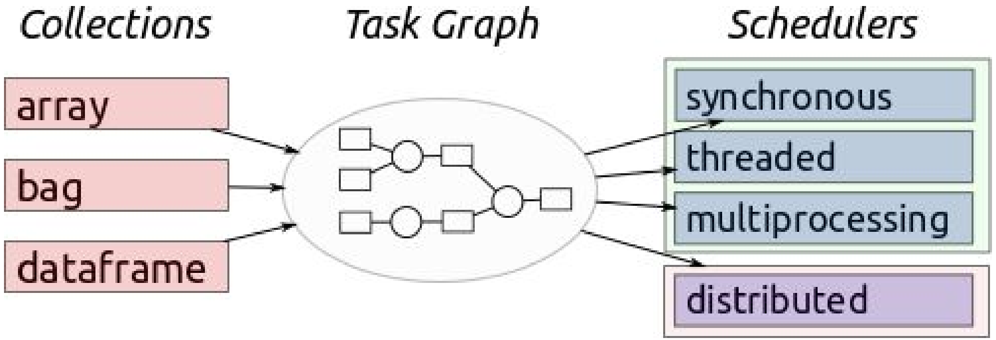
DASK

Parallel Computing Library for Python

Introduction

- Python Library
- Multi-core (single machine) and distributed parallel execution (multi-machines).

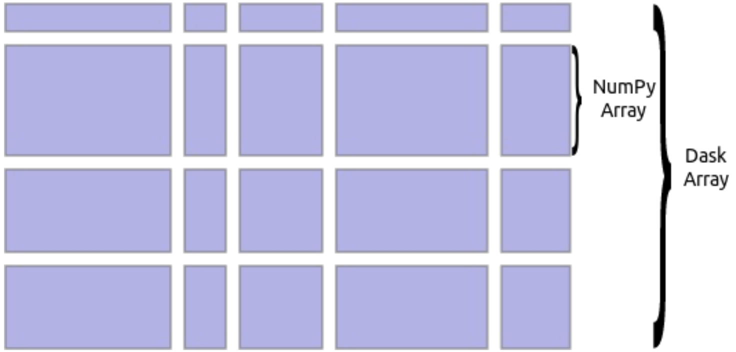


High Level collections: similar to Spark, Databases…
It mimics functionalities of numpy, lists & pandas.

Low Level collections: similar to IPython Parallel, Airflow, Celery.

dask.array

- Similar interface as Numpy.
- Compliments large on-disk array stored like HDF5, NetCDF, BColz.
- Dask.array coordinates many numpy arrays arranged into grid.



- Commonly used to speedup in-memory computations using multiple cores.

http://dask.pydata.org/en/latest/array.html

dask.bag

- dask.Bag often used to parallelize simple computations on unstructured or semi-structured data like text data, log files, JSON records, or user defined Python object.

Disadvantages:

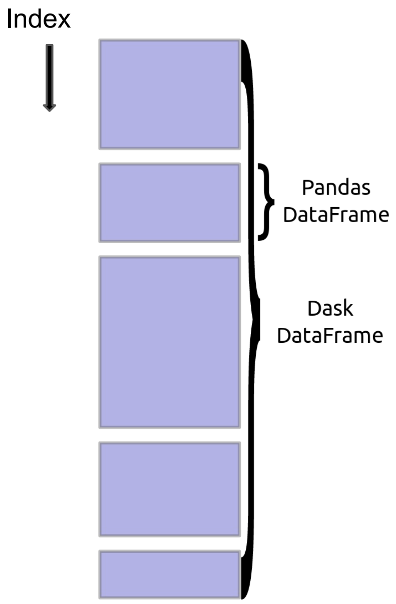
- It utilizes dask.multiprocessing as default Scheduler. Hence it does not perform very well for tasks that include computations with many inter-worker communications.
- Slower than array/dataframe computations.

http://dask.pydata.org/en/latest/bag.html

dask.dataframe

- Dataframes from various data storage formats like CSV, HDF, Apache Parquet, and others can be created.
- For most formats this data can live on various storage systems including local disk, network file systems (NFS), the Hadoop File System (HDFS), and Amazon’s S3.
- Dask dataframes coordinate many Pandas DataFrames/Series arranged along the index.
- Dask.dataframe is partitioned row-wise, grouping rows by index value for efficiency.

