V-NOVA
VIDEO AS NATURE INTENDED™

Determinism and Low-Latency GPU Scheduling in OpenCL

July 2019

Balázs Keszthelyi &
Dr Michele Sanna

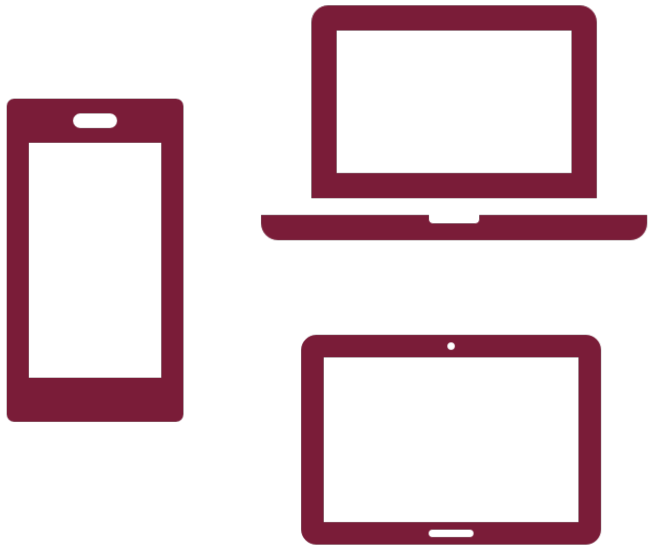
V-NOVA

AI-based compression technologies to make the future of digital come alive

- Worldwide IP coverage from 300+ patents
- Industry award-winning
- Powering major operators

Multi-Stream Challenges

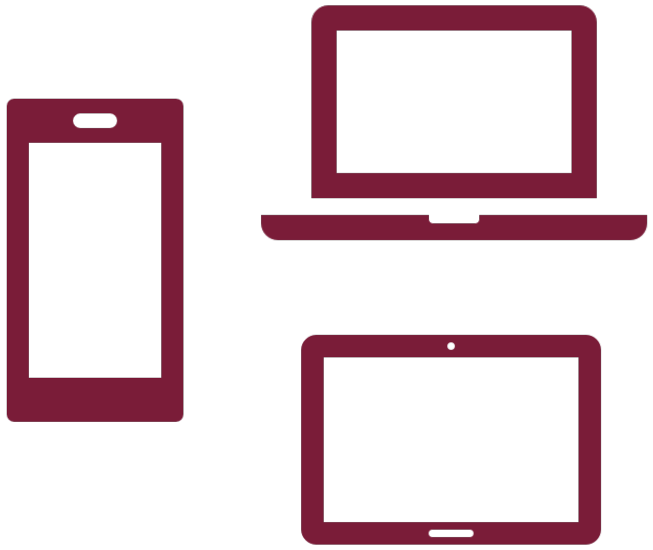
- similar compute load
- no rigid ordering of input
- QoS is expected, latency should be capped
- throughput minimum per stream
- logically independent to the user
- failure should happen per stream only



