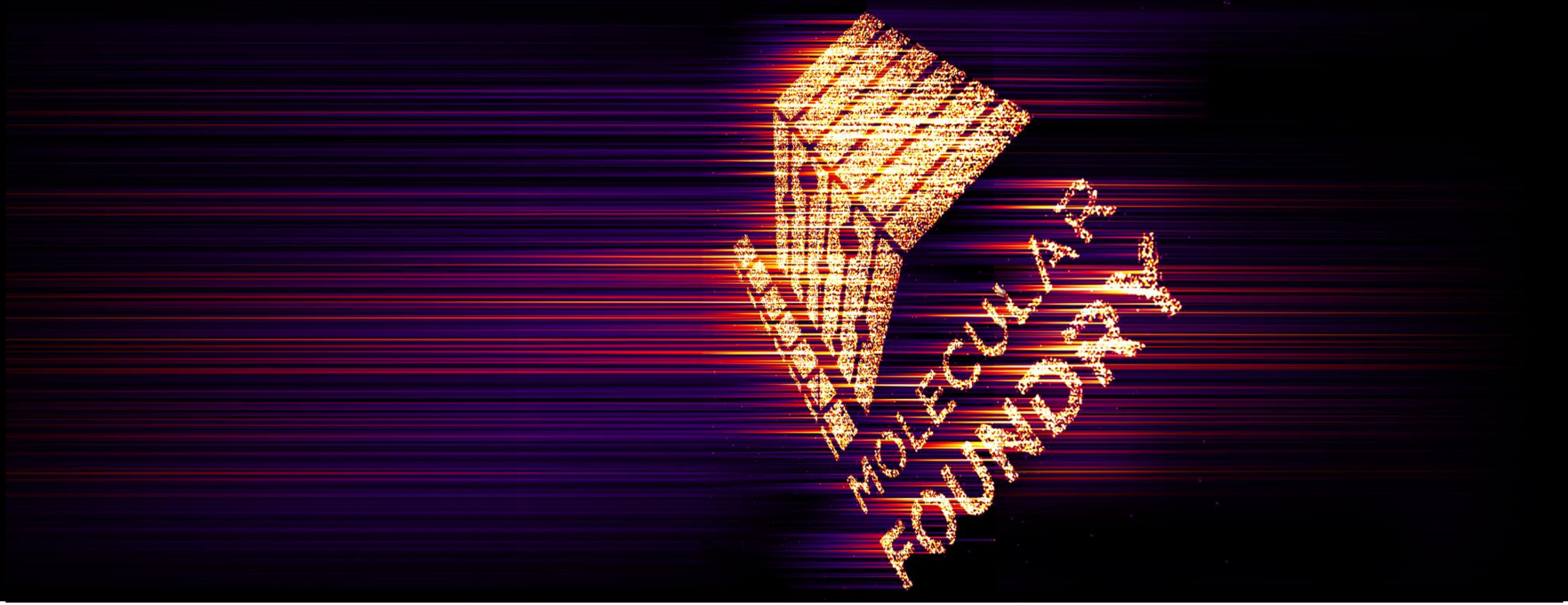


Sparse Data in Scientific Imaging Applications: Transmission Electron Microscopy

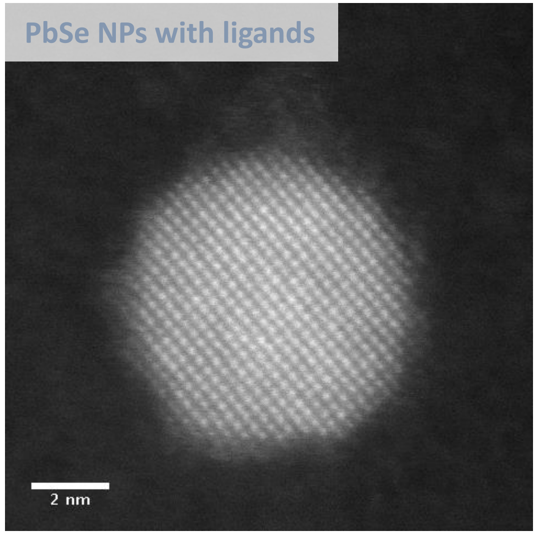
- *Peter Ercius (percius@lbl.gov)*

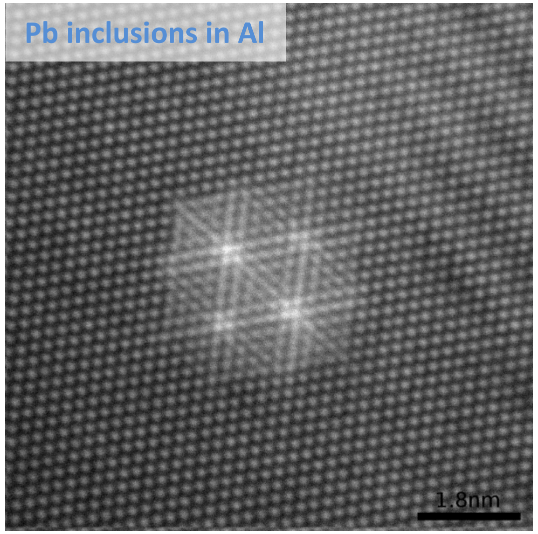
- *HDF5 Users Group Meeting 2021*
- *2021/10/12*

