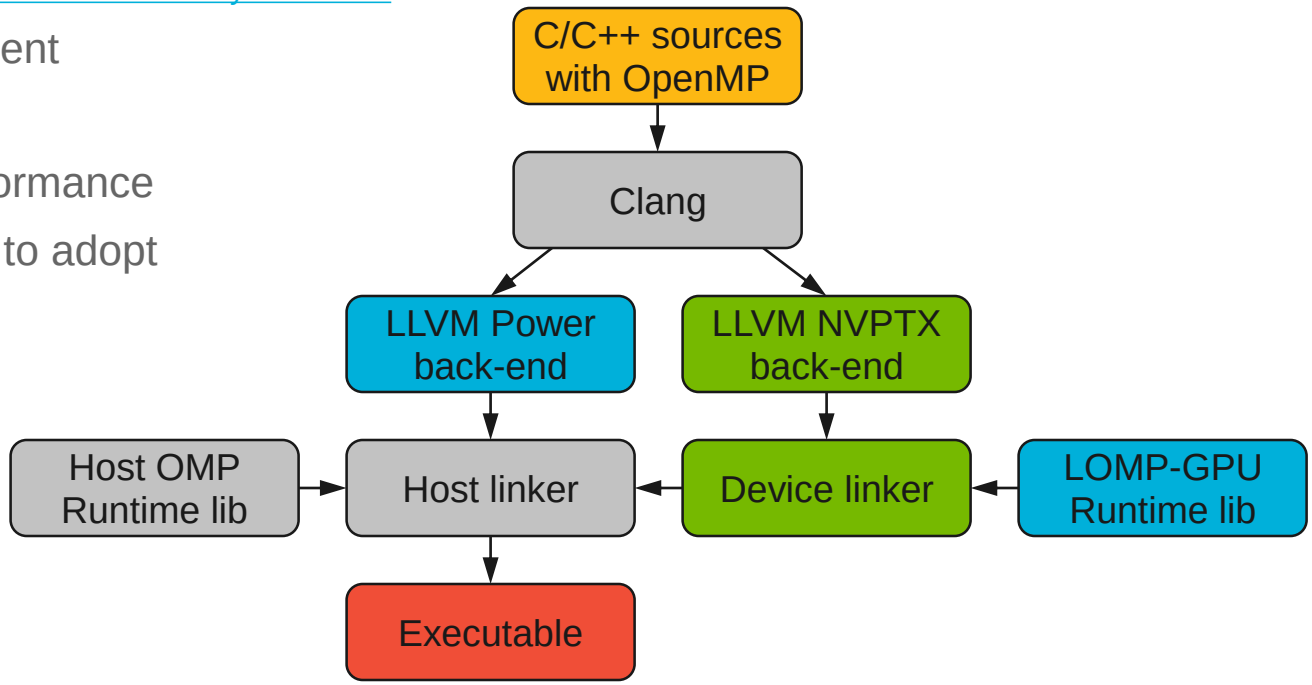
Caveats when targeting GPUs

- OpenMP 4.5
  - Map of static allocated fields excluded.
  - (In)compatible with Unified (Virtual) Memory
  - Asynchronous device computation but no control over GPU-streams
  - No native device-to-device communication
- OpenACC 2.5
  - Multi-device support only via explicit use of executable pragma `set device`
  - No native device-to-device communication

