Big Data Processing

Carter Francis

Direct Electron

A Couple of Things Before we Start:

- Doing this yourself is hard, but there are lots of tools available!
  - I've spent lots of time writing and thinking about this, as has everyone else teaching you
  - Don't try to write your own code; try using other people's.
  - Have fun! Your data is ***really*** **cool!**
- Software engineers are (believe it or not) not great at naming things
  - Embarrassingly parallel doesn't mean you should be embarrassed if it doesn't work or seems difficult!
  - Lazy processing is good!
- Ask Questions:
  - Don't be embarrassed!
  - You can always open a discussion on Hyperspy with any questions.

Starting Questions

Where do you analyze data?

a. Laptop
b. Desktop
c. Cluster Computing
d. Other?

How do you store data?

a. Binary File (.mib, .seq, .mrc)
b. HDF5 Format
c. Zarr Format
d. Other?

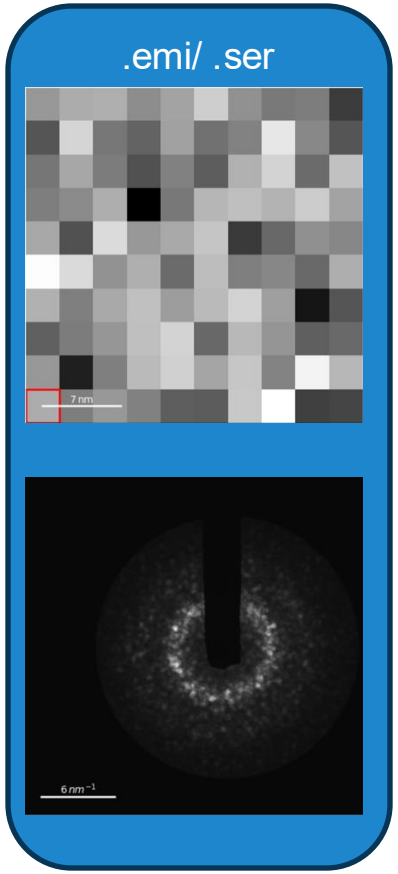
How do you process data?

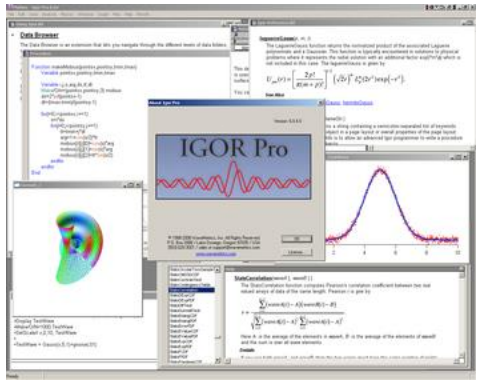
a. Single CPU (python, matplotlib, igor pro)
b. Parallel CPU (python +dask etc., hyperspy, liberTEM)
c. Multiple Node
d. GPU Processing
e. How can I tell?!

What do you want to learn?

a. Processing out of Memory (lazy)
b. Writing code for doing parallel CPU computing
c. How to do machine learning on big datasets
d. How to efficiently build a workflow
e. All of the Above!

