JURECA DC Evaluation Platform: AMD Instinct MI250

JSC AMD GPU Porting Workshop

16 May 2022 | A. Herten, K. Haghighi-Mood, S. Achilles | Jülich Supercompting Centre, Forschungszentrum Jülich

JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE

Outline

Node Overview

- 2 nodes, as part of JURECA DC
- **CPU**: AMD EPYC 7443 processor (*Milan*); 2 sockets, 24 cores per socket, SMT-2 (total: $2 \times 24 \times 2 = 96$ threads) in NPS-4 configuration
- **Memory**: 512 GB DDR4-3200 RAM (256 GB per socket; 8 memory channels per socket (2 channels per NUMA domain))
- **GPU**: 4$\times$ AMD Instinct MI250 GPUs (each 128 GB memory); shown as 8 GPUs with 64 GB GB memory each
- **Network**: 1$\times$ Mellanox HDR InfiniBand ConnectX 6 (100 Gbit$/$s) (*preliminary*)

GPU Status and Configuration via SMI

`rocm-smi`

```
================================ Concise Info ================================
GPU  Temp   AvgPwr  SCLK    MCLK     Fan  Perf  PwrCap  VRAM%  GPU%
0    34.0c  77.0W   800Mhz  1600Mhz  0%   auto  500.0W    0%   0%
1    39.0c  N/A     800Mhz  1600Mhz  0%   auto  0.0W      0%   0%
2    35.0c  115.0W  800Mhz  1600Mhz  0%   auto  500.0W    0%   0%
3    31.0c  N/A     800Mhz  1600Mhz  0%   auto  0.0W      0%   0%
4    31.0c  92.0W   800Mhz  1600Mhz  0%   auto  500.0W    0%   0%
5    33.0c  N/A     800Mhz  1600Mhz  0%   auto  0.0W      0%   0%
6    33.0c  93.0W   800Mhz  1600Mhz  0%   auto  500.0W    0%   0%
7    36.0c  N/A     800Mhz  1600Mhz  0%   auto  0.0W      0%   0%
==============================================================================
```

GPU Status and Configuration via SMI

`rocm-smi --showtopo`

```
========================== Weight between two GPUs ===========================
       GPU0       GPU1       GPU2       GPU3       GPU4       GPU5       GPU6       GPU7
GPU0   0          15         30         30         15         30         15         30
GPU1   15         0          30         15         30         45         30         15
GPU2   30         30         0          15         15         30         15         30
GPU3   30         15         15         0          30         15         30         45
GPU4   15         30         15         30         0          15         30         30
GPU5   30         45         30         15         15         0          30         15
GPU6   15         30         15         30         30         30         0          15
GPU7   30         15         30         45         30         15         15         0
```

GPU Status and Configuration via SMI

`rocm-smi --showtopo`

```
============================ Hops between two GPUs ============================
       GPU0         GPU1         GPU2         GPU3         GPU4         GPU5         GPU6         GPU7
GPU0   0            1            1            1            1            1            1            1
GPU1   1            0            1            1            1            1            1            1
GPU2   1            1            0            1            1            1            1            1
GPU3   1            1            1            0            1            1            1            1
GPU4   1            1            1            1            0            1            1            1
GPU5   1            1            1            1            1            0            1            1
GPU6   1            1            1            1            1            1            0            1
GPU7   1            1            1            1            1            1            1            0
```

GPU Status and Configuration via SMI

`rocm-smi --showtopo`

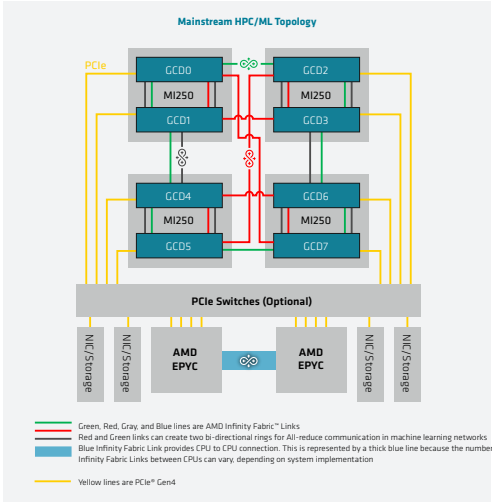
```
========================== Link Type between two GPUs ==========================
       GPU0         GPU1         GPU2         GPU3         GPU4         GPU5         GPU6         GPU7
GPU0   0            XGMI         XGMI         XGMI         XGMI         XGMI         XGMI         XGMI
GPU1   XGMI         0            XGMI         XGMI         XGMI         XGMI         XGMI         XGMI
GPU2   XGMI         XGMI         0            XGMI         XGMI         XGMI         XGMI         XGMI
GPU3   XGMI         XGMI         XGMI         0            XGMI         XGMI         XGMI         XGMI
GPU4   XGMI         XGMI         XGMI         XGMI         0            XGMI         XGMI         XGMI
GPU5   XGMI         XGMI         XGMI         XGMI         XGMI         0            XGMI         XGMI
GPU6   XGMI         XGMI         XGMI         XGMI         XGMI         XGMI         0            XGMI
GPU7   XGMI         XGMI         XGMI         XGMI         XGMI         XGMI         XGMI         0
```

