

Analysis Report

```
void cudnn::detail::wgrad_alg0_engine<float, int=512, int=6, int=5, int=3, int=3,
int=3, bool=1, int=512>(int, int, int, float const *, int,
cudnn::detail::wgrad_alg0_engine<float, int=512, int=6, int=5, int=3, int=3, int=3,
bool=1, int=512>*, float const , kernel_grad_params, int, float, int, int)
```

Duration	304.19 μ s
Grid Size	[1,1,256]
Block Size	[8,8,1]
Registers/Thread	79
Shared Memory/Block	3.25 KiB
Shared Memory Requested	96 KiB
Shared Memory Executed	96 KiB
Shared Memory Bank Size	4 B

[0] GeForce GTX 1080

GPU UUID	GPU-edd19385-a5f1-ce46-e1d9-61408827cccc
Compute Capability	6.1
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	72.9 GigaFLOP/s
Single Precision FLOP/s	9.331 TeraFLOP/s
Double Precision FLOP/s	291.6 GigaFLOP/s
Number of Multiprocessors	20
Multiprocessor Clock Rate	1.823 GHz
Concurrent Kernel	true
Max IPC	6
Threads per Warp	32
Global Memory Bandwidth	320.32 GB/s
Global Memory Size	7.923 GiB
Constant Memory Size	64 KiB
L2 Cache Size	2 MiB
Memcpy Engines	2
PCIe Generation	3

[0] GeForce GTX 1080

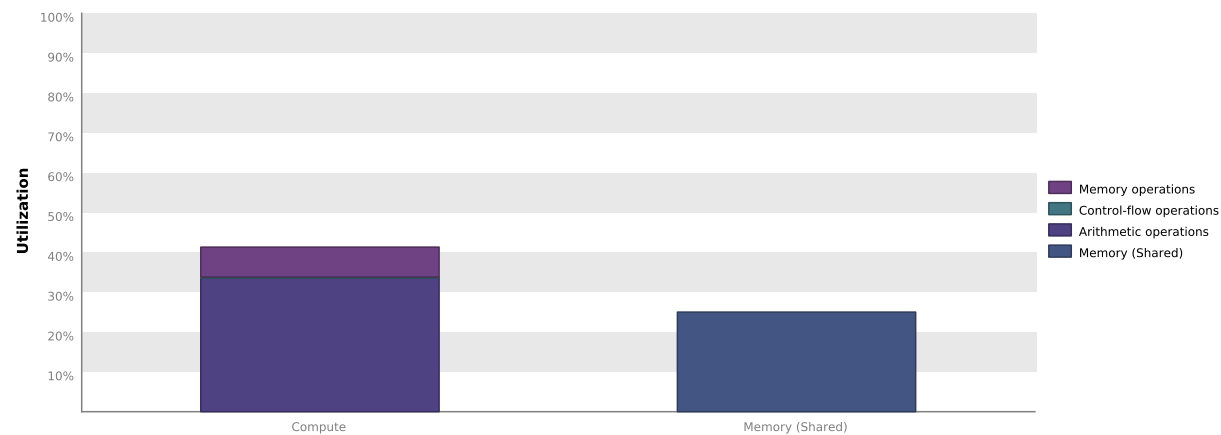
PCIe Link Rate	8 Gbit/s
PCIe Link Width	16

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 cudnn::detail::wgrad_a..." 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 1080". 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 number of registers used by the kernel.

2.1. GPU Utilization Is Limited By Register Usage

The kernel uses 79 registers for each thread (5056 registers for each block). This register usage is likely preventing the kernel from fully utilizing the GPU. Device "GeForce GTX 1080" provides up to 65536 registers for each block. Because the kernel uses 5056 registers for each block each SM is limited to simultaneously executing 12 blocks (24 warps). Chart "Varying Register Count" below shows how changing register usage will change the number of blocks that can execute on each SM.

