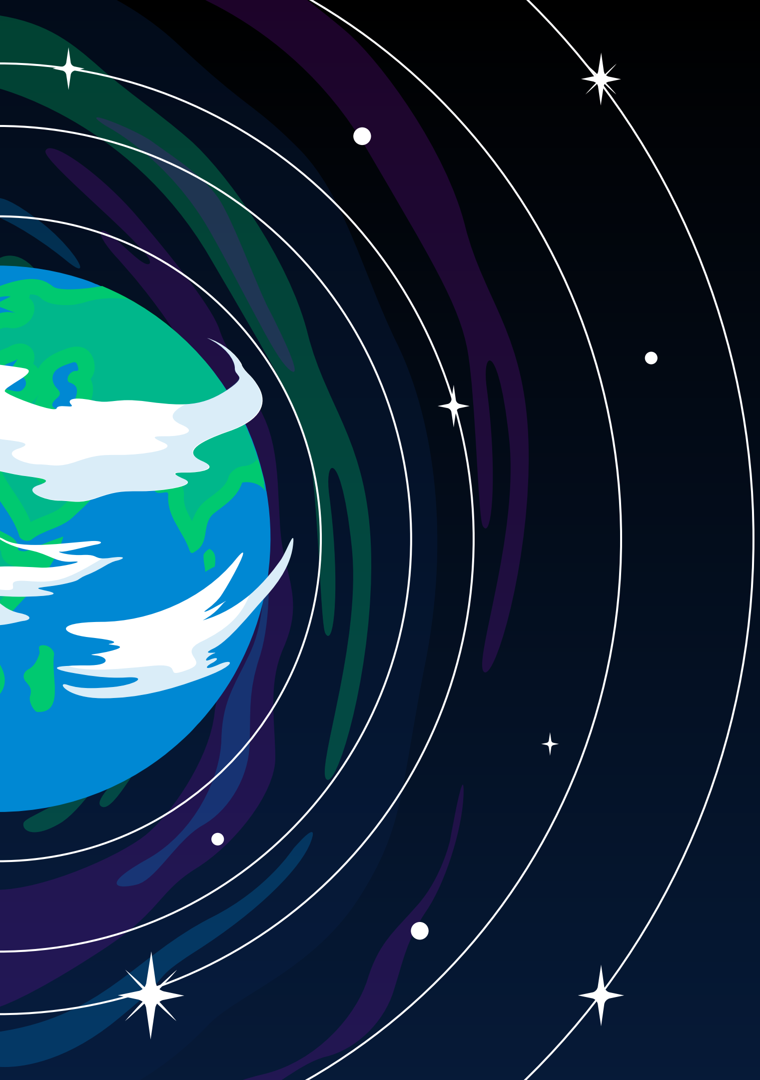
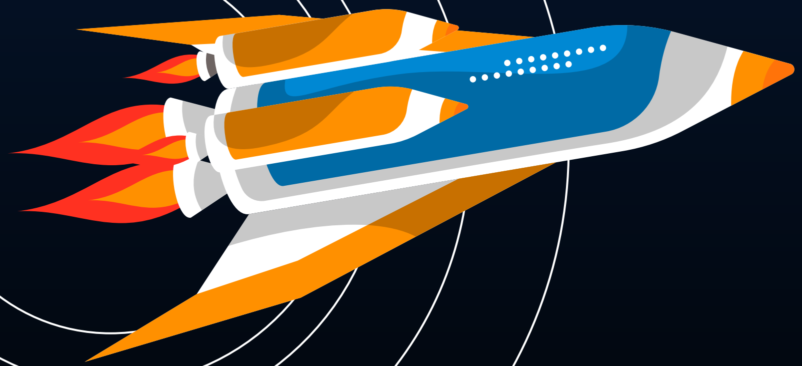


