Introducing HDF5 - A new storage format for your data

Chris Hall

IMBL

Our current storage demands

- IMBL CT requires > 1000 images to be collected and stored.
- IMBL allows moderate resolution for each image over a wide area => many pixels per image
- The SR beam and detector limits means individual projections can be made of several images.
- => Many arrays need to be stored.
- E.g. Elvis the rhino: raw data 1.31 TB in 30 CT sets: 336,390 files.

Future storage issue

- IMBL is designed for moderate spatial resolution imaging of large objects. We have not yet fully exploited this capability
- As well as X, Y, theta, & Z we might also add time, and even a spectral dimension to the data stream
- => The quantity and speed of data that needs to be stored is going to increase

Data storage efficiency

- Storing each image in a single file is very inefficient.
  - Data storage is slow
  - Directories take long times to list
  - Management of the data is slow and awkward
  - It is separated from the ‘meta-data’ (everything you would normally record in your experiment log book.)

Towards a data store standard…

- There have been several attempts to agree on a tomography data storage format e.g.
  - APS : Scientific Data Exchange
  - International: NeXus and PANData Formats
- Many rely on the concept of a data ‘container’ which keeps relevant data together in one place.

Hierarchical Data Format 5 (HDF5)

- An HDF5 file is a container for storing various data
- An HDF5 file is self describing… You can figure out where and what the data is by looking at it
- HDF5 is mature, and used in many other areas e.g. financial services
- It is composed of two primary types of objects: groups and datasets.
  - **HDF5 group:** a grouping structure containing zero or more HDF5 objects, together with supporting metadata
  - **HDF5 dataset:** a multidimensional array of data elements, together with supporting metadata

Introduction to HDF5

- Any HDF5 group or dataset may have an associated attribute list.
  - An **HDF5 attribute** is a user-defined HDF5 structure that provides extra information about an HDF5 object.
- Working with groups and datasets is similar in many ways to working with directories and files in Linux. As with Linux directories and files, an HDF5 object in an HDF5 file is often referred to by its **full path name** (also called an **absolute path name**).
  - / signifies the root group.
    /blah signifies a member of the root group called blah.
    /blah/blah signifies a member of the group blah (which in turn is a member of the root group blah)

The goal for HDF5 on IMBL:

- Each sample will have all its raw data saved in a single HDF5 file. (This will eventually include the calibration images (F&D), but probably not in the first instance)
- Serial scans and other protocols will be kept in an N-dimensional data array within the HDF5 file
- Stitched and corrected projection images will be stored in a separate HDF5 file
- Reconstructed data will be stored in a third HDF5 file.

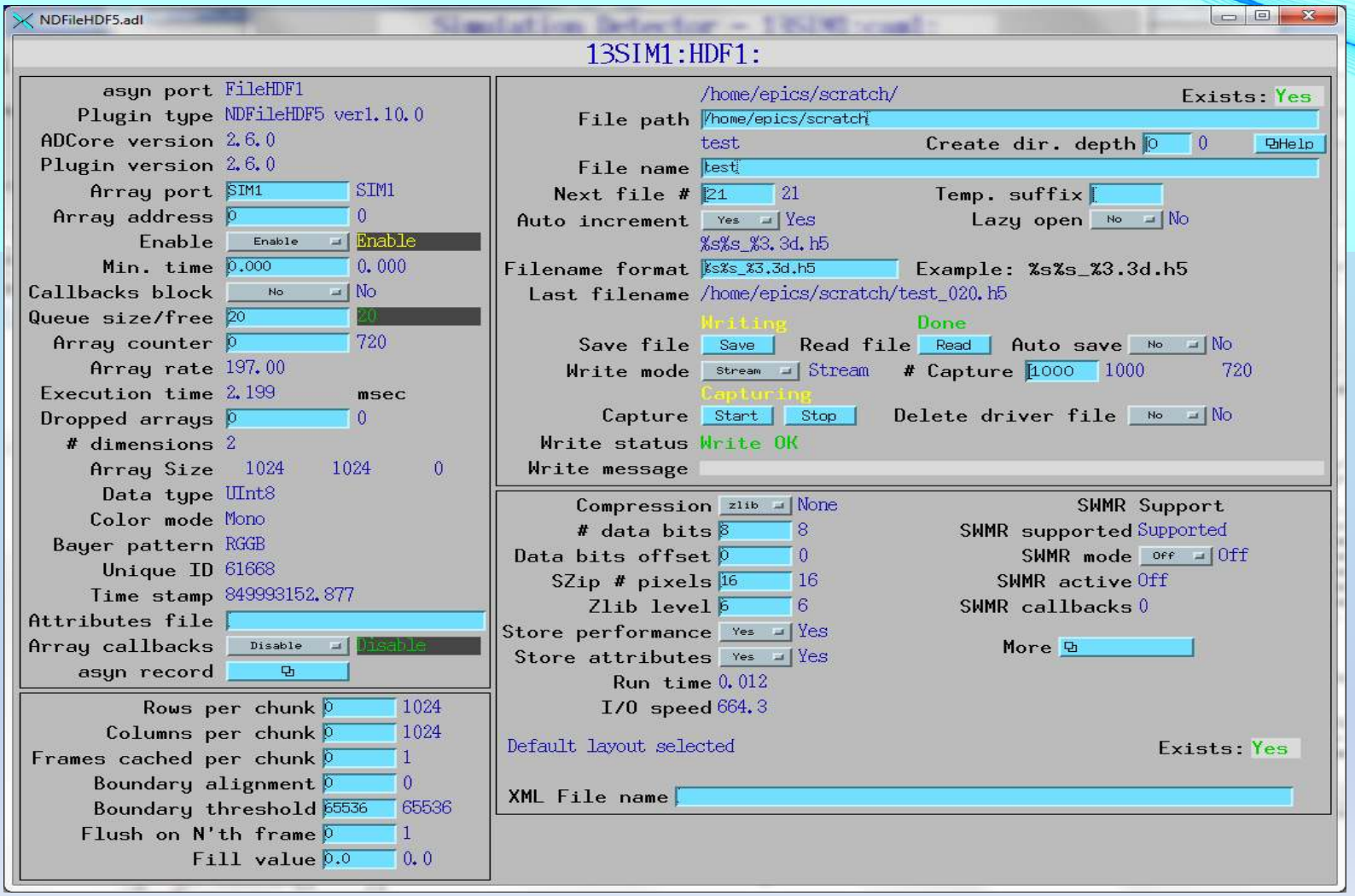
Why this protocol?

