Dask for HEP

Nick Smith

PyHEP Module of the Month

5 May 2021

Outline

- Setup
- Overview of dask and its place in the ecosystem
- Abridged notebook-based generic tutorial
- Notebook-based tutorial on HEP-specific considerations

Dask tutorial setup

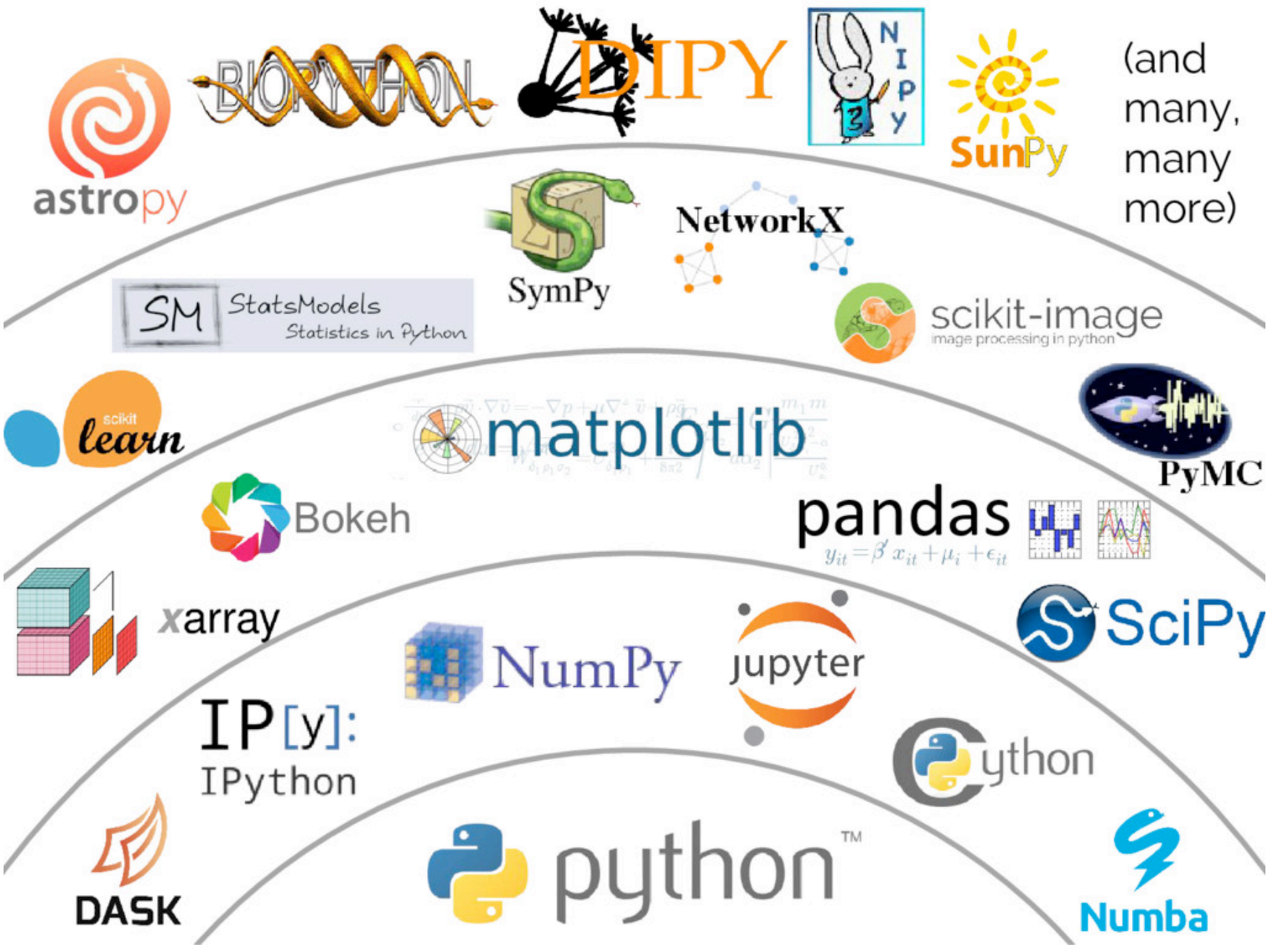
- I will give an abridged version of https://tutorial.dask.org
  - See the SciPy2020 youtube tutorial for an (almost complete) 4h tutorial
- I will also show a HEP example analysis task using dask
- If you want to follow along, please start with the instructions at
  - https://github.com/nsmith-/dask-hep-tutorial#readme

Dask

- Dask is a parallel computing library that scales the existing Python ecosystem:
  - **High level collections**: Dask provides high-level *Array*, *Bag*, and *DataFrame* collections that mimic NumPy, lists of objects, and Pandas (resp.) but can operate in parallel on datasets that don't fit into memory. Dask's high-level collections are alternatives to NumPy and Pandas for large datasets.
  - **Low Level schedulers**: Dask provides dynamic task schedulers that execute task graphs in parallel. These execution engines power the high-level collections mentioned above but can also power custom, user-defined workloads. These schedulers are low-latency (around 1ms) and work hard to run computations in a small memory footprint. Dask's schedulers are an alternative to direct use of threading or multiprocessing libraries in complex cases or other task scheduling systems like Luigi or IPython parallel.

Ecosystem

- Dask is considered a component of the scientific python ecosystem:



Alternatives

- Dask provides functionality similar to that of other projects:
  - Apache Spark http://spark.apache.org/
    - Scala with python bindings, *very large* community
  - Joblib: https://joblib.readthedocs.io/en/latest/
    - Common in scikit-learn community, targeting CPU-bound tasks
  - Celery: https://docs.celeryproject.org/en/stable/getting-started/introduction.html
    - Generic task queue leveraging modern distributed foundations: rabbitMQ, redis, etc.
  - Scoop: https://scoop.readthedocs.io/en/0.7/
    - Another multi-machine DAG library
  - stdlib concurrent.futures: https://www.python.org/dev/peps/pep-3148/
    - Many of the other projects (incl. dask) implement this protocol
  - Ray: https://ray.io/
    - Multi-scheduler design, more focused on distributed ML tasks
  - Parsl: http://parsl-project.org/
    - Popular in academic/HPC communities
- Higher-level task DAG libraries:
  - Apache airflow: https://airflow.apache.org/
  - Luigi: https://github.com/spotify/luigi

More resources

- Dask summit
  - https://summit.dask.org/schedule/
  - https://summit.dask.org/schedule/presentation/24/dask-in-high-energy-physics-community/
    - Organized by Oksana Shadura, Lukas Heinrich
- Dask resources:
  - Docs: https://dask.org/
  - Stackoverflow: http://stackoverflow.com/questions/tagged/dask
  - Github issues: https://github.com/dask/dask/issues/new
- Scaling options for HEP
  - Dask-jobqueue: https://jobqueue.dask.org/en/latest/
    - Easily deploy Dask on job queuing systems like PBS, Slurm, MOAB, SGE, LSF, and HTCondor
    - Caveats apply for HTCondor @ FNAL LPC, CERN LXPLUS
      - See https://github.com/CoffeaTeam/lpcjobqueue for a canned solution (issue #2 for lxplus)
  - Coffea-casa: https://github.com/CoffeaTeam/coffea-casa
    - An “analysis facility in a box” using JupyterHub + kubernetes + HTCondor to provision workers