Optimizing the data-analysis codes for future CMB experiments

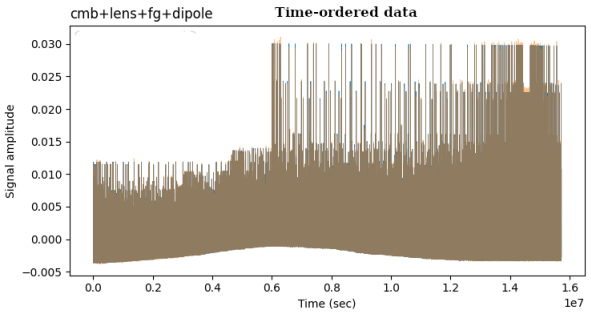
Avinash Anand

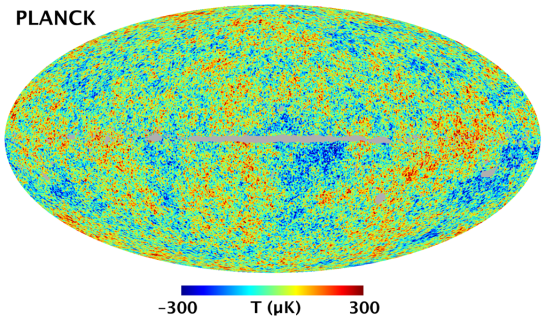
AASSPhD
38^{th} cycle
University of Rome "Tor Vergata"

Supervisor:
Dr. Giuseppe Puglisi
University of Catania

June 14, 2023

Introduction





Contents

CMB map-making

Introduction:

- Map-making is a data-reduction problem: from time-ordered data to maps
- Methods are based on Generalized Least Squares approach
- Data reduction need to handle matrices of size $\mathcal{O}(10^9 \times 10^9)$
- Computation involves complex operations like matrix-vector multiplication and inversion or preconditioner

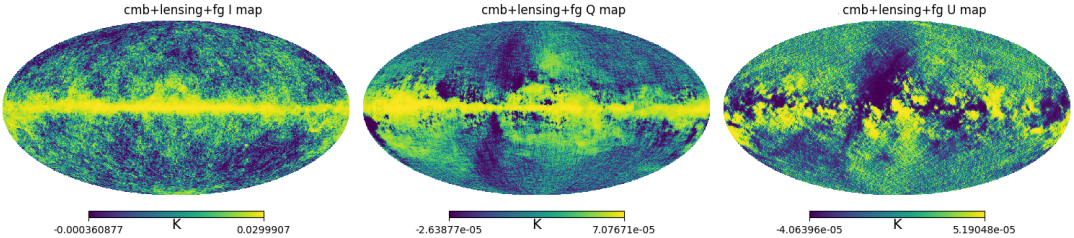
Goal:

- Optimization of ROMA (De Gasperis et al. 2003, Natoli et al. 2000) and Sanepic (Patanchon et al. 2010) ; adding the interface to end-to-end simulation pipeline
- Offload repetitive matrix operations on GPU

Contents

fgcluster

- Map-making algorithms produce I, Q, U maps - include CMB, Galactic foregrounds and imprints of systematic effects



- We would like to partition the full sky into multiple domains obtained with Clustering techniques `fgcluster` (Puglisi et al. 2021, Carones et al. 2023)

Bottlenecks:

- Spectral proximity of pixels are computed in the matrix of size $(npix, npix)$
- The size of matrix scales by power of 4 as the `nside` parameter of the healpix map is increased
- So far the code runs on maps with coarse pixelizations ($n_{pix} < 10^4$, need to identify clusters with $n_{pix} \sim 10^6$
- Problem becomes data intensive - MPI based distribution - large communication overhead

Solution: Dask[1] - A python library for parallel computing

[1] https://docs.dask.org/en/stable/

fgcluster - Dask-based optimization

Why Dask?

- Dask data objects are collection of coordinated numpy arrays
- Out-of-box support for most of the `numpy` API functions
- **Out-of-memory** computation support
- Distributed computing environment - Support for multiple processes and nodes with implicit TCP based communication
- Automatic dynamic task scheduling - implicit load balancing

fgcluster - Status

- Successfully ported most of the code to Dask
- So far, no significant difference in computation time (tested on low $n_{pix} \sim 10^3$)
- Work in progress on implementing/finding alternatives for sparse matrix operations

Thank You!