Analysis Report

DHashData(unsigned char*, unsigned long, unsigned char*, unsigned long, unsigned long, int)

Duration	306.308 µs
Grid Size	[1024,1,1]
Block Size	[2,1,1]
Registers/Thread	90
Shared Memory/Block	0 B
Shared Memory Requested	48 KiB
Shared Memory Executed	48 KiB
Shared Memory Bank Size	4 B

[0] Tesla K20c

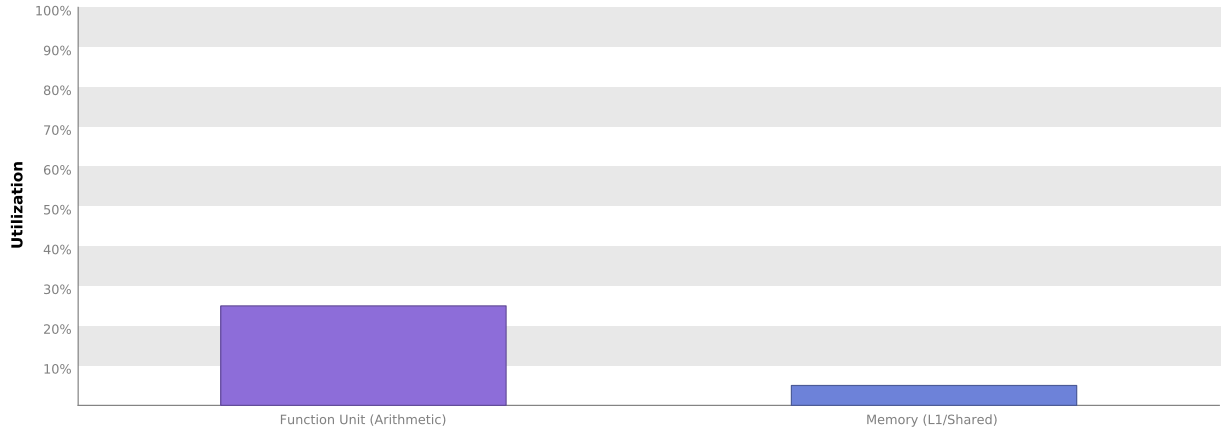
GPU UUID	GPU-8755f09f-ebc6-0c4c-349a-22c22f181d94
Compute Capability	3.5
Max. Threads per Block	1024
Max. Shared Memory per Block	48 KiB
Max. Registers per Block	65536
Max. Grid Dimensions	[2147483647, 65535, 65535]
Max. Block Dimensions	[1024, 1024, 64]
Max. Warps per Multiprocessor	64
Max. Blocks per Multiprocessor	16
Single Precision FLOP/s	3.522 TeraFLOP/s
Double Precision FLOP/s	1.174 TeraFLOP/s
Number of Multiprocessors	13
Multiprocessor Clock Rate	705.5 MHz
Concurrent Kernel	true
Max IPC	7
Threads per Warp	32
Global Memory Bandwidth	208 GB/s
Global Memory Size	4.687 GiB
Constant Memory Size	64 KiB
L2 Cache Size	1.25 MiB
Memcpy Engines	2
PCIe Generation	2
PCIe Link Rate	5 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 "DHashData" is most likely limited by instruction and memory latency. You should first examine the information in the "Instruction And Memory Latency" section to determine how it is limiting performance.

1.1. Kernel Performance Is Bound By Instruction And Memory Latency

This kernel exhibits low compute throughput and memory bandwidth utilization relative to the peak performance of "Tesla K20c". These utilization levels indicate that the performance of the kernel is most likely limited by the latency of arithmetic or memory operations. Achieved compute throughput and/or memory bandwidth below 60% of peak typically indicates latency issues.



2. 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 increasing the number of threads in each block.

