# Analysis Report

## run_lbm(double[27]*, double[27]*, unsigned long, double, IndexBlock)

| | |
|---|---|
| Duration | 183.365 μs |
| Grid Size | [ 11,1,1 ] |
| Block Size | [ 11,11,1 ] |
| Registers/Thread | 60 |
| Shared  Memory/Block | 0 B |
| Shared Memory Requested | 48 KiB |
| Shared Memory Executed | 48 KiB |
| Shared Memory Bank Size | 4 B |

[0] GeForce GTX 765M

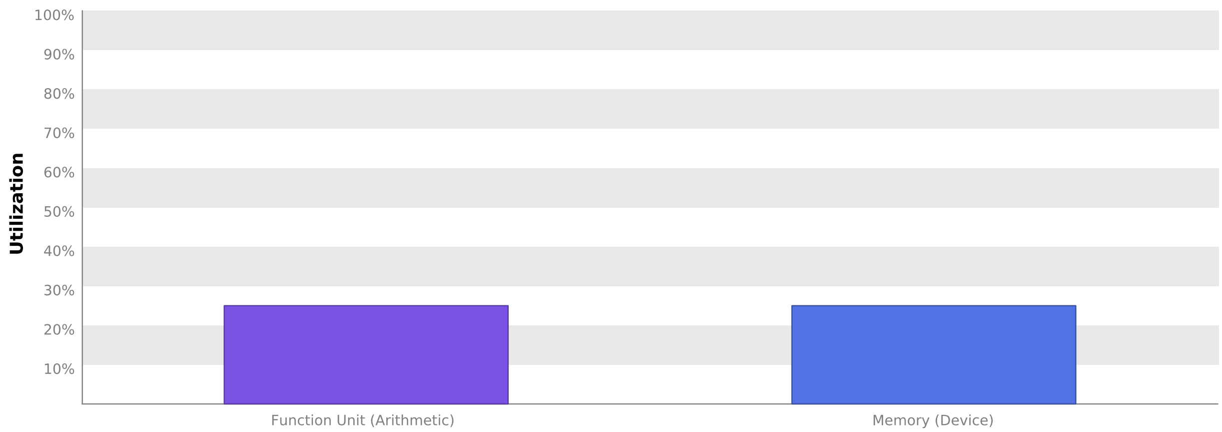
| | |
|---|---|
| GPU UUID | GPU-fe6c21ea-b7dc-9c6c-991c-33e89452d625 |
| Compute Capability | 3.0 |
| Max. Threads per Block | 1024 |
| Max. Threads per Multiprocessor | 2048 |
| Max. Shared Memory per Block | 48 KiB |
| Max. Shared Memory per Multiprocessor | 48 KiB |
| Max. Registers per Block | 65536 |
| 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 | 16 |
| Single Precision FLOP/s | 1.325 TeraFLOP/s |
| Double Precision FLOP/s | 55.2 GigaFLOP/s |
| Number of Multiprocessors | 4 |
| Multiprocessor Clock Rate | 862.5 MHz |
| Concurrent Kernel | true |
| Max IPC | 7 |
| Threads per Warp | 32 |
| Global Memory Bandwidth | 64.128 GB/s |
| Global Memory Size | 1.952 GiB |
| Constant Memory Size | 64 KiB |
| L2 Cache Size | 256 KiB |
| Memcpy Engines | 1 |
| 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 "run_lbm" 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 765M". 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 "GeForce GTX 765M" can simultaneously execute 8 blocks on each of the 4 SMs, so the kernel may need to execute a multiple of 32 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. GPU Utilization May Be Limited By Register Usage

Theoretical occupancy is less than 100% but is large enough that increasing occupancy may not improve performance. You can attempt the following optimization to increase the number of warps on each SM but it may not lead to increased performance.

