New Servers and Using GPU Nodes

"New Servers and Using GPU Nodes on Psychology Research Cluster"

Weijun Gao @ UTSC Psychology, February 2021

https://psycomp.utsc.utoronto.ca

Contents

- New Servers
- Slurm Partitions
- Using “gpu” and “gpudebug” Partitions
- Tensor Cores and Mixed Precision
- Using GPU Devices
- Demo

New Servers

https://psycomp.utsc.utoronto.ca/support/index.php/resources/psychology-cluster-hardware/

- 1 x CPU Server
  - Dell R740
  - 2 x Intel Xeon Platinum 8268 2.9G, 24C/48T
  - 384G RAM
  - 18TB usable storage space for computing (HDDs)
- 1 x GPU Server for Debugging
  - Dell R740
  - 2 x Intel Xeon Silver 4208 2.1G, 8C/16T
  - 32G RAM
  - 2 x NVIDIA T4 16 GB, 2560 Cuda cores, 320 Tensor cores
- 2 x GPU Servers for Computing (with good CPU cores too)
  - Dell R740
  - 2 x Intel Xeon Gold 6238R 2.2G, 28C/56T
  - 384G RAM
  - 1 x NVIDIA Quadro RTX 8000 48 GB, 4608 Cuda cores, 576 Tensor cores, 72 RT cores
  - 18TB usable storage space for computing (HDDs)
- 1 x Storage Server
  - Silicon Mechanics Storform R513.v7
  - 80TB usable backup and archive space (HDDs)

GPU servers are also good for CPU computing.

Home folders will be moved to the new servers.

Slurm Partitions

https://psycomp.utsc.utoronto.ca/support/index.php/resources/cluster-status/

- "cpu" – wall time – 30 days
  - 7 x Dell R430
    - 32 CPU cores, 64G RAM
  - 1 x Dell R740
    - 96 CPU cores, 384G RAM
- "interactive" – wall time – 1 day
  - 1 x Dell R430
    - 32 CPU cores, 64G RAM
- "gpu" – wall time – 2 days (will be adjusted according to needs)
  - 2 x Dell R740
    - 112 CPU cores, 384G RAM
    - 1 x GPU with 48GB memory, 4608 Cuda cores, 576 Tensor cores
- "gpudebug" – wall time – 1 day
  - Dell R740
    - 32 CPU cores, 32G RAM
    - 2 x GPU with 16GB memory, 2560 Cuda cores

Admin users can increase the wall-time of a Slurm job.

Using “gpu” and “gpudebug” Partitions

- Start a “gpudebug” job
  - *gpudebug [your-email-address]*
  - *srun -p gpudebug --gpus=1 --mail-type=ALL --mail-user=EMAIL --pty bash -i*
  - *srun -p gpudebug --gpus=1 --mail-type=ALL --mail-user=EMAIL --x11 --pty bash -i*
- Start a “gpu” job
  - *srun -p gpu -c 20 --gpus=1 --mail-type=ALL --mail-user=EMAIL --x11 --pty bash -i*
  - sbatch sbatchScript.sh

    #SBATCH --gpus=1

A user can use maximum 1 GPU at the same time.

May run a script to monitor the “gpu” and “gpudebug” partitions.

