Efficient Programming of HPC Systems: A Study on the AMReX Framework

Luis Traffa

May 25, 2023

Contents

1 Introduction

1.1 Area and Problem

HPC systems are used to quikcly solve mathematical problems. Often times these problems are near impossible to solve in a reasonable amount of time. Therefore, these systems employ techniques from numerival analysis to give approximate solutions, instead of exact ones.

One such technique is called adaptive mesh refinement (AMR). It is used during simulations of various natural phenomena, such as weather, climate, and astrophysics. The idea is to use a coarse mesh to simulate the phenomena, and then refine the mesh in areas of interest. This allows for a more accurate simulation, while keeping the computational cost low.

1.2 AMReX Framework

The AMReX framework is a software library that provides a set of tools to implement AMR algorithms. It is developed by the Center for Computational Sciences and Engineering (CCSE) at the Lawrence Berkeley National Laboratory (LBNL). The framework is written in C++ and is open source. It is used in a variety of projects, such as the Exascale Computing Project (ECP) and the Energy Exascale Earth System Model (E3SM).

This framework aids in developing block-strucutred AMR algorithms by providing a set of tools to solve systems of partial differential equations and manage data structures and the parallelization of the algorithms. It is designed to be highy performant, easy-to-use, flexible and portable across different architectures. Hence, most of the parallelization was abstarcted away, to enable the users to focus on the mathematical and physical aspects of their problems. However, users can still access lower level features if the need arises.

2 Background

2.1 Vocabulary

- Meshes: A mesh is a geometrical representation of a physical domain, often a space, which is broken down into small, discrete elements. These elements can be in various shapes such as triangles, rectangles, or hexagons in 2D, and tetrahedra, prisms, or hexahedra in 3D. These small units are interconnected by nodes or vertices, forming a network that covers the entire domain. Mesh is typically used in numerical simulations, such as Finite Element Method (FEM), Finite Volume Method (FVM), or Finite Difference Method (FDM), to discretize a continuum domain for solving complex mathematical problems.

- Partial Differential Equations (PDEs): A PDE is a type of differential equation that contains unknown multivariable functions and their partial derivatives. PDEs are used to formulate problems involving functions of several variables, and are prevalent in physics and engineering. They are often used to describe wave propagation, heat diffusion, fluid flow, or quantum mechanics, amongst others.

- Mesh Simulation: Mesh simulation involves the use of computational methods to solve equations over the domain represented by the mesh. This can involve a va-

riety of different equations depending on the problem at hand, but often involves the solution of PDEs. The domain of interest is discretized into a mesh, and the equations are solved at each node or cell of the mesh. The goal of the simulation is often to predict physical behavior or solve complex engineering problems.

- (Adaptive) Mesh Refinement: Mesh refinement, or adaptive mesh refinement (AMR), is a computational technique used to adaptively refine a computational mesh. The concept behind mesh refinement is to use fine mesh (small cells) in areas where the solution has high gradients or requires high accuracy, and coarse mesh (large cells) in areas where the solution is smooth or less important. This technique allows the efficient use of computational resources, improving the accuracy of the solution while reducing computational cost.

2.2 Meshes and PEDs

The relationshiop between meshes and PEDs is at the center of many numerical methods in computational physics and engineering. Therefore it is important to understand how the concepts relate to one another.

2.3 Programming Models

2.4 Algorithms

2.5 Hardware Architectures

3 AMReX: Approach

3.1 Introduction to AMReX

3.2 Efficiency Techniques

Several techniques for efficient usage of resources are used in AMReX.

- Adaptive Mesh Refinement: AMReX uses a block-structured AMR, which selectively refines regions of the computational grid that require more resolution. This technique reduces computational cost by allocating resources only where needed.

- Parallelization: AMReX has strong support for parallelization with both MPI (Message Passing Interface) and OpenMP, allowing it to run efficiently on distributed-memory and shared-memory systems. It uses a combination of space-filling curve (SFC) based algorithms and two-level hierarchical algorithms for parallelization.

- Load Balancing: AMReX uses a dynamic load balancing algorithm to ensure that computational work is evenly distributed across all available processors, thus minimizing idle time and enhancing overall performance.

- Vectorization: AMReX supports the use of modern many-core and multi-core architectures and uses vectorization to exploit these architectures to the fullest.

- Asynchronous I/O: AMReX supports parallel I/O, which is particularly useful when dealing with large datasets.

3.3 Portability Across Architectures

One of AMReX's strengths is its ability to provide portability across a wide variety of computing architectures. It has been successfully deployed on a variety of high-performance computing systems, including CPUs and GPUs, and is designed to work seamlessly with upcoming exascale architectures.

4 Evaluation

4.1 Publication 1

4.2 Publication 2

4.3 Publication 3

4.4 Publication 4

5 Discussion

5.1 Advantages of AMReX

5.2 Drawbacks of AMReX

5.3 Comparison with Other Approaches

6 Summary and Outlook

6.1 Key Findings

6.2 Future Research Directions