Codee Training Series

April 26-27, 2022

Shift Left Performance

Automated Code inspection for Performance

Finally: A systematic, more predictable path !

#4 **Putting it all together**

- Hands-on: **Optimizing LULESHmk** on Perlmutter
- Hands-on: **Work on your own code**

Format:

- Remote demos and hands-on sessions

The journey towards GPU in this workshop

		Challenges of GPU acceleration addressed in introductory course			Other GPU programming challenges to be addressed in next advanced course			
		Find opportunities for offloading	Optimize memory layout for data transfers	Identify defects in data transfers	Exploit massive parallelism through loop nest collapsing	Minimize data transfers across consecutive loop nests	Minimize data transfers through convergence loops	Identify auxiliary functions to be offloaded
Example codes used in this introductory course	**PI**	X	-	-	-	-	-	-
	MATMUL	X	X	X	X	X	-	-
	LULESHmk	X	X	X	X	X	X	X
	HEAT	X	-	-	-	X	X	-
	Your code!	Probably all of these challenges apply, and even more!						

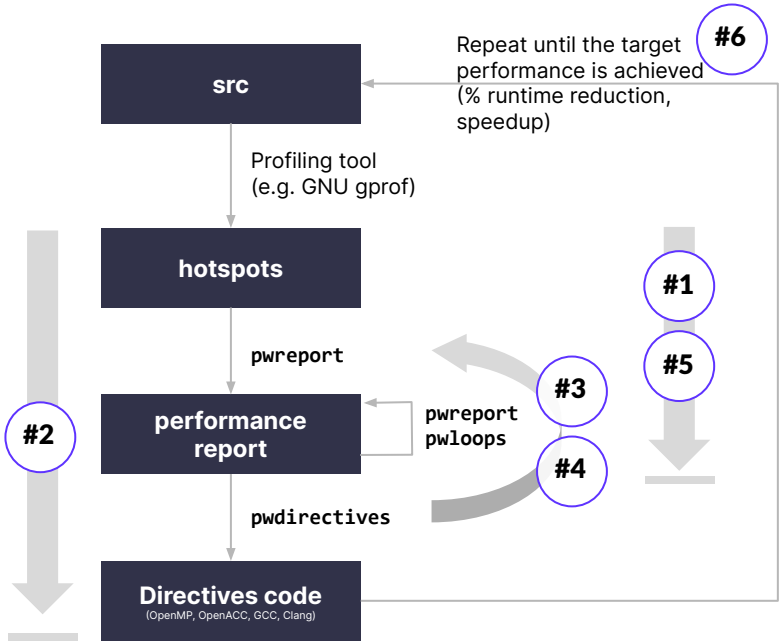
Why using additional tools apart from APIs?

- **The OpenACC Application Programming Interface. Version 2.7 (November 2018)** 🔗
  - “does **not describe automatic detection of parallel regions or automatic offloading of regions of code to an accelerator by a compiler or other tool**.”
  - “if one thread updates a memory location and another reads the same location, or two threads store a value to the same location, **the hardware may not guarantee the same result** for each execution.”
  - “it is (...) **possible to write a compute region that produces inconsistent numerical results**.”
  - “**Programmers need to be very careful that the program uses appropriate synchronization** to ensure that an assignment or modification by a thread on any device to data in shared memory is complete and available before that data is used by another thread on the same or another device.”
- **Programmers are responsible for making good use of Application Programming Interface (API)**
  - This applies to OpenACC, OpenMP
  - But also to any other API, such as MPI, compiler pragmas, and even the programming language itself

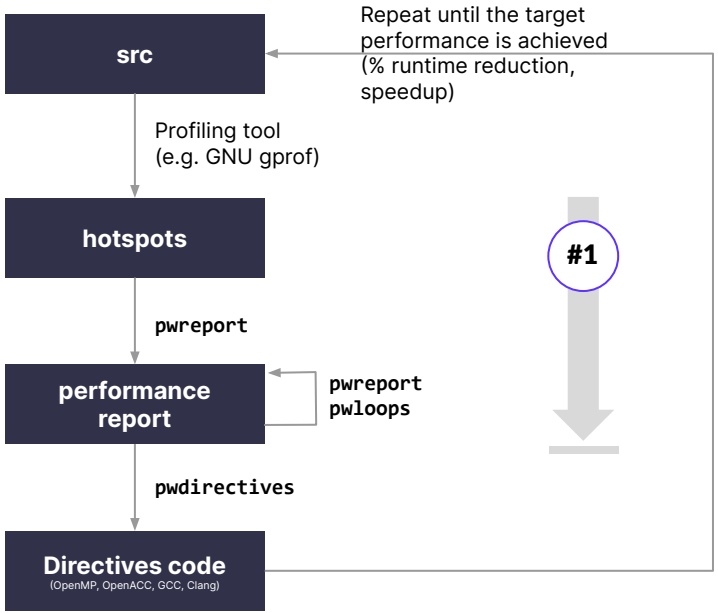
The Challenges of Real Application Codes

