EXTENDING MAGMA PORTABILITY

Final Presentation

Anna Fortenberry, UNT
UTK RECSEM REU 2022

Mentors: Dr. Stan Tomov, UTK and Dr. Kwai Wong, UTK

CONTENTS

1

PROBLEM OVERVIEW

SIGNIFICANCE OF SUPERCOMPUTING

- Supercomputers provide the computational power necessary to resolve problems in a vast number of important domains

data science

quantum information science

applied mathematics

high performance computing

cybersecurity

artificial intelligence research

[1], [2], [3]

EVOLUTION OF SUPERCOMPUTER SYSTEM DESIGN

- NVIDIA opened a new door for supercomputing (SC) capabilities with the invention of the GPU in 1999
- NVIDIA Tesla K20X GPU powered the first successful hybrid SC system in 2012
- SC Systems are continually increasing in diversity

[4], [5]

SC SYSTEM VENDORS

TOP500 The List			
JUNE 2022	CPU/ Accelerator	JUNE 2019	CPU/ Accelerator
Frontier	AMD, AMD	Summit	IBM, NVIDIA
S.C. Fugaku	Fugaku	Sierra	IBM, NVIDIA
LUMI	AMD, AMD	Sunway TaihuLight	Sunway
Summit	IBM, NVIDIA	Tianhe-2A	Intel
Sierra	IBM, NVIDIA	Frontera	Intel
Sunway TaihuLight	Sunway	Piz Daint	Intel, NVIDIA
Perlmutter	AMD, NVIDIA	Trinity	Intel
Selene	AMD, NVIDIA	ABCI	Intel, NVIDIA
Tianhe-2A	Intel, NUDT	SuperMUC-NG	Intel
Adastra	AMD, AMD	Lassen	IBM, NVIDIA

[6], [7]

FIRST INTEL GPU POWERED SUPERCOMPUTER

Anticipated for release in late 2022, Intel hopes to enter the supercomputer GPU vendor domain by powering the Aurora supercomputer at Argonne National Laboratory

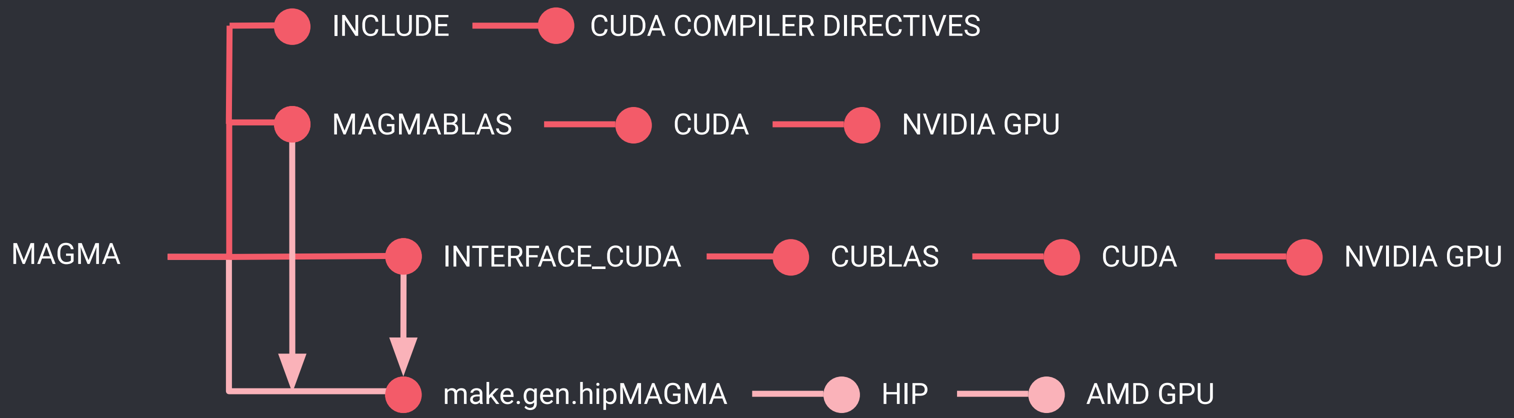
[8]

INTEL ONEAPI

- Intel recently released a new programming model called **oneAPI**
- Applications that take advantage of oneAPI gain portability to all supported hardware platforms
  - CPUs (Scalar Architecture)
  - GPUs (Vector Architecture)
  - FPGAs (Spatial Architecture)
  - Other Accelerators (Matrix Architecture)

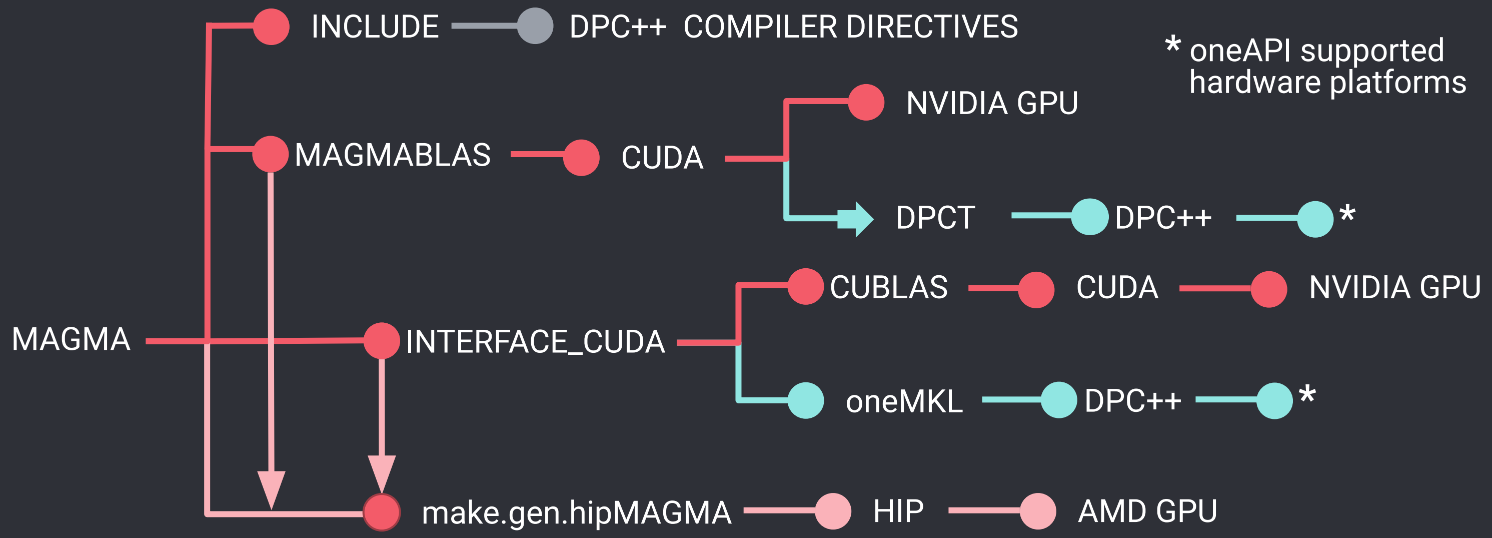
[9], [10]

