Page: Details | Result: 0 - 23 - nqfaf | Add Baseline | Apply Rules | Occupancy Calculator | Source Comparison | Save as PDF

	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	23 - nqfaf (9248, 1, 1)x(64, 1, 1)	651,37 msecond	884.271.856	38	0 - NVIDIA GeForce GTX 1650 Ti	1,35 cycle/nsecond	7.5	[10612] NQueensFAF.exe

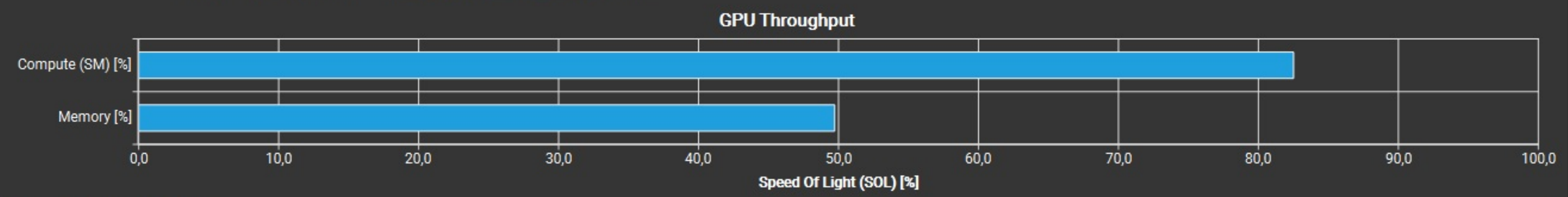
GPU Speed Of Light Throughput

GPU Throughput Chart

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.

Compute (SM) Throughput [%]	82,51	Duration [msecond]	651,37
Memory Throughput [%]	49,73	Elapsed Cycles [cycle]	884.271.856
L1/TEX Cache Throughput [%]	87,45	SM Active Cycles [cycle]	881.714.192,69
L2 Cache Throughput [%]	0,23	SM Frequency [cycle/nsecond]	1,35
DRAM Throughput [%]	0,28	DRAM Frequency [cycle/nsecond]	6,02

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.

