sarmap
your information gateway

Computationally efficient and operational solutions for the processing of very large SAR datasets based on modern multicore and GPGPU technologies

Achille Peternier, Marco Defilippi,
Alessio Cantone and Paolo Pasquali

sarmap
your information gateway

Remote Sensing: the big data issue

- New sensors such as ALOS-2, Sentinel-1:
  - High resolution, large files (e.g., S1A SLC ~8 GB).
  - Short-revisit time.
  - High availability (e.g., ESA SciHub).

- New algorithms and techniques:
  - Multi-temporal Analysis (SBAS, PSInSAR):
    - Using dozens of high-resolution images.
  - Larger series of images.

⇨ **Big data requires big computational power**

sarmap
your information gateway

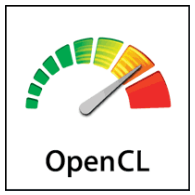
Remote Sensing: the big data issue

- What technology to use (from a SME point of view)?
  - Single-core, sequential code is a dead-end.
  - Multicore CPUs provide a high level of parallelism.
  - GPUs provide an even higher level of parallelism.

- Many (industrial) constraints:
  - Software must run on a wide range of heterogeneous customer machines:
    - Windows/Linux.
    - From laptops to high-end workstations and server-class computers:
      - From one single CPU up to multiple CPUs/GPUs.
  - Maintenance and development costs in line with a SME.

sarmap
your information gateway

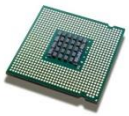
OpenCL



- After investigating most of the available options (i.e., OpenMP, TBB, C++AMP, CUDA and derivate projects), we decided to use OpenCL *(Peternier et al., APSAR 2013)*.
- **Open C**omputing **L**anguage.
- First specification released in December 2008:
  - Current version 2.1 (but effectively supported up to version 1.2).
- Maintained by the non-profit Khronos Group:
  - Khronos is the entity behind first-class standards like OpenGL, COLLADA, WebGL, etc.
- Supported by most of the key hardware and software manufacturers.

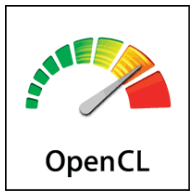
sarmap
your information gateway

OpenCL



sarmap
your information gateway

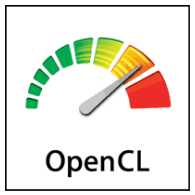
OpenCL



- "Write once, run everywhere" philosophy.
  - Runs on CPUs, GPUs, accelerators, FPGAs, etc.
    - OpenCL kernels are written in a C-like language and are compiled on-the-fly on the target machine using all the available optimizations it provides.
  - Code is portable, performance isn't:
    - …but reasonable performance is usually achieved by writing optimization-neutral code.
- Writing parallel algorithms is still difficult, but APIs and code-portability are now much easier.

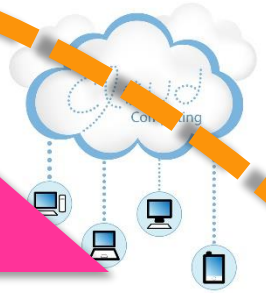
sarmap
your information gateway

OpenCL



- Mature ecosystem:
  - Documentation, tutorials, examples, books, …
  - Development tools (debuggers, profilers, ...)
  - External libraries (BLAS, FFT, abstraction layers, ...)
- Limitations:
  - Moving data to/from system/OpenCL is expensive.
  - OpenCL memory is a fraction of the system memory.
  - Requires an OpenCL platform to be first installed on the target system:
    - No Windows XP…
    - …but there are CPU-only versions that run even within virtual machines.

sarmap
your information gateway



DISTRIBUTED COMPUTING

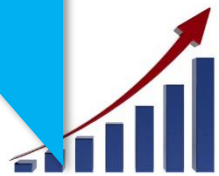
$$\langle \bar{R}_N + \frac{Z_N}{2} n_t | \hat{\rho}(t) | \bar{R}_N - \frac{Z_N}{2} n'_t \rangle$$
$$= \sum_{n_0, n'_0} \int d\bar{R}_0 dq_0 dp_0 dq'_0 dp'_0$$
$$\times \frac{1}{4} (q_{n_t} + ip_{n_t})(q_{n_0} \quad ip'_{n'_t})(q'_{n'_0} + ip'_{n'_0})$$
$$\times \int \prod_{k=1}^{N-1} d\bar{R}_k$$
$$\times \prod_{k}^{N-1}$$