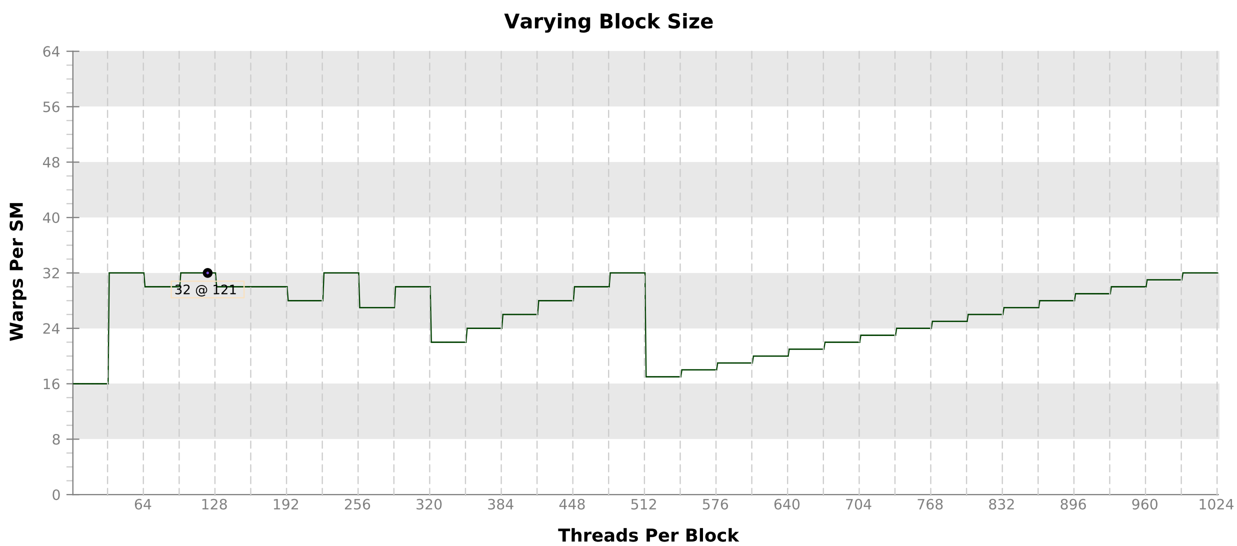
The kernel uses 60 registers for each thread (7260 registers for each block). This register usage is likely preventing the kernel from fully utilizing the GPU. Device "GeForce GTX 765M" provides up to 65536 registers for each block. Because the kernel uses 7260 registers for each block each SM is limited to simultaneously executing 8 blocks (32 warps). Chart "Varying Register Count" below shows how changing register usage will change the number of blocks that can execute on each SM.

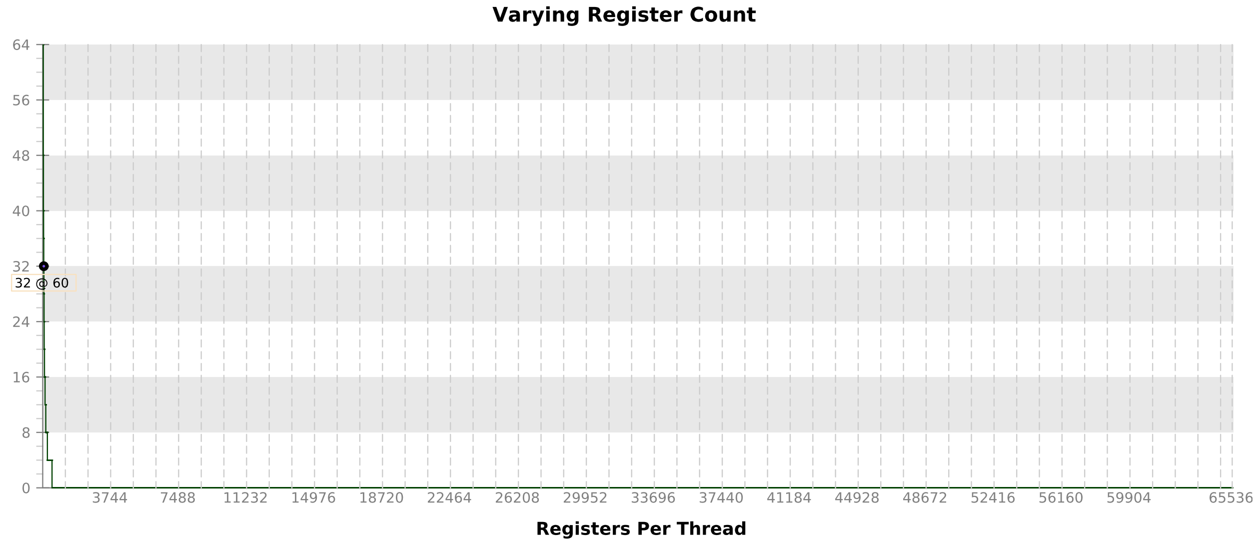
*Optimization: Use the -maxrregcount flag or the __launch_bounds__ qualifier to decrease the number of registers used by each thread. This will increase the number of blocks that can execute on each SM. On devices with Compute Capability 5.2 turning global cache off can increase the occupancy limited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

