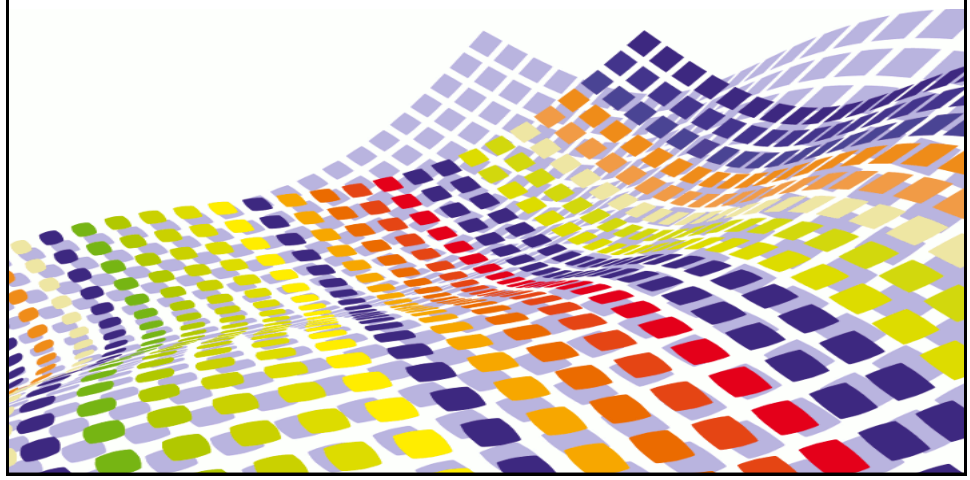
HMPP Origins

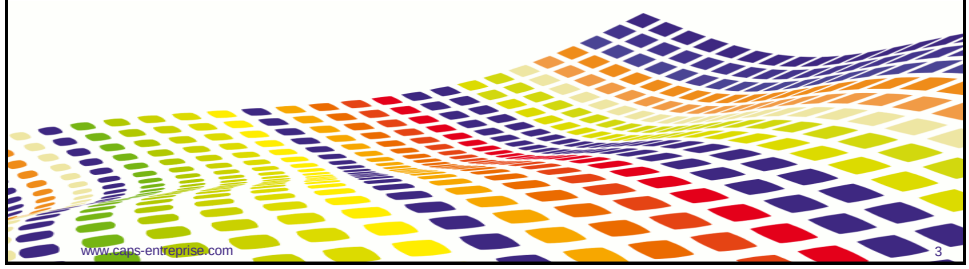


Agenda

- GPGPU & Hybrid Parallel computing
- HMPP Concepts and Overview
- Starting with HMPP
- Addressing hardware accelerators with HMPP
- HMPP Toolchain
- HMPP Runtime
- Managing Data with HMPP
- Grouping Codelets
- Sharing Data with HMPP
- HMPP Features & Roadmap

CAPS

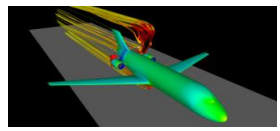
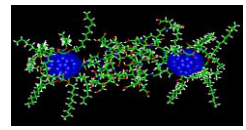
GPGPU & Parallel Hybrid Computing



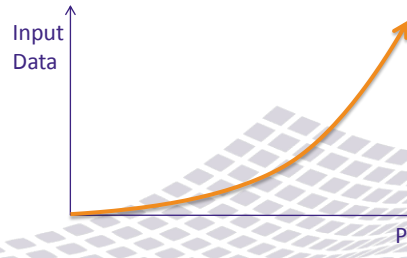
Industry and Business Facts

CAPS

- Modeling & Simulation are pervasive

- Precision is the key to success



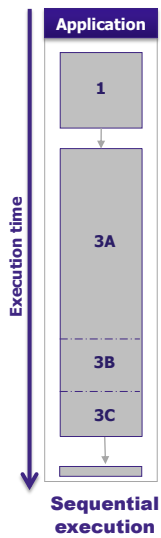
Why Hybrid Computing?

