HDF5: An Introduction

Adam Carter
EPCC, The University of Edinburgh

What is HDF5?

Hierarchical Data Format (version 5)

From www.hdfgroup.org:

HDF5 is a unique technology suite that makes possible the management of extremely large and complex data collections.

What is HDF5?

- A versatile **data model** that can represent very complex data objects and a wide variety of metadata.
- A completely **portable file format** with no limit on the number or size of data objects in the collection.
- A **software library that** runs on a range of computational platforms, from laptops to massively parallel systems, and implements a high-level API with C, C++, Fortran 90, and Java interfaces.
- A rich set of integrated **performance** features that allow for access time and storage space optimizations.
- Tools and applications for managing, manipulating, viewing, and analyzing the data in the collection.

Source: www.hdf5group.org (emphasis mine)

Data Model

- In very basic terms, HDF is like a directory and file *hierarchy* in a file
- The data model is based on *groups* and *datasets*
  - can think of groups like directories/folders, datasets like files
  - both can have (user-defined) *attributes*

Portable File Format

- HDF5 files are binary but portable
  - HDF5 model take care of types, endianness etc.

Why HDF5?

- Structure
- Portability
- Performance
- Free & Open Source!

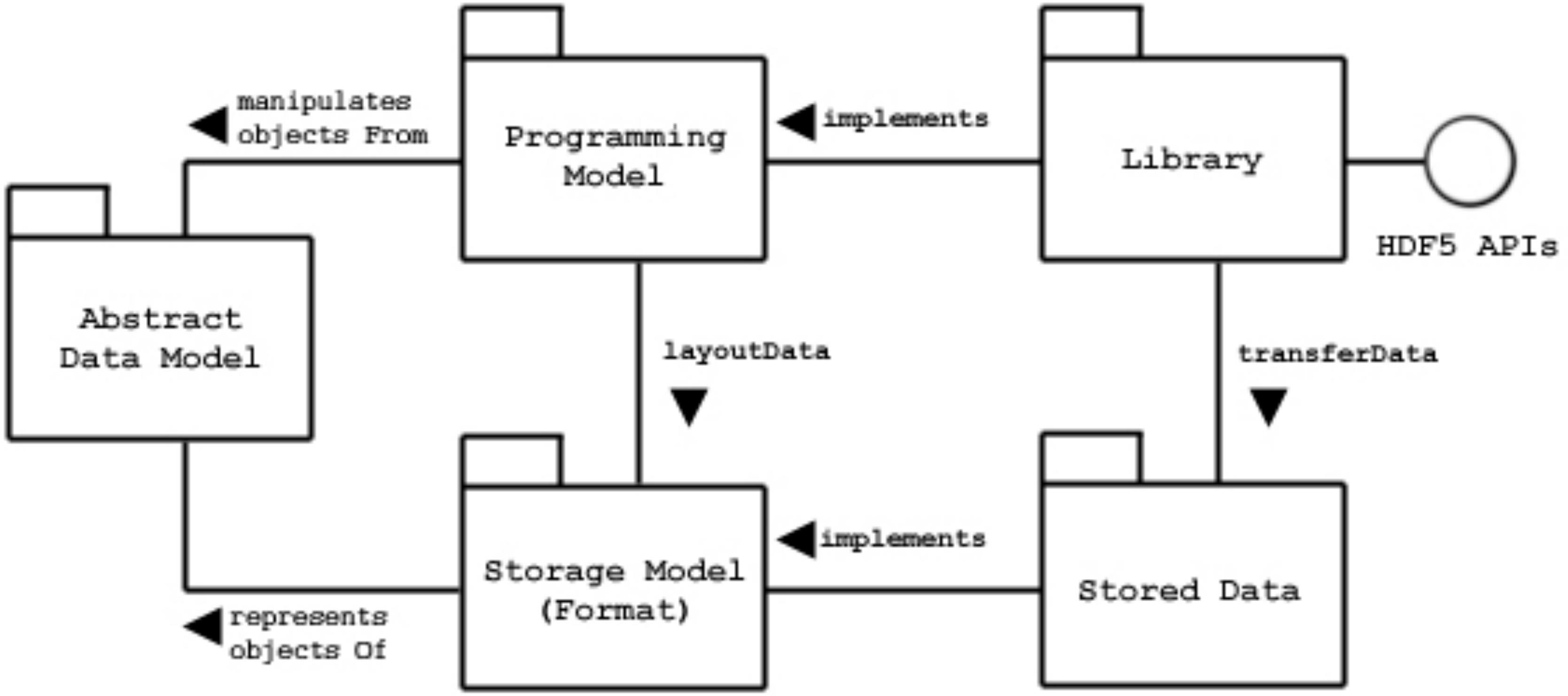
HDF5 files are Self-Describing

Tool Support

Pre-Optimised

Parallel-Ready

HDF5 Data Model and File Structure



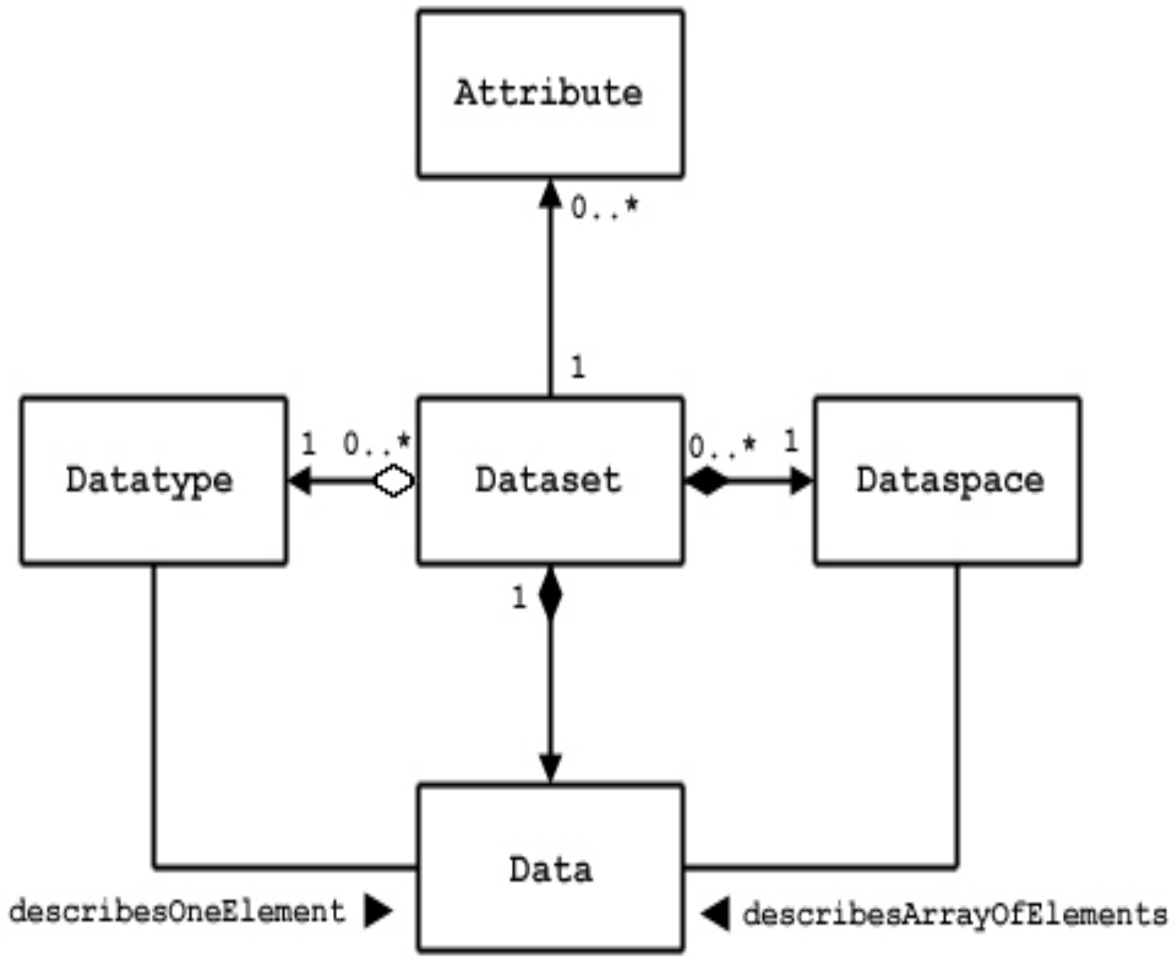
Source: http://www.hdfgroup.org/HDF5/doc/UG/Images/Dmodel_fig1.JPG