https://psycomp.utsc.utoronto.ca/support/index.php/resources/gpu-usage/

Using “gpu” and “gpudebug” Partitions

a top like utility for GPU

- nvidia-smi
  - nvidia-smi

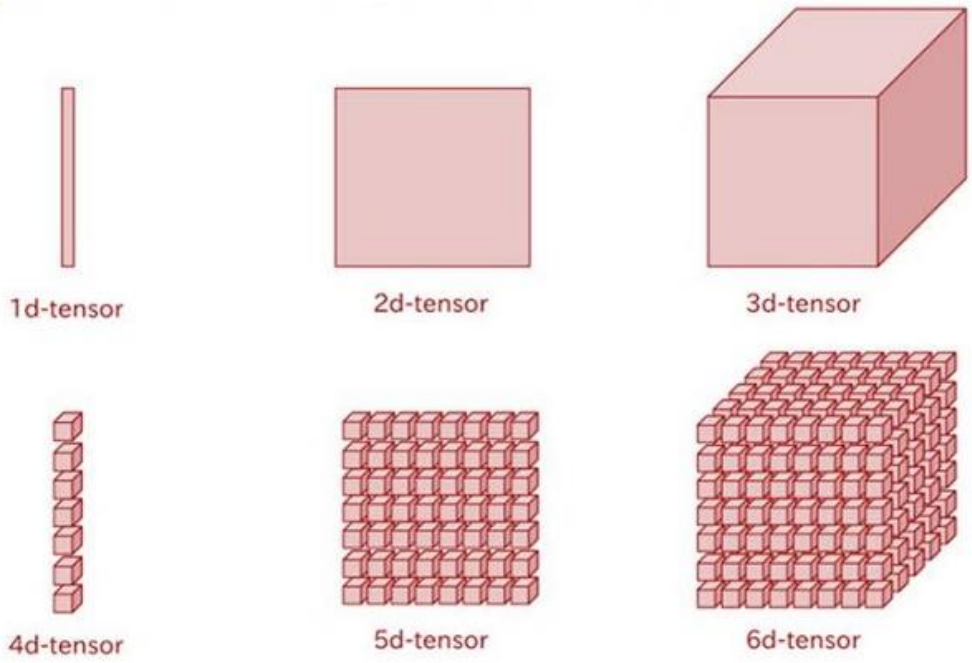
```
Sat Feb  6 18:24:04 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:3B:00.0 Off |                    0 |
| N/A   26C    P8     8W /  70W |      4MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:D8:00.0 Off |                    0 |
| N/A   25C    P8     9W /  70W |      4MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      1409      G   /usr/lib/xorg/Xorg                  4MiB |
|    1   N/A  N/A      1409      G   /usr/lib/xorg/Xorg                  4MiB |
+-----------------------------------------------------------------------------+
```

identify processes using a GPU

- fuser -v
  - fsuer -v /dev/nvidia0

```
root@neurocompl2:~# fuser -v /dev/nvidia0
                     USER        PID ACCESS COMMAND
/dev/nvidia0:        root       1409 F...m Xorg
                     wgao     105253 F...m python
root@neurocompl2:~# fuser -v /dev/nvidia1
                     USER        PID ACCESS COMMAND
/dev/nvidia1:        root       1409 F...m Xorg
```

Tensor Cores and Mixed Precision



https://www.tensorflow.org/guide/tensor

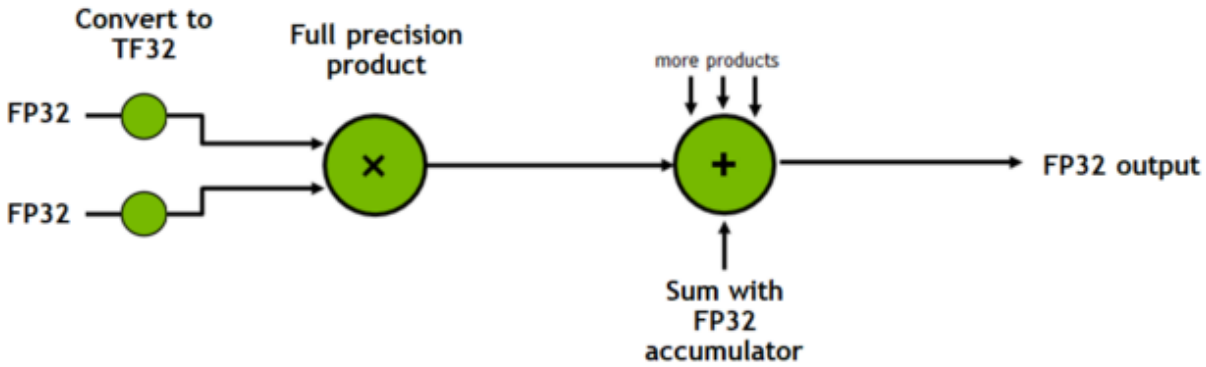
CUDA cores are for general computing using CUDA SDKs, Tensor cores are optimized for AI training and inferencing, and RT cores are optimized for ray tracing calculations.

Tensor Cores and Mixed Precision

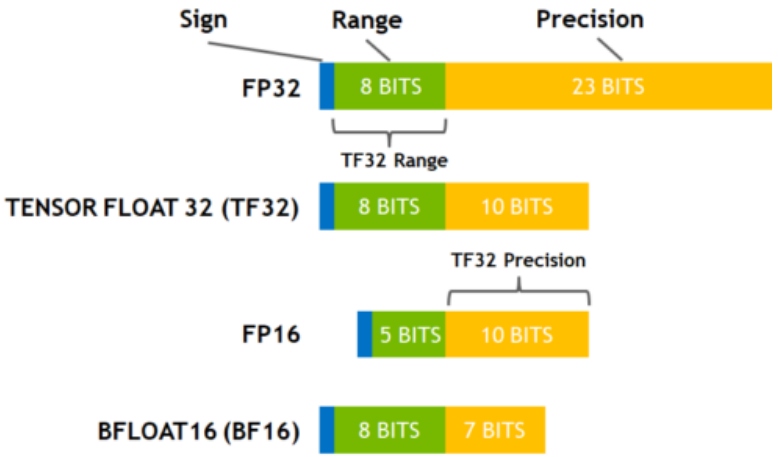
- "Tensor Cores enable mixed-precision computing, dynamically adapting calculations to accelerate throughput while preserving accuracy. The latest generation expands these speedups to a full range of workloads. From 10X speedups in AI training with Tensor Float 32 (TF32), a revolutionary new precision, to 2.5X boosts for high-performance computing with floating point 64 (FP64), NVIDIA Tensor Cores deliver new capabilities to all workloads."

