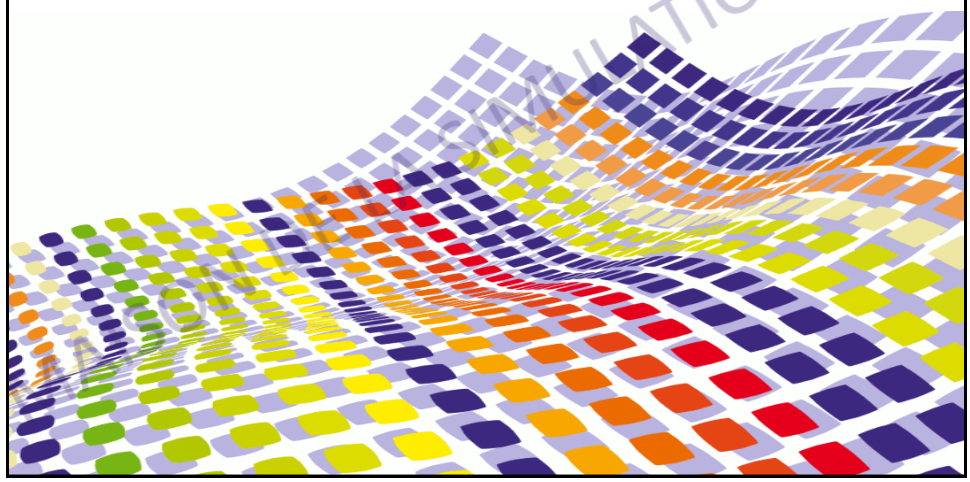
HMPP Origins



MAISON DE LA SIMULATION 2012

Agenda

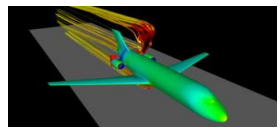
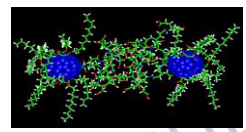
- GPGPU & Hybrid Parallel computing
- HMPP Concepts and Overview
- Starting with HMPP
- Addressing hardware accelerators with HMPP
- HMPP Toolchain
- HMPP Runtime
- Managing Data with HMPP
- Grouping Codelets
- Sharing Data with HMPP
- HMPP Features & Roadmap