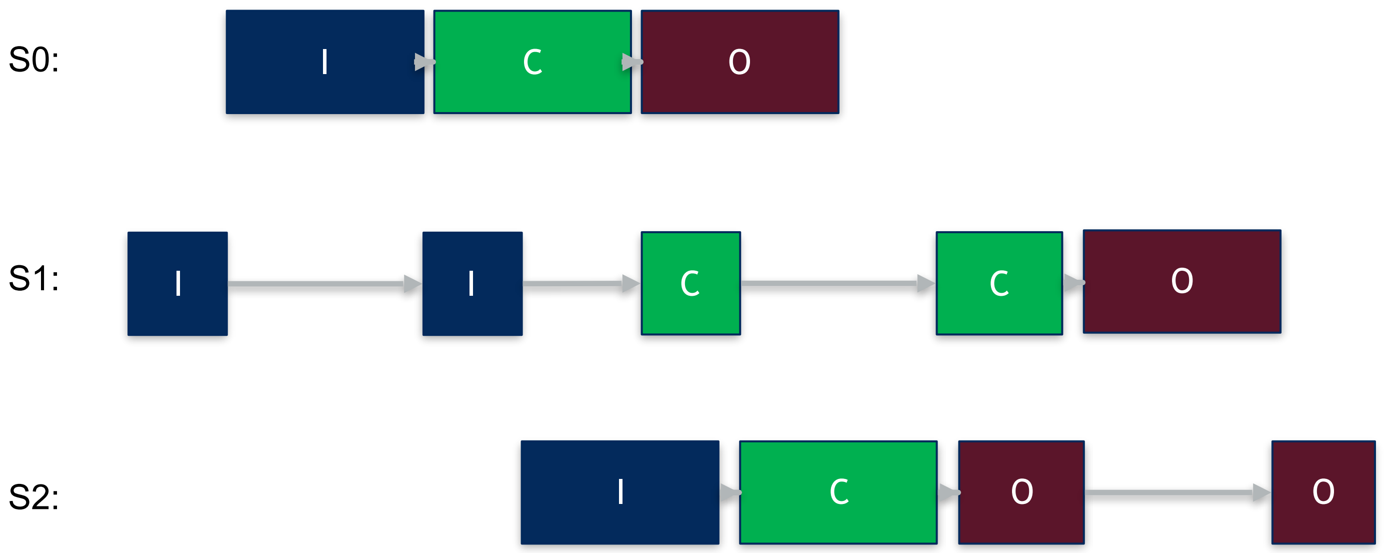
Multi-Stream Challenges

Multi-process:
+ in theory process isolation enables streams to fail alone
- in practice GPU can get locked up inter-process
- doesn't allow for advanced GPU synchronisation primitives

Multi-threaded:
+ enables GPU synchronisation within one OpenCL context
- lack of process isolation (theoretic)



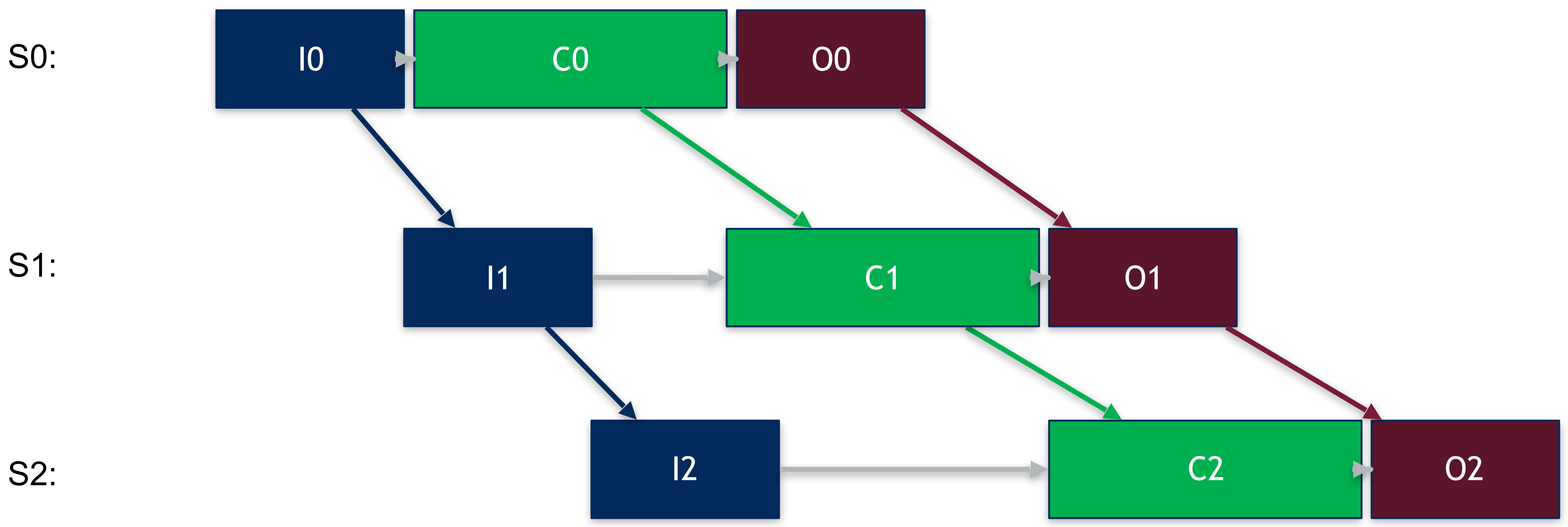
Multi-Process Scheduling – 1 Process Per Stream