2.1. GPU Utilization Is Limited By Block Size

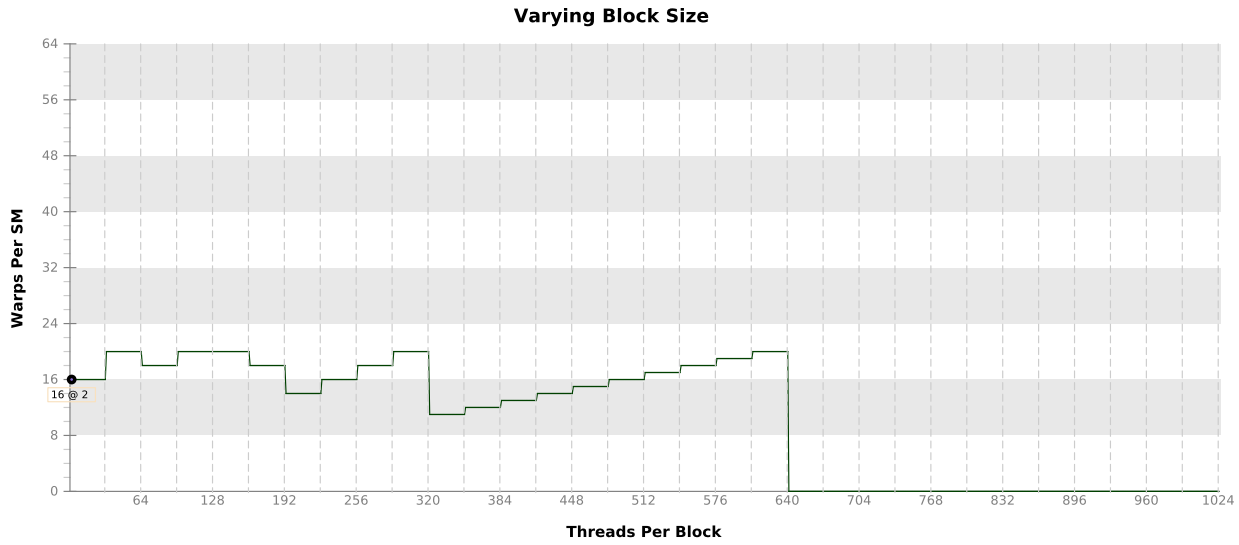
The kernel has a block size of 2 threads. This block size is likely preventing the kernel from fully utilizing the GPU. Device "Tesla K20c" can simultaneously execute up to 16 blocks on each SM. Because each block uses 1 warp to execute the block's 2 threads, the kernel is using only 16 warps on each SM. Chart "Varying Block Size" below shows how changing the block size will change the number of warps that can execute on each SM.

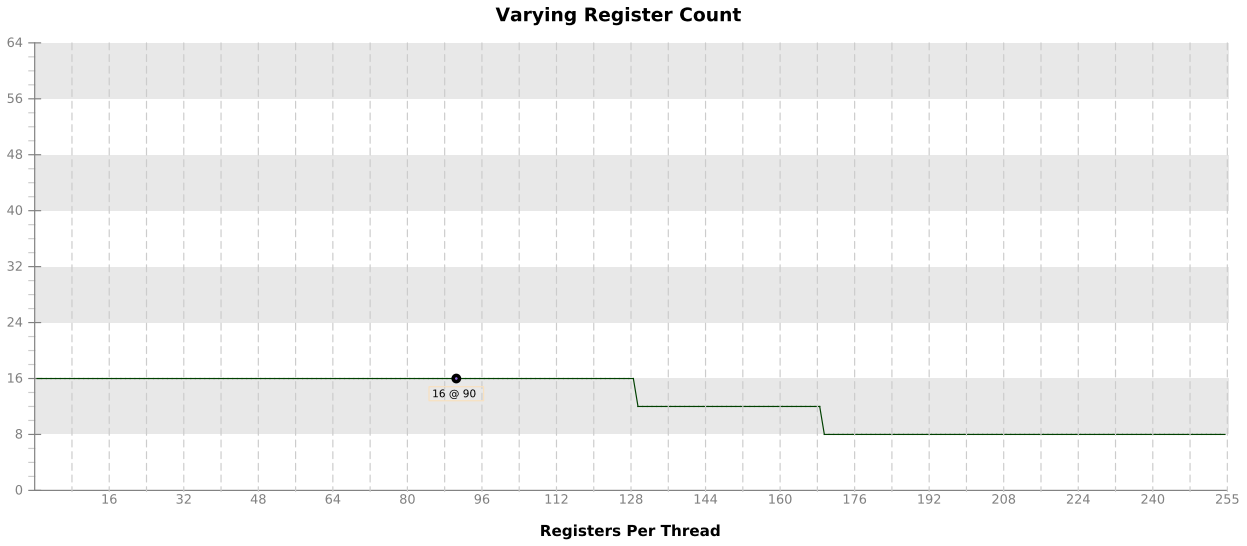
Optimization: Increase the number of threads in each block to increase the number of warps that can execute on each SM.

