Porting our astrophysics application to Arm64FX and adding Arm64FX support using kokkos

Patrick Diehl and Gregor Daiß

Louisiana State University and University of Stuttgart

patrickdiehl@lsu.edu and Gregor.Daiss@ipvs.uni-stuttgart.de

02/10/2022

LSU Center for Computation & Technology

STE||AR GROUP

Motivation

At peak brightness, the rare 2002 red nova V838 Monocerotis briefly rivalled the most powerful stars in the Galaxy. Credit: NASA/ESA/H. E. Bond (STScI)

Goal

Simulate the merger and obtain the light curve to understand the observations better:

Multi-physic is need:

- Hydro
- Gravity
- Radiation

Reference

- Tylenda, R., et al. "V1309 Scorpii: merger of a contact binary." Astronomy & Astrophysics 528 (2011): A114.

Overview

Software framework

Octo-Tiger

Astrophysics open source program[1] simulating the evolution of star systems based on the fast multipole method on adaptive Octrees.

Octo-Tiger

Modules

- Hydro
- Gravity
- Radiation (benchmarking)

Supports

- Communication: MPI/libfabric
- Backends: CUDA, HIP, Kokkos

Reference

- Marcello, Dominic C., et al. "octo-tiger: a new, 3D hydrodynamic code for stellar mergers that uses hpx parallelization." Monthly Notices of the Royal Astronomical Society 504.4 (2021): 5345-5382.

[1] `https://github.com/STEllAR-GROUP/octotiger`

Example of a merger simulation

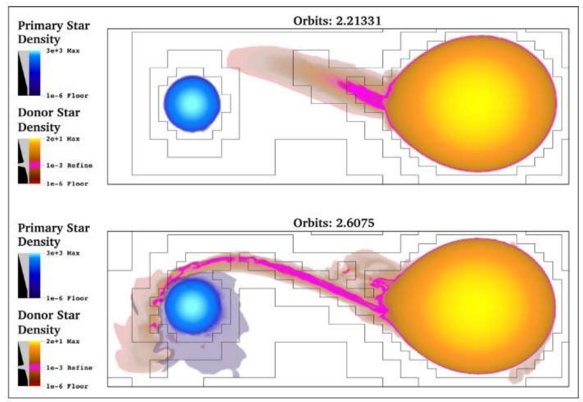


Figure 2. The early stages of mass transfer in a binary star system. The accreting star is five times more massive than the donor star.

Reference

- Heller, Thomas, et al. "Harnessing billions of tasks for a scalable portable hydrodynamic simulation of the merger of two stars." The International Journal of High Performance Computing Applications 33.4 (2019): 699-715.

Example of a merger simulation

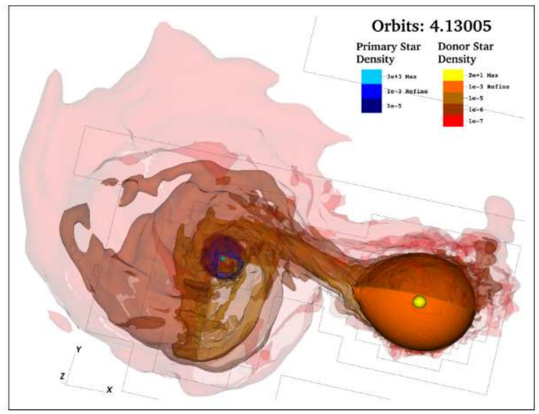


Figure 3. A 3-D contour plot of the system in Figure 2 after an accretion disc begins to form. 3-D: three-dimensional.

Reference

- Heller, Thomas, et al. "Harnessing billions of tasks for a scalable portable hydrodynamic simulation of the merger of two stars." The International Journal of High Performance Computing Applications 33.4 (2019): 699-715.

HPX

HPX is an open source C++ Standard Library for Concurrency and Parallelism[2].

Features

- HPX exposes a uniform, standards-oriented API for ease of programming parallel and distributed applications.
- HPX provides unified syntax and semantics for local and remote operations.
- HPX exposes a uniform, flexible, and extendable performance counter framework which can enable runtime adaptivity.

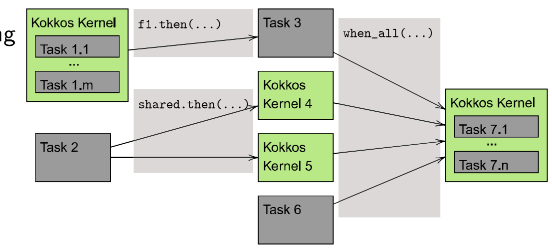
Reference

- Kaiser, Hartmut, et al. "Hpx-the c++ standard library for parallelism and concurrency." Journal of Open Source Software 5.53 (2020): 2352.