NEW ALGORITHMS AND MORE EFFICIENT CODE

next-gen SARscape

MULTICORE, GPGPU AND PARALLEL COMPUTING

OpenCL



sarmap
your information gateway

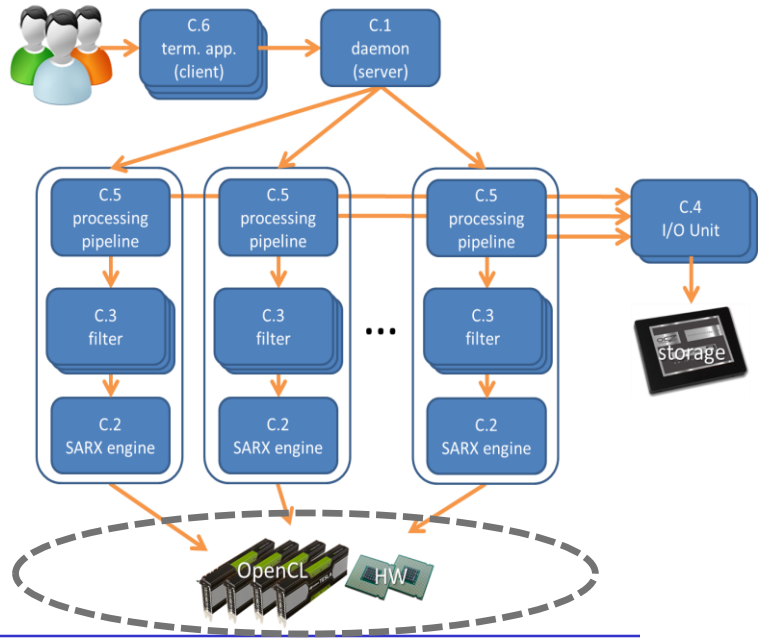
OpenCL ENVISAT focusing

- The SARIPA project (**SAR**scape **I**mage **P**rocessing **A**ccelerator): focusing of ENVISAT and Sentinel-1 images in realtime (< than the acquisition time).
- Migrate the SARscape ENVISAT and Sentinel-1 focusing pipeline to OpenCL.
- Target machine: 2x Intel Xeon 8 core CPUs, 64 of RAM, 4x Nvidia Tesla K20 GPUs, high performance SSD HD.

