Analysis Report

mxnet::op::matrixMultiplyShared(float*, float*, float*, int, int, int, int, int, int, int, int, int, int, int)

Duration	2.93131 ms (2,931,312 ns)
Grid Size	[27,1,1000]
Block Size	[8,8,1]
Registers/Thread	72
Shared Memory/Block	8 KiB
Shared Memory Executed	0 B
Shared Memory Bank Size	4 B

[0] TITAN V

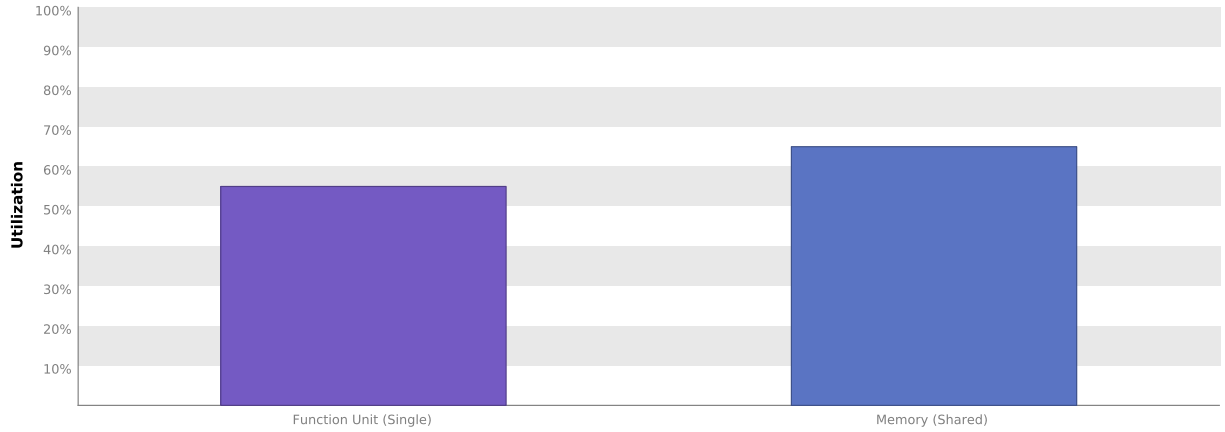
GPU UUID	GPU-01e2078d-caf5-3bc2-aeb7-80d9b3d6e701
Compute Capability	7.0
Max. Threads per Block	1024
Max. Threads per Multiprocessor	2048
Max. Shared Memory per Block	48 KiB
Max. Shared Memory per Multiprocessor	96 KiB
Max. Registers per Block	65536
Max. Registers per Multiprocessor	65536
Max. Grid Dimensions	[2147483647, 65535, 65535]
Max. Block Dimensions	[1024, 1024, 64]
Max. Warps per Multiprocessor	64
Max. Blocks per Multiprocessor	32
Half Precision FLOP/s	29.798 TeraFLOP/s
Single Precision FLOP/s	14.899 TeraFLOP/s
Double Precision FLOP/s	7.45 TeraFLOP/s
Number of Multiprocessors	80
Multiprocessor Clock Rate	1.455 GHz
Concurrent Kernel	true
Max IPC	4
Threads per Warp	32
Global Memory Bandwidth	652.8 GB/s
Global Memory Size	11.755 GiB
Constant Memory Size	64 KiB
L2 Cache Size	4.5 MiB
Memcpy Engines	7
PCIe Generation	3
PCIe Link Rate	8 Gbit/s
PCIe Link Width	8

1. Compute, Bandwidth, or Latency Bound

The first step in analyzing an individual kernel is to determine if the performance of the kernel is bounded by computation, memory bandwidth, or instruction/memory latency. The results below indicate that the performance of kernel "mxnet::op::matrixMultiplySh..." is most likely limited by both compute and memory bandwidth. You should first examine the information in the "Compute Resources" section to determine how it is limiting performance.

1.1. Kernel Performance Is Bound By Compute And Memory Bandwidth

For device "TITAN V" compute and memory utilization are balanced. These utilization levels indicate that kernel performance is good, but that additional performance improvement may be possible if either of both of compute and memory utilization levels are increased.