Binary Black Hole Mergers

Charles Njoroge

TABLE OF CONTENTS

INTRODUCTION



A binary black hole is a system of two black holes that orbit closely

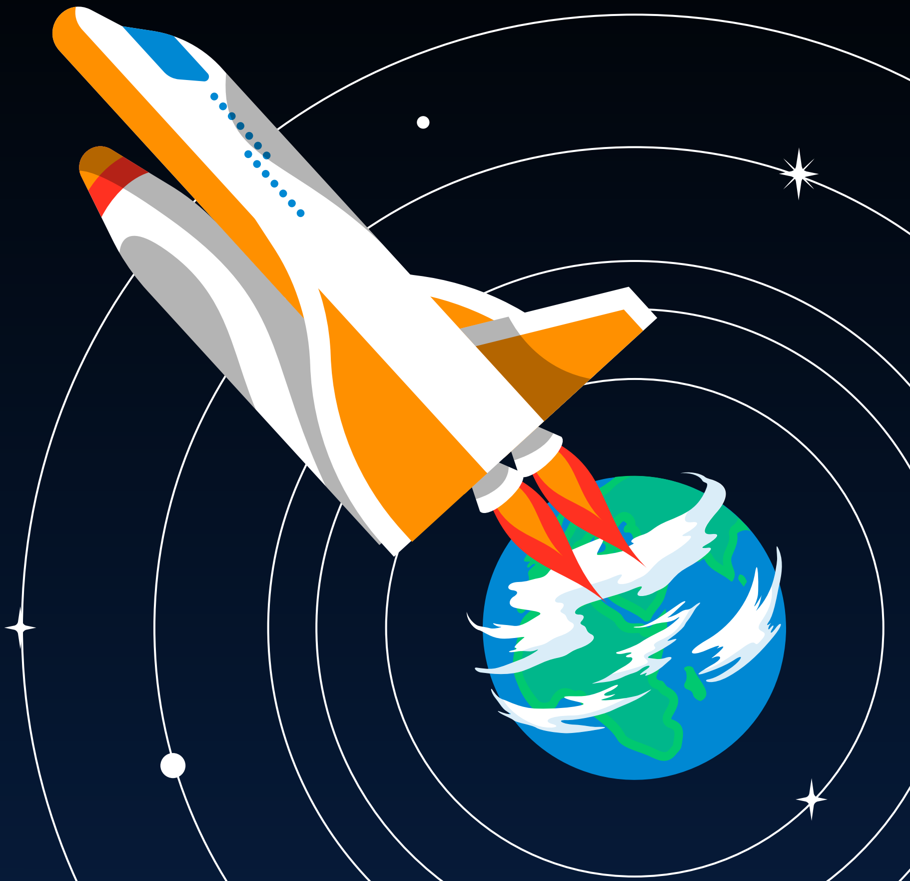
Discovering black holes was and is very challenging due to the nature of black holes

Gravitational waves are 'ripples' in space-time produced by some of the most violent events in the cosmos, such as the collisions and mergers of massive stars.

WHOA!

First detected by Laser Interferometer Gravitational-Wave Observatory (LIGO) detected that picked up on the distinct gravitational signature of GW150914 **1.3 billion light-years away**

For the final **20 ms** of spiraling inward and merging, **GW150914** released around **3 solar masses** as gravitational energy, which is **more wattage** than the **combined power of all light radiated** by all the stars in the **observable universe** put together



Experimentation Review

Einstein Toolkit

A comprehensive, open-source computational infrastructure for relativistic astrophysics.

GW150914 Simulation

Simulates the gravitational wave event **GW150914** by modeling the dynamics of two merging black holes

Core tools

Utilizes specialized modules like T**wo Punctures** for initial data generation, **Carpet** for mesh refinement, **AHFinderDirect** for horizon tracking, and **McLachlan** for solving Einstein's equations.

