GPU programming made easy with OpenMP

Aditya Nitsure (anitsure@in.ibm.com)

Pidad D’Souza (pidsouza@in.ibm.com)

OpenMP®

Enabling HPC since 1997

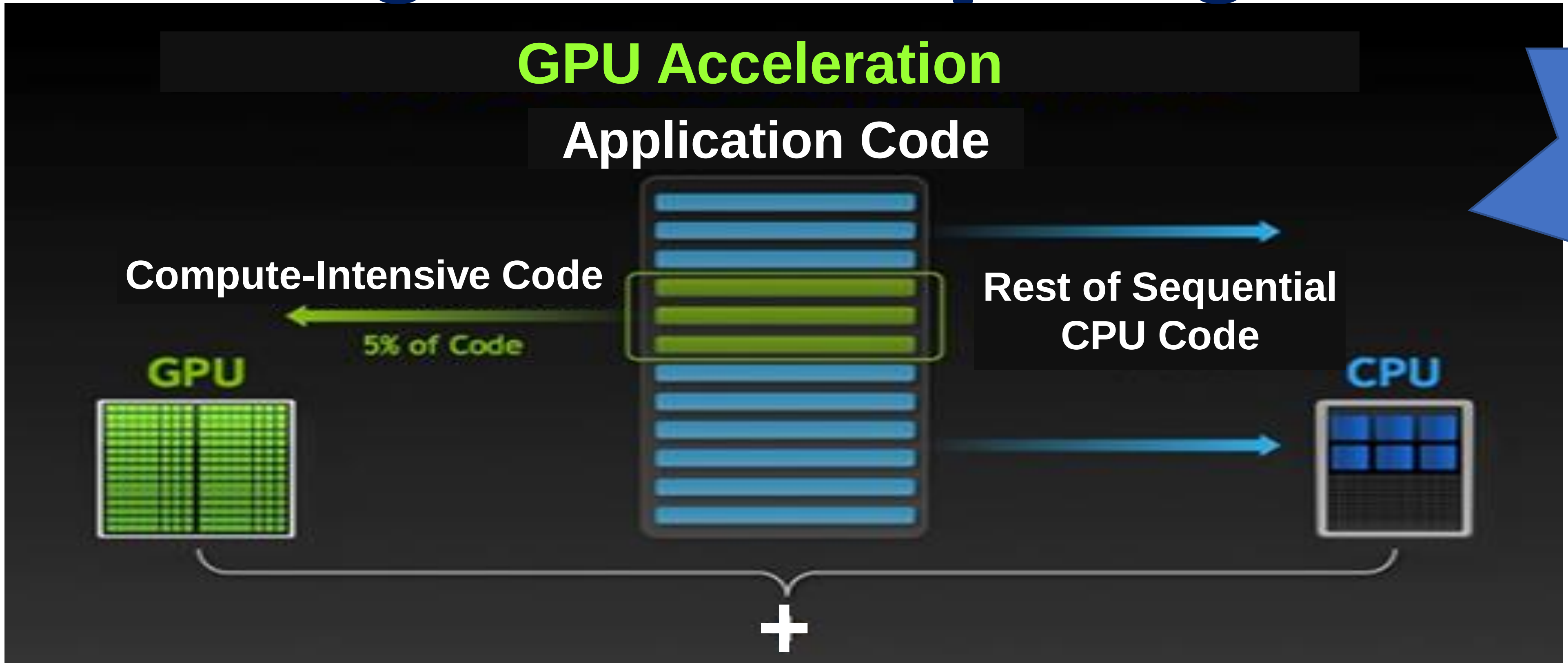
Tutorial Overview

- ➢Heterogeneous system overview
- ➢Introduction to parallel programming
- ➢OpenMP programming on CPU and GPU
- ➢Profiling and monitoring
- ➢HPC application performance

