Walberla

Jayesh Badwaik, Chelsea Maria John, Radim Vavrik, Kristian Kadlubiak

April 17, 2023

Overview and Code Structure

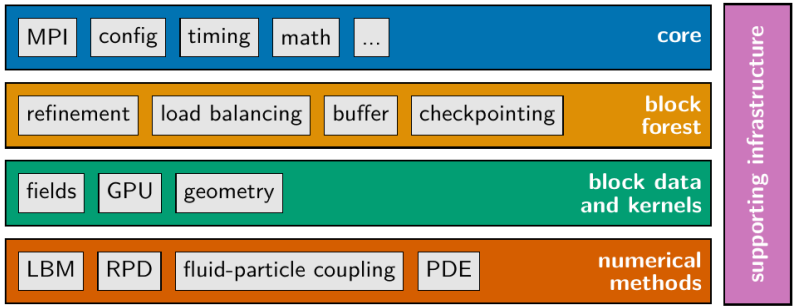


Figure: Structure of the Walberla Framework

Blockforest

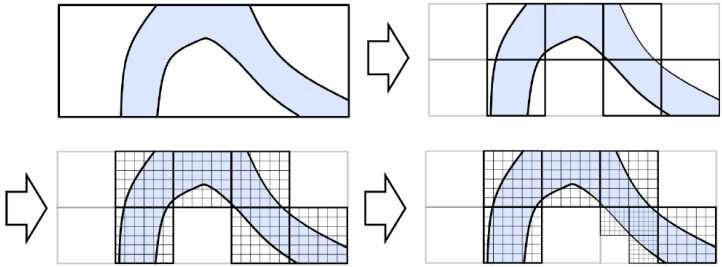


Figure 4: Block-structured domain setup. From left to right: defining domain using a surface mesh, decomposition into coarse blocks, allocation of cells in blocks, block refinement

Figure: Distributed Block Forest Structure

Blockforest

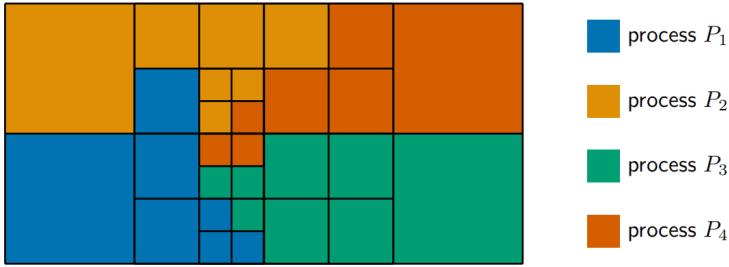


Figure: Distributed Block Forest Structure

- Node-Level Performance: `lbmpy`, `pystencils`

Execution Graph

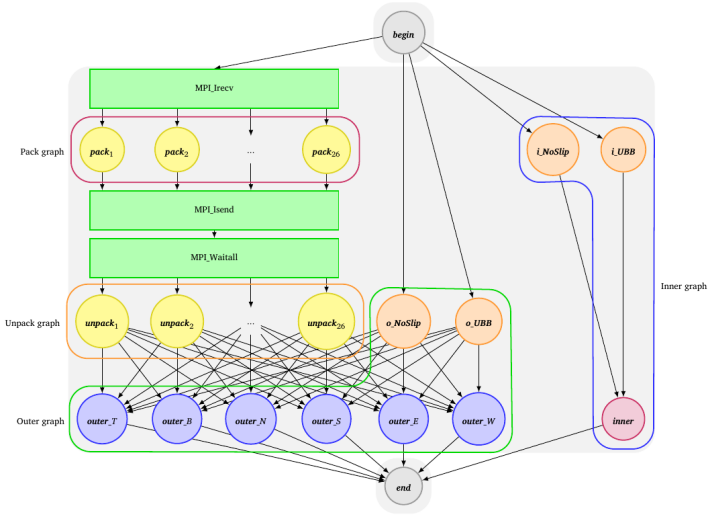


Figure: Execution Graph (With Communication Hiding)