- If you know more, please let me know

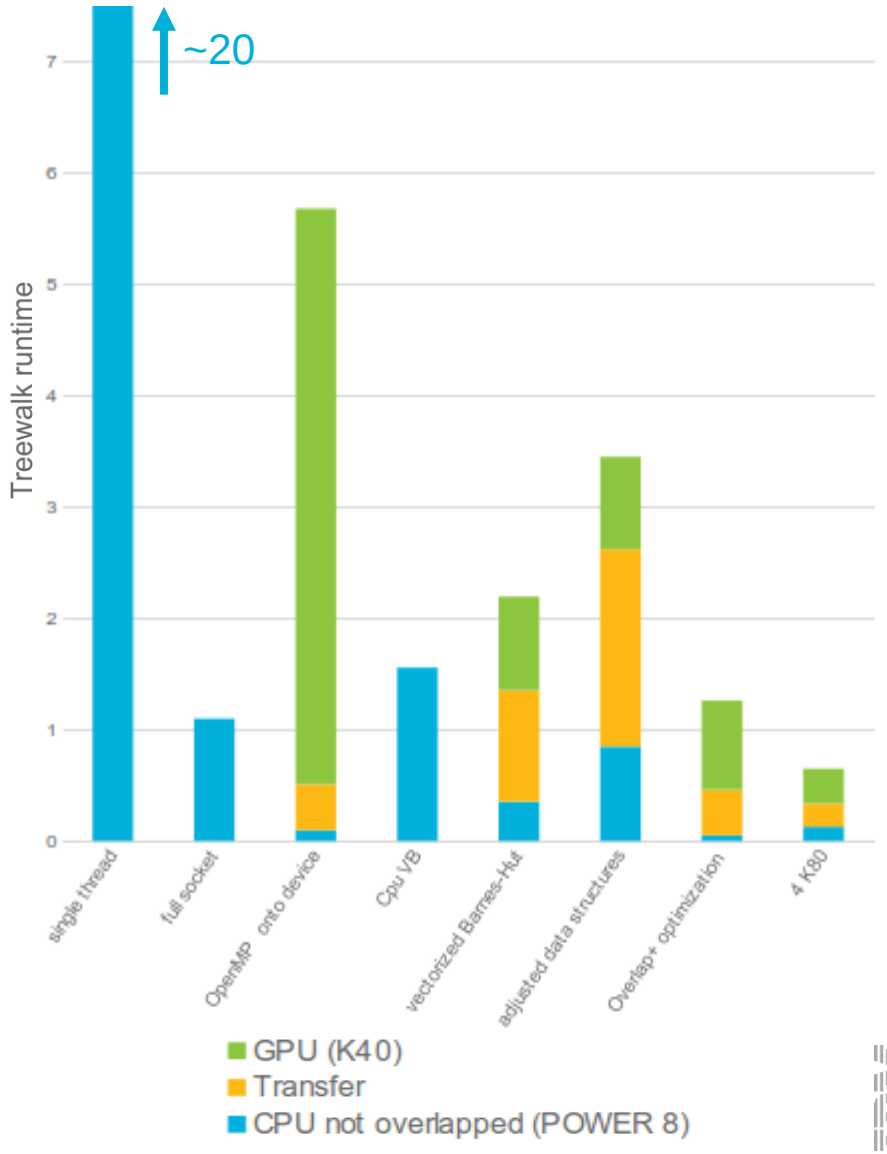
LLVM C/C++ Compiler with OpenMP4.5

- Free & OpenSource
- Try Now: https://www.ibm.com/developerworks/community/blogs/8e0d7b52-b996-424b-bb33-345205594e0d/entry/installation
- Ongoing development
- Power back-end
  - Preliminary performance
  - XL C/C++ going to adopt methodology



Compute Device Enablement of Gadget4

- Gadget4
  - App for cosmological N-body/SPH simulations
  - Part of the PRACE Unified European Applications Benchmark Suite
  - Initiated by Volker Springel at the Uni-versity of Heidelberg, v4 in development
  - MPI+OpenMP parallelized
  - v4: CUDA/OpenCL for costly parts
- This work: v4 extension
  - Proof of Concept: Vectorised Barnes Hut
    - Tree walk on host
    - Force calculation on device
    - Employed group-size determines work-share
  - Offloading to compute device using OpenMP/OpenACC



References

- Presentations
  - http://on-demand.gputechconf.com/gtc/2016/presentation/s6510-jeff-larkin-targeting-gpus-openmp.pdf
  - http://on-demand.gputechconf.com/gtc/2016/presentation/s6709-michael-wolfe-openacc-for-gpus-open-power.pdf
  - http://on-demand.gputechconf.com/gtc/2016/presentation/s6410-jeff-larkin-beyer-comparing-open-acc-openmp.pdf
  - https://wiki.scinet.utoronto.ca/wiki/images/9/9b/Ds-openmp.pdf

- OpenMP compilers for CORAL/OpenPOWER Heterogeneous Systems
  - IBM XL http://openpowerfoundation.org/presentations/openmp-accelerator-support-for-gpu/
  - LLVM
    - https://www.ibm.com/developerworks/community/blogs/8e0d7b52-b996-424b-bb33-345205594e0d/entry/installation
    - http://on-demand.gputechconf.com/gtc/2016/presentation/s6240-arpith-jacob-programming-using-Openmp.pdf

- Specifications
  - OpenMP 4.5 http://www.openmp.org/mp-documents/openmp-4.5.pdf
  - OpenACC 2.5 http://www.openacc.org/sites/default/files/OpenACC_2pt5.pdf

Important Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the act of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Trademarks

IBM, the IBM logo, ibm.com, AIX, DB2 and BLUEMIX are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their respective occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM mat the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Windows is a trademark of Microsoft Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. SAP. R/3 and all other SAP product and service names mentioned herein are trademarks or registered trademarks of SAP AG in Germany and several other countries.

Other company, product, or service names may be trademarks or service marks of others.