Parallelized Nearest Neighbor Upscaler

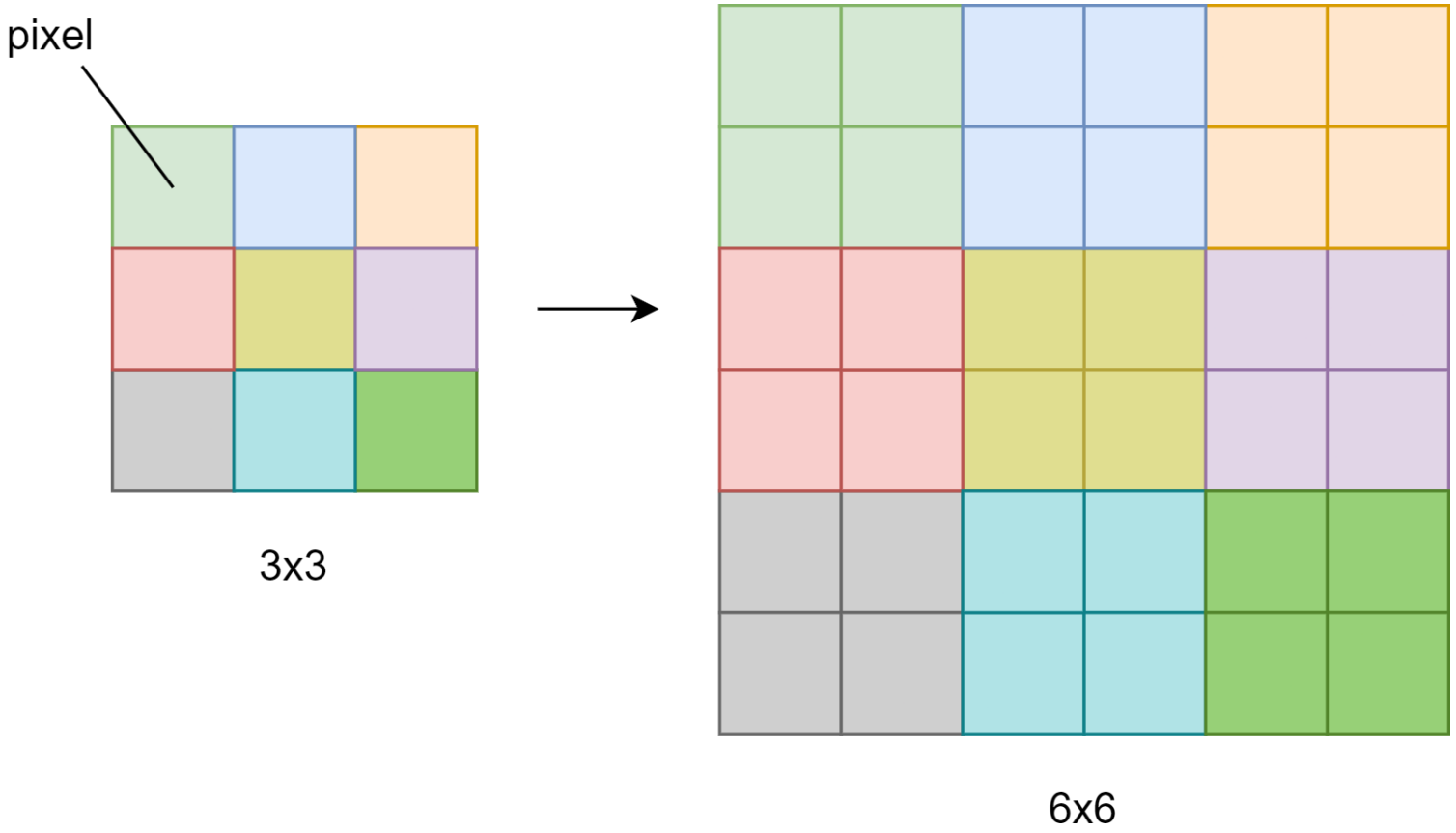
Biagio Cornacchia

Gianluca Gemini

Matteo Abaterusso

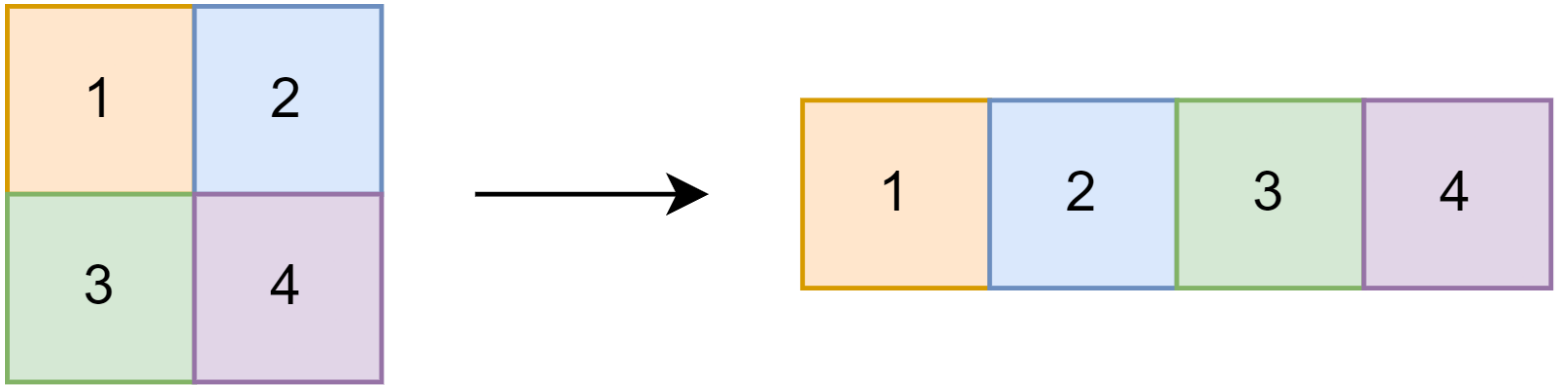
Overview

Image upscaling is the process that allows to increase the resolution of an image, trying to minimize the loss in image quality. The chosen algorithm is the **nearest-neighbor**.



