session3-intro-to-python-dask-on-ceres

August 24, 2020

1 Parallel Computing on Ceres with Python and Dask

adapted from https://github.com/willirath/dask_jobqueue_workshop_materials by Kerrie Geil, USDA-ARS August 2020

1.1 The Goal

Interactive data analysis on very large datasets. The tools in this tutorial are most appropriate for analysis of large earth-science-type datasets.

For large dataset analysis, you'll want to run parallel (instead of serial) computations to save time. On a high-performance computer (HPC), you could divide your computing into independent segments (batching) and submit multiple batch scripts to run compute jobs simultaneously or you could parallelize your codes using MPI (Message Passing Interface), a traditional method for parallel computing if your code is in C or Fortran. Actually, there is also an "MPI for Python" package, but the methods in this tutorial are much *much* simpler. Both the batching and MPI methods of parallelization do not allow for interactive analysis, such as analysis using a Jupyter notebook, which is often desired by the earth science research community. Note that interactive analysis here does *not* mean constant visual presentation of the data with a graphical user interface (GUI) such as in ArcGIS.

For earth-science-type data and analysis with Python, one of the simplest ways to run parallel computations in an interactive environment is with the Dask package.

1.2 Core Lessons

Using Dask to: - set up SLURM clusters - scale clusters - use adaptive clusters - view Dask diagnostics

This tutorial will demonstrate how to use Dask to manage compute jobs on a SLURM cluster (including setting up your SLURM compute cluster, scaling the cluster, and how to use an adaptive cluster to save compute resources for others). The tutorial will also explain how to access the Dask diagnostics dashboard to view the cluster working in real time.

---

2 Begin Tutorial: Parallel Computing on Ceres with Python and Dask

In this tutorial we will compute in parallel using Python's Dask package to communicate with the Ceres HPC SLURM job scheduler.

SLURM (Simple Linux Utility for Resource Management) is a workload manager for HPC systems. From the SLURM documentation, SLURM is "an open source… cluster management and job scheduling system for large and small Linux clusters. As a cluster workload manager, SLURM has three key functions. First, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work. Second, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes. Finally, it arbitrates contention for resources by managing a queue of pending work."

2.1 First Set up Your File Space

Create a folder in your home directory for the Dask worker error files

[1]:

```
import os

homedir = os.environ['HOME']
daskpath=os.path.join(homedir, "dask-worker-space-can-be-deleted")

try:
    os.mkdir(daskpath)
except OSError as error:
    print(error)
```

```
[Errno 17] File exists: '/home/kerrie.geil/dask-worker-space-can-be-deleted'
```

2.2 Set up a SLURM Cluster with Dask

The first step is to create a SLURM cluster using the dask.distributed and dask_jobqueue packages. The SLURMCluster function can be interpreted as the settings/parameters for 1 SLURM compute job. Later, we can increase our compute power by "scaling our cluster", which means Dask will ask the SLURM scheduler to execute more than one job at a time for any given computation.

Here's a key to the dask_jobqueue.SLURMCluster input parameters in the code block below:

cores = Number of logical cores per job. This will be divided among the processes/workers. Can't be more than the lowest number of logical cores per node in the queue you choose, see https://scinet.usda.gov/guide/ceres/#partitions-or-queues.

processes = Number of processes per job (also known as Dask "workers" or "worker processes"). The number of cores per worker will be cores/processes. Can use 1 but more than 1 may help keep your computations running if cores/workers fail. For numeric computations (Numpy, Pandas, xarray, etc.), less workers may run significantly faster due to reduction in communication time. If your computations are mostly pure Python, it may be better to run with many workers each associated with only 1 core. Here is more info than you'll probably ever want to know about Dask workers.

memory = Memory per job. This will be divided among the processes/workers. See https://scinet.usda.gov/guide/ceres/#partitions-or-queues for the maximum memory per core you can request on each queue.

queue = Name of the Ceres queue, a.k.a. partition (e.g. short, medium, long, long60, mem, longmem, mem768, debug, brief-low, scavenger, etc.).

walltime = Time allowed before the job is timed out.

local_directory = local spill location if the core memory is exceeded, use /local/scratch a.k.a $TMPDIR

log_directory = Location to write the stdout and stderr files for each worker process. Simplest choice may be the directory you are running your code from.

python = The python executable. Add this parameter if you are running in a container to tell SLURM what container and conda env to use. Otherwise, it's not needed.

