

# Optimizing SPH-EXA for AMD GPUs



Florina Ciorba (PI)  
Ruben Cabezon (Co-PI)  
Osman Seckin Simsek  
Yiqing Zhu  
Lukas Schmidt  
José Escartin



Lucio Mayer (Co-PI)  
Noah Kubli  
Darren Reed



Sebastian Keller  
Jean-Guillaume Piccinali  
Jean Favre  
Jonathan Coles



Axel Sanz (UPC)  
Joseph Touzet (Paris-Saclay)

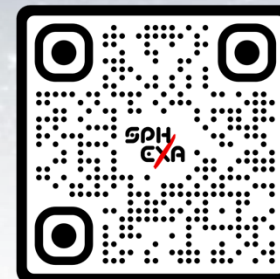
**LUMI Optimizing for AMD GPUs Hackathon**

October 14-18, 2024, Brussels

**Osman Seckin Simsek**

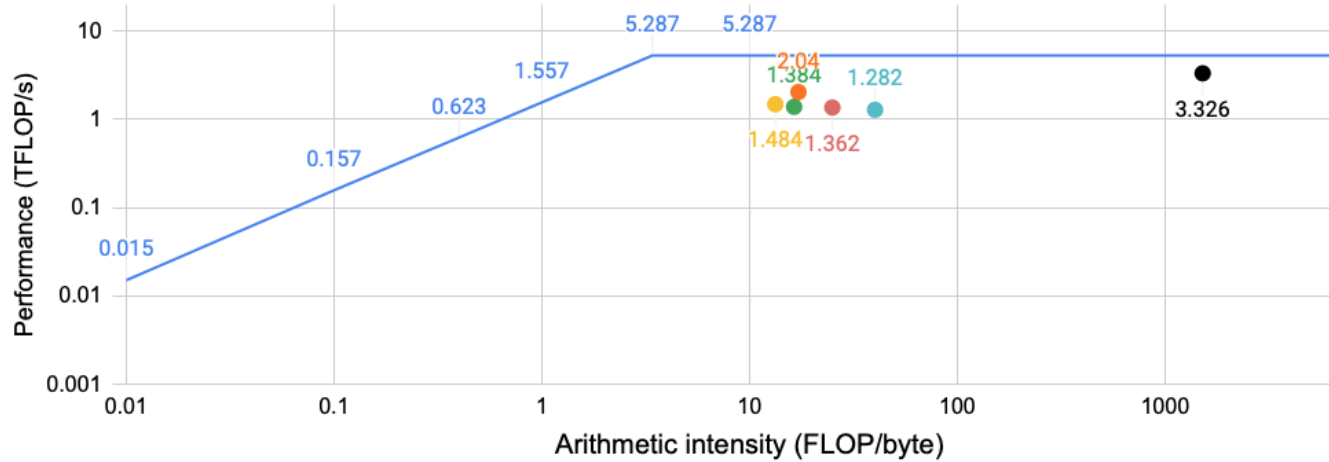


<https://github.com/unibas-dmi-hpc/SPH-EXA>

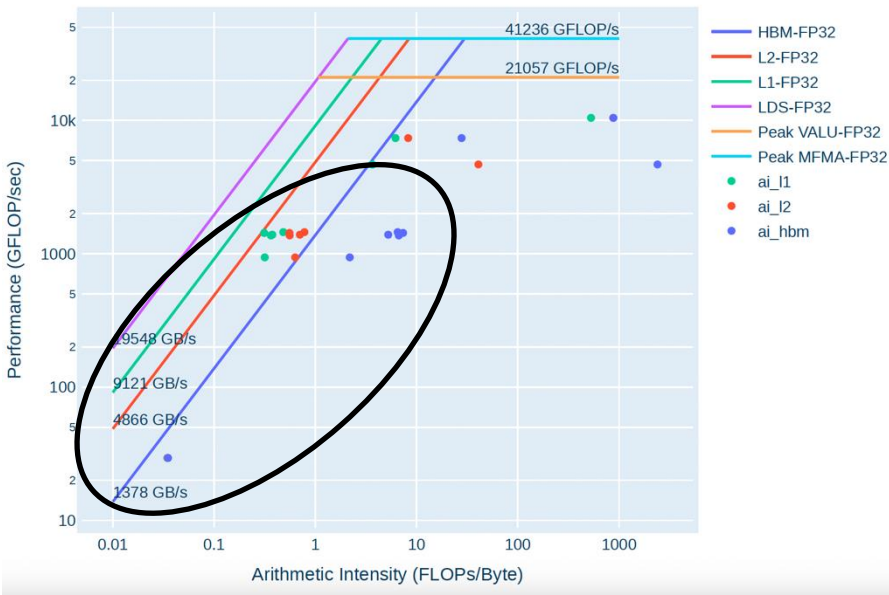


# Motivation for Joining the LUMI Hackathon

— Roofline 
 ● Momentum&Energy 
 ● Derivatives 
 ● Artificial Viscosity 
 ● Density 
 ● Gradh 
 ● Gravity

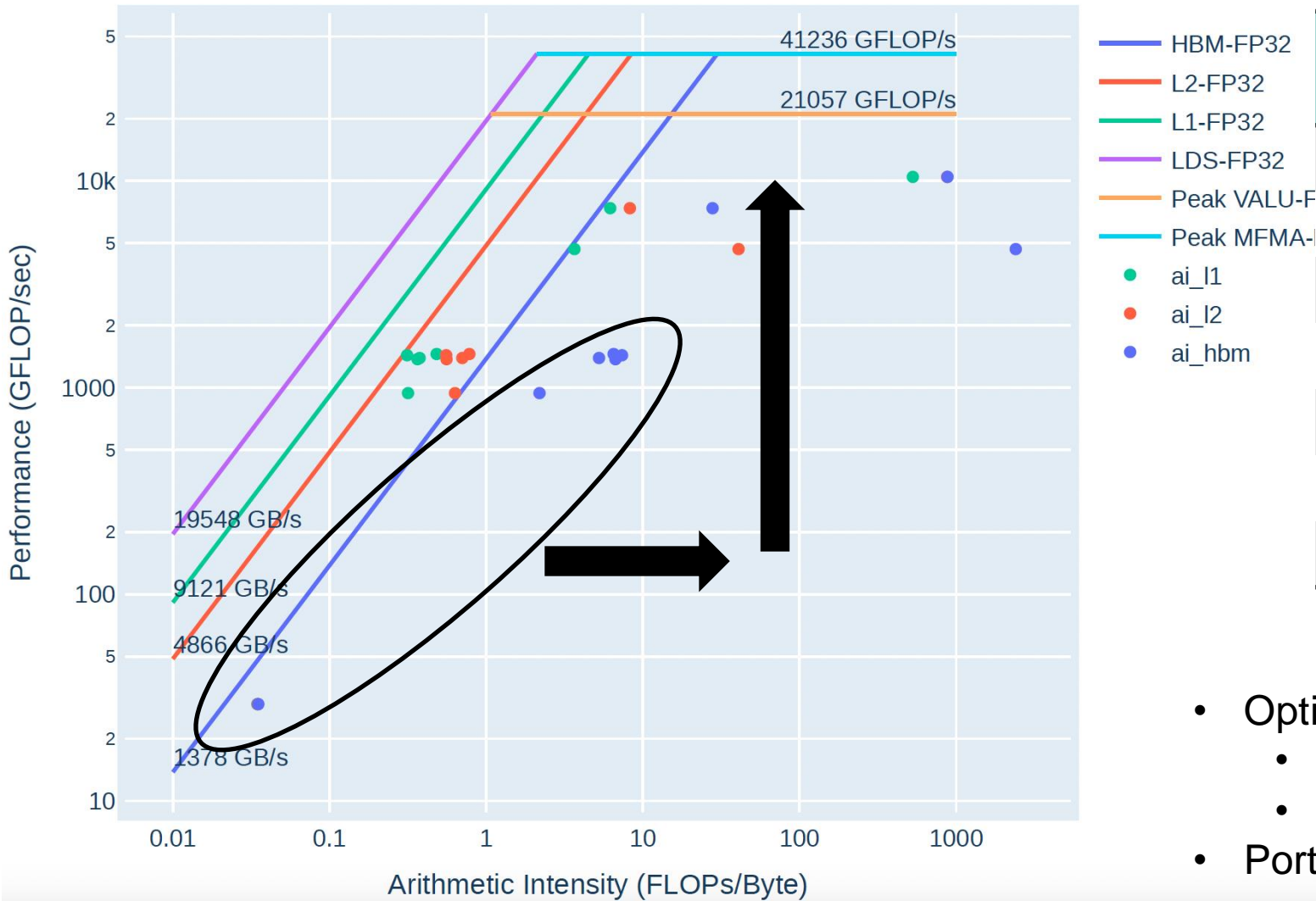


Performance of the most time consuming kernels on a single Nvidia A100 GPU.