2. Compute Resources

GPU compute resources limit the performance of a kernel when those resources are insufficient or poorly utilized. Compute resources are used most efficiently when all threads in a warp have the same branching and predication behavior. The results below indicate that a significant fraction of the available compute performance is being wasted because branch and predication behavior is differing for threads within a warp. Compute resources are used most efficiently when instructions do not overuse a function unit. The results below indicate that compute performance may be limited by overuse of a function unit.

2.1. Divergent Branches

Compute resource are used most efficiently when all threads in a warp have the same branching behavior. When this does not occur the branch is said to be divergent. Divergent branches lower warp execution efficiency which leads to inefficient use of the GPU's compute resources.

Optimization: Each entry below points to a divergent branch within the kernel. For each branch reduce the amount of intra-warp divergence.

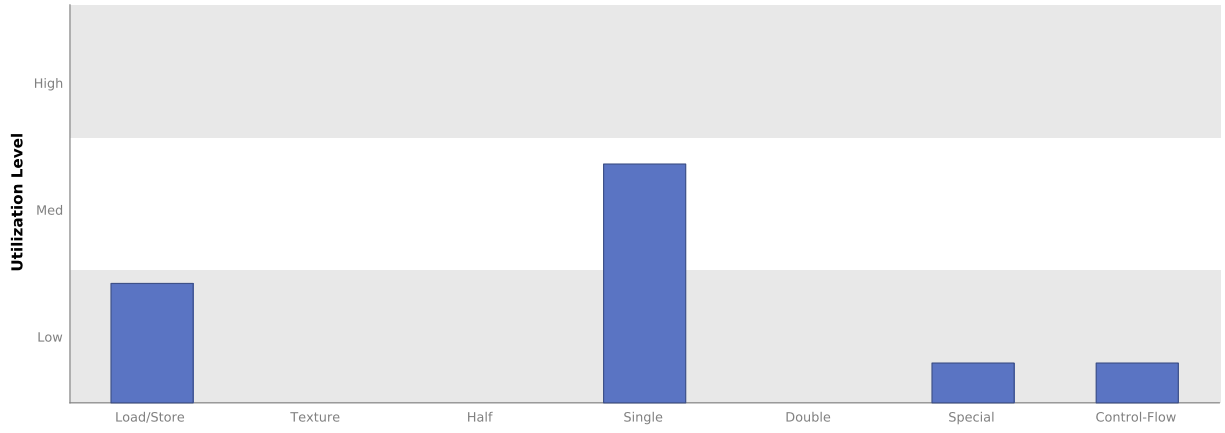
/mxnet/src/operator/custom/./new-forward.cuh

Line 147	Divergence = 0% [0 divergent executions out of 54000 total executions]
Line 148	Divergence = 0% [0 divergent executions out of 540000 total executions]
Line 148	Divergence = 0% [0 divergent executions out of 54000 total executions]
Line 213	Divergence = 51.9% [28000 divergent executions out of 54000 total executions]
Line 225	Divergence = 0% [0 divergent executions out of 54000 total executions]

2.2. GPU Utilization Is Limited By Function Unit Usage

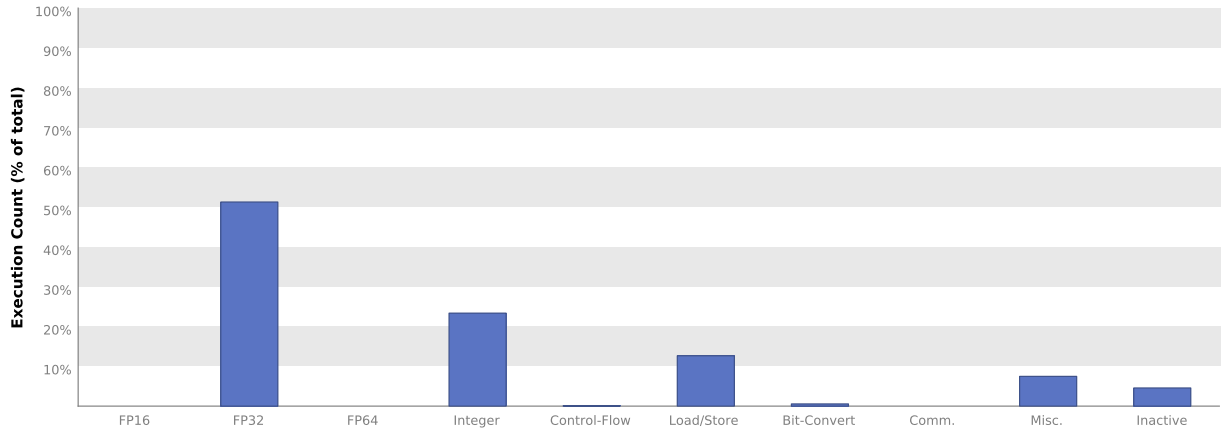
Different types of instructions are executed on different function units within each SM. Performance can be limited if a function unit is over-used by the instructions executed by the kernel. The following results show that the kernel's performance is potentially limited by overuse of the following function units: Single.