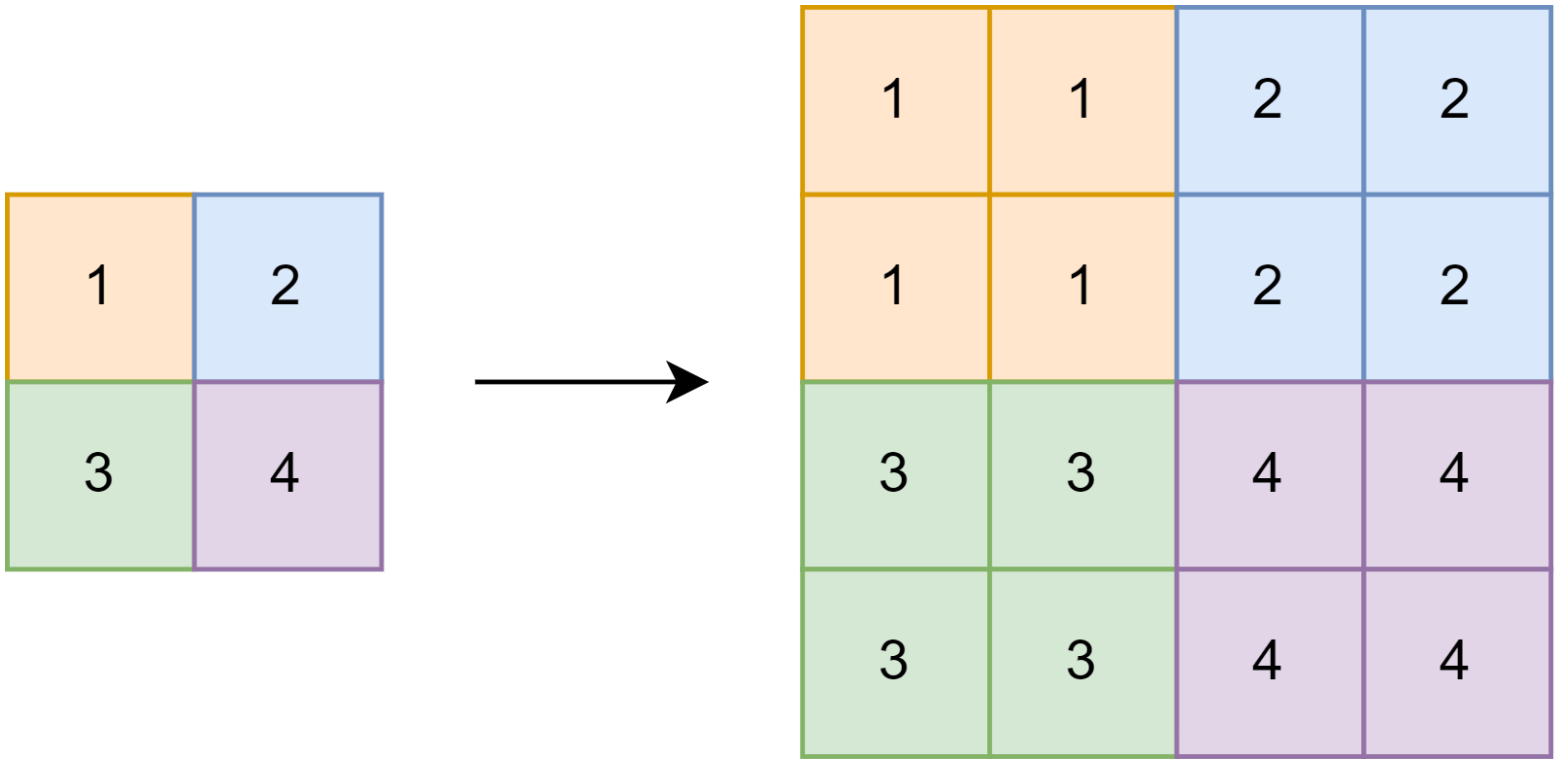
CPU Version

The image to upscale is loaded into a **one-dimensional byte array**. Each pixel is made by 4 byte which correspond to the four channels named red, green, blue and alpha.



