What NetCDF users should know about HDF5?

Elena Pourmal

The HDF Group

July 20, 2007

Outline

- The HDF Group and HDF software
- HDF5 Data Model
- Using HDF5 tools to work with NetCDF-4 programs files
- Performance issues
  - ✓ Chunking
  - ✓ Variable-length datatypes
  - ✓ Parallel performance
- Crash proofing in HDF5

The HDF Group

- Non-for-profit company with a mission to sustain and develop HDF technology affiliated with University of Illinois
- Spun-off NCSA University of Illinois in July 2006
- Located at the U of I Campus South Research Park
- 17 team members, 5 graduate and undergraduate students
- Owns IP for HDF fie format and software
- Funded by NASA, DOE, others

HDF5 file format and I/O library

- General
  - ✓ simple data model
- Flexible
  - ✓ store data of diverse origins, sizes, types
  - ✓ supports complex data structures
- Portable
  - ✓ available for many operating systems and machines
- Scalable
  - ✓ works in high end computing environments
  - ✓ accommodates date of any size or multiplicity
- Efficient
  - ✓ fast access, including parallel i/o
  - ✓ stores big data efficiently

HDF5 file format and I/O library

- File format
  - ✓ Complex
    - Objects headers
    - Raw data
    - B-trees
    - Local and Global heaps
    - etc
- C Library
  - ✓ 500+ APIs
  - ✓ C++, Fortran90 and Java wrappers
  - ✓ High-level APIs (images, tables, packets)

Apps: simulation, visualization, remote sensing…

Examples: Thermonuclear simulations
Product modeling
Data mining tools
Visualization tools
Climate models

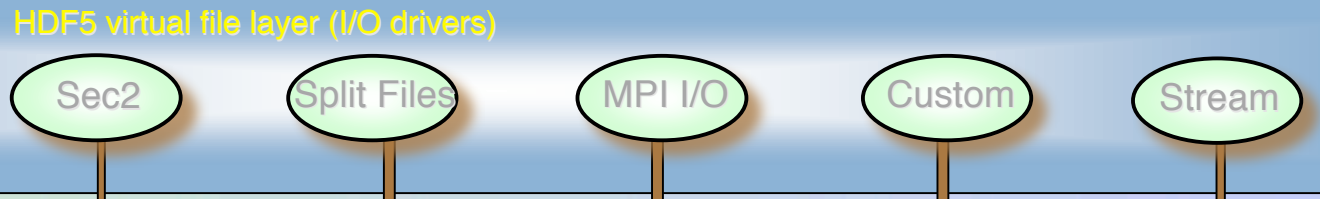
Common application-specific data models

appl-specific APIs

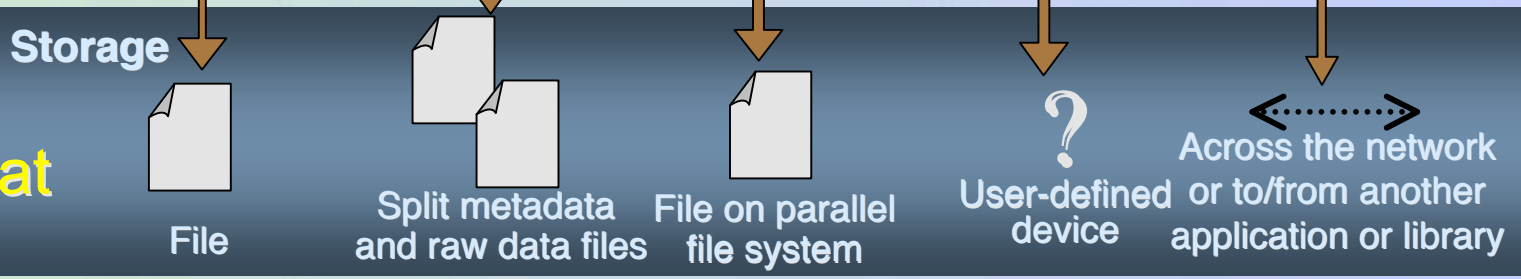
NetCDF-4	SAF	hdf5mesh	IDL	HDF-EOS
UnidataLANL	LLNL, SNL	Grids	COTS	NASA

HDF5 data model & API

HDF5 serial & parallel I/O



HDF5 format



HDF5 file format and I/O library

For NetCDF-4 users HDF5 complexity is hidden behind NetCDF-4 APIs

HDF5 Tools

- Command line utilities
  http://www.hdfgroup.org/hdf5tools.html
  - Readers
    - ✓ h5dump
    - ✓ h5ls
  - Writers
    - ✓ h5repack
    - ✓ h5copy
    - ✓ h5import
  - Miscellaneous
    - ✓ h5diff, h5repart, h5mkgrp, h5stat, h5debug, h5jam/h5unjam
  - Converters
    - ✓ h52gif, gif2h5, h4toh5, h5toh4
- HDFView (Java browser and editor)

Other HDF5 Tools

- ✓ HDF Explorer

  Windows only, works with NetCDF-4 files

  http://www.space-research.org/
- ✓ PyTables
- ✓ IDL
- ✓ Matlab
- ✓ Labview
- ✓ Mathematica
- ✓ See

  http://www.hdfgroup.org/tools5app.html

HDF Information

- HDF Information Center

  http://hdfgroup.org
- HDF Help email address

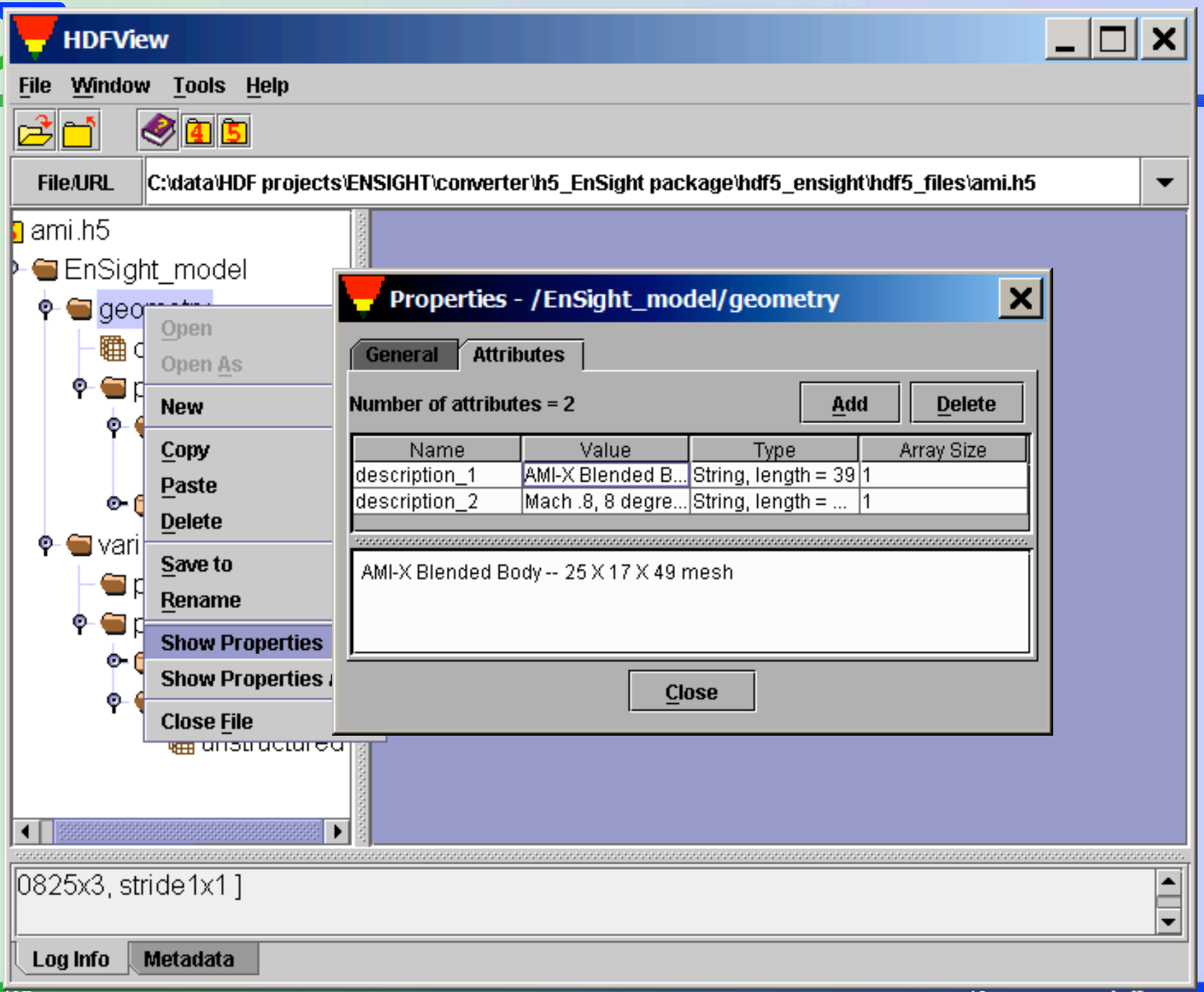
  help@hdfgroup.org
- HDF users mailing lists

  news@hdfgroup.org

  hdf-forum@hdfgroup.org

NetCDF and HDF5 terminology

NetCDF	HDF5
Dataset	HDF5 file
Dimensions	Dataspace
Attribute	Attribute
Variable	Dataset
Coordinate variable	Dimension scale


