Parallel In-situ querying over AMR data in scientific simulations

Presenter: Kushal Bansal, NC State University, kbansal@ncsu.edu
Mentors: Suren Byna, John Wu, Lawrence Berkeley National Laboratory, (sbyna, kwu)@lbl.gov

PROBLEM STATEMENT

Querying the AMR (Adaptive Mesh Refinement) data generated by a scientific simulation during the run (in-situ) in a parallel fashion.

INTRODUCTION

- Adaptive Mesh Refinement (AMR) data structure is compute and memory efficient for large array(uniform mesh) scientific datasets.
- Widely used by scientific applications in climate, astrophysics domains e.g. BISICLES[1], MAESTRO[2].

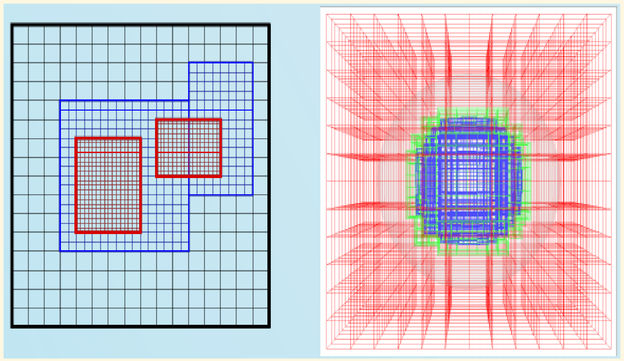


Fig 1: AMR structure [3]

- No existing framework to query AMR data as and when it is being generated by simulation. This capability is very essential for following reasons:
  - ➢ Efficient tracking of scientific phenomenon in real time.
  - ➢ Steering the simulation as per generated data (interactive queries)

CHALLENGES

- Minimal existing work for in-situ context.
- Least performance overhead to the simulation.
- Scalability especially for parallel simulations.

APPROACH

- Identify use-case applications.
- Study the requirements by collaborating with application users.
- Develop specific solutions and generalize.

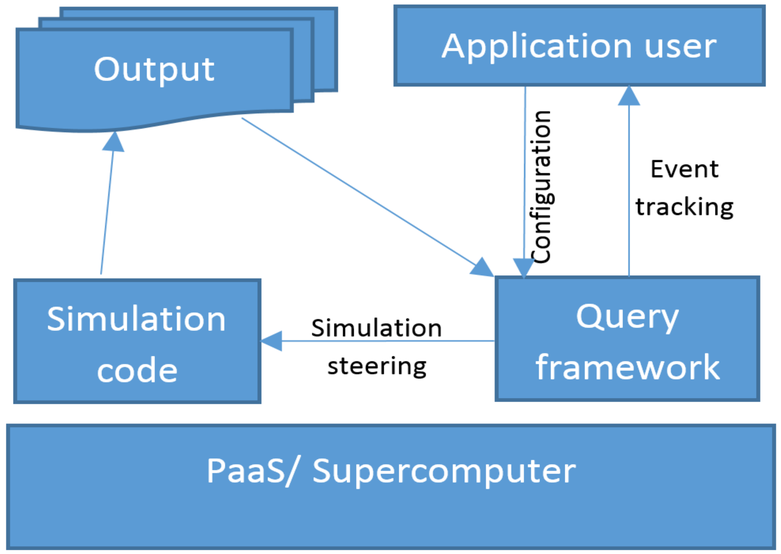
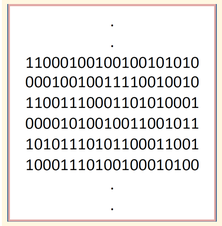


Fig 2: AMR Parallel In-situ Query Framework

REFERENCES:
[1] BISICLES: AMR ice-sheet modeling, http://bisicles.lbl.gov/
[2]MAESTRO: Low Mach Number Astrophysics, https://ccse.lbl.gov/Research/MAESTRO/index.html
[3] Block-structured AMR: Algorithms & Applications (https://www.orau.gov/SciDAC3PI2012/presentations/Posters_Day1/A02_Almgren.pdf)
[4]Marchok, T., 2010: Use of the GFDL vortex tracker. *WRF Tutorial for Hurricanes*

WORK DONE

- Identified Tropical Cyclone detection and tracking as one of the use-case application.
- Requirements study
  - Collaborating with application developers/users.



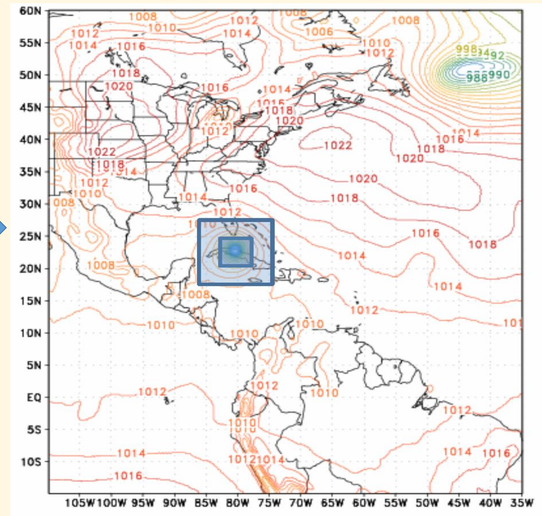


Fig 3: Vortex (storm center) tracking [4]

IN PROGRESS

- Documentation study
  - Knuston et al (2007): Atmospheric model to simulate TC activity in Atlantic.
  - GFDL vortex tracker (2010): Average over Barnes analysis and Linear extrapolation for next lead time.
- Study of TECA code
  - Parallel implementation of TC tracking algorithm in Knuston et al (2007).
  - Input: List of netCDF files. Configuration based on total timesteps, grid spacing.
  - Outputs detailed TC statistics. Trajectory analysis to separate results by year.
- Simulation code study.