KHARMA: Fast & Flexible GRMHD

Ben Prather, LANL

Parthenon users’ meeting

Los Alamos, 10/3/2022

Requirements

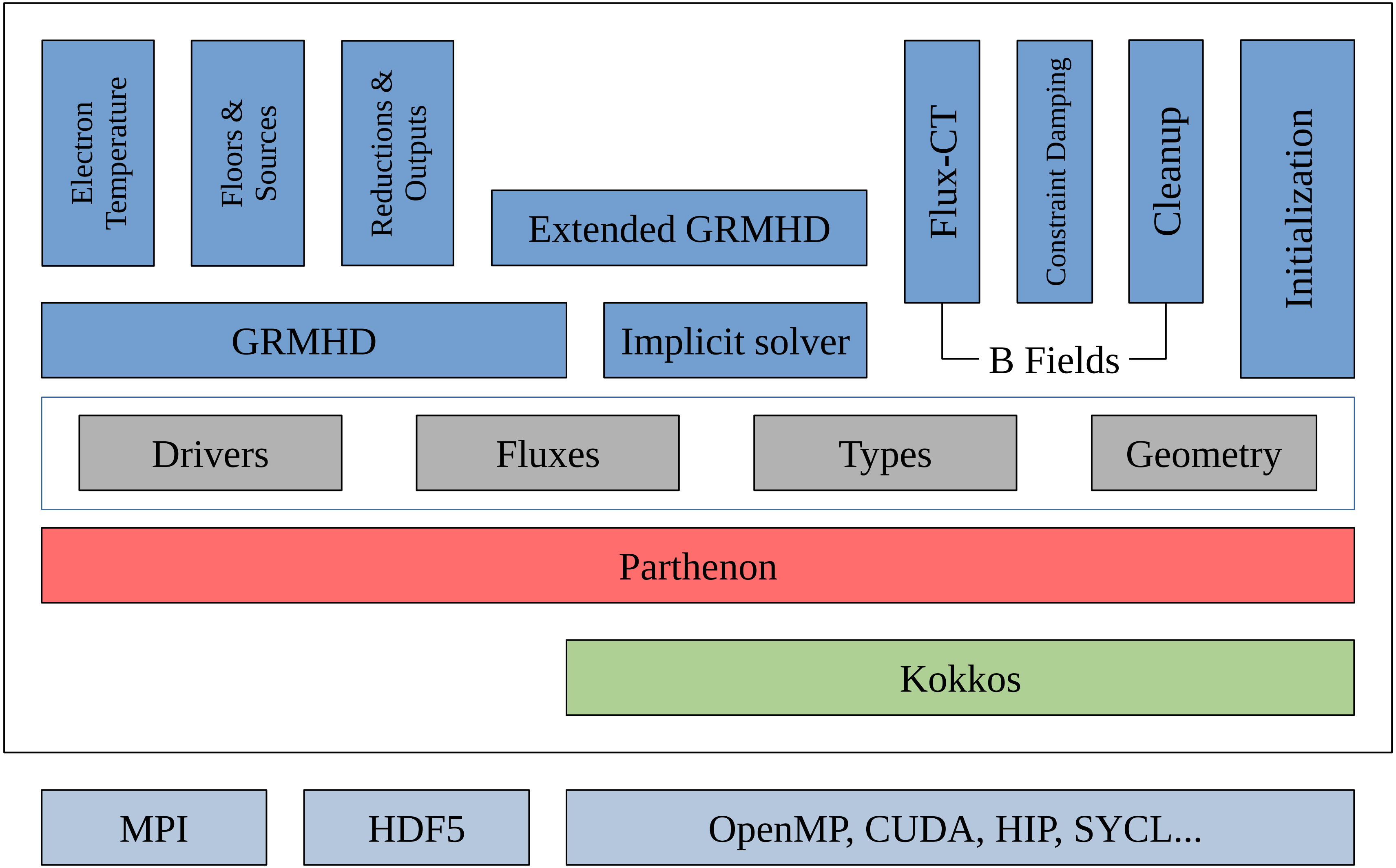
- Many long simulations
  - Best possible single-node speed: must support **accelerators**
- Code longevity
  - Must be **performance-portable** and supported by current- and next-gen machines
- Code simplicity/extensibility
  - Must (after some reading) be **understandable** to the next grad student primarily interested in writing an extension
  - Must be capable of supporting extensions implementing solves, transfer, etc

Algorithm

- HARM: 2nd-order conservative shock-capturing scheme
- LLF fluxes
- (Usually) WENO5 reconstruction
- Any (rotationally symmetric) stationary spacetime background
- Flux-CT transport for B
- 1-dimensional Newton-Raphson primitive variable recovery

Design

- Not overcomplicated: core GRMHD ~5k SLOC (KHARMA is ~10k total)
- Modular: built in layers, split into “packages” with a consistent interface separating common curved-space conservative scheme operations from specific physics (GRMHD and additions)
- Swap/enable/disable modules for different schemes, without paying in performance or cross-talk bugs