HDFView
File Window Tools Help
File/URL
C:\data\HDF projects\ENSIGHT\converter\h5_EnSight package\hdf5_ensight\hdf5_files\ami.h5
ami.h5
EnSight_model
Open
Open As
New
Copy
Paste
Delete
Save to
Rename
Show Properties
Close File
Properties - /EnSight_model/geometry
General
Attributes
Number of attributes = 2
Add
Delete
Name
Value
Type
Array Size
description_1
AMI-X Blended B...
String, length = 39
1
description_2
Mach .8, 8 degre...
String, length = ...
1
AMI-X Blended Body -- 25 X 17 X 49 mesh
Close
0825x3, stride1x1]
Log Info
Metadata

HDF5 Data Model

HDF5 data model

- HDF5 file – container for scientific data
- Primary Objects
  - Groups
  - Datasets
- Additional ways to organize data
  - Attributes
  - Sharable objects
  - Storage and access properties

NetCDF-4 builds from these parts.

HDF5 Dataset

Metadata

Dataspace

Rank

3

Dimensions

Dim_1 = 4

Dim_2 = 5

Dim_3 = 7

Datatype

IEEE 32-bit float

Storage info

chunked

compressed

checksum

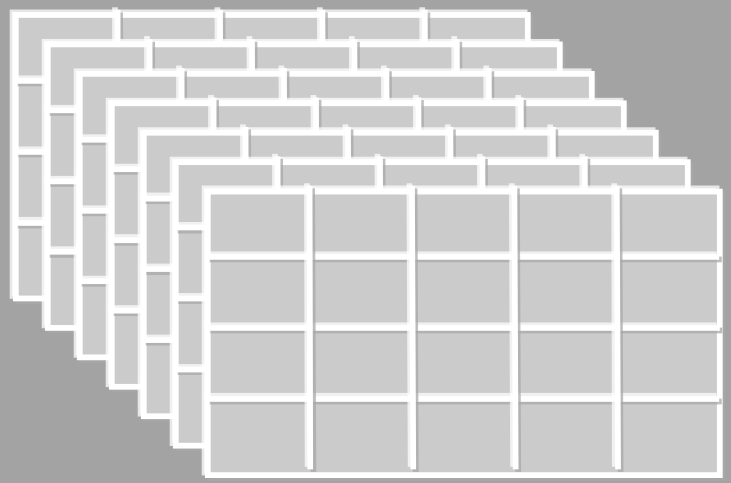
Attributes

time = 32.4

pressure = 987

temp = 56

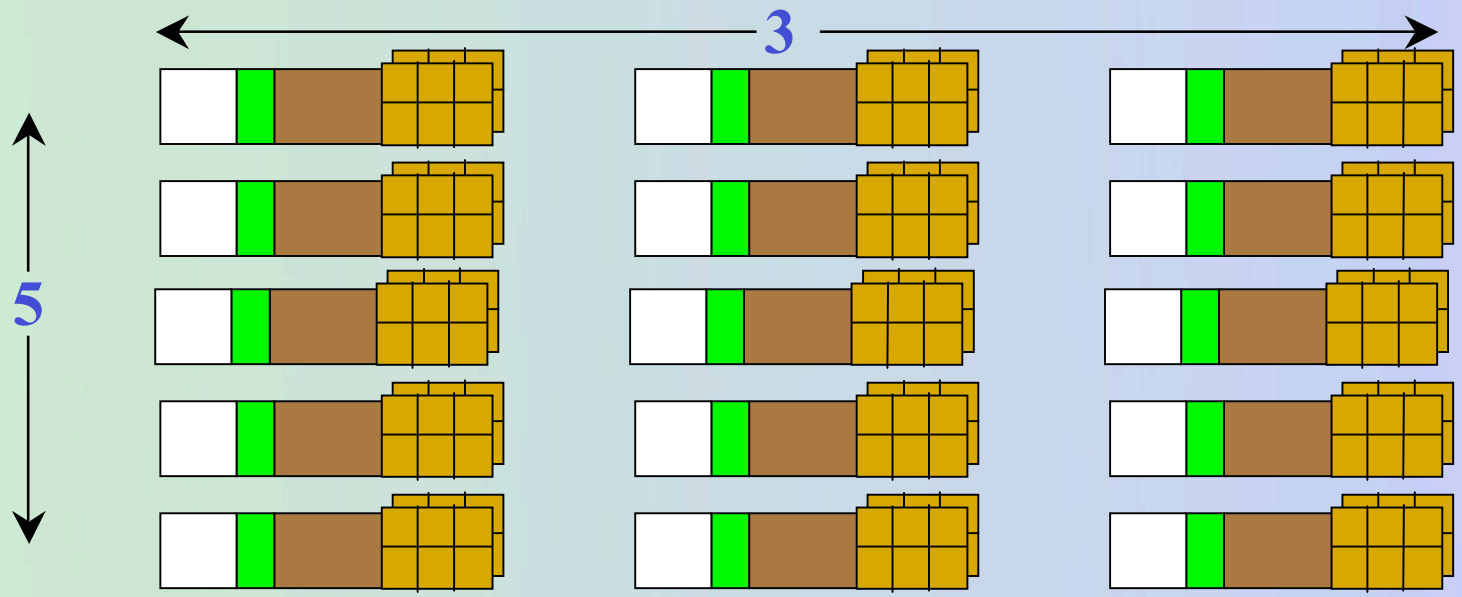
Data



Datatypes

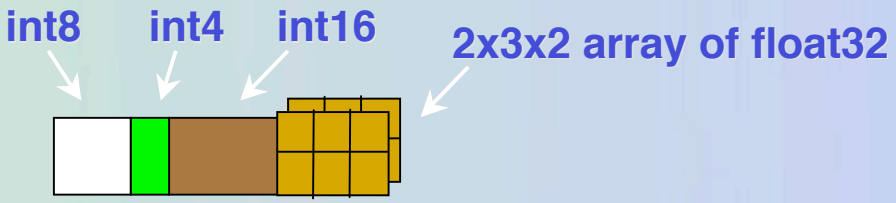
- HDF5 atomic types
  - ✓ normal integer & float
  - ✓ user-definable (e.g. 13-bit integer)
  - ✓ variable length types (e.g. strings, ragged arrays)
  - ✓ pointers - references to objects/dataset regions
  - ✓ enumeration - names mapped to integers
  - ✓ array
  - ✓ opaque
- HDF5 compound types
  - ✓ Comparable to C structs
  - ✓ Members can be atomic or compound types
  - ✓ No restriction on comlexity

HDF5 dataset: array of records



Dimensionality: 5 x 3

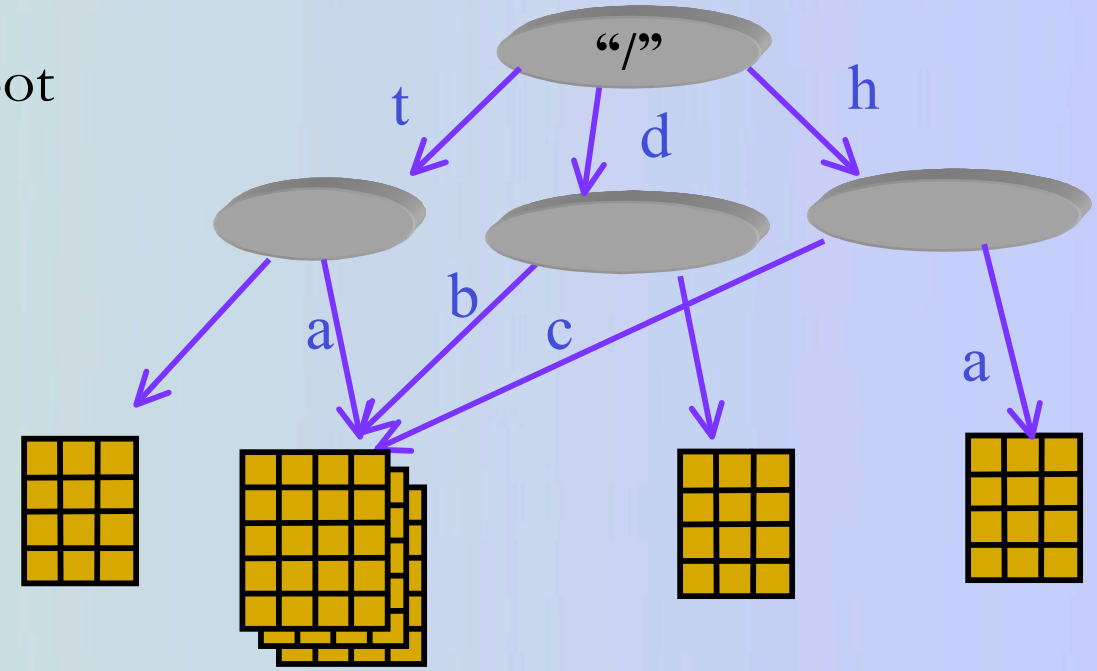
Datatype:



Record

Groups

- A mechanism for collections of related objects
- Every file starts with a root group
- Similar to UNIX directories
- Can have attributes
- Objects are identified by a path e.g. /d/b, /t/a



Attributes

- Attribute – data of the form "name = value", attached to an object (group, dataset, named datatype)
- Operations scaled down versions of dataset operations
  - ✓Not extendible
  - ✓No compression
  - ✓No partial I/O
- Optional
- Can be overwritten, deleted, added during the "life" of a dataset
- Size under 64K in releases before HDF5 1.8.0

Using HDF5 tools with NetCDF-4 programs and files

Example

- Create netCDF-4 file
- /Users/epourmal/Working/_NetCDF-4
  - s.c creates simple_xy.nc (NetCDF3 file)
  - sh5.c creates simple_xy_h5.nc (NetCDF4 file)
  - Use h5cc script to compile both examples
  - See contents simple_xy_h5.nc with ncdump and h5dump
    - Useful flags
      - ✓ -h to print help menu
      - ✓ -b to export data to binary file
      - ✓ -H to display metadata information only
- HDF Explorer

NetCDF view: ncdump output

```
% ncdump -h simple_xy_h5.nc
netcdf simple_xy_h5 {
dimensions:
        x = 6 ;
        y = 12 ;
variables:
         int data(x, y) ;
data:
}
```

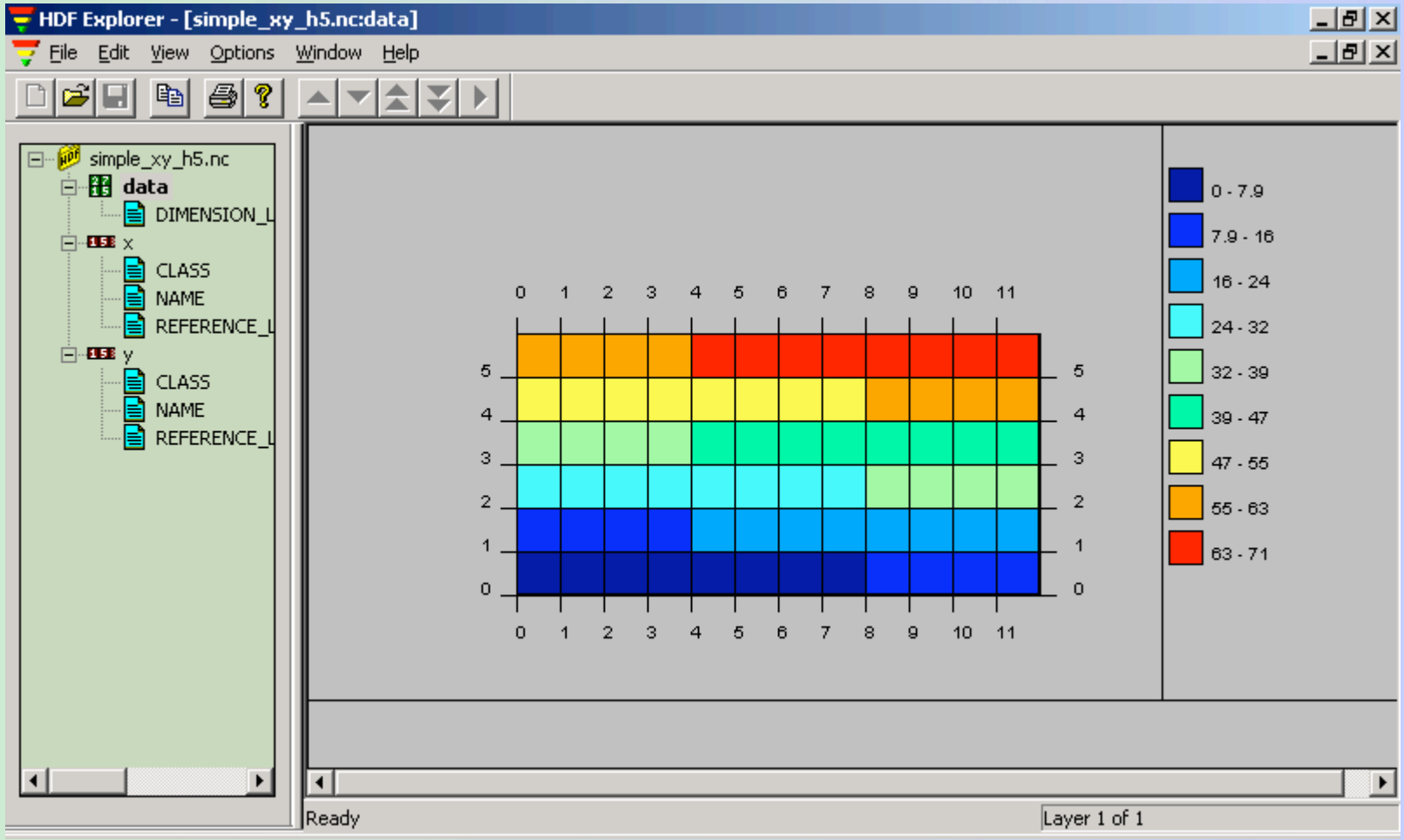
```
% h5dump -H simple_xy.nc
h5dump error: unable to open file "simple_xy.nc"
```

✓ This is NetCDF3 file, h5dump will not work

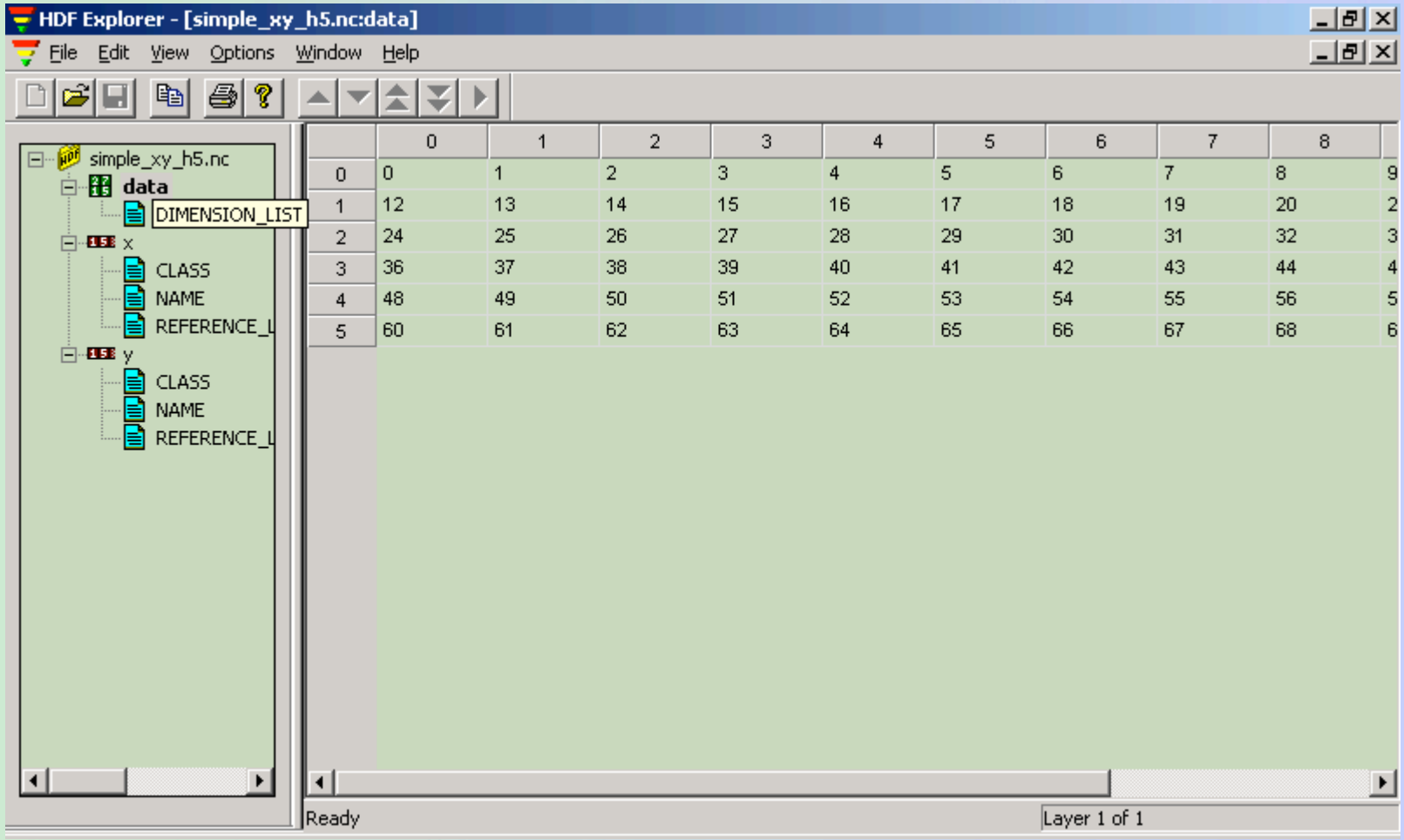
HDF5 view: h5dump output

```
% h5dump -H simple_xy_h5.nc
HDF5 "simple_xy_h5.nc" {
GROUP "/" {
  DATASET "data" {
    DATATYPE  H5T_STD_I32LE
    DATASPACE  SIMPLE { ( 6, 12 ) / ( 6, 12 ) }
    ATTRIBUTE "DIMENSION_LIST" {
      DATATYPE  H5T_VLEN { H5T_REFERENCE}
      DATASPACE  SIMPLE { ( 2 ) / ( 2 ) }
    }
  }
  DATASET "x" {
    DATATYPE  H5T_IEEE_F32BE
    DATASPACE  SIMPLE { ( 6 ) / ( 6 ) }
    .......
 }
```