CAPS

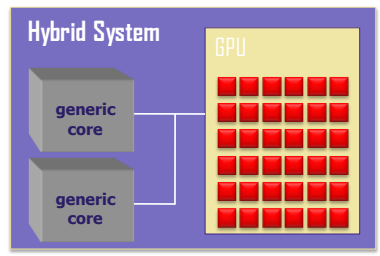
- More precision implies more data to process
  - But still in the same time, so more speed in computations!
- Current technologies reached a limit (in terms of frequency)
- One solution is to use parallelism (increase # of cores)
  - We do more things in parallel instead of doing it quicker
- The efficiency scale evolves according economic issues
  - Performance / Power consumption
  - Performance / Development time
  - …
- Mainstream applications will rely on these multicore / manycore architectures
- Various heterogeneous hardware
  - General purpose cores
  - Application specific cores (e.g. GPUs)

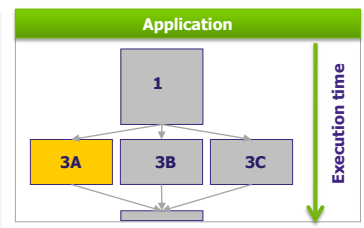
You said Parallel Hybrid Application?

CAPS



Sequential execution



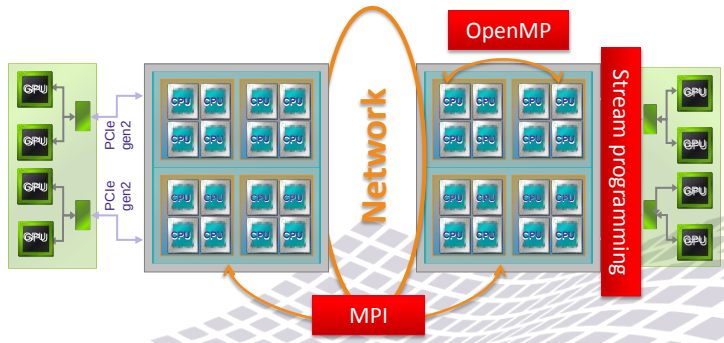


Parallel and distributed execution

Multiple Parallelism Levels

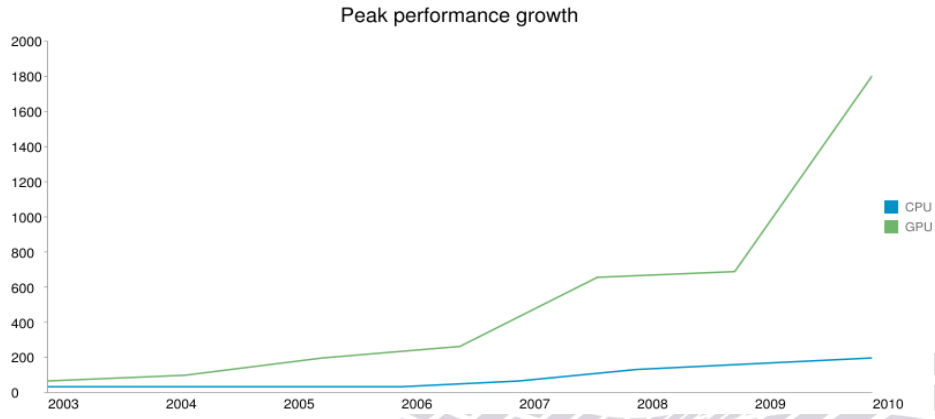
CAPS

- Adding a new layer of specific hardware is adding a new workload to the developer
- Programming various hardware components of a node cannot be done separately



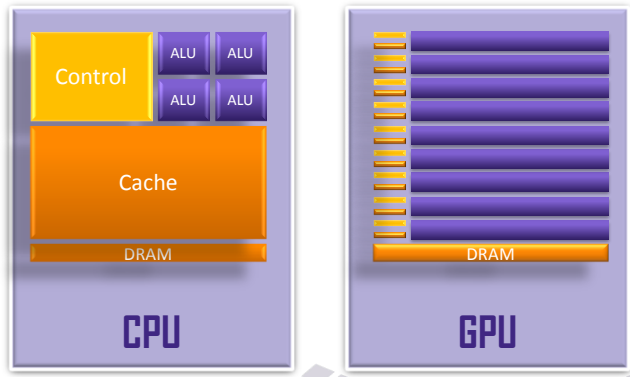
Parallel Processor Architectures

CAPS



Parallel Processor Architectures

CAPS



General purpose architecture

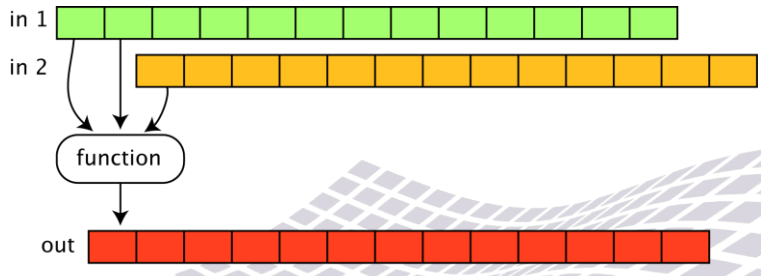
Massively data parallel

Needs 1000s of computation threads to be efficient

Stream computing

CAPS

- Stream programming is well suited to GPU
  - But memory hierarchy is exposed
- A similar computation is performed on a collection of data (*stream*)
  - There is no data dependence between the computation on different stream elements