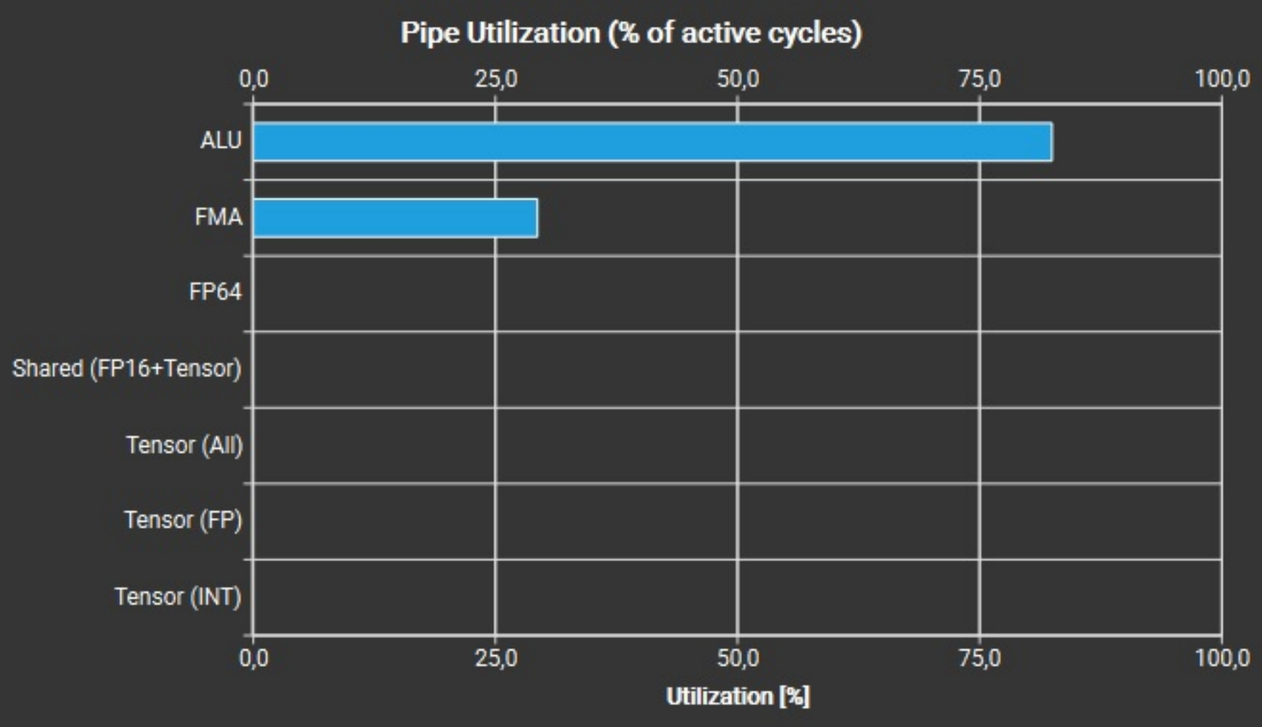


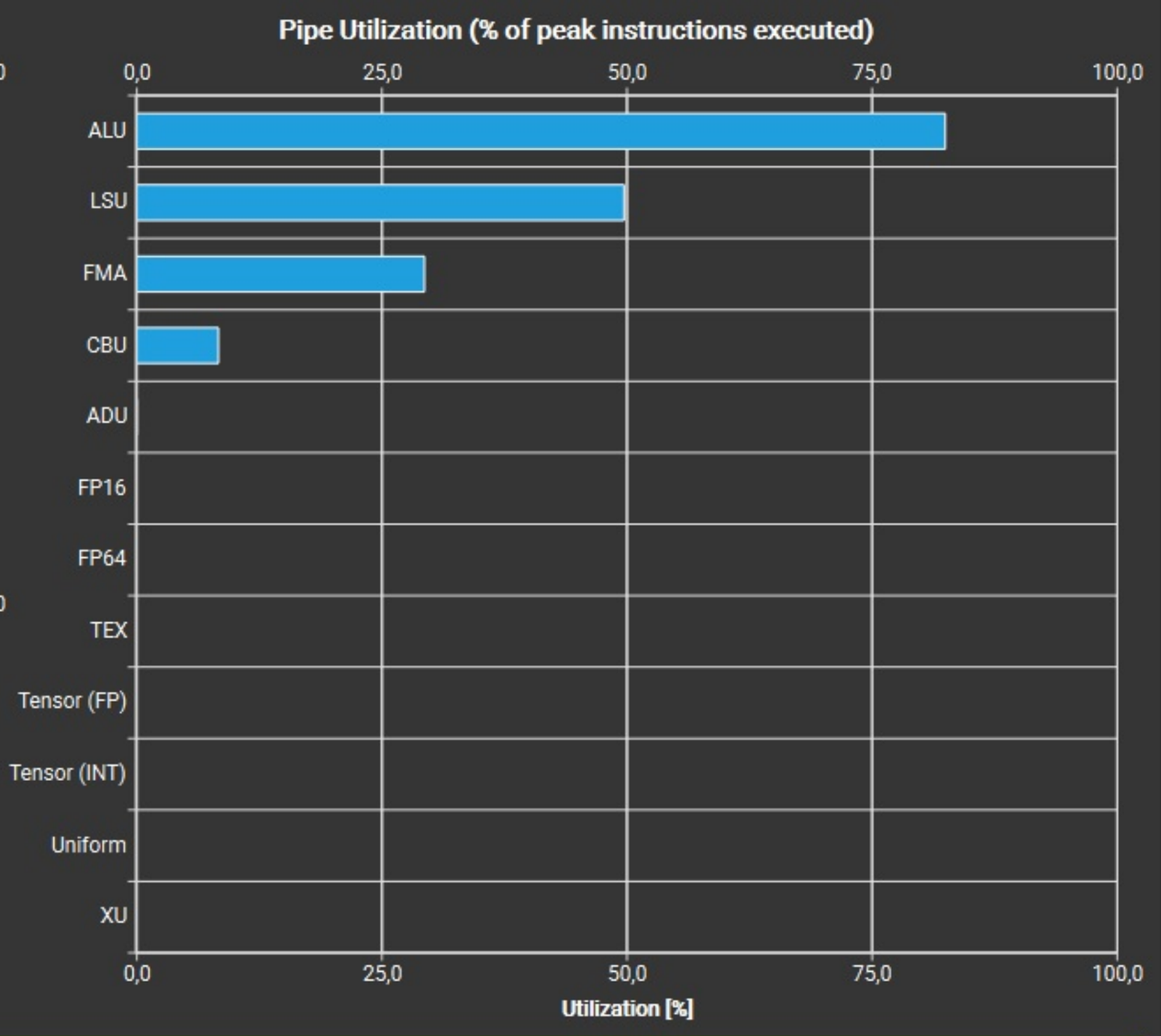
Compute Workload Analysis

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]	3,15	SM Busy [%]	82,47
Executed Ipc Active [inst/cycle]	3,15	Issue Slots Busy [%]	78,84
Issued Ipc Active [inst/cycle]	3,15		

Very High Utilization ALU is the highest-utilized pipeline (82.5%) based on active cycles, taking into account the rates of its different instructions. It executes integer and logic operations. The pipeline is over-utilized and likely a performance bottleneck. Based on the number of executed instructions, the highest utilized pipeline (82.5%) is ALU. It executes integer and logic 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.





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	9.248	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	38	Static Shared Memory Per Block [Kbyte/block]	4,68
Block Size	64	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	591.872	Driver Shared Memory Per Block [byte/block]	0
Waves Per SM	44,46	Shared Memory Configuration Size [Kbyte]	65,54

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 [%]	81,25	Block Limit Registers [block]	24
Theoretical Active Warps per SM [warp]	26	Block Limit Shared Mem [block]	13
Achieved Occupancy [%]	73,15	Block Limit Warps [block]	16
Achieved Active Warps Per SM [warp]	23,41	Block Limit SM [block]	16

To customize your report even further, you might want to learn about custom sections and writing your own rules.
You might also want to consider adding individual metrics.