- Raw data can be collected at the highest speeds into a single file
- Kept in the /input tree of the filestore this will be archived automatically.
- Processed data requires programs to read the data. Currently these work on TIFF file stacks. It is easy to unload data from the HDF5 file to TIFF stacks.

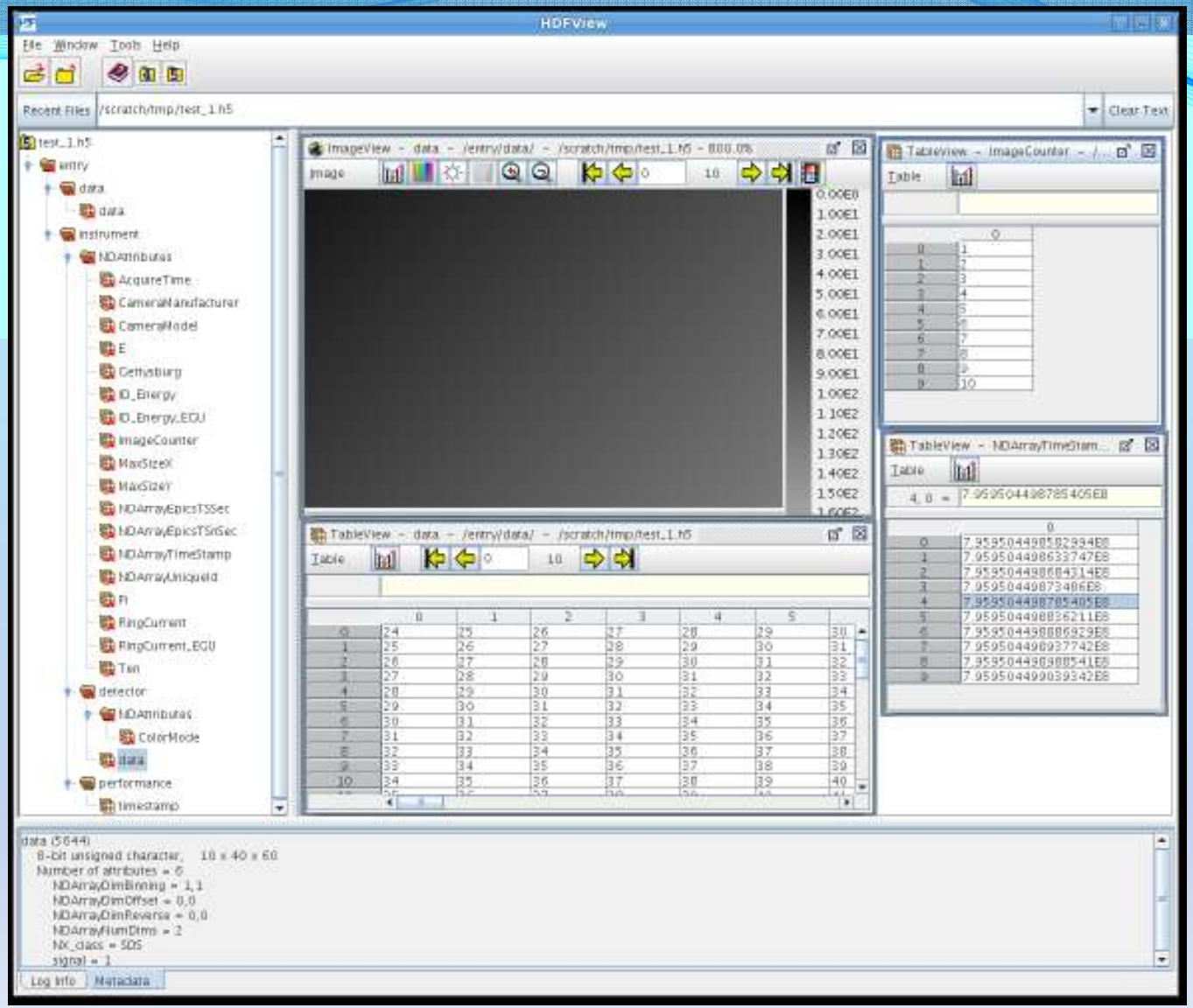
The role of AreaDetector

- All our imagers use an EPICS AreaDetector system to control, read, and store image data
- AreaDetector has a plugin which will take the data and save it to the HDF5 file along with the instrument attributes
- Storing to HDF5 works in either Stream or Capture mode

An example HDF5 set-up



Example HDF5 on IMBL



Take home message:

- If you want to collect a CT set with a short exposure times
- If you are interested in switching to a more manageable data format

- **Ask us about using HDF5**