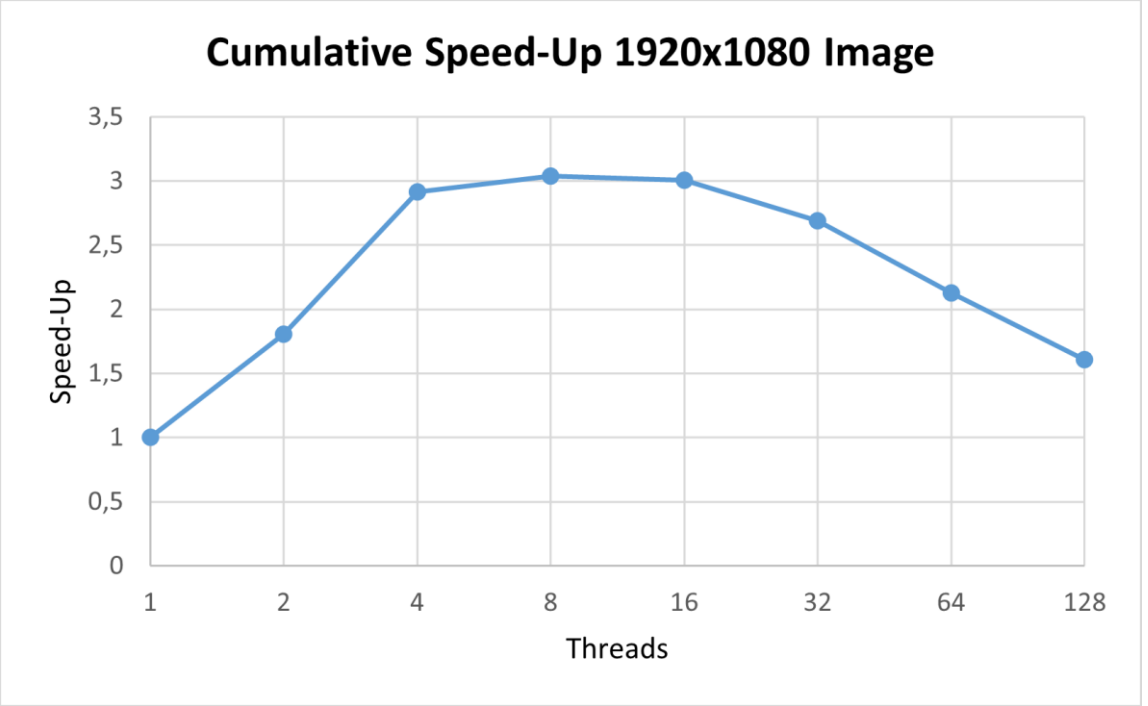
CPU Version

A **thread** handles a subset of **consecutive pixels** of the original image. Each of them is copied into the upscaled image, and the number of copies is defined by the **upscaling factor**.



