Impact of CPU-GPU data transfers on mobile device GPGPU

Tommaso Maestri
Samsung R&D Institute UK
Staines-upon-Thames, Middlesex, UK
t.maestri@samsung.com

Keywords: GPGPU, OpenCL, shared memory, mobile device

1 Introduction

Current mobile devices have powerful GPUs capable of running both graphics related tasks and general purpose computation (GPGPU) tasks, but their architecture is different compared to Desktop machines. Mobile devices are SoC and one of their main characteristics is that the CPU and GPU share the same off-chip (system) memory.

System memory represents a key factor in the cooperation between CPU and GPU when dealing with GPGPU. They both have to share data and if this is not managed in the proper way it can lead to poor performance or even waste any improvement the GPU can provide. This session will focus on the key issues related to data sharing that influence performance and explains how to address them.

2 An in-depth look on the shared memory

On mobile devices CPUs and GPUs use the same physical memory, and it is quite different from the architecture used in desktop machines, where the GPU has a dedicated off-chip memory. This means that both units should be able to access the shared data in a simple way without any additional cost and avoiding data copy; ideally just passing a pointer to an address in memory.

But even if the memory is shared, the mechanisms to map and unmap buffers take time and they have the final effect of reducing any possible speed-up. Moreover, creating a buffer requires time so to have an optimize solution the memory should be allocated during the initialization phase. But this is not suitable for all cases because the size should be known in advance.

CPUs and GPUs also share a common bus in order to access the off-chip memory and this could represent a bottleneck when both units are running code that requires frequent read or write operations.

3 Load distribution and data transfer

Any data copy introduced into the process due to the parallelization represents a performance reduction and should be avoided when possible. The data can be shared using mapping mechanisms. This way it can be accessed by both the CPU and GPU.

Another important aspect is the interoperation between graphic and GPGPU, in particular the possibility to work using the same data structures (textures). This represents a good way to save time and avoid additional copies.

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792550

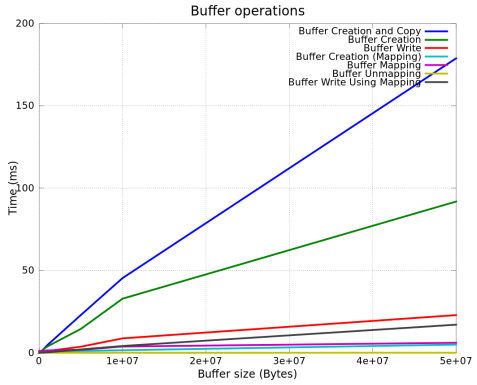

Figure 1: *Cost of mapping operations on buffer*

4 Profiling results

Examples are going to be shown about how the performance can depend on CPU-GPU data transfer, covering all the aspects cited so far using the OpenCL API as framework to execute GPGPU tasks on different vendors' devices.

5 Conclusion

The analysis and results clearly show the bottlenecks of the current mobile devices' SoC architecture. Often a good part of the gain obtained by parallelising the code is wasted due to additional data copying or mapping. Data transfer must always be considered when profiling for performance.

Techniques on how manage data transfer should be carefully investigated and different solutions should be trialed with extensive profiling. The investigation should also involve different vendors GPUs because this depends on how the memory is managed and is strictly related to the hardware architecture which is transparent to the user but with effects on performance.

References

2011. The opencl specification - version 1.1. Khronos OpenCL Working Group.