MAISON DE LA SIMULATION 2012

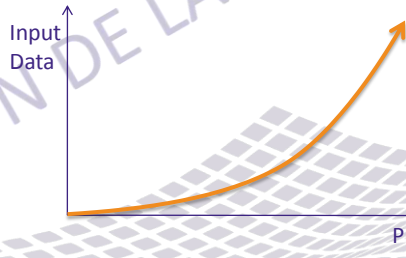
GPGPU & Parallel Hybrid Computing

Industry and Business Facts

- Modeling & Simulation are pervasive


- Precision is the key to success

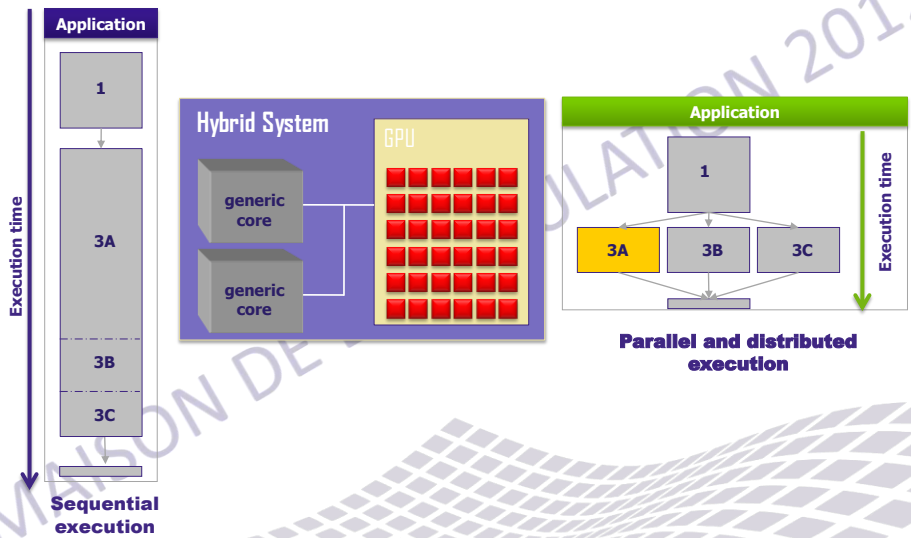


Why Hybrid Computing?

- 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)

MAISON DE LA SIMULATION 2012

You said Parallel Hybrid Application?

