HydroFunToy overview

Roland Haas
University of Illinois at Urbana-Champaign, National Center for Supercomputing Applications

January 30th 2017

GRHydro scheduling

- static scheduling using one block per MPI rank
- OpenMP parallelization with static scheduling
- multiple (18) Cactus scheduled functions to compute RHS
- SYNC happens in lockstep
- minimal amount of recomputing values

```
GRHydro::SourceTerms: Source term calculation
GRHydro::GRHydroStartLoop: [level] Set the flux_direction vari
while (GRHydro::flux_direction)
  GROUP FluxTerms: Calculation of intercell fluxes
    GRHydro::GRHydro_RefinementLevel: Calculate current refine
    GRHydro::Reconstruct: Reconstruct the functions at the cel
    GRHydro::Riemann: Solve the local Riemann problems
    GRHydro::UpdateCalcul: Calculate the update term from the
    GRHydro::GRHydroAdvanceLoop: [level] Decrement the flux_di
GRHydro::GRHydroUpdateAtmosphereMask: Alter the update terms i
```

HydroFunToy scheduling

- dynamic scheduling using many tiles per MPI rank and task based concepts to work through set of tiles
- overlaps computation and communication
- single Cactus scheduled function to compute RHS
- recomputes quantities multiple times (4x the reconstruction work, twice the prim2con work)

```
HydroFunToy::HydroFunToy_Con2Prim: Convert back to primitive v
HydroFunToy::HydroFunToy_CalcRHS: Compute RHS variables for hy
GROUP HydroBase_PostStep: Post step tasks for hydro thorns
  GROUP HydroBase_Boundaries: HydroBase-internal Boundary conc
    GROUP HydroBase_Select_Boundaries: Group to schedule the b
      HydroFunToy::HydroFunToy_Boundaries: [level] Select Hydr
    ...
  HydroFunToy::HydroFunToy_SyncDone: [level] Fullfill SYNC pro
```

HydroFunToy design

- proof of concept hydro code for FunHPC. Identify how a hydro differs from McLachlan.
- no options or complex methods, keep it simple
- don't worry about pointwise efficiency yet
- single scheduled routine with a pointwise kernel that computes the RHS, looks like GR RHS computation
- re-use inner code from GRhydro where possible
- allow for overlap between computation and communication
- currently all code for a single scheduled function in a single C++ file

HydroFunToy code

- HLLE Riemann solver from C++ GRHydro
- WENO reconstruction from C++ GRHydro
- Tmunu computation from C++ GRHydro
- Prim2Con from GRHydro
- Con2Prim using Wolfang Kastaun's bracketing, derivative-free method
- parallelism via Erik's FunHPC, which is a thin layer around qthreads

Simple code, no options, choose methods that are easy to parallelize over complex methods (eg WENO rather than PPM).

HydroFunToy test

- code compiles on wheeler and my laptop
- need funhpc branch of Carpet, flesh, hydrofuntoy branch of Zelmani
- not yet tested with ML_BSSN_FH
- unigrid only since there is no support in Carpet for task based prolongation yet
- should be possible to port kernel to GPUs fairly straightforwardly, at least much more easily than GRHydro

HydroFunToy status

```
GetComponets --root Cactus_FunHPC \
  https://docs.einsteintoolkit.org/et-docs/images/c/c9/Zelmani.th
cd Cactus_FunHPC
simfactory/bin/sim sync wheeler
simfactory/bin/sim --remote wheeler build \
  --thornlist thornlists/Zelmani.th
simfactory/bin/sim --remote wheeler submit --procs 24 \
  --walltime 0:10:00 \
  --parfile repos/Zelmani/HydroFunToy/test/static_tov.par \
  static_tov00
```