HDF Explorer



HDF Explorer

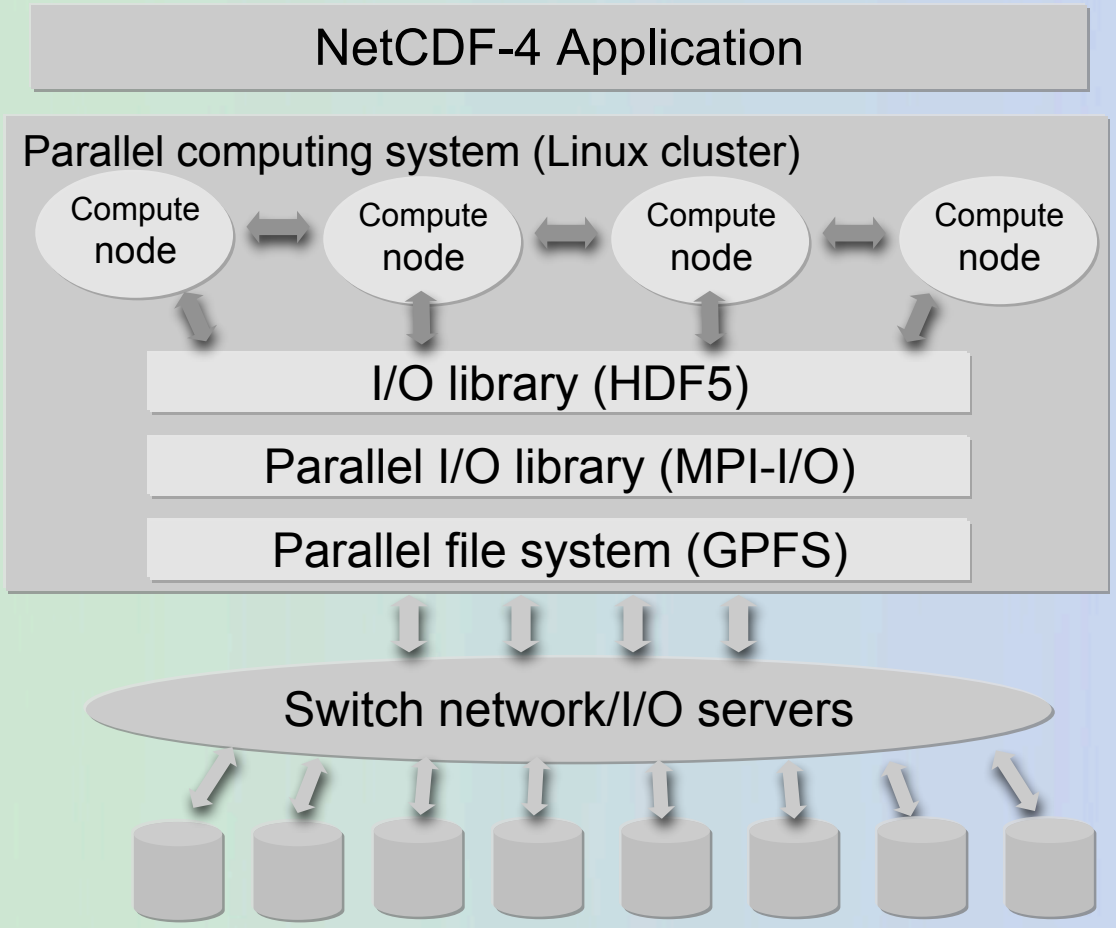


Performance issues

Performance issues

- Choose appropriate HDF5 library features to organize and access data in HDF5 files
- Three examples:
  - Collective vs. Independent access in parallel HDF5 library
  - Chunking
  - Variable length data

Layers – parallel example



Disk architecture & layout of data on disk

I/O flows through many layers from application to disk.

h5perf

- An I/O performance measurement tool
- Test 3 File I/O API
  - Posix I/O (open/write/read/close…)
  - MPIO (MPI_File_{open,write,read.close})
  - PHDF5
    - H5Pset_fapl_mpio (using MPI-IO)
    - H5Pset_fapl_mpiposix (using Posix I/O)

H5perf: Some features

- Check (-c) verify data correctness
- Added 2-D chunk patterns in v1.8

My PHDF5 Application I/O "inhales"

- If my application I/O performance is bad, what can I do?
  - Use larger I/O data sizes
  - Independent vs Collective I/O
  - Specific I/O system hints
  - Parallel File System limits

Independent Vs Collective Access

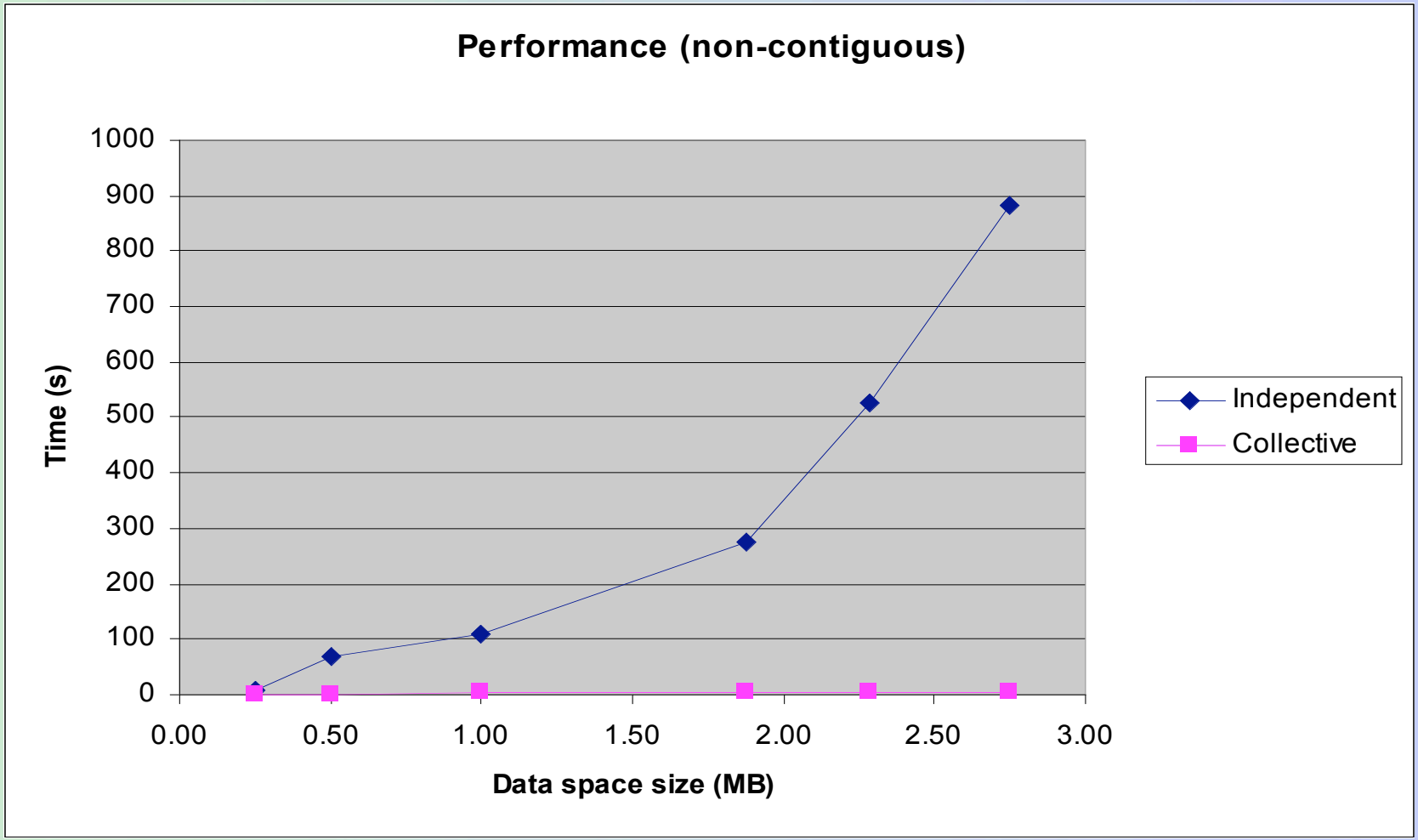
- User reported Independent data transfer was much slower than the Collective mode
- Data array was tall and thin: 230,000 rows by 6 columns