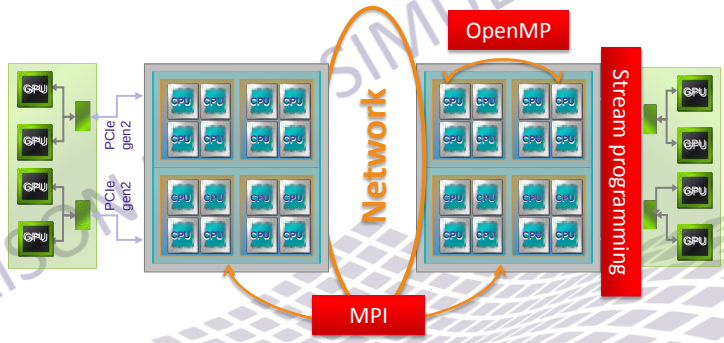


MAISON DE LA SIMULATION 2012

Multiple Parallelism Levels

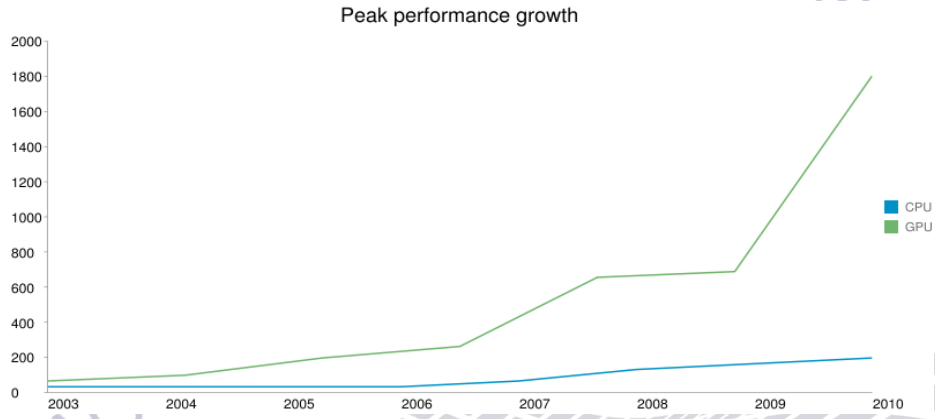
CAPS

- 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

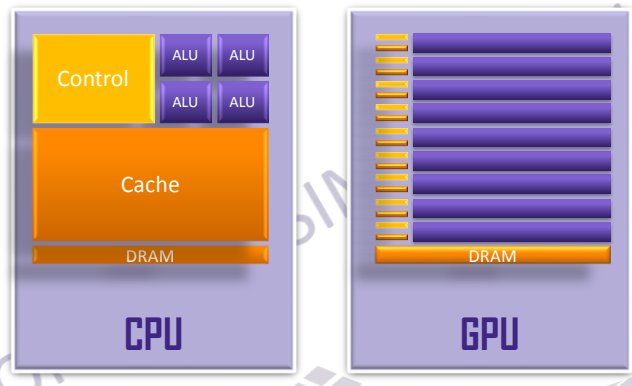


Parallel Processor Architectures

CAPS



Parallel Processor Architectures



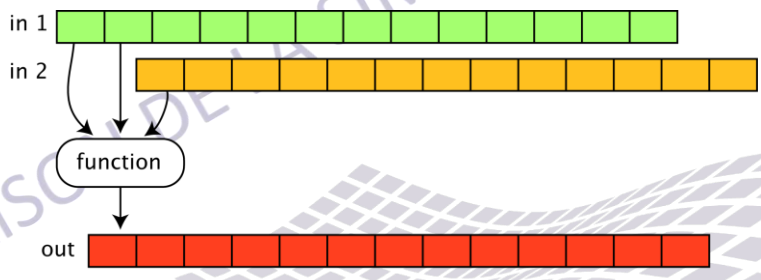
General purpose architecture

Massively data parallel

Needs 1000s of computation threads to be efficient

Stream computing

- 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



Current GPUs for GPU Computing

CAPS

- NVIDIA GeForce 200, for instance GTX 285
  - Max device memory size is 4 GB
  - Peak FP SP is 1 teraflops
  - Peak FP DP is 0.1 teraflops
  - Memory bandwidth is 159 GB/s
  - Power consumption 204 Watts
  - CUDA and OpenCL
- AMD Firestream, for instance Radeon HD 5870
  - Max device memory size is 2 GB
  - Peak FP SP is 2,5 teraflops
  - Peak FP DP is 0.5 teraflops
  - Memory bandwidth is 153 GB/s
  - Power consumption 228 Watts
  - OpenCL and CAL/IL

MAISON DE LA SIMULATION 2012

Offloading computations

CAPS

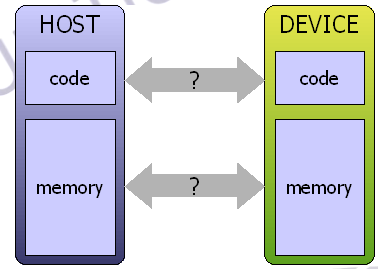
- Host: General purpose cores
  - Share a main memory
  - Core ISA provides fast SIMD instructions
- Device: Streaming cores
  - GPU, DSP, FPGA… (vector, SIMD)
  - Application specific architectures ("*narrowband*")
  - Can be extremely fast
- Hundreds of GigaOps
  - But not easy to leverage
  - Restriction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

MAISON DE LA SIMULATION 2012

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

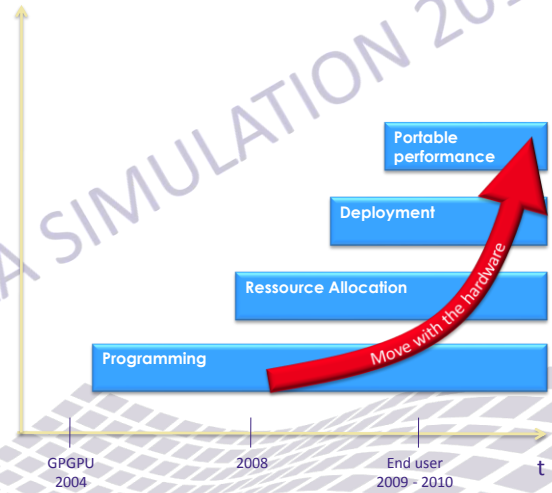


MAISON DE LA SIMULATION 2012

Manycore Challenges

CAPS

- Programming
  - Medium
- Resources management
  - Medium
- Application deployment
  - Hard
- Portable performance
  - Extremely hard



MAISON DE LA SIMULATION 2012

Why hybrid/heterogeneous may be unavailable

- Share some goals with embedded systems that lead to the same technological issues in HPC
  - Optimizing energy consumption leads to specialized architectures
- No common programming API
  - APIs always make some underlying architecture assumptions
  - No low level programming API common to all devices
  - An API mainly addresses a specific hardware component, as a consequence we need many

MAISON DE LA SIMULATION 2012

Accelerator Programming Model
Parallelization
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

http://www.caps-entreprise.com
http://twitter.com/CAPSentreprise

http://www.openhmpp.org