- ➢Hands-on

Heterogenous Systems

Heterogenous Computing

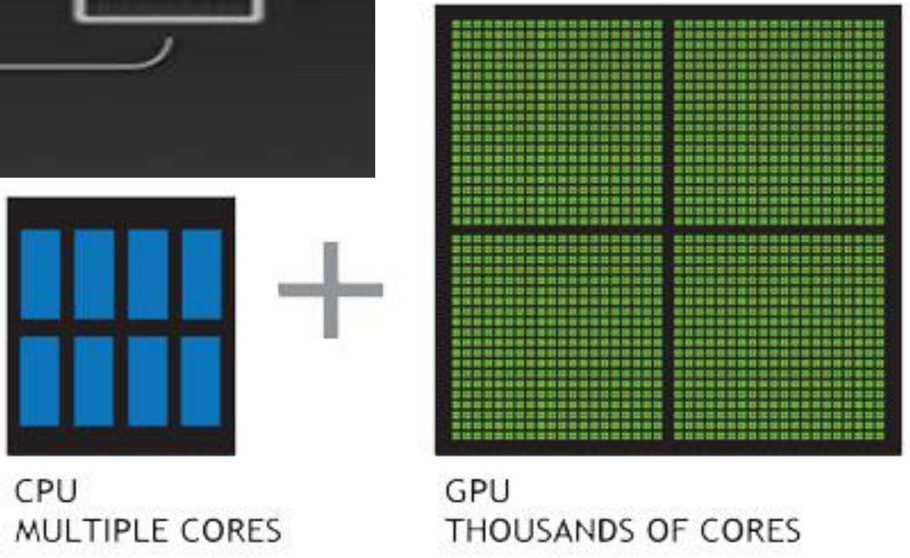


CPU

- Large and broad instruction set to perform complex operations

GPU

- High throughput – Massive parallelization through large number of cores
- Specialized for SIMD/SIMT

