

Performance Portable

Offline Highlights

- "R&D exploiting GPUs gaining traction, even for the most complex workflows:
 - Montecarlo simulation (Adept, Celeritas),
 - Tracking (ACTS/traccc) and primary vertex reconstruction."
 - "Simulation on GPUs: AdePT" : géométrie spéciale, bibliothèque de physique speciale...
 - Reconstruction with ACTS/traccc : "GPU becoming competitive at high pileup".
-

Results from HEP-CCE

Hardware support for each portability layer

	CUDA	Kokkos	SYCL	HIP	OpenMP	alpaka	std::par
NVIDIA GPU			<i>intel/llvm compute-cpp</i>	<i>hipcc</i>	<i>nvc++ LLVM, Cray GCC, XL</i>		<i>nvc++</i>
AMD GPU			<i>openSYCL intel/llvm</i>	<i>hipcc</i>	<i>AOMP LLVM Cray</i>		
Intel GPU			<i>oneAPI intel/llvm</i>	<i>CHIP-SPV: early prototype</i>	<i>Intel OneAPI compiler</i>	<i>prototype</i>	<i>oneapi::dpl</i>
x86 CPU			<i>oneAPI intel/llvm compute-cpp</i>	<i>via HIP-CPU Runtime</i>	<i>nvc++ LLVM, CCE, GCC, XL</i>		
FPGA				<i>via Xilinx Runtime</i>	<i>prototype compilers (OpenArc, Intel, etc.)</i>	<i>prototype via SYCL</i>	

Ported representative testbeds from ATLAS, CMS and DUNE to each portability layer

/er.

	Kokkos	SYCL	OpenMP	Alpaka	std::par
Patatrack	Done	Done*	WIP	Done*	Done compiler bugs
Wirecell	Done	Done	Done	no	Done
FastCaloSim	Done	Done	Done	Done	Done
P2R	done	Done	OpenACC	Done	Done

Portable Parallelization Strategies: Recommendations

Software and hardware are still rapidly changing

- Lots of interactions with API developers in hackathons and to fix bugs
- Results remain preliminary

API recommendations are very application dependent

- All perform approximately equally for simple kernels
- Complex algorithms and chained kernels bring out weaknesses of all APIs
 - interaction with external libraries adds extra complexities
 - even compilation can be an issue

Learning curve / language complexity of APIs not all the same

- `std::par` → OMP → Kokkos / SYCL → alpaka → OMP
- subjective and dependent on code complexity and previous experience

Porting from Serial CPU code → GPU concepts is the biggest hurdle

- starting with optimized code is extra challenging

Very hard to extrapolate to next five years or beyond

- Vendors are pushing in different directions (but towards standards)
 - Increasing proximity of CPU / GPU / memory will have significant impact
-

Mon sentiment subjectif

- Utilisation accrue des GPUs, mais reste cantonnés dans des tâches spécifiques, à commencer par le online.
- Toujours compliqué à coder, donc ca reste entre les mains de spécialistes.
- Pas vu de code capable de fonctionner efficacement à la fois sur CPU et GPU.
- Pour la portabilité entre GPUs, Alpaka prend le dessus.
- SYCL maintenant assimilé à Intel, et pris en compte uniquement pour les GPUs Intel, présents et à venir.
- SYCL sur FPGA ?!?