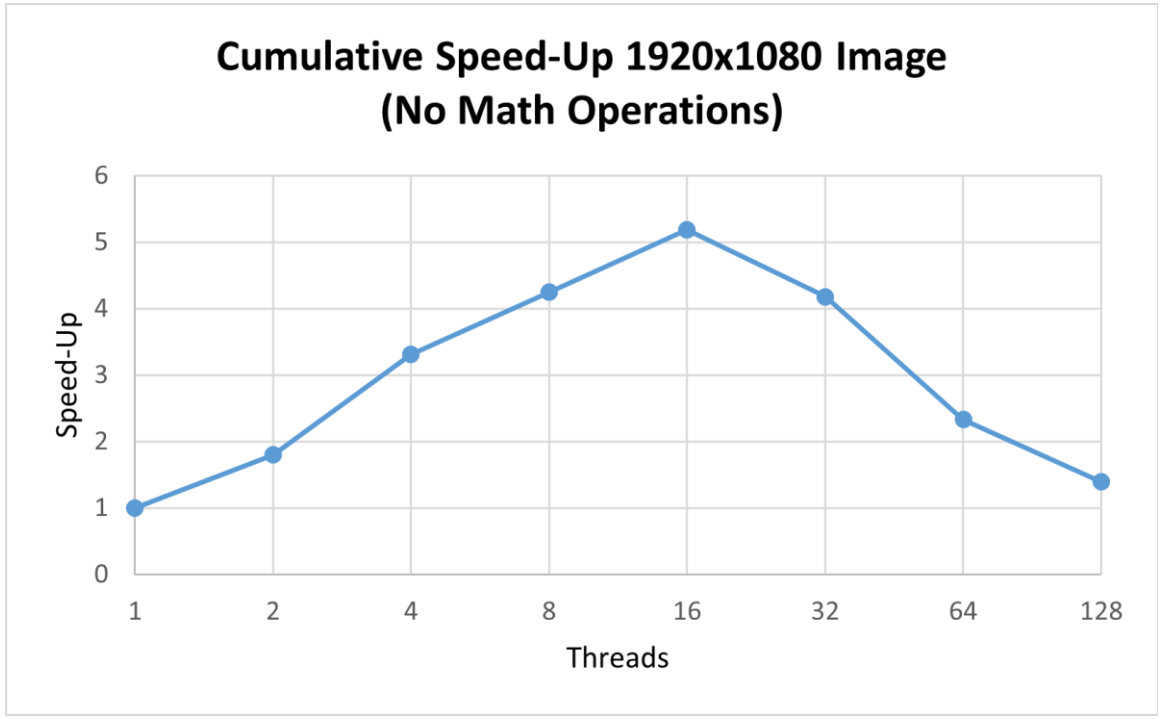
CPU Version Results

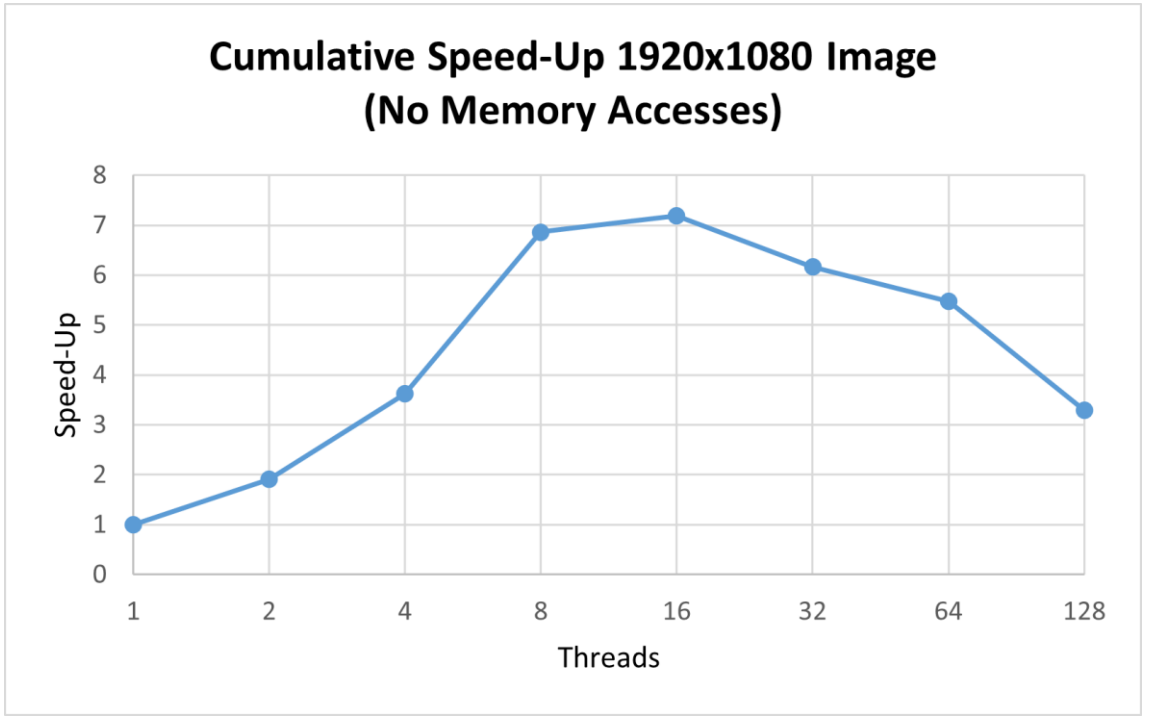
The upscaler has been tested on a **Ryzen 7 5800H** with **8 cores**, using different image sizes and varying the number of threads.



CPU Version Results

The algorithm consists of a **CPU bound** part and an **I/O bound** part. To figure out what cause the saturation, tests have been repeated excluding one of the two parties in turn.





GPU Version

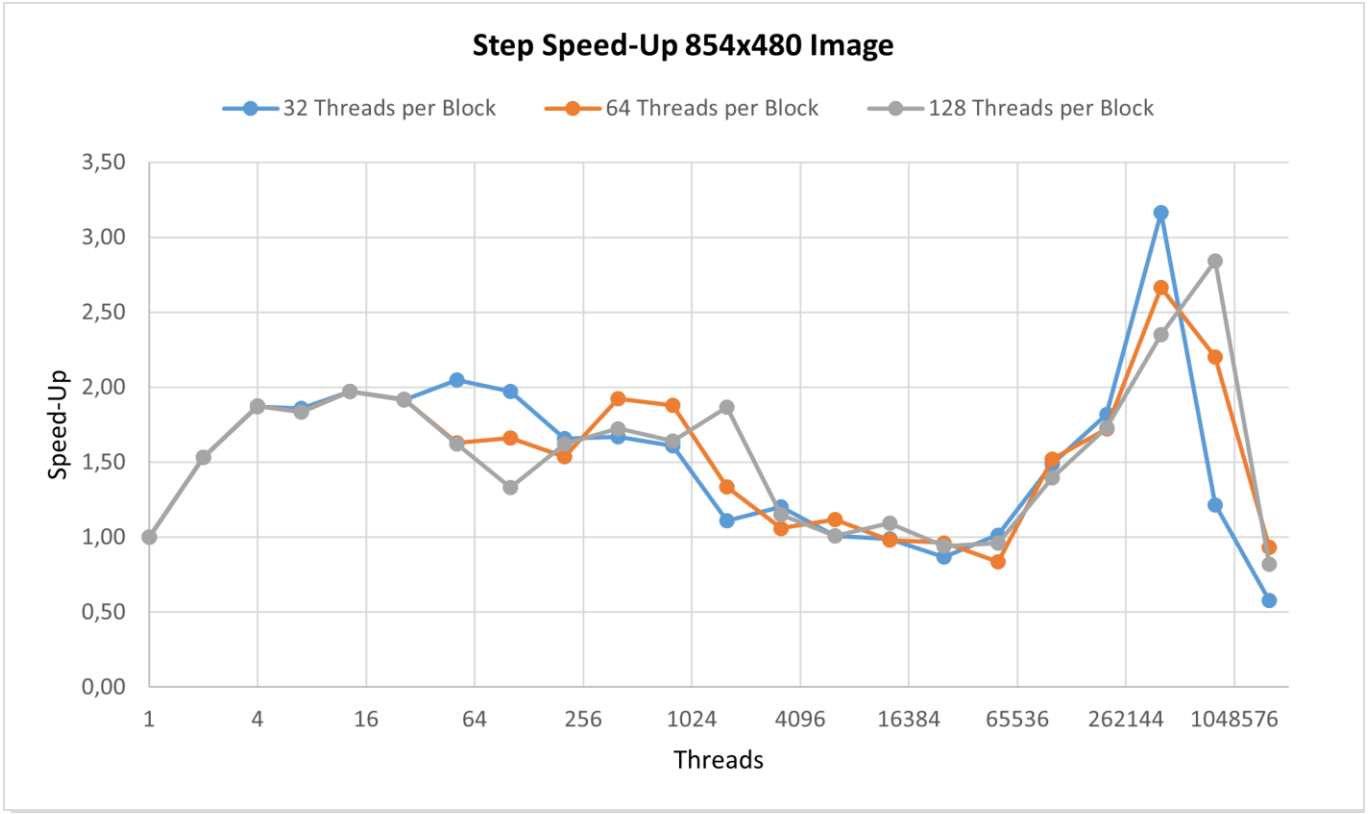
In the GPU version the kernel iterates the **upscaled image pixels** instead of the original ones. This **increases** the number of total **threads**, but it allows to **better manage** the **memory locality**.