Performance of the most time consuming kernels on a single MI250X GCD.

# SPH-EXA: Plan for LUMI Hackathon



Function Name	Percentage of time per time-step
momentumEnergy	22.24%
gravity	35.63%
iadDivvCurlv	11.18%
AVSwitches	10.12%
veDefGradh	8.49%
xMass	7.47%
<b>Total</b>	<b>95.14%</b>

- Optimize per function for:
  - Increasing the arithmetic intensity
  - Increasing the performance
- Porting Nuclear-networks computations to GPU

# SPH-EXA: LUMI Hackathon Improvements Tried

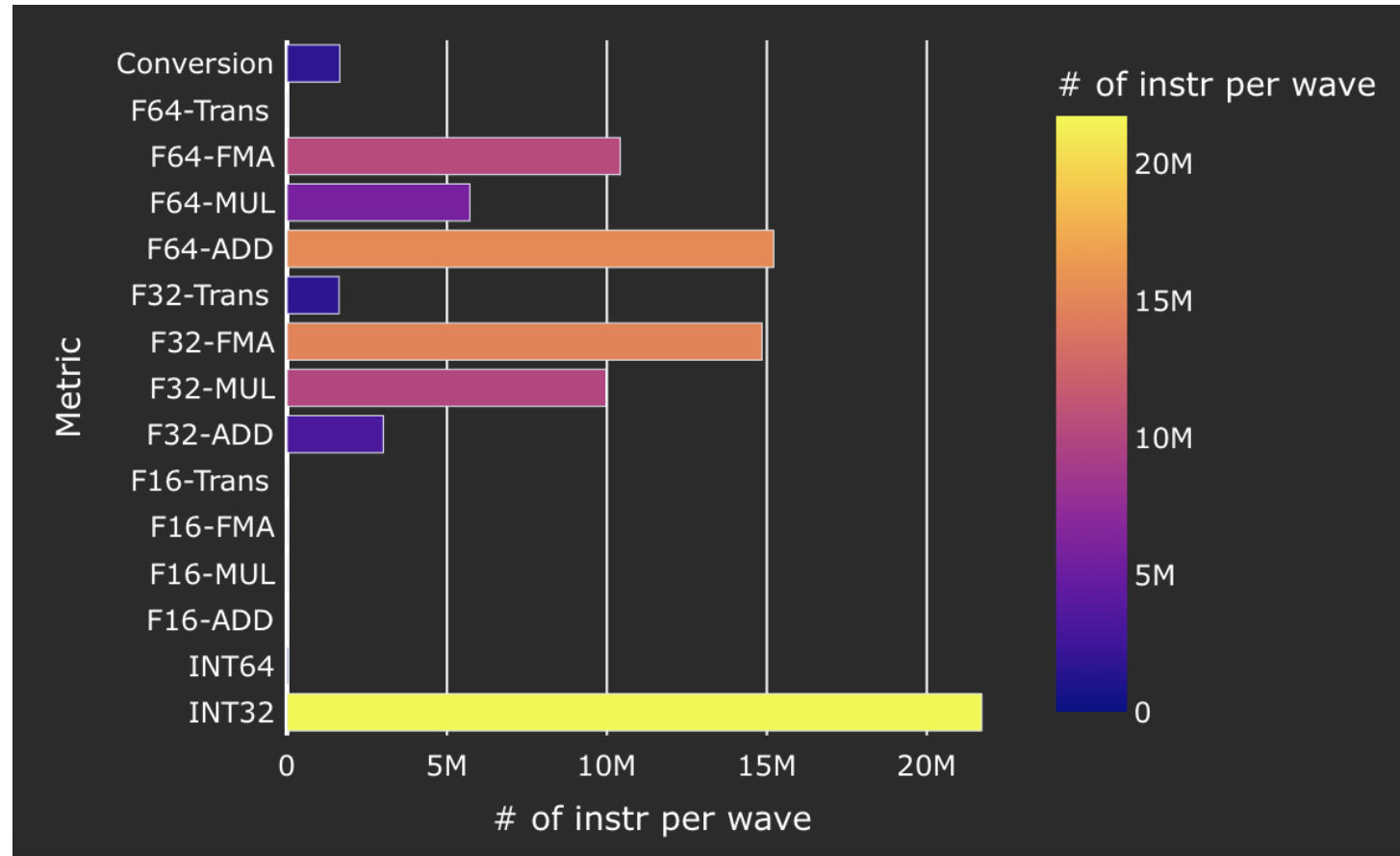
---

- Use rocm/6.2.2 instead of rocm/6.0.3



Function Name	Rocm/6.0.3	Rocm/6.2.2	Performance Increase
momentumEnergy	5.45	5.47	-0.37%
gravity	8.73	6.91	20.85%
iadDivvCurlv	2.74	2.72	0.73%
AVSwitches	2.48	2.36	4.84%
veDefGradh	2.08	2.06	0.96%
xMass	1.83	1.84	-0.55%

# SPH-EXA: LUMI Hackathon Improvements Tried

- Use rocm/6.2.2 instead of rocm/6.0.3 
- Reduce conversion instructions









# SPH-EXA: LUMI Hackathon Improvements Tried

- Use rocm/6.2.2 instead of rocm/6.0.3 
- Reduce conversion instructions 
- Using `_launch_bounds_(256,1)` in the kernels increased the performance of momentumEnergy function

Function Name	Rocm/6.0.3	Rocm/6.2.2 (launch_bounds)	Performance Increase
momentumEnergy	5.45	3.69	32.29%
gravity	8.73	6.91	20.85%
iadDivvCurlv	2.74	2.73	0.36%
AVSwitches	2.48	2.37	4.44%
veDefGradh	2.08	2.06	0.96%
xMass	1.83	1.84	-0.55%

# SPH-EXA: LUMI Hackathon Improvements Tried

---

- Use rocm/6.2.2 instead of rocm/6.0.3 
- Reduce conversion instructions 
- Using `_launch_bounds_(256,1)` in the kernels increased the performance of momentumEnergy function 
