

# AMD ROCm GPU profiling in Trace Compass

Arnaud Fiorini with Pr. Michel Dagenais May 8th, 2020

Polytechnique Montreal

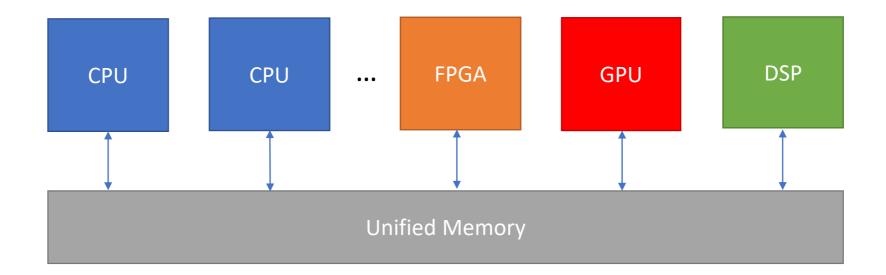
DORSAL Laboratory

- I. Introduction
  - 1. GPU Development
  - 2. Optimization Problems
- II. Tracing and profiling of CPU-GPU systems
  - 1. ROC Platform
  - 2. Tracing GPUs
  - 3. Profiling GPUs

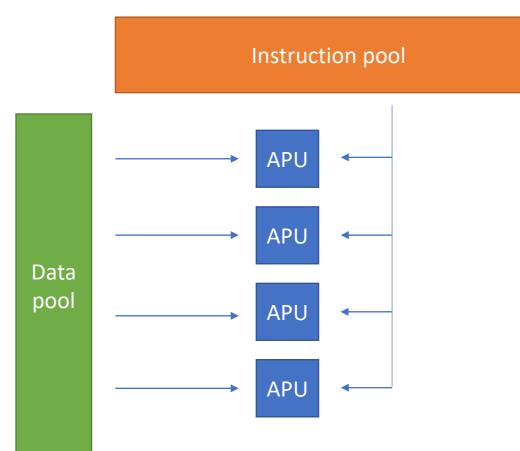
- A few definitions :
  - Kernel : A small piece of code executed on the device.

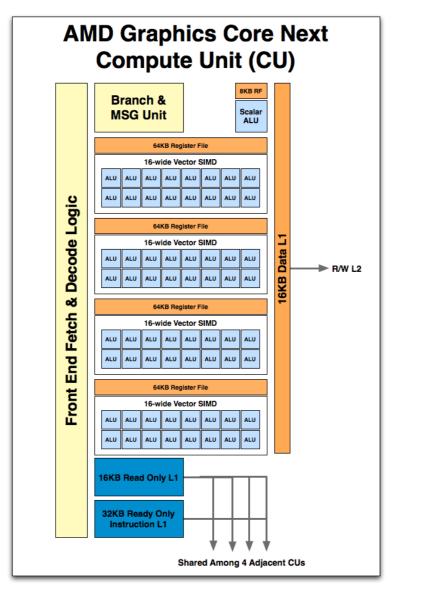
```
__kernel void saxpy(__global float *src, __global float *dst, float factor)
{
    long i = get_global_id(0);
    dst[i] += src[i] * factor;
}
```

- A few definitions :
  - Kernel : A small piece of code executed on the device.
  - Heterogeneous system : system mixing multiple types of processors



• SIMD Architecture :







#### © 2019 AMD Corporation

# **Optimization Problems - Introduction**

- Communication Overhead :
  - Memory synchronisation
  - Interprocessor Communication
- Scheduling and load balancing :
  - Benchmarking
  - Load characteristics of kernels
- Shared Cache :
  - Cache misses, thrashing

