

Adaptation of a HPC system to FPGA

Georgios Christodoulis¹, François Broquedis¹, Olivier Muller²,
Frederic Desprez^{1,3}

LIG¹, TIMA², INRIA³

georgios.christodoulis@inria.fr

August 31, 2016



Heterogeneity in HPC

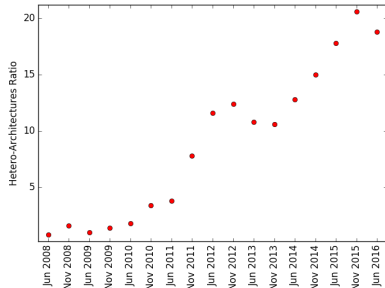


Figure: Heterogeneous Architectures, Top500

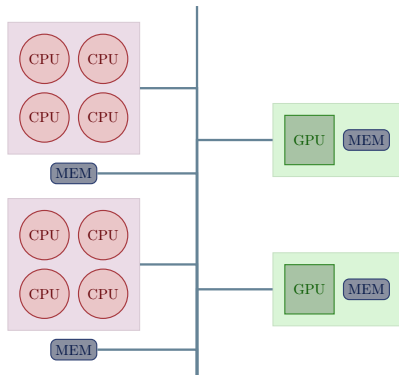
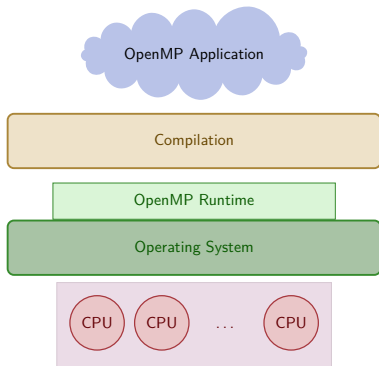


Figure: Accelerators in HPC

Standard application development in HPC

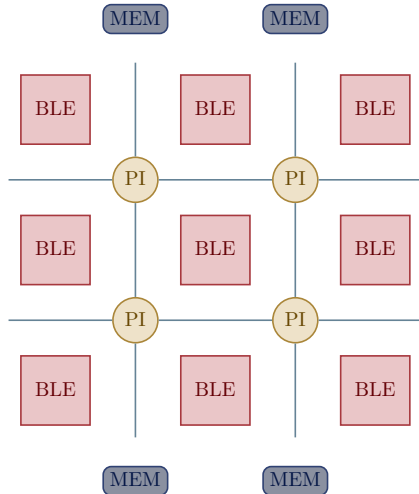
OpenMP is a **directive based** parallel programming environment for shared memory applications.



- ▶ Broadly used programming languages: C/C++, Fortran
- ▶ Relaxed-consistency shared memory model
- ▶ Fork-Join execution Model

OpenMP 4.0: **Accelerator** support

What an FPGA is



An FPGA is an array of:

- ▶ Logic Elements
- ▶ Interconnection Network
- ▶ Memory Elements

Motivation

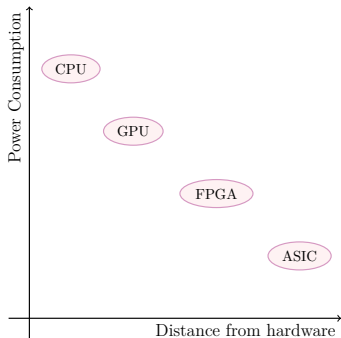


Figure: Energy efficiency associated with the level of abstraction

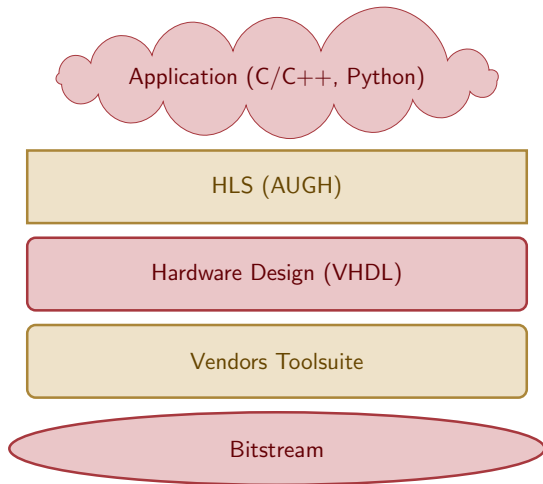
Pros:

- ▶ Energy efficiency
- ▶ Impressive acceleration

Cons:

- ▶ Programming in hardware level
- ▶ No programming model:
Complicated Integration

FPGA: Standard Design flow



Commodity frameworks in HPC

(multicore) OpenMP, TBB (accelerators) OpenCL, CUDA	Application	OpenCL C, C++, SystemC, VHDL, Verilog
--	--------------------	--

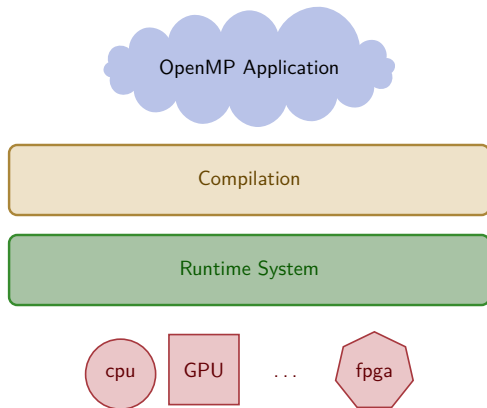
GCC, ICC, NVCC, PGI, IBM XL LLVM (CLANG)	Compiler	CatapultC, Vivado HLS AUGH
---	-----------------	-------------------------------

GOMP, IOMP, OmpSs, StarPU, Xkaapi	Runtime support	Proprietary or adhoc
--------------------------------------	----------------------------	----------------------

Heterogeneous Architectures: CPU, GPU, Xion Phi	Architecture	FPGA
--	---------------------	------

Our Horizon

Build a runtime Environment for complete support of FPGAs as accelerator for HPC applications.



With emphasis in:

- ▶ Portability
- ▶ Performance
- ▶ Application Development

Essential Tools

LLVM/ Clang

Using Clang we have:

- ▷ Compiler diagnostic tools
- ▷ OpenMP support
- ▷ Efficient Compilation Speeds

Compiler Level

Runtime Level

AUGH

AUGH can achieve:

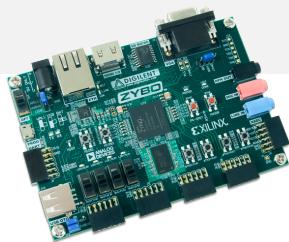
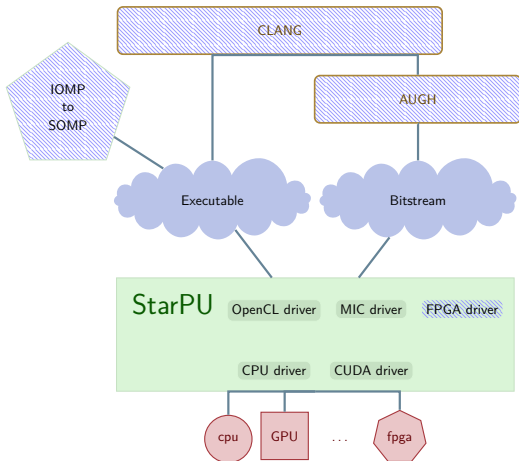
- ▷ Great speed on hardware design
- ▷ Hardware design under provided spatial restrictions

StarPU

StarPU can efficiently manage:

- ▷ Task dependences
- ▷ Heterogeneous scheduling
- ▷ Data transfers (Coherence-Consistency)
- ▷ OpenMP 3.0 support (somp)

Ongoing Development



Primitive Results:

- ▶ Compiler Cooperation (Clang + AUGH)
- ▶ OpenMP compatibility for basic directives (parallel, single, task)
- ▶ Successful Hardware task offloading