DevDeck

Global Solution for Many-Core Programming

OLCF Spring Training and Users' Meeting 2012 – April 17, 2011

Vasnier Jean-charles – CAPS enterprise

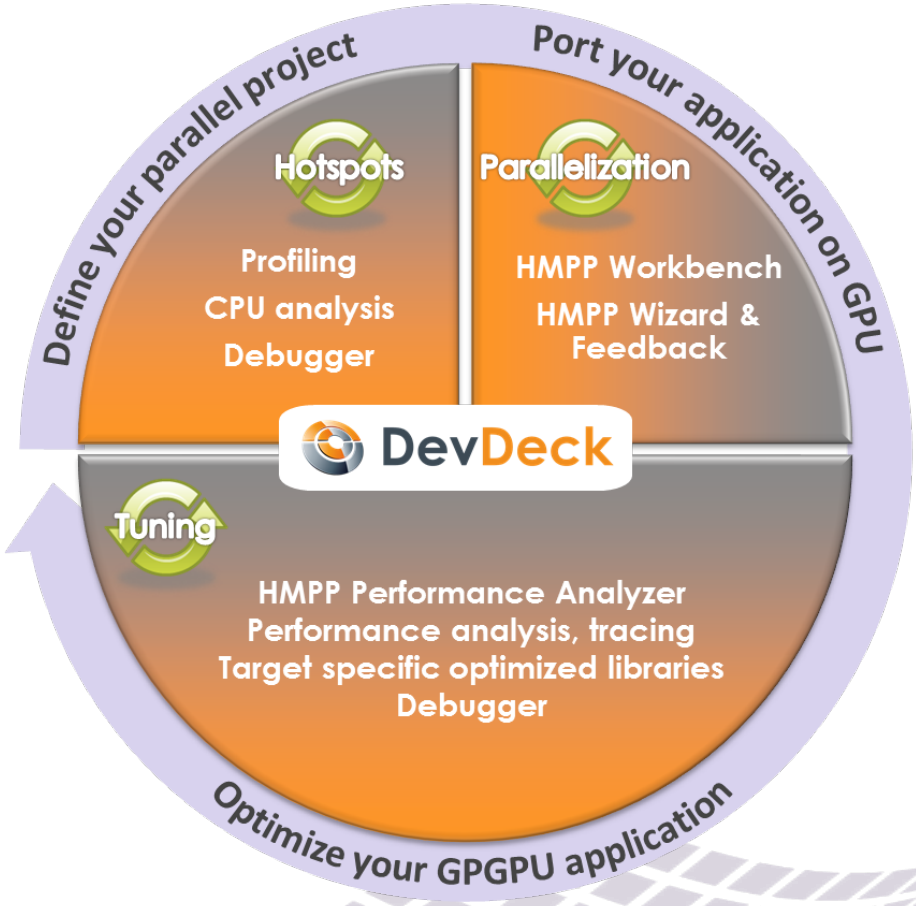
OAK RIDGE National Laboratory CAPS

What is DevDeck ?

A Global Solution for Many-Core Programming

- A single offer including different software development tools
- Based on the porting methodology developed by CAPS
- Involving partners to answer needs

CAPS porting methodology overview



What tools do we need?

- A profiler: to detect where to spend time effectively for code porting
- Optimized libraries: to take advantage of optimized resources already available on some architectures
- A debugger: to assist the development
- Tools to efficiently port the application to GPU

