Experience with Beignet OpenCL on Low Power Intel SoCs

Felice Pantaleo – felice.pantaleo@cern.ch

CHEP 2016 - 22nd International Conference on Computing in High Energy and Nuclear Physics – 10th – 14 th October 2016

Low Power Architectures

Power consumption is becoming a hot-spot in data centres' total bill

At CERN, this will be even more important with the HL-LHC upgrade: LHC experiments will have to cope with 2-3x the amount of data

Increasing interest in alternative low power architectures based on ARM

Increasing interest in complementary highly efficient accelerators like GPUs

Intel Skylake

	Intel® Core™ m3-6Y30	Intel® Core™ i7-6700K
Processor Number	M3-6Y30	i7-6700K
# of Cores	2	4
# of Threads	4	8
Processor Base Frequency	900.00 MHz	4.00 GHz
Max Turbo Frequency	2.20 GHz	4.20 GHz
Cache	4 MB SmartCache	8 MB SmartCache
TDP	4.5 W	91 W
Processor Graphics	Intel® HD Graphics 515	Intel® HD Graphics 530
Graphics Base Frequency	300.00 MHz	350.00 MHz
Graphics Max Dynamic Frequency	850.00 MHz	1.15 GHz

Credits: Intel

Beignet OpenCL

Open-source implementation

70000+ lines of C and C++

Distributed under the LGPLv2.1

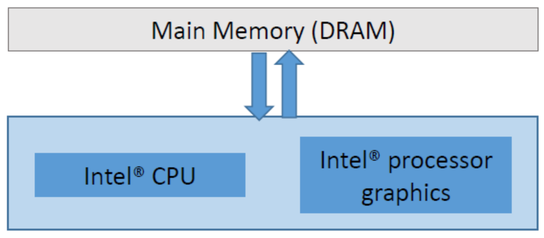
Supported GPUs: Intel HD, Iris, Iris Pro

Supported CPUs: Intel Core, Atom

Memory Management

Applications can inform the driver of their memory usage scenarios

- during allocation
- memory transfer API



Driver implementations create internal copies of memory buffers

- beneficial for improving caching behavior
- dramatic impact on performance
- device-specific knowledge to avoid these copies

Integrated graphics:

best performance when using zero copy

no need to create a host and a device version of data

No NUMA effects: memory shared between the CPU and GPU can be efficiently accessed by both devices.

Adding OpenCL in existing codebase:

must create a buffer that is aligned to a 4096 byte boundary and have a size that is a multiple of 64 bytes

```
int *pbuf = (int)_aligned_malloc(sizeof(int)* 256, 4096);
cl_mem myZeroCopyCLMem0bj = clCreateBuffer(ctx,…CL_MEM_USE_H0ST_PTR…);
```

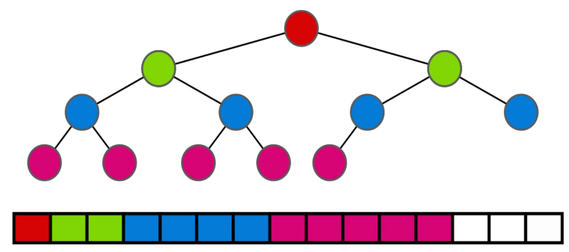
OpenCL-managed host allocation

```
buf=clCreateBuffer(ctx,CL_MEM_ALL0C_H0ST_PTR, …)
```