Offloading computations

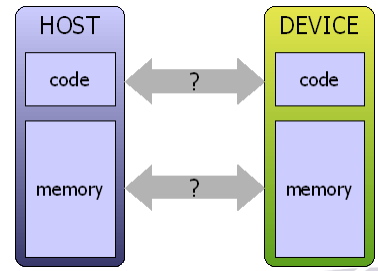
CAPS

- Host: General purpose cores
  - Share a main memory
  - Core ISA provides fast SIMD instructions
- Device: Streaming cores
  - GPU, DSP, FPGA… (vector, SIMD)
  - Application specific architectures
  - Can be extremely fast
- Hundreds of GigaOps
  - But not easy to leverage
  - Restriction to one platform is not acceptable

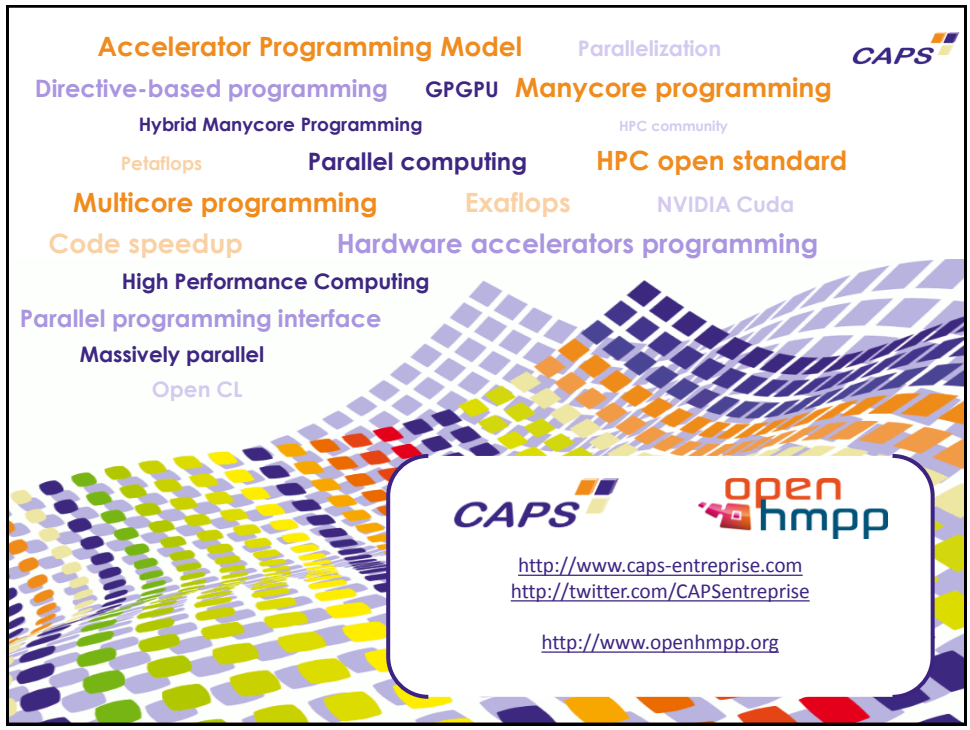
Key Issues

CAPS

- Using accelerators makes you stick to a proprietary language/environment/technology
- Huge potential performance but accelerators are far from host memory
  - Data must be copied on the remote device
  - Due to narrowband links between CPU/HWA, data transfers are critical



- Therefore rethink the computation organization/algorithm


Accelerator Programming Model
Parallelization
CAPS
Directive-based programming
GPGPU
Manycore programming
Hybrid Manycore Programming
HPC community
Petaflops
Parallel computing
HPC open standard
Multicore programming
Exaflops
NVIDIA Cuda
Code speedup
Hardware accelerators programming
High Performance Computing
Parallel programming interface
Massively parallel
Open CL
CAPS
open hmpp
http://www.caps-entreprise.com
http://twitter.com/CAPSentreprise
http://www.openhmpp.org