Provide the right tool at the right time

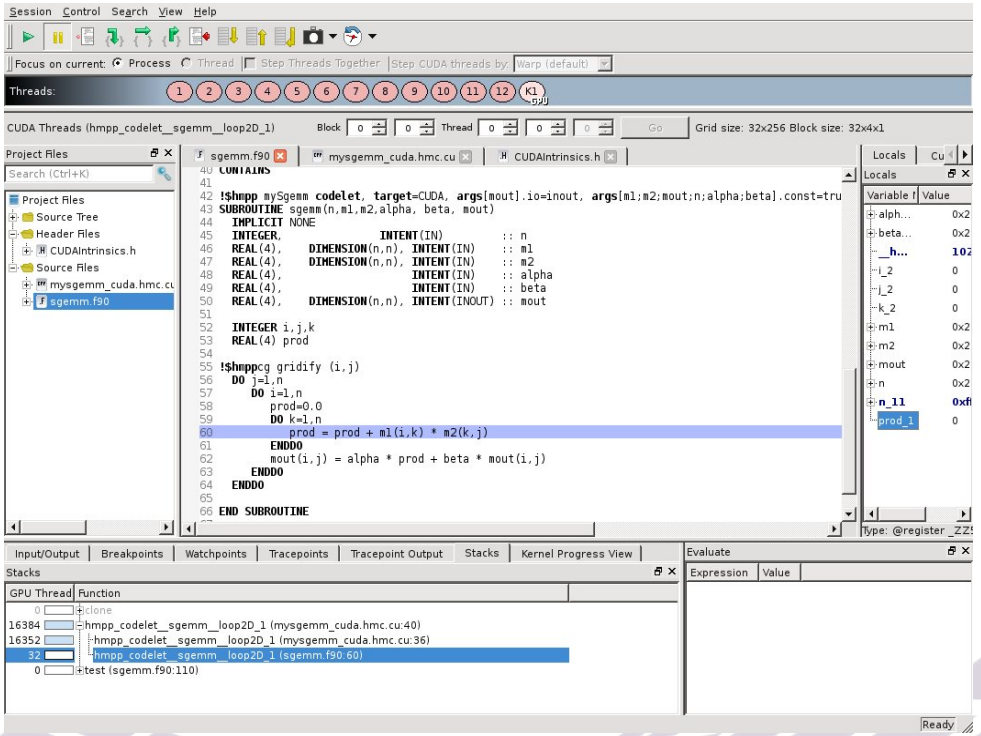
DevDeck: current configuration

CAPS: HMPP Workbench

- A **Directive**-based Programming Model for C/C++/ FORTRAN
  - They address the **remote execution** of functions or regions of code on many-core accelerators
  - And **the transfer of data** to and from the target device memory.
- A Source-to-source Compiler with CUDA and OpenCL Back-ends
  - Front-end: C/C++/Fortran (based on Intel, GNU, PGI, …)
  - Back-end: CUDA & OpenCL
- A Runtime to Manage Data & Workload Distribution
- Works with HMPP/HMPPCG directives
  - OpenACC directives (soon)

Allinea DDT – Debugging Tool for CPU/GPU

- Debug your kernels:
  - Debug CPU and GPU concurrently
- Examine thread data
  - Display variables
- Integrated with HMPP
  - Allows HMPP directives breakpoints
  - Step into HMPP codelets



CULA GPU-accelerated linear algebra library

OAK RIDGE National Laboratory CAPS

CULA | tools

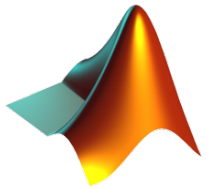
CULA is a GPU-accelerated linear algebra library that utilizes the NVIDIA CUDA parallel computing architecture to dramatically improve the computation speed of sophisticated mathematics. Used by thousands of developers and engineers worldwide, CULA is #1 in ease of use, functionality, and performance.

CULA R14 (CUDA 4.1)

- » Familiar LAPACK interface
- » C/C++ & FORTRAN
- » 170+ Routines
- » Cross platform support
- » Linux, Windows, Mac OS X

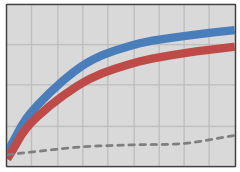
MATLAB Acceleration

- » Seamless integration with new link interface in R14



Supercomputer Speeds

Performance 7x of Intel's MKL LAPACK. To view performance charts: www.culatools.com



Accelereyes-accelerated linear algebra library

ArrayFire is a fast software library for GPU computing with an easy-to-use API. Its array-based function set makes GPU programming simple. ArrayFire is available for C, C++, Fortran, and Python and integrates with AMD, Intel, and NVIDIA hardware.

ArrayFire is simple. A few lines of code in ArrayFire can replace dozens of lines of raw GPU code, saving you valuable time and lowering development costs. No need to waste time with building block functions. Plug-in ArrayFire and jump start your code!

reductions
- sum, min, max, count, prod
- vectors, columns, rows, etc

dense linear algebra
- LU, QR, Cholesky, SVD, Eigenvalues, Inversion, Solvers, Determinant, Matrix Power

convolutions
- 2D, 3D, ND

FFTs
- 2D, 3D, ND

image processing
- filter, rotate, erode, dilate, morph, resize, rgb2gray, histograms

interpolate & scale
- vectors, matrices
- rescaling

sorting
- along any dimension
- sort detection

and many more…

Accelerated linear algebra library using HMPPAlt

- Accelerated library to improve the computation speed of sophisticated mathematics (FTT, LAPACK, …)
  - Take advantage of optimized libraries that already exist
  - Keep a unique version of the code for these library call

- Can be used with HMPP ALTernative directives
  - Bring interoperability between HMPP code and GPU libraries

Example of use with HMPP ALT Directives

```
…
! Regular call of the CPU version of sgeqrf
  CALL sgeqrf(M1, N1, A1, LDA1, TAU1, WORK1, LWORK1, INFO);
 …
  ! Implement sgeqrf using CULA for large matrices only
  !$HMPPALT CULA call, name="sgeqrf", cond="M*N>1000000"
  CALL sgeqrf(M2, N2, A2, LDA2, TAU2, WORK2, LWORK2, INFO)
…
```

CAPS: HMPP Wizard & Performance Analyzer

Help you to make your code Many-Core Friendly

- Play the role of a programming manual for Many-Core software development
- Bring in the same tool different perspectives/views of the code performance
- Focus on the most important performance issues
  - Gather information at compile and run time
  - Infer pertinent feedbacks from the compilation and the execution of the application

CAPS: HMPP Wizard & Performance Analyzer

- Offer different levels of information:
  - Static:
    - Analysis of the source code
    - Take into account HMPP Code Generation directives
  - Dynamic:
    - CPU profiling
    - Feedback on GPU execution with the Performance Analyzer

- 2 kinds of analysis
  - Inside HMPP code: guide the optimization of the code (advice)
  - Outside the HMPP code:
    - Provide information on potential parallel loops
    - Library usage opportunity

HMPP Wizard & Performance Analyzer

- Command line

  $ hmppReport **–tools=[Tools]** -o [OutputDirectory] <Usual Compilation Line>

- List of supported tools:
  - WIZARD — Analyze the application and provide some advice
  - GPROF — Profiling analysis by using GNU profiler
  - PERFANALYZER — To get feedback about the GPU execution

- Then get the consolidated results with an HTML report:

  $ hmppReport --build -o [OutputDirectory]