MAGMA STRUCTURE



- Designed originally to run on NVIDIA GPUs
- Extended to support AMD GPUs

EXTENDED MAGMA STRUCTURE



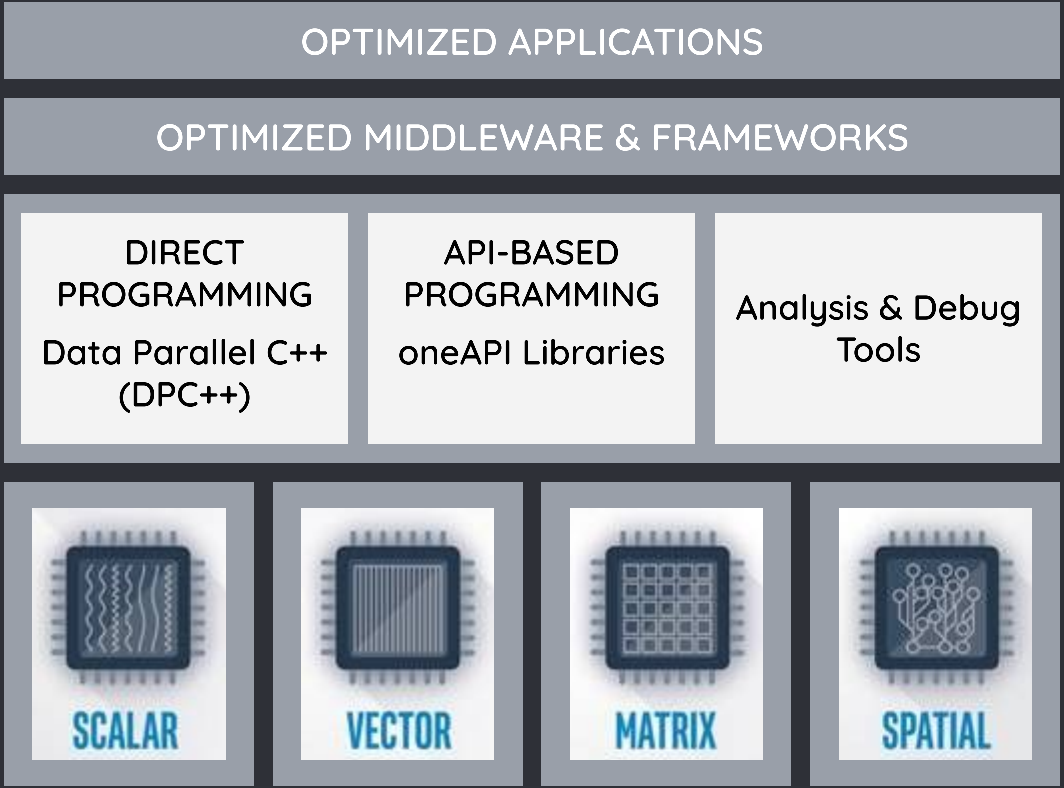
- oneAPI includes tools for adopting the model
  - Data Parallel C++ (DPC++) Translation Tool (DPCT)
  - oneAPI Math Kernel Library (oneMKL)

RESEARCH QUESTIONS

- How well does the DPCT tool translate CUDA code to DPC++ code?
- What are the common translation errors?
- Can this tool be used to translate MAGMA?
- Is DPC++ portable to Nvidia and AMD GPUs, and multicore CPUs?
- What is the performance of DPC++ on each of these accelerators comparative to CUDA?

2 SOFTWARE AND HARDWARE

STRUCTURE OF ONEAPI



- DPC++ is a oneAPI implementation of the Khronos standard **SYCL**
- SYCL is an accelerator language that allows code reuse across hardware targets
- SYCL adds data parallelism and heterogeneous programming to standard ISO C++

[10], [11]

SOFTWARE OVERVIEW

DPC++ Compatibility Tool (DPCT)

oneAPI tool to assist with migrating CUDA code to DPC++ code; translates with high accuracy

oneAPI Math Kernel (oneMKL)

set of math routines for use in high performance computing on a variety of computational devices

Compute Unified Device Architecture (CUDA)

NVIDIA parallel computing platform for harnessing power of GPUs

DPC++-LLVM (CLang-LLVM)

LLVM-based compiler project that supports SYCL language

DPC++ LLVM NVIDIA*

CLANG-LLVM build on Linux with CUDA NVIDIA support; allows DPC++ to port to NVIDIA GPUs

Intel DevCloud

Remote development environments that grant access to Intel hardware for testing oneAPI projects*

[12], [13], [14], [15], [20]

CENTRAL PROCESSING UNITS

AMD EPYC 7742 PROCESSOR

Cores:	64
Base Clock:	2.25 Ghz
# of Threads:	128
Cache:	256 MB

INTEL® XEON® PROCESSOR E5-2698 V4

Cores:	20
Base Clock:	2.20 Ghz
# of Threads:	40
Cache:	50 MB

[16], [17]

GRAPHICS PROCESSING UNITS

NVIDIA GEFORCE RTX 3060 (Discrete)

GPU Cores:	3584
Base Clock:	1320 MHz
Memory Size:	12 GB

INTEL UHD GRAPHICS P630 [0x3e96] (Integrated)

GPU Cores:	192
Base Clock:	350 MHz
Memory Size:	Shared System

[18], [19]

3 METHODOLOGY

METHODOLOGY

1. Configure system to run DPC++ code and document the process
2. Translate different structures of CUDA files to DPC++ with DPCT for correctness
3. Document translation process and common errors
4. Configure system to run DPC++ code on Nvidia GPU
5. Repeat steps 1-4 on Innovative Computing Lab (ICL) Account
6. Set up directory with MAGMA CUDA sgemm and dependencies
7. Migrate CUDA code in directory to DPC++ with DPCT and correct errors
8. Test and compare performance of sgemm on available hardware

