PyStormTracker: A Parallel Object-Oriented Cyclone Tracker in Python

Stony Brook University School of Marine and Atmospheric Sciences

NCAR National Center for Atmospheric Research

Albert M.W. Yau[1]*, Kevin Paul[2], John Dennis[2]

[1] School of Marine and Atmospheric Sciences, Stony Brook University, [2] Computational and Information Systems Laboratory, National Center for Atmospheric Research *albert.yau@stonybrook.edu

ABSTRACT

Extra-tropical cyclones moving along storm tracks are responsible for much of the day-to-day weather variability and extreme weather events at mid-latitudes. Hence, it is of great interest to study and understand cyclone track dynamics and variability. With the availability of large ensemble datasets, such as the CMIP5 archive, it is critical that any cyclone tracking program used to study these dynamics is capable of processing such large datasets.

We have begun development on a parallel Python package, named PyStormTracker, which locates cyclone centers from model outputs and compiles a list of cyclone tracks. In the PyStormTracker's initial implementation, we define cyclones as local minima of the mean sea level pressure field. The cyclones are detected using SciPy's multi-dimensional image processing library. Cyclone center detection is accelerated using the MPI4Py message passing library, parallelizing over time steps in the input dataset. The tracks are generated by linking cyclone centers together, in time, using the nearest neighborhood method. We have evaluated the performance of the package on NCAR's Yellowstone cluster. The source code is available at GitHub, and is designed to be extensible such that new detection and linking algorithms can be independently implemented.

INTRODUCTION

- Extra-tropical cyclones are responsible for much of the day-to-day weather variability and extreme weather events at mid-latitudes
- Most current cyclone trackers are written in FORTRAN, source codes are not readily available; hard to modify, optimize, or compare existing cyclone trackers
- A review of fifteen cyclone trackers (Neu et al., 2013): Tracking extra-tropical cyclones is challenging; cyclones vary greatly in size and shape, move with different velocities, split and merge over the lifetime

OBJECTIVES

- To implement a user and developer friendly cyclone tracker in Python using an object-oriented approach
  - Flexible and extensible by inheriting or replacing the classes and methods
  - Use existing scientific computing libraries such as NumPy and SciPy
  - The source codes will be made publicly available
- To be able to process data from different datasets
  - GRIB: most commonly used in reanalysis and weather models: NCEP/NCAR Reanalysis and GEFS
  - netCDF: a more recent standard commonly used in climate models, for example, in the CMIP5 ensemble of more than 30 climate models
  - PyNIO is chosen for file I/O
- To utilize parallel computing technique to accelerate processing of large amount of data
  - MPI4Py as an interface to the MPI message passing library