TEM in Materials Science

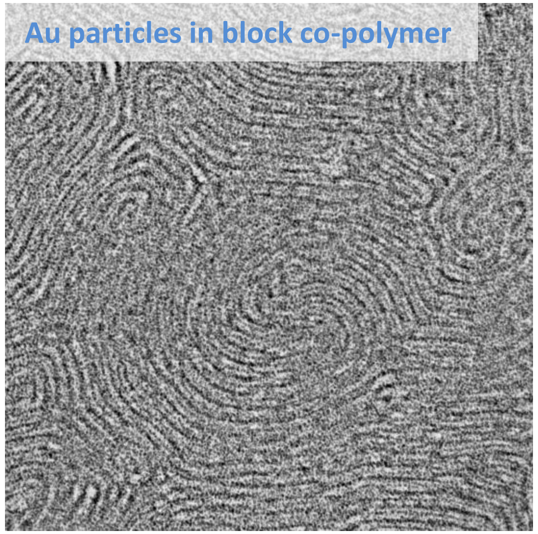
MOLECULAR FOUNDRY

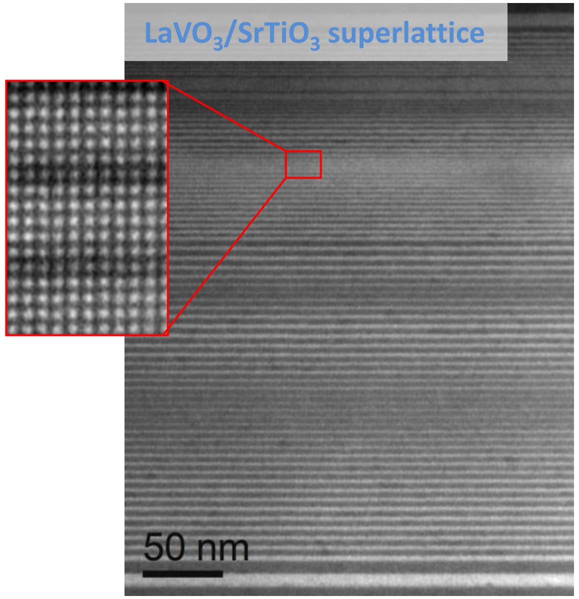
Nanoparticles



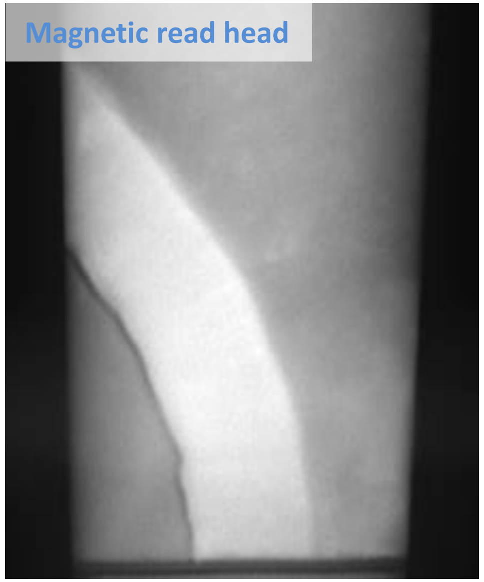


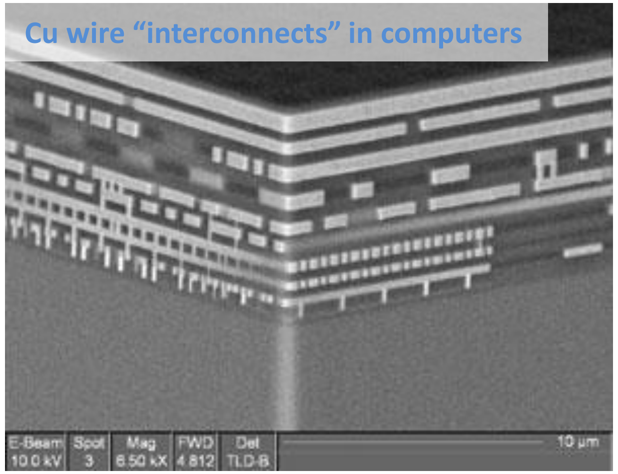
New Materials





Integrated Circuits



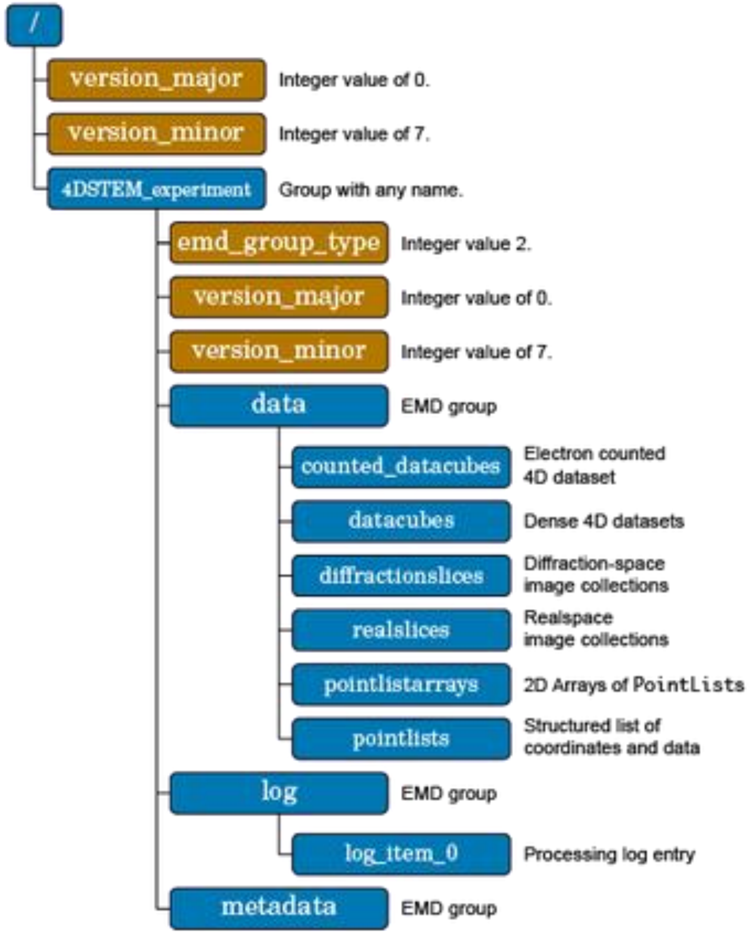


How is HDF5 Used Now In TEM?

MOLECULAR FOUNDRY

- Berkeley Electron Microscopy Dataset (EMD) file
  - Several open-source projects support it: py4DSTEM, ncempy, hyperspy, etc.
  - https://emdatasets.com/
- Detector vendors are switching from closed proprietary formats:
  - Thermo Fischer Velox Files (EMD)
  - Ametek Gatan DM5
- Dataset size is driving the community to adopt an open file type and HDF5 in python (h5py) seems the likely winner.
  - Open, extensible, large HDF5 ecosystem
- Very large data generator: 4D Camera at LBL