Independent vs. Collective write

(6 processes, IBM p-690, AIX, GPFS)

# of Rows	Data Size (MB)	Independent (Sec.)	Collective (Sec.)
16384	0.25	8.26	1.72
32768	0.50	65.12	1.80
65536	1.00	108.20	2.68
122918	1.88	276.57	3.11
150000	2.29	528.15	3.63
180300	2.75	881.39	4.12

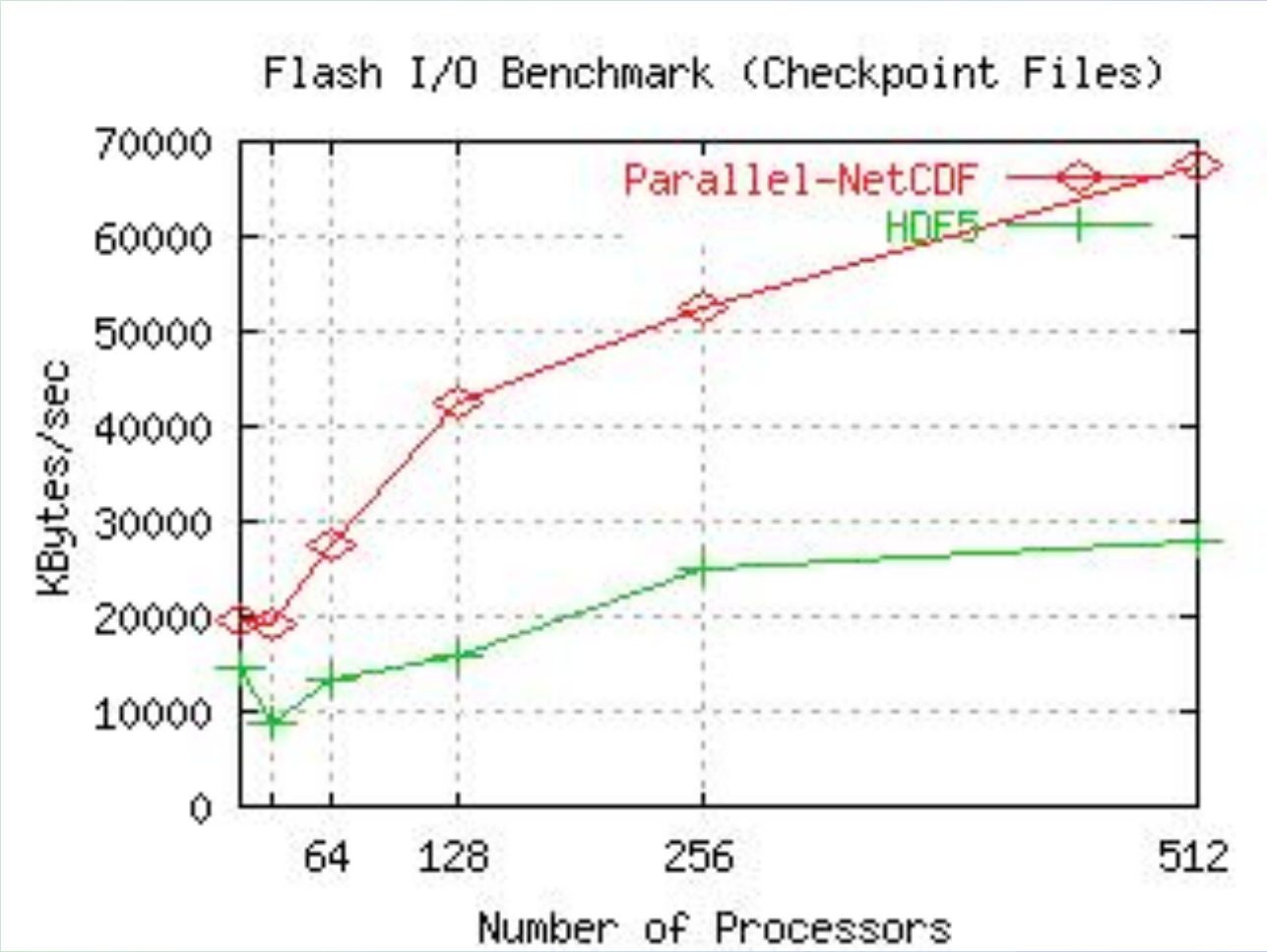
Independent vs Collective write
(6 processes, IBM p-690, AIX, GPFS)



Some performance results

1. A parallel version of NetCDF-3 from ANL/Northwestern University/University of Chicago (PnetCDF)
2. HDF5 parallel library 1.6.5
3. NetCDF-4 beta1
4. For more details see http://www.hdfgroup.uiuc.edu/papers/papers/ParallelPerformance.pdf