IMPLEMENTATION

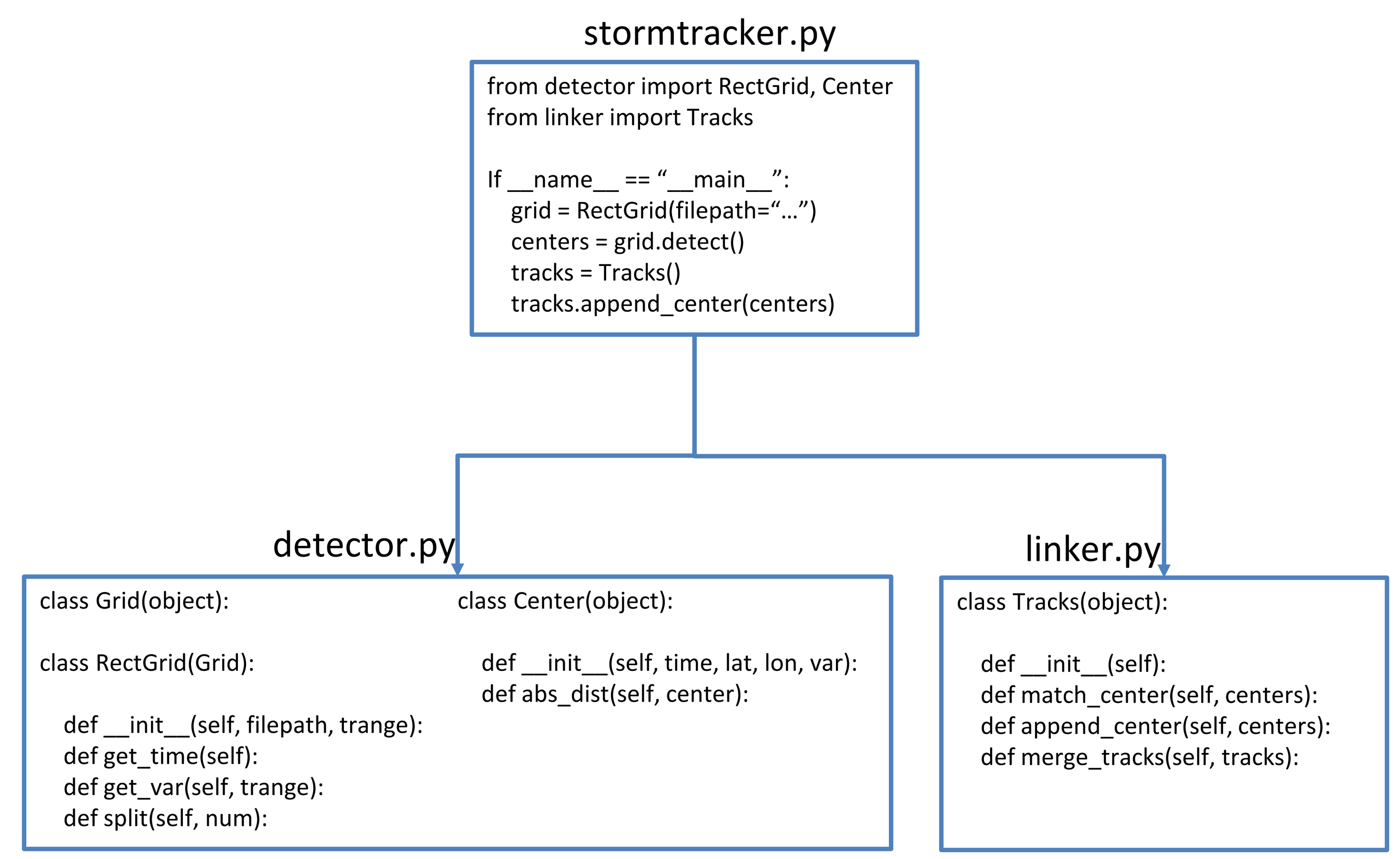
stormtracker.py
- The main controller for generating Grid, Center and Track objects
- Manages parallel MPI processes

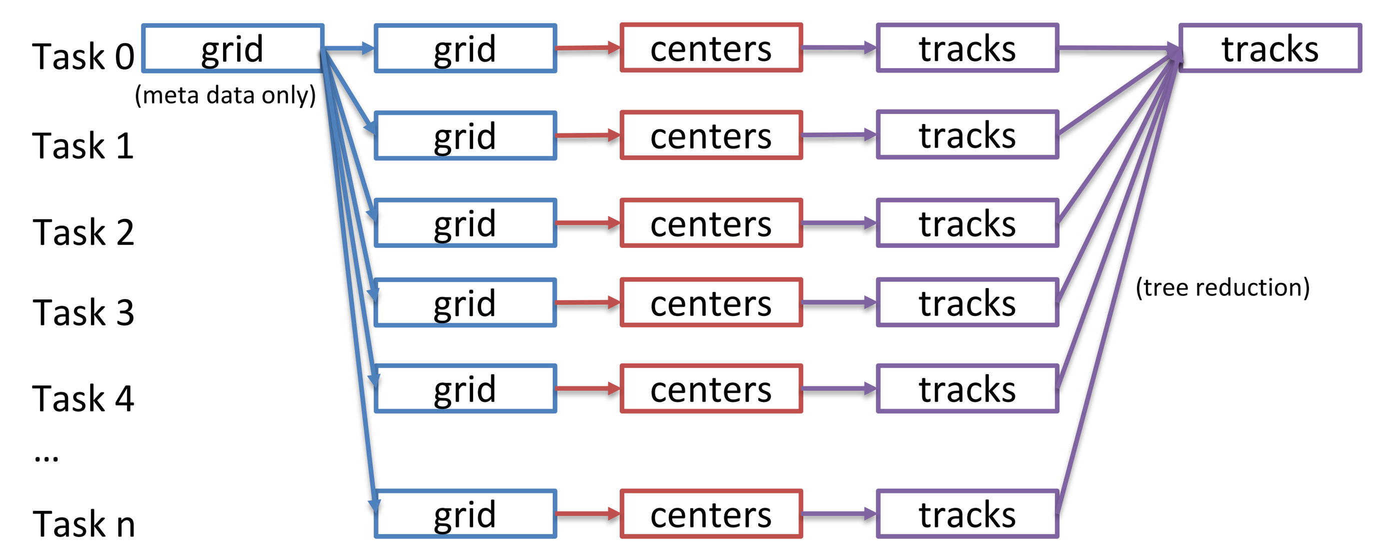
RectGrid class
- Rectilinear grid derived from Grid class
- Does not read data until actually required
- Splits self into smaller objects by time section

