Analysis Report

void computeHOGSharedPred<float, int=8, int=8, int=16, int=16, int=64>(float const *, float const *, float*, float const *, int, int, int, int)

Duration	1.489 ms (1,489,498 ns)
Grid Size	[70,1,1]
Block Size	[256,1,1]
Registers/Thread	40
Shared Memory/Block	36 KiB
Shared Memory Requested	96 KiB
Shared Memory Executed	96 KiB
Shared Memory Bank Size	4 B

[0] GeForce GTX 960

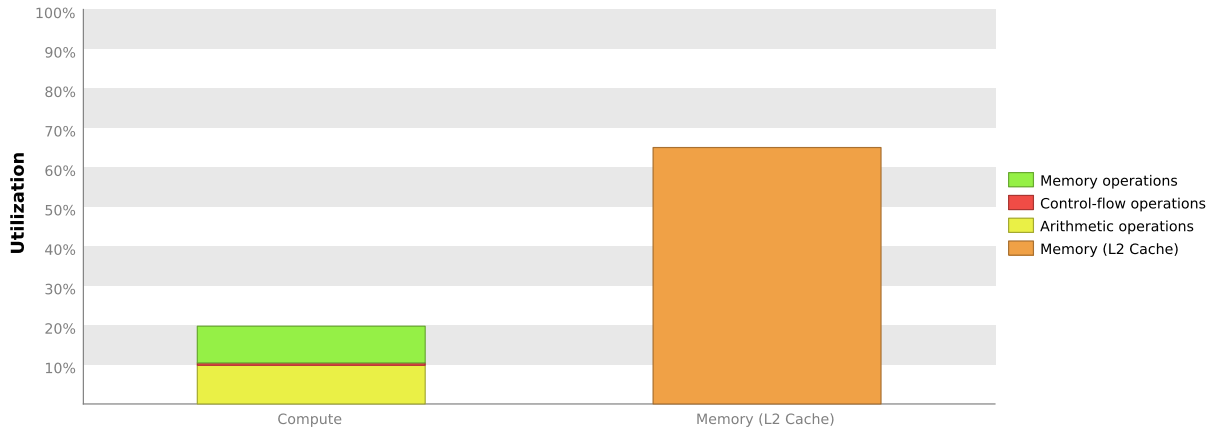
GPU UUID	GPU-0db32734-f94e-48a7-8b5d-4604317dc554
Compute Capability	5.2
Max. Threads per Block	1024
Max. Shared Memory per Block	48 KiB
Max. Registers per Block	65536
Max. Grid Dimensions	[2147483647, 65535, 65535]
Max. Block Dimensions	[1024, 1024, 64]
Max. Warps per Multiprocessor	64
Max. Blocks per Multiprocessor	32
Single Precision FLOP/s	2.644 TeraFLOP/s
Double Precision FLOP/s	82.624 GigaFLOP/s
Number of Multiprocessors	8
Multiprocessor Clock Rate	1.291 GHz
Concurrent Kernel	true
Max IPC	6
Threads per Warp	32
Global Memory Bandwidth	112.16 GB/s
Global Memory Size	4 GiB
Constant Memory Size	64 KiB
L2 Cache Size	1 MiB
Memcpy Engines	2
PCIe Generation	2
PCIe Link Rate	5 Gbit/s
PCIe Link Width	16

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 "void computeHOGSharedPred<f..." is most likely limited by memory bandwidth. You should first examine the information in the "Memory Bandwidth" section to determine how it is limiting performance.

1.1. Kernel Performance Is Bound By Memory Bandwidth

For device "GeForce GTX 960" the kernel's compute utilization is significantly lower than its memory utilization. These utilization levels indicate that the performance of the kernel is most likely being limited by the memory system. For this kernel the limiting factor in the memory system is the bandwidth of the L2 Cache memory.



2. 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 L2 cache.

2.1. Global Memory Alignment and Access Pattern

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

Optimization: Each entry below points to a global 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.

