# Analysis Report

## mxnet::op::unroll_kernel(float*, float*, int, int, int, int)

| | |
|---|---|
| Duration | 8.008 µs |
| Grid Size | [ 5,1,1 ] |
| Block Size | [ 1024,1,1 ] |
| Registers/Thread | 32 |
| Shared  Memory/Block | 0 B |
| Shared Memory Executed | 0 B |
| Shared Memory Bank Size | 4 B |

[0] TITAN V

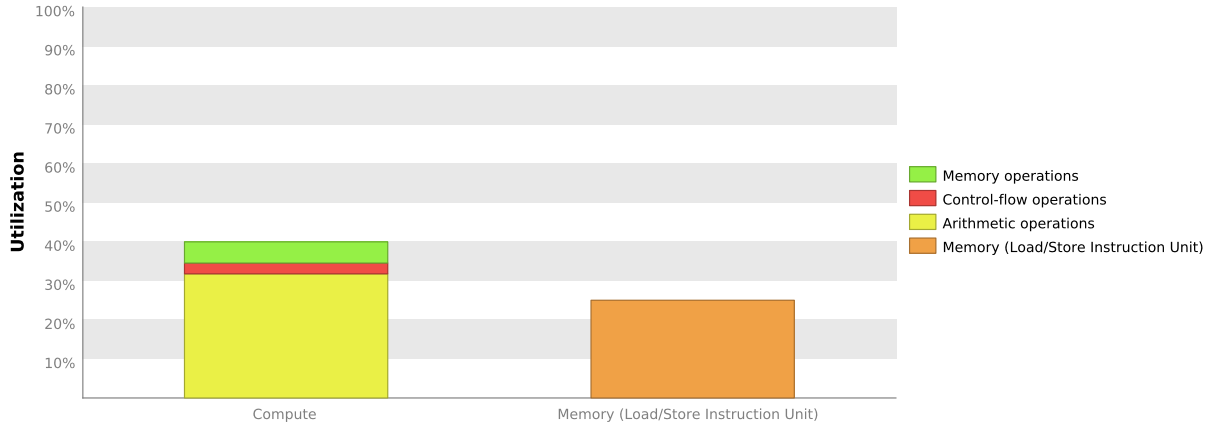
| | |
|---|---|
| GPU UUID | GPU-e3ce02be-cdd2-1d0b-a44d-7eda09667ab3 |
| Compute Capability | 7.0 |
| Max. Threads per Block | 1024 |
| Max. Threads per Multiprocessor | 2048 |
| Max. Shared Memory per Block | 48 KiB |
| Max. Shared Memory per Multiprocessor | 96 KiB |
| Max. Registers per Block | 65536 |
| 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 | 11.752 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 | 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 "mxnet::op::unroll_kernel" 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 "TITAN V". 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 results below indicate that the GPU does not have enough work because the kernel does not execute enough blocks.

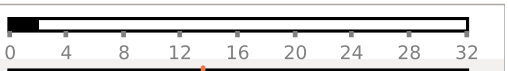
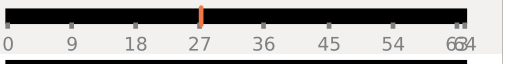
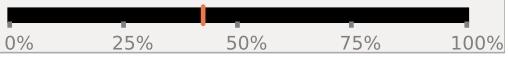
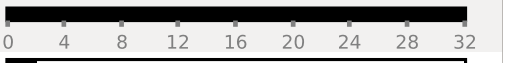
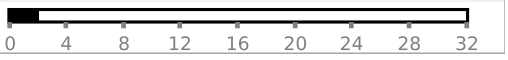
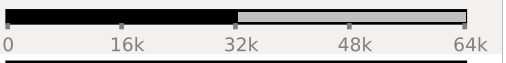
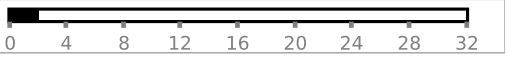
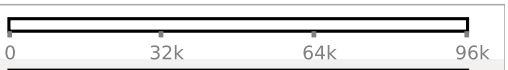
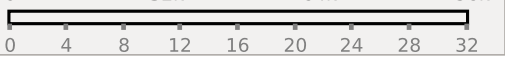
### 2.1. Grid Size Too Small To Hide Compute And Memory Latency