Center class
- Detects centers from Grid object using scipy.ndimage.filters
- Calculates great circle distances between centers

Tracks class
- A collection of tracks, which are lists of Center objects
- Matches centers to existing Tracks using nearest neighborhood method
- If matched, appends to existing track; else appends center as a new track
- Merges two Tracks objects

Task Parallelism
- The grid is initialized using meta data only, such as data file path, time range and tracking parameters
- The grid is sliced into different time sections and distributed to different MPI tasks
- Each MPI task reads the data file individually, creates a list of Center using local minima detection and connects the centers together to form a Tracks object for the assigned time section
- The tracks are combined using tree reduction



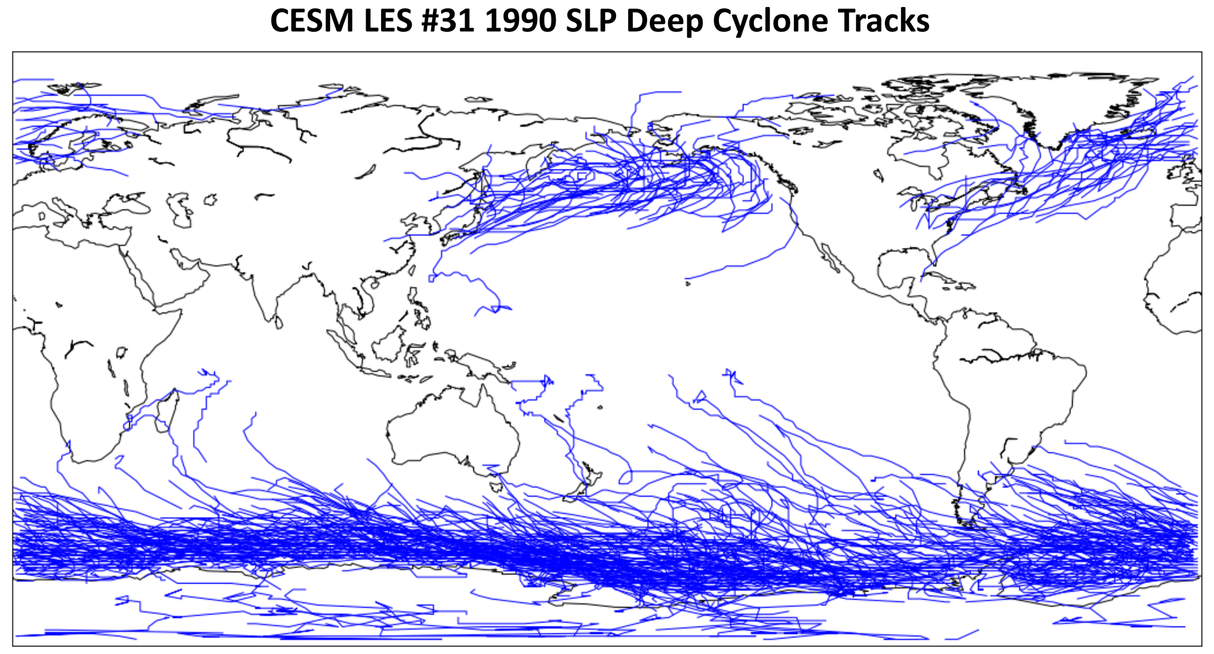


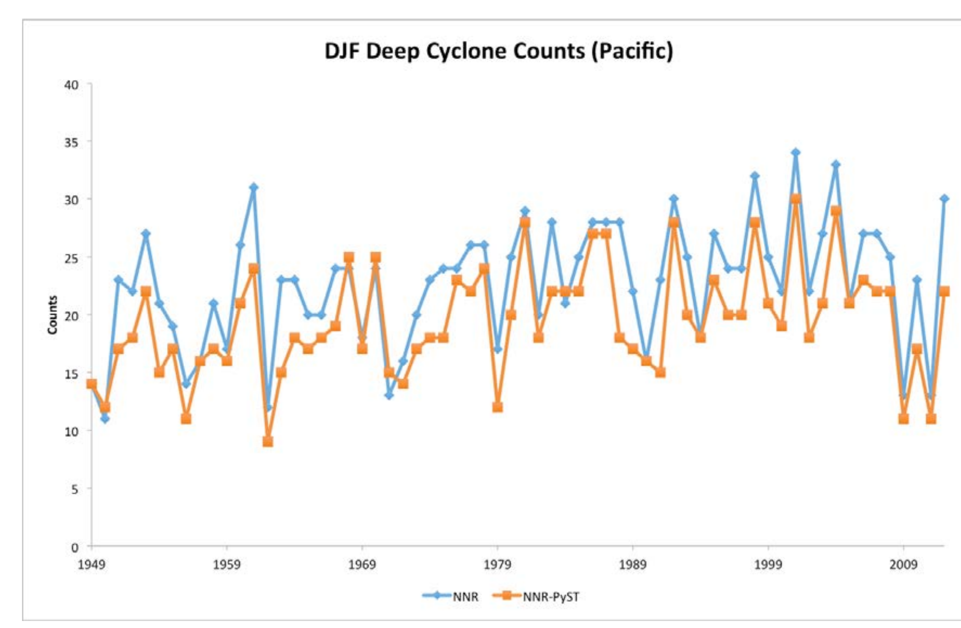
RESULTS

(left) Tracks from CESM Large Ensemble #31 SLP in 1990 showing 769 deep cyclone tracks (cyclone centers deeper than 975hPa, moved 1000km and longer than 2 days)

(right) Counts of northern hemisphere winter (Dec-Feb) Pacific deep cyclone tracks in NCEP/NCAR Reanalysis compared with Hodges (1999) tracker

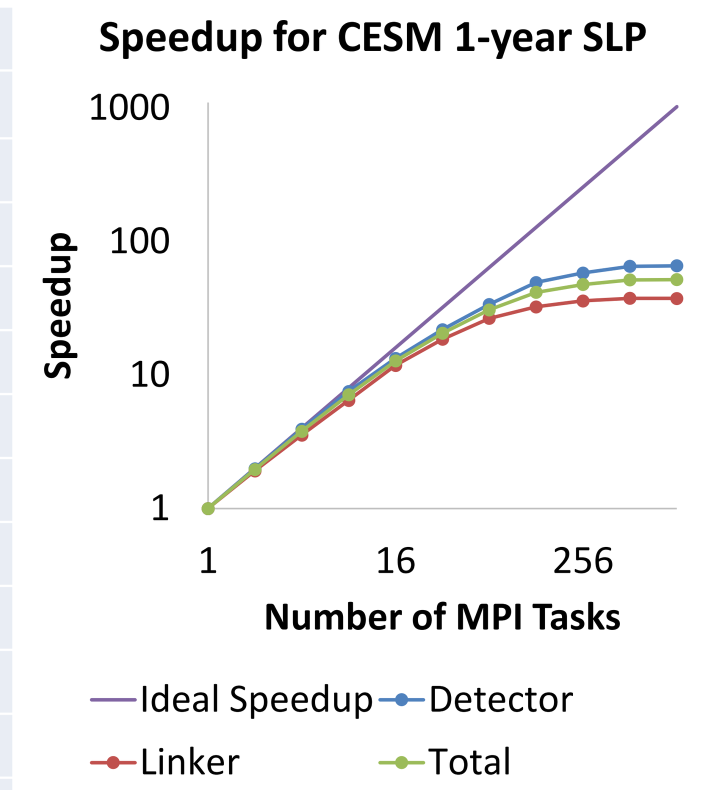
CESM LES #31 1990 SLP Deep Cyclone Tracks