Load/Store - Load and store instructions for shared and constant memory.
Texture - Load and store instructions for local, global, and texture memory.
Half - Half-precision floating-point arithmetic instructions.
Single - Single-precision integer and floating-point arithmetic instructions.
Double - Double-precision floating-point arithmetic instructions.
Special - Special arithmetic instructions such as sin, cos, popc, etc.
Control-Flow - Direct and indirect branches, jumps, and calls.



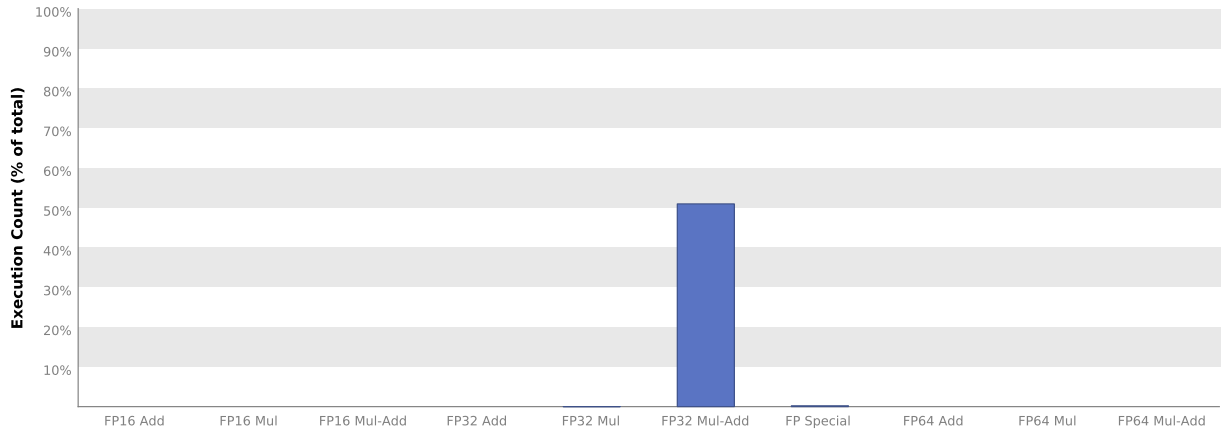
2.3. Instruction Execution Counts

The following chart shows the mix of instructions executed by the kernel. The instructions are grouped into classes and for each class the chart shows the percentage of thread execution cycles that were devoted to executing instructions in that class. The "Inactive" result shows the thread executions that did not execute any instruction because the thread was predicated or inactive due to divergence.



2.4. Floating-Point Operation Counts

The following chart shows the mix of floating-point operations executed by the kernel. The operations are grouped into classes and for each class the chart shows the percentage of thread execution cycles that were devoted to executing operations in that class. The results do not sum to 100% because non-floating-point operations executed by the kernel are not shown in this chart.



3. Memory Bandwidth

Memory bandwidth limits the performance of a kernel when one or more memories in the GPU cannot provide data at the rate requested by the kernel. The results below indicate that the kernel is limited by the bandwidth available to the shared memory.

3.1. Shared Memory Alignment and Access Pattern

Memory bandwidth is used most efficiently when each shared memory load and store has proper alignment and access pattern.

Optimization: Select each entry below to open the source code to a shared load or store within the kernel with an inefficient alignment or access pattern. For each load or store improve the alignment and access pattern of the memory access.