## **Optimization Problems - Introduction**

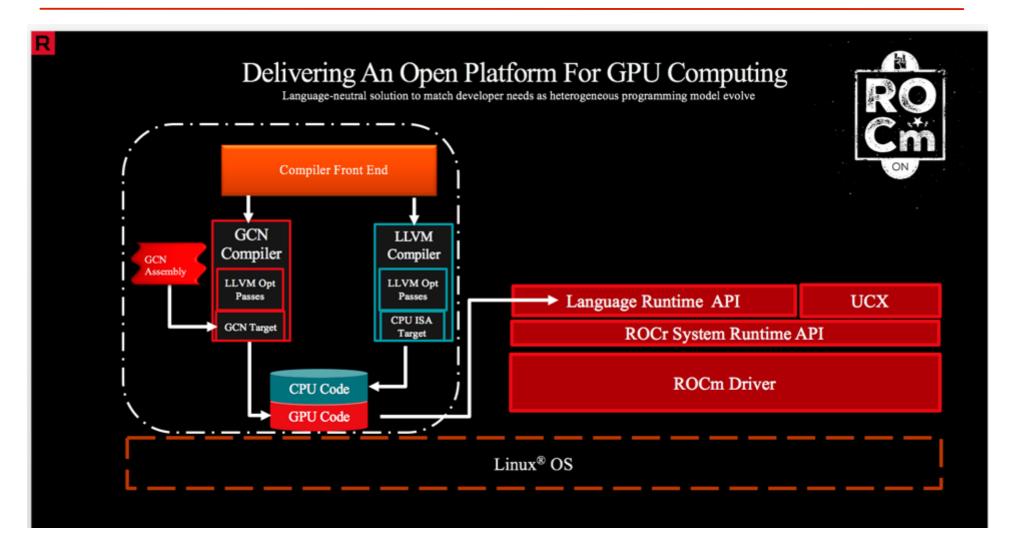
- Communication Overhead :
  - Memory synchronisation \_\_\_\_\_ Tracing
  - Interprocessor Communication
- Scheduling and load balancing :
  - Benchmarking
  - Load characteristics of kernels

Profiling (Performance Counters)

- Shared Cache :
  - Cache misses, thrashing

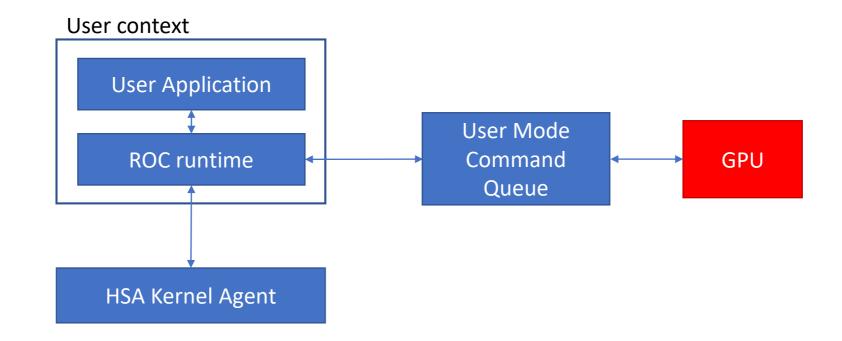
# Agenda

- I. Introduction
  - 1. GPU Development
  - 2. Optimization Problems
- II. Tracing and profiling of CPU-GPU systems
  - 1. ROC Platform
  - 2. Tracing GPUs
  - 3. Profiling GPUs

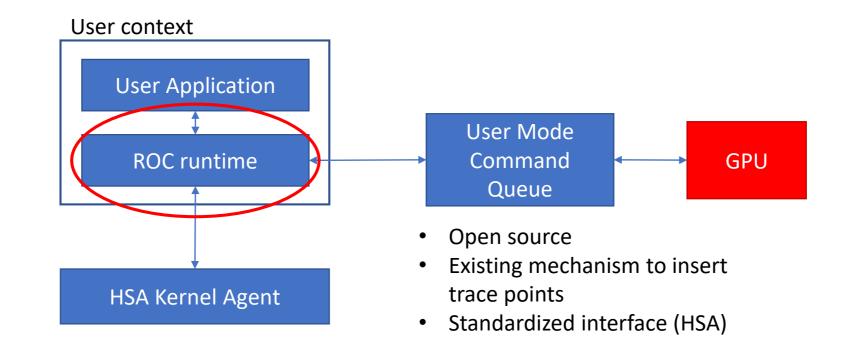


© 2019 AMD Corporation <a href="https://rocm.github.io/">https://rocm.github.io/</a>

ROCm functioning summarized :