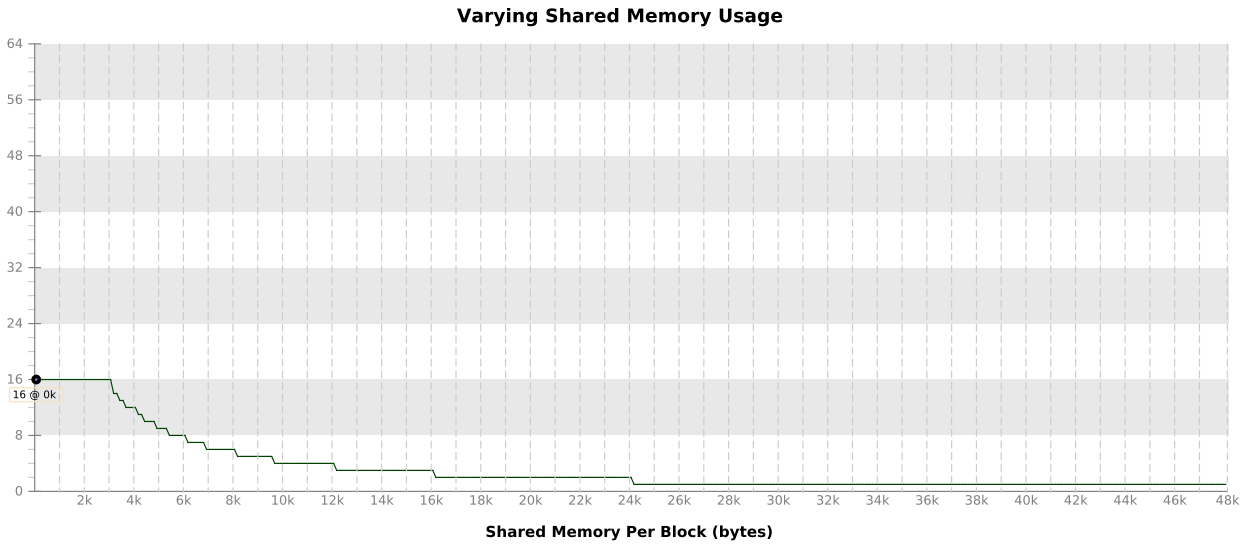
Variable	Achieved	Theoretical	Device Limit	Grid Size: [1024,1,1] (1024 blocks) Block Size: [2,1,1] (2 threads
Occupancy Per SM				
Active Blocks		16	16	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Active Warps	13.36	16	64	0 7 14 21 28 35 42 49 56 64
Active Threads		512	2048	0 256 512 768 1024 1280 1536 1792 2048
Occupancy	20.9%	25%	100%	0% 25% 50% 75% 100%
Warps				
Threads/Block		2	1024	0 128 256 384 512 640 768 896 1024
Warps/Block		1	32	0 3 6 9 12 15 18 21 24 27 30 32
Block Limit		64	16	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Registers				
Registers/Thread		90	255	0 32 64 96 128 160 192 224 255
Registers/Block		3072	65536	0 16k 32k 48k 64k
Block Limit		20	16	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Shared Memory				
Shared Memory/Block		0	49152	0 16k 32k 48k
Block Limit			16	

2.2. 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
16 @ 2
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
16 @ 90
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 @ 0k
2k
4k
6k
8k
10k
12k
14k
16k
18k
20k
22k
24k
26k
28k
30k
32k
34k
36k
38k
40k
42k
44k
46k
48k
Shared Memory Per Block (bytes)

3. 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.

3.1. Low Warp Execution Efficiency

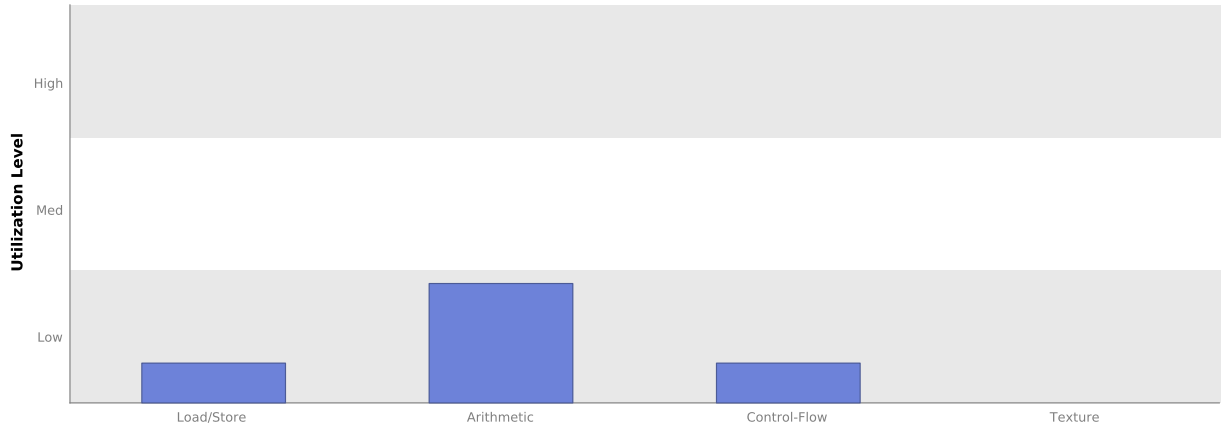
Warp execution efficiency is the average percentage of active threads in each executed warp. Increasing warp execution efficiency will increase utilization of the GPU's compute resources. The kernel's maximum warp execution efficiency is 6.2% because the number of threads per block is not a multiple of the warp size.