sarmap
your information gateway

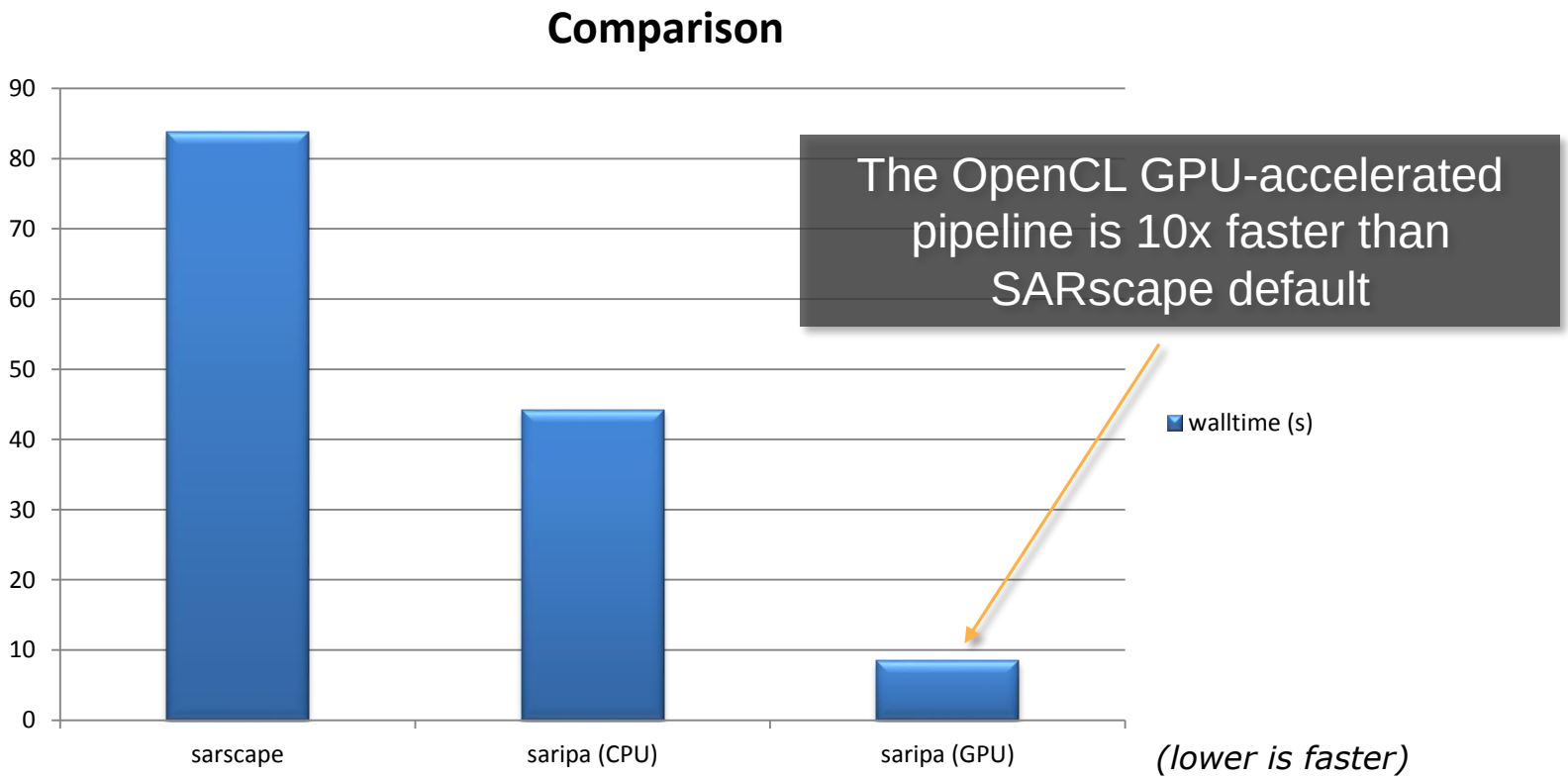
OpenCL ENVISAT focusing

- SARIPA is a compact local-cluster abstraction layer on top of OpenCL, Boost, and in-house libraries.
- Automatic features:
  - Multiple parallel pipelines.
  - Hardware awareness:
    - Hardware discovery and auto-configuration, NUMA nodes, etc.
  - Self-tuning (e.g., according to available memory and number of cores).
  - Deferred I/O operations.