My First “4D” STEM Experiment

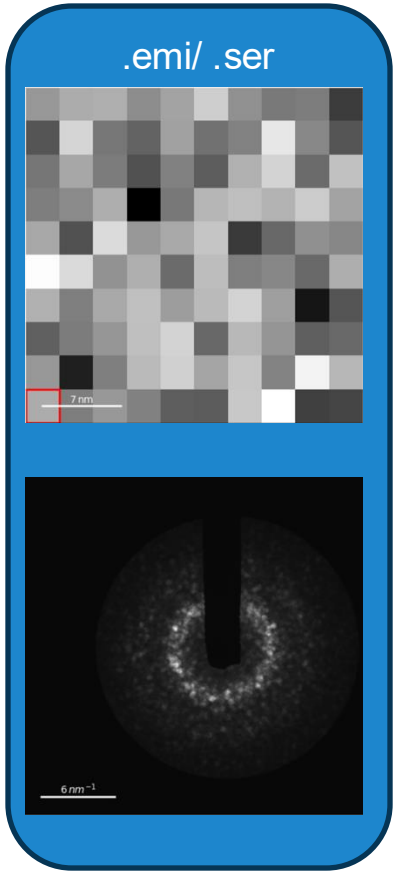


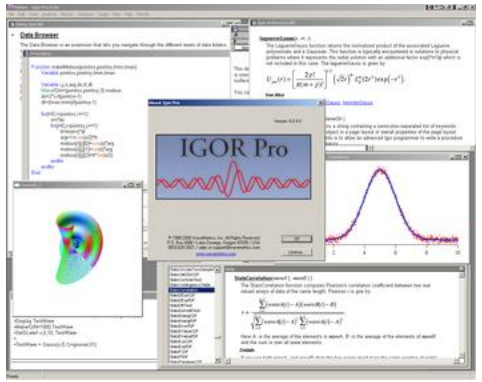


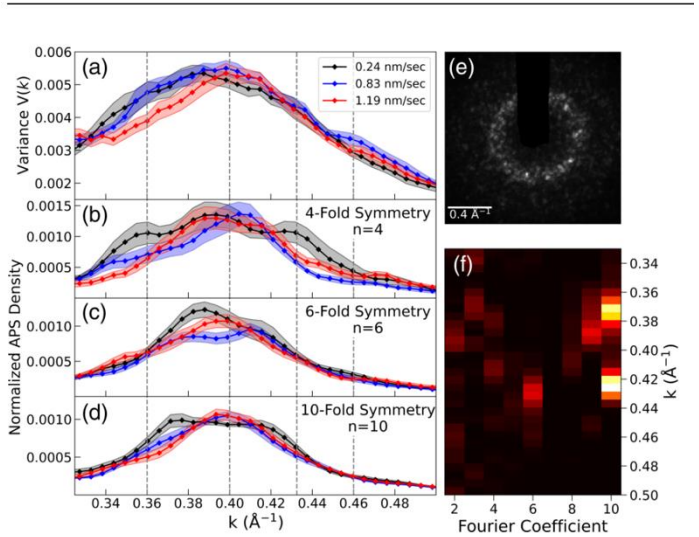
Ran out of Memory!