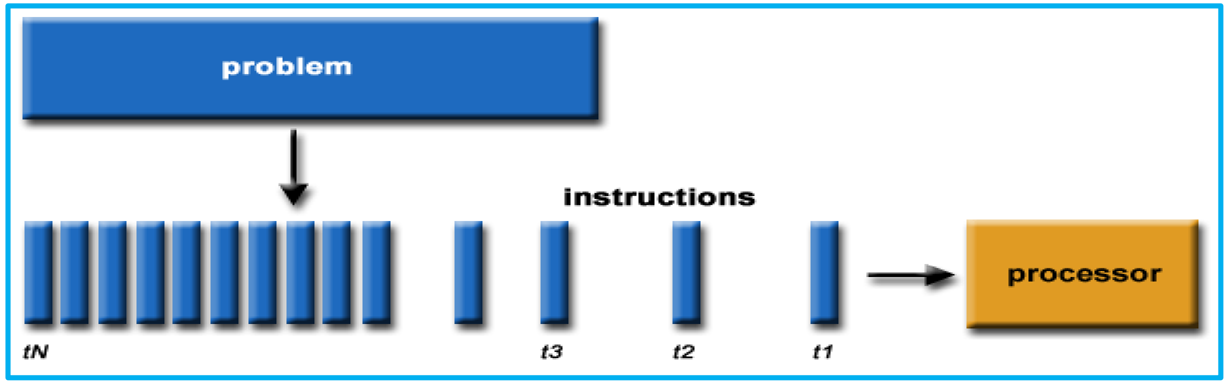


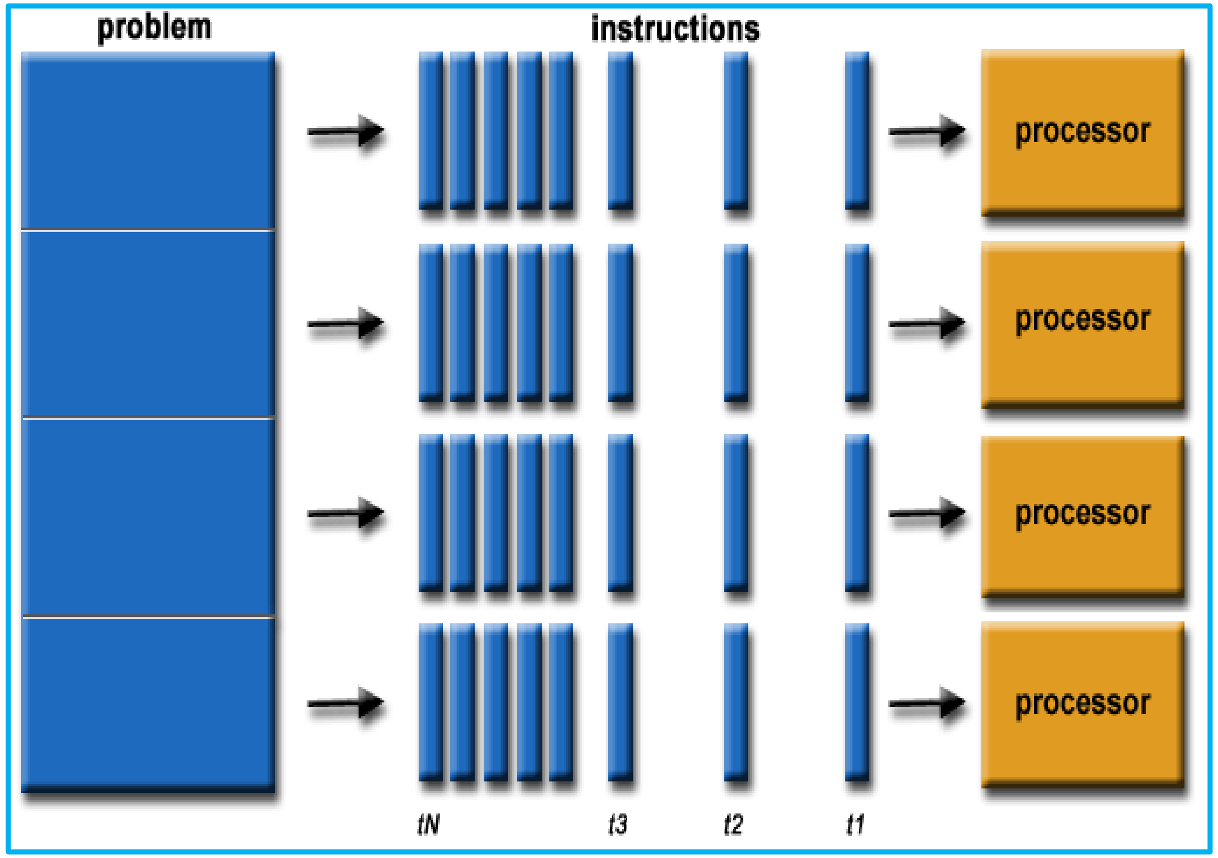
Heterogenous System configurations

Todo

Introduction to Parallel computing

What is Parallel Computing?