[2] `https://github.com/STEllAR-GROUP/hpx`

HPX-Kokkos in Octo-Tiger

- HPX: Combine Tasks via futures in different ways (DAG)
- Octo-Tiger: Uses Kokkos for compute-intensive kernels (running on CPU+GPU)
- Uses HPX-Kokkos integrations (futurized Kokkos kernels to integrate into the dependency graph + kernels as tasks)
- Octo-Tiger's Kokkos kernels use Kokkos SIMD types for explicit vectorization



Reference

- Edwards, H. Carter, Christian R. Trott, and Daniel Sunderland. "Kokkos: Enabling manycore performance portability through polymorphic memory access patterns." Journal of parallel and distributed computing 74.12 (2014): 3202-3216.
- Daiß, Gregor, et al. "Beyond Fork-Join: Integration of Performance Portable Kokkos Kernels with HPX." 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2021.

Scaling Results on Ookami

Test Setup

Purpose of the Test:

- Basic check for (node-level, distributed) scaling with HPX on ARM
- Basic check for SIMD speedup using the Kokkos SIMD (NEON) types

Test Setup

Purpose of the Test:

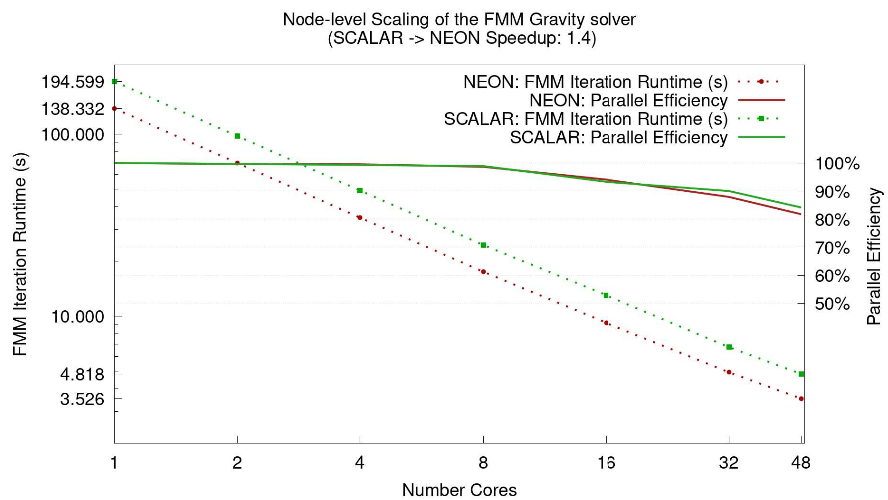
- Basic check for (node-level, distributed) scaling with HPX on ARM
- Basic check for SIMD speedup using the Kokkos SIMD (NEON) types

Setup:

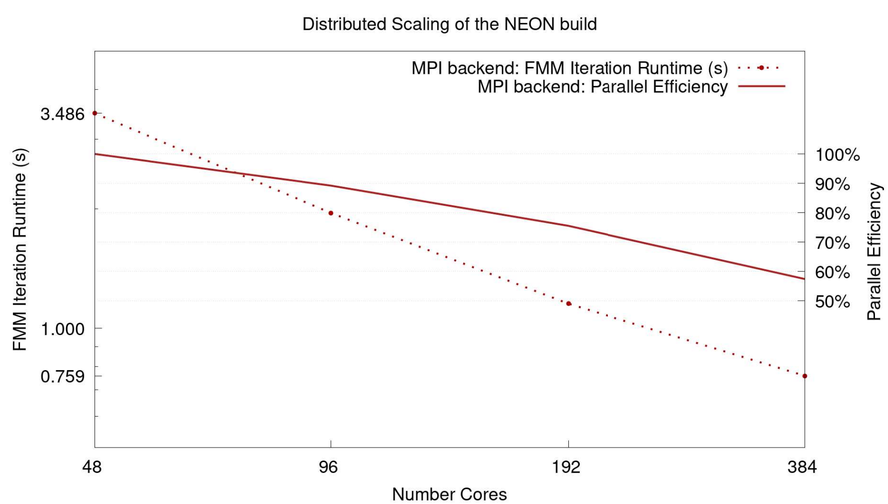
- Gravity-only scenario ("solid sphere") using the Fast Multipole Method (FMM)
- Runtime averages over 10 FMM iterations
- Level 5 AMR oct-tree, each node has one sub-grid of the size 8x8x8
- Overall, using 7625 Subgrids with 512 cells each

→ **Small scenario that can still be run on one core for scaling tests**

Node-Level on 1 ARM64FX Node



Distributed on 1 to 8 Nodes



Conclusion and Outlook

Conclusion and Outlook

- Octo-Tiger, HPX and the rest of the toolchain work on Ookami
- Node-Level scaling look as expected, distributed scaling needs more (larger) tests
- Kokkos SIMD implementation works using the NEON types
- Kokkos SIMD implementation does not work using the SVE types (yet)

Outlook

- Scaling results with the new Kokkos/HPX implementation
- Get Octo-Tiger to properly use SVE and vectorize more methods
- Benchmark the radiation and port to Kokkos
- Optimize for AMD GPUs

Thanks for your attention! Questions?