EMD v0.1



https://emdatasets.com/format/

Exponentially Increasing Data Rates

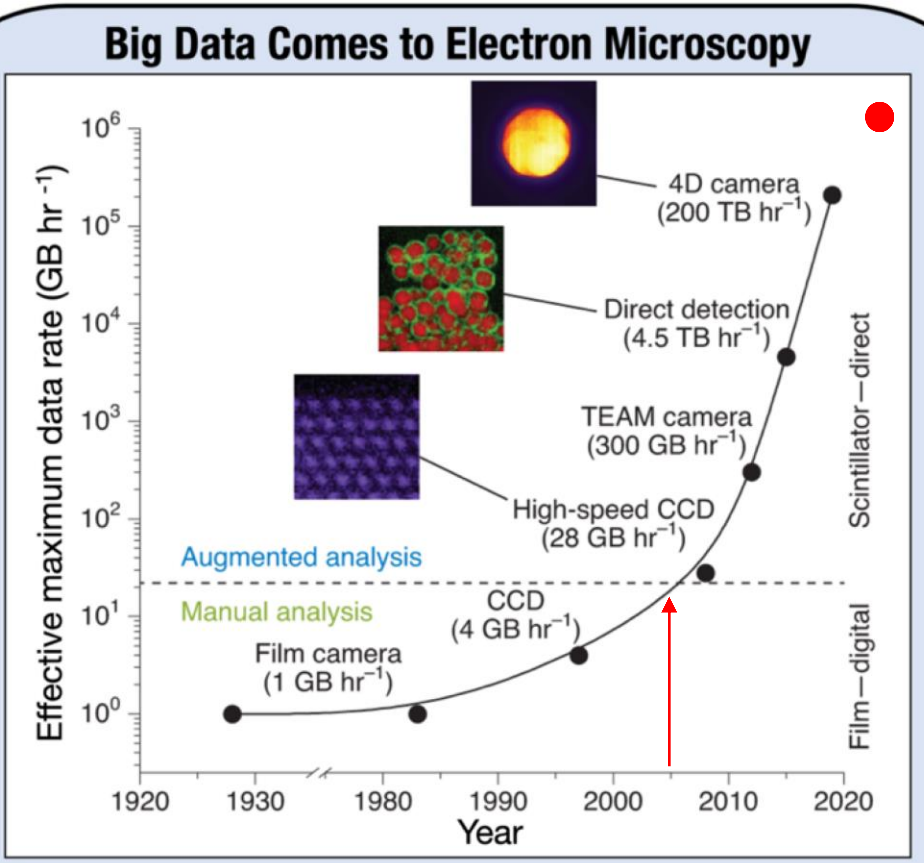
My background in TEM big data

2005 – 2009:
- Tomography (0.5 -1 GB)
- Movies: 1s / frame (< 1GB)

2012 – 2015:
- Faster movies: 0.1 sec / frame (10's GB)

2015 –
- Even faster movies: 0.002 sec / frame (100's GB)

2019 –
- Super fast movies: 10e-6 sec / frame (1000's GBs)



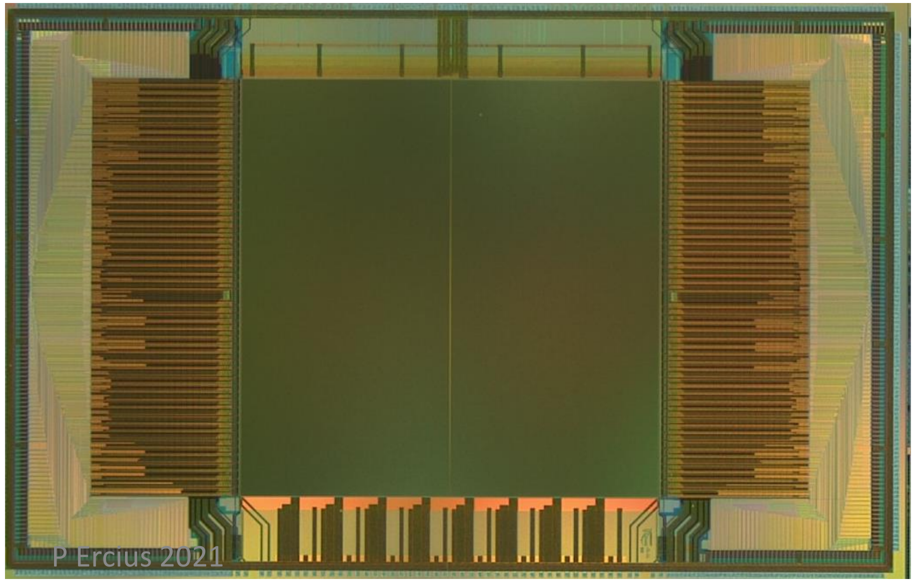
Effective maximum microscope data-production rates by year showing the rapid increase associated with better detector technologies (Spurgeon et al., 2021).

Spurgeon et al, Nature Materials, (2021)

4D Camera Parameters

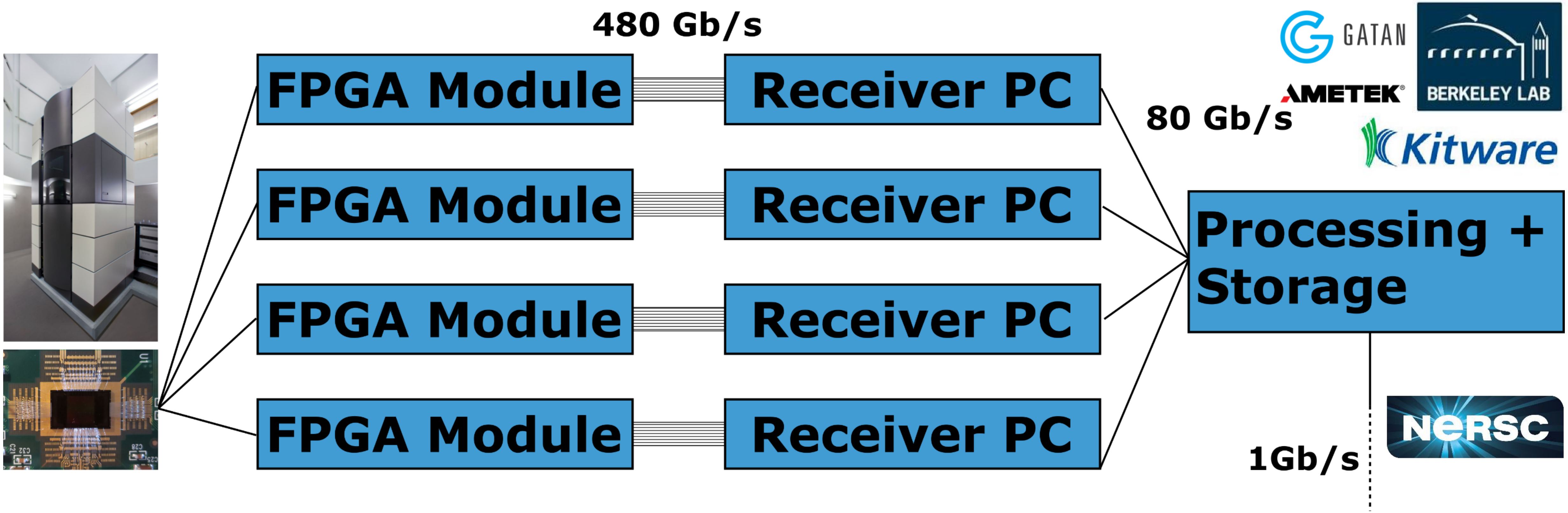
MOLECULAR FOUNDRY

- CMOS Active Pixel Sensor
- 10 μm^2 pixel size
- 11 μsec read out
- Rolling shutter
- 576 x 576 pixels
- Uint16 data output



