

Training Videos

HDF5 Introduction



Core Topics

Data Model and Basic Usage, Core Topic #1

What is HDF5?

THE PROBLEM

All these problems in dealing with data are solved by the flexibility of images and data that work any way you want.

We need a way to store data and to access it in a consistent way. It's just as important.

- Time spent writing another CSV parser is time not spent on analysis.
- Scattered data and metadata prevents us from seeing the whole, consistent picture.

Datasets, Core Topic #2

Creating a Dataset

```
CREATE A DATASET
# Use the create dataset
# method on File to create
# an empty dataset of the
# desired datatype and
# datatype (shape)
>>> import h5py
>>> f = h5py File("file5.
>>> f.create_dataset("dataset", dtype="float64", data=(10, 10))
>>> f.close()
>>> CLOSE OBJECTS AND FILE
```

Attributes, Core Topic #3

Groups and Links, Core Topic #4

Creating Groups

```
CREATING GROUPS
# To create a group in the file
>>> grp = myFile.create_group("mygroup")
# Note that methods like create_group
# and create_group are available
# group, not just the File object
>>> subgrp = grp.create_group("subgroup")
# There's an interesting attribute
# dataset and group objects, .name:
>>> subgrp.name
"/mygroup/subgroup"
```

Discovering File Structure, Core Topic #5

What Does "Self-Describing" Mean?

- HDF5 files contain both structural information and data
- Groups and links provide structure
- Attributes provide metadata
- All of this is discoverable; you don't need to consult file format documentation or write a parser

Partial IO, Core Topic #6

Partial I/O: Indexing

WHAT'S GOING ON?

- When we index into a dataset, h5py locates the corresponding element on disk, and loads it into memory.
- HDF5 then performs any type conversion necessary to fit the data into one of the NumPy types.
- A NumPy value is created, and returned to the Python caller.

Compound Datatype, Core Topic #7

Compound Datatype

COMPOUND DATATYPE

- Compound datatypes allow you to compose **aggregate records** from multiple **heterogeneous pieces** and store them in a **single data**.
- Fields of compound datatype may have any HDF5 datatype - including compound datatype.
- Once you've created a compound datatype, it's a C++ class citizen - you can use it in any place where you can specify a datatype.

Name	Age (integer)	Weight (float)
John Smith	27	180.4
Walter Brown	19	220.2
Lisa Miller	32	135.0
Oliver Jones	21	120.6
Christina White	66	180.1
Andrew Wilson	33	121.4
Michael Taylor	35	120.2
Ella Parker	80	155.0
Sarah Lee-Scott	33	189.4
Donna Walker	74	127.1

Advanced Topics:

Dataset Storage Layouts, Advanced Topic #1

HDF5 Storage Layouts

EXAMPLE: CHUNKING WITH h5py

```

# Use the "chunks" keyword to
# create dataset.

# For example, suppose we have
# images which are 640x480,
# application will access it
# 320x240 slices:
>>> dset = h5file.create_data(
    "mydataset", (100, 640, 480),
    chunks=(1, 320, 240) )
    
```

Using Compression and Filters, Advanced Topic #2

HDF5 Filters

FILTER PIPELINE

- More than one filter can be applied to a chunk.
- For example, you could combine a compression filter with a checksum filter to verify data integrity.
- The filters form a "pipeline" which processes data on its way to and from disk.

Using Command Line Tools, Advanced Topic #3

Command-Line Tools

Tool	Function
lsf	List contents of the HDF5 file
lsdump	Examine structure and retrieve data
h5pack	Process file, add or remove filters
h5diff	Compare two HDF5 files
h5import	Import ASCII or binary data into HDF5
h5importsum	Add user block data to an HDF5 file
h5copy	Copy objects to a new file
h5stat	Display object and metadata information
h5perf	Tools to interact with compilers, measure performance, and convert formats