Performance

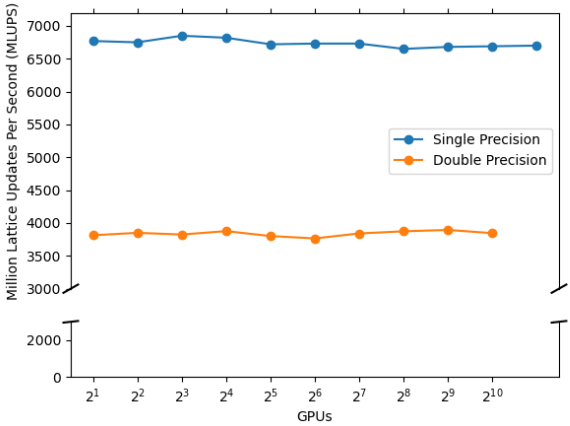


Figure: Performance on Juwels Booster (A100 CUDA)

- Bandwidth-Bound Code ($85 - 90\%$ of the Bandwidth)
- MLUPS -> Million Lattice Updates Per Seconds

Porting and Benchmarking Methodology

Method

- Replace CUDA kernel calls by HIP kernel calls
- Replace CUDA Memory Allocations by Hip Memory Allocations
- No HIP specific optimization

Consequences

- Register use not optimized

Modules Loaded

MI250 on AMD Cloud

- cuda
- python
- rocm-afar
- openmpi
- rocm

Output

```
$ module list
Currently Loaded Modules:
1) StdEnv   2) cuda/11.7.0   3) python/3.9.13   4) rocm-afar/001-732
5) openmpi/5.0.0-rocm-afar001-732   6) rocm/5.4.0
```

Single GPU Results

Block Size	**A100**	**MI250 (Per GCD)**	**SpeedUp**
Bandwidth	1500 GB/s	1638 GB/s	1.05
$64 \times 64 \times 64$	1200 MLUPS	2000 MLUPS	1.65
$128 \times 128 \times 128$	3800 MLUPS	3600 MLUPS	0.95

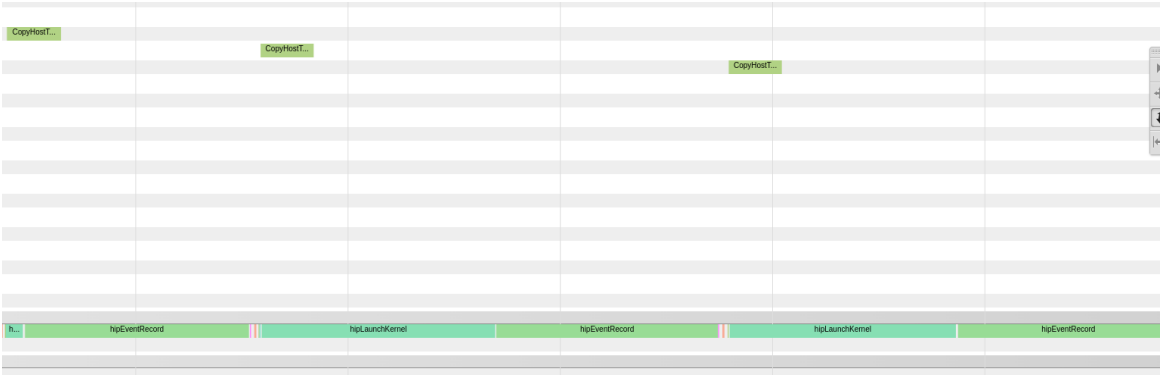
- Launch times are probably much smaller

Multiple GPU Result

- Turning on HIP-Aware MPI Explicitly Doesn't Work
- Depending on PSMPI vs OpenMPI, 50 MLUPS vs 1200 MLUPS
- A different problem from last time

Block Size	**A100**	**MI250 (Per GCD)**	**SpeedUp**
$256 \times 256 \times 256$	3800 MLUPS	1400 MLUPS	0.4
$512 \times 512 \times 512$	3800 MLUPS	2800 MLUPS	0.75