You can view additional parameters, methods, and attributes in the Dask documentation for dask_jobqueue.SLURMCluster.

[2]:

```
from dask.distributed import Client
from dask_jobqueue import SLURMCluster

container='/lustre/project/geospatial_tutorials/wg_2020_ws/
 ↪data_science_im_rs_vSCINetGeoWS_2020.sif'
env='py_geo'

cluster = SLURMCluster(
    cores=40,
    processes=1,
    memory="120GB", #40 cores x 3GB/core
    queue="short",
    local_directory='$TMPDIR',
    walltime="00:10:00",
    log_directory=daskpath,
    python="singularity -vv exec {} /opt/conda/envs/{}/bin/python".
 ↪format(container,env)) #tell the cluster what container and conda env we're␣
 ↪using
```

So far we have only set up a cluster, we have not started any compute jobs/workers running yet. We can verify this by issuing the following command in a Ceres terminal. Launch a terminal from the JupyterLab launcher and type:

```
squeue -u firstname.lastname
```

To see the job script that will be used to start a job running on the Ceres HPC use the method .job_script() as shown in the code block below.

Here's a key to the output of the cluster.job_script() command below:

-J = Name of the job. This will appear in the “Name” column of the squeue output. “dask-worker” is the default value.

-e and -o = Name/Location of the stdout and stderr files for each job. This comes from the SLURMCLuster “log_directory” parameter.

-p = Name of the Ceres queue/partition. This comes from the SLURMCLuster “queue” parameter.

-n = Number of nodes.

–cpus-per-task = Number of cores per job (same as -N). This comes from the SLURMCluster “cores” parameter.

–mem = Memory per job. This comes from the SLURMCluster “memory” parameter.

-t = Time allowed before the job is timed out. This comes from the SLURMCluster parameter “walltime”.

[3]:
```
print(cluster.job_script())
```

```
#!/usr/bin/env bash

#SBATCH -J dask-worker
#SBATCH -e /home/kerrie.geil/dask-worker-space-can-be-deleted/dask-worker-%J.err
#SBATCH -o /home/kerrie.geil/dask-worker-space-can-be-deleted/dask-worker-%J.out
#SBATCH -p short
#SBATCH -n 1
#SBATCH --cpus-per-task=40
#SBATCH --mem=112G
#SBATCH -t 00:10:00

singularity -vv exec /lustre/project/geospatial_tutorials/wg_2020_ws/data_scienc
e_im_rs_vSCINetGeoWS_2020.sif /opt/conda/envs/py_geo/bin/python -m
distributed.cli.dask_worker tcp://10.1.4.113:40929 --nthreads 40 --memory-limit
120.00GB --name name --nanny --death-timeout 60 --local-directory $TMPDIR
```

Next, we must initialize a Dask Client, which opens a line of communication between Dask worker processes and the SLURM job scheduler by pointing to the address of the scheduler (tcp://10.1.8.84:41601).

[4]:
```
client = Client(cluster)
client
```

[4]:
```
<Client: 'tcp://10.1.4.113:40929' processes=0 threads=0, memory=0 B>
```

Note: So far we have only set up a cluster and initialized a client. We still have not started any compute jobs running yet, as shown in the Cluster information above. We can also verify that no workers are running yet by issuing the squeue command in a Ceres terminal again as we did previously or we could access the Dask Diagnostics Dashboard for even more information.

2.3 Viewing the Dask Diagnostics Dashboard

We will now take a look at the Dask Dashboard to verify that there a no workers running in our cluster yet. Once we start computing, we will be able to use the Dashboard to see a visual representation of all the workers running.

At the very left edge of JupyterLab click the icon that looks like two orange leaves. If the Dask Dashboard extension is working you should see a bunch of orange colored boxes. Each one of these boxes allows you to visualize a different aspect of the compute job.

Click on the "workers" box to open a separate tab for visualizing the dask workers as they compute. Click over to that tab and right now you should see that there are no workers running yet. When you run a compute job you will see the workers populate the table on your dask workers tab.