Grid / Refinement

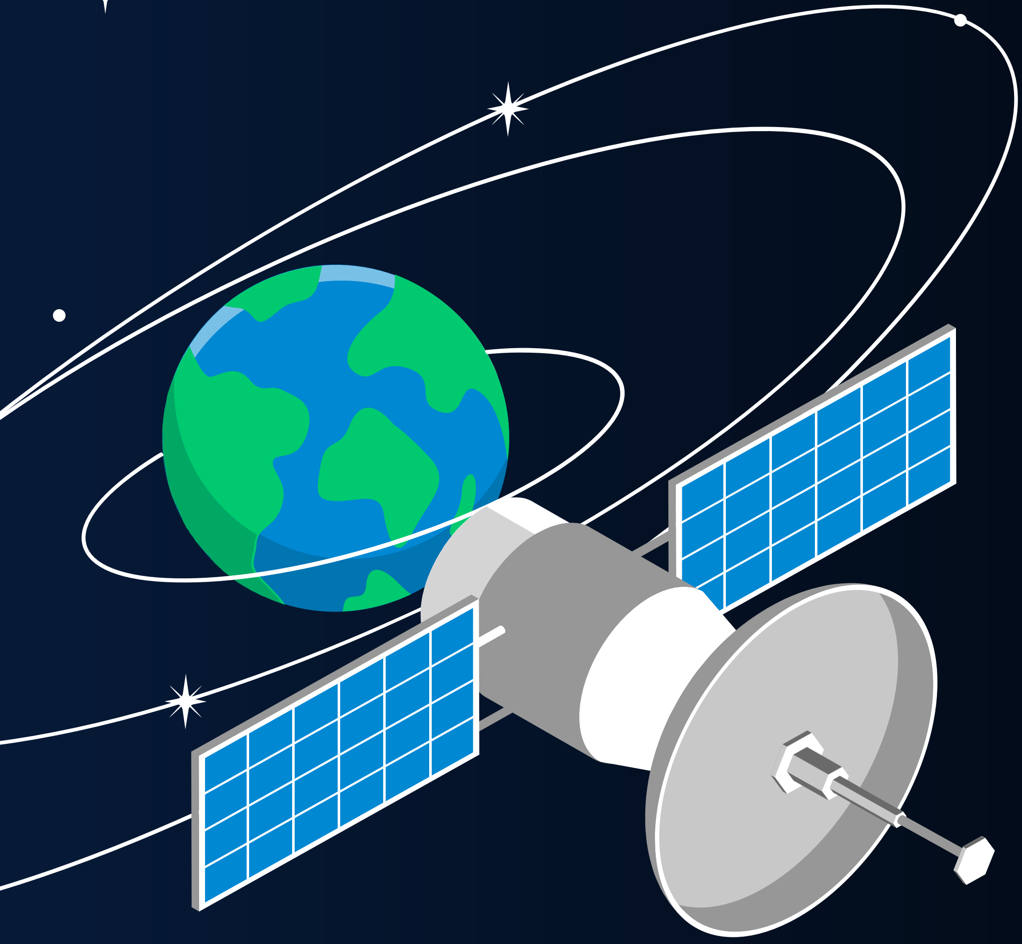
Defines a multi-level grid structure with dynamic mesh refinement to capture high-resolution

Outputs

Generates data, including gravitational waveforms, apparent horizon metrics, and spacetime curvature,

Validation

Incorporates symmetry checks and parameter logging to promote robust scientific research



Results

Pending

simulations take 3-4 days to run

Current Setup

3 caslake nodes , with ~128 cores

Discussion

- Thinking through how GPUs can be used to speed up the calculations due to mesh refinement
- Grid Calculations: GPUs can speed up solving equations on a grid, like calculating derivatives in spacetime.
- Time Stepping: Methods like Runge-Kutta, numerical analysis iterative methods, for advancing the simulation in time are ideal for GPU acceleration because each step can run in parallel.
- Wave Signal Analysis: Extracting gravitational waves (e.g., \(\Psi_4 \)) from the simulation, which involves lots of math, runs faster on GPUs.
- Horizon Finding and Interpolation: Detecting black hole horizons and interpolating data between grids are heavy tasks that GPUs handle efficiently.