4 CUDA TO DPC++ TRANSLATION

SIMPLE KERNEL TRANSLATION

```
__global__ void VectorAddKernel(float* A, float* B, float* C)
{
	A[threadIdx.x] = threadIdx.x + 1.0f;
	B[threadIdx.x] = threadIdx.x + 1.0f;
	C[threadIdx.x] = A[threadIdx.x] + B[threadIdx.x];
}

void VectorAddKernel(float* A, float* B, float* C, sycl::nd_item<3> item_ct1)
{
	A[item_ct1.get_local_id(2)] = item_ct1.get_local_id(2) + 1.0f;
	B[item_ct1.get_local_id(2)] = item_ct1.get_local_id(2) + 1.0f;
	C[item_ct1.get_local_id(2)] =
	A[item_ct1.get_local_id(2)] + B[item_ct1.get_local_id(2)];
}
```

TEST 1: ISOLATED FILE

- Translated files for CUDA vector addition and vector-matrix multiplication
- 100% compilation and execution accuracy
- CUDA error handling dead code clean up for file readability

```
/* DPCT1003:30: Migrated API does not return error code. (*, 0) is inserted.
You may need to rewrite this code. */

checkCudaErrors((h_C = (float *)sycl::malloc_host(mem_size_C,
     dpct::get_default_queue()),0));

-> h_C = (float *)sycl::malloc_host(mem_size_C,dpct::get_default_queue());
```

TEST 2: FILE WITH HEADERS

- Matrix-matrix multiplication file with six included headers
- 98.7% compilation accuracy and 98.0% execution accuracy in the main file
- 10% of the code needed dead code touchups
- Header files had 100% compilation accuracy and execution accuracies ranging from 75%-100%

COMMON TRANSLATION ERROR

```
cudaGetDeviceCount(&device_count);

while (current_device < device_count)
{
      cudaGetDeviceProperties
        (&deviceProp, current_device);

      if (deviceProp.computeMode !=
            cudaComputeModeProhibited)
      { . . . }
      else {
            devices_prohibited++;
      }

      current_device++;
}
```

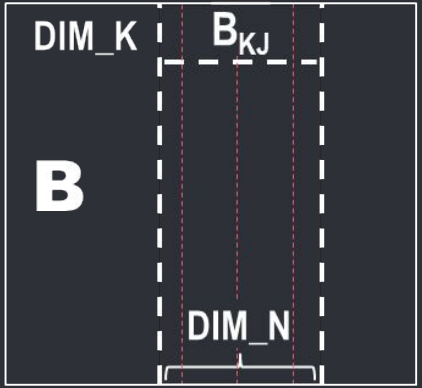
```
device_count =
dpct::dev_mgr::instance().device_count()

while (current_device < device_count)
{
      dpct::dev_mgr::instance()
       .get_device(current_device)
       .get_device_info(deviceProp);

      if (true)
      { . . . }

      else {
            devices_prohibited++;
      }

      current_device++;
}
```

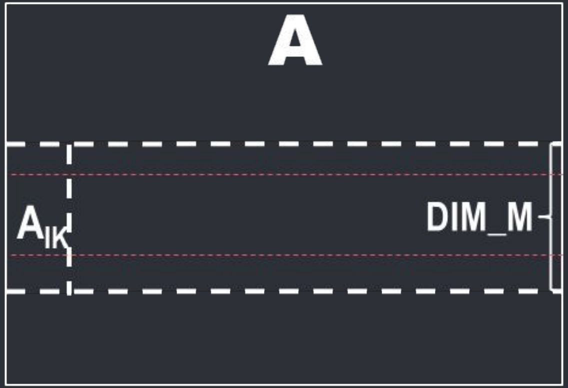
5 PORTING MAGMA SGEMM

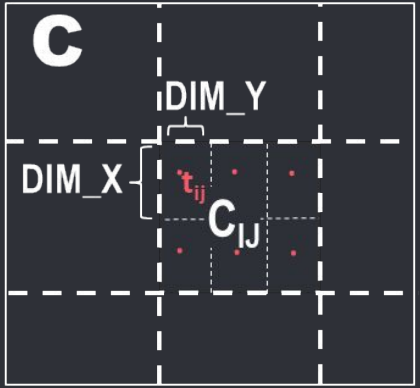
MAGMA GEMM

C = A B

```
template < DIM_X, DIM_Y,  DIM_M, DIM_N,
DIM_K, DIM_XA, DIM_YA, DIM_XB, DIM_YB>

For I = 1 .. M step DIM_M
    For J = 1 .. N step DIM_N
        For K = 1 .. K step DIM_K
            C_IJ += A_IK B_KJ
```







- Implementation is templated with **9 parameters**
- Computation is done with thread blocks of size **[DIM_X , DIM_Y]**
- Thread t_{ij} computes [DIM_M / DIM_X, DIM_N / DIM_Y] elements of C_{IJ}
- A_{IK} gets loaded in **shared memory** by [DIM_XA , DIM_YA] threads
- B_{KJ} gets loaded in **shared memory** by [DIM_XB , DIM_YB] threads
- C_{IJ} is held and computed in **registers**