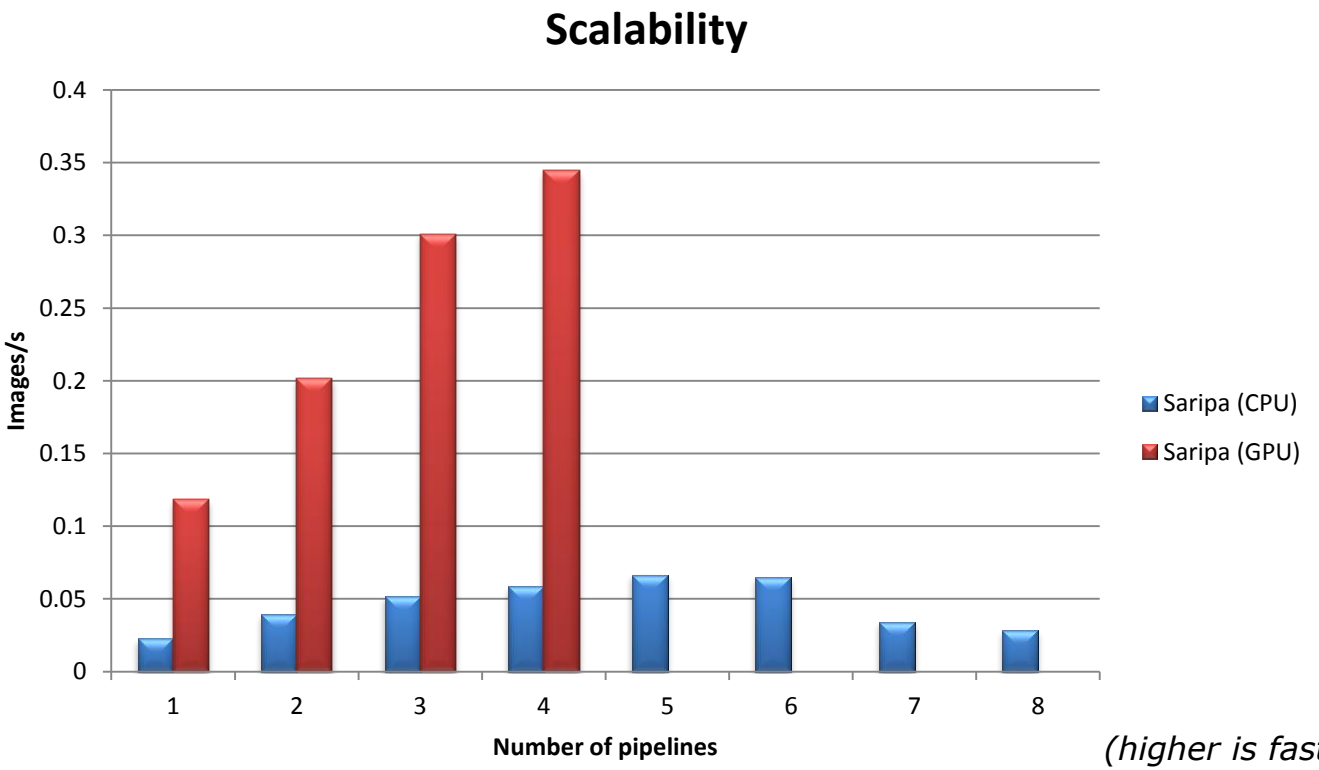
OpenCL ENVISAT focusing

Focusing of one single image



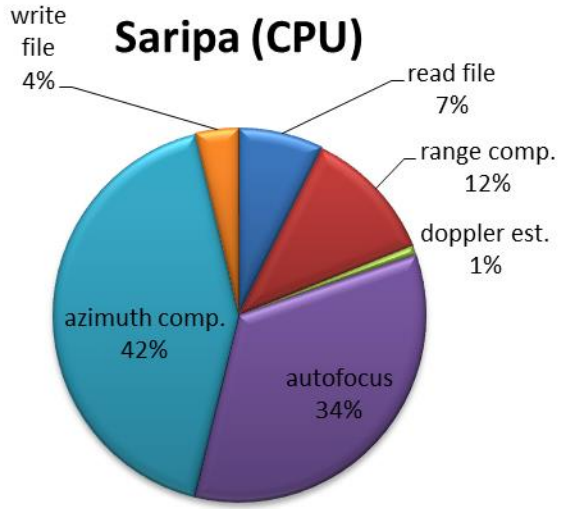
OpenCL ENVISAT focusing

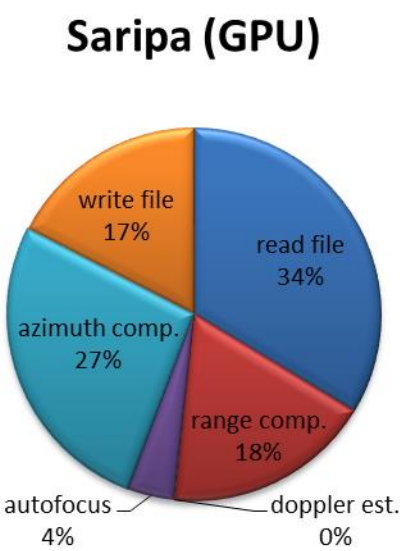
Focusing of multiple images



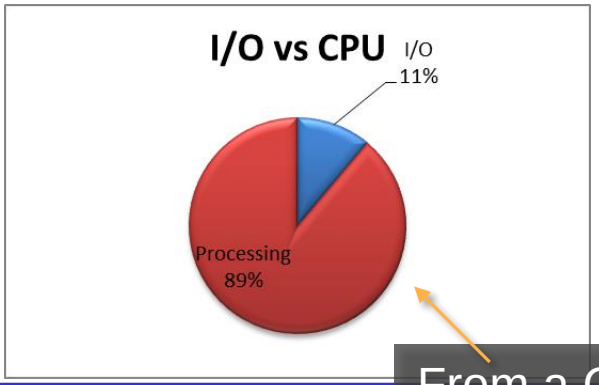
(higher is faster)