https://www.nvidia.com/en-us/data-center/tensor-cores/

Tensor Cores and Mixed Precision



Dot product computation, which forms the building block for both matrix multiplications and convolutions, rounds FP32 inputs to TF32, computes the products without loss of precision, then accumulates those products into an FP32 output.



- https://developer.nvidia.com/blog/accelerating-ai-training-with-tf32-tensor-cores/
- https://blogs.nvidia.com/blog/2020/05/14/tensorfloat-32-precision-format/
- https://en.wikipedia.org/wiki/Bfloat16_floating-point_format

Using GPU Devices

- Tensorflow

```
import tensorflow as tf
# Place tensors on the GPU
with tf.device('/GPU:0'):
  a = tf.constant([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])
  b = tf.constant([[1.0, 2.0], [3.0, 4.0], [5.0, 6.0]])

# Run on the GPU
c = tf.matmul(a, b)
print(c)
```

Using GPU Devices

- PyTorch

```
import torch
if torch.cuda.is_available():
  dev = "cuda:0"
else:
  dev = "cpu"
device = torch.device(dev)
a = torch.zeros(4,3)
a = a.to(device)
```

Using GPU Devices

- Python with Numba and Cudatoolkit

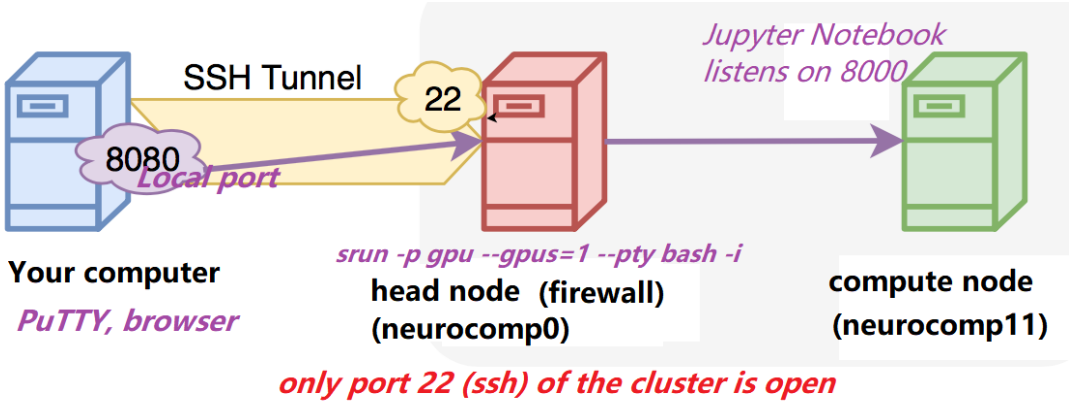
```
from numba cuda
import numpy as np

# normal function to run on cpu
def func(a,n):
    for i in range(n):
        a[i]+= 1

# function optimized to run on gpu
@cuda.jit
def func2(a,n):
    for i in range(n):
        a[i]+= 1
…
```

Using GPU Devices

- Matlab

```
…
for ii=1:numel(sizes)
    % First do it on the host
    A = rand( N(ii), N(ii) );
    B = rand( N(ii), N(ii) );
    mmTimesHost(ii) = timeit(@() A*B);
    % Now on the GPU
    A = gpuArray(A);
    B = gpuArray(B);
    mmTimesGPU(ii) = gputimeit(@() A*B);
…
```

Demo



- *module* command
- Jupyter Notebook

  PuTTY + SSH Tunnel + Slurm + Jupyter Notebook

  https://psycomp.utsc.utoronto.ca/support/index.php/2019/06/14/connecting-to-jupyter-notebook-running-on-psy-cluster-using-your-laptop-desktop-browser/