/mxnet/src/operator/custom/./new-forward.cuh

Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 182	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total

	executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 188	Shared Store Transactions/Access = 4, Ideal Transactions/Access = 1 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total

	executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total

	executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total

	executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 4, Ideal Transactions/Access = 4 [2160000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]
Line 202	Shared Load Transactions/Access = 2, Ideal Transactions/Access = 2 [1080000 transactions for 540000 total executions]

3.2. GPU Utilization Is Limited By Memory Bandwidth

The following table shows the memory bandwidth used by this kernel for the various types of memory on the device. The table also shows the utilization of each memory type relative to the maximum throughput supported by the memory. The results show that the kernel's performance is potentially limited by the bandwidth available from one or more of the memories on the device.

Optimization: Try the following optimizations for the memory with high bandwidth utilization.
Shared Memory - If possible use 64-bit accesses to shared memory and 8-byte bank mode to achieved 2x throughput.
L2 Cache - Align and block kernel data to maximize L2 cache efficiency.
Unified Cache - Reallocate texture data to shared or global memory. Resolve alignment and access pattern issues for global loads and stores.
Device Memory - Resolve alignment and access pattern issues for global loads and stores.

System Memory (via PCIe) - Make sure performance critical data is placed in device or shared memory.

Transactions	Bandwidth	Utilization	
Shared Memory			
Shared Loads	140759189	6,146.455 GB/s	
Shared Stores	71072477	3,103.483 GB/s	
Shared Total	211831666	9,249.938 GB/s	Idle Low Medium High Max
L2 Cache			
Reads	41110086	448.783 GB/s	
Writes	11388016	124.319 GB/s	
Total	52498102	573.101 GB/s	Idle Low Medium High Max
Unified Cache			
Local Loads	0	0 B/s	
Local Stores	0	0 B/s	
Global Loads	269540486	2,942.469 GB/s	
Global Stores	11388000	124.318 GB/s	
Texture Reads	154015490	6,725.31 GB/s	
Unified Total	434943976	9,792.098 GB/s	Idle Low Medium High Max
Device Memory			
Reads	1636423	17.864 GB/s	
Writes	2532071	27.642 GB/s	
Total	4168494	45.506 GB/s	Idle Low Medium High Max
System Memory			
[PCIe configuration: Gen3 x8, 8 Gbit/s]			
Reads	0	0 B/s	Idle Low Medium High Max
Writes	5	54.583 kB/s	Idle Low Medium High Max

3.3. Memory Statistics

The following chart shows a summary view of the memory hierarchy of the CUDA programming model. The green nodes in the diagram depict logical memory space whereas blue nodes depicts actual hardware unit on the chip. For the various caches the reported percentage number states the cache hit rate; that is the ratio of requests that could be served with data locally available to the cache over all requests made.
The links between the nodes in the diagram depict the data paths between the SMs to the memory spaces into the memory system. Different metrics are shown per data path. The data paths from the SMs to the memory spaces report the total number of memory instructions executed, it includes both read and write operations. The data path between memory spaces and "Unified Cache" or "Shared Memory" reports the total amount of memory requests made (read or write). All other data paths report the total amount of transferred memory in bytes.

4. Instruction and Memory Latency

Instruction and memory latency limit the performance of a kernel when the GPU does not have enough work to keep busy. The performance of latency-limited kernels can often be improved by increasing occupancy. Occupancy is a measure of how many warps the kernel has active on the GPU, relative to the maximum number of warps supported by the GPU. Theoretical occupancy provides an upper bound while achieved occupancy indicates the kernel's actual occupancy. The results below indicate that occupancy can be improved by reducing the amount of shared memory used by the kernel.