- <u>Disadvantage:</u> Setting a new Index from unsorted column is expensive; also for groupby-apply and join on unsorted columns.



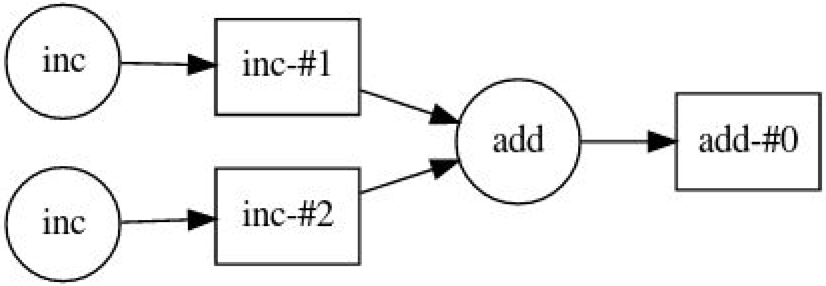
http://dask.pydata.org/en/latest/dataframe-api.html

dask.delayed

- To parallelize custom algorithms.
- Implemented by delaying the computations and turning them into a dask-graph.

Example:

```python
x = dask.delayed(inc)(1)
y = dask.delayed(inc)(2)
z = dask.delayed(add)(x, y)
z.compute()
z.visualize()
```

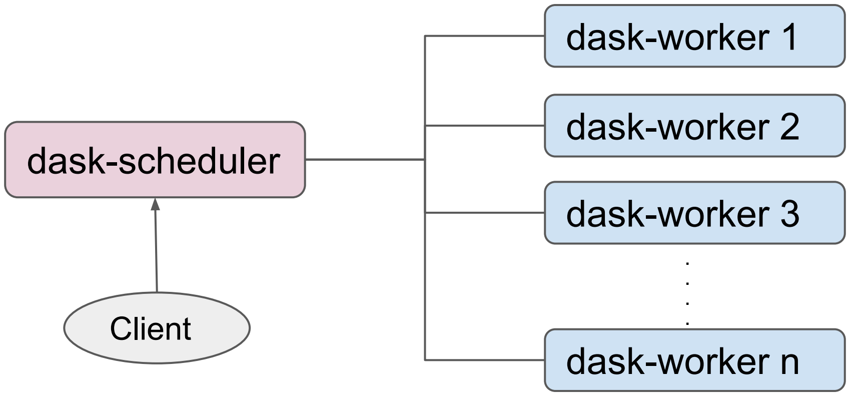


<u>Operations not supported by dask.delayed:</u>

- Mutating operators (a += 1).
- Iteration (for, while loops).
- Use as a predicate (if/else …).

dask.distributed

- Dask can run on a cluster of hundreds of machines and thousands of cores.
- The central dask-scheduler process coordinates the actions of several dask-worker processes spread across multiple machines and the concurrent requests of several clients.



- Users interact by connecting a local Python session to the scheduler and submitting work, by individual calls to the simple interface: **client.submit(function, *args, **kwargs).**

https://distributed.readthedocs.io/en/latest/

dask.distributed Network setup

dask.scheduler and dask.worker network can be initiated/connected in following ways:

- Command line
- SSH
- Shared Network FIle System (NFS)
- Python interface
- LocalCluster
- Amazon EC2

https://distributed.readthedocs.io/en/latest/setup.html

Choosing the Scheduler

dask.threaded.get

dask.multiprocessing.get

dask.get

Computation on Single machine, multiple cores

distributed.Client.get

Computation on multiple machines

dask.distributed.Client

- The Client is the primary entry point for users of dask.distributed.
- When we create a Client object it registers itself as the default Dask scheduler.