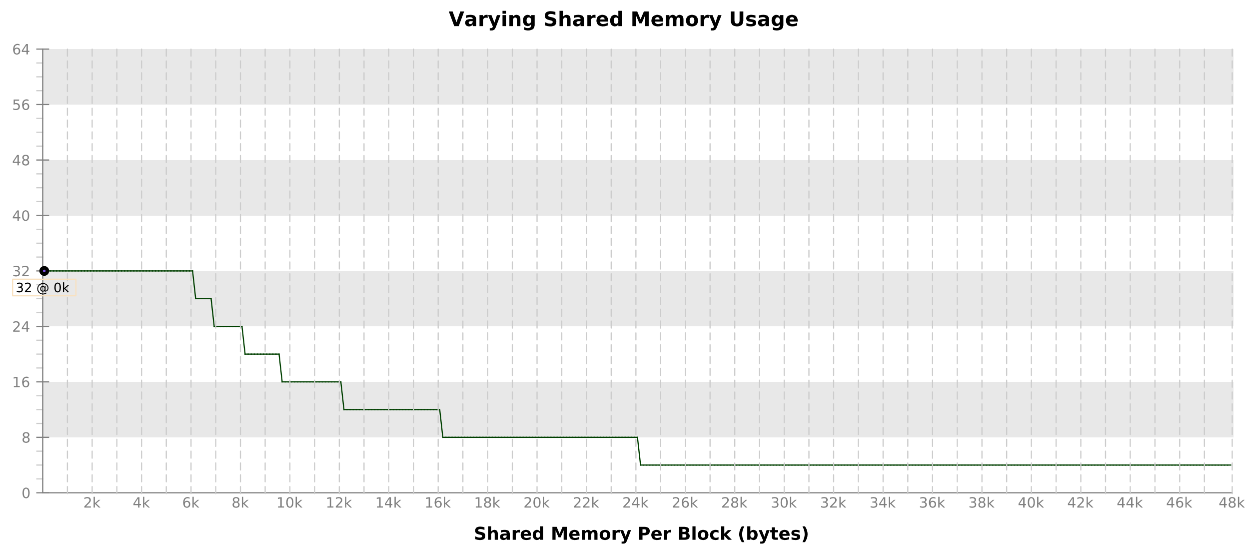
| Variable | Achieved | Theoretical | Device Limit | Grid Size: [ 11,1,1 ] (11 blocks) Block Size: [ 11,11,1 ] (121 threads |
|---|---|---|---|---|
| **Occupancy Per SM** | | | | |
| Active Blocks | | 8 | 16 | |
| Active Warps | 10.4 | 32 | 64 | |
| Active Threads | | 1024 | 2048 | |
| Occupancy | 16.2% | 50% | 100% | |
| **Warps** | | | | |
| Threads/Block | | 121 | 1024 | |
| Warps/Block | | 4 | 32 | |
| Block Limit | | 16 | 16 | |
| **Registers** | | | | |
| Registers/Thread | | 60 | 65536 | |
| Registers/Block | | 8192 | 65536 | |
| Block Limit | | 8 | 16 | |
| **Shared Memory** | | | | |
| Shared Memory/Block | | 0 | 49152 | |
| Block Limit | | 0 | 16 | |