2.4 Scale the Cluster to Start Computing

Now let's start multiple SLURM jobs computing.

```
[5]: from time import time, sleep   #time for timing computation length, sleep for␣
      ↪pausing while SLURM starts the requested jobs

     cluster.scale(jobs=3)  # scale to more jobs
     sleep(15)              # pause while SLURM starts up the jobs
     client
```

```
[5]: <Client: 'tcp://10.1.4.113:40929' processes=3 threads=120, memory=360.00 GB>
```

The .scale() method actually starts the jobs running as shown in the Cluster information above.

A quick check of squeue will now show your multiple jobs running as well. Or click over to your Dask Workers tab and you'll see you have workers ready to compute.

When we set up our original cluster (equivalent of 1 SLURM job) we requested 40 cores spread over 2 workers. When we scaled our cluster to 3 jobs we now have 40x3=120 cores spread over 2x3=6 workers, as shown above. Note: you can also scale your cluster by cores, workers or memory as opposed to jobs.

2.5 Monte-Carlo Estimate of π

Now we will use the Monte-Carlo method of estimating π to demonstrate how Dask can execute parallel computations with the SLURM Cluster we've built and scaled.

We estimate the number π by exploiting that the area of a quarter circle of unit radius is $\pi/4$ and that hence the probability of any randomly chosen point in a unit square to lie in a unit circle centerd at a corner of the unit square is $\pi/4$ as well.

So for N randomly chosen pairs (x, y) with $x \in [0, 1)$ and $y \in [0, 1)$, we count the number N_{circ} of pairs that also satisfy $(x^2 + y^2) < 1$ and estimate $\pi \approx 4 \cdot N_{circ}/N$.

Let's define a function to compute π and another function to print out some info during the compute.

[6]:

```
import dask.array as da
import numpy as np

def calc_pi_mc(size_in_bytes, chunksize_in_bytes=200e6):
    """Calculate PI using a Monte Carlo estimate."""

    size = int(size_in_bytes / 8)       # size= no. of random numbers to␣
 ↪generate (x & y vals), divide 8 bcz numpy float64's generated by random.
 ↪uniform are 8 bytes
    chunksize = int(chunksize_in_bytes / 8)

    xy = da.random.uniform(0, 1,                             # this generates a␣
 ↪set of x and y value pairs on the interval [0,1) of type float64
                           size=(size / 2, 2),               # divide 2 because we␣
 ↪are generating an equal number of x and y values (to get our points)
                           chunks=(chunksize / 2, 2))

    in_circle = ((xy ** 2).sum(axis=-1) < 1)      # a boolean array, True for␣
 ↪points that fall inside the unit circle (x**2 + y**2 < 1)
    pi = 4 * in_circle.mean()                     # mean= sum the number of True␣
 ↪elements, divide by the total number of elements in the array

    return pi

def print_pi_stats(size, pi, time_delta, num_workers):
    """Print pi, calculate offset from true value, and print some stats."""
    print(f"{size / 1e9} GB\n"
          f"\tMC pi: {pi : 13.11f}"
          f"\tErr: {abs(pi - np.pi) : 10.3e}\n"
          f"\tWorkers: {num_workers}"
          f"\t\tTime: {time_delta : 7.3f}s")
```

2.6 The Actual Calculations

We loop over different volumes (1GB, 10GB, and 100GB) of double-precision random numbers (float64, 8 bytes each) and estimate π as described above. Note, we call the function with the .compute() method to start the computations. To see the dask workers computing, execute the code block below and then quickly click over to your dask workers tab.

[7]:

```
for size in (1e9 * n for n in (1, 10, 100)):

    start = time()
    pi = calc_pi_mc(size).compute()
    elaps = time() - start

    print_pi_stats(size, pi, time_delta=elaps,
                   num_workers=len(cluster.scheduler.workers))
```

```
1.0 GB
        MC pi:  3.14163468800   Err:  4.203e-05
        Workers: 3              Time:   1.156s
10.0 GB
        MC pi:  3.14153445760   Err:  5.820e-05
        Workers: 3              Time:   1.139s
100.0 GB
        MC pi:  3.14159745088   Err:  4.797e-06
        Workers: 3              Time:   8.593s
```

2.7 Scale the Cluster to Twice its Size and Re-run the Same Calculations

We increase the number of workers times 2 and the re-run the experiments. You could also double the size of the cluster by doubling the number of jobs, cores, or memory.