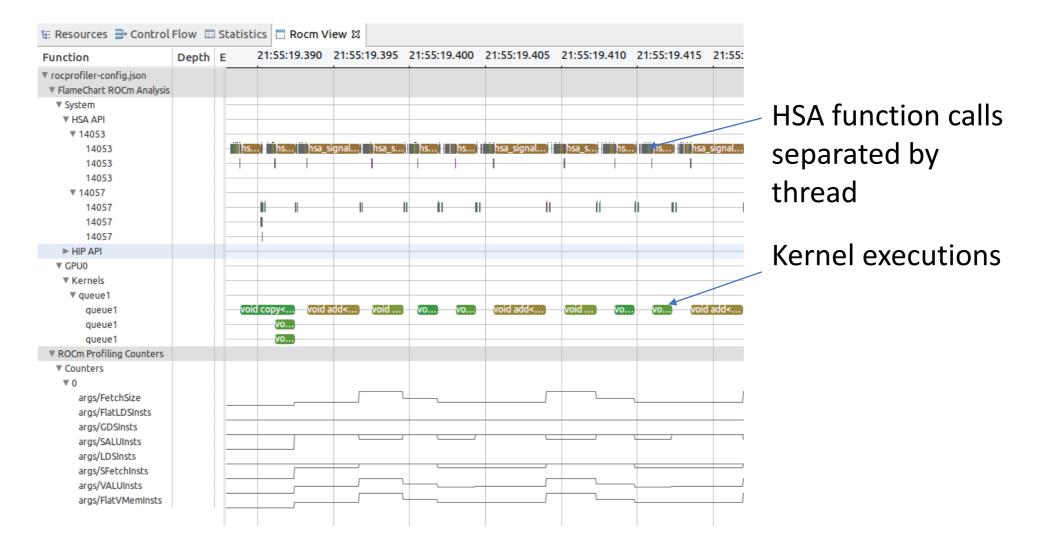


ROCm functioning summarized :

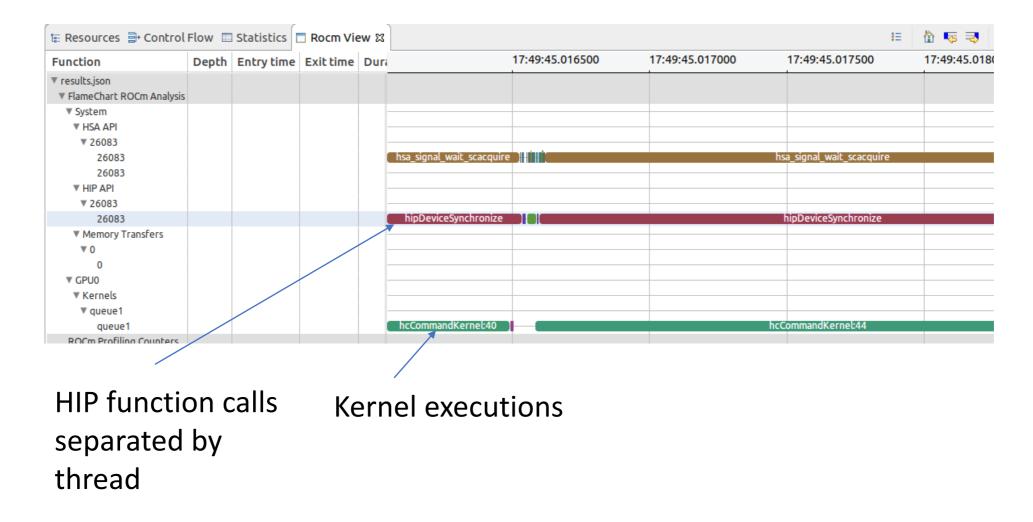


- This work has already been done by AMD and is open source : <u>https://github.com/ROCm-Developer-Tools/rocprofiler</u> <u>https://github.com/ROCm-Developer-Tools/roctracer</u>
- AMD has released a few other libraries and tools thanks to their Radeon Open Compute initiative.