HMPP Wizard and Performance Analyzer

- Visualize the result:

  $firefox OutputDirectory/index.hmtl

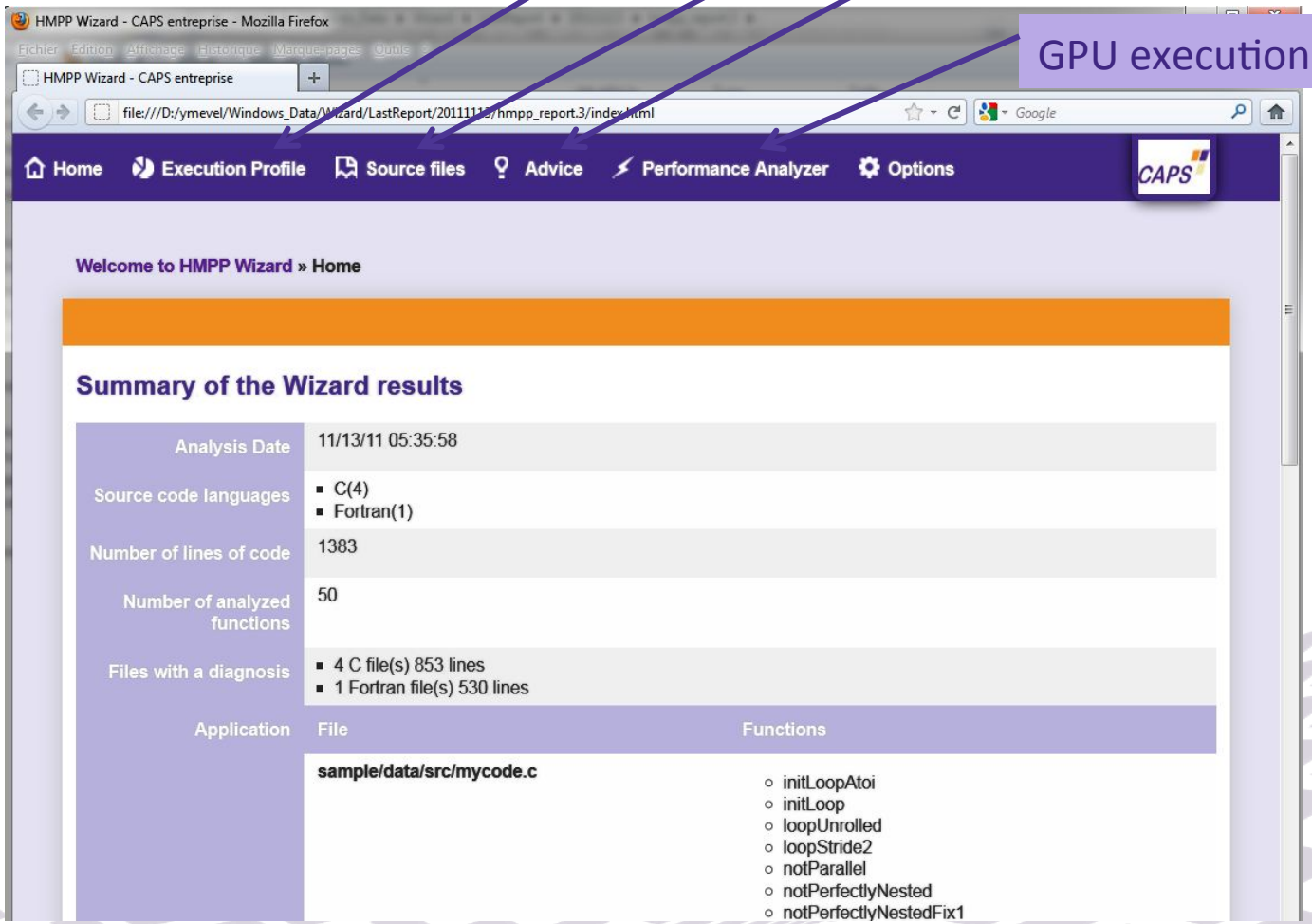
- Different tabs available

CPU Profiling view

Source files view

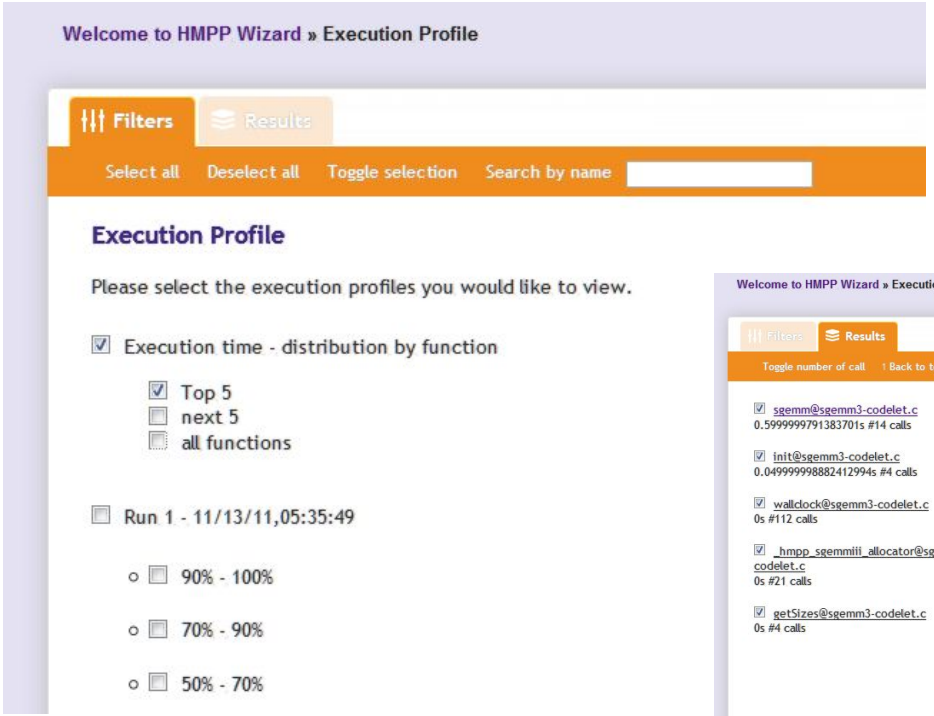