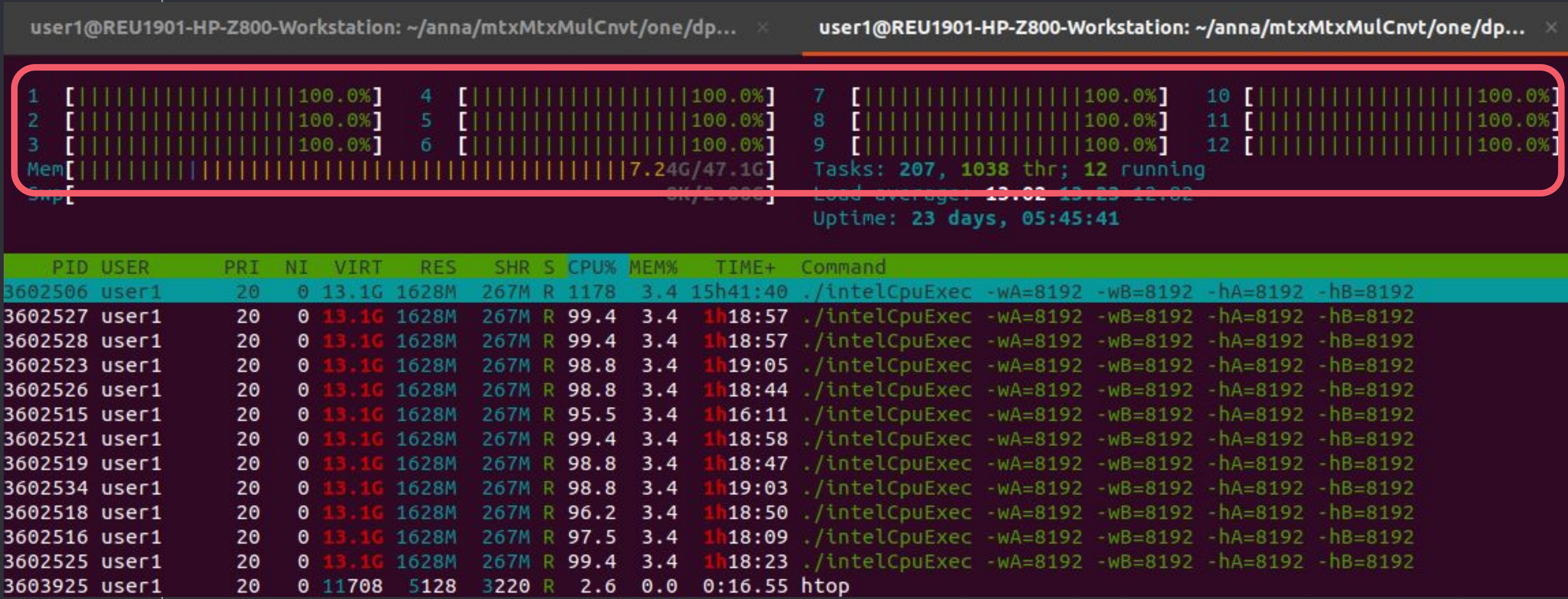
MAGMA SGEMM TRANSLATION PROCESS

- Collected MAGMA SGEMM CUDA code and dependencies in one directory
- Used DPCT to recursively migrate CUDA code to DPC++
- Translated header files that did not migrate independently in a separate directory and then copied them into the MAGMA SGEMM directory
- Implemented compiler directives as needed