HDF5 Groups

- *HDF5 group:* "a grouping structure containing instances of zero or more groups or datasets, together with supporting metadata."
- A group has two parts:
  - A *group header* containing name & attributes
  - A group *symbol table* listing the group's contents
- Like UNIX directories, you can identify an object with a path name:
  - / the root group
  - **/foo** a member of the root group, called foo
  - **/foo/zoo** a member of the foo group, called zoo

HDF5 Datasets

- A dataset has two parts:
  - A *header*
  - A *data array*
- Header provides information on:
  - Name
    - The dataset name. A sequence of alphanumeric characters.
  - Datatypes
    - Atomic, Compound, NATIVE, Named
  - Dataspace
    - Describes the dimensionality (including *unlimited* option)
  - Storage Layout
    - Contiguous, compact, chunked



Source: http://www.hdfgroup.org/

HDF5 Attributes

- *Attributes* are small named datasets that are attached to primary datasets, groups, or named datatypes
- Name, value pairs
  - Value can have multiple entries of the same datatype
- There’s a separate API for attribute read/write
- Excessively large attribute sets will impact performance

The HDF5 API

- **H5F**: **F**ile-level access routines
  - e.g. H5Fopen
- **H5G**: **G**roup functions, for creating and operating on groups of objects
  - e.g. H5Gset
- **H5T:** Data**T**ype functions, for creating and operating on simple and compound datatypes to be used as the elements in data arrays
- **H5S:** Data**S**pace functions, which create and manipulate the dataspace in which the elements of a data array are stored

- **H5D: D**ataset functions, which manipulate the data within datasets and determine how the data is to be stored in the file.
- **H5P**: **P**roperty list functions, for manipulating object creation and access properties.
- **H5A**: **A**ttribute access and manipulating routines.
- **H5Z**: **C**ompression registration routine.
- **H5E**: **E**rror handling routines.
- **H5R**: **R**eference routines.
- **H5I**: **I**dentifier routine.

Include Files (and a note about use on ARCHER)

- In your program you'll need:

```
#include <hdf5.h>
```

- On ARCHER you should

```
module load cray-hdf5-parallel
```

Example: Create and Close a File

A type defined in HDF5. An HDF ID, used to keep track of objects like files

```
hid_t       file;                          /* identifier */
/*
* Create a new file using H5ACC_TRUNC access,
* default file creation properties, and default file
* access properties.
* Then close the file.
*/
file = H5Fcreate(FILE, H5ACC_TRUNC, H5P_DEFAULT, H5P_DEFAULT);
status = H5Fclose(file);
```