Advice view

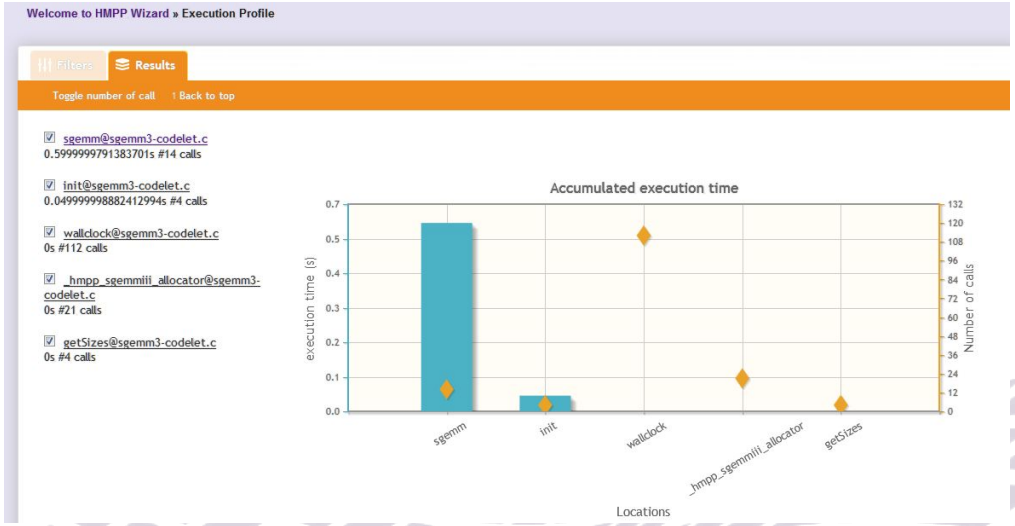
GPU execution feedback view



HMPP Wizard: CPU Profile view

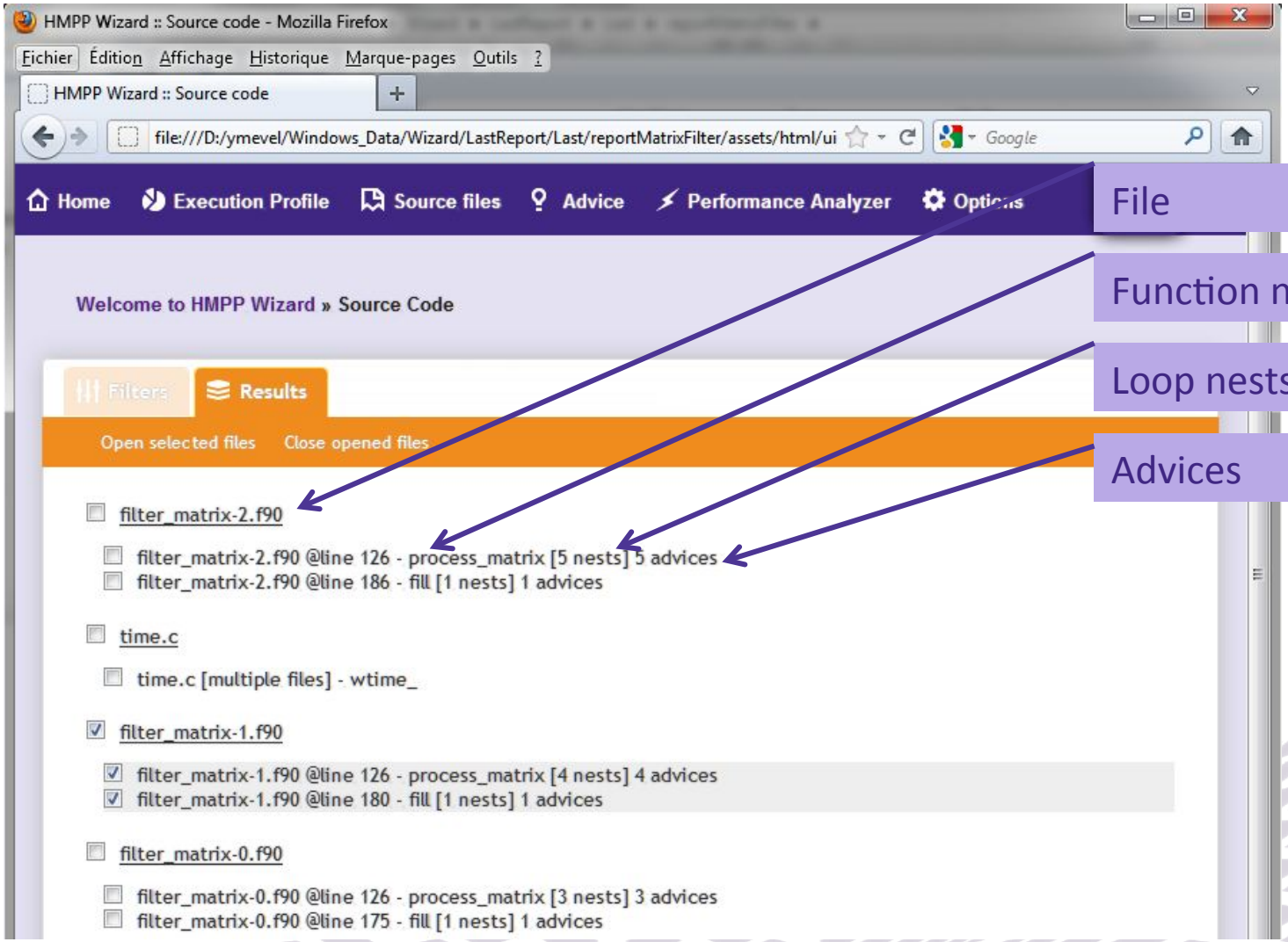
- Detect CPU hot spots
- Provide a CPU performance overview