/home/adas/cuda-workspace/CudaVisionSysDeploy/Release/../src/init/../device/HOG/HOGdescriptor.cuh

Line 149	Global Load L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [4576000 L2 transactions for 143104 total executions]
Line 149	Global Load L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [4576000 L2 transactions for 143104 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total

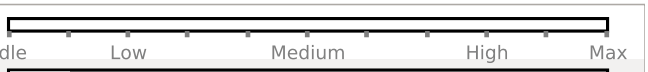
	executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total executions]
Line 159	Global Store L2 Transactions/Access = 32, Ideal Transactions/Access = 4 [17875 L2 transactions for 559 total

	executions]

2.2. 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	2636446	250.562 GB/s	
Shared Stores	2636626	250.58 GB/s	
Shared Total	5273072	501.142 GB/s	
L2 Cache			
Reads	9187697	218.295 GB/s	
Writes	643506	15.289 GB/s	
Total	9831203	233.584 GB/s	
Unified Cache			
Local Loads	0	0 B/s	
Local Stores	0	0 B/s	
Global Loads	10689680	253.981 GB/s	
Global Stores	643500	15.289 GB/s	
Texture Reads	2682000	63.723 GB/s	
Unified Total	14015180	332.994 GB/s	
Device Memory			
Reads	552860	13.136 GB/s	
Writes	109485	2.601 GB/s	
Total	662345	15.737 GB/s	
System Memory			
[PCIe configuration: Gen2 x16, 5 Gbit/s]			
Reads	0	0 B/s	
Writes	5	118.797 kB/s	

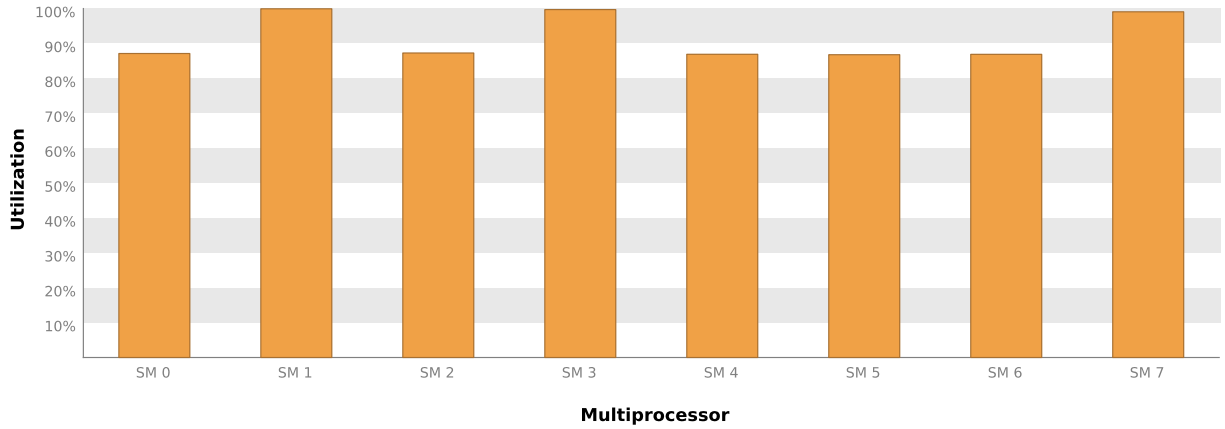
3. Instruction and Memory Latency

Instruction and memory latency limit the performance of a kernel when the GPU does not have enough work to keep busy. The results below indicate that the GPU does not have enough work because differences in the execution time of the kernel's blocks leads to poor load balancing across the SMs.

3.1. Achieved Occupancy Is Low

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 kernel's achieved occupancy of 11.8% is significantly lower than its theoretical occupancy of 25%. Most likely this indicates that there is an imbalance in how the kernel's blocks are executing on the SMs so that all SMs are not equally busy over the entire execution of the kernel. The following chart shows the utilization of each multiprocessor during execution of the kernel.

Optimization: Make sure that all blocks are doing roughly the same amount of work. It may also help to increase the number of blocks executed by the kernel.