The CPU version access pattern continually causes **jumps** to potentially **distant locations** in memory, not allowing proper utilization of the cache. To avoid this problem, the original image is loaded into the **texture memory**.

Finally, the pixels that a thread must handle are not decided a priori, but they are obtained using the **thread ID** and the **block ID** of each thread.

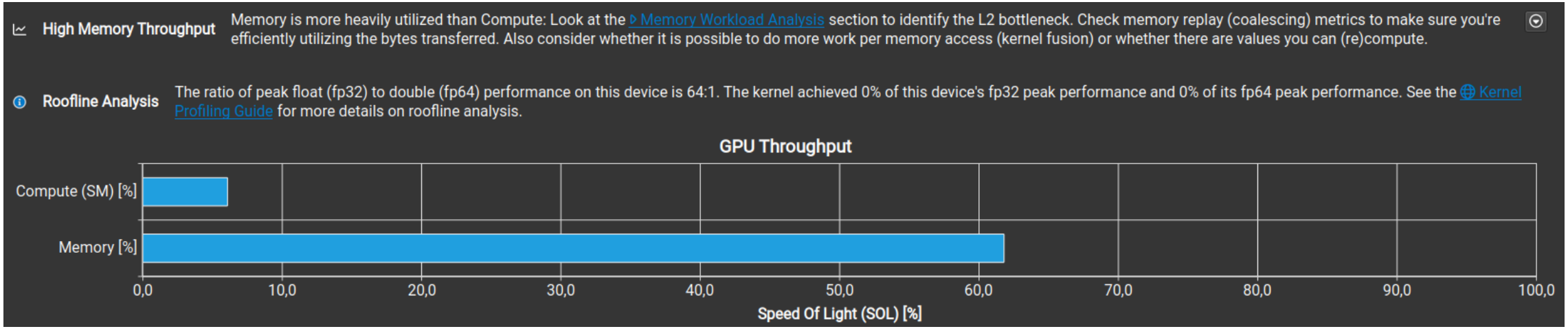
GPU Version Results

The tests have been performed on a **NVIDIA RTX 3060 mobile** with **30 multi-processors** (MP) and **128 CUDA cores per MP**. The warp size is 32, so each multi-processor has 4 partitions and the total of **CUDA cores** is **3840**.



GPU Version Performance Analysis

GPU Speed of Light Throughput section:



Scheduler Statistics section:

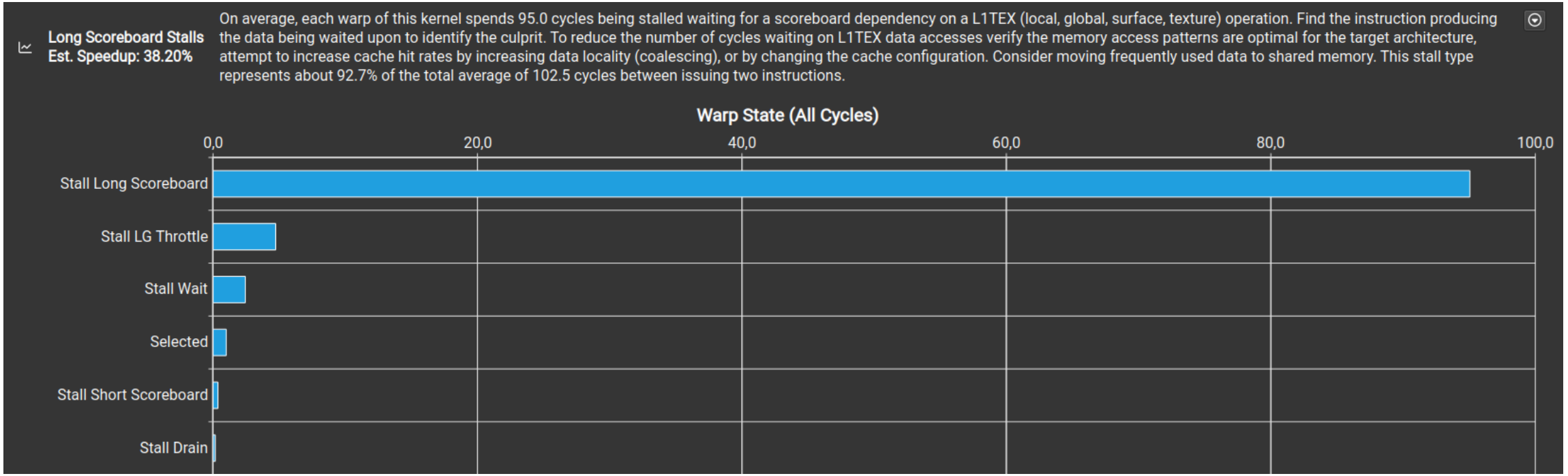
Active Warps Per Scheduler [warp]	3,34	No Eligible [%]	96,75
Eligible Warps Per Scheduler [warp]	0,03	One or More Eligible [%]	3,25
Issued Warp Per Scheduler	0,03		

