# Simulating nanoscale dragons

Towards fully open source GPU accelerated molecular dynamics simulation

<u>Vedran Miletić</u>, HITS gGmbH Szilárd Páll, KTH Frauke Gräter, HITS gGmbH

Heidelberg Institute for Theoretical Studies





#### Molecular dynamics simulation

• GROMACS simulates proteins, i.e. nanoscale dragons\*



\*careful, they can still bite

GPU accelerated app (e.g. GROMACS) and libraries



GPU accelerated app (e.g. GROMACS) and libraries				
CUDA	OpenCL			
NVIDIA proprietary compiler		AMD proprietary compiler		
NVIDIA proprietary driver		AMD proprietary driver		

GPU accelerated app (e.g. GROMACS) and libraries				
CUDA	OpenCL			
NVIDIA proprietary compiler		AMD proprietary compiler	Clang and libclc	
NVIDIA proprietary driver		AMD proprietary driver	LLVM, Mesa/DRM, and radeon/nouveau	

GPU accelerated app (e.g. GROMACS) and libraries				
	Contribute	d by StreamComputing		
CUDA		OpenCL		
NVIDIA proprietary compiler		AMD proprietary compiler	Clang and libclc	
NVIDIA proprietary driver		AMD proprietary driver	LLVM, Mesa/DRM, and radeon/nouveau	

GPU accelerated app (e.g. GROMACS) and libraries					
	Contribute	Contributed by StreamComputing			
CUDA		OpenCL Our work			
NVIDIA proprietary compiler		AMD proprietary compiler	Clang and libclc		
NVIDIA proprietary driver		AMD proprietary driver	LLVM, Mesa/DRM, and radeon/nouveau		nd

# If using **proprietary stack** is OK, NVIDIA/Intel do it better than AMD.

Accelerator/CP Family Performance Share



Nvidia Kepler
Intel Xeon Phi
Nvidia Fermi
Hybrid
ATI Radeon
PEZY-SC

Source: top500.org statistics for November 2015.

#### **Open source OpenCL stack**

- Anyone can improve code
  - Hopefully less bugs, better performance

#### **Open source OpenCL stack**

- Anyone can improve code
  - Hopefully less bugs, better performance
- Develop what you care about, no "company priorities"



James Price @jrprice89

👤 Follow

NVIDIA rep at #SC15 acknowledged that lots of customers are asking for #OpenCL, but certain individuals inside the company are pushing back.



#### Road to openness

- Remove the unused OpenCL image functions from GROMACS
  - Done by Szilard Pall, accepted upstream

# Road to openness

- Remove the unused OpenCL image functions from GROMACS
  - Done by Szilard Pall, accepted upstream
- Implement global atomic compare-and-swap in LLVM AMDGPU target
  - Custom lowering to {BUFFER,FLAT}\_ATOMIC\_CMPSWAP

00

- Reeviewed by AMDGPU target maintainers ( )
- Will be merged as soon as it has tests

#### Work in progress

Add erf() / erff() to libclc

# Work in progress

- Add erf() / erff() to libclc
- Handle struct arguments in OpenCL kernels correctly
  - Few possible approaches, cf. Beignet for Intel iGPUs

# Work in progress

- Add erf() / erff() to libclc
- Handle struct arguments in OpenCL kernels correctly
  - Few possible approaches, cf. Beignet for Intel iGPUs
- Running, with result correctness issues

radeontop v0.9-2-g0e8227	2, runnin	ng on BONAIRE,	120 samples/sec
Graphics pipe	16.67%		
Event Engine	0.00%		
Texture Addresser			-
Scan Converter Primitive Assembly	0.83% 0.83%		
Depth Block Color Block	0.83% 0.00%		

# Joys of AMDGPU target development

- LLVM ABI breakage, LLVM doesn't compile, ...
- Occasional target-specific regressions

# Joys of AMDGPU target development

- LLVM ABI breakage, LLVM doesn't compile, ...
- Occasional target-specific regressions
- Limitations regarding Volcanic Islands
- Overall resemblance to early days of Mozilla



#### The future is open and is here and now

GPU accelerated app (e.g. GROMACS) and libraries			
CUDA	OpenCL, C++17		
NVIDIA proprietary compiler		Clang, libclc, HCC/HIP	
NVIDIA proprietary driver		LLVM, Mesa/DRM, and radeon/nouveau	

# Acknowledgments

- Tom Stellard, AMD
- Matt Arsenault, AMD
- Edward O'Callaghan, Freenode channel #radeon
- Serge Martin, Freenode channel #radeon