5

1

Compare the performance with sse2 and avx2 extensions in regards to “vectMul.c” at ~/csds312/simd (i.e. sse2 vs avx2)

```
[tln32@class-login2 ~]$ cd ~/csds312/simd
[tln32@class-login2 simd]$ srun -A csds312 -p markov_cpu --pty /bin/bash
[tln32@classct005 simd]$ module load iccifort
[tln32@classct005 simd]$ icc -o novect -O0 vectMul.c -lm
Error: A license for Comp-CL is not available now (-15,570,115).
A connection to the license server could not be made.  You should
make sure that your license daemon process is running: both an
lmgrd process and an INTEL process should be running
if your license limits you to a specified number of licenses in use
at a time.  Also, check to see if the wrong port@host or the wrong
license file is being used, or if the port or hostname in the license
file has changed.
License file(s) used were (in this order):
**  1.  27009@smaster.case.edu
**  2.
/usr/local/easybuild_allnodes/software/iccifort/2020.4.304/compilers_and
_libraries_2020.4.304/linux/bin/intel64/../../Licenses
**  3.  /home/tln32/Licenses
**  4.  /home/tln32/intel/licenses
**  5.  /opt/intel/licenses
**  6.  /Users/Shared/Library/Application Support/Intel/Licenses
**  7.
/usr/local/easybuild_allnodes/software/iccifort/2020.4.304/compilers_and
_libraries_2020.4.304/linux/bin/intel64/*.lic
Please refer http://software.intel.com/sites/support/ for more
information..
icc: error #10052: could not checkout FLEXlm license
[tln32@classct005 simd]$ # I failed to recreate the recompilation of the
source given, thus I will just use the given compiled files
[tln32@classct005 simd]$ time ./novect
Multiplication vector:
74433190797631.000000
real    0m0.233s
user    0m0.180s
sys 0m0.040s
[tln32@classct005 simd]$ time ./autovect
Multiplication vector:
74433190797631.000000
real    0m0.072s
```

```
user    0m0.022s
sys 0m0.045s
[tln32@classct005 simd]$
```

This gives us the result of:

No Vectorization	Auto Vectorization
0.233	0.072

2

Report GPU utilization - https://sites.google.com/a/case.edu/hpcc/guides-and-training/helpful-references/hpc-environment/gpu-benchmark--hpc. (use 2 Gpus to avoid conflict at /tmp/GPUBenchReport though only one GPU is utilized)

```
[tln32@class-login2 ~]$ cd ~/csds312
[tln32@class-login2 ~]$ cp -r /usr/local/doc/MATLAB/GPUBench-v2p1 .
[tln32@class-login2]~/csds312% cd GPUBench-v2p1/
[tln32@class-login2]~/csds312/GPUBench-v2p1% srun -A csds312 -p
markov_gpu --gres=gpu:2 --exclusive --mem=90g --pty /bin/bash
[I have no name!@classt06 GPUBench-v2p1]$ module load matlab
[I have no name!@classt06 GPUBench-v2p1]$ matlab
> gpuBench
```