Issue Slot Utilization
Est. Local Speedup: 38.20%

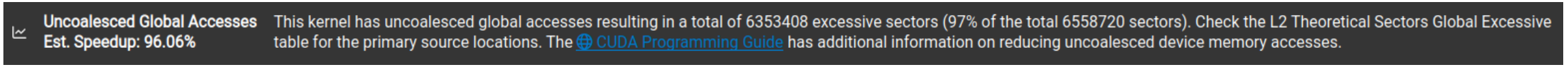
Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 30.7 cycles. This might leave hardware resources underutilized and may lead to less optimal performance. Out of the maximum of 12 warps per scheduler, this kernel allocates an average of 3.34 active warps per scheduler, but only an average of 0.03 warps were eligible per cycle. Eligible warps are the subset of active warps that are ready to issue their next instruction. Every cycle with no eligible warp results in no instruction being issued and the issue slot remains unused. To increase the number of eligible warps, reduce the time the active warps are stalled by inspecting the top stall reasons on the Warp State Statistics and Source Counters sections.

GPU Version Performance Analysis

Warp State Statistics section:

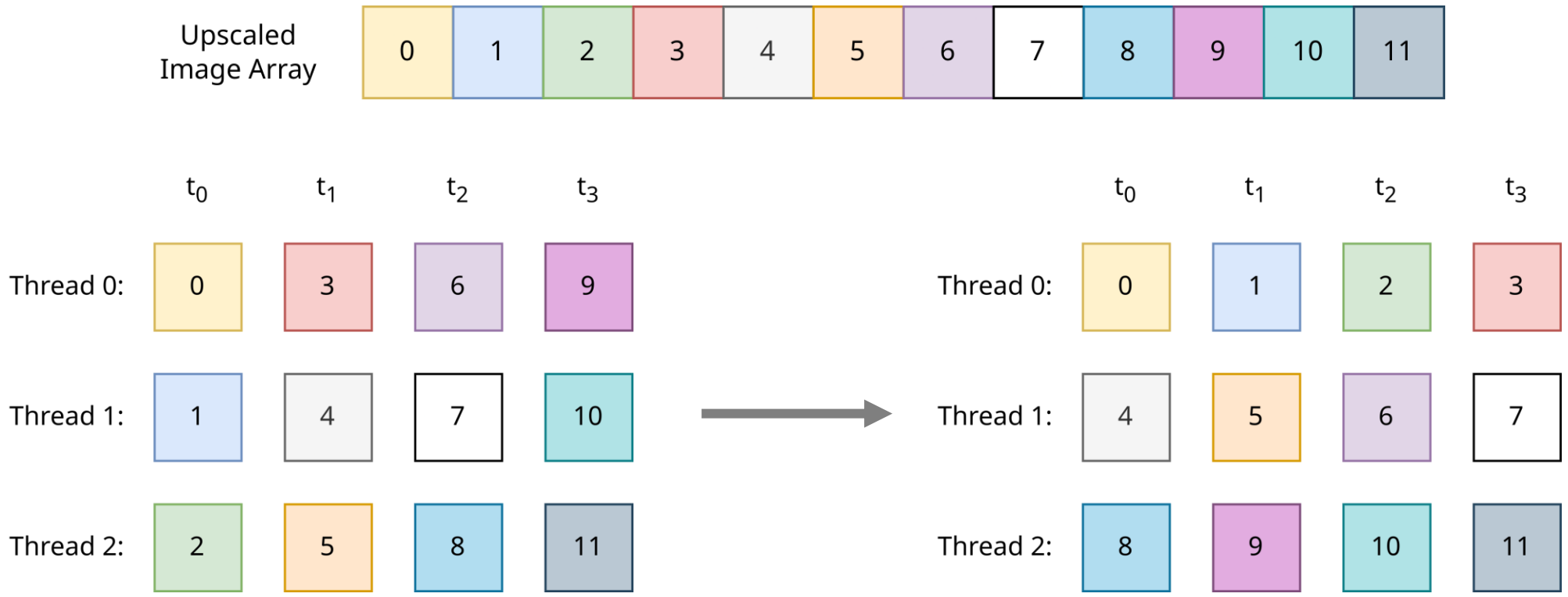


Source Counter section:



GPU Optimized Version

Uncoalesced global accesses occur if threads belonging to the same warp perform a memory operation at the **same instant** at **addresses** that are **not consecutive**.



