Octree Frontend for Enzo-E in YT

Bolun Thompson
2023 Summer REU Extra Help Intern

Matthew Turk
Assistant Professor at the School of Information Sciences

June 20th, 2023

Goal

`yt` currently interprets Enzo-E data in a grid-based format, despite the data natively being octree-based. The goal of this project is to write a frontend usable in the real-world to load Enzo-E data into `yt` in an octree-based format.

Background

`yt` is a software package for the analysis of volumetric data [1]. It has been used for the analysis of data in a variety of domains, including astrophysics, seismology, and molecular dynamics. It can interpret data stored as particles, unstructured meshes, and, of note here, both grid-based and octree-based adaptive mesh refinement (AMR) data. It uses a variety of frontends to load datasets from various simulation codes.

Enzo is a grid-based parallel adaptive mesh refinement simulation code for the investigation of astrophysical phenomena such as the formation of stars and galaxies [2]. Enzo-E is a branch of the Enzo code rewritten to use Cello, a highly scalable parallel array-of-octree AMR framework [3].

Problem

The eventual purpose of Enzo-E is to run exascale astrophysical simulations; however, the current Enzo-E frontend for `yt` restricts the analysis of the generated datasets of size 2048^3 blocks ($\approx$ 1 TB) due to performance bottlenecks. The primary performance bottleneck is loading the data and building an index of that data.

One reason for this is that the current Enzo-E frontend interprets the octree data as a collection of grids, not as an array of octrees—a legacy of Enzo's grid-based design. Building the index for a grid-based format necessitates the inefficient instantiation of a Python object for each grid, in contrast to Octs, which are instead efficiently indexed as structures at most 88 bytes large. This leads to octree-based frontends generally being faster to load than grid-based frontends, motivating our project to implement an octree-based Enzo-E frontend for `yt`. However, further work may be necessary to optimize the new frontend to take advantage of the new design.

Objectives

1. Write and test a frontend for `yt` to load Enzo-E datasets in an octree format
2. Optimize the frontend so that it can be used for real world exascale datasets
3. Demonstrate that the octree frontend can be used in a real-world style problem

Previous Work

In the summer of 2022, preliminary work was done refactoring the `yt` code to support loading in Enzo-E data as an octree, as well as exploratory scripts.

Future Work

The Flash and GAMER frontends are other examples of octree-based simulation codes where `yt` loads their datasets as grids, not as octrees. Scaling issues may also impact these frontends, so future work could be to rewrite those frontends using the octree facilities.

References

1. **yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data** Matthew J Turk, Britton D Smith, Jeffrey S Oishi, Stephen Skory, Samuel W Skillman, Tom Abel, Michael L Norman (2011-01) https://arxiv.org/abs/1011.3514 DOI: 10.1088/0067-0049/192/1/9

2. **ENZO: AN ADAPTIVE MESH REFINEMENT CODE FOR ASTROPHYSICS** Greg L Bryan, Michael L Norman, Brian W OShea, Tom Abel, John H Wise, Matthew J Turk, Daniel R Reynolds, David C Collins, Peng Wang, Samuel W Skillman, ... Yuan Li and *The Astrophysical Journal Supplement Series* (2014-03) https://doi.org/10.1088%2F0067-0049%2F211%2F2%2F19 DOI: 10.1088/0067-0049/211/2/19

3. **Computational cosmology and astrophysics on adaptive meshes using charm++** James Bordner, Michael L Norman (2018) https://arxiv.org/abs/1810.01319