Synchronisation Options

- Inter-process synchronisation: CPU-driven, inconvenient and would mean a performance penalty due to CPU-GPU round trips
- If we were using a single process instead, GPU synchronisation would become possible via OpenCL events
- The OpenCL context itself supports multiple threads

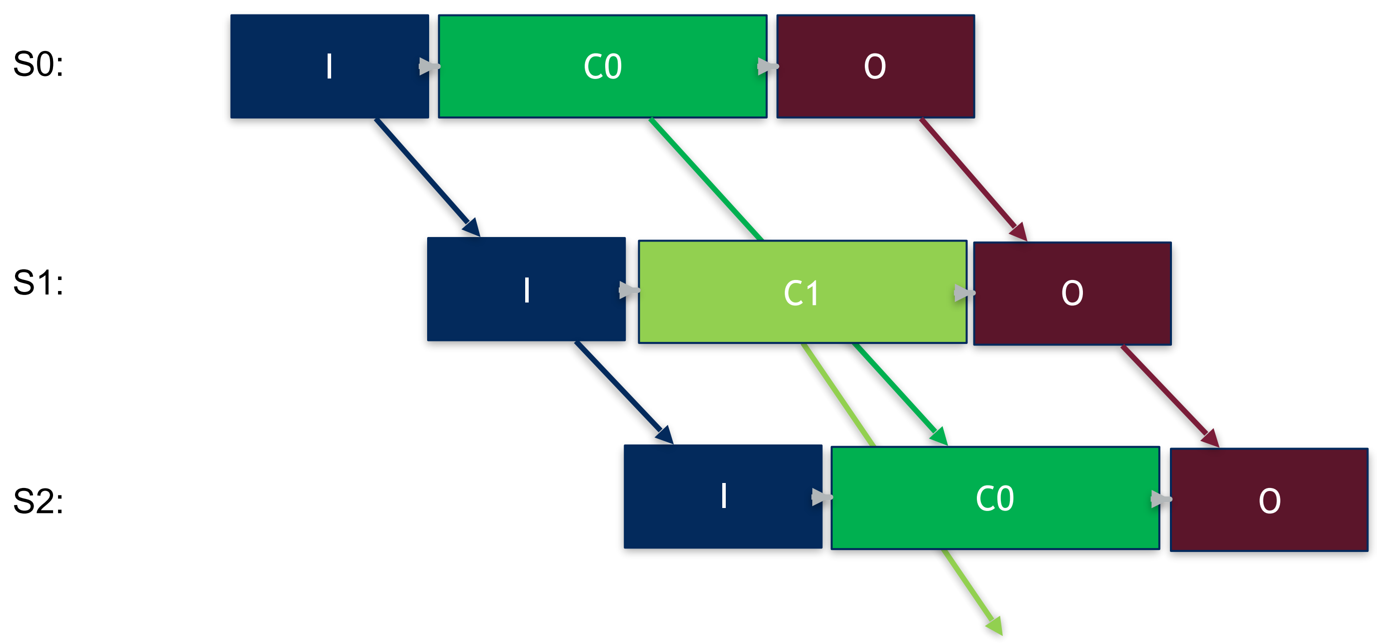
Multi-Threaded Scheduling – 1 Queue Per Stream



Mind Map of the GPU HW

- More efficient utilisation is possible by mapping the OpenCL command queues to HW queues
- 1x input queue – corresponding DMA engine and PCI-E path
- 1x output queue – corresponding DMA ending and PCI-E path
- Cx compute queues - in this example we are going to use 2x
- Note: using multiple compute queues may increase the achieved GPU utilisation by offering more parallel workload for the GPU to execute