Total File Size:100 MB

My First “4D” STEM Experiment

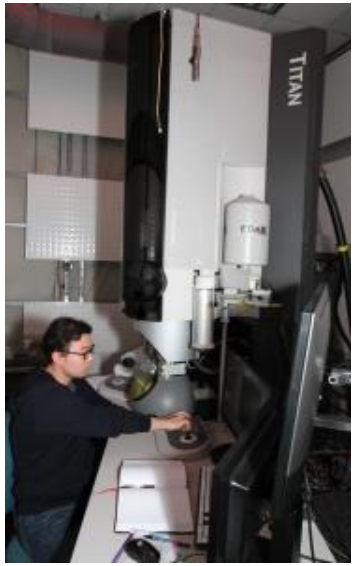


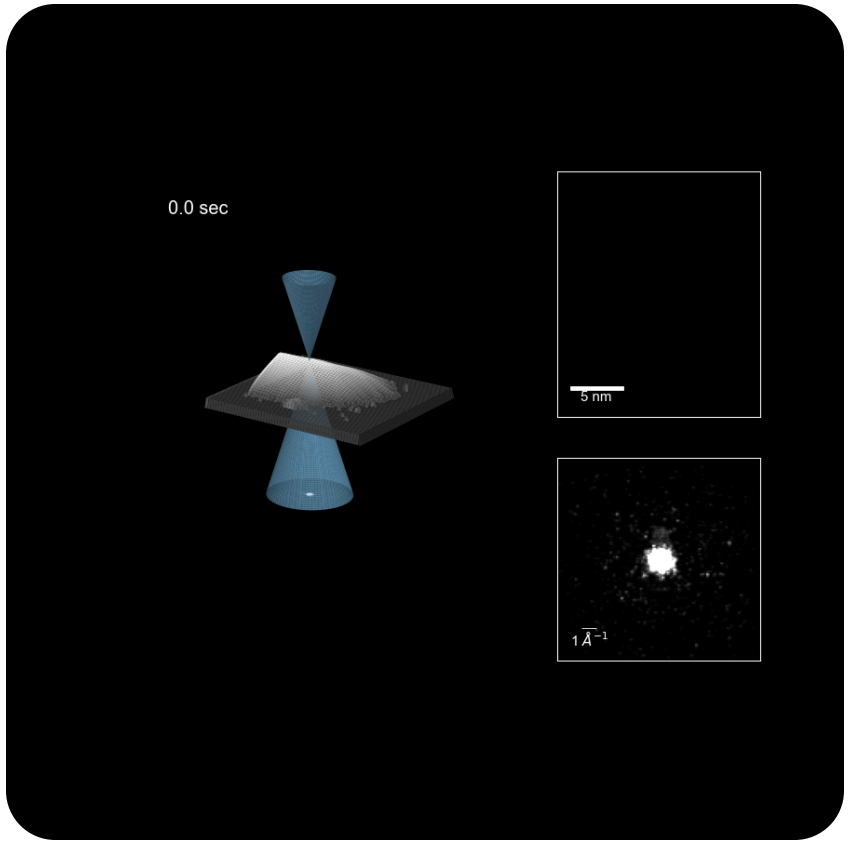




Total File Size:100 MB

My Most Recent “4D” STEM Experiment





Total File Size:1.6 TB

pyxem

HyperSpy

RosettaSciIO

Max Memory usage of ~50GB

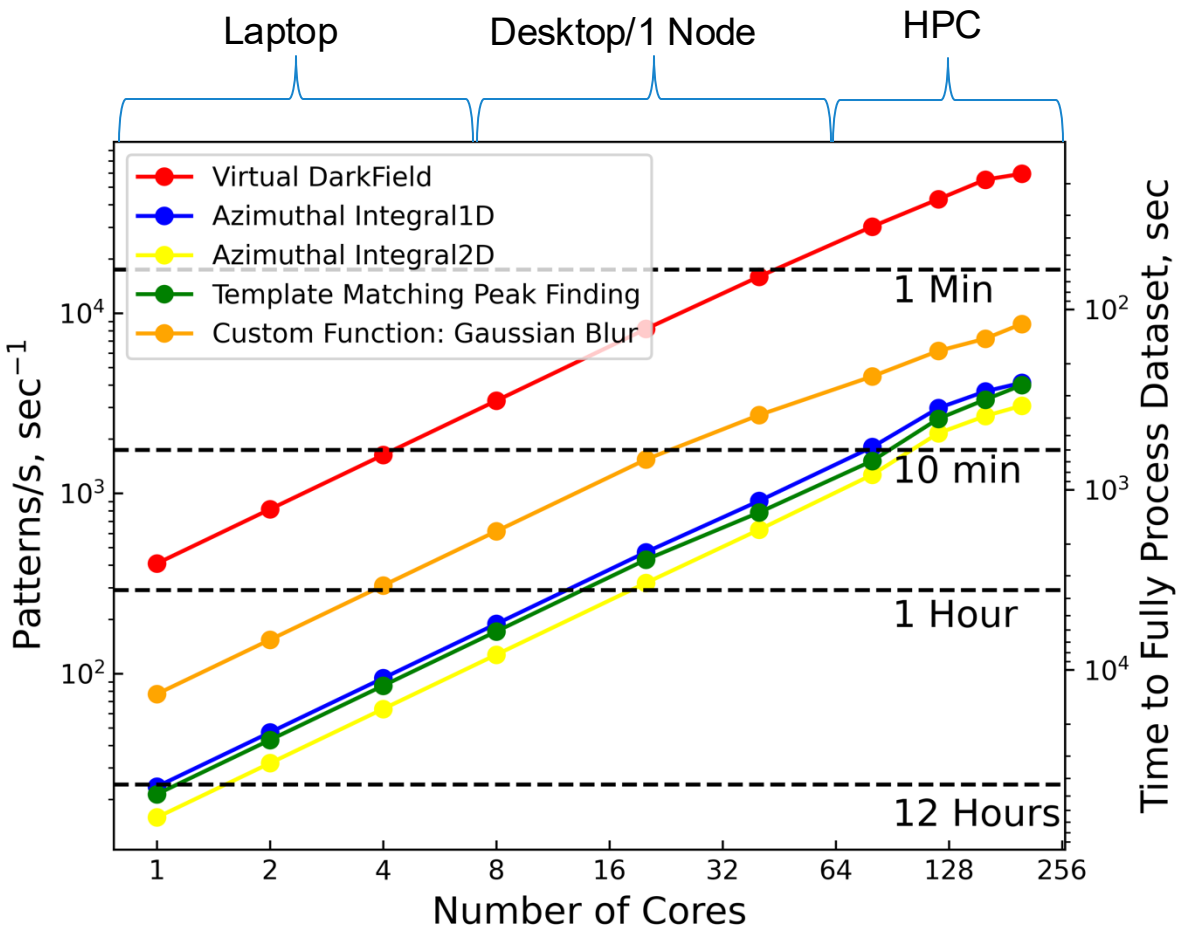
My First "4D" STEM Experiment

Scalable Performance

The entire data cube is 1024x1024x256x256 at 32 bit

This translates to 256 GB!

- Hyperspy and pyxem scales linearly from:
  - Laptop or desktop
  - High powered workstation
  - Multi-node cluster
  - ***All with little to no set up and the same syntax!***
- Every function in hyperspy runs in parallel by default
- Every function works with lazy, out of memory data ***meaning that you are never limited by RAM***