[8]:

```
new_num_workers = 2 * len(cluster.scheduler.workers)

print(f"Scaling from {len(cluster.scheduler.workers)} to {new_num_workers}␣
 ↪workers.")

cluster.scale(new_num_workers)

# the following commands all get you the same amount of compute resources as␣
 ↪above
#cluster.scale(12)                 # same as code above. default parameter is␣
 ↪workers. (original num workers was 6)
#cluster.scale(jobs=6)             # can scale by number of jobs
#cluster.scale(cores=240)          # can also scale by cores
#cluster.scale(memory=600)         # can also scale by memory

sleep(15)
client
```

```
Scaling from 3 to 6 workers.
```

[8]:

```
<Client: 'tcp://10.1.4.113:40929' processes=6 threads=240, memory=720.00 GB>
```

[9]:

```
for size in (1e9 * n for n in (1, 10, 100)):


    start = time()
    pi = calc_pi_mc(size).compute()
    elaps = time() - start

    print_pi_stats(size, pi,
                   time_delta=elaps,
                   num_workers=len(cluster.scheduler.workers))
```

```
1.0 GB
        MC pi:  3.14218400000   Err:  5.913e-04
        Workers: 6              Time:   1.004s
10.0 GB
        MC pi:  3.14166279680   Err:  7.014e-05
        Workers: 6              Time:   1.070s
100.0 GB
        MC pi:  3.14158664128   Err:  6.012e-06
        Workers: 6              Time:   4.554s
```

2.8 Automatically Scale the Cluster Up and Down (Adaptive Cluster)

Using the .adapt() method will dynamically scale up the cluster when necessary but scale it down and save compute resources when not actively computing. Dask will ask the SLURM job scheduler to run more jobs, scaling up the cluster, when workload is high and shut the extra jobs down when workload is smaller.

Note that cluster scaling is bound by SCINet HPC user limitations. These limitations on the Ceres HPC are 400 cores, 1512GB memory, and 100 jobs max running simultaneously per user. So for example, if you set your cluster up with 40 cores per job and scale to 20 jobs (40x20=800cores) you will only get 400 cores (10 jobs) running at any time and the remaining requested jobs will not run. Your computations will still run successfully, but they will run on 10 jobs/400 cores instead of the requested 20 jobs/800 cores.

Watch *how the cluster will scale down to the minimum a few seconds after being made adaptive.*

[10]:

```
ca = cluster.adapt(minimum_jobs=1, maximum_jobs=9);

sleep(5)  # Allow for scale-down
client
```

[10]:

```
<Client: 'tcp://10.1.4.113:40929' processes=1 threads=40, memory=120.00 GB>
```

Now, we'll repeat the calculations with our adaptive cluster and a larger workload. Watch the dash board!

[11]:

```
for size in (n * 1e9 for n in (1, 10, 100, 1000)):

    start = time()
    pi = calc_pi_mc(size, min(size / 1000, 500e6)).compute()
    elaps = time() - start

    print_pi_stats(size, pi, time_delta=elaps,
                   num_workers=len(cluster.scheduler.workers))

sleep(5)  # allow for scale-down time
client
```

```
1.0 GB
        MC pi:  3.14130771200   Err:  2.849e-04
        Workers: 2              Time:   6.426s
10.0 GB
        MC pi:  3.14160750080   Err:  1.485e-05
        Workers: 6              Time:   3.515s
100.0 GB
        MC pi:  3.14158415680   Err:  8.497e-06
        Workers: 9              Time:   3.359s
1000.0 GB
        MC pi:  3.14158967315   Err:  2.980e-06
        Workers: 9              Time:  25.888s
```

```
[11]: <Client: 'tcp://10.1.4.113:40929' processes=1 threads=40, memory=120.00 GB>
```

What are the use cases for the adaptive cluster feature? Personally, I will be using the adaptive cluster when I have a code that contains a mix of lighter and heavier computations so I can use the minimum number of cores necessary for the lighter parts of the code and then have my cluster automagically scale up to handle heavier parts of the code without me having to think about it.

