Page: Details | Result: 0 - 517 - mandel_kernel | Add Baseline | Apply Rules | Occupancy Calculator | Save as PDF

	Report	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	re...e1	517 - ...	753.44 msecond	1,039,830,114	24	0 - NVIDIA GeForce I	1.38 cycle/nsecond	7.5	190621] benchmark
Baseline 1	rem...ile	517 - ...	30.94 msecond	42,699,894	24	0 - NVIDIA GeForce I	1.38 cycle/nsecond	7.5	178927] benchmark

GPU Speed Of Light Throughput

High-level overview of the throughput for compute and memory resources of the GPU. For each unit, the throughput reports the achieved percentage of utilization with respect to the theoretical maximum. Breakdowns show the throughput for each individual sub-metric of Compute and Memory to clearly identify the highest contributor. High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

Compute (SM) Throughput [%]	84.08 (+0.06%)	Duration [msecond]	753.44 (+2,335.27%)
Memory Throughput [%]	0.02 (-34.65%)	Elapsed Cycles [cycle]	1,039,830,114 (+2,335.21%)
L1/TEX Cache Throughput [%]	0.03 (-34.54%)	SM Active Cycles [cycle]	1,024,731,742.63 (+2,332.74%)
L2 Cache Throughput [%]	0.02 (-34.65%)	SM Frequency [cycle/nsecond]	1.38 (-0.00%)
DRAM Throughput [%]	0.01 (-74.22%)	DRAM Frequency [cycle/nsecond]	6.79 (-0.00%)

High Throughput — The kernel is utilizing greater than 80.0% of the available compute or memory performance of the device. To further improve performance, work will likely need to be shifted from the most utilized to another unit. Start by analyzing workloads in the Compute Workload Analysis section.

FP64/32 Utilization — The ratio of peak float (fp32) to double (fp64) performance on this device is 32:1. The kernel achieved 0% of this device's fp32 peak performance and 26% of its fp64 peak performance. If Compute Workload Analysis determines that this kernel is fp64 bound, consider using 32-bit precision floating point operations to improve its performance. See the Kernel Profiling Guide for more details on roofline analysis.

FP64/32 Utilization — The achieved fp64 performance is 59% lower than the fp64 pipeline utilization. Check the Instruction Statistics section to see if using fused instructions can benefit this kernel.

Compute Workload Analysis

Detailed analysis of the compute resources of the streaming multiprocessors (SM), including the achieved instructions per clock (IPC) and the utilization of each available pipeline. Pipelines with very high utilization might limit the overall performance.

Executed Ipc Elapsed [inst/cycle]	0.08 (-1.85%)	SM Busy [%]	85.32 (+0.16%)
Executed Ipc Active [inst/cycle]	0.08 (-1.75%)	Issue Slots Busy [%]	1.93 (-1.75%)
Issued Ipc Active [inst/cycle]	0.08 (-1.75%)		

Very High Utilization — FP64 is the highest-utilized pipeline (85.3%) based on active cycles, taking into account the rates of its different instructions. It executes 64-bit floating point operations. The pipeline is over-utilized and likely a performance bottleneck. Based on the number of executed instructions, the highest utilized pipeline (85.3%) is FP64. It executes 64-bit floating point operations. Comparing the two, the overall pipeline utilization appears to be caused by frequent, low-latency instructions. See the Kernel Profiling Guide or hover over the pipeline name to understand the workloads handled by each pipeline. The Instruction Statistics section shows the mix of executed instructions in this kernel.

Memory Workload Analysis

Detailed analysis of the memory resources of the GPU. Memory can become a limiting factor for the overall kernel performance when fully utilizing the involved hardware units (Mem Busy), exhausting the available communication bandwidth between those units (Max Bandwidth), or by reaching the maximum throughput of issuing memory instructions (Mem Pipes Busy). Detailed chart of the memory units.

Memory Throughput [Mbyte/second]	24.40 (-74.22%)	Mem Busy [%]	0.02 (-34.70%)
L1/TEX Hit Rate [%]	87.80 (+50.46%)	Max Bandwidth [%]	0.02 (-34.65%)
L2 Hit Rate [%]	99.07 (-0.12%)	Mem Pipes Busy [%]	21.02 (+0.11%)

Launch Statistics

Summary of the configuration used to launch the kernel. The launch configuration defines the size of the kernel grid, the division of the grid into blocks, and the GPU resources needed to execute the kernel. Choosing an efficient launch configuration maximizes device utilization.

Grid Size	512 (+0.00%)	Function Cache Configuration	CachePreferNone (CachePreferNone)
Registers Per Thread [register/thread]	24 (+0.00%)	Static Shared Memory Per Block [byte/block]	0 (+0.00%)
Block Size	1,024 (+0.00%)	Dynamic Shared Memory Per Block [byte/block]	0 (+0.00%)
Threads [thread]	524,288 (+0.00%)	Driver Shared Memory Per Block [byte/block]	0 (+0.00%)
Waves Per SM	17.07 (+0.00%)	Shared Memory Configuration Size [Kbyte]	32.77 (+0.00%)

Occupancy

Occupancy is the ratio of the number of active warps per multiprocessor to the maximum number of possible active warps. Another way to view occupancy is the percentage of the hardware's ability to process warps that is actively in use. Higher occupancy does not always result in higher performance, however, low occupancy always reduces the ability to hide latencies, resulting in overall performance degradation. Large discrepancies between the theoretical and the achieved occupancy during execution typically indicates highly imbalanced workloads.

Theoretical Occupancy [%]	100 (+0.00%)	Block Limit Registers [block]	2 (+0.00%)
Theoretical Active Warps per SM [warp]	32 (+0.00%)	Block Limit Shared Mem [block]	16 (+0.00%)
Achieved Occupancy [%]	48.90 (+15.22%)	Block Limit Warps [block]	1 (+0.00%)
Achieved Active Warps Per SM [warp]	15.65 (+15.22%)	Block Limit SM [block]	16 (+0.00%)

Occupancy Limiters — This kernel's theoretical occupancy is not impacted by any block limit. The difference between calculated theoretical (100.0%) and measured achieved occupancy (48.9%) can be the result of warp scheduling overheads or workload imbalances during the kernel execution. Load imbalances can occur between warps within a block as well as across blocks of the same kernel. See the CUDA Best Practices Guide for more details on optimizing occupancy.

Source Counters

Source metrics, including branch efficiency and sampled warp stall reasons. Warp Stall Sampling metrics are periodically sampled over the kernel runtime. They indicate when warps were stalled and couldn't be scheduled. See the documentation for a description of all stall reasons. Only focus on stalls if the schedulers fail to issue every cycle.

Branch Instructions [inst]	364,832,223 (+2,313.70%)	Branch Efficiency [%]	99.89 (+0.37%)
Branch Instructions Ratio [%]	0.15 (+0.98%)	Avg. Divergent Branches	3,454.42 (+476.06%)

Uncoalesced Global Accesses — This kernel has uncoalesced global accesses resulting in a total of 3717438 excessive sectors (83% of the total 4464016 sectors). Check the L2 Theoretical Sectors Global Excessive table for the primary source locations. The CUDA Programming Guide had additional information on reducing uncoalesced device memory accesses.

L2 Theoretical Sectors Global Excessive

Location	Value	Value (%)
0x7f6252e35b10 in mandel_kernel	3,670,016	99
0x7f6252e35ab0 in mandel_kernel	47,424	1

Still missing data? Collect the *full* metric set, enable individual sections or learn how to select individual metrics for collection.