OpenCL ENVISAT focusing

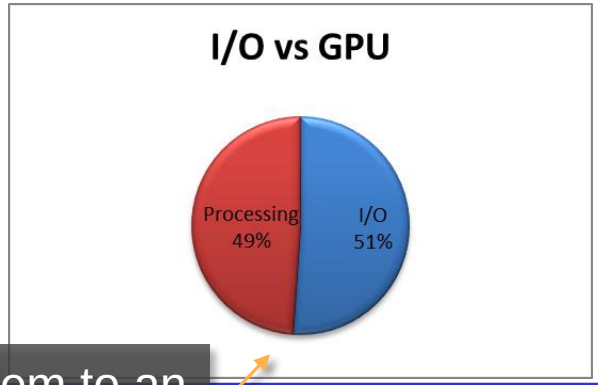




(tot. time = 44.25 s)



(tot. time = 8.45 s)



From a CPU-bound problem to an I/O-bound problem

sarmap
your information gateway

OpenCL + Cluster PS

- Algorithm considered:
  - Computing the Persistent Scatterer (PS) on a stack of 30 COSMO Skymed images using the distributed (cluster) version of SARscape.
  - Each physical node runs a local instance of the SARIPA, which can have one or more pipelines (NxM total logical nodes).

- Target cluster configuration of 4 physical nodes:
  - Each node has 2x Xeon 8 core CPUs, 128 GB of RAM.
  - 2 nodes have 4x Nvidia Tesla K80, 2 nodes have 2x Nvidia Tesla K20.
  - We assigned one logical node to each GPU, since each physical node has between 2 and 4 GPUs each.

sarmap
your information gateway

OpenCL + Cluster PS

- The test has been executed on a window of 5000x10000 pixels during the first inversion step of the PS:
  - The first inversion step is particularly suited for GPU performance evaluation since it pushes a lot of stress on OpenCL and takes a significant advantage of a higher number of cluster nodes.
  - Peak data size of about 400 GB (including temporary files, multilooking, etc.).