The kernel does not execute enough blocks to hide memory and operation latency. Typically the kernel grid size must be large enough to fill the GPU with multiple "waves" of blocks. Based on theoretical occupancy, device "TITAN V" can simultaneously execute 2 blocks on each of the 80 SMs, so the kernel may need to execute a multiple of 160 blocks to hide the compute and memory latency. If the kernel is executing concurrently with other kernels then fewer blocks will be required because the kernel is sharing the SMs with those kernels.

*Optimization: Increase the number of blocks executed by the kernel.*

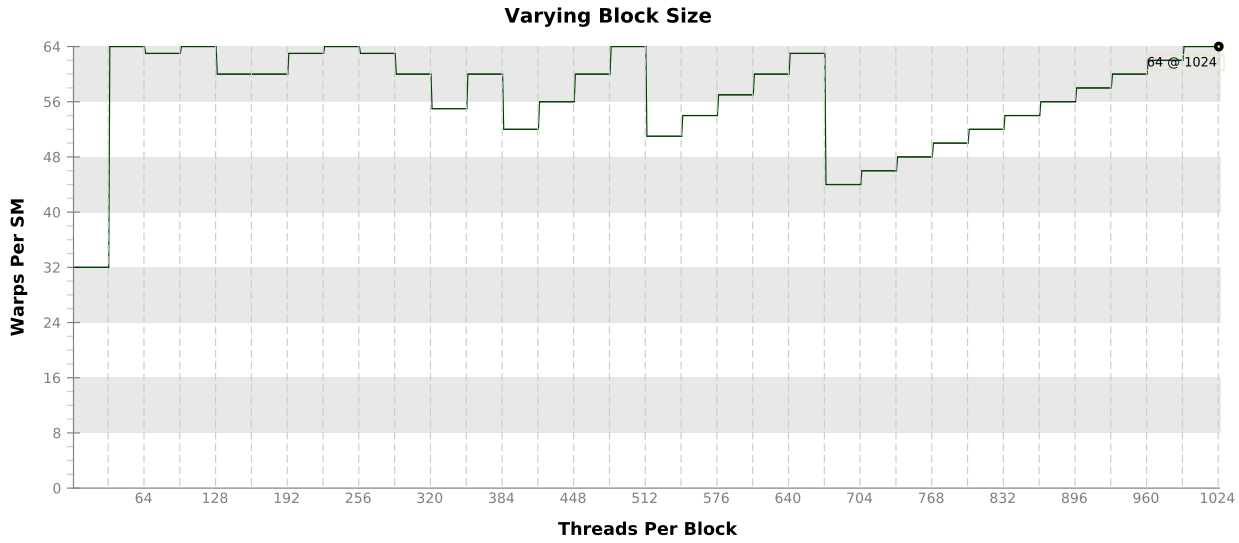
### 2.2. Occupancy Is Not Limiting Kernel Performance