6

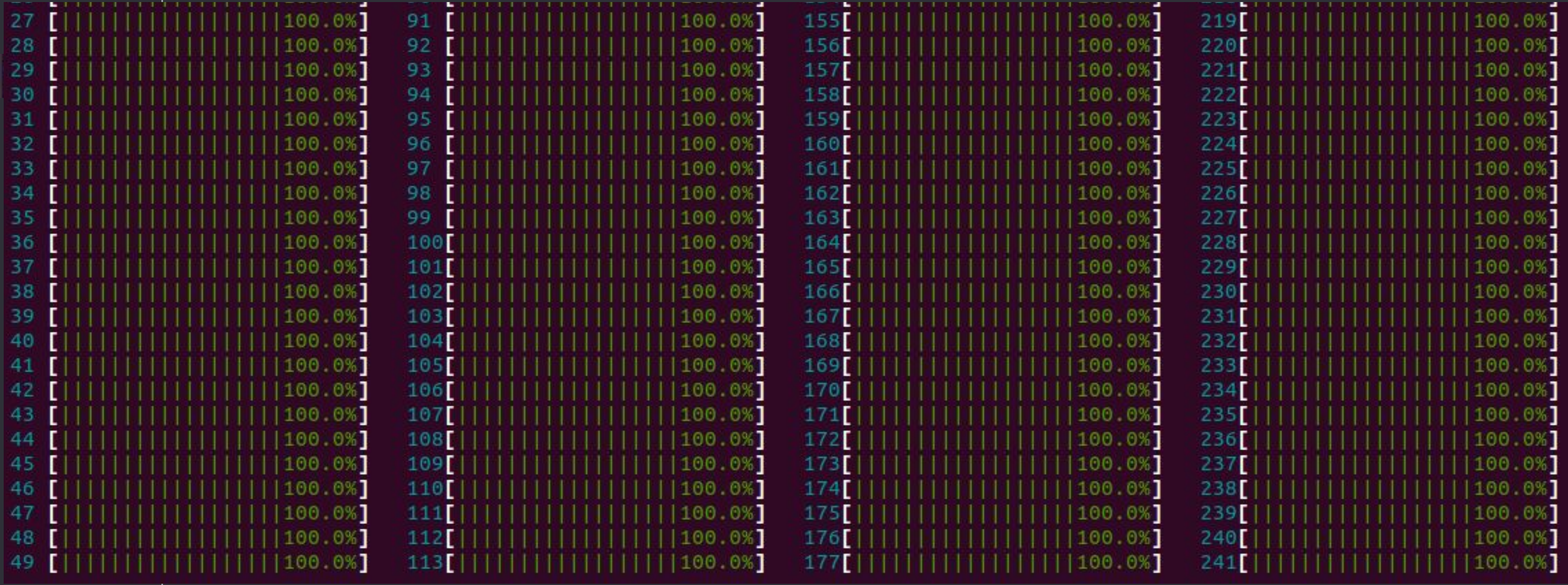
HARDWARE USAGE

MULTICORE CPUS



Intel(R) Xeon(R) CPU X5650 @ 2.67GHz

MULTICORE CPUS



AMD EPYC 7742 64-Core Processor

NVIDIA GPUS

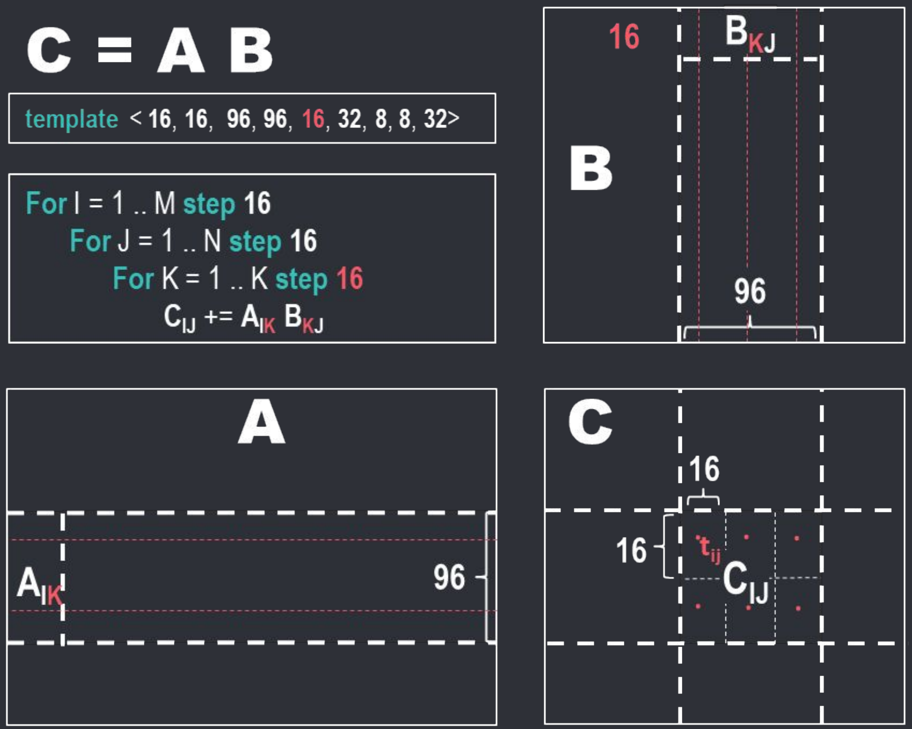
NVIDIA GeForce GTX 1650

```
Every 0.5s: nvidia-smi     REU1901-HP-Z800-Workstation: Fri Jul  8 10:16:58 202

Fri Jul  8 10:16:58 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.129.06   Driver Version: 470.129.06   CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| 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  NVIDIA GeForce ...  Off  | 00000000:0F:00.0  On |                  N/A |
| 35%   63C    P3    60W / 100W |   1345MiB /  3909MiB |    100%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      1239      G   /usr/lib/xorg/Xorg                 23MiB |
|    0   N/A  N/A    240692      G   /usr/lib/xorg/Xorg                241MiB |
|    0   N/A  N/A    240820      G   /usr/bin/gnome-shell               25MiB |
|    0   N/A  N/A    258953      G   ...RendererForSitePerProcess       13MiB |
|    0   N/A  N/A   3368334      G   /usr/lib/firefox/firefox          111MiB |
|    0   N/A  N/A   3634687      C   ...ffice/program/soffice.bin       47MiB |
|    0   N/A  N/A   3643058      C   ./cudaGpuExec                     862MiB |
+-----------------------------------------------------------------------------+
```

7 PERFORMANCE

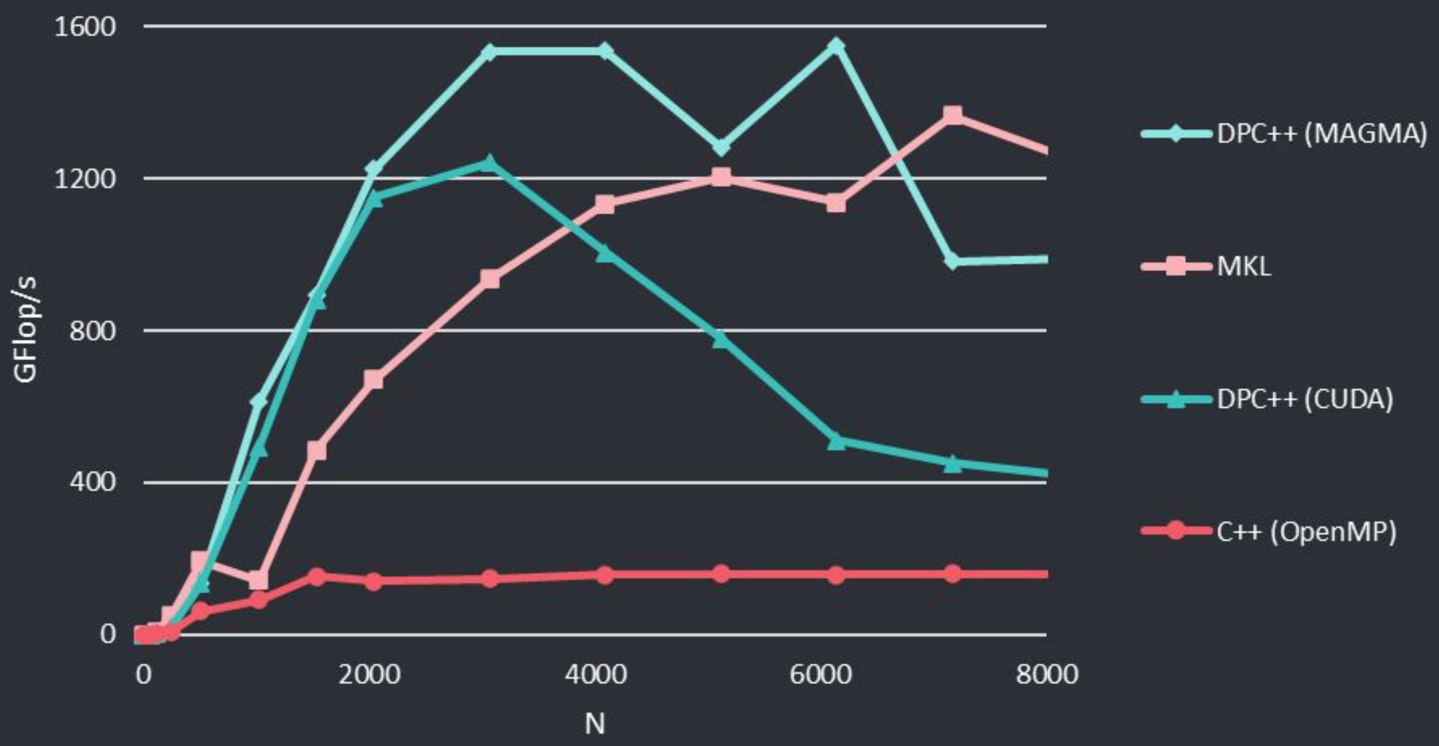
TEST PARAMETERS

cuda =
-DMAGMA_TUNING
-DDIM_X=16
-DDIM_Y=16
-DBLK_M_nn=96
-DBLK_N_nn=96
-DBLK_K_nn=16
-DDIM_XA=32
-DDIM_YA=8
-DDIM_XB=8
-DDIM_YB=32