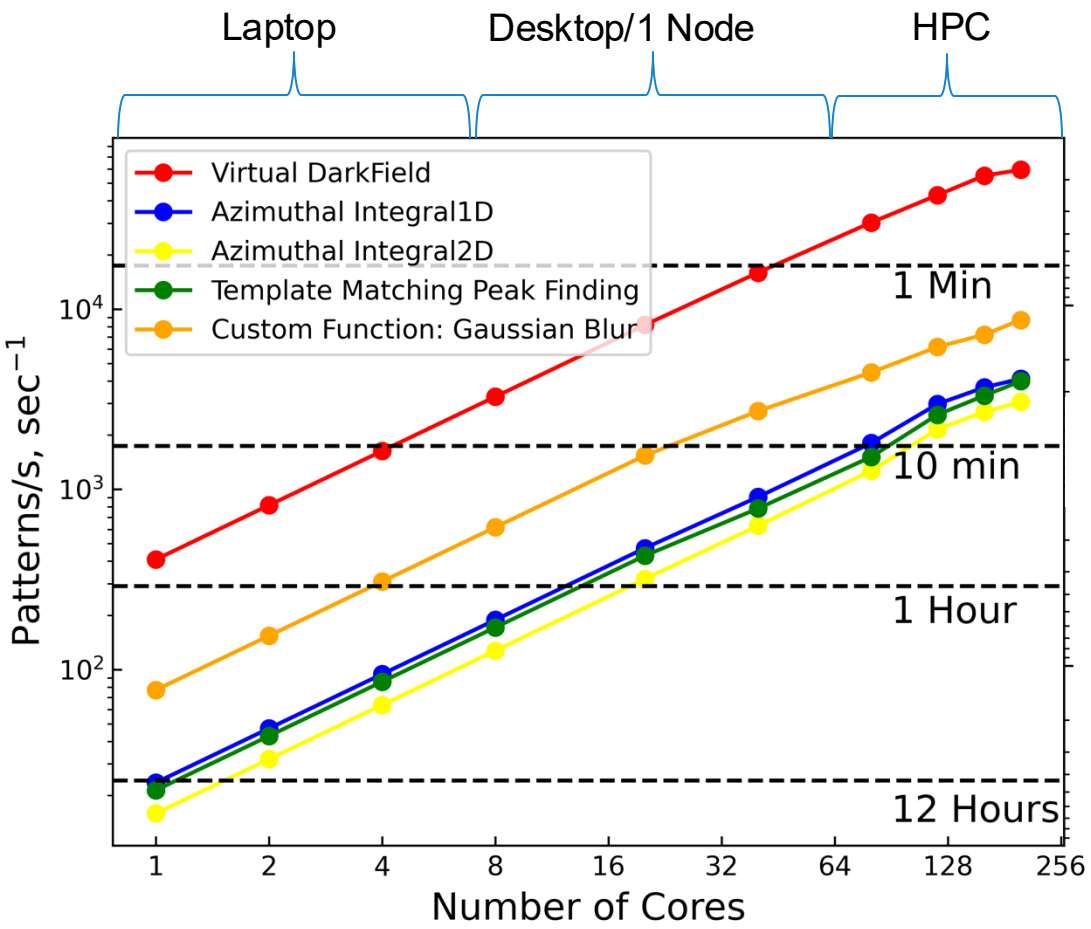


Scalable Performance

The entire data cube is 1024x1024x256x256 at 32 bit

This translates to 256 GB!

- Hyperspy and pyxem scales linearly from:
  - Laptop or desktop
  - High powered workstation
  - Multi-node cluster
  - ***All with little to no set up and the same syntax!***
- Every function in hyperspy runs in parallel by default
- Every function works with lazy, out of memory data ***meaning that you are never limited by RAM***

