Analysis Report

compute_util_table_ver_1(int*, int**, unsigned int, unsigned int, int, int, int, int, int, int, int, bool)

Duration	448.085 ms (448,085,196 ns)
Grid Size	[762940,1,1]
Block Size	[64,1,1]
Registers/Thread	32
Shared Memory/Block	6.75 KiB
Shared Memory Requested	48 KiB
Shared Memory Executed	48 KiB
Shared Memory Bank Size	4 B

| [0] GeForce GTX TITAN	
GPU UUID	GPU-c2e0ee17-52b9-35a2-645d-ea8f92144945
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	4.707 TeraFLOP/s
Double Precision FLOP/s	196.112 GigaFLOP/s
Number of Multiprocessors	14
Multiprocessor Clock Rate	875.5 MHz
Concurrent Kernel	true
Max IPC	7
Threads per Warp	32
Global Memory Bandwidth	288.384 GB/s
Global Memory Size	5.999 GiB
Constant Memory Size	64 KiB
L2 Cache Size	1.5 MiB
Memcpy Engines	1
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 "compute_util_table_ver_1" 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 "GeForce GTX TITAN". 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.
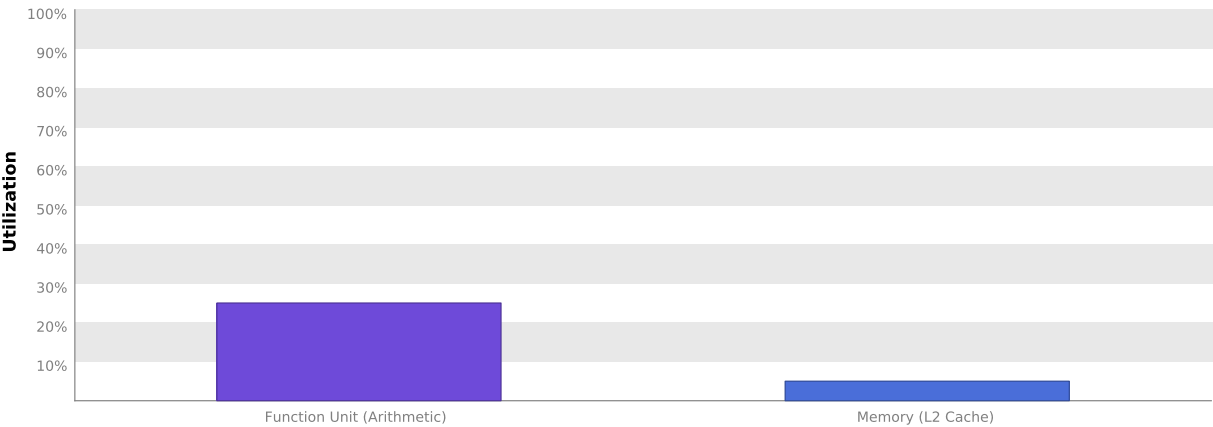
2. Instruction and Memory Latency

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 reducing the amount of shared memory used by the kernel.