GPU Optimized Version

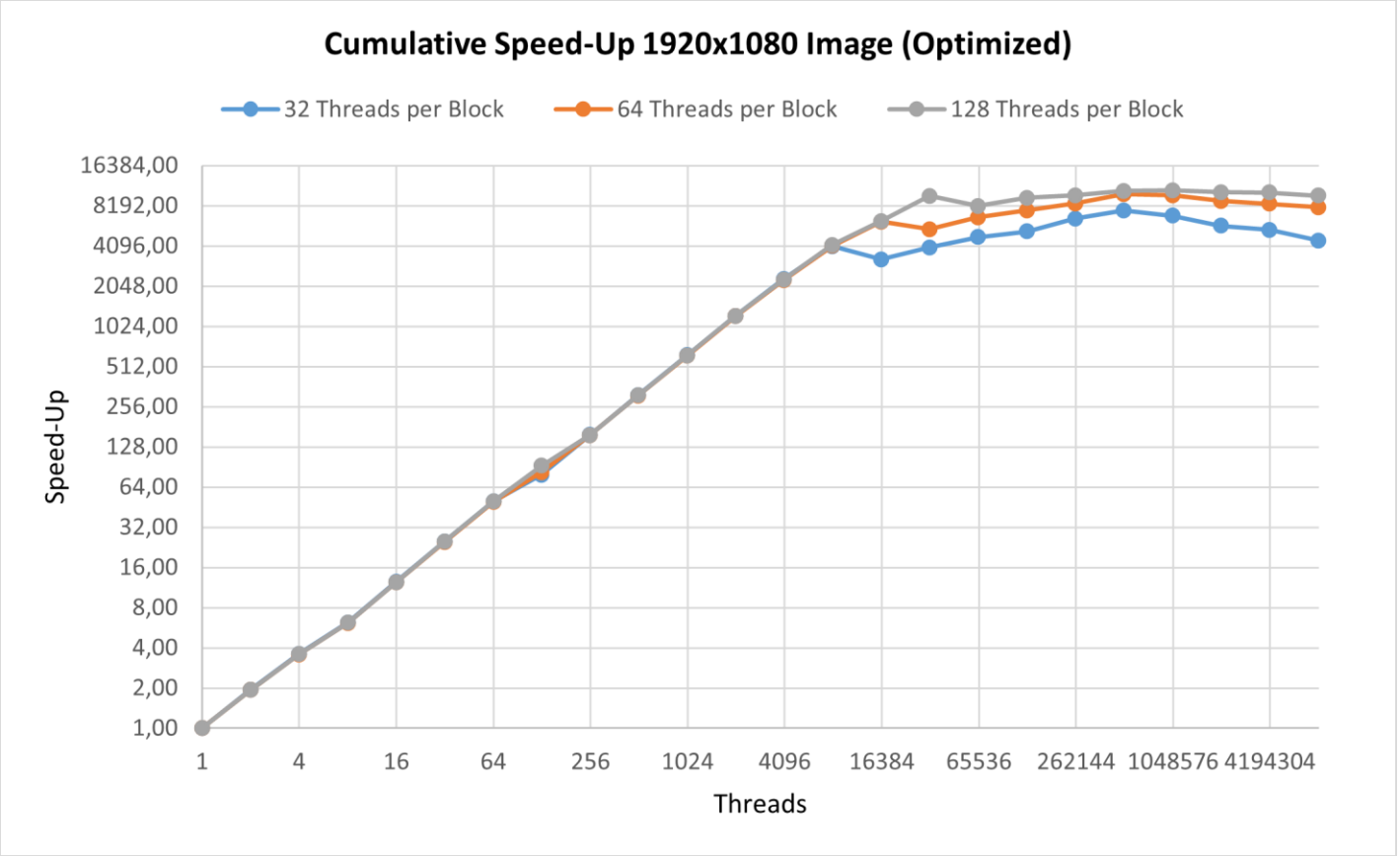
The main optimization concerns the **indexes system**, that is modified to solve the **uncoalesced global accesses** problem.

Read-only used variables are computed a priori and loaded into **constant memory**. This reduces the **mathematical operations** of each thread and the **access time** to the values.

CUDA built-in structure **uchar4** replaced **memcpy** to ensure that copying a pixel is done through a **single memory transaction**.