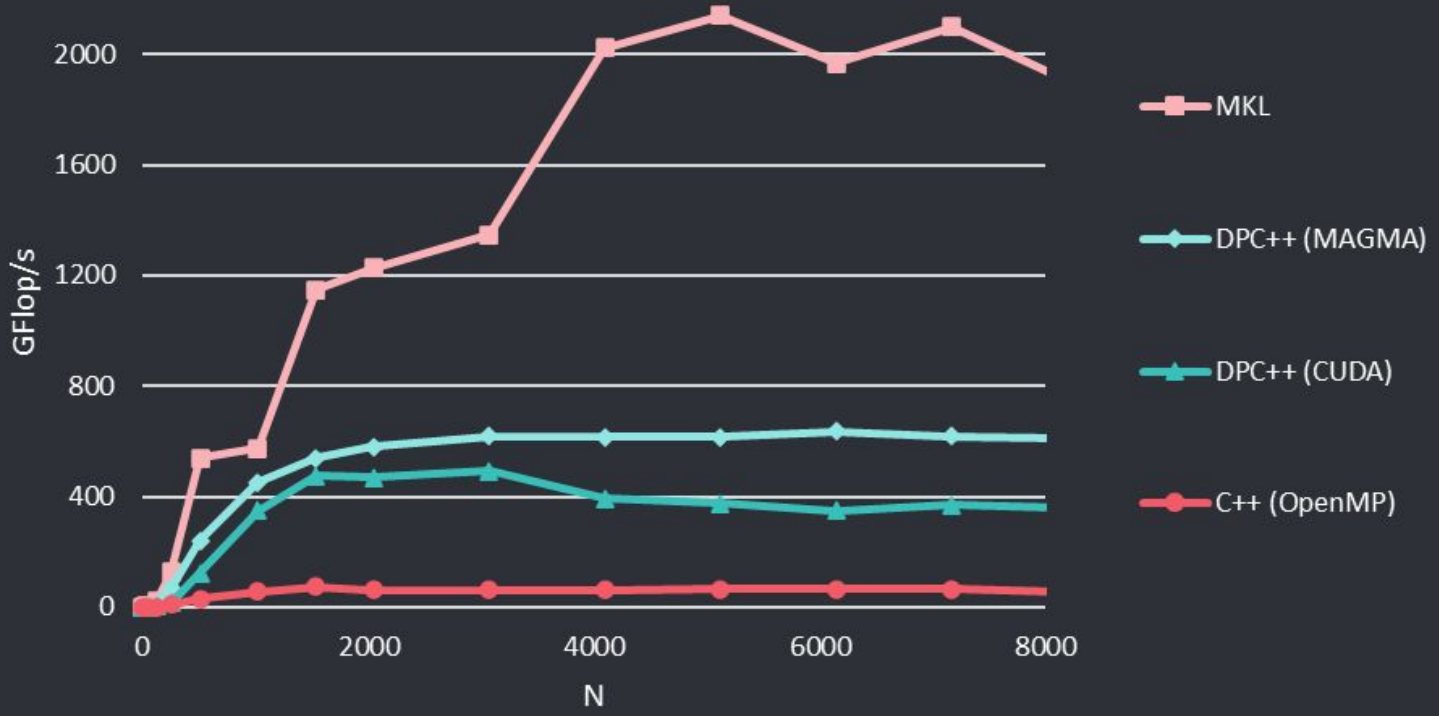


- Thread t_{ij} computes [96 / 16 , 96 / 16] elements of C_{IJ}
- A_{IK} gets loaded in shared memory by [32, 8] threads
- B_{KJ} gets loaded in shared memory by [8, 32] threads
- C_{IJ} is held and computed in registers

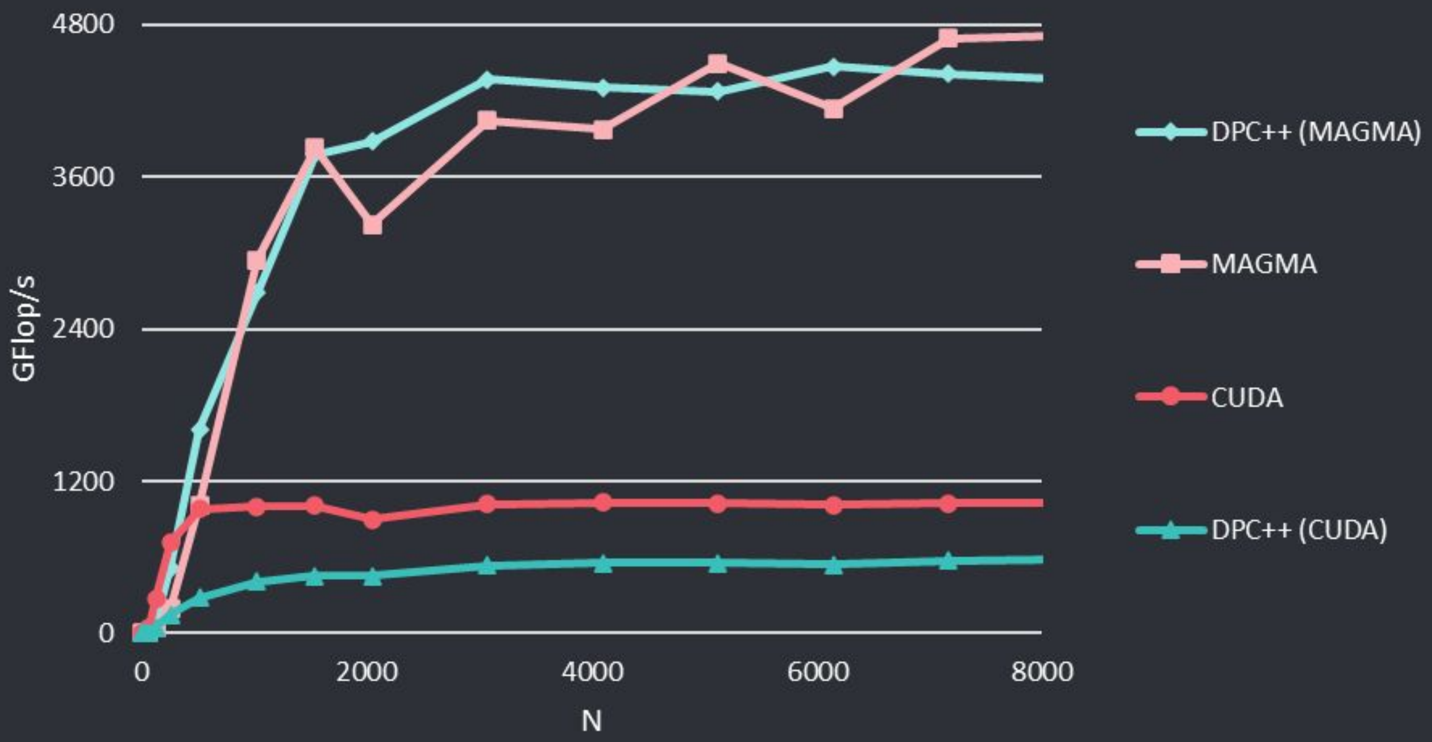
AMD EPYC 7742 64-CORE PROCESSOR @ 2.25GHZ