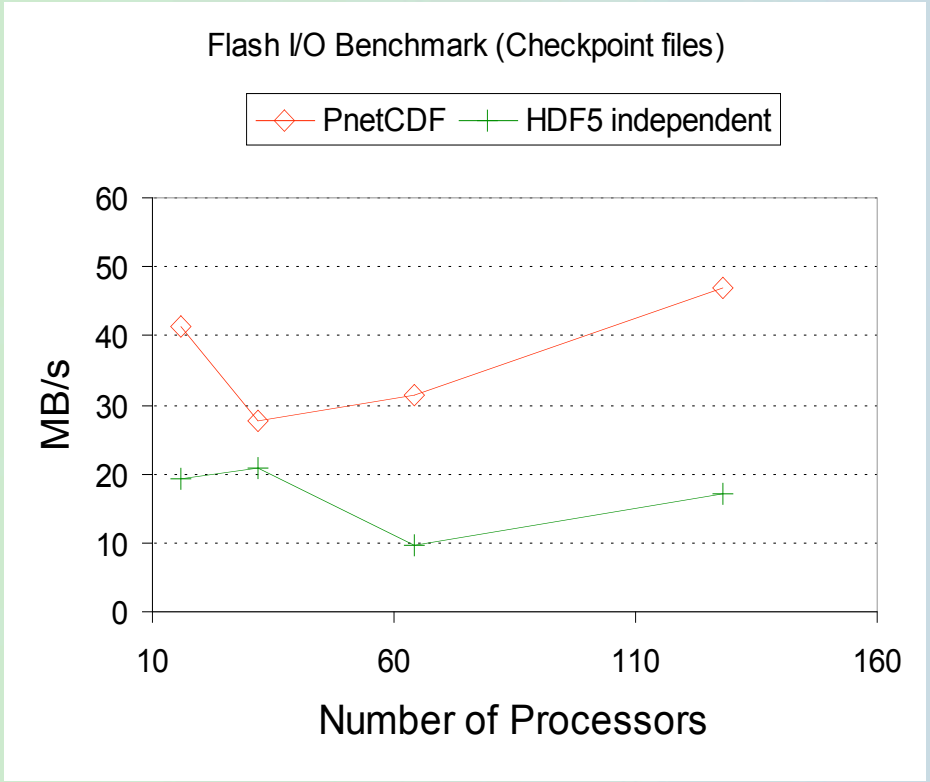
HDF5 and PnetCDF Performance Comparison



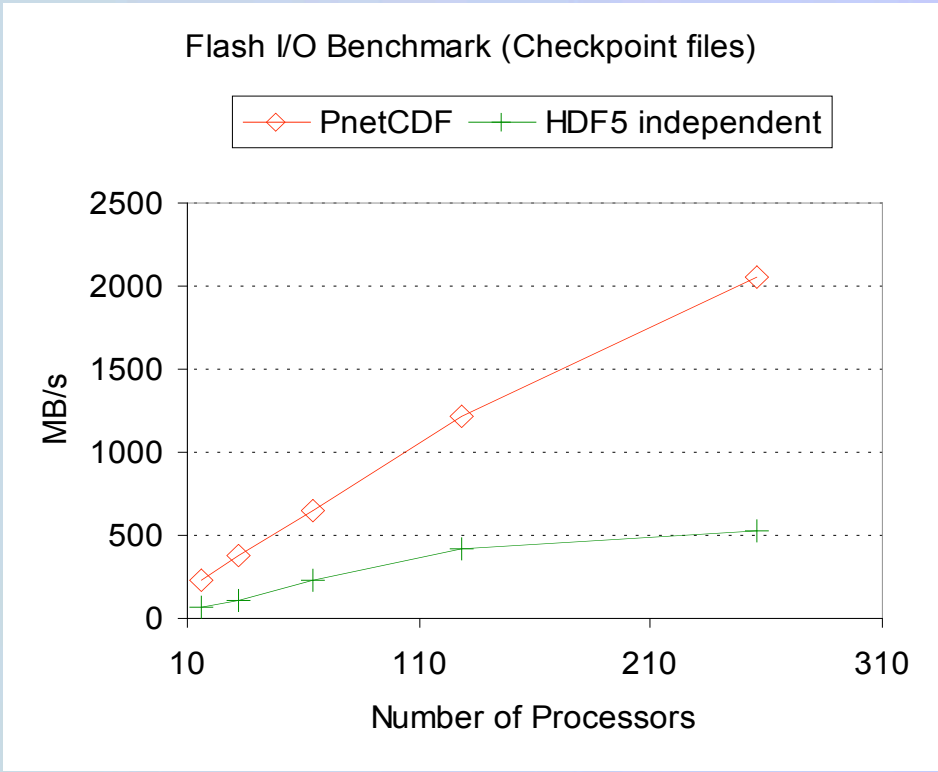
Flash I/O Website http://flash.uchicago.edu/~zingale/flash_benchmark_io/

