max\_blocks\_per\_multiprocessor 16

max\_warps\_per\_multiprocessor 32

max\_warps\_per\_scheduler 8

Di seguito solo 32 thread per blocco

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Mentre qui 64 thread per blocco. SI raggiunge una maggiore utilizzazione!

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

640 blocchi con 64 thread dovrebbe essere il massimo che si può raggiungere

Infatti dopo che si runna con 640 non ci sono più miglioramenti significativi che vengono suggeriti…

Immagine che contiene testo, software, Software multimediale, schermata

Descrizione generata automaticamente

Immagine che contiene schermata, testo, software, linea

Descrizione generata automaticamenteImmagine che contiene software, Software multimediale, Software per la grafica, Icona del computer

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Immagine che contiene testo, software, Software multimediale, linea

Descrizione generata automaticamente

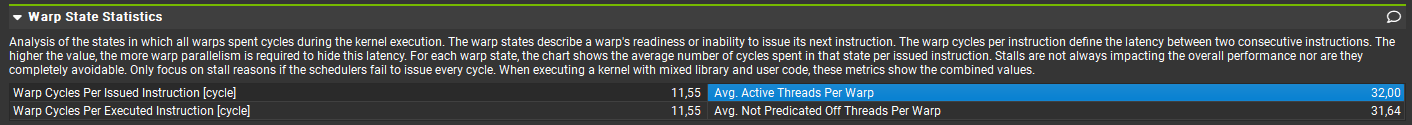
Immagine che contiene schermata, Software multimediale, Software per la grafica, software

Descrizione generata automaticamente

Immagine che contiene schermata, testo, linea, software

Descrizione generata automaticamente

Vedi anche file Nsight salvato…



In NVIDIA Nsight Compute, the "**Not Predicated Off Threads per Warp**" metric provides an average count of threads per warp that are not predicated off during the execution of an instruction.

To understand this metric more clearly, let's break down the key components:

1. \*\*Warp\*\*: In NVIDIA's GPU architecture, a warp consists of 32 threads that execute the same instruction simultaneously.

2. \*\*Predicated Off\*\*: In SIMD (Single Instruction, Multiple Data) architectures like those used by NVIDIA GPUs, instructions can be conditionally executed using predicates. When a thread in a warp is "predicated off," it means that the thread does not execute the instruction due to the evaluation of the predicate condition.

Therefore, "Not Predicated Off Threads per Warp" essentially counts the number of threads within a warp that are actively executing an instruction, as opposed to being skipped due to predicate conditions.

### Example Calculation

Suppose a warp consists of 32 threads. If an instruction is predicated such that only 20 threads out of 32 execute it, the remaining 12 threads are predicated off. The metric "Not Predicated Off Threads per Warp" would be 20 in this case.

### Average Not Predicated Off Threads per Warp

This metric is averaged over all warps and instructions during a kernel execution. If multiple warps are considered, the metric provides an average value indicating the typical number of active threads per warp for the instructions being analyzed.

### Usage

This metric is useful for understanding the efficiency of instruction execution within warps. A lower value indicates that more threads are being predicated off, which may point to inefficiencies in how the code is utilizing the GPU's parallel processing capabilities. Conversely, a higher value (closer to 32) indicates that most threads are actively participating in instruction execution, which is generally more efficient.

In summary, the "average not predicated off threads per warp" metric provides insight into how effectively the threads within each warp are being utilized during the execution of a kernel, by measuring the average number of threads that are not skipped by predicate conditions.