3.2. GPU Utilization Is Limited By Shared Memory Usage

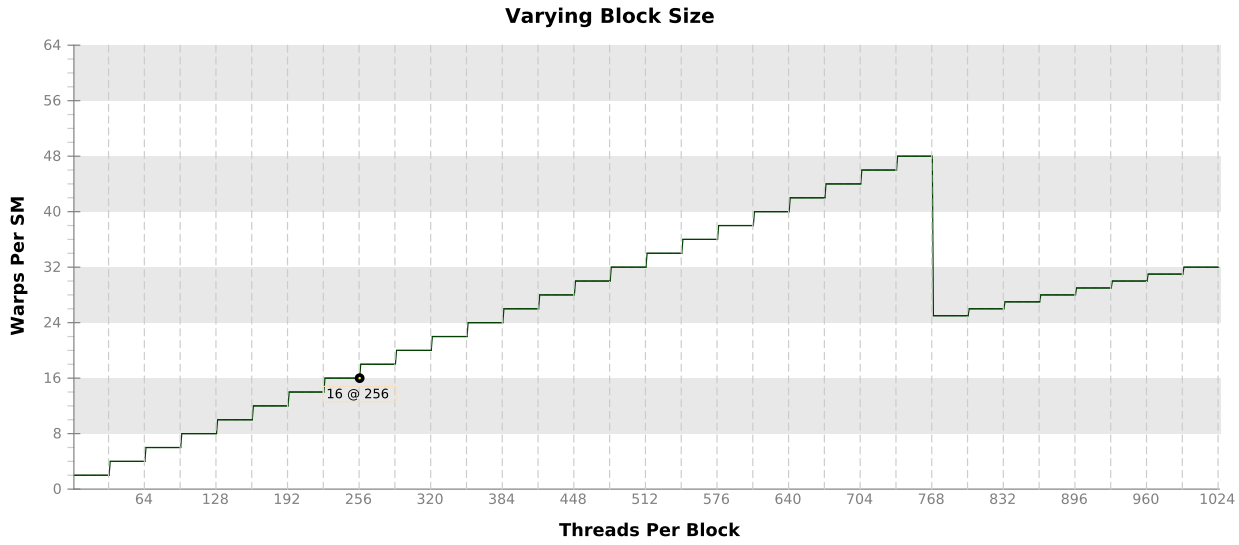
The kernel uses 36 KiB of shared memory for each block. This shared memory usage is likely preventing the kernel from fully utilizing the GPU. Device "GeForce GTX 960" is configured to have 96 KiB of shared memory for each SM. Because the kernel uses 36 KiB of shared memory for each block each SM is limited to simultaneously executing 2 blocks (16 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.

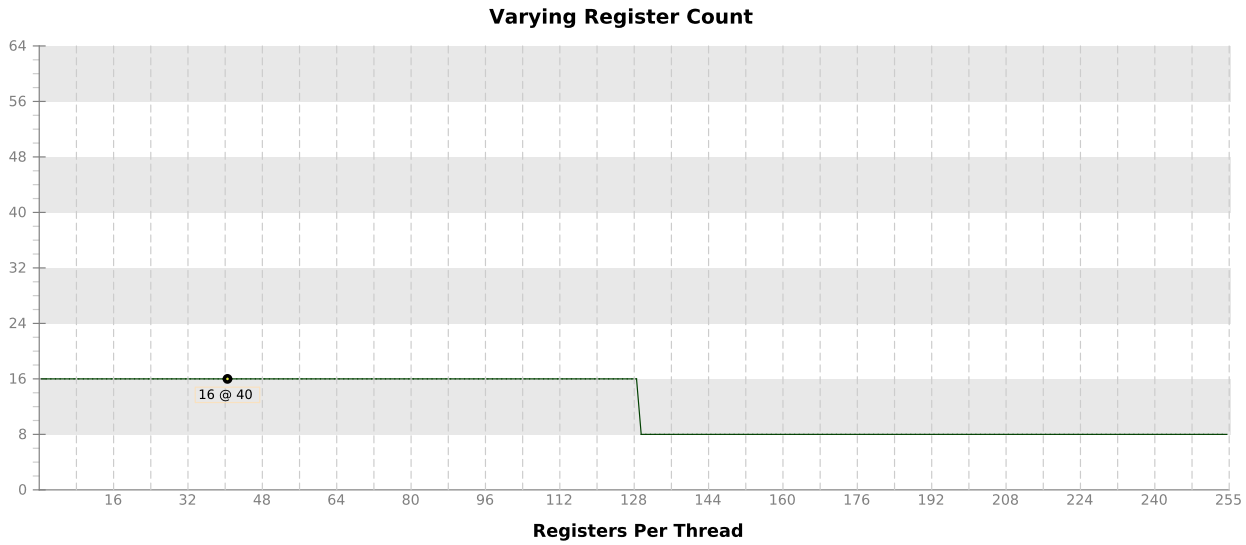
Optimization: Reduce shared memory usage to increase the number of blocks that can execute on each SM. You can also increase the number of blocks that can execute on each SM by increasing the amount of shared memory available to your kernel. You do this by setting the preferred cache configuration to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