576x576x2x87000 = 58 GB / sec

Data Acquisition, Processing, and Storage



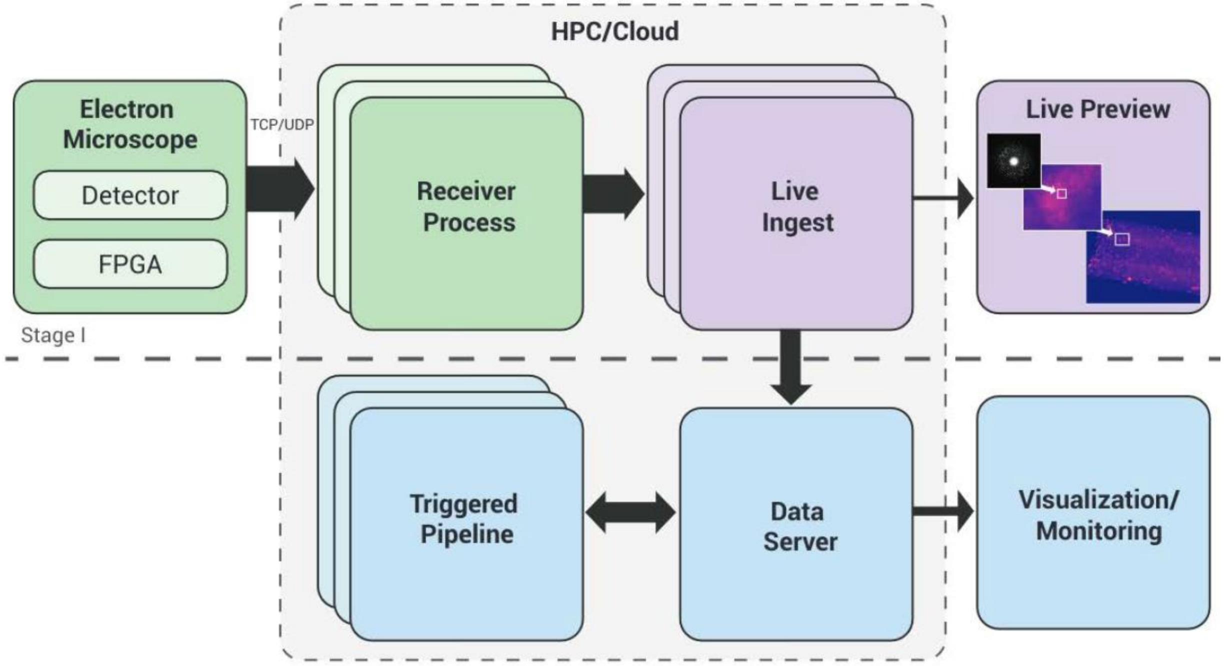
- 87,000 Hz readout
- Typical data set is 650 GB captured in 15 seconds (480 Gbit/s)
- Data pipeline: FPGA → RAM → Flash storage → Sparse HDF5