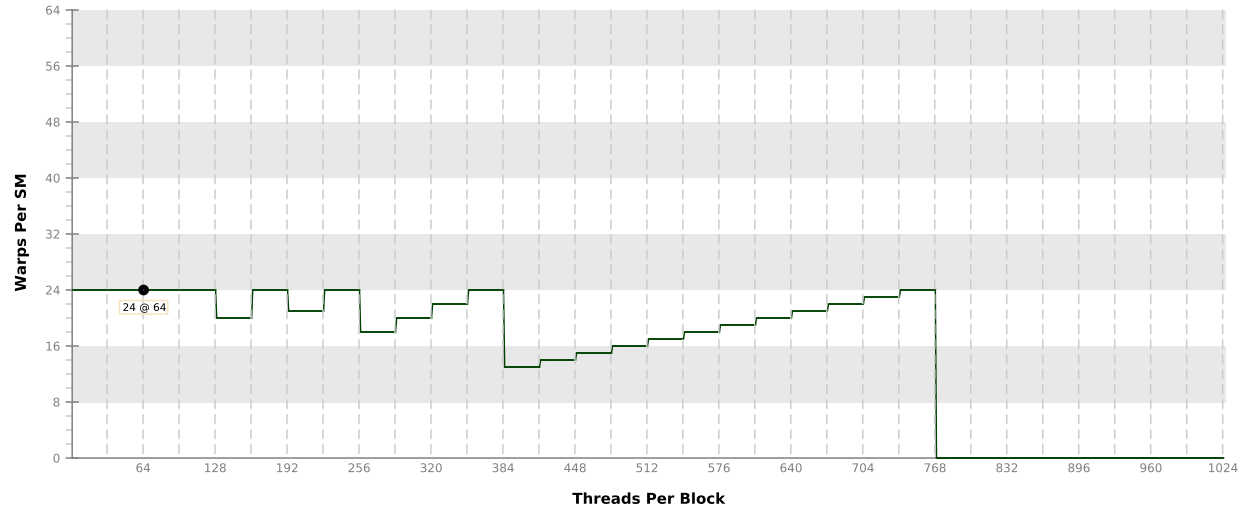
Optimization: Use the `-maxrregcount` flag or the `__launch_bounds__` qualifier to decrease the number of registers used by each thread. This will increase the number of blocks that can execute on each SM. On devices with Compute Capability 5.2 turning global cache off can increase the occupancy limited by register usage.

Variable	Achieved	Theoretical	Device Limit	Grid Size: [1,1,256] (256 blocks) Block Size: [8,8,1] (64 threads)
Occupancy Per SM				
Active Blocks		12	32	
Active Warps	20.91	24	64	
Active Threads		768	2048	
Occupancy	32.7%	37.5%	100%	
Warps				
Threads/Block		64	1024	
Warps/Block		2	32	
Block Limit		32	32	
Registers				
Registers/Thread		79	65536	
Registers/Block		5120	65536	
Block Limit		12	32	
Shared Memory				
Shared Memory/Block		3328	98304	
Block Limit		29	32	

