hmpp

Many-Core Programming Environment

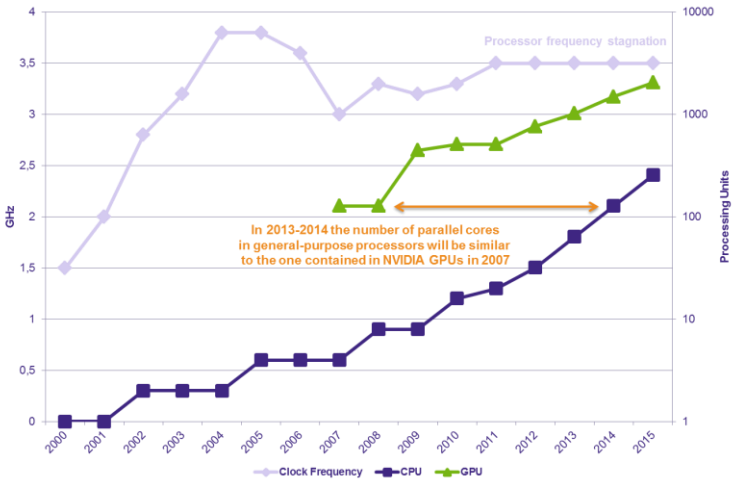
Break the Many-Core Wall

How to Write Code to Survive the Many-Core Revolution?

Pushed by the pace of innovation in the many-core technologies, including graphic processing units (GPUs), **the processor landscape is moving fast**.

As a consequence of the processor frequency stagnation, in **2013-2014**, **the number of parallel cores in general-purpose processors will be comparable to the number of cores contained in NVIDIA™ GPUs in 2007**.

By being a high level model, directive-based approaches like HMPP™ abstract the programming of many-core applications, keep them **hardware independent** and **ensure their portability** across new generations of hardware.



HMPP, a Directive-based Multi-language and Multi-target Programming Model

Based on a set of **OpenMP™-like directives** that **preserve legacy codes**, HMPP fully leverages the performance offered by most of today's stream processors. You keep your software **independent** from hardware targets **while preparing for future architectures** (Sandy Bridge, MIC, AMD Fusion, NVIDIA Denver…).

Complementary to **OpenMP** and **MPI™**, HMPP lets you **incrementally develop or port** existing applications to many-core without the complexity associated with many-core programming.

Pioneer in the directive-based approach with **its HMPP flagship product**, CAPS also delivers software development tools, solutions and expertise that help organizations to adapt the way their applications are developed in order to **benefit from the performance of many-core architectures**.

What You Get with HMPP

- With **one source code**, target **multiple many-core architectures**
- Distribute computation over CPU and GPU cores (**Multi-GPU**)
- High performance with optimized **data management**
- **Interoperability with libraries**
- Protect your software investment by using an **Open Standard**

From C/C++/FORTRAN,

Using your Compiler

intel · The Portland Group · GCC · absoft

to

NVIDIA CUDA · OpenCL

BE PART of a Worldwide **Ecosystem**

hmpp Competence Center

open hmpp

A Directive-based Programming Model for C/C++/FORTRAN

HMPP directives are **meta-information** added in the application source code that **do not change the semantic of the original code**. They address the **remote execution** (RPC) of functions or regions of code on GPUs and many-core accelerators as well as **the transfer of data** to and from the target device memory.

HMPP offers an **incremental way** of migrating applications by first **declaring** and **generating** kernels of critical computations, then by **managing data transfers** and finally by **optimizing kernel performance and data synchronization**.

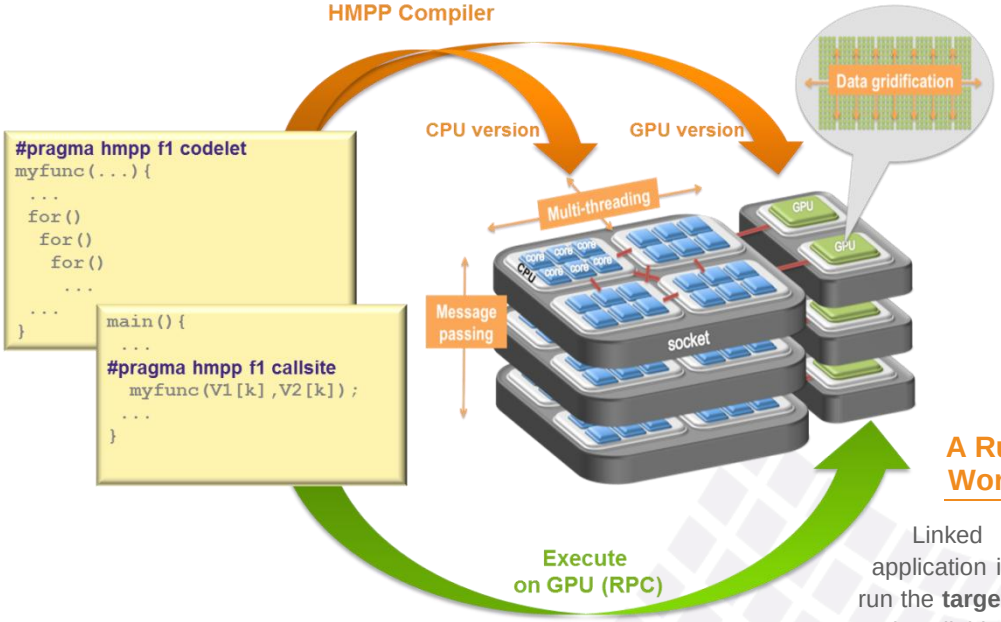
A Source-to-source Compiler with CUDA and OpenCL Back-ends

HMPP is **complementary to existing parallel APIs** (OpenMP or MPI).

From the HMPP annotated application, HMPP **separately compiles the native host application and the GPU accelerated codelet functions** as software plugins so that CPU code **does not require any compiler change**.

The codelets are translated in **NVIDIA CUDA** and **OpenCL** languages by the HMPP back-ends and compiled with the hardware vendor tools, leveraging existing SDKs.

HMPP offers a mechanism to obtain **interoperability between user's code and highly optimized libraries** such as cuFFT…



10 REASONS to Use Directives

- Preserve serial code
- Incremental programming reduces risk and cost
- Do not commit to a hardware platform-specific dialect
- Use a standardized approach (e.g. OpenHMPP)
- Focus on highlighting parallelism, not its implementation
- Do not require changing the coding base language
- Avoid mixing languages
- Allow to keep a unique source code
- Do not create dependencies on a runtime library
- Ease interaction between application and computer science people

A Runtime to Manage Data & Workload Distribution

Linked with the HMPP runtime, the native host application is able to **execute standalone** or to load and run the **targeted codelet libraries** when GPUs are present and available.

HMPP runtime scales to **multi-GPUs** and **is free of use.**

Supported Platforms and Compilers

Linux

Compilers:
- Intel 11.1+
- GNU gcc 4.1+
- GNU gfortran 4.3+
- Open64 4.2+
- PGI 10.0+
- SUN 12.1+
- Absoft Pro Fortran 11.0+

Operating Systems:
- Debian 5.0+
- RedHat Entreprise
- Linux 5.3+
- OpenSuse 11.x
- SLES 11.0
- Ubuntu 9.10 & 10.04

Windows

Compilers:
- Visual Studio 2008 SP1 IDE
- Absoft Pro Fortran 11.1.2+

Operating Systems:
- Microsoft Windows HPC Server 2008 R2
- Windows 7

With All NVIDIA® Tesla™ and AMD-ATI FireStream™

Headquarters - FRANCE
Immeuble CAP Nord
4A Allée Marie Berhaut
35000 Rennes
France
Tel.: +33 (0)2 22 51 16 00
info@caps-entreprise.com

CAPS - USA
4701 Patrick Drive Bldg 12
Santa Clara
CA 95054
Tel.: +1 408 550 2887 x70
usa@caps-entreprise.com

CAPS - CHINA
Suite E2, 30/F
JuneYao International Plaza
789, Zhaojiabang Road,
Shanghai 200032
Tel.: +86 21 3363 0057
apac@caps-entreprise.com