4.1. GPU Utilization Is Limited By Shared Memory Usage

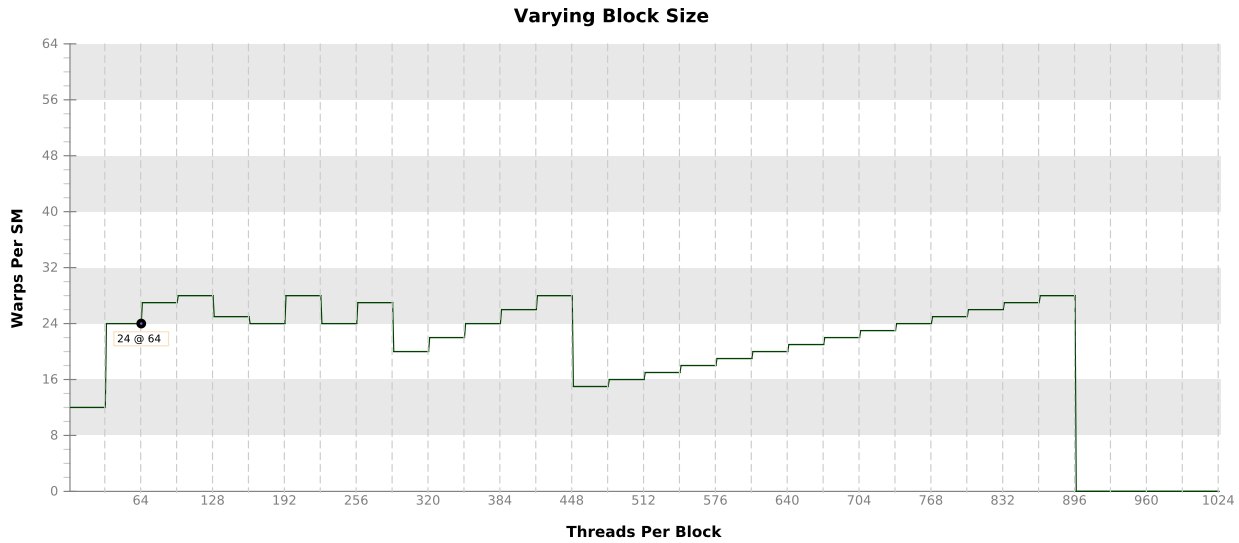
The kernel uses 8 KiB of shared memory for each block. This shared memory usage is likely preventing the kernel from fully utilizing the GPU. Device "TITAN V" is configured to have 96 KiB of shared memory for each SM. Because the kernel uses 8 KiB of shared memory for each block each SM is limited to simultaneously executing 12 blocks (24 warps). Chart "Varying Shared Memory Usage" below shows how changing shared memory usage will change the number of blocks that can execute on each SM.

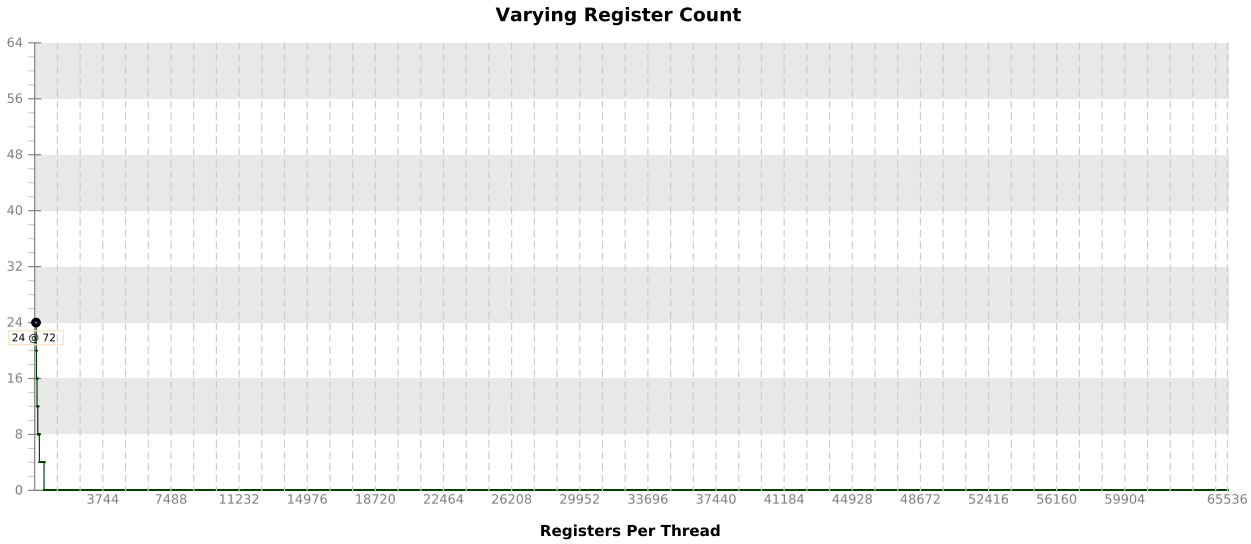
Optimization: Reduce shared memory usage to increase the number of blocks that can execute on each SM. You can also increase the number of blocks that can execute on each SM by increasing the amount of shared memory available to your kernel. You do this by setting the preferred cache configuration to "prefer shared".