HMPP Wizard: source code view

- file-by-file synthesis view

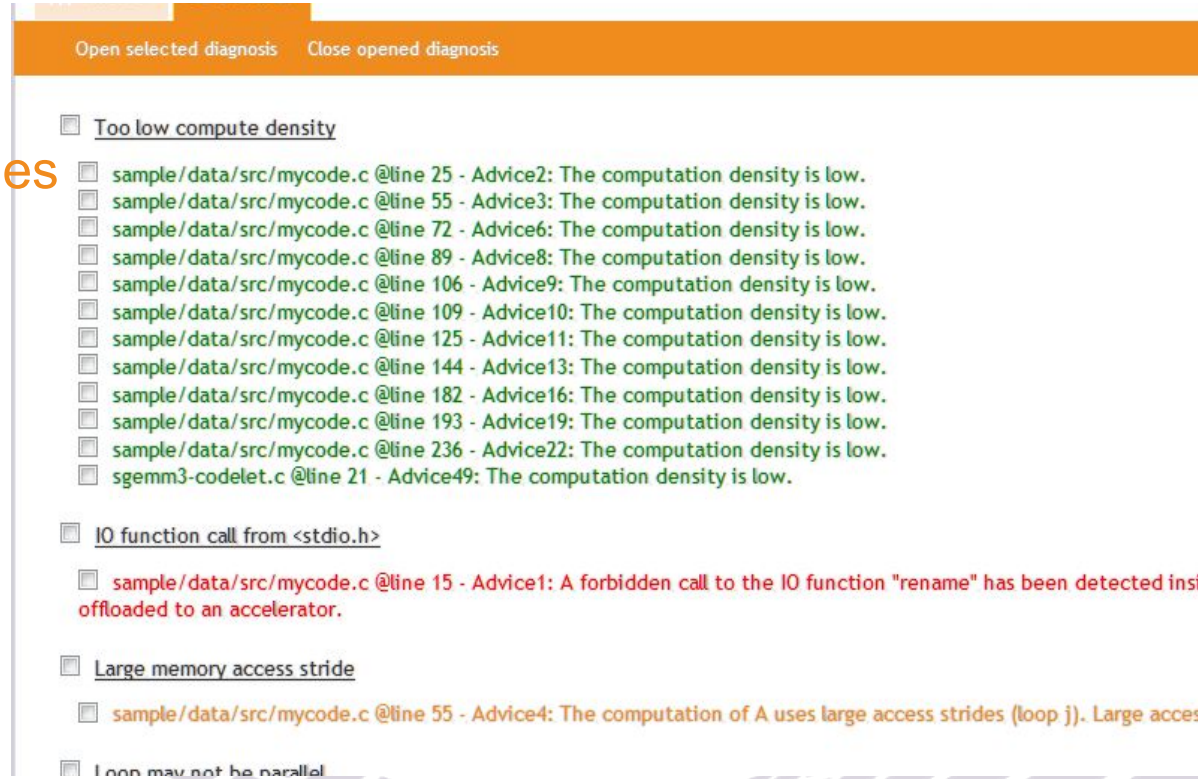


HMPP Wizard: advice view

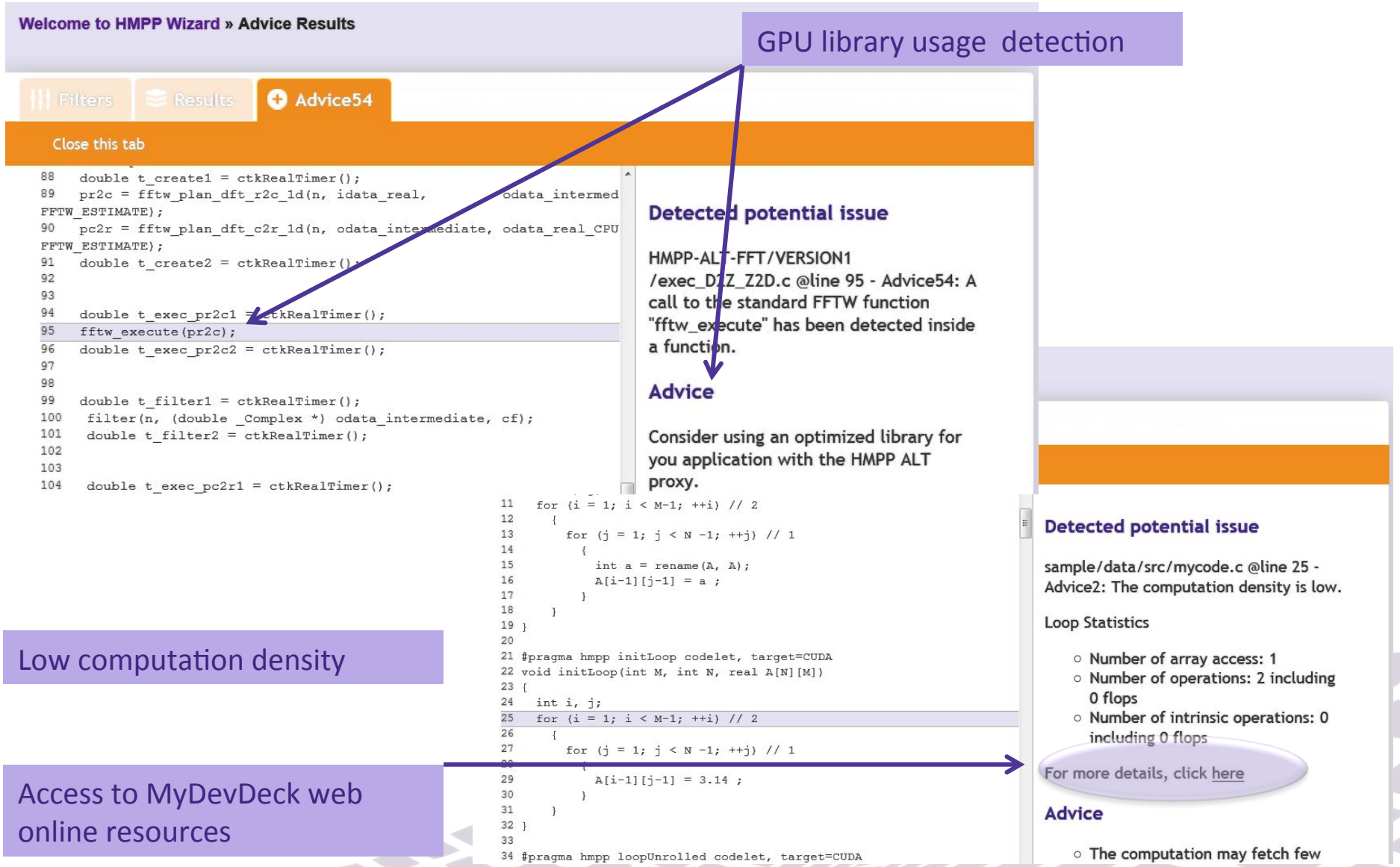
OAK RIDGE National Laboratory CAPS

Help to make your code GPU-friendly

- Detect potential GPU issues
- Provide **tuning advice**
  - Check memory access patterns
  - Improve parallelism
  - …
- Take into account HMPP Code Generation directives