https://computing.llnl.gov/tutorials/parallel_comp/

Sequential Computing

- Sequential execution of series of instructions
- Only one computing resource

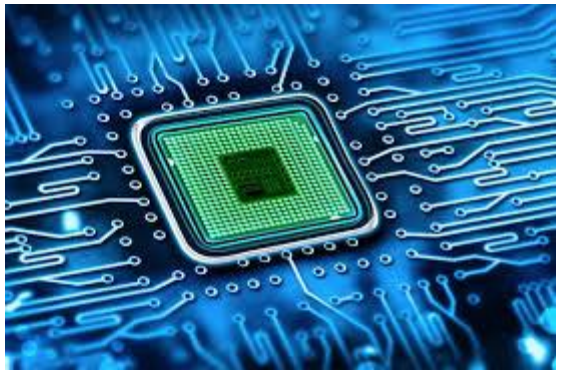
Parallel Computing

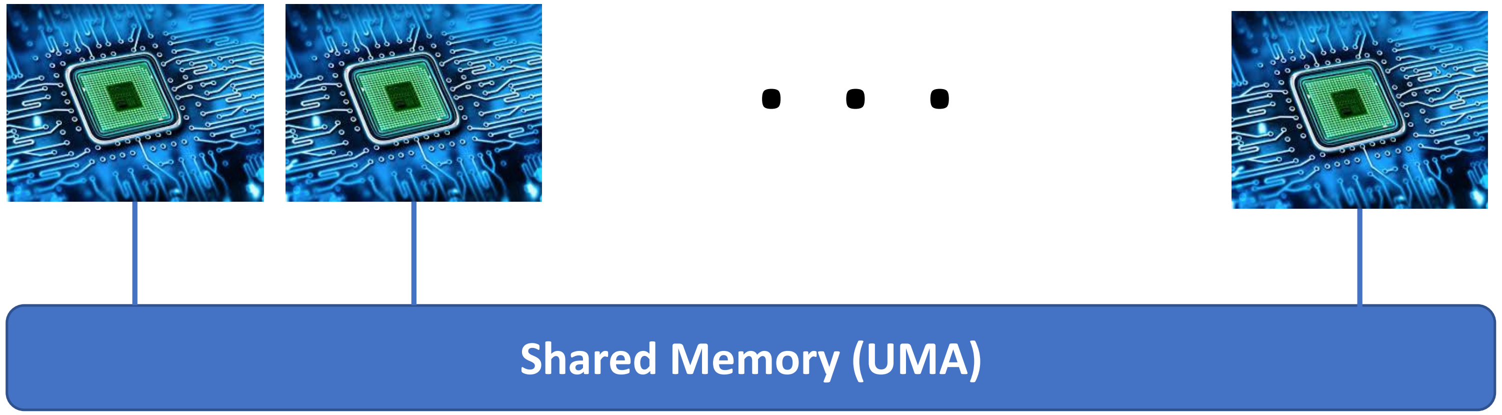
Simultaneous use of multiple compute resources to solve a computational problem

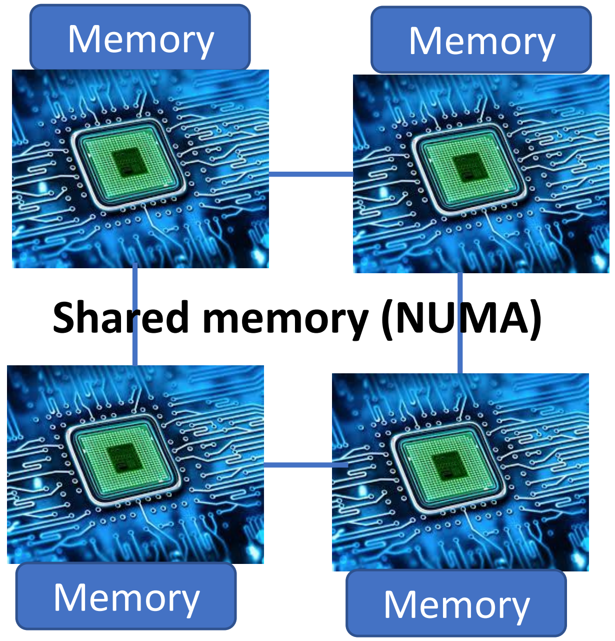
- Divided problem into discrete sub-problems
- Execute sub-problems in in simultaneously on different compute resource
- More than one compute resource