Robb Ross, etc."Parallel NetCDF: A Scientific High-Performance I/O Interface

HDF5 and PnetCDF performance comparison

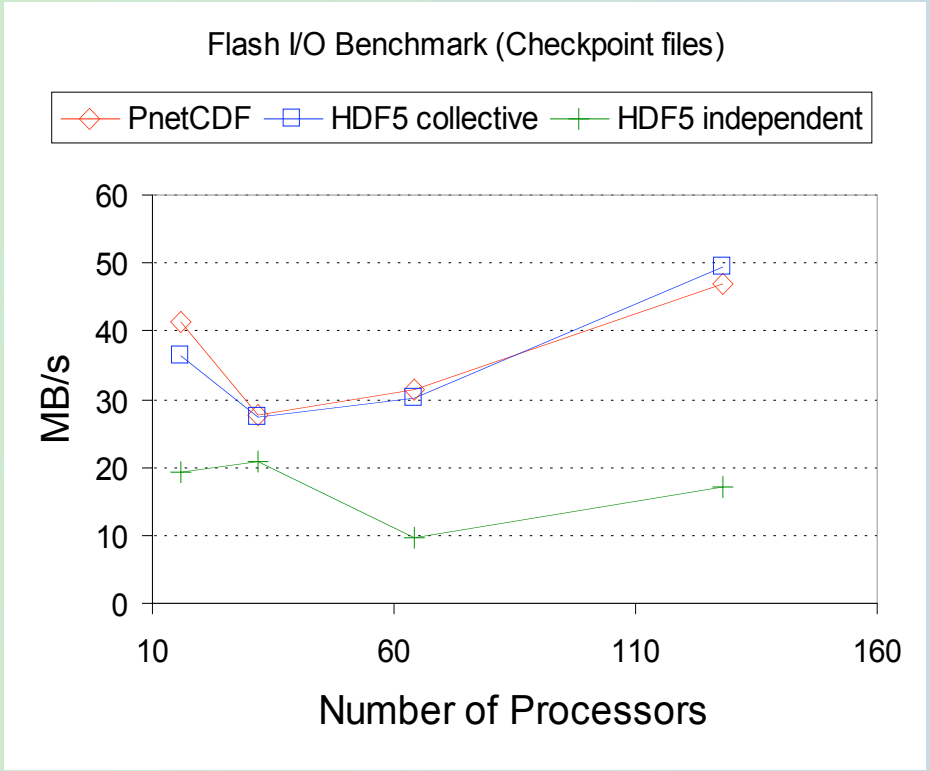


Bluesky: Power 4

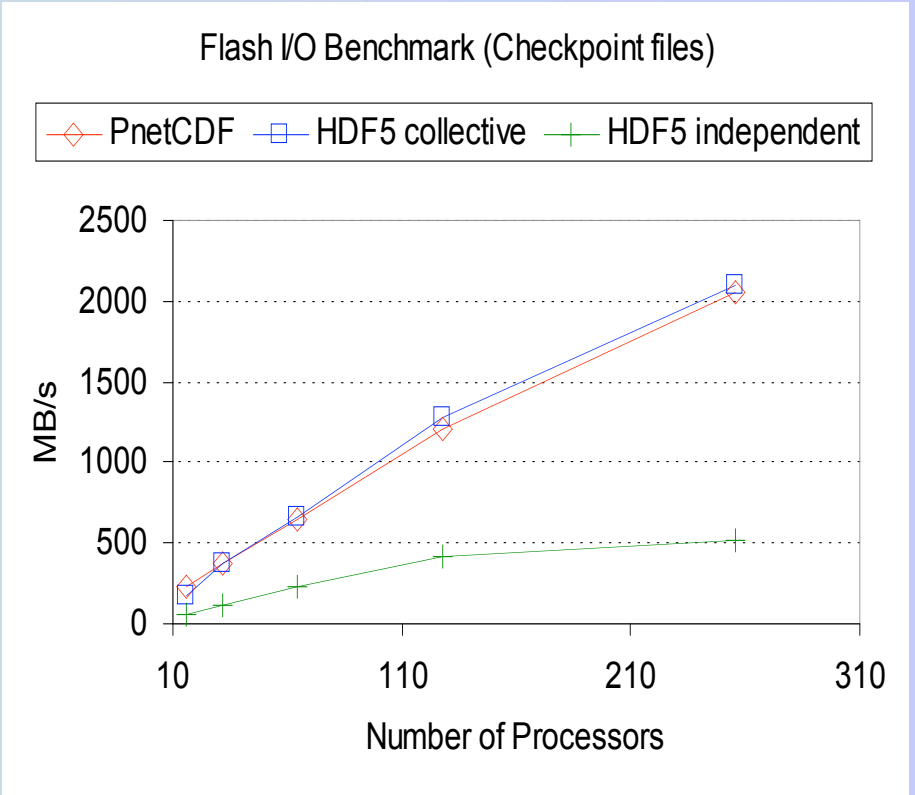


uP: Power 5

HDF5 and PnetCDF performance comparison

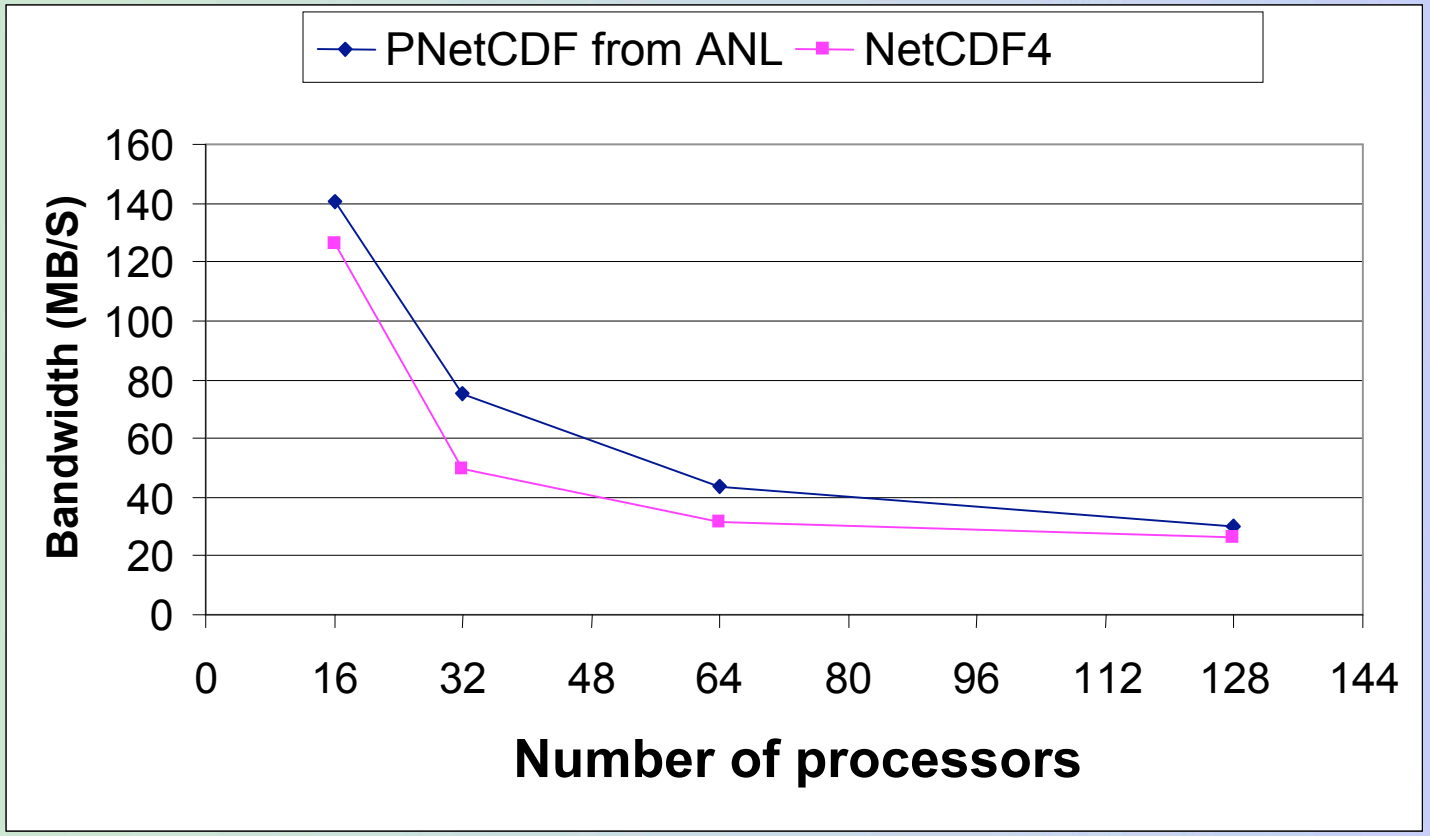


Bluesky: Power 4



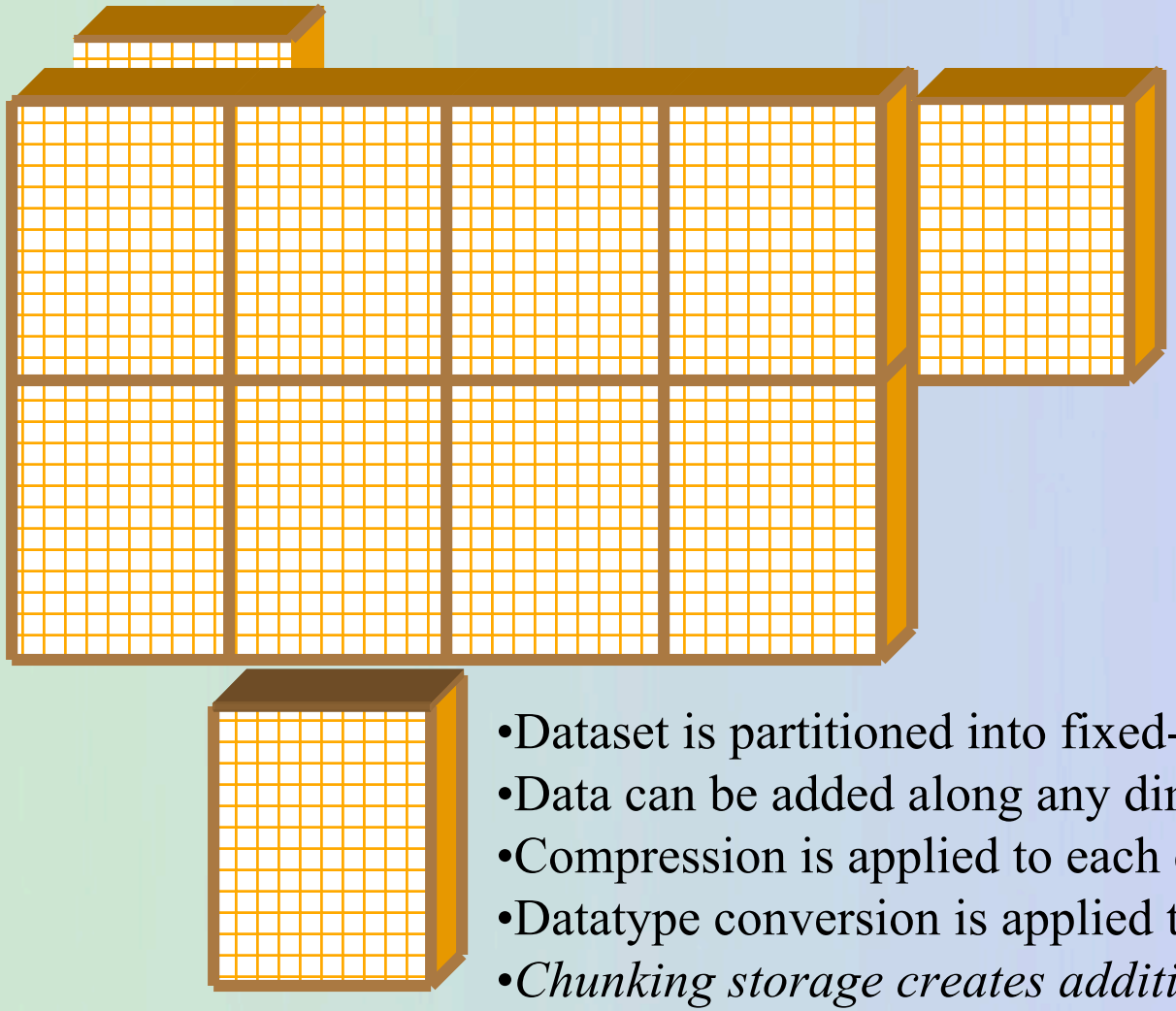
uP: Power 5

Parallel NetCDF-4 and PnetCDF



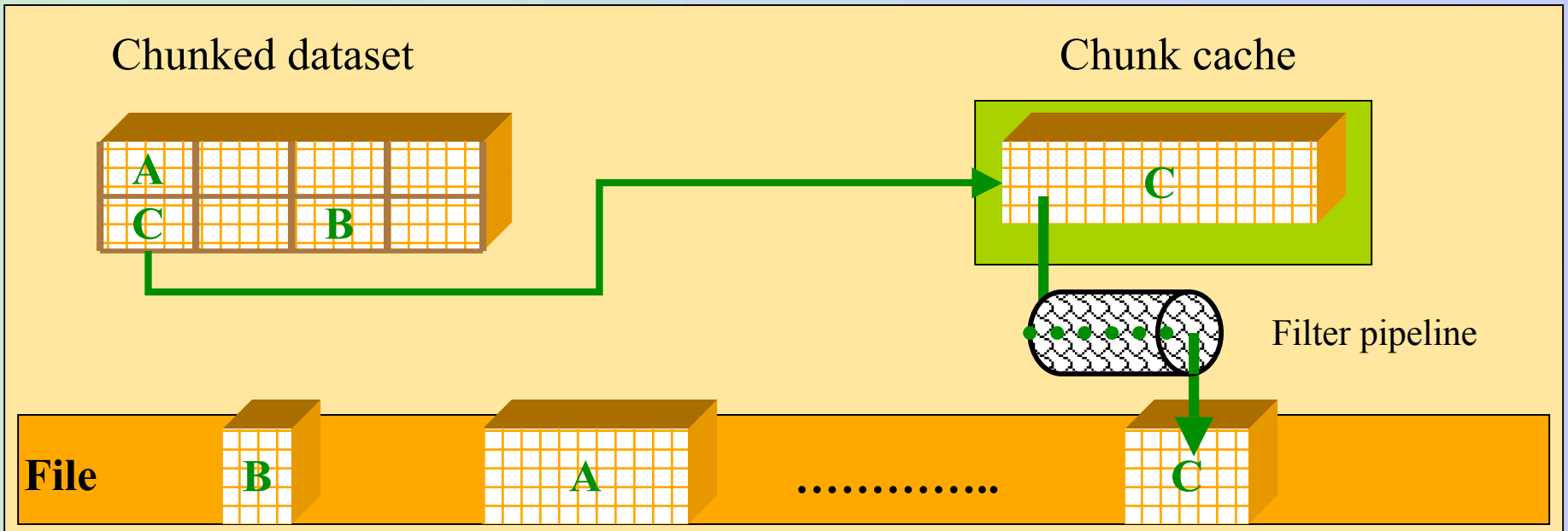
- Fixed problem size = 995 MB
- Performance of PnetCDF4 is close to PnetCDF