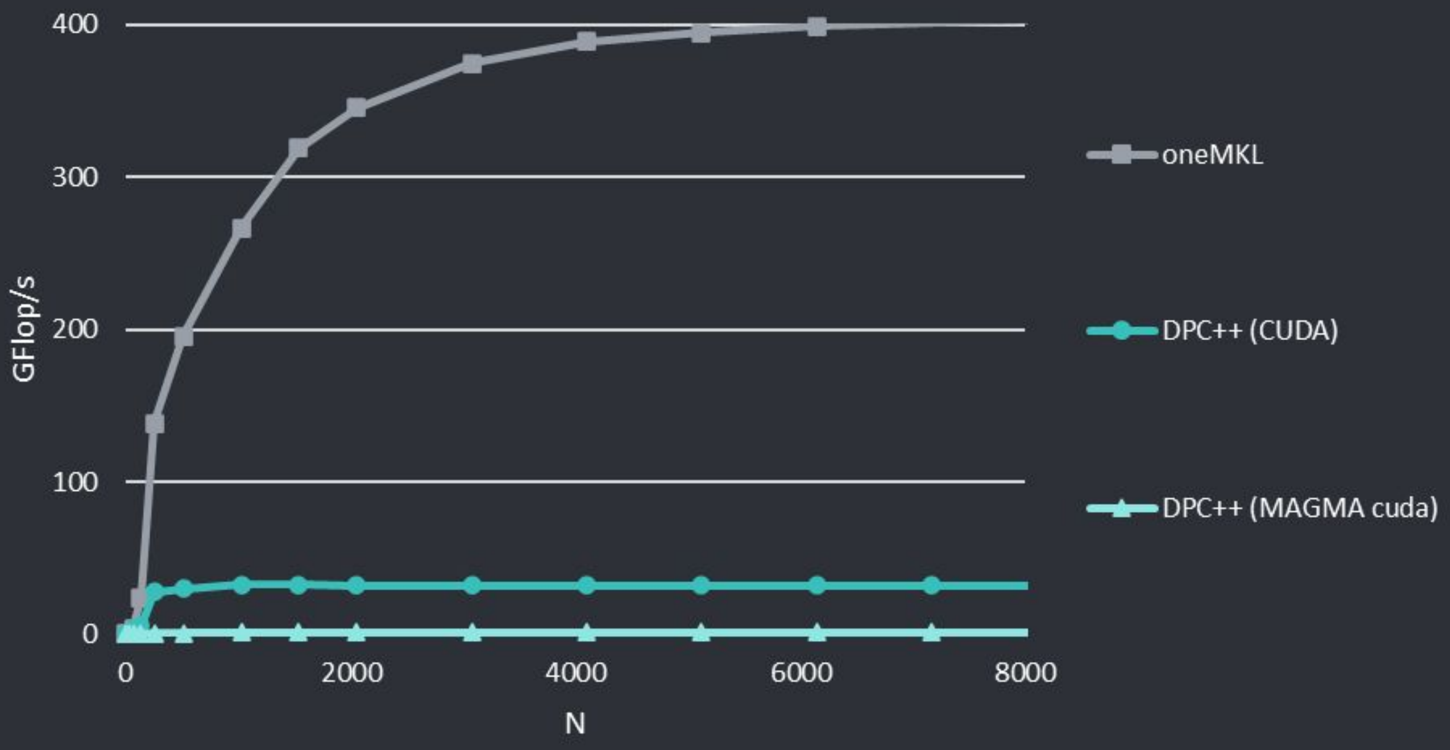
INTEL® XEON® CPU E5-2698 V4 20-CORE PROCESSOR @ 2.20GHZ



NVIDIA GEFORCE RTX 3060



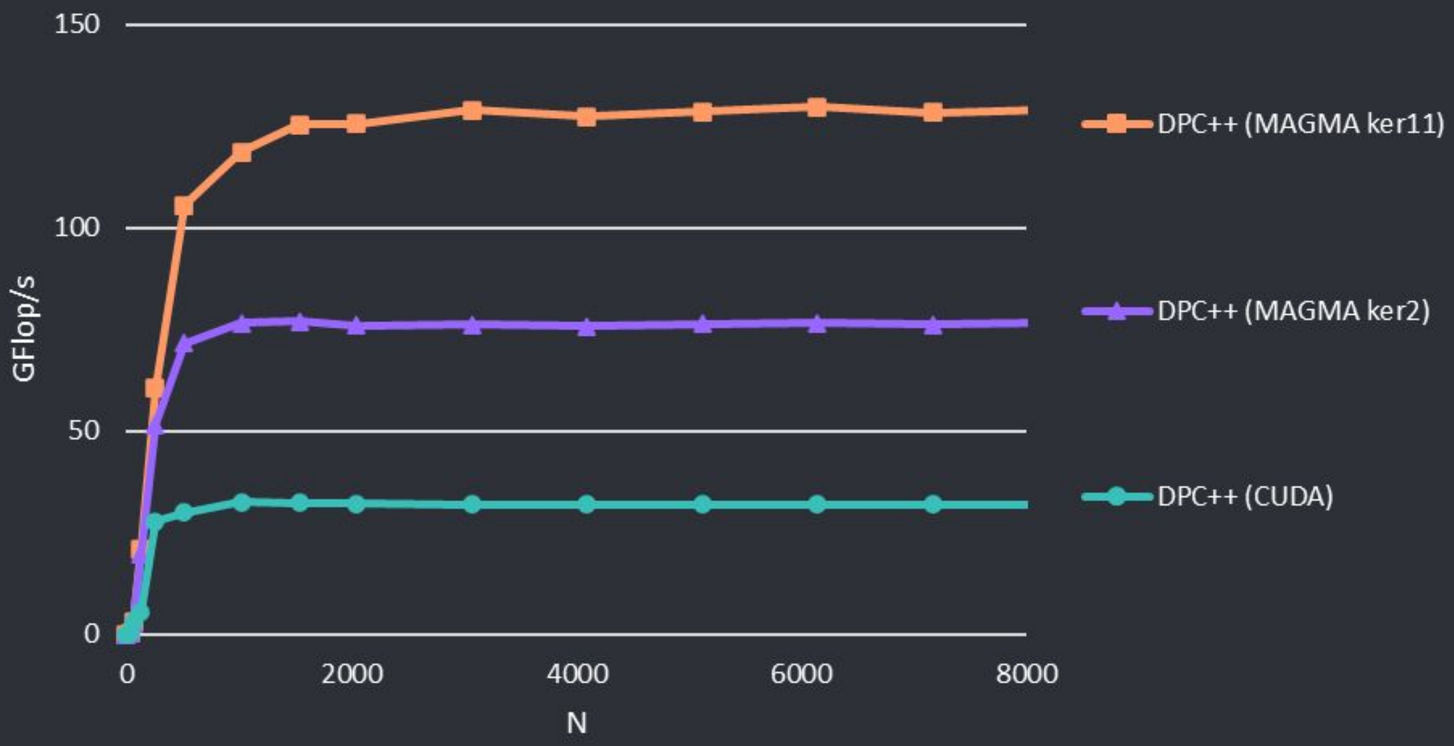
INTEL UHD GRAPHICS P630 [0x3e96]