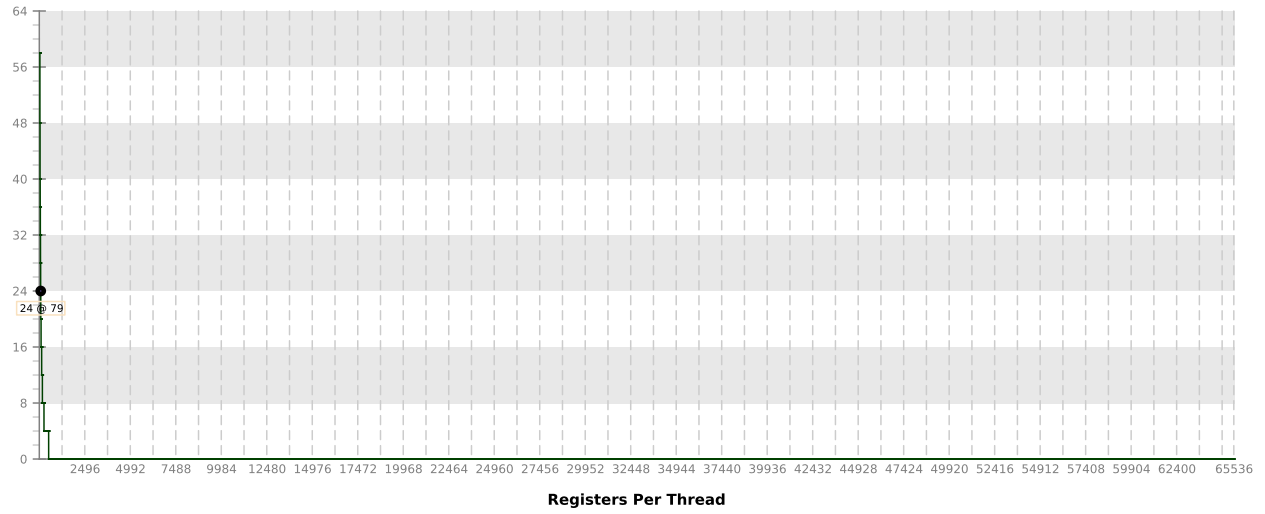
2.2. Occupancy Charts

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

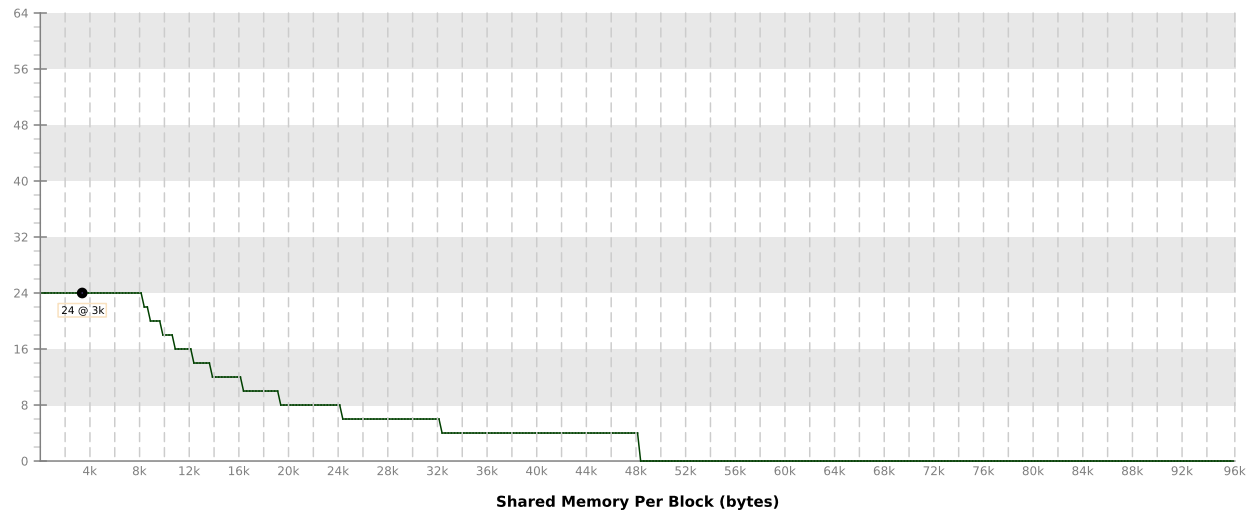
Varying Block Size



Varying Register Count



Varying Shared Memory Usage



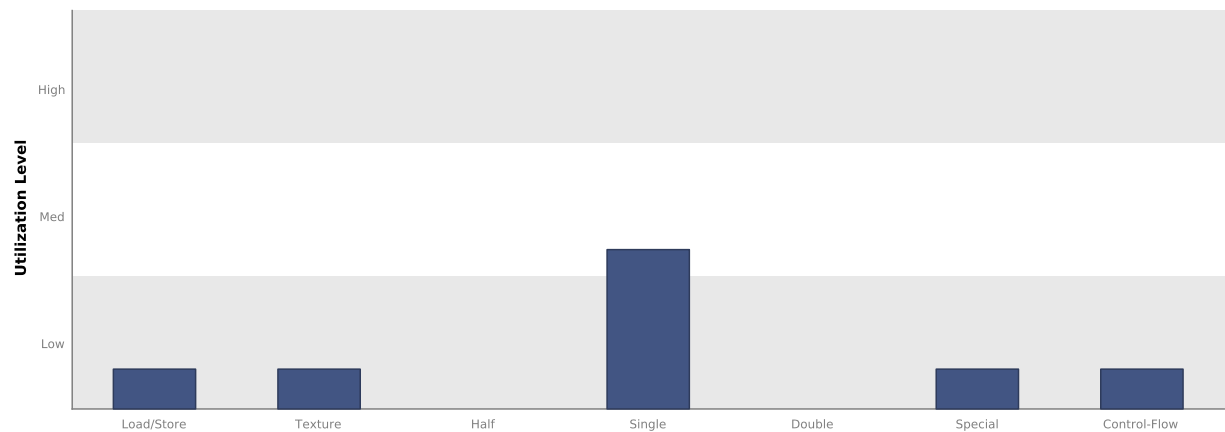
3. Compute Resources

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

3.1. Function Unit Utilization

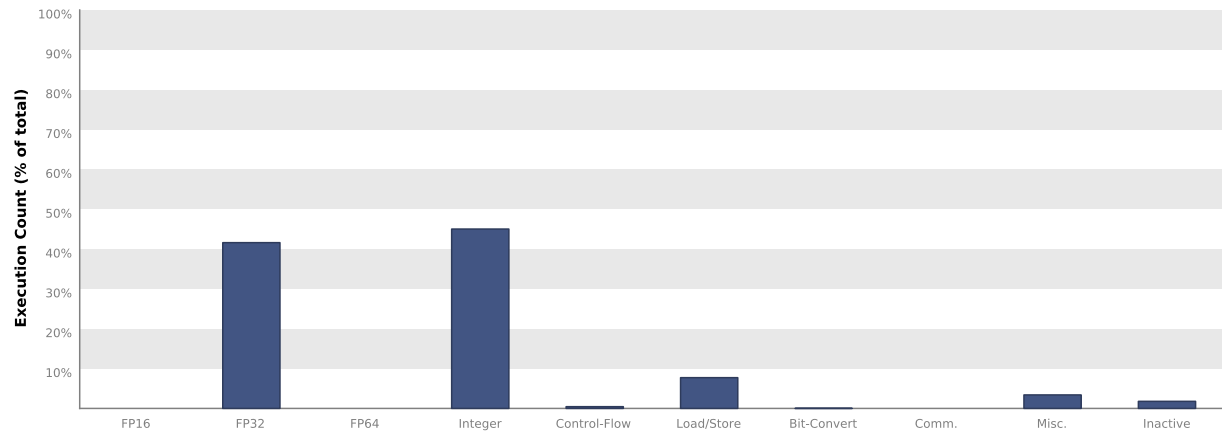
Different types of instructions are executed on different function units within each SM. Performance can be limited if a function unit is over-used by the instructions executed by the kernel. The following results show that the kernel's performance is not limited by overuse of any function unit.