Allows an existing file (if present) to be overwritten

Example: Creating a dataset so that data can be written to it

```
hid_t    dataset, datatype, dataspace;   /* declare identifiers */
/* Create dataspace: Describe the size of the array and
 * create the data space for fixed size dataset.
 */
dimsf[0] = NX;
dimsf[1] = NY;
dataspace = H5Screate_simple(RANK, dimsf, NULL);
/* Define datatype for the data in the file.
 * We will store little endian integer numbers.
 */
datatype = H5Tcopy(H5T_NATIVE_INT);
status = H5Tset_order(datatype, H5T_ORDER_LE);
/* Create a new dataset within the file using defined
 * dataspace and datatype and default dataset creation
 * properties.
*/
dataset = H5Dcreate(file, DATASETNAME, datatype, dataspace, H5P_DEFAULT);
```

Example: Write data to a file

```
/*
* Write the data to the dataset using default transfer
* properties.
*/
status = H5Dwrite(dataset, H5T_NATIVE_INT, H5S_ALL,
      H5S_ALL, H5P_DEFAULT, data);
```

Here's the data itself, stored as a standard array of ints in C.

Example: Read data from a file

```
/*
* Write the data to the dataset using default transfer
* properties.
*/
status = H5Dread(dataset, H5T_NATIVE_INT, H5S_ALL,
      H5S_ALL, H5P_DEFAULT, data);
```

- Exactly analogous to write!

Hyperslabs

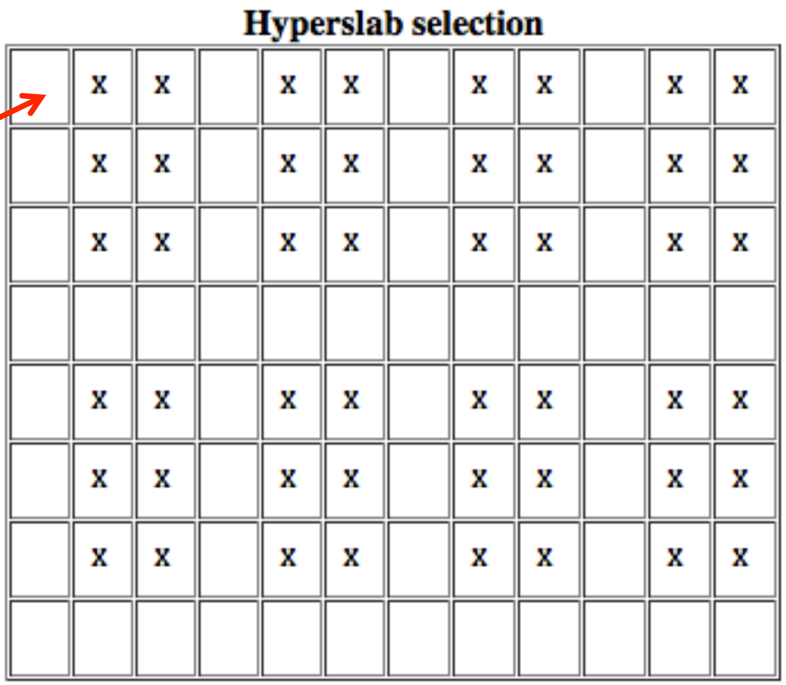
- Hyperslabs are portions of datasets

start = (0,1)

stride = (4,3)

