DASK FOR PARALLEL COMPUTING CHEAT SHEET

See full Dask documentation at: http://dask.pydata.org/
These instructions use conda environment manager. Get yours at http://bit.ly/getconda
TIP: Use help(object) to get help about any Python object

DASK QUICK INSTALL

Install Dask with conda	conda install dask
Install Dask with pip	pip install dask[complete]

DASK COLLECTIONS

DASK ARRAYS

Import dask.array library	import dask.array as da
Create a Dask Array from NumPy-like array	x = da.from_array(d, chunks=(m, n, ...))

Example Dask Array from HDF5 file	import h5py
	
	f = h5py.File('datafile.hdf5', 'r')
	x = f['/group1/dataset1']
	d = da.from_array(x, chunks = (1000, 1000))

Store Dask Array in array-like object	da.store(x , array)

Example Store Dask Array into HDF5 file	x = da.random.normal(10, .3, size=(5,5), chunks=(5,1))
	f = h5py.File('myfile.hdf5')
	dset = f.create_dataset(...)
	da.store(x, dset)

Arithmetic element-wise & scalar operations	*, +, -, **, /, exp, log
Example Arithmetic element-wise & scalar operations	y = da.sin(x)**2 + da.cos(x)**2
Reduction along axes	sum(), prod(), mean(), std()
Example Sum reduction along t	y = x.mean(axis=t)
Matrix multiplication and dot product	dot(), tensordot()
Axis reordering	transpose()
Slicing	x[:5, 20:10:-1]
Fancy indexing	x[[1, 3], :]

DASK BAGS

Import dask.bag library	import dask.bag as db
Create Dask Bag from a sequence	db.from_sequence(seq, npartitions)
Example	b = db.from_sequence([1, 2, 3 ,4 ,5 ,6], npartitions=2)
Create Dask Bag from text files	b = db.read_text('data.*.json')
Map function across all elements in a Dask Bag	map()

Example use read_text and json.loads together	import json
	b = db.read_text('data.*.json.gz').map(json.loads)

Trigger computations	compute()

Example	b = db.from_sequence([2, 3, 5, 7, 11, 13], npartitions=2)
	c = b.map(lambda x: x + 1)
	c.compute()

ANACONDA
Powered by Continuum Analytics®

CONTINUED ON BACK →

DASK COLLECTIONS (CONTINUED)

DASK BAGS (CONTINUED)

Some useful functions supported by Dask Bags	max(), min(), mean(), sum(), std(), filter(), fold(), foldby(), frequencies(), groupby(), join(), pluck(), product(), remove(), take(), topk(), var()
Convert to Dask DataFrame	to_dataframe()
Write Dask Bag to disk	to_textfiles('path')

DASK DATAFRAMES

Import dask.dataframe library	import dask.dataframe as dd
Create Dask DataFrame from CSV files	df = dd.read_csv('filenames-*.csv')
Element-wise operations	*, +, /, -
Row-wise selection	df[df.x > 0]
Selection by label	df.loc['2015-01': '2015-05']
Common aggregations	max(), min(), mean(), std(), sum(), count(), var()
pandas operations supported by Dask DataFrames	groupby(), value_counts(), drop_duplicates(), merge(), set_index()
Trigger computations	compute()

Example

```
df = dd.read_csv('filenames.*.csv')
df.sample(frac=0.1, replace=True)
    .groupby(df.timestamp.day)
    .value.mean().compute()
```

GRAPHS

TIP: Use single-threaded scheduler for debugging, dask.set_options(get=dask.async.get_sync)

Scheduler backed by thread pool	dask.threaded.get()
Scheduler backed by process pool	dask.multiprocessing.get()
Synchronous scheduler	dask.async.get_sync()

Example

```
from dask.threaded import get
from operator import add
dsk = {'a' : 1,
         'b' : 2,
         'c' : (add, 'a', 'b')}
get(dsk, 'c')
```

MORE RESOURCES

Support	https://www.continuum.io/support-plan
Training	http://bit.ly/continuumtraining
Consulting	http://bit.ly/continuumconsulting
Dask gitter chat room	https://gitter.im/dask/dask
Report a bug	https://github.com/dask/dask/issues
Dask mailing list	https://groups.google.com/a/continuum.io/forum/#!forum/blaze-dev

Join the #AnacondaCrew!

Connect with other talented, like-minded data scientists and developers while contributing to the open source movement. Visit https://continuum.io/community