ADDITIONAL TEST PARAMETERS

	DIM_X	DIM_Y	DIM_M	DIM_N	DIM_K	DIM_XA	DIM_YA	DIM_XB	DIM_YB
cuda	16	16	96	96	16	32	8	8	32
ker2	16	16	64	64	8	32	8	8	32
ker11	12	4	48	48	2	24	2	24	2

INTEL UHD GRAPHICS P630 [0x3e96]



8

CONCLUSION

SUMMARY

- oneAPI is a promising approach for parallel programming across various architectures
- DPCT tool can be used successfully for an initial port of CUDA code to DPC++
- Large numerical libraries like MAGMA, originally written in CUDA to support Nvidia GPUs, can be easily translated to DPC++ to provide functional portability to different vendor GPUs, as well as multicore CPUs

SUMMARY

- Initial migrated code tuned for Nvidia GPUs performs well on multicore CPUs
- Initial migrated code tuned for Nvidia GPUs retains performance on Nvidia GPUs
- Initial migrated code tuned for Nvidia GPUs performs poorly on the available Intel GPU
  - Tuning is required, but optimal parameters are difficult to find without further knowledge on the hardware design

ONGOING AND FUTURE WORKS

- Full translation of MAGMA
- ICL account configuration
- Finding near optimal parameters for the Intel integrated GPU
- Testing migrated code on discrete Intel GPU upon release

ACKNOWLEDGEMENTS

This research was funded by the National Science Foundation.

This research was conducted at the University of Tennessee at Knoxville through the RECSEM REU.

Home University

University of North Texas

REFERENCES

[1] *Advancing computing and data capabilities for scientific discovery and continued U.S. technological leadership.* Oak Ridge National Lab. *https://www.ornl.gov/directorate/ccsd*

[2] https://thenounproject.com/search/icons/?iconspage=1&q=quantum

[3] *Computing at LLNL.* Lawrence Livermore National Laboratory. https://computing.llnl.gov/

[4] *NVIDIA HISTORY.* Nvidia. https://www.nvidia.com/en-us/about-nvidia/corporate-timeline/

[5] *New Titan Supercomputer Named Fastest in the World.* Department of Energy. https://www.energy.gov/articles/new-titan-supercomputer-named-fastest-world-0

[6] *June 2019.* The Top 500 List. https://www.top500.org/lists/top500/2019/06/

[7] *June 2022.* The Top 500 List. https://www.top500.org/lists/top500/2022/06/

REFERENCES

[8] *Aurora: HPC and AI at Exascale.* Intel. https://www.intel.com/content/www/us/en/high-performance-computing/supercomputing/exascale-computing.html

[9] *Compare Benefits of CPUs, GPUs, and FPGAs for Different oneAPI Compute Workloads.* Intel. https://www.intel.com/content/www/us/en/developer/articles/technical/comparing-cpus-gpus-and-fpgas-for-oneapi.html#gs.83qstn

[10] *Intel oneAPI Programming Overview.* Intel. https://www.intel.com/content/www/us/en/develop/documentation/oneapi-programming-guide/top/introduction-to-oneapi-programming/intel-oneapi-programming-overview.html

[11] *Data Parallel C++: the oneAPI Implementation of SYCL*.* Intel. https://www.intel.com/content/www/us/en/developer/tools/oneapi/data-parallel-c-plus-plus.html#gs.83xmmq

REFERENCES

[12] *Intel® DPC++ Compatibility Tool.* Intel. https://www.intel.com/content/www/us/en/developer/tools/oneapi/dpc-compatibility-tool.html#gs.83zp77

[13] *oneMKL.* Intel. https://spec.oneapi.io/versions/latest/elements/oneMKL/source/index.html

[14] *What Is CUDA?* NVIDIA. https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/

[15] *Compiling SYCL* for Different GPUs.* Intel. https://www.intel.com/content/www/us/en/developer/articles/technical/compiling-sycl-with-different-gpus.html

[16] *AMD EPYC™ 7742.* AMD. https://www.amd.com/en/products/cpu/amd-epyc-7742

REFERENCES

[17] *Intel® Xeon® Processor E5-2698 v4.* Intel. https://ark.intel.com/content/www/us/en/ark/products/91753/intel-xeon-processor-e52698-v4-50m-cache-2-20-ghz.html

[18] *GEFORCE RTX 3060 FAMILY.* Nvidia. https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3060-3060ti/

[19] *Intel UHD Graphics P630.* TechPowerUp. https://www.techpowerup.com/gpu-specs/uhd-graphics-p630.c3676

[20] *Intel® DevCloud.* Intel. Intel® DevCloud

Presentation Template:

Catalina, J. (n.d.). Minimal business. Free PowerPoint Template & Google Slides theme. SlidesCarnival. Retrieved July 5, 2022, from https://www.slidescarnival.com/eleanor-free-presentation-template/308#preview