2.9 Complete listing of software used here

```
[12]: %conda list --explicit
```

```
# This file may be used to create an environment using:
# $ conda create --name <env> --file <this file>
# platform: linux-64
@EXPLICIT
https://conda.anaconda.org/conda-
forge/linux-64/_libgcc_mutex-0.1-conda_forge.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/ca-
certificates-2020.6.20-hecda079_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/hicolor-icon-
theme-0.17-0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/ld_impl_linux-64-2.34-h53a641e_7.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libgfortran-
ng-7.5.0-hdf63c60_6.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libstdcxx-
ng-9.2.0-hdf63c60_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pandoc-2.10-h14c3975_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/poppler-data-0.4.9-1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libgomp-9.2.0-h24d8f2e_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/_openmp_mutex-4.5-0_gnu.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libgcc-
ng-9.2.0-h24d8f2e_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/abseil-
cpp-20200225.2-he1b5a44_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/blosc-1.19.0-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-
```

```
forge/linux-64/brotli-1.0.7-he1b5a44_1004.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/bzip2-1.0.8-h516909a_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/c-ares-1.16.1-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/charls-2.1.0-he1b5a44_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/epoxy-1.5.4-h516909a_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/expat-2.2.9-he1b5a44_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/firefox-78.0esr-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/freexl-1.0.5-h516909a_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/fribidi-1.0.10-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/geckodriver-0.26.0-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/geos-3.8.1-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gflags-2.2.2-he1b5a44_1004.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/giflib-5.2.1-h516909a_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gmp-6.2.0-he1b5a44_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/graphite2-1.3.13-he1b5a44_1001.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/icu-64.2-he1b5a44_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/jpeg-9d-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/json-c-0.13.1-hbfbb72e_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/jxrlib-1.1-h516909a_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/lerc-2.2-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libaec-1.0.4-he1b5a44_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libffi-3.2.1-he1b5a44_1007.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libiconv-1.15-h516909a_1006.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libopenblas-0.3.10-pthreads_hb3c22a3_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libsodium-1.0.17-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libspatialindex-1.9.3-he1b5a44_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libtool-2.4.6-h14c3975_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libuuid-2.32.1-h14c3975_1000.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libwebp-base-1.1.0-h516909a_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libzopfli-1.0.3-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/lz4-c-1.9.2-he1b5a44_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/lzo-2.10-h14c3975_1000.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/linux-64/ncurses-6.2-he1b5a44_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/openssl-1.1.1g-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pcre-8.44-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pixman-0.38.0-h516909a_1003.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pthread-stubs-0.4-h14c3975_1001.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/re2-2020.07.06-he1b5a44_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/snappy-1.1.8-he1b5a44_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/tbb-2020.1-hc9558a2_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/tzcode-2020a-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-kbproto-1.0.7-h14c3975_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libice-1.0.10-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libxau-1.0.9-h14c3975_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libxdmcp-1.1.3-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-renderproto-0.11.1-h14c3975_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-xextproto-7.3.0-h14c3975_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-xproto-7.0.31-h14c3975_1007.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xz-5.2.5-h516909a_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/yaml-0.2.5-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/zfp-0.5.5-he1b5a44_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/zlib-1.2.11-h516909a_1006.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/boost-cpp-1.72.0-h8e57a91_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/brunsli-0.1-he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gettext-0.19.8.1-hc5be6a0_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/glog-0.4.0-h49b9bf7_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/hdf4-4.2.13-hf30be14_1003.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/hdf5-1.10.6-nompi_h3c11f04_100.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libblas-3.8.0-17_openblas.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libedit-3.1.20191231-h46ee950_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libevent-2.1.10-hcdb4288_1.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/linux-64/libllvm9-9.0.1-he513fc3_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libpng-1.6.37-hed695b0_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libprotobuf-3.12.3-h8b12597_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libssh2-1.9.0-hab1572f_4.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libxcb-1.13-h14c3975_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libxml2-2.9.10-hee79883_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/mpfr-4.0.2-he80fd80_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/readline-8.0-he28a2e2_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/tk-8.6.10-hed695b0_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xerces-c-3.2.2-h8412b87_1004.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libsm-1.2.3-h84519dc_1000.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/zeromq-4.3.2-he1b5a44_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/zstd-1.4.5-h6597ccf_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/freetype-2.10.2-he06d7ca_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/grpc-cpp-1.30.2-heedbac9_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/kealib-1.4.13-h33137a7_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/krb5-1.17.1-hfafb76e_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libcblas-3.8.0-17_openblas.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libkml-1.3.0-hb574062_1011.