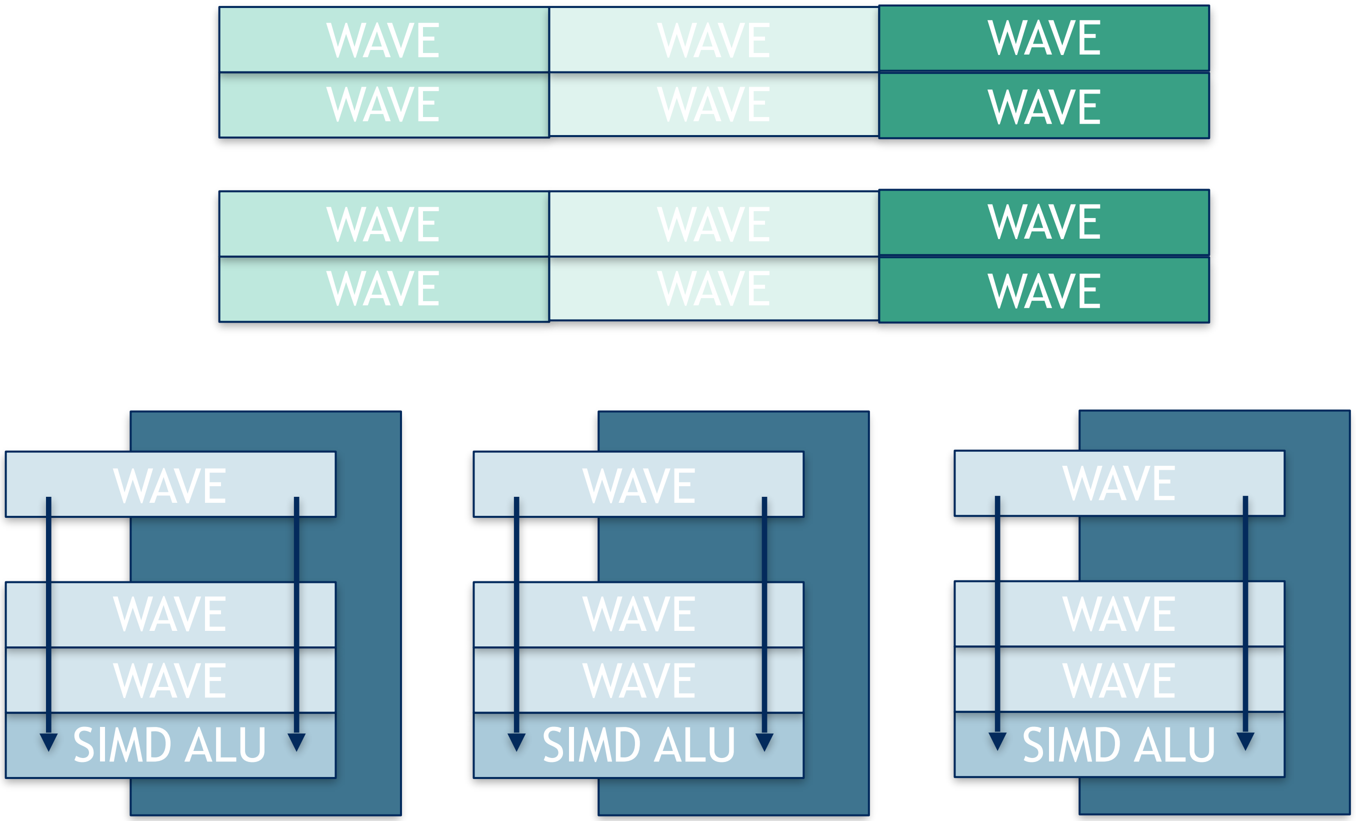
Multi-Threaded Scheduling – 1 Queue Per HW Queue



Poor Man's Device Fission – Kernel Fission (Experimental)

- At this point, reasonable performance and stability are possible
- But the independent forward progress of tasks running on different compute queues may lead to non-deterministic performance due to resource virtualisation
- Albeit device fission is usually not supported, knowing the parameters of the target GPU, by limiting the size of a kernel launch, it is possible to limit the maximum number of GPU "cores" working on the particular problem…

Poor Man's Device Fission – Mind Map (Experimental)



- Number of waves per core is kernel specific (register/local memory)
- Larger batches amortize the cost of the kernel launch
- Smaller batches could prevent one kernel overtaking all

Pre-Emptive Kernel Launches (Experimental)

- In the previous diagrams, small gaps could be seen between execution stages
- In practice, there are gaps between kernel launches, that may be considerable between kernels where their runtime is otherwise negligible
- It is possible to pull the gaps closer by issuing the commands well in advance to the GPU
- Question: User Events?

Results

Throughput increase

- From +4%
- Up-To +40%

Latency Decrease

- From -33%
- Up-To -80%
- Generally more stable operating latency

We are hiring!

Roles:

- > GPU SW Engineer
- > Research Engineer
- > Software Engineer

Contacts
https://careers-v-nova.icims.com
emma.gilbert@v-nova.com
stergios.poularakis@v-nova.com

Thank You

info@v-nova.com
www.v-nova.com