https://distributed.readthedocs.io/en/latest/client.html

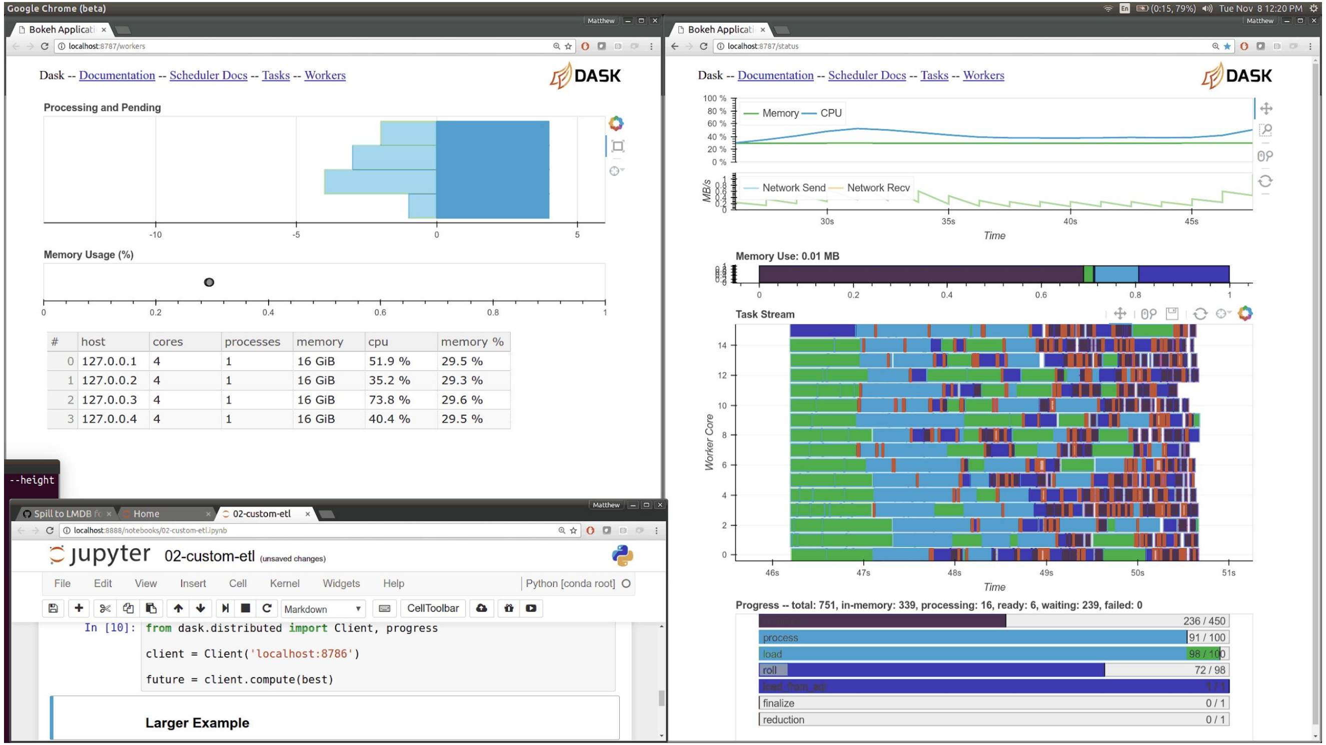
Dask for Machine Learning

- Model Selection and hyperparameter search: **dask-searchcv:** GridSearchCV, RandomSearchCV, Pipeline (Similar to functions in sklearn.model_selection).
- With Joblib: **with** joblib.parallel_backend('dask.distributed', scheduler_host='localhost:8786').
- Convex Optimization: **dask-glm.**
- **dask-xgboost:** dask based parallel implementation of XGboost.
- **dask-tensorflow**: starts Tensorflow clusters from Dask.

https://github.com/dask/dask-searchcv
https://matthewrocklin.com/blog//work/2017/03/28/dask-xgboost
http://matthewrocklin.com/blog/work/2017/03/22/dask-glm-1
http://matthewrocklin.com/blog/work/2017/02/11/dask-tensorflow

Web Interface

Built over Bokeh server



Displays:

- Worker Status, load
- Tasks Distribution
- Progress bar