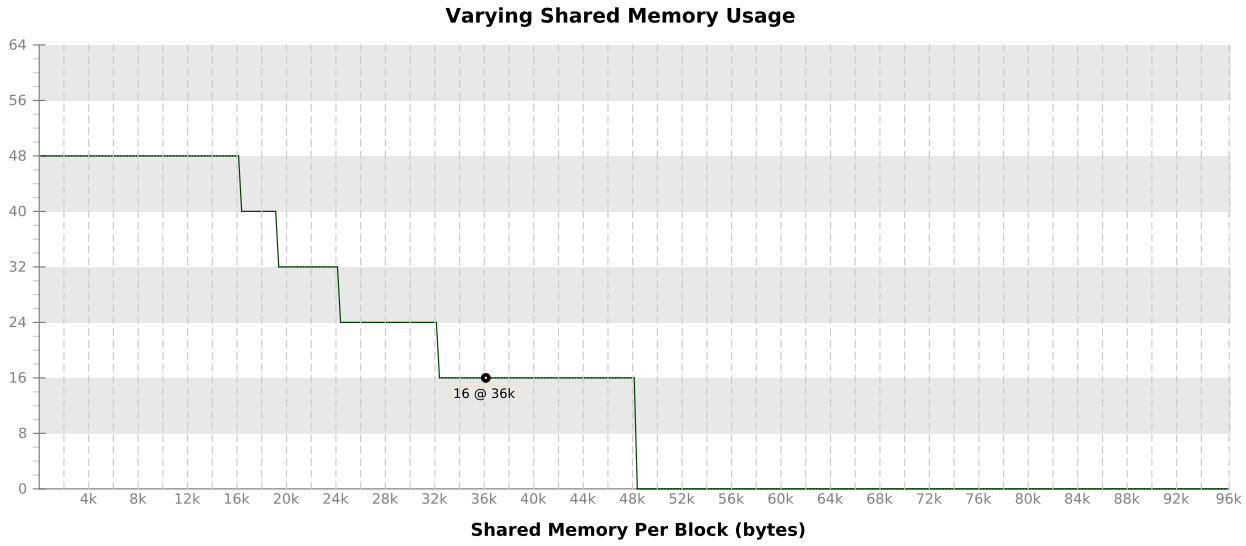
Variable	Achieved	Theoretical	Device Limit	Grid Size: [70,1,1] (70 blocks) Block Size: [256,1,1] (256 threads
Occupancy Per SM				
Active Blocks		2	32	0 3 6 9 12 15 18 21 24 27 30 32
Active Warps	7.55	16	64	0 7 14 21 28 35 42 49 56 64
Active Threads		512	2048	0 256 512 768 1024 1280 1536 1792 2048
Occupancy	11.8%	25%	100%	0% 25% 50% 75% 100%
Warps				
Threads/Block		256	1024	0 128 256 384 512 640 768 896 1024
Warps/Block		8	32	0 3 6 9 12 15 18 21 24 27 30 32
Block Limit		8	32	0 3 6 9 12 15 18 21 24 27 30 32
Registers				
Registers/Thread		40	255	0 32 64 96 128 160 192 224 255
Registers/Block		10240	65536	0 16k 32k 48k 64k
Block Limit		6	32	0 3 6 9 12 15 18 21 24 27 30 32
Shared Memory				
Shared Memory/Block		36864	98304	0 32k 64k 96k
Block Limit		2	32	0 3 6 9 12 15 18 21 24 27 30 32

3.3. Occupancy Charts

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




Varying Register Count
64
56
48
40
32
24
16
8
0
16 @ 40
16
32
48
64
80
96
112
128
144
160
176
192
208
224
240
255
Registers Per Thread


