* What do you think the most useful DASK feature is?
* Why is the advent of DASK so important?
* About what would you like to learn more?

User face:

1. Dask DataFrame mimcs Pandas
   1. Best used for manipulating large datasets – where datasets do not fit into internal pc storage
   2. Accelerating long computations
   3. Distributed computing on large datasets with standard operations like ‘groupby’, ‘join’, etc.
2. Dask Array mimcs NumPy
   1. Computes on arrays larger than memory
   2. Used in fields like atmospheric and oceanographic science
   3. Uses the threaded scheduler in order to avoid data transfer

Scheduling:

There are two families of schedulers: single machine and distributed scheduler. A single machine scheduler provides basic features on a local process or thread pool. It can only be used on a single machine, and it does not scale. A distributed scheduler offers more features than a single machine scheduler but requires more effort to set up. It can run locally or across a cluster.

1. Local Threads
   1. Executes computations with a local thread pool executor.
   2. Lightweight and requires no setup.
   3. Introduces very little task overhead.
   4. Incurs no costs to transfer data between tasks since everything occurs in the same process
   5. Only provides parallelism when code is majority non-Python code.
2. Local Processes
   1. Executes multiprocessing with a local process pool executor
   2. Lightweight and require no setup.
   3. Able to bypass issues with Python’s Global Interpreter Lock
   4. Provides parallelism on pure Python code
   5. Excellent choice when workflows are linear

The most useful DASK feature is using the familiar user interfaces, especially since we’ve used pandas and numpy in python religiously, coding in either Dask DataFrame or Dask Array would be easier to understand. DASK is important because it allows its user to manage parallel processing in Python. Learning more about Dask DataFrame or Dask Array would be interesting to see how it differs from Pandas and NumPy.