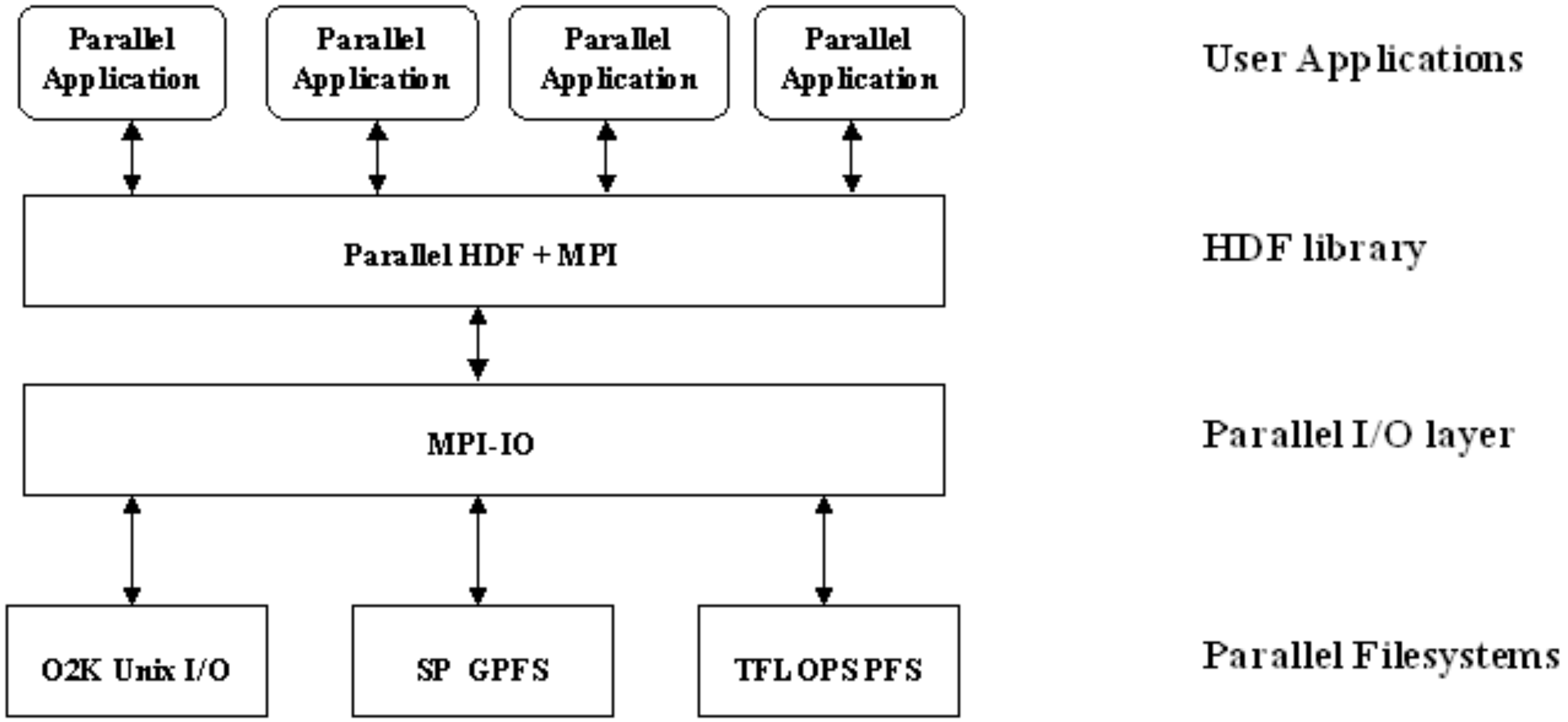
count = (2,4)

block = (3,2)



Parallel HDF5

- Designed to work with MPI and MPI-IO
- Parallel HDF5 files are compatible with serial HDF5 files and sharable between different serial and parallel platforms
- Parallel HDF5 had to be designed to have a single file image to all processes, rather than having one file per process. Having one file per process can cause expensive post processing, and the files are not usable by different processes.
- A standard parallel I/O interface had to be portable to different platforms.



Source: http://www.hdfgroup.org/HDF5/Tutor/poverview.html

Further Reading

- *Introduction to HDF5*
  - www.hdfgroup.org/HDF5/doc/H5.intro.html
- *HDF5 User Guide*
  - www.hdfgroup.org/HDF5/doc/UG/index.html
- *HDF5 Reference Manual*
  - www.hdfgroup.org/HDF5/doc/RM/RM_H5Front.html
- *Introduction to Scientific I/O*
  - http://www.nersc.gov/users/training/online-tutorials/introduction-to-scientific-i-o/

Acknowledgements & Re Use

- The presentation was created by Adam Carter, EPCC, The University of Edinburgh

- Portions of this presentation are based on the HDF5 documentation