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/liblapack-3.8.0-17_openblas.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libtiff-4.1.0-hc7e4089_6.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libxslt-1.1.33-h31b3aaa_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/mpc-1.1.0-h04dde30_1007.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/sqlite-3.32.3-hcee41ef_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/thrift-cpp-0.13.0-h62aa4f2_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libx11-1.6.9-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/fontconfig-2.13.1-h86ecdb6_1001.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/lcms2-2.11-hbd6801e_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libcups-2.2.12-hf10b501_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libcurl-7.71.1-hcdd3856_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libpq-12.3-h5513abc_0.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/linux-64/openjpeg-2.3.1-h981e76c_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/python-3.7.6-cpython_h8356626_6.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libxext-1.3.4-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libxrender-0.9.10-h516909a_1002.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libxt-1.1.5-h516909a_1003.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/affine-2.3.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/appdirs-1.4.3-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/asciitree-0.3.3-py_2.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/async-timeout-3.0.1-py_1000.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/attrs-19.3.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/backcall-0.2.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/blinker-1.4-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/cachetools-4.1.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/cfitsio-3.470-h3eac812_5.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/click-7.1.2-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/cloudpickle-1.5.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/curl-7.71.1-he644dc0_3.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/decorator-4.4.2-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/defusedxml-0.6.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/docopt-0.6.2-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/docutils-0.15.2-py37_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/et_xmlfile-1.0.1-py_1001.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/fastprogress-0.2.3-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/fsspec-0.7.4-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/glib-2.65.0-h6f030ca_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gmpy2-2.1.0b1-py37h04dde30_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/heapdict-1.0.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/httplib2-0.18.1-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/idna-2.10-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/ipython_genutils-0.2.0-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/jdcal-1.4.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/jmespath-0.10.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/locket-0.2.0-py_2.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/monotonic-1.5-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/more-itertools-8.4.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/mpmath-1.1.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/olefile-0.46-py_0.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/noarch/pandocfilters-1.4.2-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/param-1.9.3-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/parso-0.7.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/phantomjs-2.1.1-1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/postgresql-12.3-h8573dbc_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/proj-7.0.0-h966b41f_5.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/prometheus_client-0.8.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/ptyprocess-0.6.0-py_1001.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/py-1.9.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyasn1-0.4.8-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pycparser-2.20-pyh9f0ad1d_2.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyparsing-2.4.7-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyshp-2.1.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/python_abi-3.7-1_cp37m.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pytz-2020.1-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/send2trash-1.5.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/simpervisor-0.3-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/six-1.15.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/sortedcontainers-2.2.2-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/tblib-1.6.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/testpath-0.4.4-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/threadpoolctl-2.1.0-pyh5ca1d4c_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/toolz-0.10.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/tqdm-4.48.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/typing_extensions-3.7.4.2-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/wcwidth-0.2.5-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/webencodings-0.5.1-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/webob-1.8.6-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/xorg-libxpm-3.5.13-h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/zipp-3.1.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/aws-sdk-cpp-1.7.164-hc831370_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/cairo-1.16.0-hcf35c78_1003.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/certifi-2020.6.20-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/cffi-1.14.0-py37hd463f26_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/chardet-3.0.4-py37hc8dfbb8_1006.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/noarch/click-plugins-1.1.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/cligj-0.5.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/cycler-0.10.0-py_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/cytoolz-0.10.1-py37h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/dbus-1.13.6-he372182_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/entrypoints-0.3-py37hc8dfbb8_1001.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/fastcache-1.1.0-py37h8f50634_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/fasteners-0.14.1-py_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/future-0.18.2-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gdb-9.2-py37h7d4168f_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/geotiff-1.6.0-h05acad5_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/google-resumable-media-0.5.1-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/importlib-metadata-1.7.0-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/jedi-0.17.2-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/kiwisolver-1.2.0-py37h99015e2_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libdap4-3.20.6-h1d1bd15_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libnetcdf-4.7.4-nompi_h84807e1_104.