- `_restrict_` did not have any effect on momentumEnergyKern when used in all const pointers 
- `__builtin_nontemporal_load` or store did not seem to affect at least these are not used everywhere to check. 
- float2 tried on xmassgpu kernel and performance didn't improve 

# SPH-EXA: LUMI Hackathon Improvements Tried

```
struct TravConfig
{
    /// @brief size of global workspace memory per warp, must be a power of 2
    static constexpr unsigned memPerWarp = 256 * GpuConfig::warpSize;
    static_assert((memPerWarp & (memPerWarp - 1)) == 0);

    /// @brief number of threads per block for the traversal kernel
    static constexpr unsigned numThreads = 128;

    static constexpr unsigned numWarpsPerSm = 40;
    /// @brief maximum number of simultaneously active blocks
    inline static unsigned maxNumActiveBlocks =
        GpuConfig::smCount * (TravConfig::numWarpsPerSm / (TravConfig::numThreads / GpuConfig::warpSize));

    /// @brief number of particles per target, i.e. per warp
    static constexpr unsigned targetSize = GpuConfig::warpSize;

    /// @brief number of warps per target, used all over the place, hence the short name
    static constexpr unsigned nwt = targetSize / GpuConfig::warpSize;

    /// @brief number of blocks to launch, no longer adapts to grids that are too small to saturate all SMs
    static unsigned numBlocks() { return TravConfig::maxNumActiveBlocks; }

    /// @brief compute storage needed for traversal stack
    static unsigned poolSize()
    {
        unsigned numWarpsPerBlock = TravConfig::numThreads / GpuConfig::warpSize;
        return TravConfig::memPerWarp * numWarpsPerBlock * maxNumActiveBlocks;
    }
};
```