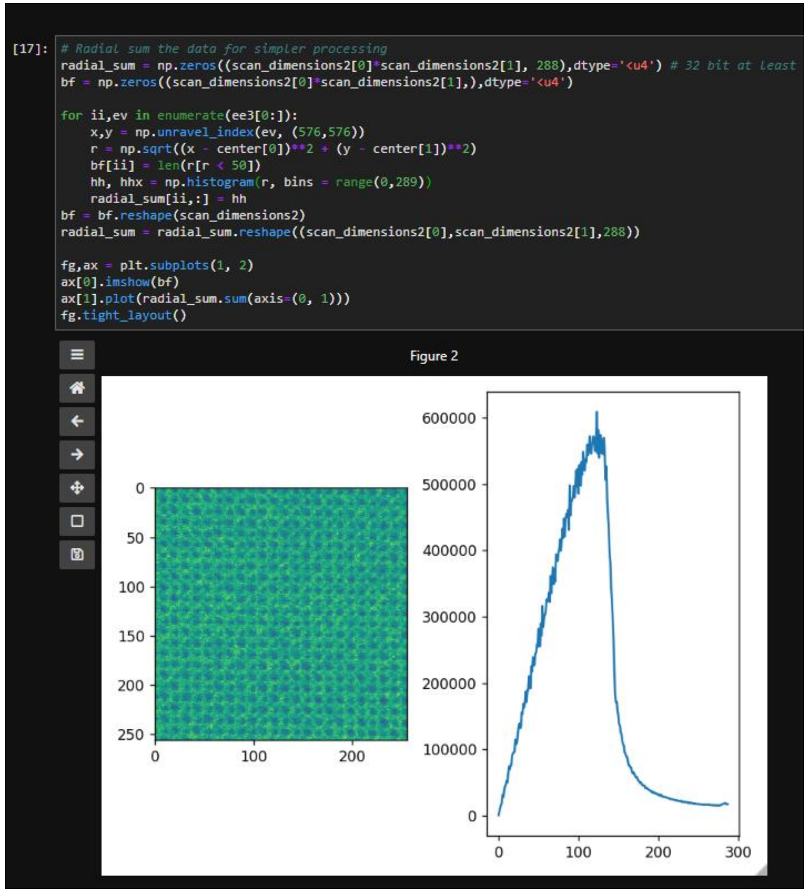
15 sec **120 sec** **8.5 min**

Rapid to Live Processing

Kitware https://github.com/OpenChemistry/stempy

jupyter NERSC



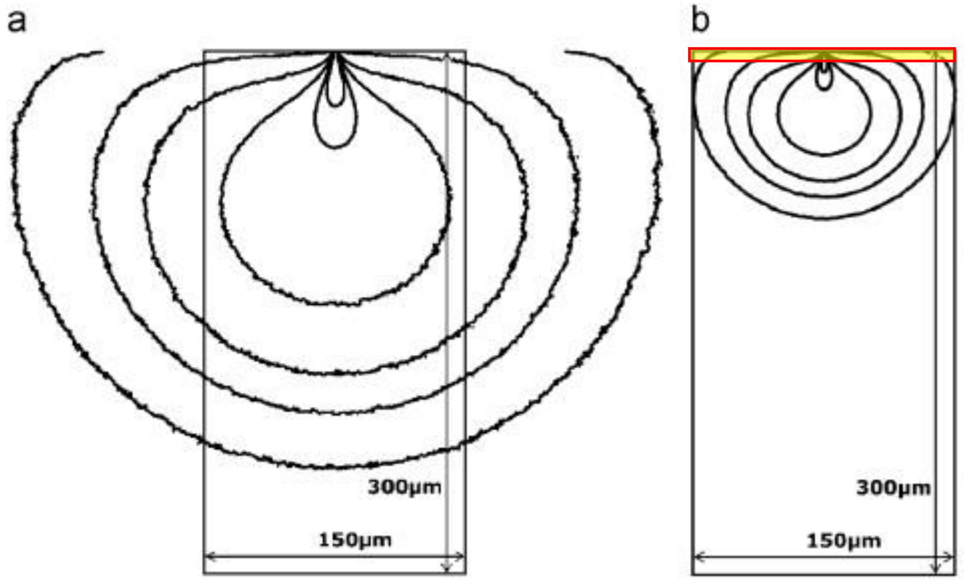


- Powerful open-source processing ecosystem
  - Local and remote high-performance computing
- Is the data useful?
  - Scientists need minute scale reduction, efficient data representation and storage
  - Improve time to useful information
  - Fits in memory, easily processed

Kitware M Hanwell, BNL
C Harris, Kitware, Inc.

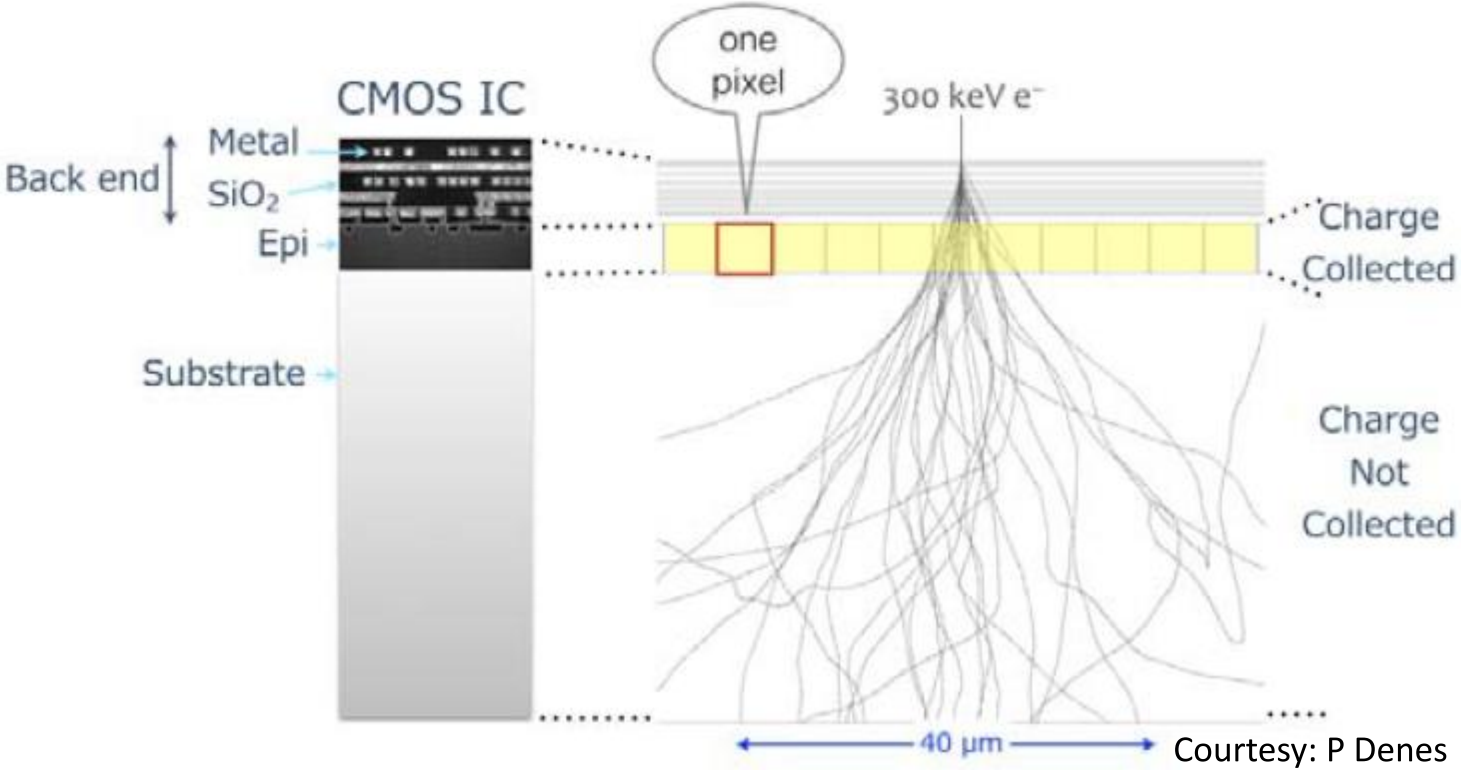
Detector Energy Deposition

<u>300 kV</u> <u>120 kV</u>



TA Caswell, P Ercius, et al., Ultramicroscopy (2009)

<u>4D Camera Pixel</u>

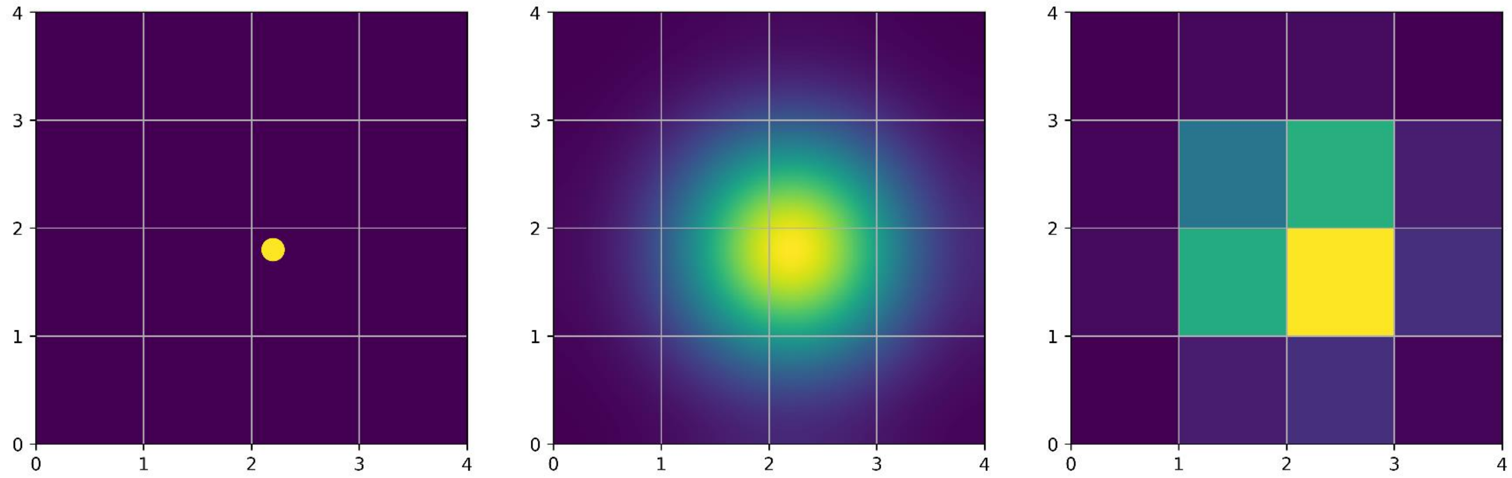


Courtesy: P Denes

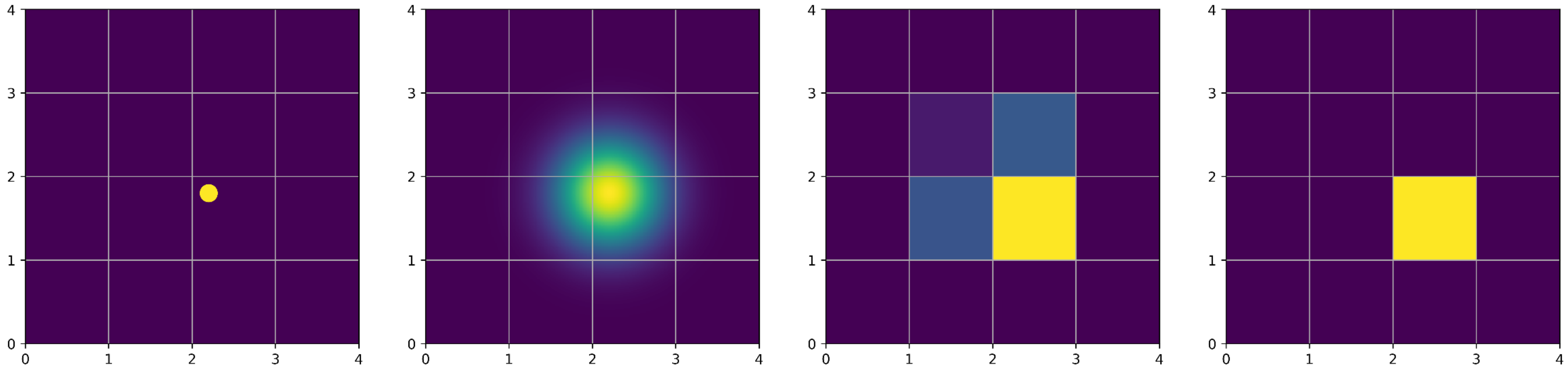
- Electrons scatter very strongly and deposit energy in depth and laterally
- Big thick pixels (EMPAD, etc.) or small thin pixels (**K3 and 4D Camera**)

Signal Summation vs Electron Counting

CCD Signal Summation PSF

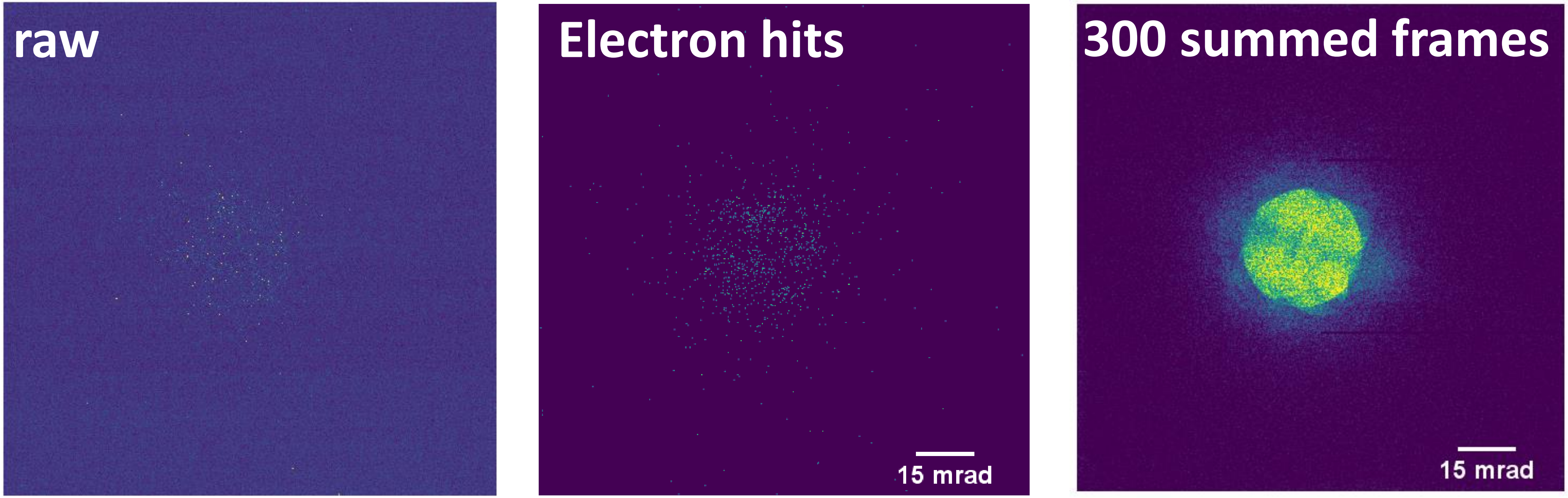


CMOS Electron Counting



M Battaglia, et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 608, 363–365.

What single frames look like

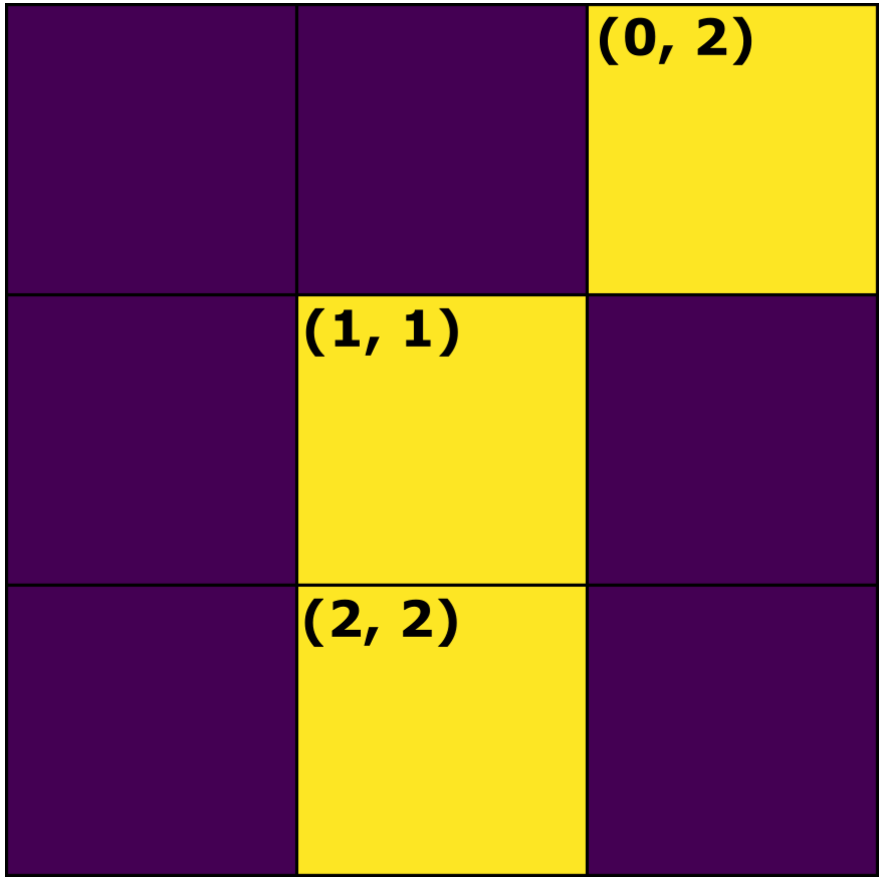


- Raw noisy frame, counted frame, sum of 300 frames
- 1% fill ‘factor’ allows ~3300 electrons per frame
- **30 – 100x** data reduction (650 GB → <20 GB)

HDF5: Where's the Sparcity?

- In numerical analysis and scientific computing, a sparse matrix or sparse array is a matrix in which most of the elements are zero.
  - There is no strict definition how many elements need to be zero for a matrix to be considered sparse but a common criterion is that the number of non-zero elements is roughly the number of rows or columns.
  - The number of zero-valued elements divided by the total number of elements is sometimes referred to as the sparsity of the matrix.
- HDF5 is a very popular open source I/O middleware package
  - Developed primarily by teams at the HDF Group and Berkeley Lab
  - Broadly regarded as most widely adopted I/O middleware in DOE
  - Widely used beyond science - also engineering, finance, and many other communities
- Does not store sparse data in an efficient, performant, or portable way

Linear Encoding Sparse Data Representation

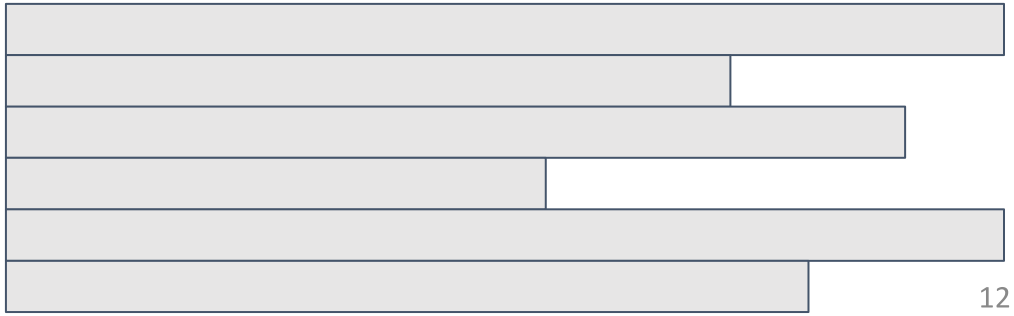


0	1	2
2	1	2

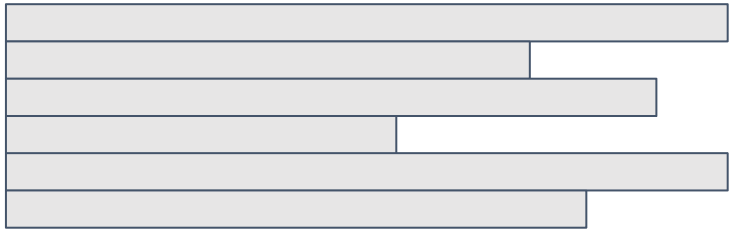
2	4	7

```
[array([  3157,   3286,   4484, ..., 323468, 328455, 329234], dtype=uint32),
     array([], dtype=uint32),
     array([   863,   3619,   4126, ..., 323910, 331405, 331406], dtype=uint32),
     ...,
     array([  2618,   7713,   7897, ..., 326856, 328006, 329049], dtype=uint32),
```

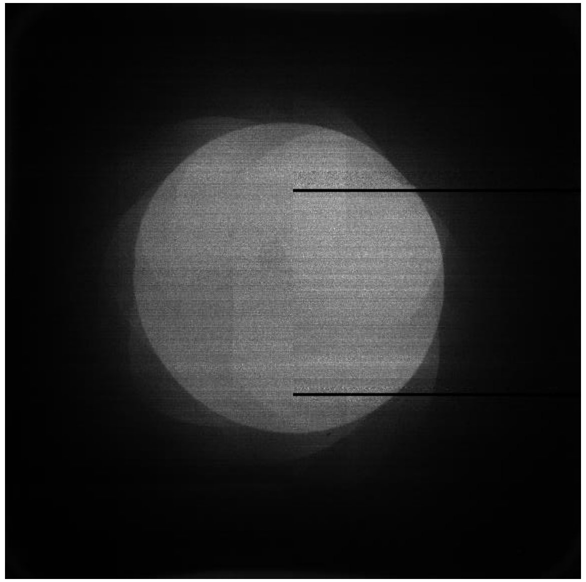
Sparse “Ragged” array in HDF5



Requirement: Sparse → Dense Data

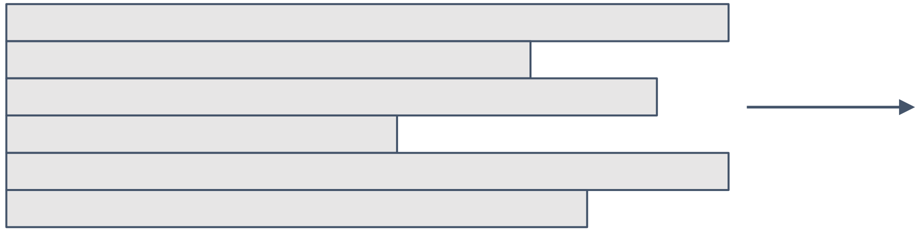


```
def sparse_to_dense():
    …
    return 2d_array
```



- HDF5 returns a normal “Dense” array
  - Transparent to the user and upstream packages
  - Can slice and operate like a normal array
  - “Lazy” loading to only compute what you need when you need it
  - Sparse data operates within “dense only” existing code

Requirement: Sparse Domain Operations



```
def sparse_domain():
    …
    return 2d_array
```

	dense format	linear index encoding	run-length encoding
bin	$O(\mathsf{M} \cdot \mathsf{F})$	$O(\mathsf{C} \cdot \mathsf{F})$	$O(2\mathsf{C} \cdot \mathsf{F})$
crop	$O(\mathsf{M} \cdot \mathsf{F})$	$O(\mathsf{C} \cdot \mathsf{F})$	$O(2\mathsf{C} \cdot \mathsf{F})$
center of mass	$O(\mathsf{M} \cdot \mathsf{F})$	$O(\mathsf{C} \cdot \mathsf{F})$	$O(2\mathsf{C} \cdot \mathsf{F})$
radial sum	$O(\mathsf{r}_1^2 - \mathsf{r}_2^2) \cdot \mathsf{F}$	$O(\mathsf{C} \cdot \mathsf{F})$	$O(2\mathsf{C} \cdot \mathsf{F})$
sum all frames	$O(\mathsf{M} \cdot \mathsf{F})$	$O(\mathsf{C} \cdot \mathsf{F})$	$O(2\mathsf{C} \cdot \mathsf{F})$

TABLE I

COMPUTATIONAL COMPLEXITY OF COMMON OPERATIONS IN 4D-STEM WITH DIFFERENT ENCODING SCHEMES, FOR M DETECTOR PIXELS, F IMAGE FRAMES, C ELECTRON COUNTS, AND AN r_1 TO r_2 RADIAL RANGE.

P Pelz, et al, arXiv:2104.06336

P Pelz, et al, IEEE Signal Processing Magazine (accepted)

“Extreme Scale Sparsity” Science Use Cases

MOLECULAR FOUNDRY

- NCEM 4D Camera
  - General sparse array
  - Sparsity >100x
- SLAC LCLS-II Data Reduction Pipeline
  - Regions-of-interest and point-lists
  - Sparsity of 10-1000x
- DUNE Experiment Particle Detectors
  - Point-lists
  - Sparsity of >1000x
- Graph neural networks (GNNs)
  - Adjacency matrices
  - Sparsity 1000-10,000x
- Dense storage of this data is ~TB scale, sparse storage is ~GB scale
- Need a solution that enables all users of sparse data to benefit

Acknowledgements

4D Camera Team and Collaborators

P Pelz, A Minor, MC Scott, C Ophus, J Ciston, P Ercius
National Center for Electron Microscopy, Molecular Foundry, LBL

I Johnson, J Joseph, A Goldschmidt, P Denes
Detector group, Engineering Division, LBL

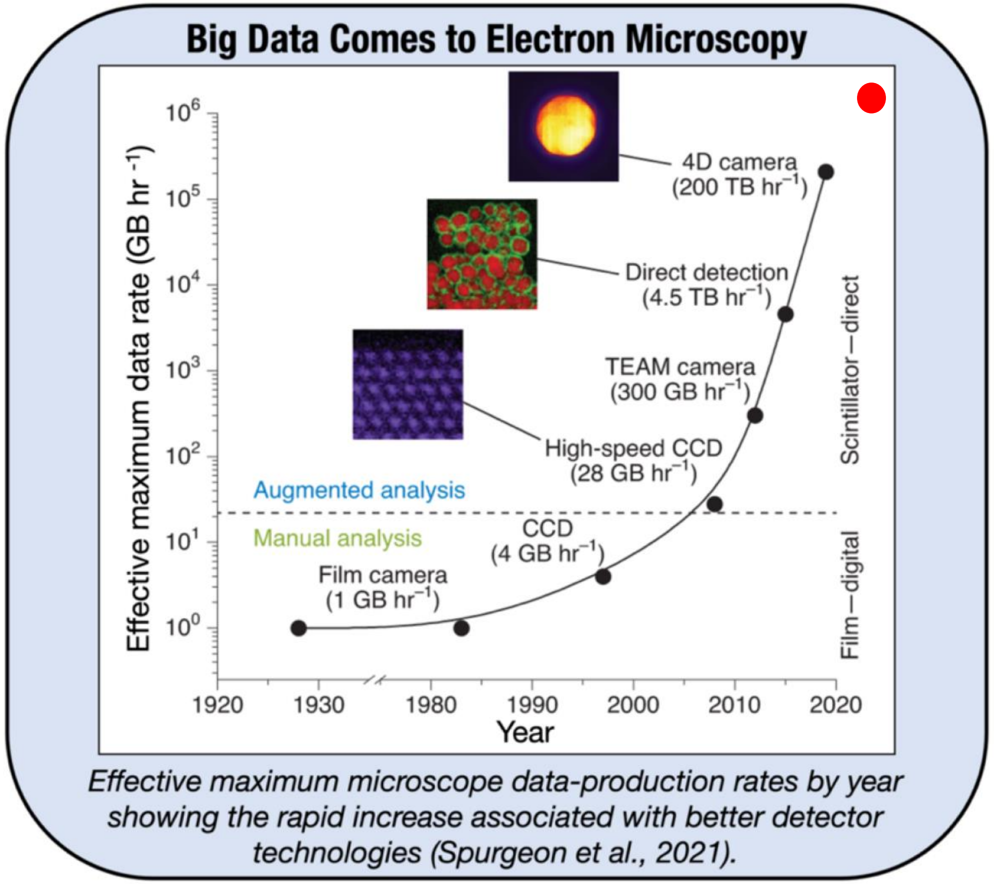
D Paul, B Draney, J Lee, D Skinner, A Selvarajan
National Energy Research Scientific Computing Center (NERSC), LBL

M Lent, P Manacop, C Czarnik
Gatan, Inc.; Ametek

C Harris, P Avery, A Genova

Kitware

M Hanwell
Brookhaven National Laboratory



Effective maximum microscope data-production rates by year showing the rapid increase associated with better detector technologies (Spurgeon et al., 2021).

Spurgeon et al, Nature Materials, (2021)