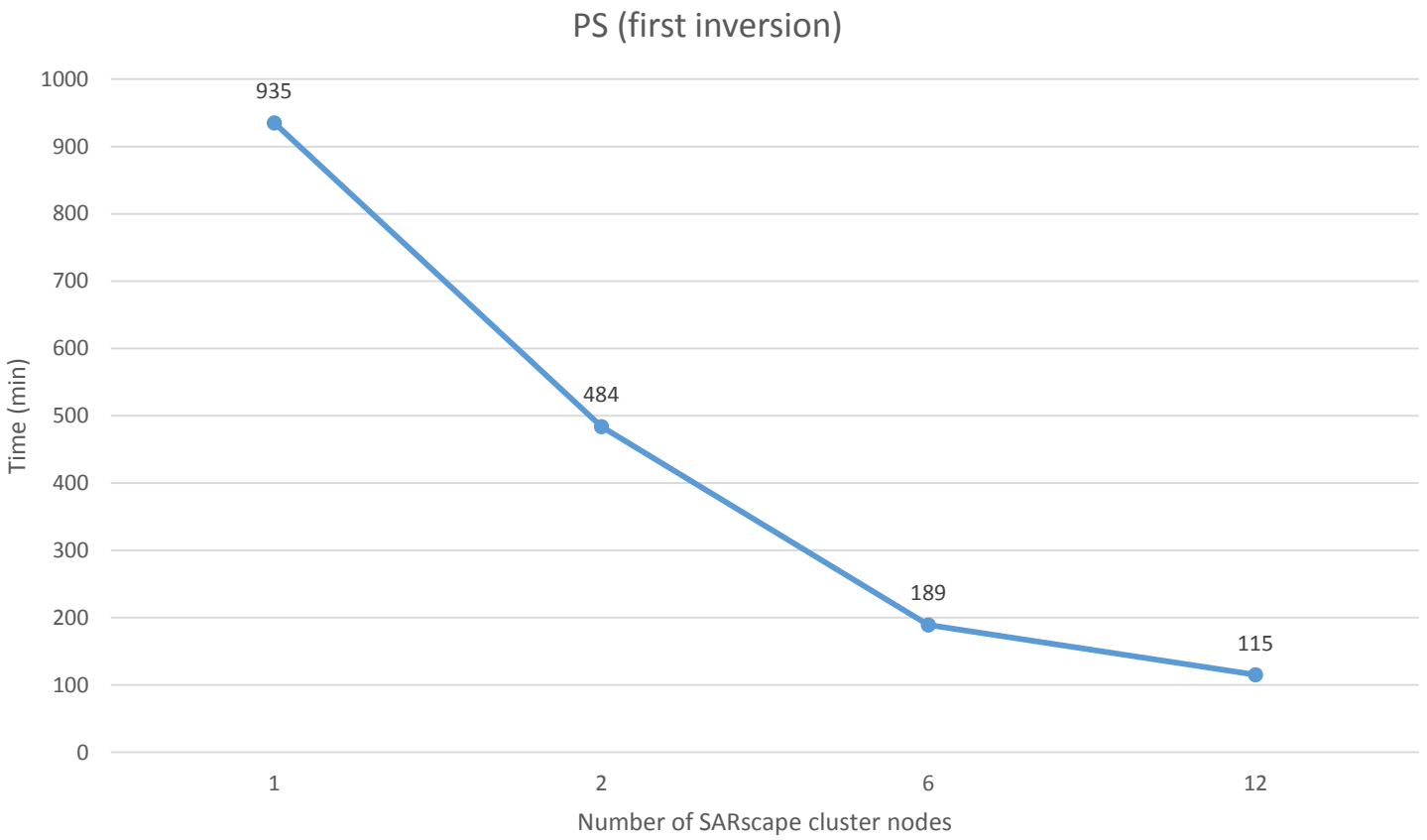
sarmap
your information gateway

OpenCL + Cluster PS

- The test has been repeated using different numbers of SARIPA pipelines (from 1 to 12) over one or more physical nodes to evaluate the speedup and scalability of the performance.

NxM	Nr. of physical computer nodes	Nr. of SARIPA pipelines	Nr. of GPUs per computer node
1x1	1	1	1 (K80)
2x1	2	2	1 (K80)
2x3	2	6	3 (K80)
2x4 + 2x2	4	12	4 (K80), 2 (K20)

OpenCL + Cluster PS



(lower is faster)

sarmap
your information gateway

OpenCL + Cluster PS

- Our software takes advantage of multiple computational nodes by distributing tasks to available logical nodes:
  - Using 12 nodes can speedup the processing by about one order of magnitude (from more than 15 hours to less than 2 hours).
  - I/O operations are distributed among multiple physical nodes to reduce local contention.
  - But additional care is required to balance the load over available nodes (mainly in the heterogeneous deployment used in the last case with NxM = 12).

sarmap
your information gateway

Conclusion

- GPGPU works at best when you can migrate/refactor large portions of the software processing pipeline to run on OpenCL.
- GPU-based SAR processing provides a significantly faster performance when compared to high-end alternate options such as dual-CPU computers.
- Clustering is still a valid option when a massive amount of data is involved:
  - Local processing is still improved through GPGPU.
  - Multiple machines reduce the stress to the local I/O system (hard disks, PCIE bus, etc.).
  - Thanks to GPUs and (now) less expensive 10 GBit networking devices, SME-sized clusters are a viable cost- and performance-efficient solution.