256

32

220



# SPH-EXA: LUMI Hackathon Improvements Tried

---

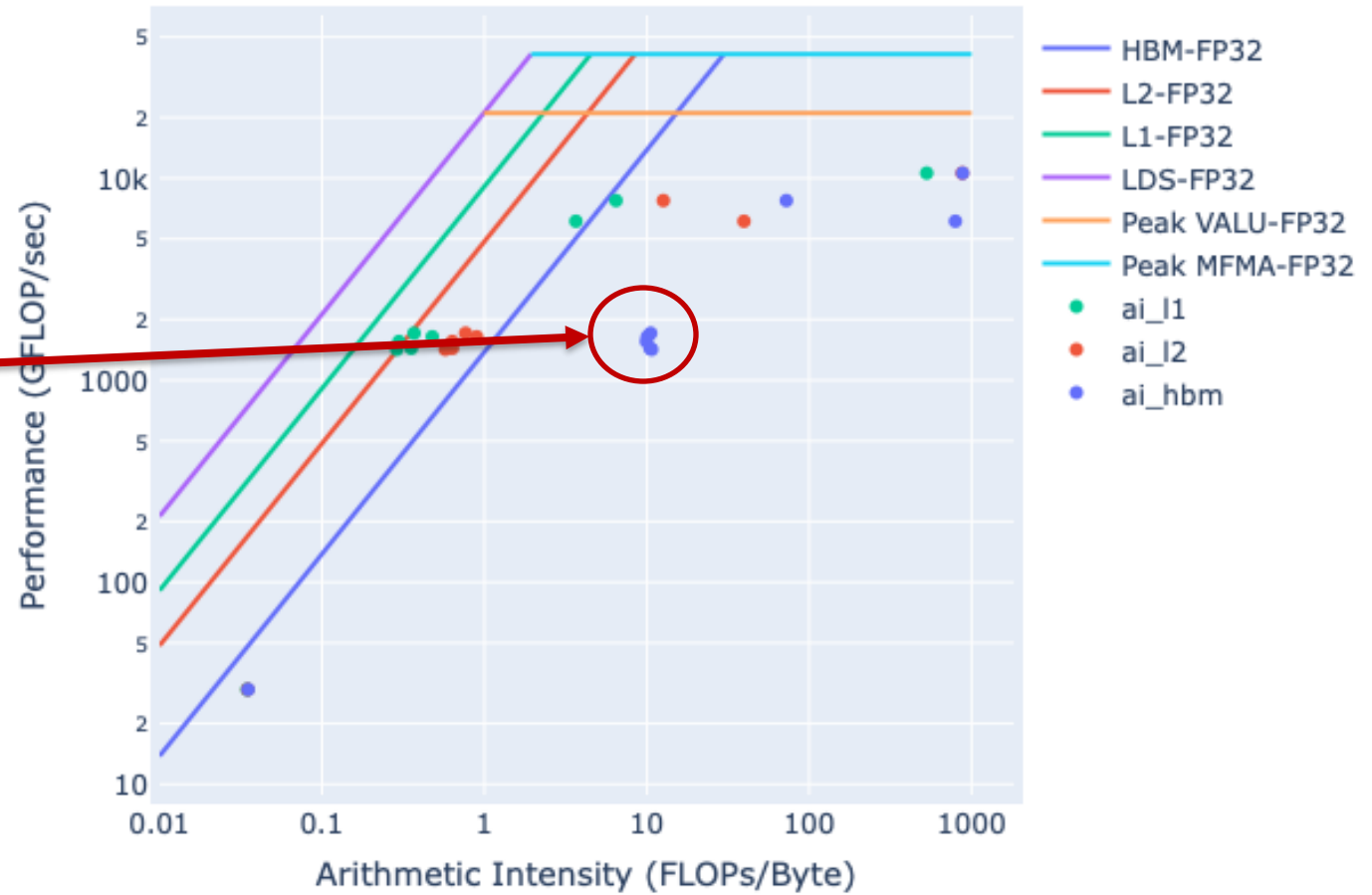
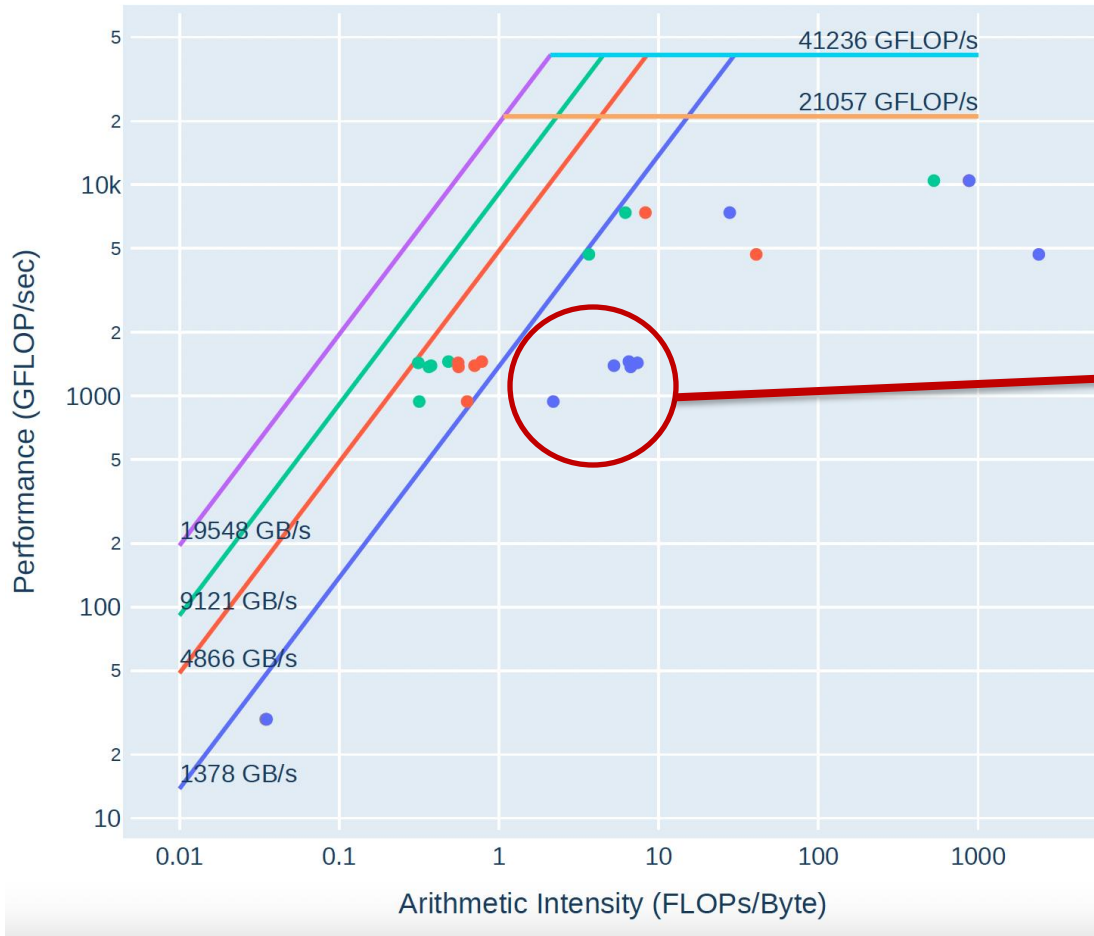
- Updated thread and block values

Function Name	Rocm/6.0.3	Rocm/6.2.2 (launch_bounds)	Performance Increase
momentumEnergy	5.45	3.09	43.30%
gravity	8.73	6.78	22.34%
iadDivvCurlv	2.74	2.51	8.39%
AVSwitches	2.48	1.89	23.79%
veDefGradh	2.08	1.91	8.17%
xMass	1.83	1.6	12.57%

---

# SPH-EXA: LUMI Hackathon Improvements Tried

- Updated thread and block values 



# Where are we now?

---

- AMD MI250X vs Nvidia A100

Function Name	MI250X 1 GCD	A100	MI250X 2 GCDs
momentumEnergy	3.09	1.51	1.54
gravity	6.78	6.82	6.92
iadDivvCurlv	2.51	1.17	1.28
AVSwitches	1.89	1.06	0.95
veDefGradh	1.91	0.95	0.95
xMass	1.6	0.83	0.8

---