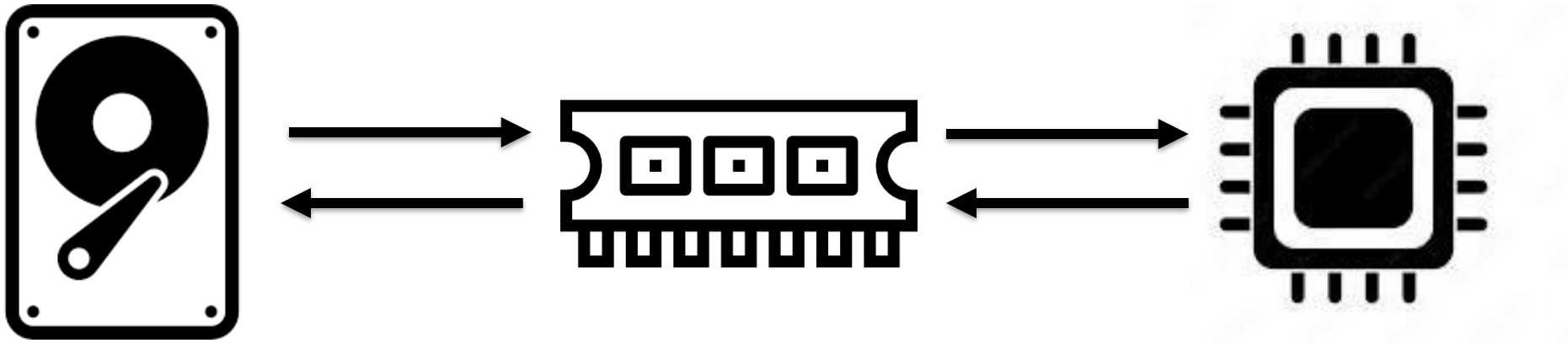


Time to get a coffee

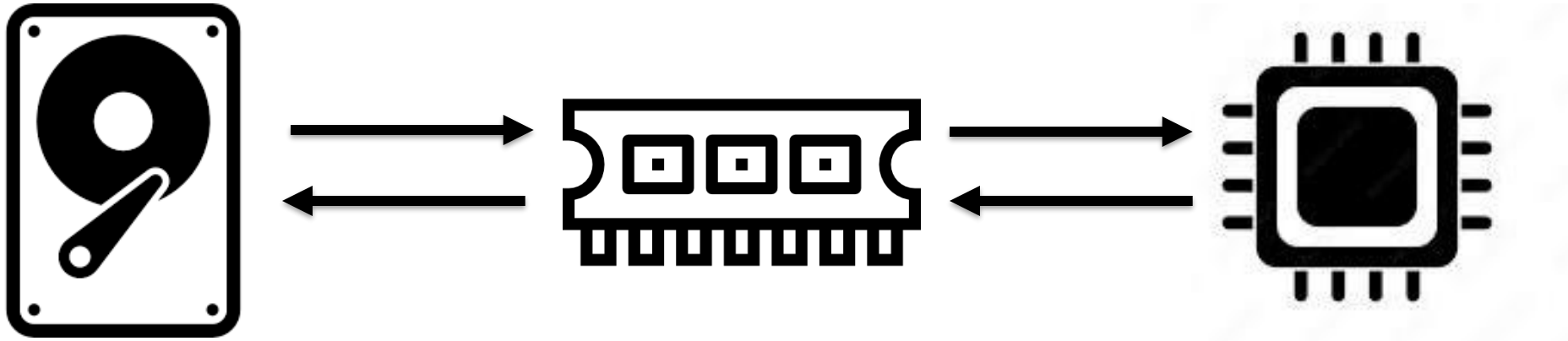
Time to go to the bathroom

Time to get lunch

Time to go home



Hard Drive RAM CPU



Hard Drive

- What kind of hard drive do you have?
  - SSD
  - HDD
  - External Hard Drive
  - Do you have multiple?
  - RAID Array?

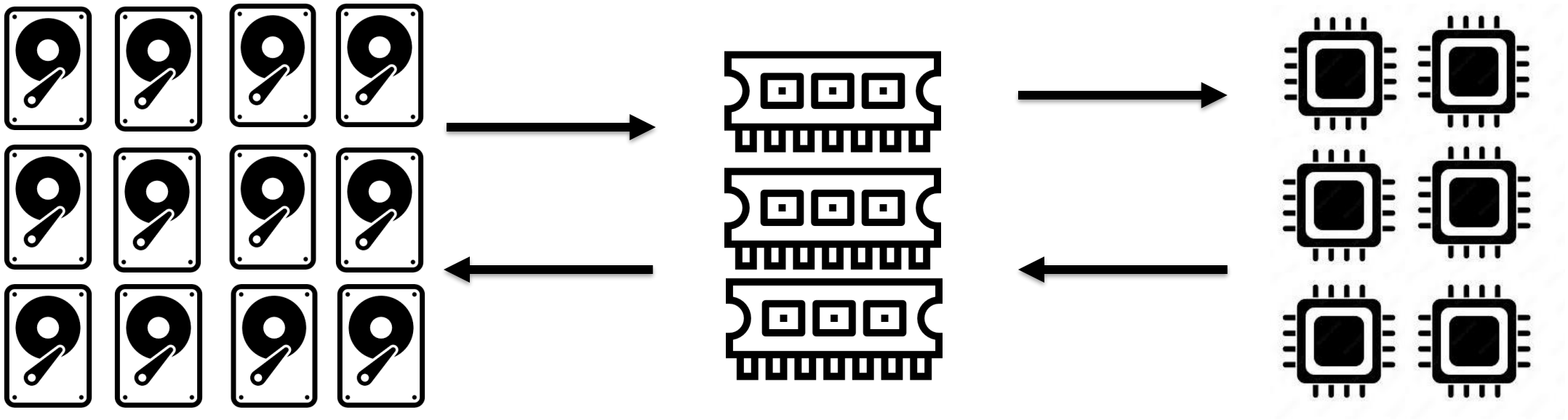
RAM

- How much RAM do you have?
  - Can you load the entire dataset into memory?
  - Can you copy the dataset when it is in memory?

CPU

- How many CPU cores do you have?
  - Are they all running when you try to do something?
  - Are you running on a laptop? Is it charging?

How much money can you spend on computing hardware? What are your bottlenecks?



Hard Drive

RAM

CPU

More Hard Drives = More I/O

More CPUs =
More Processing

Where to go from here?

- Remove barriers to doing experiments:
  - ***Experiments shouldn't be limited by processing***
  - ***Ideas should be easy to implement and explore with minimal setup.***
- Live processing to reduce data size
  - Live finding of diffraction vectors
  - Live preprocessing like direct beam centering
  - Better lazy visualization tools for optimizing parameters
- Don't be afraid to take a large Dataset!