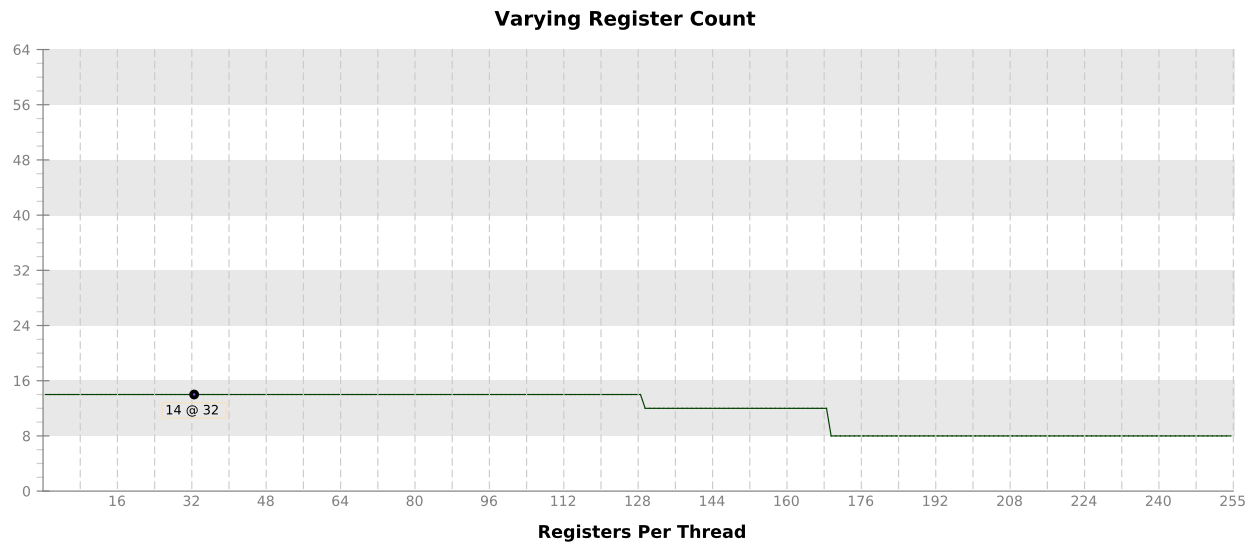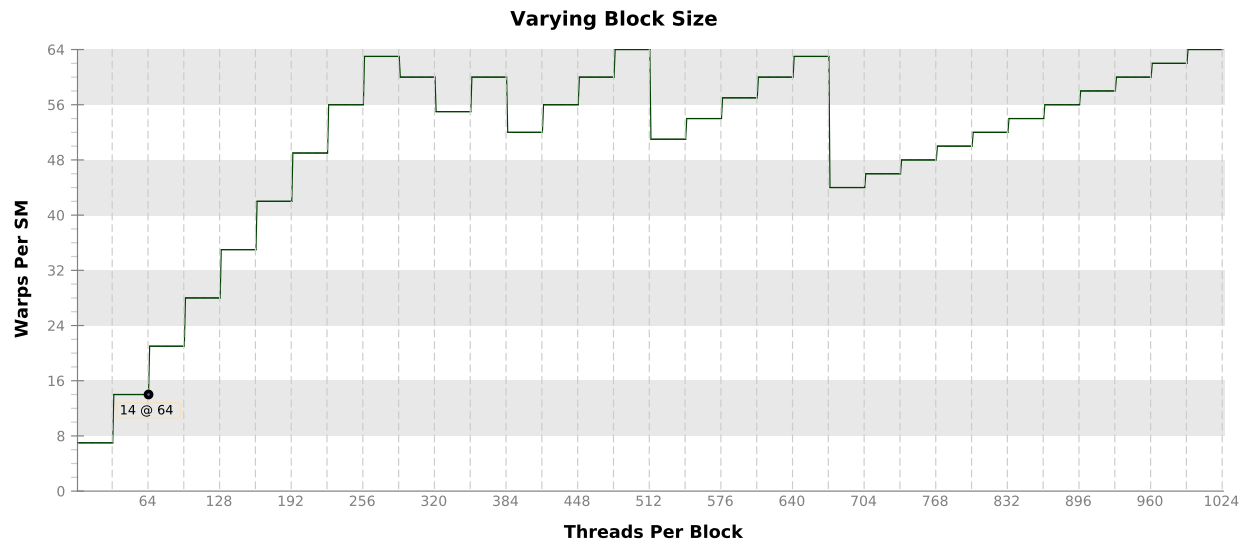## 2.1. GPU Utilization Is Limited By Shared Memory Usage

