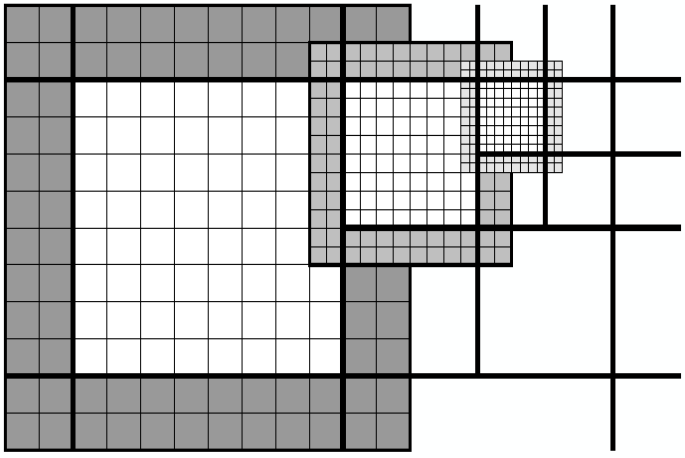
GPU accelerated adaptive wave propagation algorithm

Brian KYANJO Prof. Donna Calhoun

Department of Mathematics
Boise State University

May 18, 2022

ForestClaw

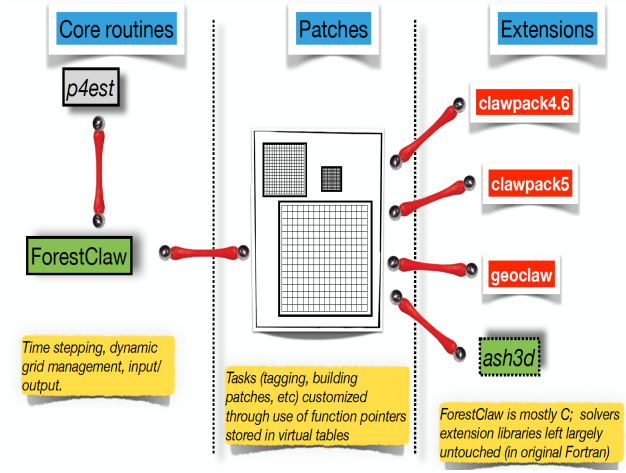


In the ”clawpatch” patch (used for finite volume solvers), each p4est quadrant is occupied by a single logically Cartesian grid, stored in contiguous memory, including ghost cells.

ForestClaw is a p4est PDE layer.

- Written mostly in object oriented C
- Core routines are agnostic as patch data, solvers used, etc.
- Most aspects of the PDE layer, including type of patches used, solver, interpolation and averaging, ghost-filling, can be customized
- Support for legacy codes
- Several extensions include Clawpack extension, GeoClaw, Ash3d and others.
- Current solvers are designed for FV meshes

Code layout



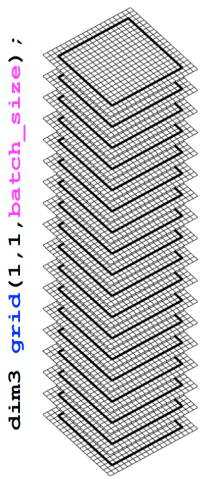
ForestClaw on GPUS

```
block_size = 128;  batch_size = 4000;
mwork = 9*meqn + 9*maux + mwaves + meqn*mwaves;
bytes_per_thread = sizeof(double)*mwork;
bytes = bytes_per_thread*block_size;

dim3 block(block_size,1,1);
dim3 grid(1,1,batch_size);

claw_flux2<<<grid,block,bytes>>>(mx,my,meqn,..)
```

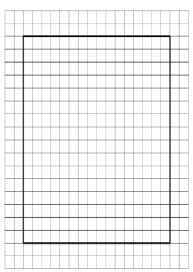
1d thread blocks
3d grid

~4000 patches in a batch

~128 threads per block



One ForestClaw patch per CUDA block



Patch layout with valid ghost cell data

Ported fully unsplit wave propagation algorithm for hyperbolic conservation laws (implemnted in clawpack) to CUDA.

- Copy time solution on all patches to single contiguous block of CPU memory
- Copy contiguous block of CPU memory to the GPU
- Configure the GPU to assign one 1d thread block to each single ForestClaw patch
- Divide Shared memory equally among thread blocks=patches
- All solution data resides in global memory; shared memory is only used for temporary data
- CUDA function pointers used to provide custom Riemann solvers