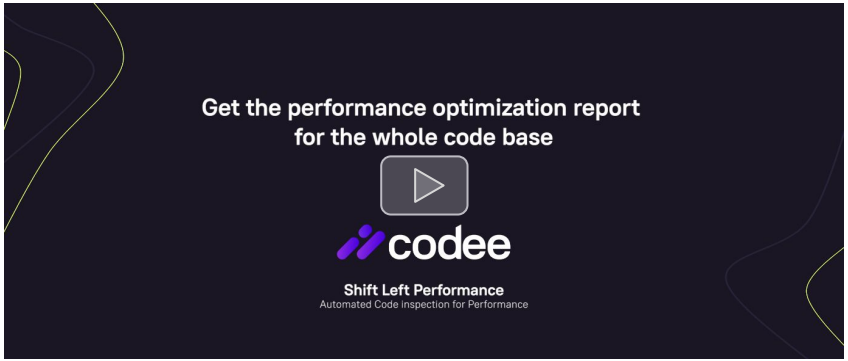
- **Challenges of real application codes on real hardware platforms include but are not limited to…**
  - Dealing with several programming languages
  - Dealing with several compilers
  - Dealing with several target hardware platforms
  - Dealing with several runtime systems
  - Dealing with several build systems
  - Dealing with several Operating systems (OS)
  - Properly doing the benchmarking of the performance-optimized code
  - etc...
- **Applications code being optimized may have different requirements or a tradeoff between several of them…**
  - performance
  - code maintainability
  - code readability
  - code portability
  - etc....

Typical Use Cases for C/C++ Developers: : Profile guided!



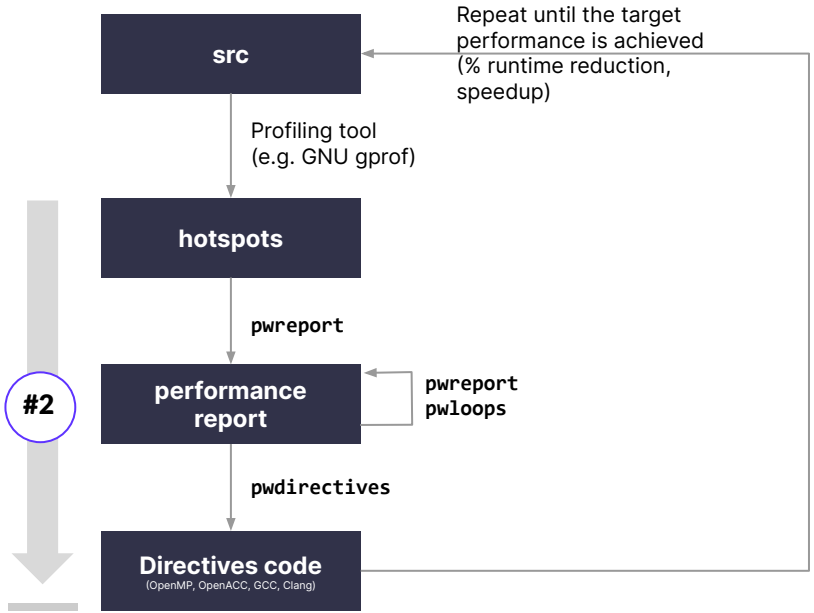
- #1 **Get the performance optimization report for the whole code base**
- #2 **Create performance-optimized code for the hotspot automatically**
- #3 **Unlock new performance optimization opportunities in the code**
- #4 **Integration with compilers**
- #5 **Integration with build systems**
- #6 **Benchmark performance impact on your hardware platform**

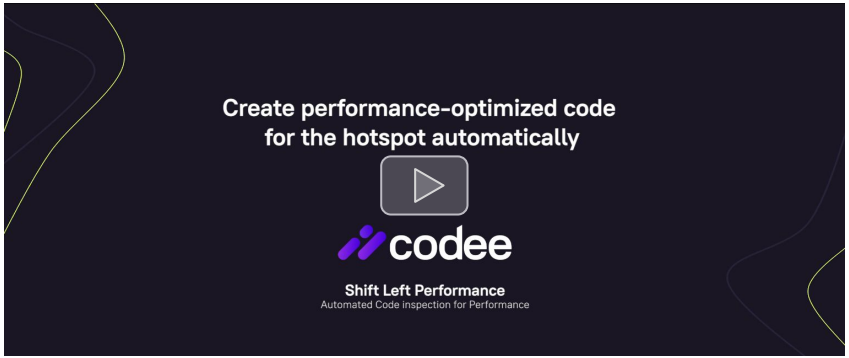




https://www.codee.com/wp-content/uploads/2022/01/Demo1-pwa-canny-pwreport-evaluation.mp4