More slides to follow

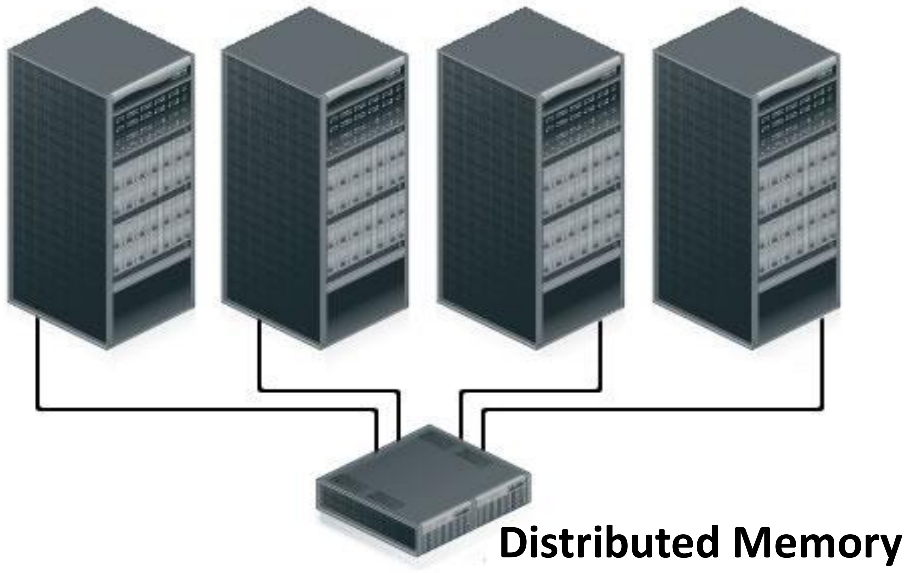
Memory Architecture







Cluster of nodes



Proprietary and Open Source Compiler Offerings supporting Acceleration Enabled Programing Models

Key Features:

- Gives direct access to the GPU instruction set
- Supports C, C++ and Fortran
- Generally achieves best leverage of GPUs for best application performance

OpenCL

MPI – Message passing

OpenMP

Key Features:

- OpenMP 4.0 introduces offloading and support for heterogeneous CPU/GPU
- Leverage existing OpenMP high level directives support

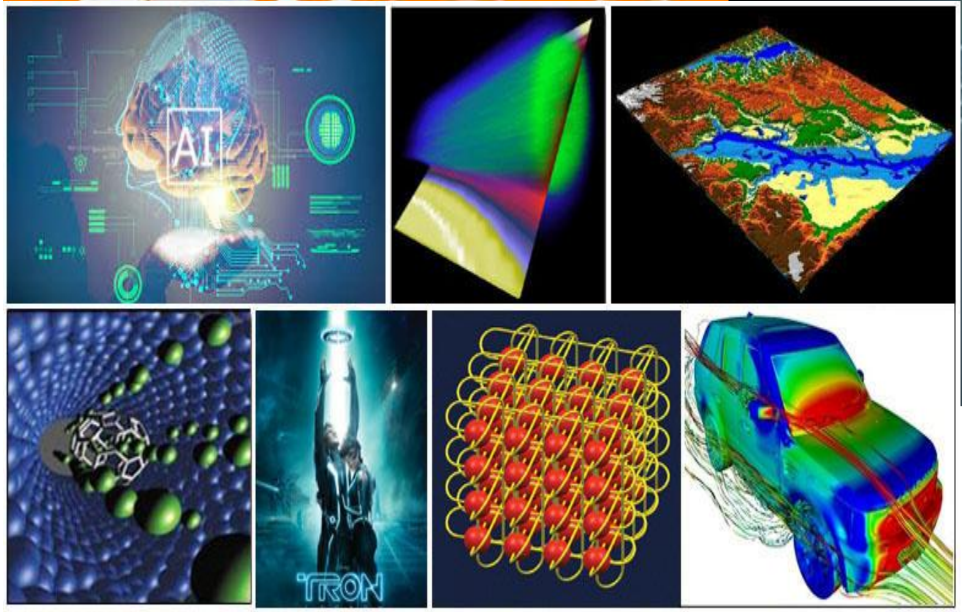
OpenACC
Directives for Accelerators

Key Features:

- Designed to simplify Programing of heterogeneous CPU/GPU systems
- Directive based parallelization for accelerator device