- Load/Store - Load and store instructions for 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.



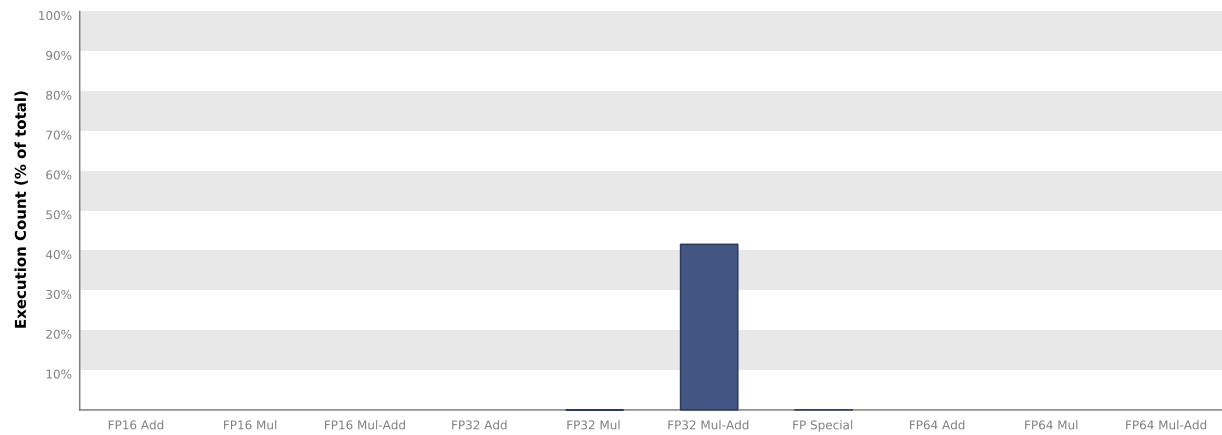
3.2. Instruction Execution Counts

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	
Shared Memory			
Shared Loads	2007040	861.422 GB/s	
Shared Stores	301056	129.213 GB/s	
Shared Total	2308096	990.636 GB/s	
L2 Cache			
Reads	674556	72.38 GB/s	
Writes	135181	14.505 GB/s	
Total	809737	86.885 GB/s	
Unified Cache			
Local Loads	0	0 B/s	
Local Stores	0	0 B/s	
Global Loads	2741632	294.177 GB/s	
Global Stores	0	0 B/s	
Texture Reads	1605376	172.257 GB/s	
Unified Total	4347008	466.434 GB/s	
Device Memory			
Reads	417708	44.82 GB/s	
Writes	26456	2.839 GB/s	
Total	444164	47.659 GB/s	
System Memory			
[PCIe configuration: Gen3 x16, 8 Gbit/s]			
Reads	0	0 B/s	
Writes	5	536.5 kB/s	