#1 **Get the performance optimization report for the whole code base**
pwreport --evaluation



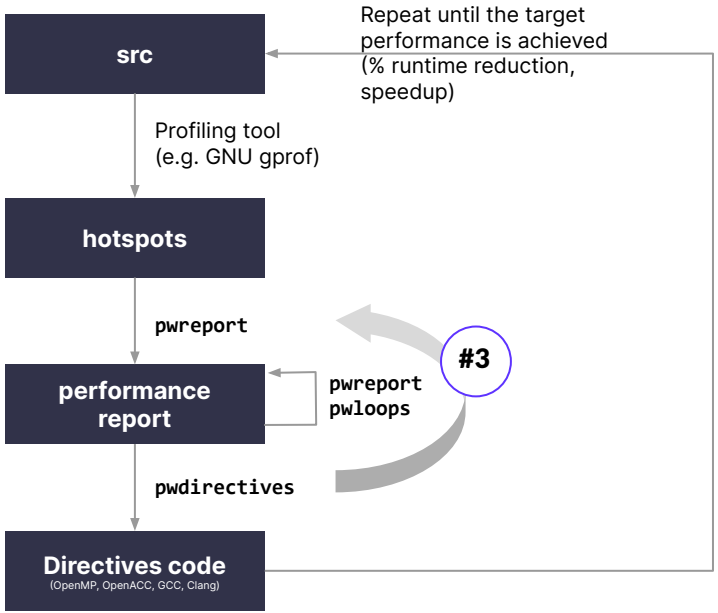


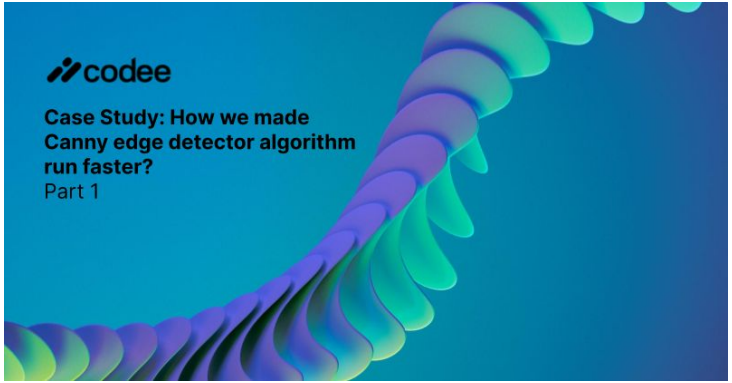
https://www.codee.com/wp-content/uploads/2022/01/Demo2-pwa-canny-pwreport-pwdirectives.mp4

#2

Create performance-optimized code for the hotspot automatically

pwreport --actions
pwdirectives --omp multi+simd





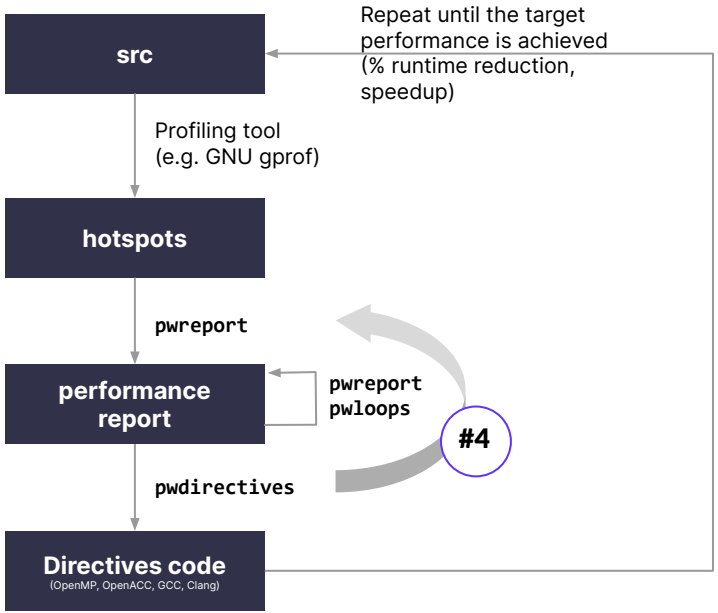
https://www.codee.com/case-study-how-we-made-canny-edge-detector-algorithm-run-faster-part-1/