GPU Status and Configuration via SMI

`rocm-smi --showtopo`

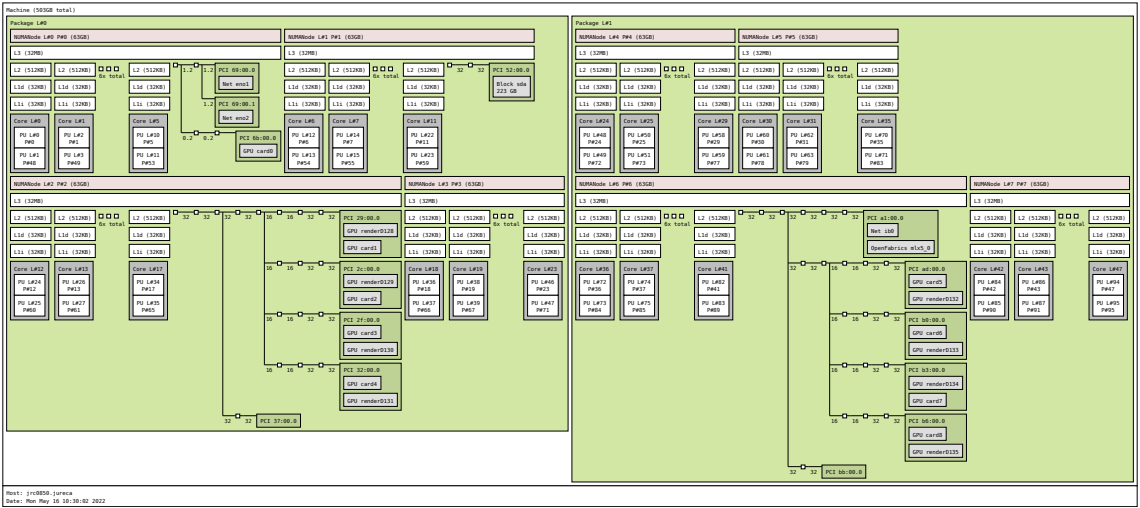
```
================================== Numa Nodes ==================================
GPU[0]  : (Topology) Numa Node: 2
GPU[0]  : (Topology) Numa Affinity: 2
GPU[1]  : (Topology) Numa Node: 2
GPU[1]  : (Topology) Numa Affinity: 2
GPU[2]  : (Topology) Numa Node: 2
GPU[2]  : (Topology) Numa Affinity: 2
GPU[3]  : (Topology) Numa Node: 2
GPU[3]  : (Topology) Numa Affinity: 2
GPU[4]  : (Topology) Numa Node: 6
GPU[4]  : (Topology) Numa Affinity: 6
GPU[5]  : (Topology) Numa Node: 6
GPU[5]  : (Topology) Numa Affinity: 6
GPU[6]  : (Topology) Numa Node: 6
GPU[6]  : (Topology) Numa Affinity: 6
GPU[7]  : (Topology) Numa Node: 6
GPU[7]  : (Topology) Numa Affinity: 6
```

CPU-GPU Topolgy



Source: AMD CDNA 2 Whitepaper, page 7.

lstopo



Compilation, Running

Compilation for AMD GPUs

- Only 2 MI250 nodes: Please don't compile on backend nodes!
- Use container on login nodes
- Prefix every command with `apptainer exec rocm-singularity.sif`
- Ignore the *Perl locale* warning
- Container has patched `hipMallocManaged()`

```
$ apptainer exec /p/project/training2215/container/rocm-singularity.sif \
    hipcc file.cpp -o out.exe
$ alias hipcc="apptainer exec /p/project/training2215/container/rocm-singularity.sif hipcc"
$ hipcc file.cpp -o out.exe
```

Using MI200 Parition

- Nodes part of `dc-mi200` partition of JURECA DC
- Use `training2215` budget

```
$ srun --nodes 1 --partition dc-mi200 --time 2 --account training2215
```

OpenACC on AMD GPUs

Converting OpenACC-C to OpenMP with Intel's Tool

- No solution from AMD to run OpenACC on AMD GPUs
- Source-to-source conversion only for OpenACC Fortran
- Options
  1. GCC
  2. Clang/Clacc
  3. Intel's Migration Tool

Intel Migration Tool

- Intel tool, mainly for their platform
- But converts to OpenMP, so works on AMD GPUs as well

→ github.com/intel/
intel-application-migration-tool-for-openacc-to-openmp

```
$ module use /p/project/training2215/tools/moduleFiles/
$ module load intel-acc-to-omp/1
$ intel-acc-to-omp poisson2d.c
```

Clacc/Flacc

- Research compiler, built on Clang
- Internal OpenMP translation
- C/C++ (Clacc), Fortran (Flacc)
- See examples at `/p/project/training2215/tools/clacc-flang-examples`

→ csmd.ornl.gov/project/clacc

```
$ module use /p/project/training2215/tools/moduleFiles/
$ module load flacc-flang/15.0.0
$ clang -fopenacc -fopenmp-targets=amdgcn-amd-amdhsa poisson2d.c
$ flang -fopenacc -fopenmp-targets=amdgcn-amd-amdhsa poisson2d.f90
```

Known Issues

Known Issues and Caveats

- Unified Memory is buggy in Linux Kernel version running on the nodes; using it will crash node
  - Please don't use `hipMallocManaged()` and similar
  - Please don't use library features which look like they could use UM
  - We patched container to error out when trying to use `hipMallocManaged()`
- Pre-GA ROCm Stack
  - We run a beta ROCm stack with preliminary drivers
  - No publication of performance numbers with this stack, please!
  - Public stack will be installed directly after workshop

Further Reading

- AMD's CDNA2 Whitepaper 
- JSC's JURECA DC Evaluation Platform Overview

Toy Example

Translate to HIP, Run

- Use a simple example from CUDA course; run it on the MI250 nodes
- Location:
  `/p/project/training2215/toy-example`
- Please copy to your own directory before compilation!
- Please don’t block nodes unnecessarily