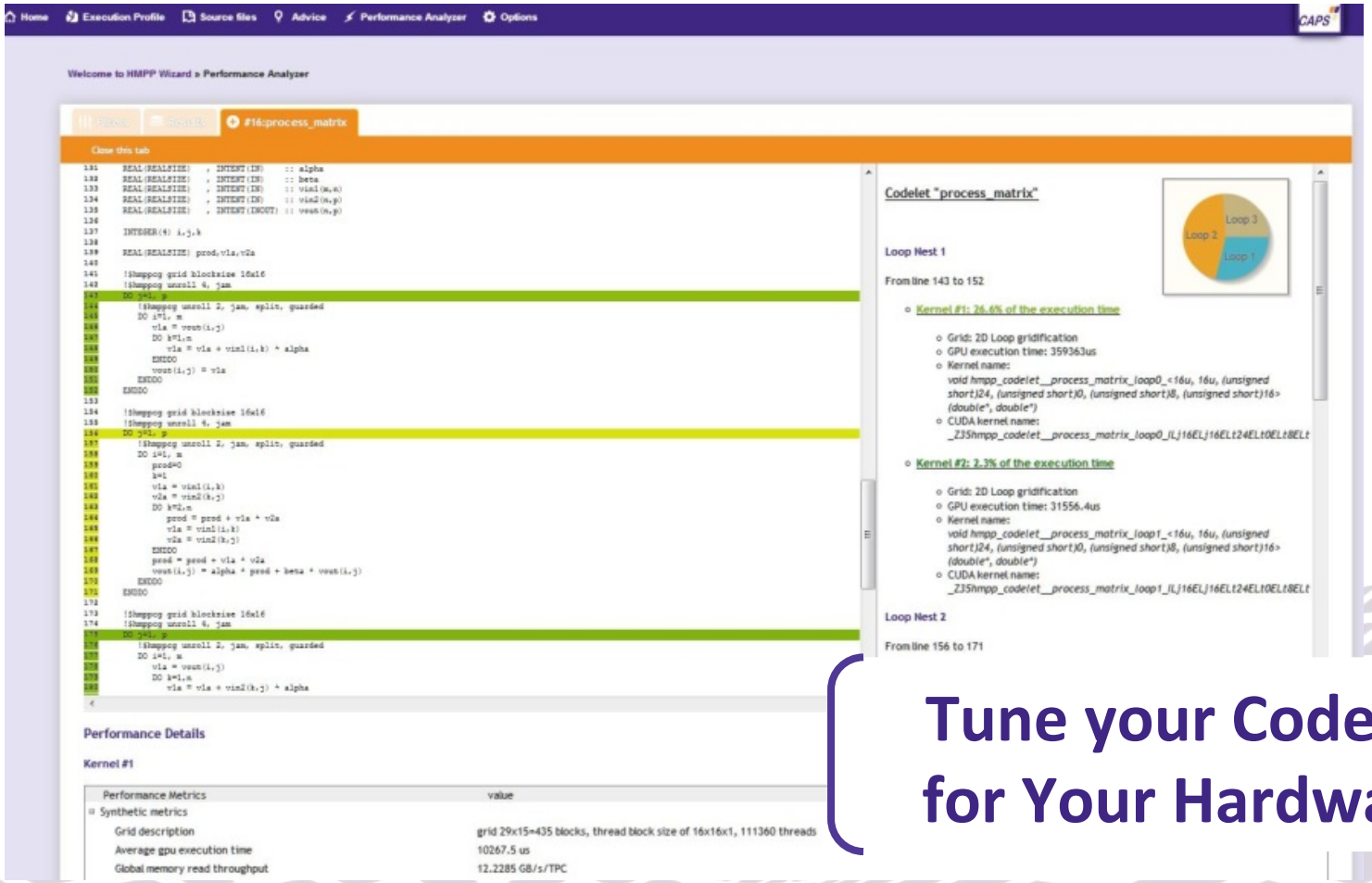


HMPP Wizard: advice view (example)



Performance Analyzer view

- GPU execution Feedback
  - Based on CUDA profile
  - Several executions may be necessary to collect all the information
- Synthetize metrics based on GPU execution profile



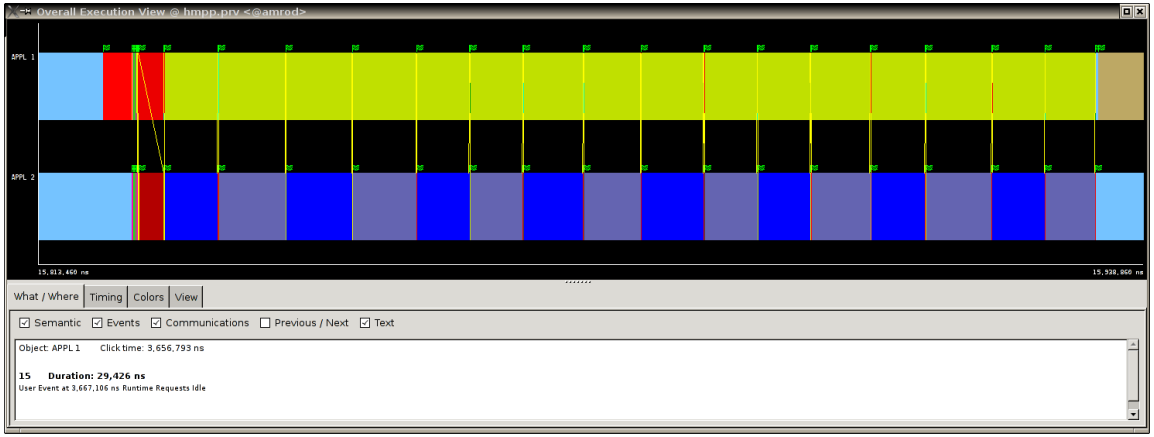
Tune your Codelet for Your Hardware

MyDevDeck: online web resources

- Online Web resources completed by CAPS

Perspectives

- Integration of a new profiling tool: based on PARAVER (from *Barcelona Supercomputing Center*). Will bring a global view of the execution by allowing to visualize:
  - GPU management
  - Kernel execution
  - Data transfer



- More GPU-accelerated libraries

- Online MyDevDeck resource will be enriched

Global Solution for Many-Core Programming

- A single package with software development tools

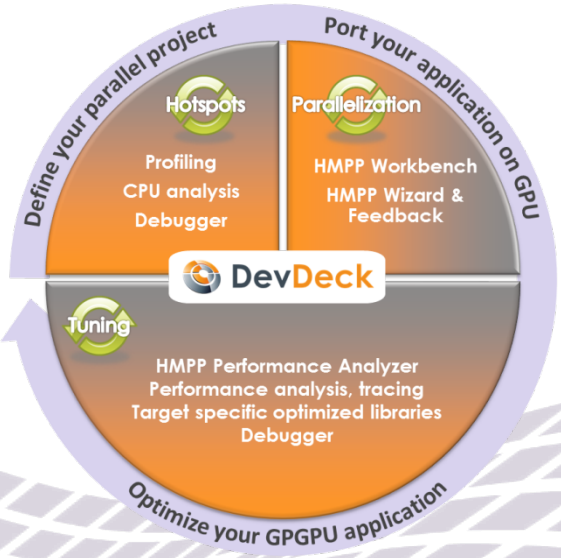
DevDeck

CAPS
- HMPP Compiler
- HMPP Wizard
- HMPP Performance Analyzer

debugger
- allinea ddt — the distributed debugging tool

libraries
- CULA tools
- OR
- ArrayFire

- Based on the porting methodology and CAPS guideline
  - MyDevDeck: Online web resources
  - Dedicated CAPS expertise