#3

Unlock new performance optimization opportunities in the code

pwreport --actions --level 2

https://www.codee.com/knowledge/

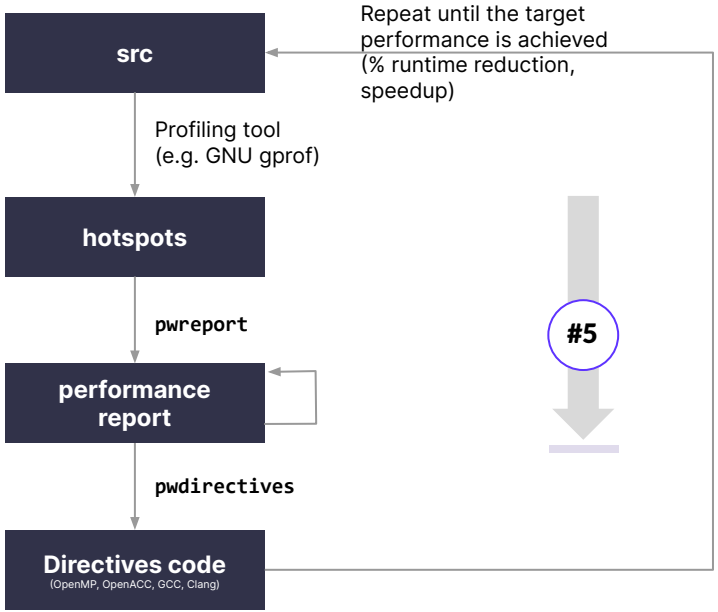


https://www.codee.com/wp-content/uploads/2022/01/Demo4-pwa-canny-pwloops-vector-support.mp4

#4

Integration with compilers

pwloops --vector-support --show-messages code.c:328

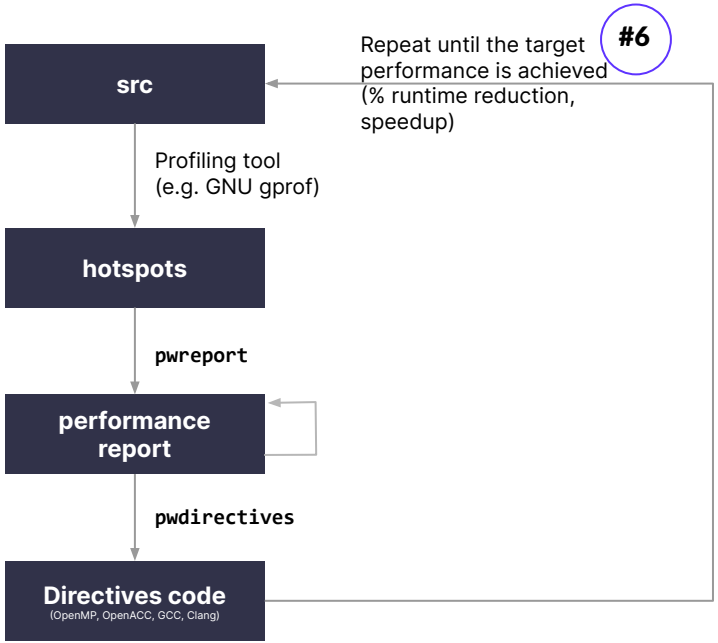


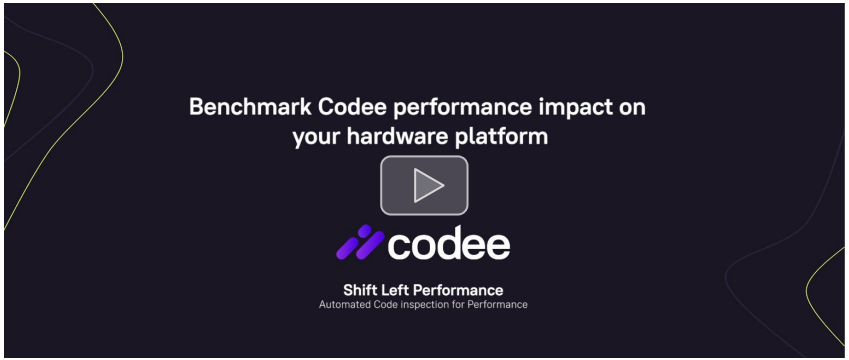
https://www.codee.com/wp-content/uploads/2022/01/Demo5-pwa-mbedtls-integration-build-systems.mp4

#5

Integration with build systems

pwreport --config compile_commands.json





https://www.codee.com/wp-content/uploads/2022/01/Demo6-pwa-performance-demos.mp4

#6 **Benchmark performance impact on your hardware platform**
https://github.com/teamappentra/performance-demos/

codee

www.codee.com

info@codee.com

Subscribe: codee.com/newsletter/

USA - Spain

codee_com

company/codee-com/