sarmap
your information gateway

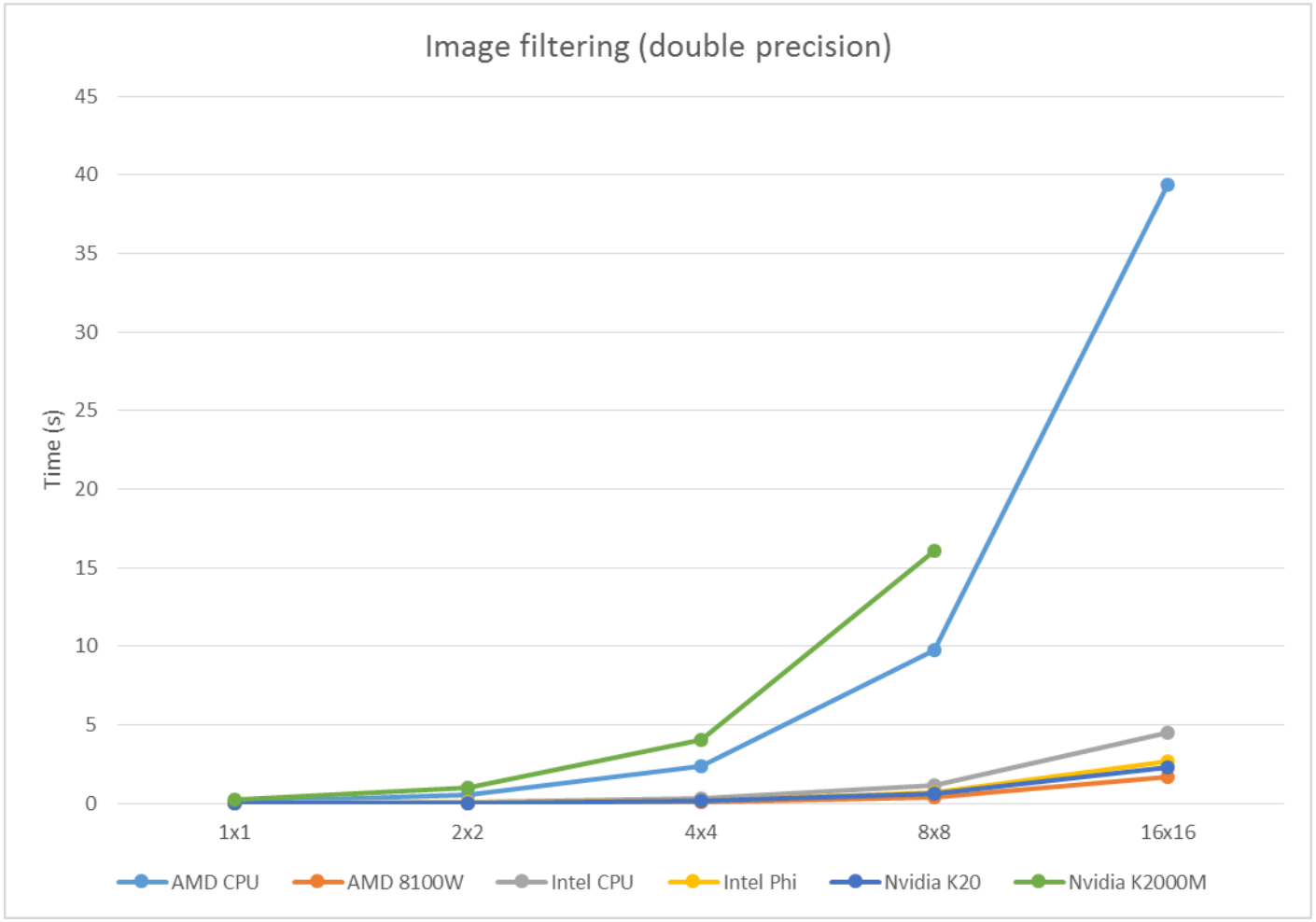
Thanks!