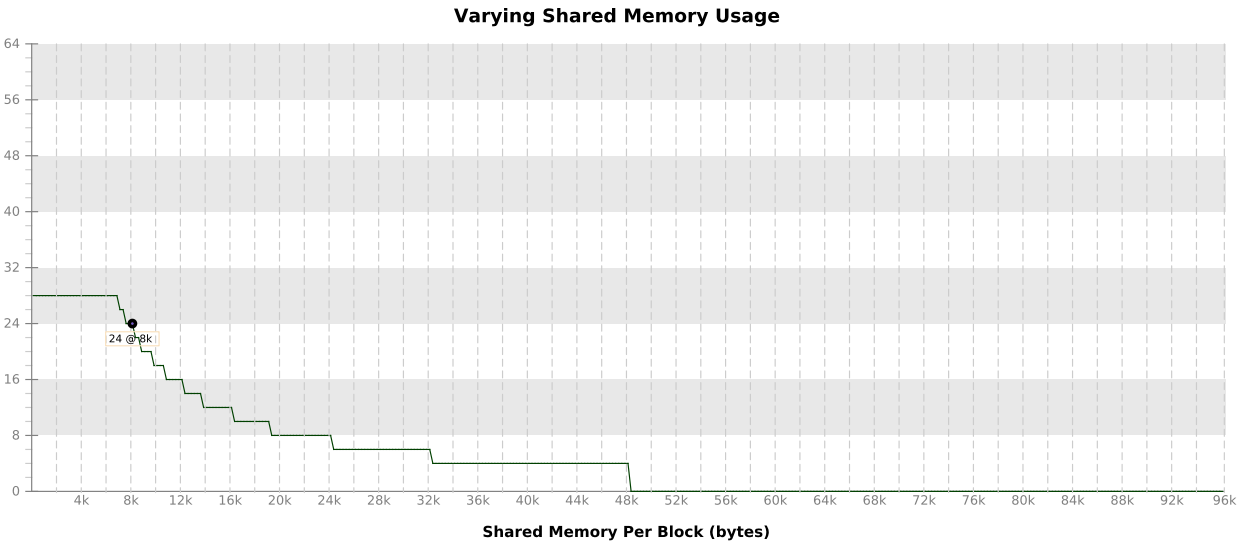
Variable	Achieved	Theoretical	Device Limit	Grid Size: [27,1,1000] (27000 blocks) Block Size: [8,8,1] (64 thre
Occupancy Per SM				
Active Blocks		12	32	0 3 6 9 12 15 18 21 24 27 30 32
Active Warps	23.69	24	64	0 7 14 21 28 35 42 49 56 64
Active Threads		768	2048	0 256 512 768 1024 1280 1536 1792 2048
Occupancy	37%	37.5%	100%	0% 25% 50% 75% 100%
Warps				
Threads/Block		64	1024	0 128 256 384 512 640 768 896 1024
Warps/Block		2	32	0 3 6 9 12 15 18 21 24 27 30 32
Block Limit		32	32	0 3 6 9 12 15 18 21 24 27 30 32
Registers				
Registers/Thread		72	65536	0 8192 16384 24576 32768 40960 49152 57344 65536
Registers/Block		4608	65536	0 16k 32k 48k 64k
Block Limit		14	32	0 3 6 9 12 15 18 21 24 27 30 32
Shared Memory				
Shared Memory/Block		8192	98304	0 32k 64k 96k
Block Limit		12	32	0 3 6 9 12 15 18 21 24 27 30 32

4.2. Occupancy Charts

The following charts show how varying different components of the kernel will impact theoretical occupancy.


Varying Block Size
Warps Per SM
64
56
48
40
32
24
16
8
0
24 @ 64
64
128
192
256
320
384
448
512
576
640
704
768
832
896
960
1024
Threads Per Block


Varying Register Count
64
56
48
40
32
24
16
8
0
24 @ 72
3744
7488
11232
14976
18720
22464
26208
29952
33696
37440
41184
44928
48672
52416
56160
59904
65536
Registers Per Thread



4.3. Multiprocessor Utilization

The kernel's blocks are distributed across the GPU's multiprocessors for execution. Depending on the number of blocks and the execution duration of each block some multiprocessors may be more highly utilized than others during execution of the kernel. The following chart shows the utilization of each multiprocessor during execution of the kernel.