FKDTree

Parallel Heapified KDTree, mainly used for track seeding or clustering

Branchless search in TBB, OpenCL, CUDA, OpenMP

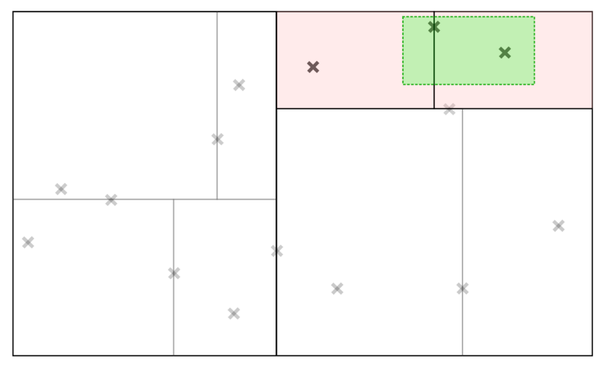


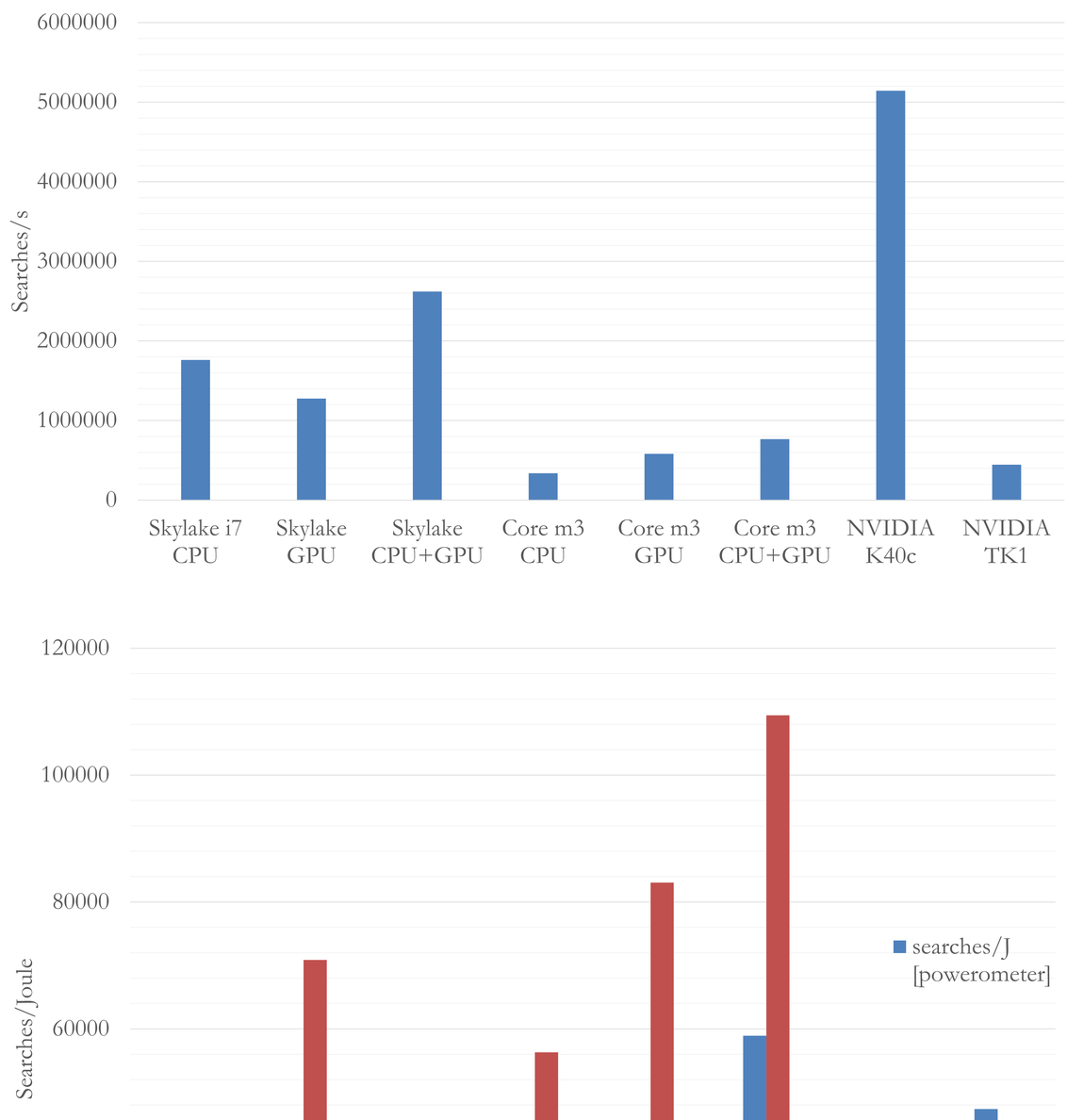
Test configuration:

3D Cloud of 500k points

Searching for points inside a box around each point

Density of points ~32 avg points inside each search box





Conclusion

Frequency is one of the dominating factors in high throughput computing

Higher frequency need improved cooling systems and decrease density

Exploiting unused SoC resources now possible and "easy"

Allows to achieve higher energy efficiency, throughput and latency

Useful to offload parallel friendly C-kernels

Bundesministerium für Bildung und Forschung