## 2.3. 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
32 @ 60
3744
7488
11232
14976
18720
22464
26208
29952
33696
37440
41184
44928
48672
52416
56160
59904
65536
Registers Per Thread


Varying Shared Memory Usage
64
56
48
40
32
24
16
8
0
32 @ 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)

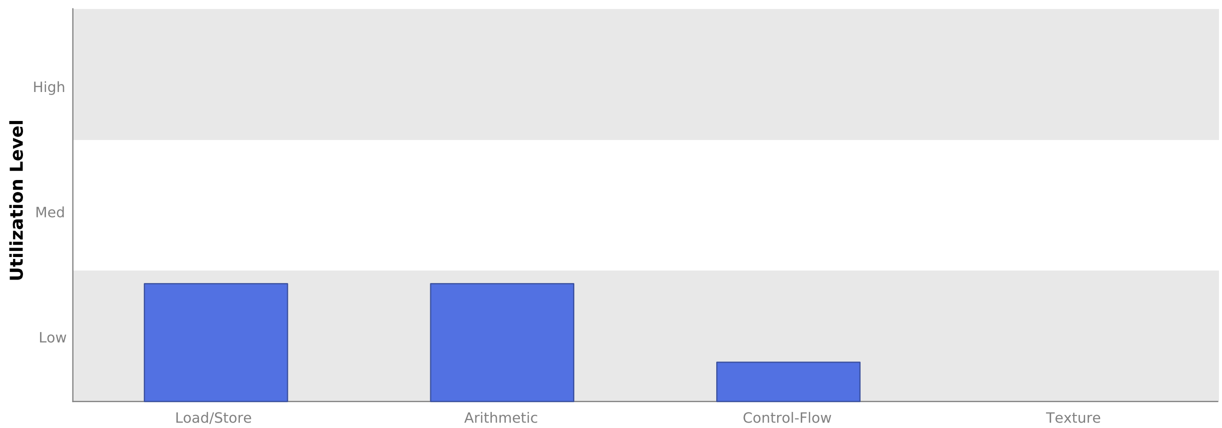
# 3. Compute Resources

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

## 3.1. Function Unit Utilization

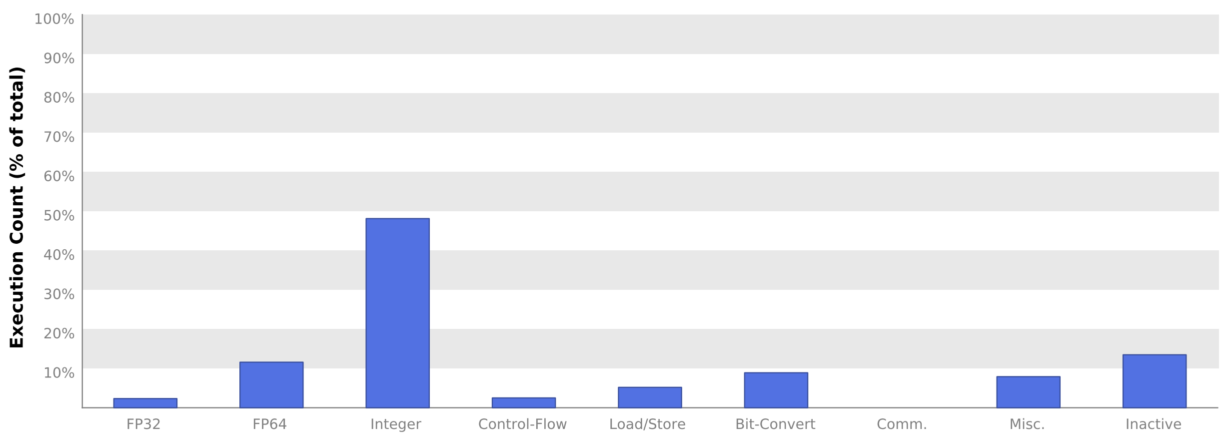
Different types of instructions are executed on different function units within each SM. Performance can be limited if a function unit is over-used by the instructions executed by the kernel. The following results show that the kernel's performance is not limited by overuse of any function unit.