Multiple GPU Result



Bottleneck in AMR

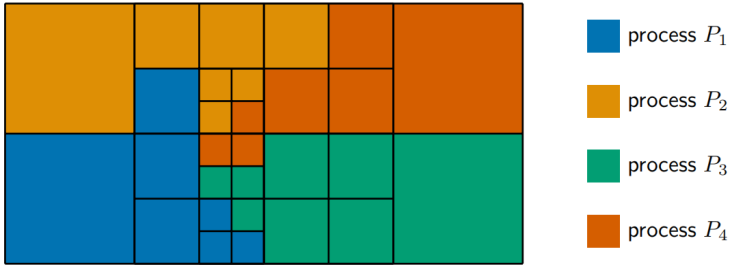


Figure: Distributed Block Forest Structure

Execution Graph

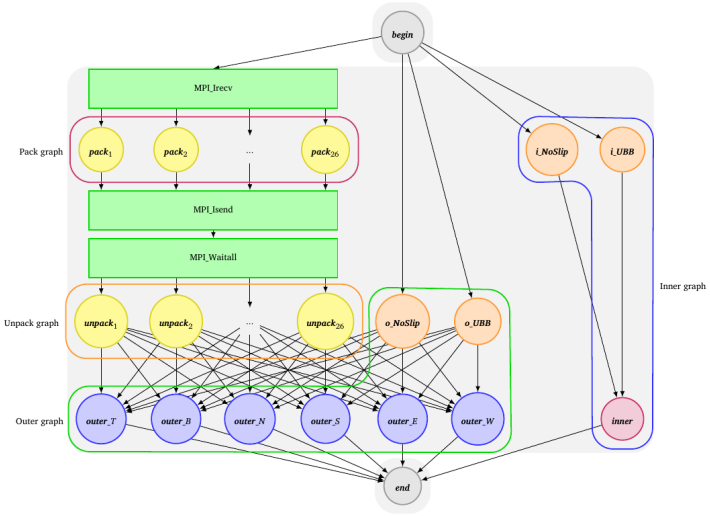
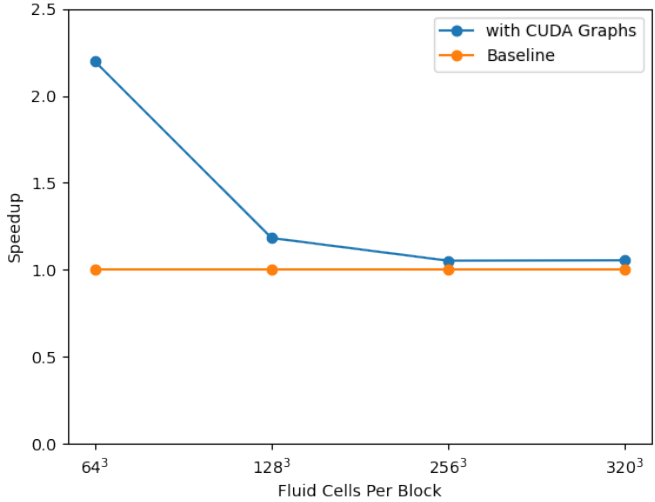


Figure: Execution Graph (With Communication Hiding)

CUDA Graphs


with CUDA Graphs
Baseline
Speedup
2.5
2.0
1.5
1.0
0.5
0.0
64^3
128^3
256^3
320^3
Fluid Cells Per Block

Objectives of this Hackathon

- Run the code on LUMI
  - Code used to run, now just compiles (probably missing some gfx9a flag)
  - Still some work to do.

```
"hipErrorNoBinaryForGPU: Unable to find code object for all current devices!"
```

- Explain Gains in Single GPU Results for Small Block Size
- Evaluate Multinode Performance
- Evaluate Using Hip Graphs for Future
- Performance Profiling and General Optimization
  - Improve GCD based communication?