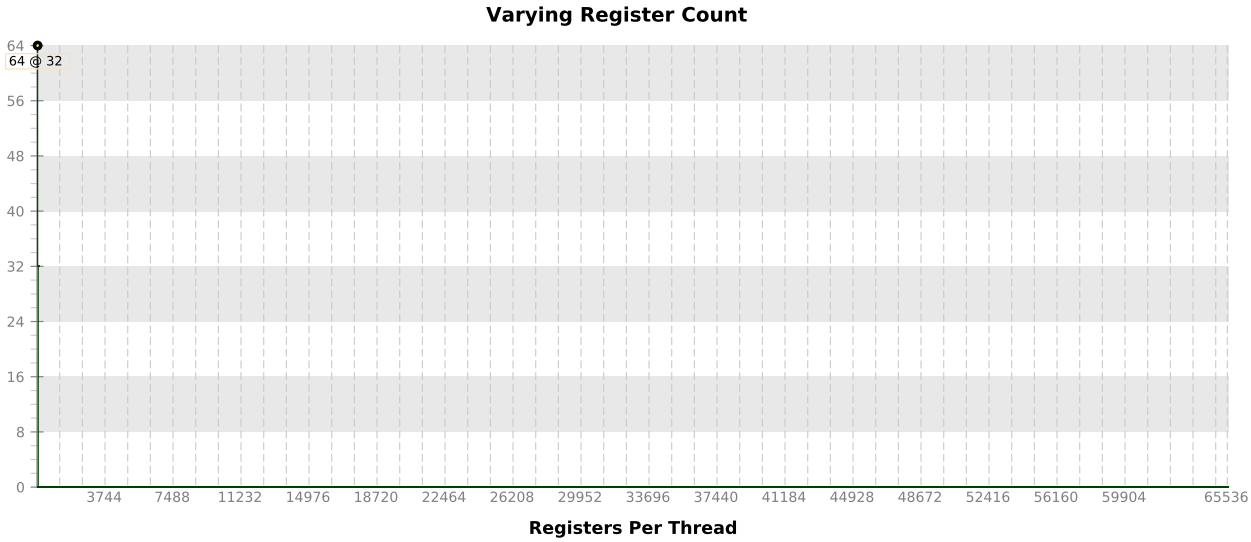
The kernel's block size, register usage, and shared memory usage allow it to fully utilize all warps on the GPU.

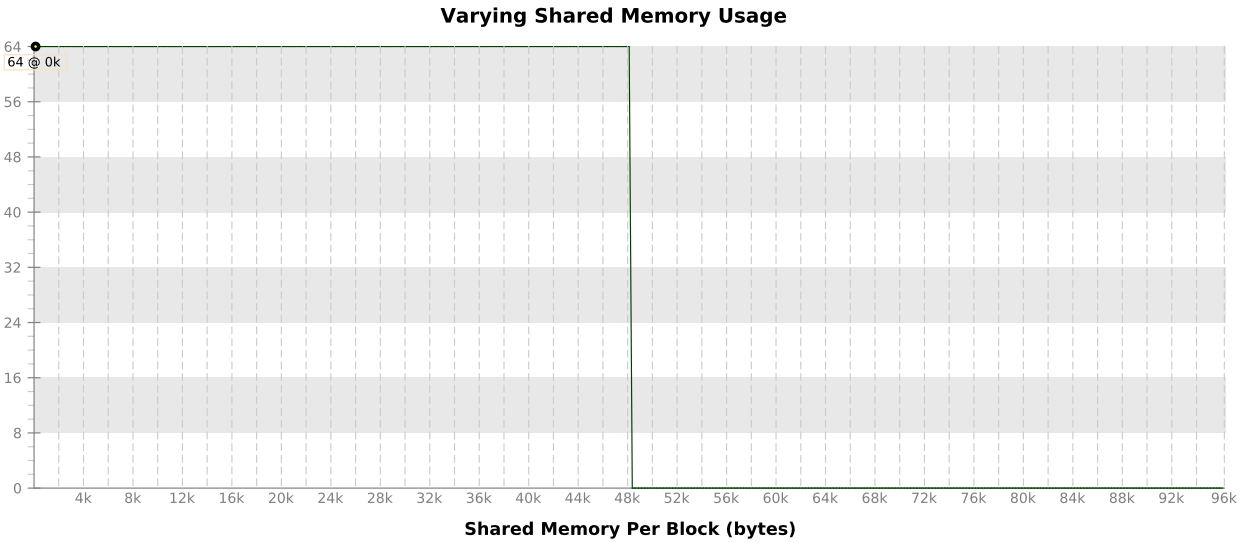
| Variable | Achieved | Theoretical | Device Limit | Grid Size: [ 5,1,1 ] (5 blocks) Block Size: [ 1024,1,1 ] (10 |
|---|---|---|---|---|
| Occupancy Per SM | | | | |
| Active Blocks | | 2 | 32 |  |
| Active Warps | 26.65 | 64 | 64 |  |
| Active Threads | | 2048 | 2048 |  |
| Occupancy | 41.6% | 100% | 100% |  |
| Warps | | | | |
| Threads/Block | | 1024 | 1024 |  |
| Warps/Block | | 32 | 32 |  |
| Block Limit | | 2 | 32 |  |
| Registers | | | | |
| Registers/Thread | | 32 | 65536 |  |
| Registers/Block | | 32768 | 65536 |  |
| Block Limit | | 2 | 32 |  |
| Shared Memory | | | | |
| Shared Memory/Block | | 0 | 98304 |  |
| Block Limit | | 0 | 32 |  |