OpenCL benchmark

- Synthetic test based on the *sarxtest* application:
  - *sarxtest* is part of sarmap's SarxTools for testing the underlying hardware OpenCL computational power.
  - *sarxtest* is conceived for evaluating the OpenCL implementation speed by using a series of common SAR-imaging algorithms (e.g., Fast-Fourier Transform and window filters).
  - SarxTools are available at: http://sarmap.ch/ocl/opencl-install.html

sarmap
your information gateway

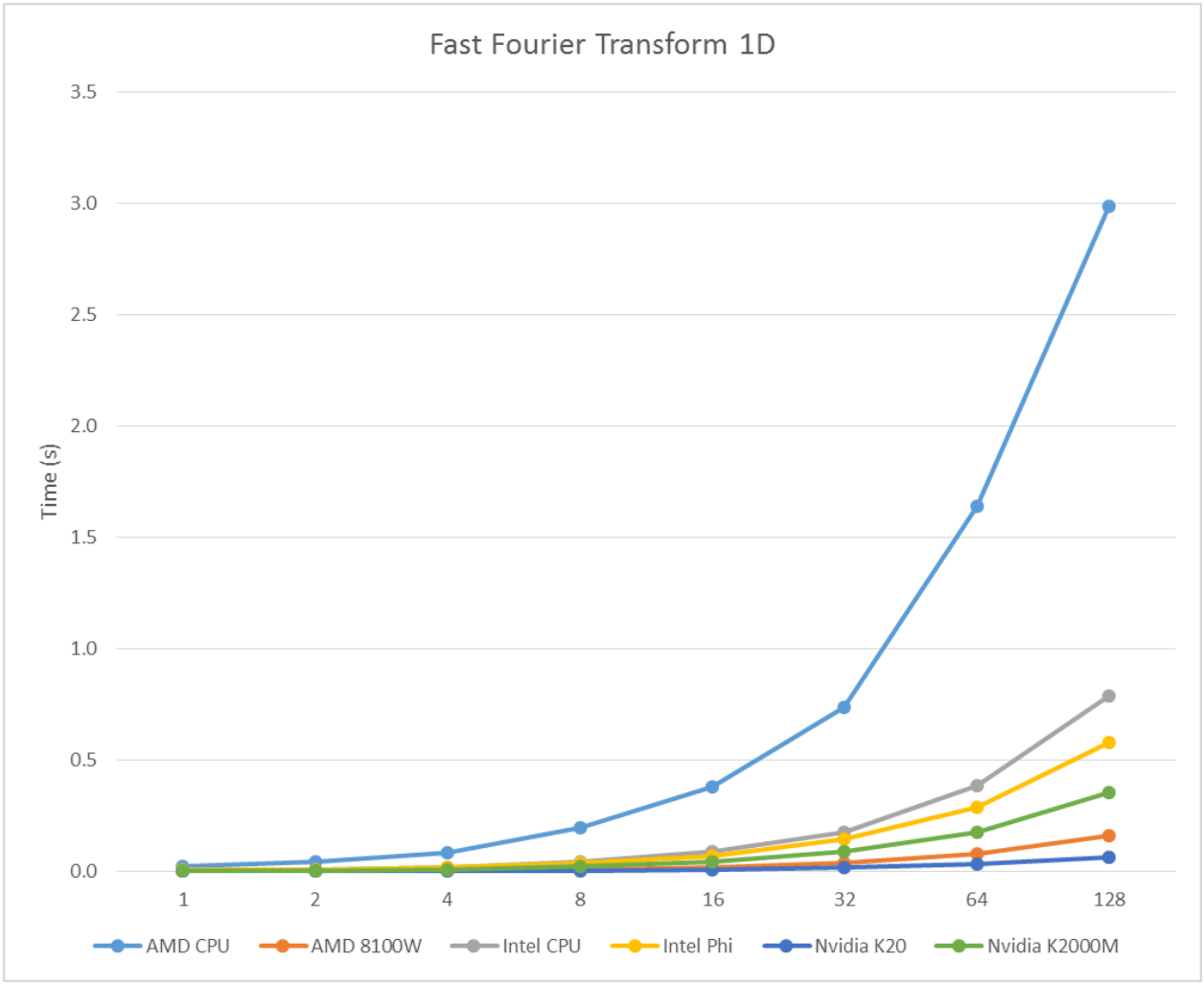
OpenCL benchmark



(CPU tests executed on 2x Intel Xeon 8 core CPUs, 64 GB of RAM) *(lower is faster)*

sarmap
your information gateway

OpenCL benchmark



(lower is faster)