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libspatialite-4.3.0a-h2482549_1038.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/llvmlite-0.33.0-py37h5202443_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/lxml-4.5.2-py37he3881c9_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/markupsafe-1.1.1-py37h8f50634_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/mistune-0.8.4-py37h8f50634_1001.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/mock-4.0.2-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/msgpack-python-1.0.0-py37h99015e2_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/multidict-4.7.5-py37h8f50634_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/multipledispatch-0.6.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/numpy-1.19.0-py37h8960a57_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/openpyxl-3.0.4-py_0.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/noarch/packaging-20.4-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/partd-1.1.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pexpect-4.8.0-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pickleshare-0.7.5-py37hc8dfbb8_1001.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pillow-7.2.0-py37h718be6c_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/psutil-5.7.2-py37h8f50634_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/psycopg2-2.8.5-py37hb09aad4_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyasn1-modules-0.2.7-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyct-core-0.4.6-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pyproj-2.6.1.post1-py37h34dd122_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pyrsistent-0.16.0-py37h8f50634_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pysocks-1.7.1-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/python-dateutil-2.7.5-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyviz_comms-0.7.6-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pyyaml-5.3.1-py37h8f50634_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pyzmq-19.0.1-py37hac76be4_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/rsa-4.6-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/rtree-0.9.4-py37h8526d28_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/simplejson-3.17.2-py37h8f50634_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/soupsieve-2.0.1-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/sqlalchemy-1.3.18-py37h8f50634_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/thrift-0.11.0-py37he1b5a44_1001.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/tiledb-1.7.7-h8efa9f0_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/tornado-6.0.4-py37h8f50634_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/traitlets-4.3.3-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/zict-2.0.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/arrow-cpp-0.17.1-py37h1234567_11_cpu.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/at-
```

```
spi2-core-2.35.1-h1369247_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/beautifulsoup4-4.9.1-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/cftime-1.2.1-py37h03ebfcd_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/cryptography-3.0-py37hb09aad4_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/dask-core-2.21.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/datashape-0.5.4-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gobject-
introspection-1.64.1-py37h619baee_1.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/h5py-2.10.0-nompi_py37h90cd8ad_104.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/harfbuzz-2.4.0-h9f30f68_3.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/imagecodecs-2020.5.30-py37hda6ee5b_2.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/imageio-2.9.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/importlib_metadata-1.7.0-0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/jupyter_core-4.6.3-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/markdown-3.2.2-py_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/numcodecs-0.6.4-py37he1b5a44_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/oauth2client-4.1.3-py_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/pandas-1.0.5-py37h0da4684_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/poppler-0.87.0-h4190859_1.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/pywavelets-1.1.1-py37h03ebfcd_1.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/scipy-1.5.1-py37ha3d9a3c_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/setuptools-49.2.0-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/shapely-1.7.0-py37hc88ce51_3.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/snuggs-1.4.7-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/spectral-0.21-pyh9f0ad1d_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/streamz-0.5.4-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/sympy-1.6.1-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/terminado-0.8.3-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/uritemplate-3.0.1-py_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/yarl-1.3.0-py37h516909a_1000.tar.bz2
https://conda.anaconda.org/conda-
```

```
forge/linux-64/aiohttp-3.6.2-py37h516909a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/atk-1.0-2.36.0-haf93ef1_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/bleach-3.1.5-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/bs4-4.9.1-0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/distributed-2.21.0-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gdk-pixbuf-2.38.2-h3f25603_4.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/google-auth-1.19.2-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/h5netcdf-0.8.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/inequality-1.0.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/jinja2-2.11.2-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/joblib-0.16.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/jsonschema-3.2.0-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/jupyter_client-6.1.6-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/libgdal-3.0.4-he6a97d6_10.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/matplotlib-base-3.3.0-py37hd478181_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/munch-2.5.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/netcdf4-1.5.3-nompi_py37hdc49583_105.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/networkx-2.4-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/nose-1.3.7-py37hc8dfbb8_1004.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/numba-0.50.1-py37h0da4684_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/numexpr-2.7.1-py37h0da4684_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pango-1.42.4-h7062337_4.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/parquet-cpp-1.5.1-2.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/patsy-0.5.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pint-0.14-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pluggy-0.13.1-py37hc8dfbb8_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/protobuf-3.12.3-py37h3340039_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pygments-2.6.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyjwt-1.7.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyopenssl-19.1.0-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/tifffile-2020.7.17-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/wheel-0.34.2-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/xarray-0.16.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/zarr-2.4.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/atk-2.36.0-0.tar.bz2