The kernel uses 6.75 KiB of shared memory for each block. This shared memory usage is likely preventing the kernel from fully utilizing the GPU. Device "GeForce GTX TITAN" is configured to have 48 KiB of shared memory for each SM. Because the kernel uses 6.75 KiB of shared memory for each block each SM is limited to simultaneously executing 7 blocks (14 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.
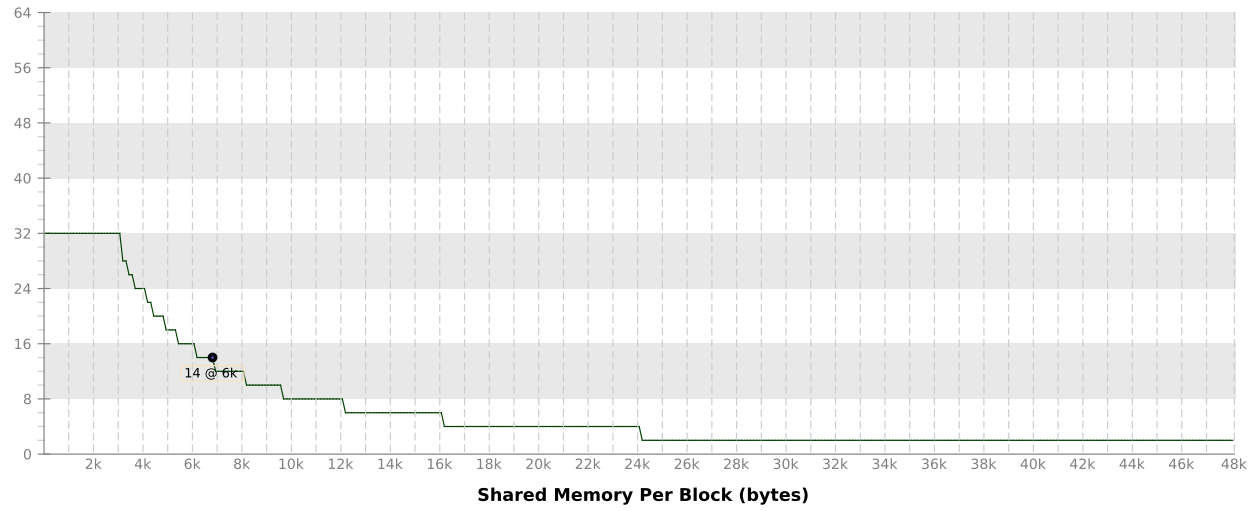
Variable	Achieved	Theoretical	Device Limit	Grid Size: [762940,1,1] (762940 blocks) Block Size: [6
Occupancy Per SM				
Active Blocks		7	16	
Active Warps	13.95	14	64	
Active Threads		448	2048	
Occupancy	21.8%	21.9%	100%	
Warps				
Threads/Block		64	1024	
Warps/Block		2	32	
Block Limit		32	16	
Registers				
Registers/Thread		32	255	
Registers/Block		2048	65536	
Block Limit		32	16	
Shared Memory				
Shared Memory/Block		6912	49152	
Block Limit		7	16	

2.2. Occupancy Charts

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

Varying Block Size

- Y-axis: Warps Per SM (0, 8, 16, 24, 32, 40, 48, 56, 64)
- X-axis: Threads Per Block (64, 128, 192, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960, 1024)
- Annotation: 14 @ 64

Varying Register Count

- Y-axis: (0, 8, 16, 24, 32, 40, 48, 56, 64)
- X-axis: Registers Per Thread (16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240, 255)
- Annotation: 14 @ 32

4

Varying Shared Memory Usage

14 @ 6k

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.