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.



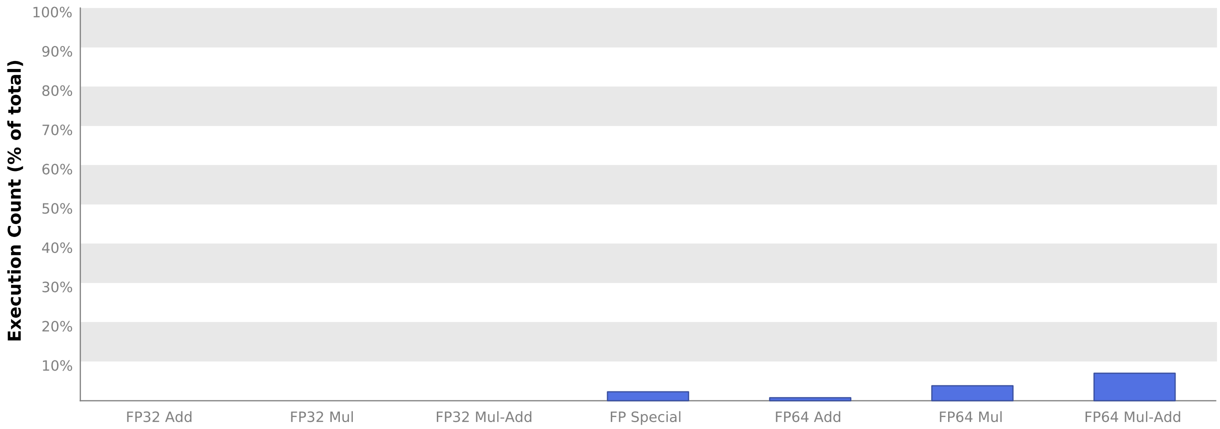

## 3.2. Instruction Execution Counts

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 | 0 | 0 B/s | |
| Shared Stores | 0 | 0 B/s | |
| Global Loads | 71874 | 13.512 GB/s | |
| Global Stores | 71874 | 13.512 GB/s | |
| Atomic | 0 | 0 B/s | |
| L1/Shared Total | 143748 | 27.025 GB/s | Idle Low Medium High Max |
| L2 Cache | | | |
| L1 Reads | 71874 | 13.512 GB/s | |
| L1 Writes | 71874 | 13.512 GB/s | |
| Texture Reads | 0 | 0 B/s | |
| Atomic | 0 | 0 B/s | |
| Total | 143748 | 27.025 GB/s | Idle Low Medium High Max |
| Texture Cache | | | |
| Reads | 0 | 0 B/s | Idle Low Medium High Max |
| Device Memory | | | |
| Reads | 40944 | 7.698 GB/s | |
| Writes | 66788 | 12.556 GB/s | |
| Total | 107732 | 20.254 GB/s | Idle Low Medium High Max |
| System Memory | | | |
| [ PCIe configuration: Gen2 x16, 5 Gbit/s ] | | | |
| Reads | 0 | 0 B/s | Idle Low Medium High Max |
| Writes | 2 | 376.003 kB/s | Idle Low Medium High Max |