#### TraceCompass ROCm plugin



#### TraceCompass ROCm plugin



#### TraceCompass ROCm plugin running on Theia front-end

	Elle Belle Oslandare Mauri Os Terminel Hele
	File Edit Selection View Go Terminal Help
ð	EXPLORER
ים	► Pictures
Le B	Programs
<u> </u>	► Public
	<ul> <li>In untime-EclipseApplication</li> </ul>
	<ul> <li>► sdcard</li> </ul>
	<ul> <li>► snap</li> </ul>
	<ul> <li>► Templates</li> </ul>
	theia-trace-extension
	<ul> <li>► Videos</li> </ul>
	<ul> <li>Workspace</li> </ul>
	<ul> <li>workspace</li> <li>workspacenew</li> </ul>
	.bash_history
	.bash_logout
	■ .bashrc
	• .dmrc
	♦ .fehbg
	♦ .gitconfig
	✿ .ICEauthority
	.node_repl_history
	notes.txt.swp
	<ul> <li>profile</li> </ul>
	<ul> <li>.python_history</li> </ul>
	.repogitconfig.json
	<ul> <li>.sudo_as_admin_successful</li> </ul>
	<ul> <li>viminfo</li> </ul>
	K .vimrc
	.wget-hsts
_	X .Xauthority
	X .xinitrc
_	X .Xresources
	X .xsession-errors
	<ul> <li>.xsession-errors.old</li> </ul>
	♦ .yarnrc
	# examples.desktop
	log
	presentation_progress_meeting.odp
	I rocprofiler-config.csv
	✓ rocprofiler-config.db
	b rocprofiler-config.hsa_stats.csv
_	<pre>support = rocprofiler-config.json</pre>
	L rocprofiler-config.stats.csv
	rocprofiler-config.txt
∞ 0	
00	

Analyzing this tracing data further, future work includes :

- Critical path analysis of CPU-GPU execution
- Determining whether the program performance is limited by the GPU or the CPU
- Extracting statistics to use in profiling analysis

#### TraceCompass ROCm plugin

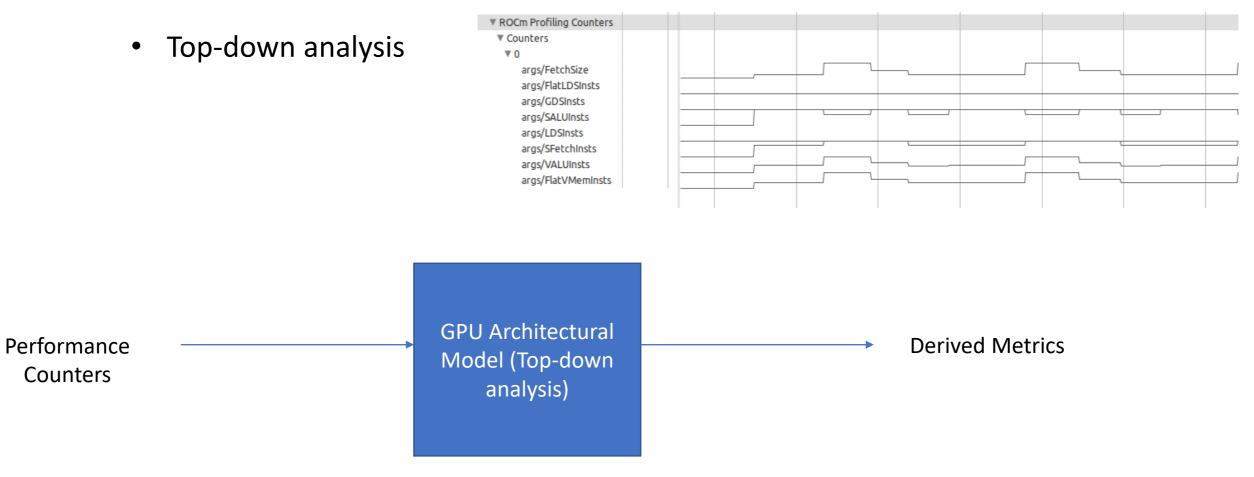
Function	Depth	E	21:55:19.390	21:55:19.395	21:55:19.400	21:55:19.405	21:55:19.410	21:55:19.415	21:55:	
rocprofiler-config.json										
▼ FlameChart ROCm Analysis										
▼ System										
▼ HSA API										
▼ 14053										
14053		– hs.	hshsa_s	signal] hsa_s	hs hs	hsa_signal	hsa_s hs	. hs hsa	_signal]	
14053				- I		1	1			
14053										
▼ 14057										
14057			<b>I</b>	-		l İl		1 11		
14057			1							
14057										
► HIP API										
▼ GPU0										
▼ Kernels										
▼ queue1										
queue1		voi	d copy <void< td=""><td>add&lt; void</td><td>vo</td><td>void add&lt;</td><td>void vo</td><td>voio voio</td><td>add&lt;)</td><td></td></void<>	add< void	vo	void add<	void vo	voio voio	add<)	
queue1			vo							
queue1			vo							Performance
ROCm Profiling Counters										Performance
▼ Counters										
▼ 0										counters
args/FetchSize					4			<i>٦</i> ــــــ		Counters
args/FlatLDSInsts										
args/GDSInsts										
args/SALUInsts								<u> </u>	1	
args/LDSInsts										
args/SFetchInsts								1	<u> </u>	
args/VALUInsts					·	+		·	<u> </u>	
args/FlatVMemInsts					4			<b>\</b>		