Varying Shared Memory Usage
64
56
48
40
32
24
16
8
0
16 @ 36k
4k
8k
12k
16k
20k
24k
28k
32k
36k
40k
44k
48k
52k
56k
60k
64k
68k
72k
76k
80k
84k
88k
92k
96k
Shared Memory Per Block (bytes)

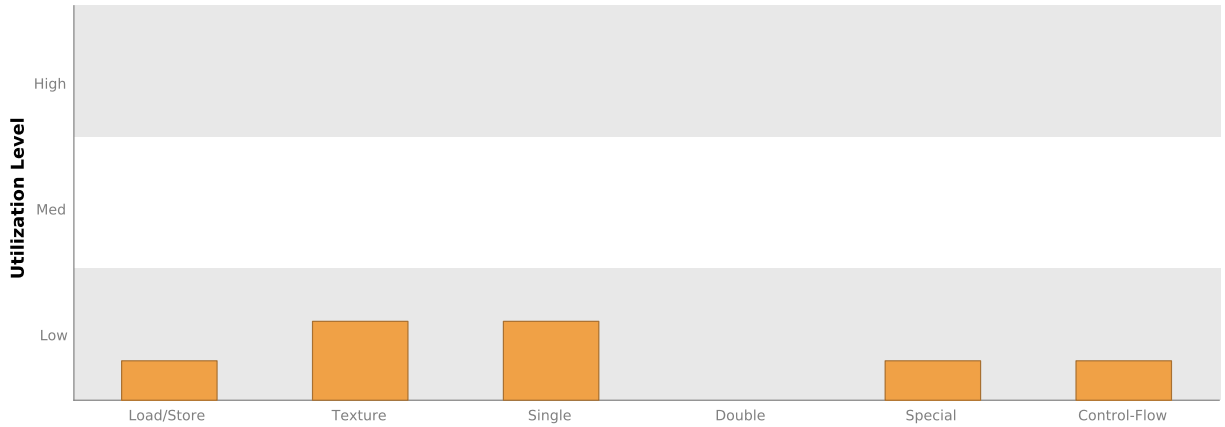
4. Compute Resources

GPU compute resources limit the performance of a kernel when those resources are insufficient or poorly utilized.

4.1. Function Unit Utilization

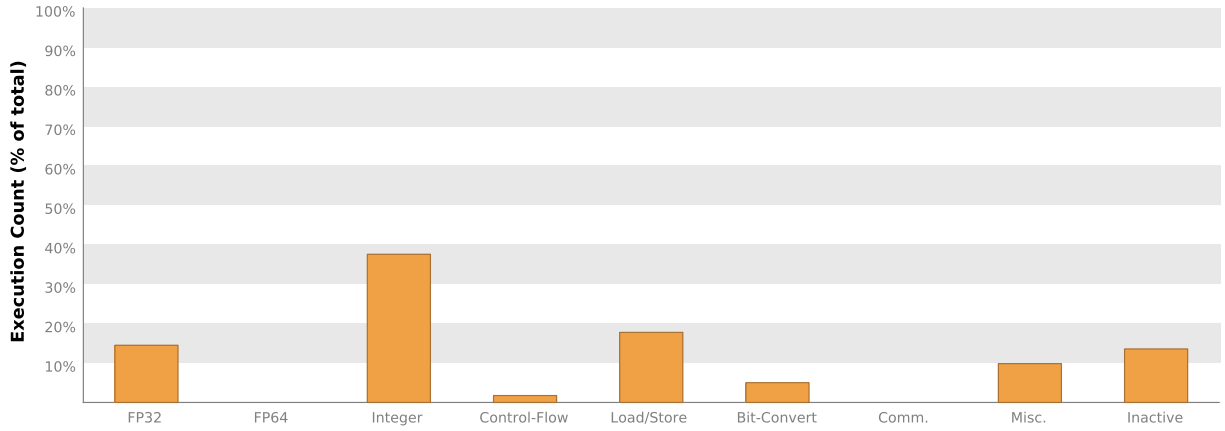
Different types of instructions are executed on different function units within each SM. Performance can be limited if a function unit is over-used by the instructions executed by the kernel. The following results show that the kernel's performance is not limited by overuse of any function unit.

Load/Store - Load and store instructions for shared and constant memory.
Texture - Load and store instructions for local, global, and texture memory.
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.



4.2. 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.



4.3. 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.