Applications of Parallel Computing

Science &
Engineering
Artificial Intelligence



Analytics
Realtime simulations
Modelling

OpenMP

Monitoring and Profiling tools

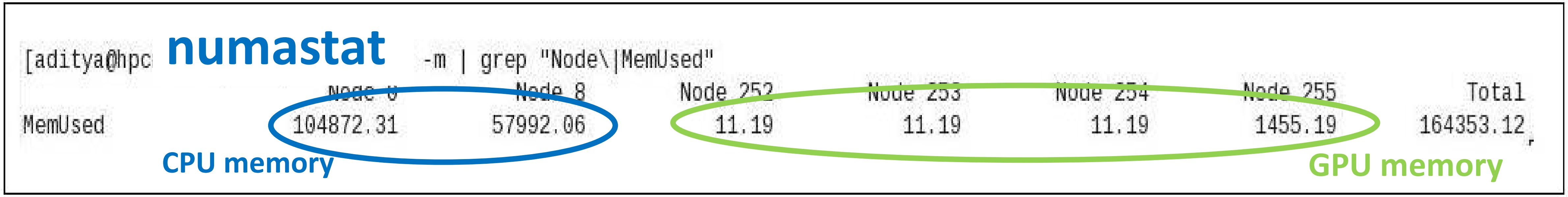
Monitoring and Profiling tools

Monitoring

- mpstat, vmstat – CPU and memory utilization
- numastat – numa memory statistics
- top/htop – real-time view of system usage

Profiling

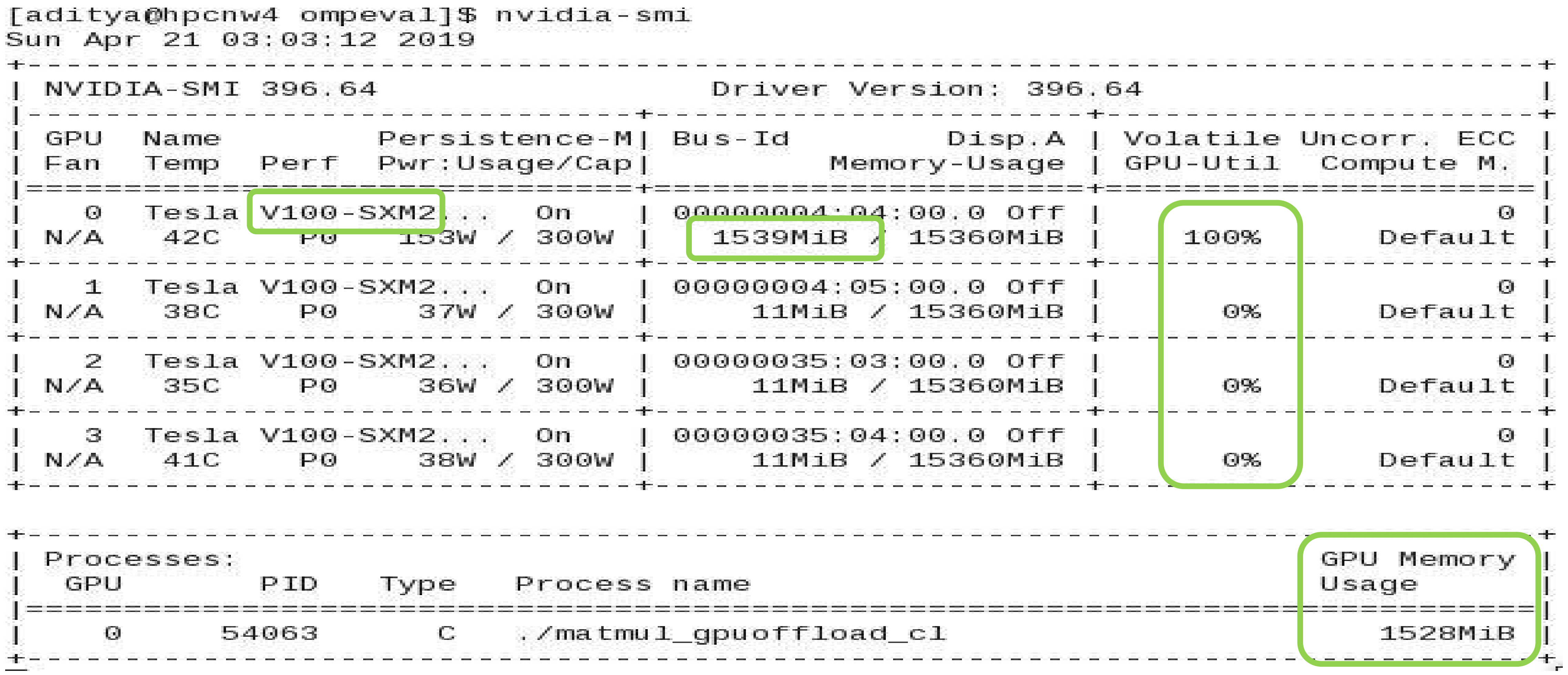
- Perf record/report – CPU profiling
- nvprof – GPU profiling



```
[aditya@hpc numastat -m | grep "Node\|MemUsed"
                       Node 0       Node 8     Node 252     Node 253     Node 254     Node 255        Total
MemUsed             104872.31     57992.06        11.19        11.19        11.19      1455.19    164353.12
```