### 2.3. Occupancy Charts

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

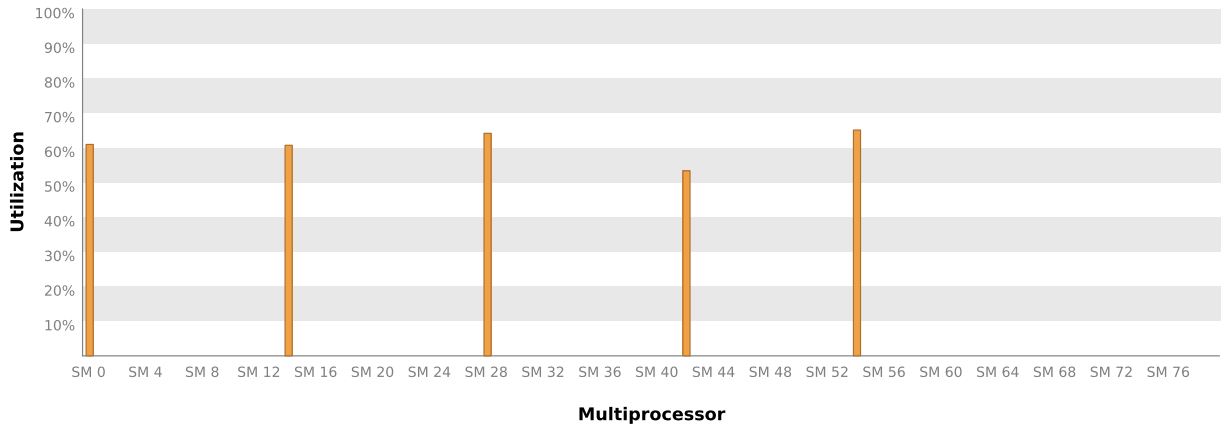

Varying Block Size
Warps Per SM
64
56
48
40
32
24
16
8
0
64 @ 1024
64
128
192
256
320
384
448
512
576
640
704
768
832
896
960
1024
Threads Per Block