- π 2 Performance [GFLOPS] 1 O<sub>2</sub>(compute-bound) O<sub>1</sub>(memory-bound) 2 1/2 1/4 1/4 1/2 32 64 128 8 16 2 Operational Intensity [FLOPS/byte]  $\beta$  : peak bandwidth
- Roofline model

- I : arithmetic intensity
- $\pi$  : peak performance

- π 2 Performance [GFLOPS] 1 Х O<sub>2</sub>(compute-bound) O<sub>1</sub>(memory-bound) хх  $\mathfrak{C}$ X 1/2 1/4 1/4 1/2 32 64 128 8 16 2 Operational Intensity [FLOPS/byte]
- **Roofline model**  $\bullet$

- $\beta$  : peak bandwidth
- I : arithmetic intensity
- $\pi$  : peak performance



r	Top-down      Grouping: Function /	-	<ul> <li>ROCm Profiling Counters</li> <li>Counters</li> <li>0 args/FetchSize args/FlatLDSInsts args/GDSInsts</li> </ul>							
-	Grouping. Tunction /					Back-End B				
	Function / Call Stack	Instructions Retired	CPI Rate	Front-End <sup>»</sup> Bound	Bad Speculation	Memory Bound»	Bound « Core Bound »	Retiring		
	price_out_impl	62,556,093,834	1.261	2.2%	7.4%	64.2%	8.4%	17.8%		Í.
	refresh_potential	17,836,026,754	3.589	3.0%	8.1%	73.2%	9.6%	6.1%		
	primal_bea_mpp	38,108,057,162	1.393	5.6%	24.3%	34.4%	21.0%	14.7%		
-	update_tree	4,092,006,138	3.373	7.2%	11.5%	62.3%	11.8%	7.2%		
-	sort_basket	12,246,018,369	1.037	20.7%	50.4%	3.8%	4.6%	20.6%		
Performance	primal_iminus	5,324,007,986	2.148	7.1%	6.7%	55.0%	20.5%	10.8%	rics	
Counters	primal_net_simple:	266,000,399	2.466	17.4%	43.4%	13.1%	13.1%	13.0%		
-			ć	anaiysis)						

# Thank you for listening !

# Questions ?

## References

- <u>https://github.com/RadeonOpenCompute/ROCm</u>
- <u>https://rocm-documentation.readthedocs.io/en/latest/</u>
- http://www.hsafoundation.com/
- HSA Runtime Programmer's Reference Manual, Version 1.2
- HSA Programmer's Reference Manual, Version 1.2
- HSA Platform System Architecture Specification, Version 1.2
- <u>https://github.com/ucb-bar/opencl-kernels/blob/master/saxpy/kernel.cl</u>
- <u>https://medium.com/@smallfishbigsea/basic-concepts-in-gpu-computing-3388710e9239</u>
- <u>https://www.techpowerup.com/gpu-specs/docs/amd-gcn1-architecture.pdf</u>
- <u>https://software.intel.com/content/www/us/en/develop/docu</u> <u>mentation/vtune-cookbook/top/methodologies/top-down-</u> <u>microarchitecture-analysis-method.html</u>