https://conda.anaconda.org/conda-
```

```
forge/linux-64/bokeh-2.1.1-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/dask-jobqueue-0.7.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/fastparquet-0.4.1-py37h03ebfcd_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gdal-3.0.4-py37h4b180d9_10.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/google-auth-httplib2-0.0.3-py_3.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/googleapis-common-protos-1.51.0-py37hc8dfbb8_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/graphviz-2.42.3-h0511662_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/nbformat-5.0.7-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/oauthlib-3.0.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pip-20.1.1-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/prompt-toolkit-3.0.5-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pyarrow-0.17.1-py37h1234567_11_cpu.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pytables-3.6.1-py37h56451d4_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pytest-5.4.3-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/rasterio-1.1.5-py37h0492a4a_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/scikit-image-0.17.2-py37h0da4684_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/scikit-learn-0.23.1-py37h8a51577_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/seaborn-base-0.10.1-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/statsmodels-0.11.1-py37h8f50634_2.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/urllib3-1.24.3-py37_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/at-spi2-atk-2.32.0-h1369247_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/botocore-1.17.25-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/dask-2.21.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/fiona-1.8.13-py37h0492a4a_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/ipython-7.16.1-py37h43977f1_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/mapclassify-2.3.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/nbconvert-5.6.1-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/requests-2.24.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/rioxarray-0.0.31-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/seaborn-0.10.1-1.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/linux-64/selenium-3.141.0-py37h8f50634_1001.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/dask-glm-0.2.0-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/geopandas-0.8.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/google-api-core-1.20.1-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/gtk3-3.24.21-h45fd312_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/intake-0.6.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/ipykernel-5.3.3-py37h43977f1_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/libpysal-4.3.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/mechanicalsoup-0.12.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/owslib-0.20.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pooch-1.1.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyct-0.4.6-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pyepsg-0.4.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/quantecon-0.4.8-py37hc8dfbb8_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/rasterstats-0.14.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/requests-oauthlib-1.2.0-pyh9f0ad1d_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/s3transfer-0.3.3-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/sat-stac-0.4.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/threddsclient-0.4.2-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/boto3-1.14.25-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/cartopy-0.18.0-py37h4b180d9_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/colorcet-2.0.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/dask-ml-1.5.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/esda-2.3.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/google-api-python-client-1.10.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/google-auth-oauthlib-0.4.1-py_2.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/google-cloud-core-1.3.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/intake-esm-2020.6.11-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/intake-parquet-0.2.3-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/intake-sql-0.2.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/intake-xarray-0.3.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/notebook-6.0.3-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/panel-0.9.7-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pointpats-2.1.0-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/linux-64/pydap-3.2.2-py37_1000.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/segregation-1.3.0-py_0.tar.bz2
```

```
https://conda.anaconda.org/conda-forge/noarch/spreg-1.1.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/tobler-0.3.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/xgeo-1.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/datashader-0.10.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/gcsfs-0.6.2-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/giddy-2.3.3-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/google-cloud-
storage-1.28.1-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-
forge/noarch/holoviews-1.13.3-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/intake-stac-0.2.3-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/jupyter-server-
proxy-1.5.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/metpy-0.12.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pydrive-1.3.1-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/s3fs-0.4.2-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/spaghetti-1.5.0-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/spglm-1.0.7-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/spvcm-0.3.0-py_0.tar.bz2
https://conda.anaconda.org/conda-
forge/linux-64/widgetsnbextension-3.5.1-py37hc8dfbb8_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/dask-
labextension-2.0.2-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/earthengine-
api-0.1.227-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/geoviews-core-1.8.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/hvplot-0.6.0-pyh9f0ad1d_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/ipywidgets-7.5.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/mgwr-2.1.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/spint-1.0.6-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/geoviews-1.8.1-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/splot-1.1.3-py_0.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/xrviz-0.1.4-py_1.tar.bz2
https://conda.anaconda.org/conda-forge/noarch/pysal-2.2.0-py_0.tar.bz2

Note: you may need to restart the kernel to use updated packages.
```

```
[ ]:
```