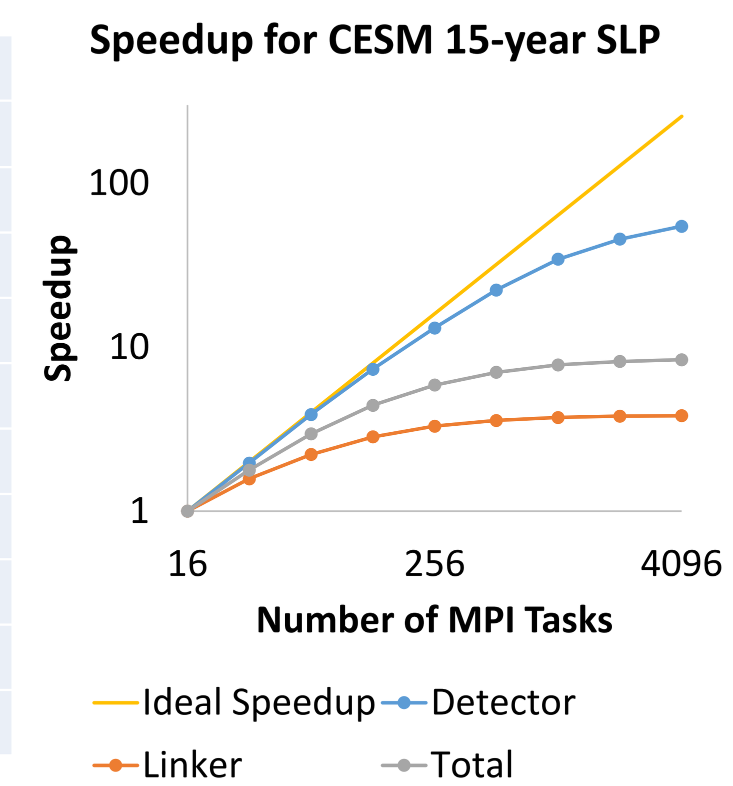


PERFORMANCE

Number of MPI Tasks	Time (s)		
	Detector	Linker	Total
1	1028.22	569.55	1597.78
2	517.25	297.34	814.58
4	261.58	160.38	421.96
8	136.39	88.17	224.56
16	77.21	48.20	125.40
32	46.99	30.73	77.72
64	30.44	21.43	51.86
128	20.80	17.59	38.39
256	17.67	15.86	33.53
512	15.77	15.17	30.93
1024	15.63	15.20	30.83



Number of MPI Tasks	Time (s)		
	Detector	Linker	Total
16	1057.31	749.45	1806.76
32	537.75	475.40	1013.15
64	272.01	337.67	609.69
128	143.80	264.06	407.86
256	80.64	226.78	307.42
512	47.35	209.60	256.95
1024	30.71	200.80	231.50
2048	23.16	197.29	220.45
4096	19.33	196.02	215.36



REFERENCES

- K. I. Hodges, 1999: Adaptive Constraints for Feature Tracking. *Mon. Wea. Rev.*, **127**, 1362–1373.
- Urs Neu, et al., 2013: IMILAST: A Community Effort to Intercompare Extratropical Cyclone Detection and Tracking Algorithms. *Bull. Amer. Meteor. Soc.*, **94**, 529–547.

ACKNOWLEDGEMENTS

- This project is supported by the Summer Internships in Parallel Computational Science (SIParCS) program in summer 2015 at the National Center for Atmospheric Research: http://www2.cisl.ucar.edu/siparcs
- PyNIO: http://www.pyngl.ucar.edu/Nio.shtml
- MPI4Py: http://pythonhosted.org/mpi4py

The source code repository is available at:
http://github.com/mwyau/PyStormTracker
Contact: albert.yau@stonybrook.edu