











Special hardware available (GPUs) & how to use it

January 15, 2024

TOP500 List November 2023

Rank	Nation	Machine	Performance	Accelerators
1.		Frontier	1194 PFLOPs/s	AMD MI250X
2.		Aurora	585 PFLOPs/s	INTEL GPU
3.		Eagle	561 PFLOPs/s	NVIDIA H100
4.		Fugaku	442 PFLOPs/s	
5.		Lumi	380 PFLOPs/s	AMD MI250X
6.		Leonardo	239 PFLOPs/s	NVIDIA A100
7.		Summit	149 PFLOPs/s	NVIDIA V100
8.		MareNostrum5	138 PFLOPs/s	NVIDIA H100
9.		EosNVIDIA DGX SPOD	121 PFLOPs/s	NVIDIA H100
10.		Sierra	95 PFLOPs/s	NVIDIA V100

GPUs @ VSC

Partition	Model	#Cores	Clock Freq (GHz)	Memory (GB)	Bandwidth (GB/s)	TDP (W)	FP32/FP64 (GFLOPs/s)
zen2_0256_a40x2	45 x 2 x A40	5376/336	1.74	48	696	300	37000/578
zen3_0512_a100x2	60 x 2 x A100	6912/432	1.40	40	1600	400	20000/10000
gpu_rtx2080ti ^[†]	19 x 1 x rtx2080ti	4352/544	1.45	11	616	255	13400/400

[†] private nodes, available only at idle times

Getting Started with GPUs, Interactive Sessions

1. VSC-5> `salloc -N 1 -p zen3_0512_a100x2 --qos zen3_0512_a100x2 --gres gpu:2`

Getting Started with GPUs, Interactive Sessions

1. VSC-5> `salloc -N 1 -p zen3_0512_a100x2 --qos zen3_0512_a100x2 --gres gpu:2`
2. VSC-5> `squeue -u $USER` (... to figure out which node has been assigned)

Getting Started with GPUs, Interactive Sessions

1. VSC-5> `salloc -N 1 -p zen3_0512_a100x2 --qos zen3_0512_a100x2 --gres gpu:2`
2. VSC-5> `squeue -u $USER` (... to figure out which node has been assigned)
3. VSC-5> `ssh n3071-003` (... or whatever else node had been assigned)

Getting Started with GPUs, Interactive Sessions

1. VSC-5> `salloc -N 1 -p zen3_0512_a100x2 --qos zen3_0512_a100x2 --gres gpu:2`
2. VSC-5> `squeue -u $USER` (... to figure out which node has been assigned)
3. VSC-5> `ssh n3071-003` (... or whatever else node had been assigned)
4. `cuda-zen sh n3071-003:~$ spack unload`
`cuda-zen sh n3071-003:~$ spack load cuda@11.8.0%gcc@9.5.0`
`cuda-zen sh n3071-003:~$ cd ~/examples/09_special_hardware/matrixMul`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$ nvcc -arch=native ./matrixMul.cu`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$./a.out`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$ cd`
`cuda-zen sh n3071-003:~$ cd ~/examples/09_special_hardware/matrixMulCUBLAS`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ which nvcc`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ export`
`LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/gpfs/opt/sw/cuda-zen/spack-0.19.0/opt/spack/linux-almalinux8-zen/gcc-9.5.0/cuda-11.8.0-ananl33ltrpp33xetcoltkbbbfuxoeez/lib64`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ nvcc -arch=native`
`./matrixMulCUBLAS.cu -lcublas`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$./a.out`

Getting Started with GPUs, Interactive Sessions

1. VSC-5> `salloc -N 1 -p zen3_0512_a100x2 --qos zen3_0512_a100x2 --gres gpu:2`
2. VSC-5> `squeue -u $USER` (... to figure out which node has been assigned)
3. VSC-5> `ssh n3071-003` (... or whatever else node had been assigned)
4. `cuda-zen sh n3071-003:~$ spack unload`
`cuda-zen sh n3071-003:~$ spack load cuda@11.8.0%gcc@9.5.0`
`cuda-zen sh n3071-003:~$ cd ~/examples/09_special_hardware/matrixMul`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$ nvcc -arch=native ./matrixMul.cu`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$./a.out`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$ cd`
`cuda-zen sh n3071-003:~$ cd ~/examples/09_special_hardware/matrixMulCUBLAS`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ which nvcc`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ export`
`LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/gpfs/opt/sw/cuda-zen/spack-0.19.0/opt/spack/linux-almalinux8-zen/gcc-9.5.0/cuda-11.8.0-ananl33ltrpp33xetcoltkbbbfuxoez/lib64`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ nvcc -arch=native`
`./matrixMulCUBLAS.cu -lcublas`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$./a.out`
5. `cuda-zen sh n3071-003:~$ nvidia-smi`

Getting Started with GPUs, Interactive Sessions

1. VSC-5> `salloc -N 1 -p zen3_0512_a100x2 --qos zen3_0512_a100x2 --gres gpu:2`
2. VSC-5> `squeue -u $USER` (... to figure out which node has been assigned)
3. VSC-5> `ssh n3071-003` (... or whatever else node had been assigned)
4. `cuda-zen sh n3071-003:~$ spack unload`
`cuda-zen sh n3071-003:~$ spack load cuda@11.8.0%gcc@9.5.0`
`cuda-zen sh n3071-003:~$ cd ~/examples/09_special_hardware/matrixMul`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$ nvcc -arch=native ./matrixMul.cu`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$./a.out`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMul$ cd`
`cuda-zen sh n3071-003:~$ cd ~/examples/09_special_hardware/matrixMulCUBLAS`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ which nvcc`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ export`
`LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/gpfs/opt/sw/cuda-zen/spack-0.19.0/opt/spack/linux-almalinux8-zen/gcc-9.5.0/cuda-11.8.0-ananl33ltrpp33xetcoltkbbbfuxoeez/lib64`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$ nvcc -arch=native`
`./matrixMulCUBLAS.cu -lcublas`
`cuda-zen sh n3071-003:~/examples/special_hardware/matrixMulCUBLAS$./a.out`
5. `cuda-zen sh n3071-003:~$ nvidia-smi`
6. `cuda-zen sh n3071-003:~$ deviceQuery` (... from the SDK now to be self-compiled)

SLURM Submission to GPU Nodes

```
#!/bin/bash
#
# usage: sbatch ./gpu_test.scrpt
#
#SBATCH -J a100
#SBATCH -N 1
#SBATCH -p zen3_0512_a100x2
#SBATCH --qos zen3_0512_a100x2
#SBATCH --gres gpu:2

spack unload
spack load cuda@11.8.0%gcc@9.5.0

nvidia-smi
```

Exercise/Example/Problem

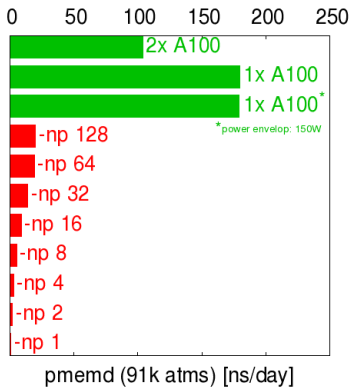
Using interactive mode or batch submission, figure out whether we have ECC enabled on GPUs of type A100 ?

Hint: `nvidia-smi -i 0 -q` will query all settings on the first device, ie number 0

GPUs are Power-Efficient

Example: AMBER-22

Performance:



Power Efficiency:

