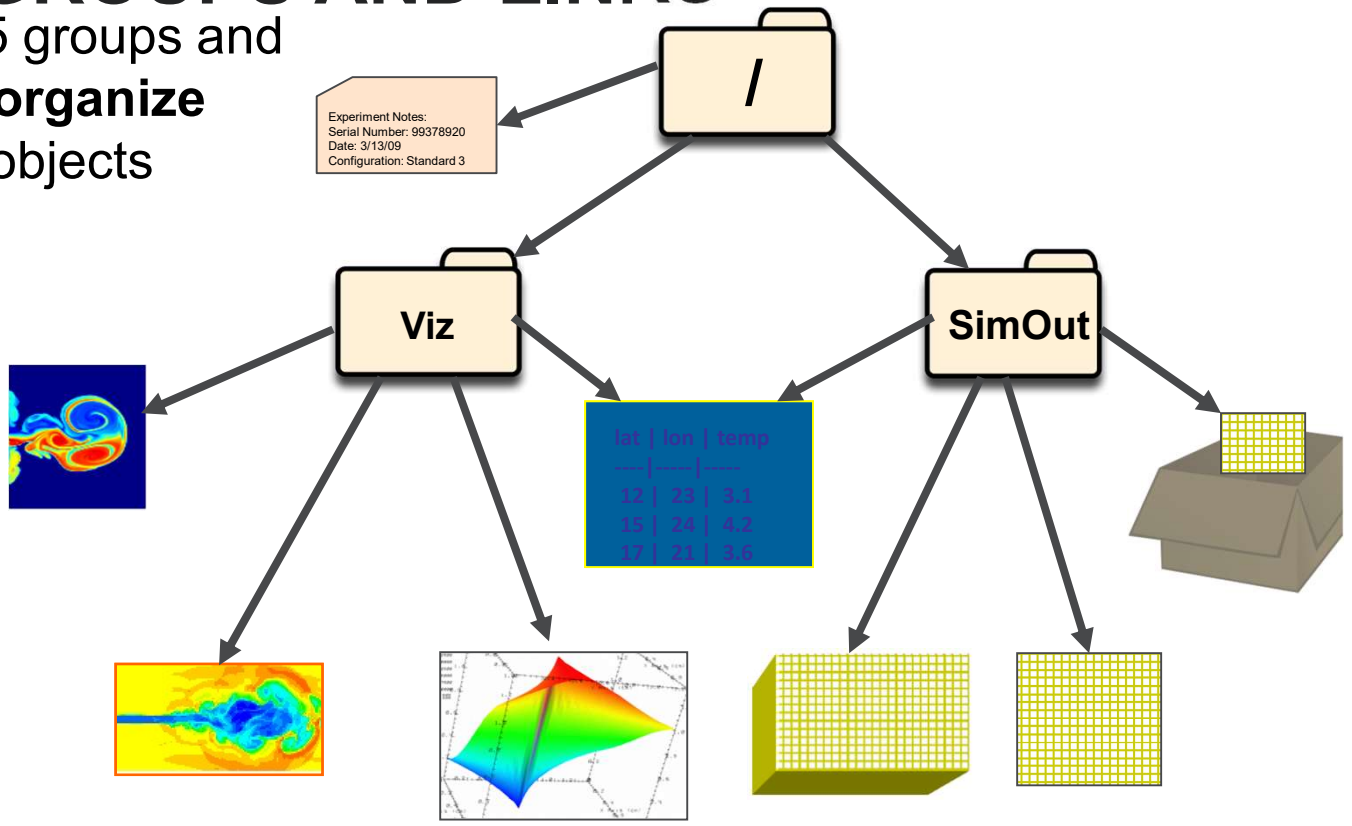
The HDF5 Interface and File Format

HDF5

- Hierarchical Data Format, from The HDF Group (formerly of NCSA)
  - https://www.hdfgroup.org/
- Data Model:
  - Hierarchical data organization in single file
  - Typed, multidimensional array storage
  - Attributes on any HDF5 "object" (dataset, data, groups)
- Features:
  - C, C++, Fortran, Java (JNI) interfaces
    - Community-supported Python, Lua, R
  - Portable data format
  - Optional compression (even in parallel I/O mode)
  - Chunking: efficient row or column oriented access
  - Noncontiguous I/O (memory and file) with hyperslabs
- Parallel HDF5 tutorial:
  - https://portal.hdfgroup.org/display/HDF5/Introduction+to+Parallel+HDF5

HDF5 GROUPS AND LINKS

HDF5 groups and links **organize** data objects



INITIALIZE THE FILE FOR PARALLEL ACCESS

```
/* first initialize MPI */

/* create access property list */
plist_id = H5Pcreate(H5P_FILE_ACCESS);

/* necessary for parallel access */
status = H5Pset_fapl_mpio(plist_id,
MPI_COMM_WORLD, MPI_INFO_NULL);

/* Create an hdf5 file */
file_id = H5Fcreate(FILENAME,
H5F_ACC_TRUNC, H5P_DEFAULT, plist_id);

status = H5Pclose(plist_id);
```

CREATE PROPERTY LIST

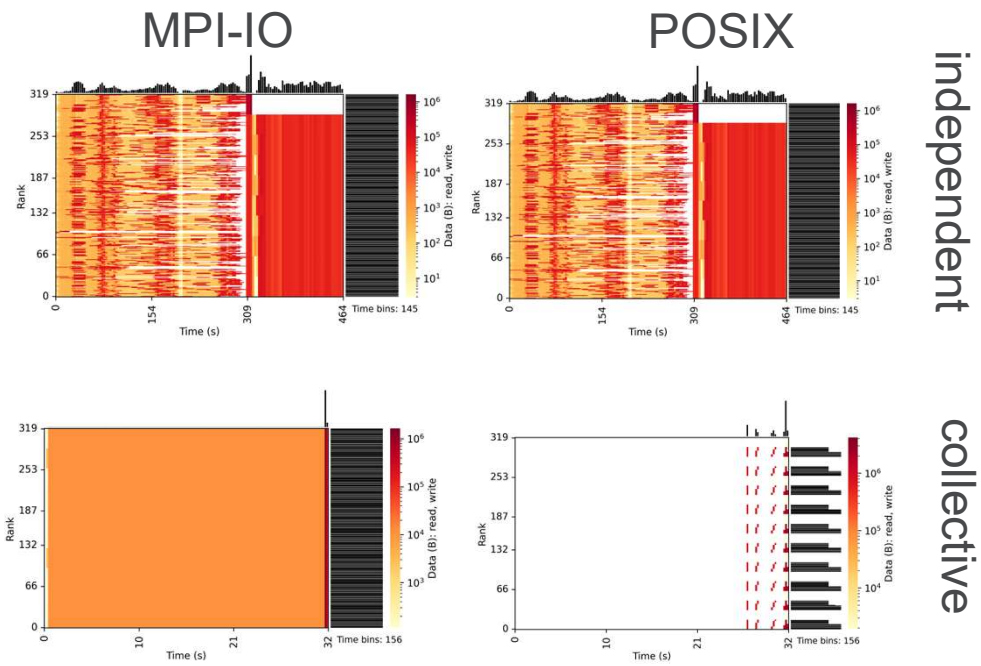
```
/* Create property list for collective dataset
write. */

plist_id = H5Pcreate(H5P_DATASET_XFER);

/* The other option is HDFD_MPIO_INDEPENDENT */
H5Pset_dxpl_mpio(plist_id,H5FD_MPIO_COLLECTIVE);
```

EFFECT OF HDF5 TUNING

- HDF5 property lists can have big impact on internal operations

- Collective I/O vs. Independent I/O
  - Huge reduction in operation count
  - Implies all processes hit I/O at same time

- Collective metadata (new in 1.10.2)
  - Further reduction in op count, especially reads (reading HDF5 internal layout information)
  - Big implications for performance at scale



https://github.com/radix-io/io-sleuthing/tree/main/examples/hdf5

NEW HDF5 FEATURES:

- New in HDF5-1.14.0
- Async operations
  - Potential for background progress
- Multi-dataset I/O
  - Similar to pnetcdf “operation combining”

DATA MODEL I/O LIBRARIES

- Parallel-NetCDF: http://www.mcs.anl.gov/pnetcdf
- HDF5: http://www.hdfgroup.org/HDF5/
- NetCDF-4: http://www.unidata.ucar.edu/software/netcdf/netcdf-4/
  - netCDF API with HDF5 back-end
- ADIOS: http://adiosapi.org
  - Configurable (xml) I/O approaches
- SILO: https://wci.llnl.gov/codes/silo/
  - A mesh and field library on top of HDF5 (and others)
- H5part: http://vis.lbl.gov/Research/AcceleratorSAPP/
  - simplified HDF5 API for particle simulations
- GIO: https://svn.pnl.gov/gcrm
  - Targeting geodesic grids as part of GCRM
- PIO:
  - climate-oriented I/O library; supports raw binary, parallel-netcdf, or serial-netcdf (from master)
- … Many more: consider existing libs before deciding to make your own.
- Note absence of a “machine learning” library – research opportunity for someone!