Optimization: Reduce the amount of intra-warp divergence and predication in the kernel.

3.2. Function Unit Utilization

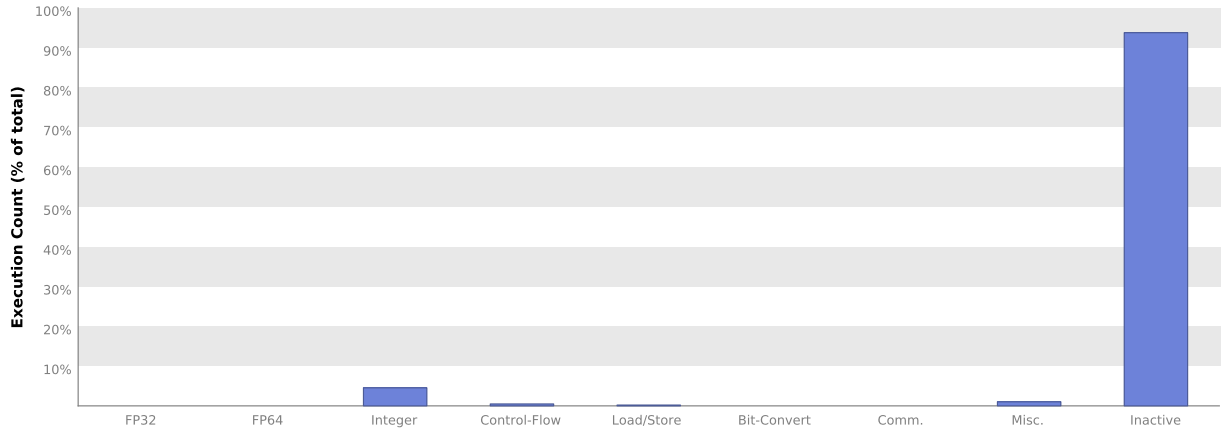
Different types of instructions are executed on different function units within each SM. Performance can be limited if a function unit is over-used by the instructions executed by the kernel. The following results show that the kernel's performance is not limited by overuse of any function unit.

Load/Store - Load and store instructions for local, shared, global, constant, etc. memory.
Arithmetic - All arithmetic instructions including integer and floating-point add and multiply, logical and binary operations, etc.
Control-Flow - Direct and indirect branches, jumps, and calls.
Texture - Texture operations.



3.3. 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.



3.4. 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.

4. 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.

4.1. Memory Bandwidth And Utilization

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.

Transactions	Bandwidth	Utilization	
L1/Shared Memory			
Local Loads	44466	15.278 GB/s	
Local Stores	36864	8.568 GB/s	
Shared Loads	0	0 B/s	
Shared Stores	0	0 B/s	
Global Loads	30208	3.13 GB/s	
Global Stores	20480	2.122 GB/s	
Atomic	0	0 B/s	
L1/Shared Total	132018	29.099 GB/s	Idle Low Medium High Max
L2 Cache			
L1 Reads	87527	9.069 GB/s	
L1 Writes	57261	5.933 GB/s	
Texture Reads	0	0 B/s	
Noncoherent Reads	0	0 B/s	
Atomic	0	0 B/s	
Total	144788	15.002 GB/s	Idle Low Medium High Max
Texture Cache			
Reads	0	0 B/s	Idle Low Medium High Max
Device Memory			
Reads	27731	2.873 GB/s	
Writes	39362	4.078 GB/s	
Total	67093	6.952 GB/s	Idle Low Medium High Max
ECC Overhead	26654	2.762 GB/s	
System Memory			
[PCIe configuration: Gen2 x8, 5 Gbit/s]			
Reads	0	0 B/s	Idle Low Medium High Max
Writes	10	1.036 MB/s	Idle Low Medium High Max