For NVIDIA CUDA & Linux

DevDeck Developer Licensing & Support

Software product	License type	Maintenance and Support
HMPP Hybrid Compiler	Perpetual License Node-locked / single concurrent user	First year included
HMPP Wizard **&** **HMPP Performance Analyzer**	Perpetual License Node-locked / single concurrent user	First year included
Allinea DDT	Perpetual License 1 GPU max / 2 CPUs / 8 Cores	First year maintenance with first level support by CAPS
***CULA Dense License**	One year subscription Per computer basis	First year maintenance with first level support by CAPS
***ArrayFire**	One Year Subscription 2 GPU max	Basic support on Accelereyes Forum

*Devdeck Developer packages include either CULA or ArrayFire libraries

DevDeck: Conclusion

- Added-value of the DevDeck: **a methodology supported through the integration of state-of-the-art products** (programming, debugging, profiling tools and libraries)
- Ideal support for the Many-core porting methodology
- Help at all stages of the development for a better productivity

Accelerator Programming Model

Directive-based programming GPGPU Manycore programming

Hybrid Manycore Programming Parallelization HPC community

Petaflops Parallel computing HPC open standard

Multicore programming Exaflops NVIDIA Cuda

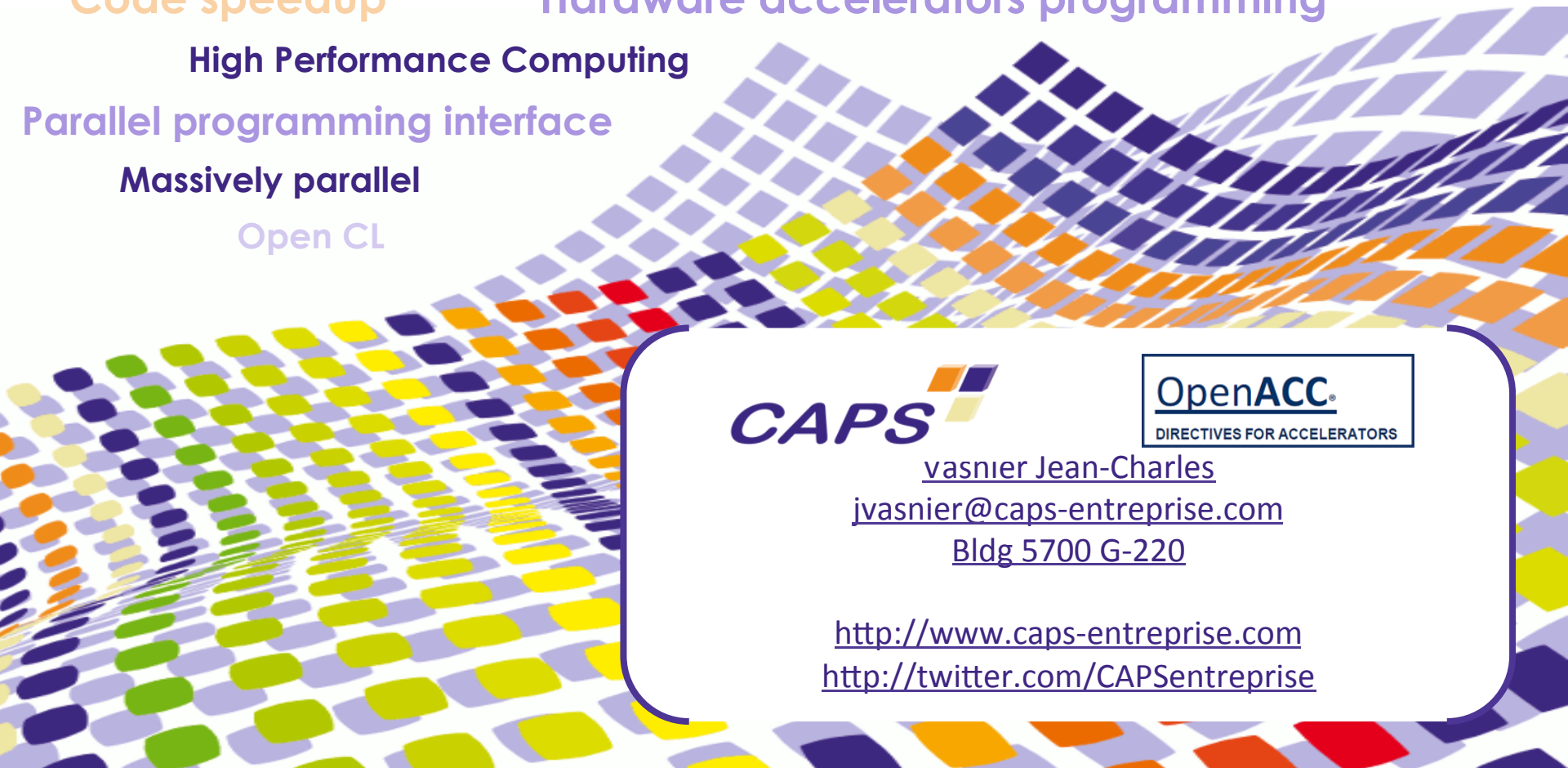
Code speedup Hardware accelerators programming

High Performance Computing

Parallel programming interface

Massively parallel

Open CL



vasnier Jean-Charles

jvasnier@caps-entreprise.com

Bldg 5700 G-220

http://www.caps-entreprise.com

http://twitter.com/CAPSentreprise