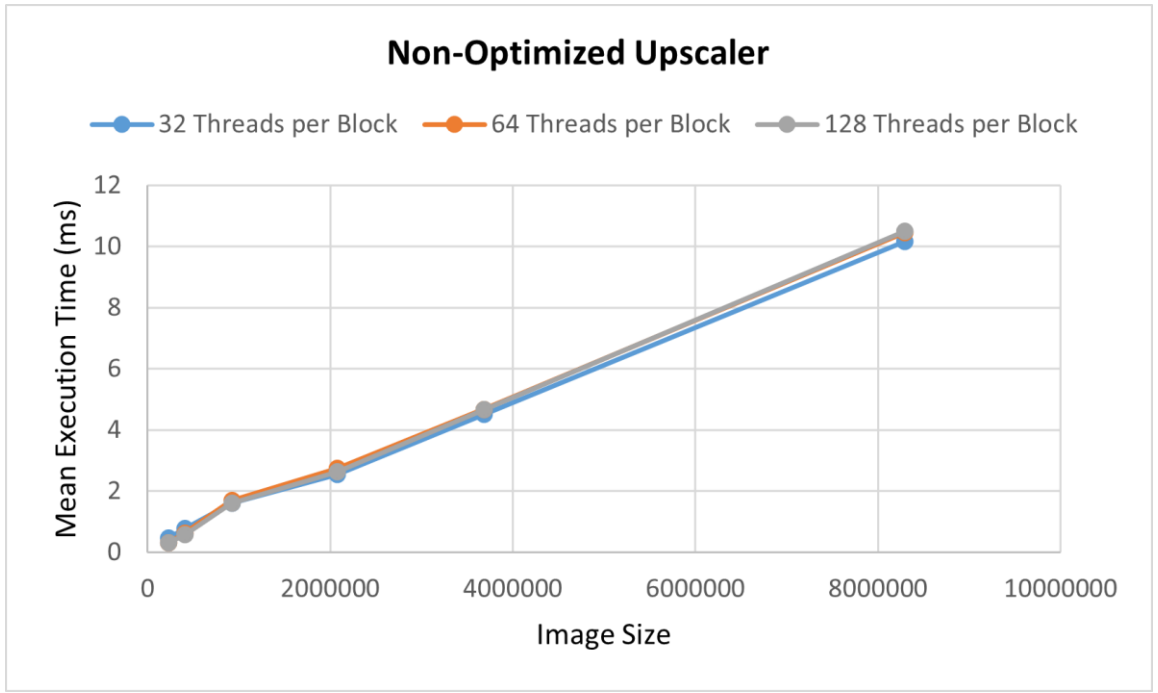
GPU Optimized Version Results

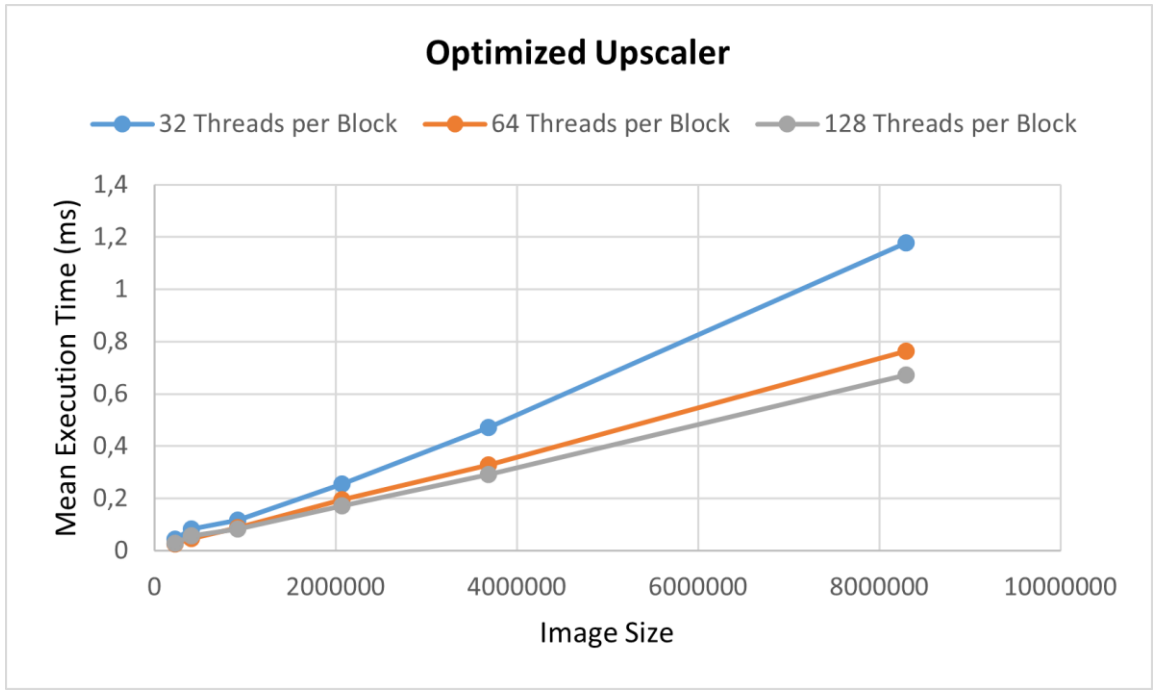
To compare the new kernel with the previous one, the same test has been performed, including also other resolutions.



GPU Optimized Version Results

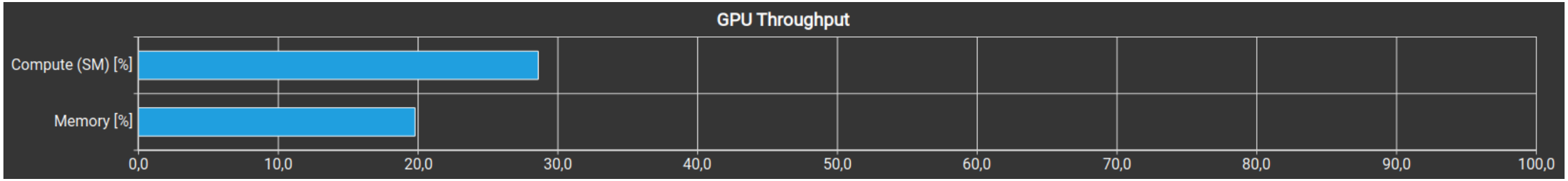
To test the algorithms scalability, the **number** of **pixels** has been fixed and the **size** of the **image** has been varied.





GPU Optimized Version Results

GPU Speed of Light Throughput section:



Warp State Statistics section: