

The programming glue between the legacy code and hardware-accelerated compute kernels

The CAPS HMPP™ Workbench enables developers to easily and rapidly integrate hardware accelerators in existing C and Fortran source code and build powerful manycore applications.

Heterogeneous processing such as CPU-GPU or CPU-FPGA hybrid processors can deliver power efficiency and performance-per-watt. On streaming computations the GPU has 20 to 50 times better performance. HMPP™ enables to seamlessly integrate these accelerating devices without having to port the applications.

Based on a set of compiler directives, HMPP™ comes with development tools and Dynamic Services to simplify the use of hardware accelerators in conventional general purpose software applications.

The directives preserve the legacy code and keep the application code portable. The hardware-assisted versions of the computations, named codelets, are built as software plugins using the hardware vendor programming tools. No need to compile a machine specific version of the application, at runtime, HMPP™ Dynamic Services take care to discover the available hardware accelerators and load the corresponding codelet plugins.

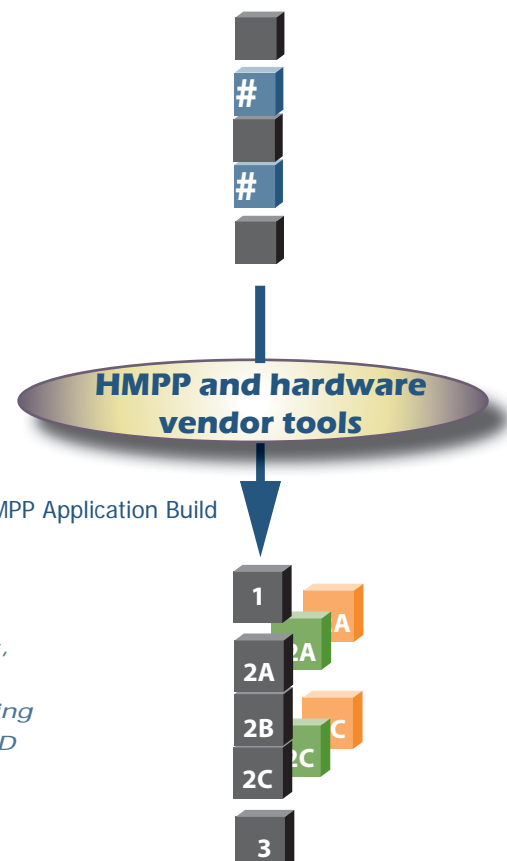
How does HMPP™ work?

Directives Use

- Declare the HMPP™ codelets, the hardware assisted implementations of critical functions.
- Indicate the asynchronous remote execution of the codelets.
- Place synchronization barriers (execution, data transfers).

HMPP™ gives programmers a simple, flexible and portable interface for developing parallel applications whose critical computations are distributed over specialized and heterogeneous cores.

HMPP™ unleashes the performance of applications, reduces their development time and cost while increasing efficiency and return on R&D investment.

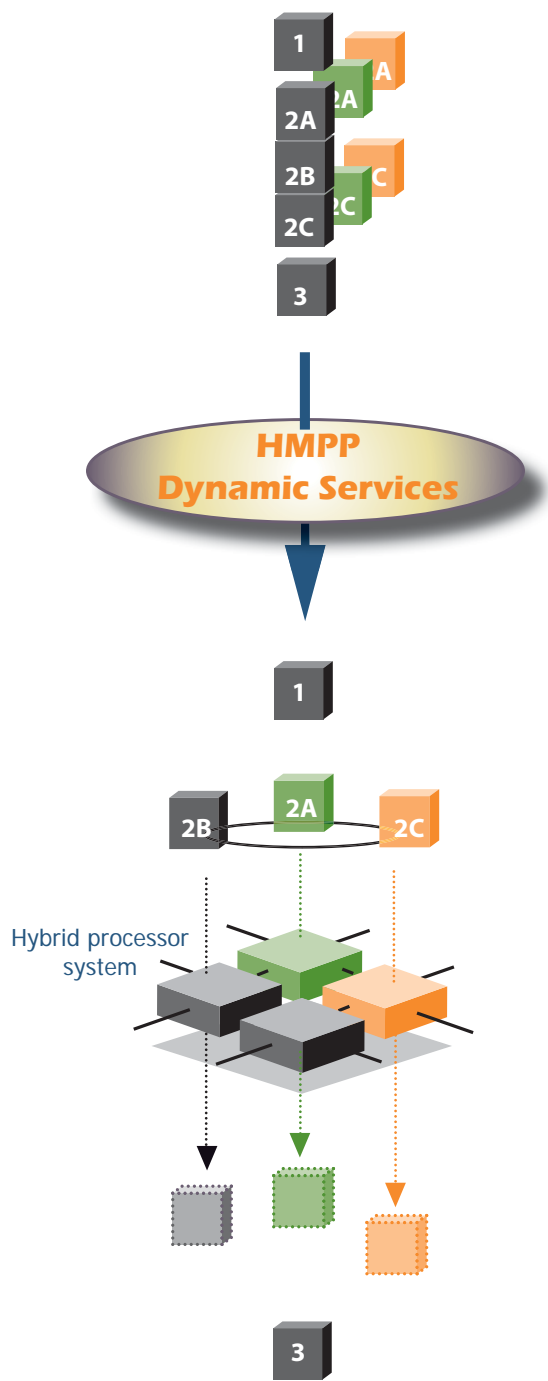


■ PORTABILITY

- Preserve the legacy code
- Keep independent from the hardware platform
- Do not lock to a vendor specific API
- Provide interoperability
- Work with any C and Fortran optimizing compilers

■ SCALABILITY

- Distribute computation between CPU and accelerators
- Integrate with Open MP and MPI



By insulating hardware specific versions of functions from the native application source, HMPP™ preserves the legacy code, keeps the application independent from fast changing hardware platforms and offers interoperability.

Application Compilation

Through the use of the HMPP™ directives, the developer declares functions, named codelets, which are suitable for hardware acceleration. The directives also specify the conditional execution of the codelets, their desired synchronous or asynchronous properties and data transfers.

The HMPP™ annotated application is preprocessed by the CAPS tools to extract the different hardware versions of the codelets which are built using the hardware vendor programming tools.

Separately compile the codelets and the application using the vendor programming tools and the host compiler.

Application Execution

At execution, the HMPP™ Dynamic Services take care to discover the available hardware accelerators. When a codelet is indicated to be run on a hardware accelerator, if the device is available and if the shared library implementing the codelet is present, HMPP™ loads it just as a software plug-in.

Otherwise the native version is run on the host CPU or in a worker thread. It is not necessary to build a machine specific version of the application, as long as the host CPU is identical, the application will make use of the available accelerators.

HMPP™ is complimentary to OpenMP and MPI. You use the same binary that is run through MPI across several nodes with different accelerating devices. The HMPP™ Dynamic Services load the available local drivers and codelets.

CAPS is committed to working with hardware vendors to provide an efficient binding between HMPP™ and the provided third party compiling tools and runtime support.

Available targets & Supported tools and platforms

NVIDIA® CUDA™
AMD-ATI® BROOK+™
Posix pthreads

C and Fortran compilers
Linux platforms