Monitoring and Profiling tools

nvidia-smi



```
[aditya@hpcnw4 ompeval]$ nvidia-smi
Sun Apr 21 03:03:12 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 396.64                 Driver Version: 396.64                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000004:04:00.0 Off |                    0 |
| N/A   42C    P0   153W / 300W |   1539MiB / 15360MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000004:05:00.0 Off |                    0 |
| N/A   38C    P0    37W / 300W |     11MiB / 15360MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000035:03:00.0 Off |                    0 |
| N/A   35C    P0    36W / 300W |     11MiB / 15360MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000035:04:00.0 Off |                    0 |
| N/A   41C    P0    38W / 300W |     11MiB / 15360MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0     54063      C   ./matmul_gpuoffload_cl                       1528MiB |
+-----------------------------------------------------------------------------+
```

Also check "nvidia-smi –query-gpu" more monitoring options

Monitoring and Profiling tools

nvprof

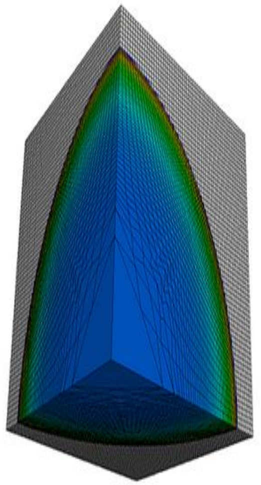
- The *nvprof* is command-line profiling tool which enables you to collect and view profiling data
- Todo

NVVP (NVIDIA Visual Profiler)

- The Visual Profiler displays a timeline of your application's activity on both the CPU and GPU so that one can identify opportunities for performance improvement.
- todo

HPC Application performance with OpenMP GPU offloading

LULESH : OpenMP 4.5 GPU Offload



LULESH

A shock hydrodynamics mini-app for computer simulations leveraging high performance computing of a wide variety of science and engineering problems that describes the motion of materials relative to each other when subject to forces.

Chart to follow

References

- GPU programming made easy with OpenMP on IBM POWER (https://developer.ibm.com/articles/gpu-programming-with-openmp)
- Offloading computations to the NVIDIA GPUs (https://www.ibm.com/support/knowledgecenter/en/SSXVZZ_16.1.0/com.ibm.xlcpp161.lelinux.doc/proguide/offloading.html)
- Parallel Computing(https://computing.llnl.gov/tutorials/parallel_comp)

Thank You