## 3.1. 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 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.
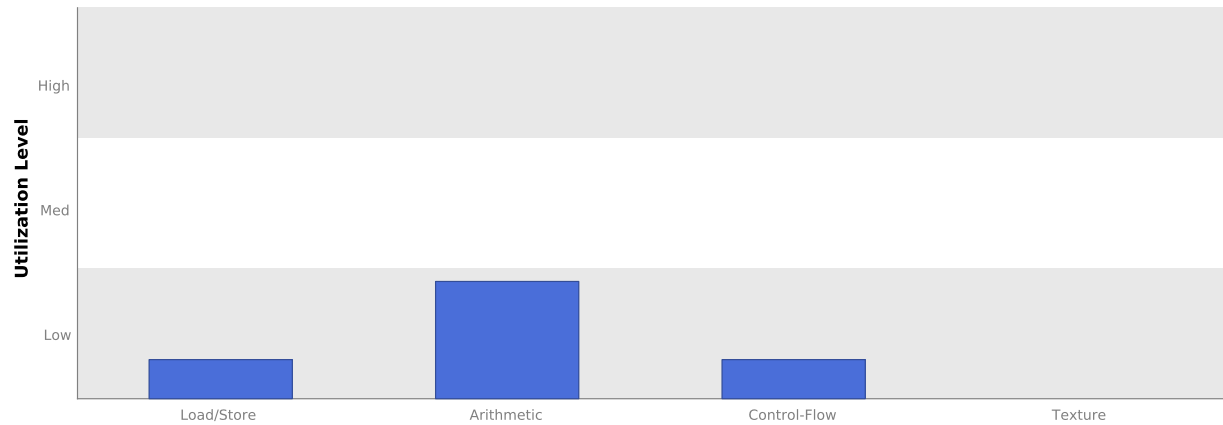
3.2. Instruction Execution Counts

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.
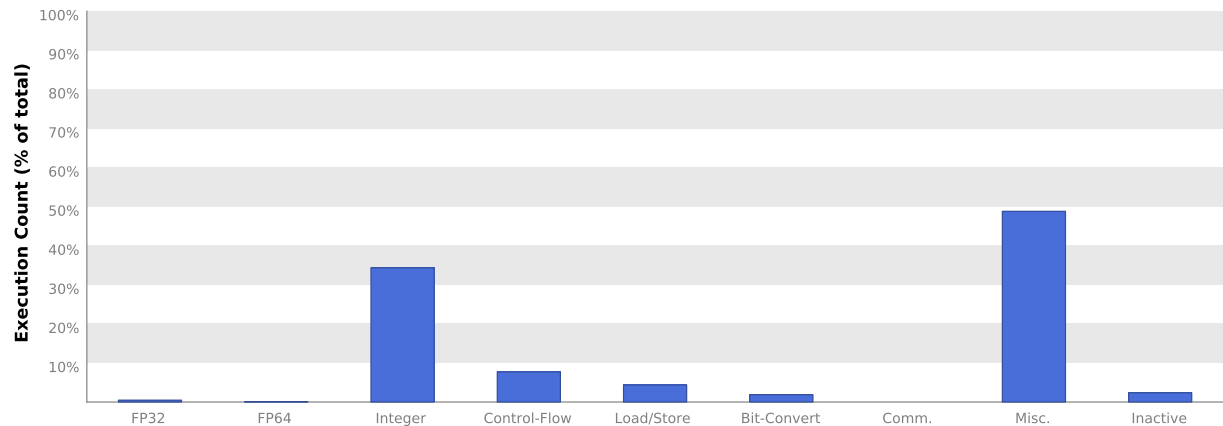
3.3. Floating-Point Operation Counts

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.

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	0	0 B/s	
Local Stores	0	0 B/s	
Shared Loads	108337409	61.999 GB/s	
Shared Stores	79345708	45.408 GB/s	
Global Loads	103859343	9.007 GB/s	
Global Stores	1525879	436.612 MB/s	
Atomic	0	0 B/s	
L1/Shared Total	293068339	116.85 GB/s	Idle Low Medium High Max
L2 Cache			
L1 Reads	125912005	9.007 GB/s	
L1 Writes	6103516	436.612 MB/s	
Texture Reads	0	0 B/s	
Noncoherent Reads	0	0 B/s	
Atomic	0	0 B/s	
Total	132015521	9.444 GB/s	Idle Low Medium High Max
Texture Cache			
Reads	0	0 B/s	Idle Low Medium High Max
Device Memory			
Reads	31937836	2.285 GB/s	
Writes	6103642	436.621 MB/s	
Total	38041478	2.721 GB/s	Idle Low Medium High Max
System Memory			
[PCIe configuration: Gen2 x8, 5 Gbit/s]			
Reads	0	0 B/s	Idle Low Medium High Max
Writes	6	429 B/s	Idle Low Medium High Max