Electron Temperature
Floors & Sources
Reductions & Outputs
Extended GRMHD
Flux-CT
Constraint Damping
Cleanup
Initialization
GRMHD
Implicit solver
B Fields
Drivers
Fluxes
Types
Geometry
Parthenon
Kokkos
MPI
HDF5
OpenMP, CUDA, HIP, SYCL…

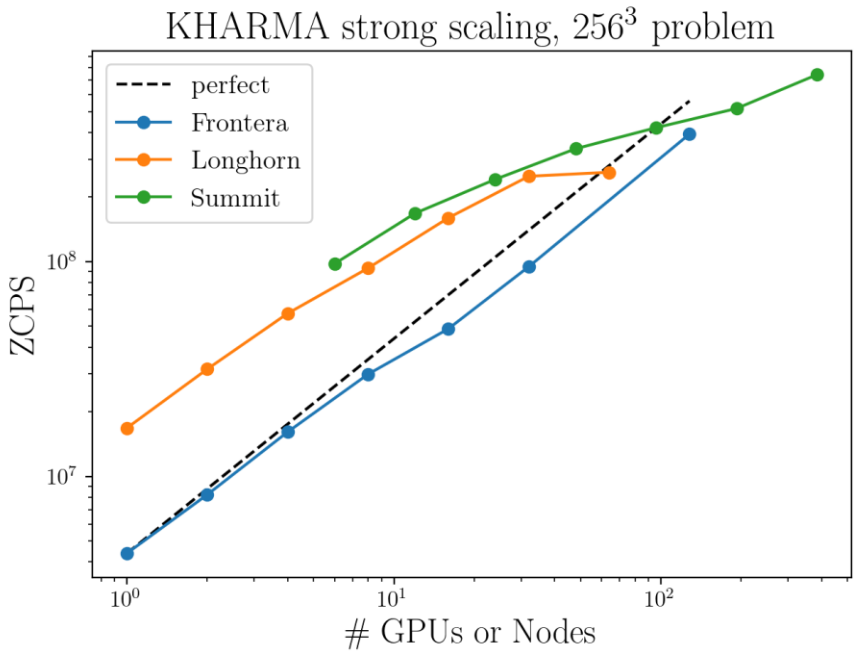
Code Features

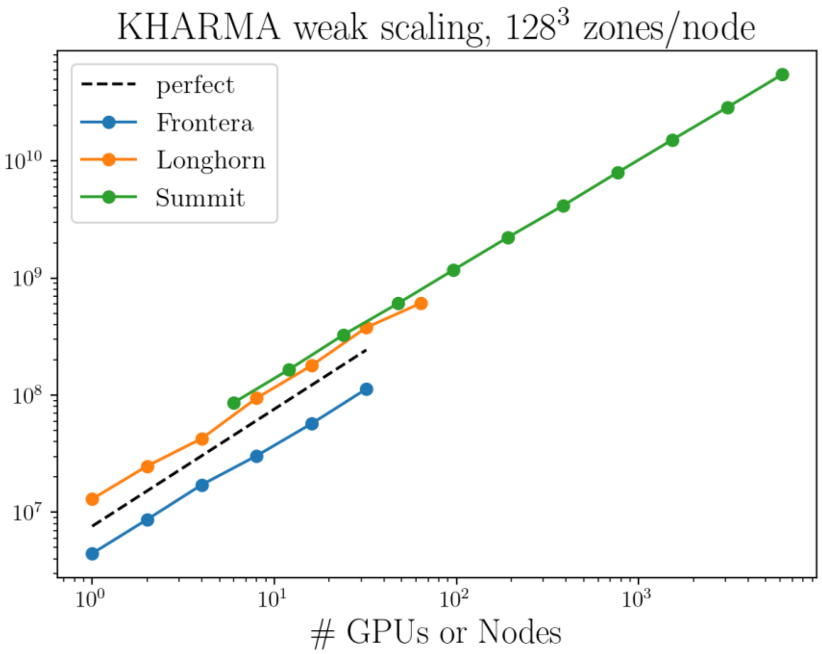
- All options are run-time: much simpler tests and containerization
- “Restart+” features: re-gridding to restart at high resolution, resizing or changing the domain, adding a hotspot, etc.
- Managed build: autodetection and machine configurations for a smoother first build experience
- Documented: introductory wiki, helpful devs

Physics Features: GRMHD+X

- Co-evolved electron temperatures
- Very soon: incorporation of leading-order viscous effects for sparse magnetized system (i.e. low-rate accretion disks). Implicit solver portion usable for other modules.
- Somewhat soon: radiative transfer – primary goal of my time here at LANL

KHARMA: Performance





~20M Zone-cycles/s on V100 running full torus problem

Conclusions

- Pushing accretion simulations forward means comparing a greater **number** and **diversity** of models.
- KHARMA suits this need by being **fast** yet **simple** and **extensible**, using a modern approach

Extra Slides

Performance *Portability*

- Raspberry Pi 3: 40k ZCPS
- Smartphone (Snapdragon 690): 300k ZCPS
- XPS 13 (before thermal throttling): 1.95M ZCPS
- XPS 13 (thermal throttled): 1.1M ZCPS