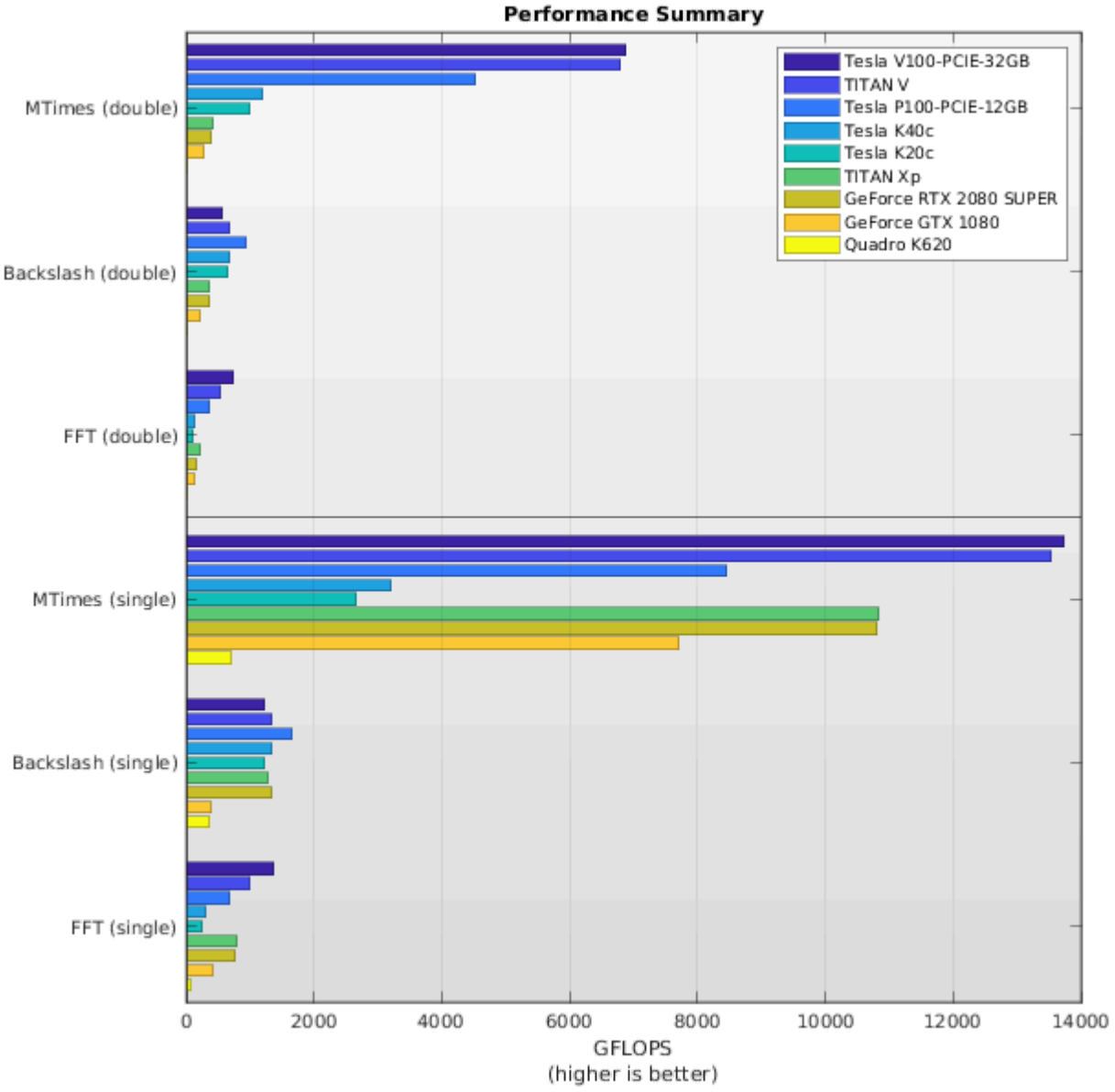
GPU Comparison Report

Summary of results

The table and chart below show the peak performance of various GPUs using the same version of MATLAB. Your results (if any) are highlighted in bold in the table and on the chart. All other results are from pre-stored data. The peak performance shown is usually achieved when dealing with extremely large arrays. Typical performance in day-to-day use will usually be much lower.

Results captured using the CPUs on the host PC (i.e., without using a GPU) are included for comparison.

Results are shown for calculations in both single and double precision. For all results, higher is better.


Performance Summary
Tesla V100-PCIE-32GB
TITAN V
Tesla P100-PCIE-12GB
Tesla K40c
Tesla K20c
TITAN Xp
GeForce RTX 2080 SUPER
GeForce GTX 1080
Quadro K620
MTimes (double)
Backslash (double)
FFT (double)
MTimes (single)
Backslash (single)
FFT (single)
0
2000
4000
6000
8000
10000
12000
14000
GFLOPS
(higher is better)