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




## 2.4. 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.



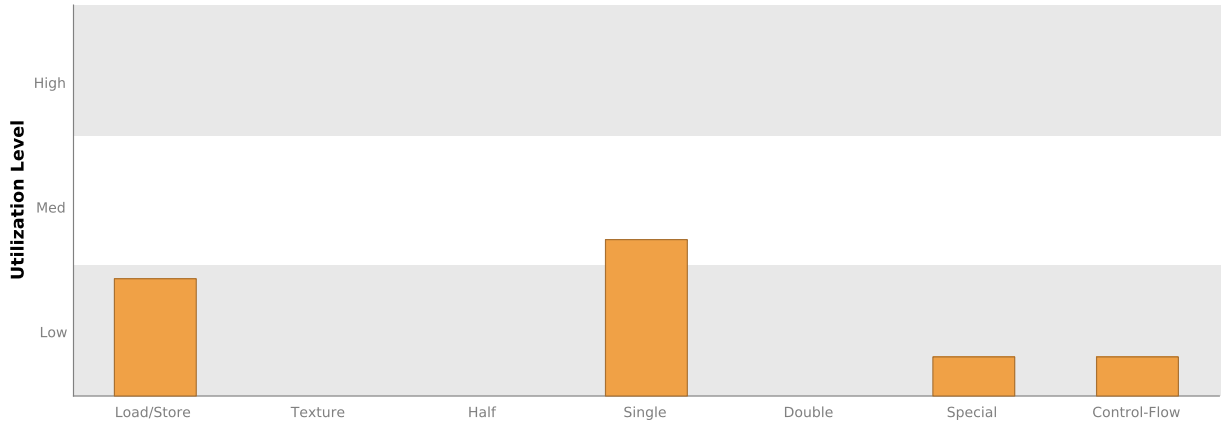
# 3. Compute Resources

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

## 3.1. Function Unit Utilization

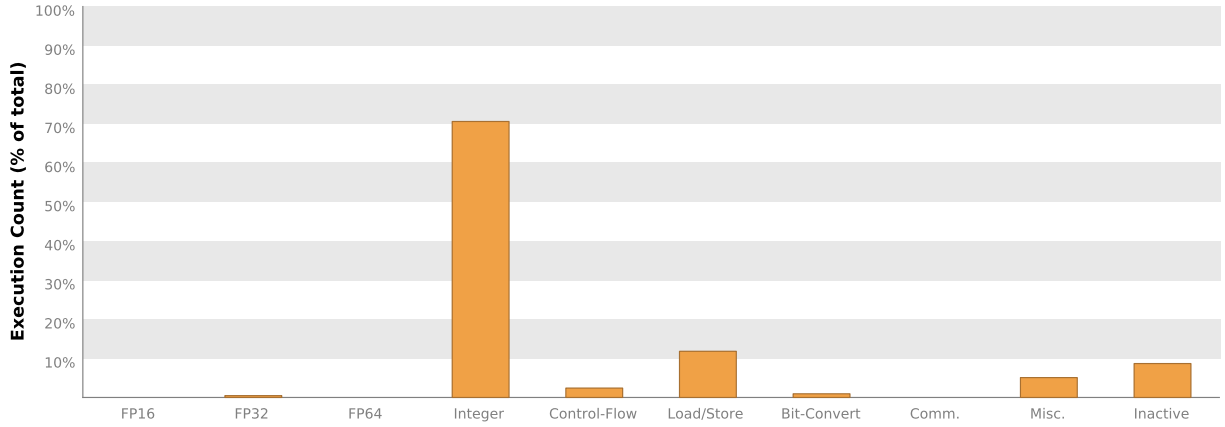
Different types of instructions are executed on different function units within each SM. Performance can be limited if a function unit is over-used by the instructions executed by the kernel. The following results show that the kernel's performance is not limited by overuse of any function unit.

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

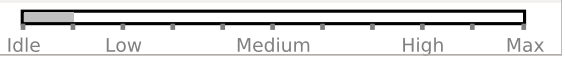
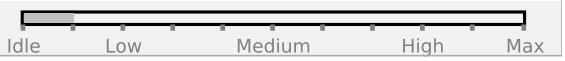
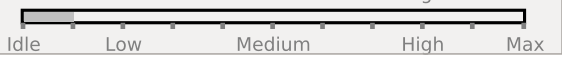
The following chart shows the mix of floating-point operations executed by the kernel. The operations are grouped into classes and for each class the chart shows the percentage of thread execution cycles that were devoted to executing operations in that class. The results do not sum to 100% because non-floating-point 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 | 0 | 0 B/s | |
| Shared Stores | 0 | 0 B/s | |
| Shared Total | 0 | 0 B/s |   |
| L2 Cache | | | |
| Reads | 837 | 3.345 GB/s | |
| Writes | 15273 | 61.031 GB/s | |
| Total | 16110 | 64.376 GB/s |   |
| Unified Cache | | | |
| Local Loads | 0 | 0 B/s | |
| Local Stores | 0 | 0 B/s | |
| Global Loads | 17147 | 68.519 GB/s | |
| Global Stores | 15257 | 60.967 GB/s | |
| Texture Reads | 3864 | 61.762 GB/s | |
| Unified Total | 36268 | 191.249 GB/s |   |
| Device Memory | | | |
| Reads | 620 | 2.478 GB/s | |
| Writes | 12326 | 49.255 GB/s | |
| Total | 12946 | 51.732 GB/s |   |
| System Memory | | | |
| [ PCIe configuration: Gen3 x16, 8 Gbit/s ] | | | |
| Reads | 0 | 0 B/s |   |
| Writes | 5 | 19.98 MB/s |   |

## 4.2. 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.