  Tensorflow, Keras, mixed-precision, PyTorch

  /pkgsGPU/scripts

- Matlab GPU and CPU benchmark

  https://www.mathworks.com/matlabcentral/fileexchange/34080-gpubench

  - GPU – single precision
  - CPU – double precision

Demo

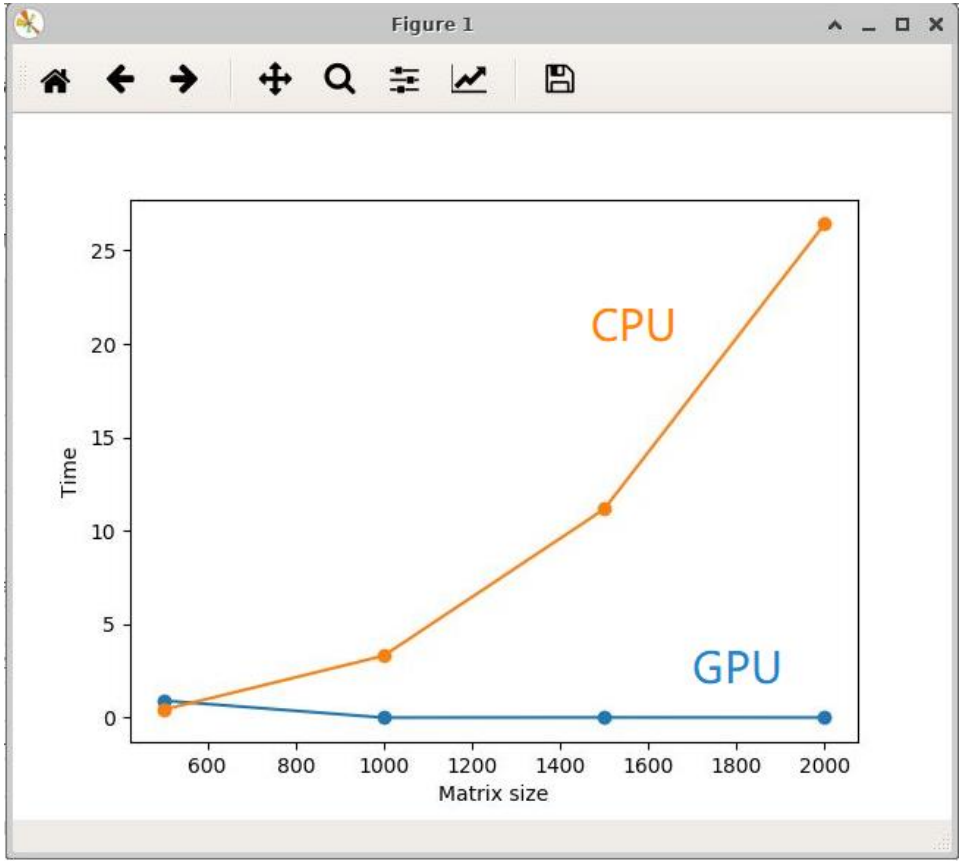
- Matlab GPU and CPU benchmark

  https://www.mathworks.com/matlabcentral/fileexchange/34080-gpubench

	Double precision results (in GFLOPS)			Single precision results (in GFLOPS)		
	MTimes	**Backslash**	**FFT**	**MTimes**	**Backslash**	**FFT**
Tesla V100-PCIE-32GB	6884.95	563.73	728.71	13727.99	1210.42	1365.11
TITAN V	6779.73	674.40	534.65	13515.42	1336.39	985.36
Tesla P100-PCIE-12GB	4510.03	929.00	357.65	8435.34	1647.83	687.13
Tesla K40c	1189.54	677.12	135.88	3187.76	1334.17	294.86
Tesla K20c	1004.06	641.42	106.09	2657.01	1230.28	235.20
Your GPU (Quadro RTX 8000)	**460.57**	**373.53**	**208.01**	**13268.24**	**2703.44**	**918.77**
TITAN Xp	421.00	369.32	209.45	10823.05	1272.06	797.17
GeForce RTX 2080 SUPER	373.37	345.32	164.30	10813.12	1330.64	746.20
Your CPU (112 CPU cores)	**1921.02**	**282.34**	**49.79**	**2654.49**	**612.79**	**79.09**

Demo

- /pkgsGPU/scripts/anaconda2.tensorflow.GPUvsCPU.py



Thank you!

- Questions?

Example scripts: /pkgsGPU/scripts

https://psycomp.utsc.utoronto.ca/support/

https://psycomp.utsc.utoronto.ca/support/index.php/contact/research-cluster-use-policy/

https://psycomp.utsc.utoronto.ca/support/index.php/resources/cluster-status/