HDF5 chunked dataset



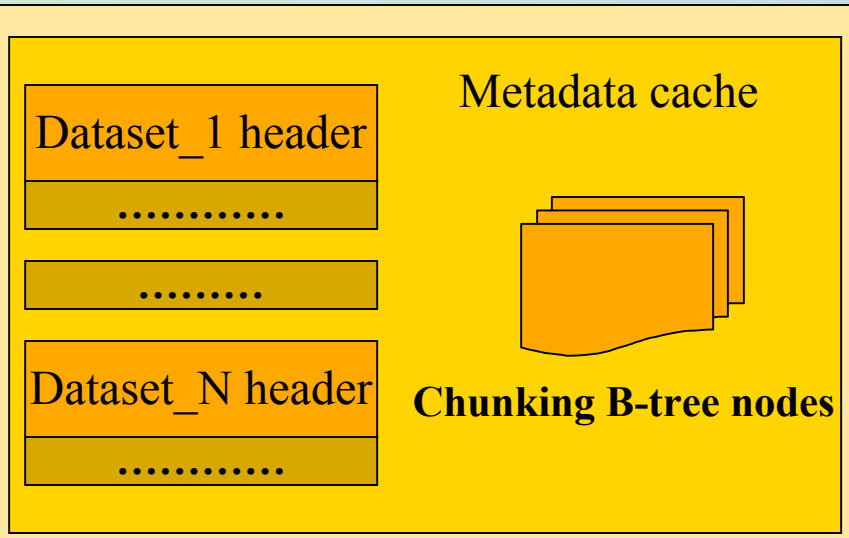
- Dataset is partitioned into fixed-size chunks
- Data can be added along any dimension
- Compression is applied to each chunk
- Datatype conversion is applied to each chunk
- *Chunking storage creates additional overhead in a file*
- *Do not use small chunks*

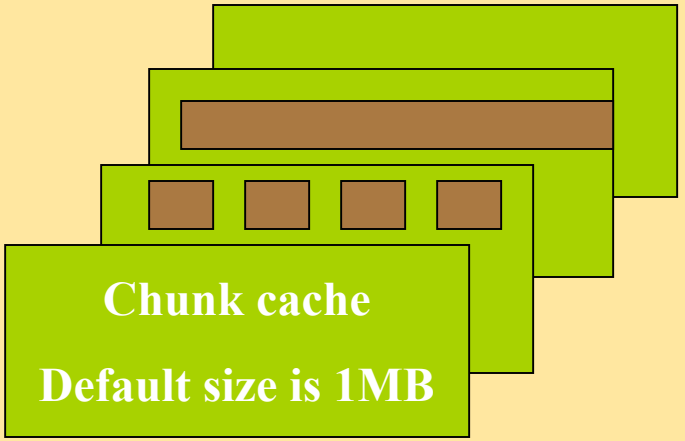
Writing chunked dataset

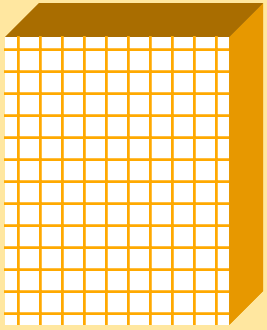


- Each chunk is written as a contiguous blob
- Chunks may be scattered all over the file
- Compression is performed when chunk is evicted from the chunk cache
- Other filters when data goes through filter pipeline (e.g. encryption)

Writing chunked datasets



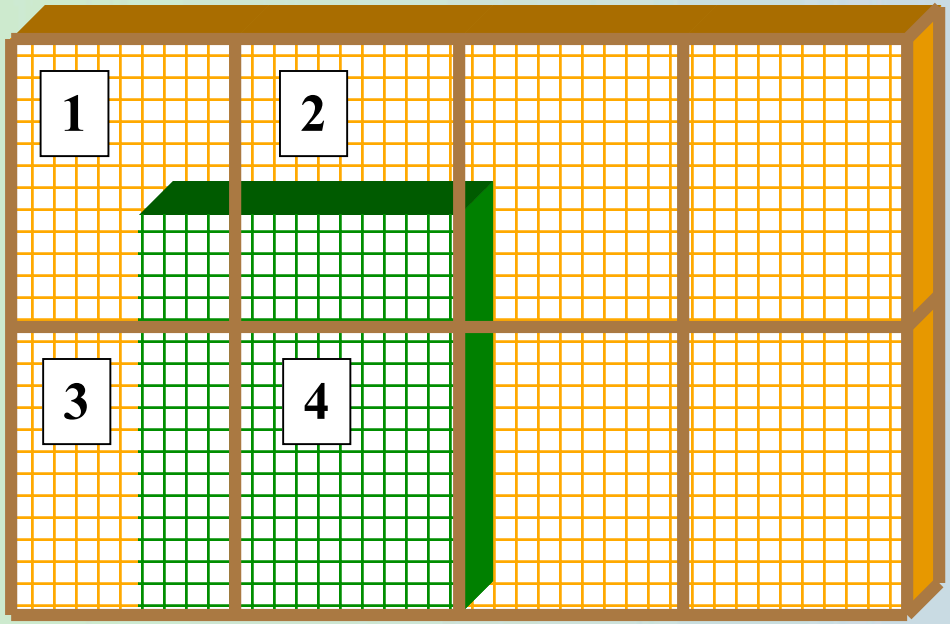




- Size of chunk cache is set for file
- Each chunked dataset has its own chunk cache
- Chunk may be too big to fit into cache
- Memory may grow if application keeps opening datasets

Application memory

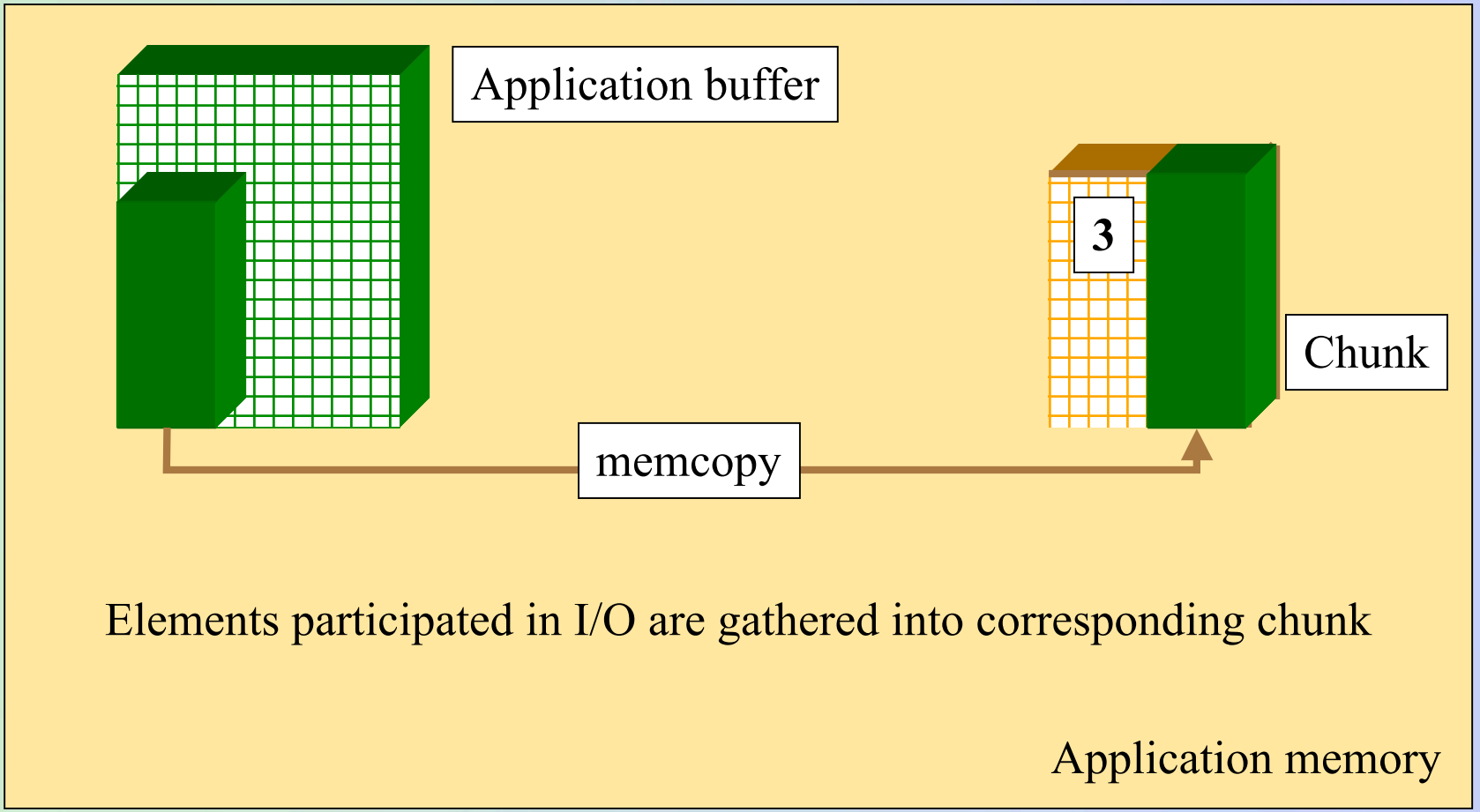
Partial I/O for chunked dataset



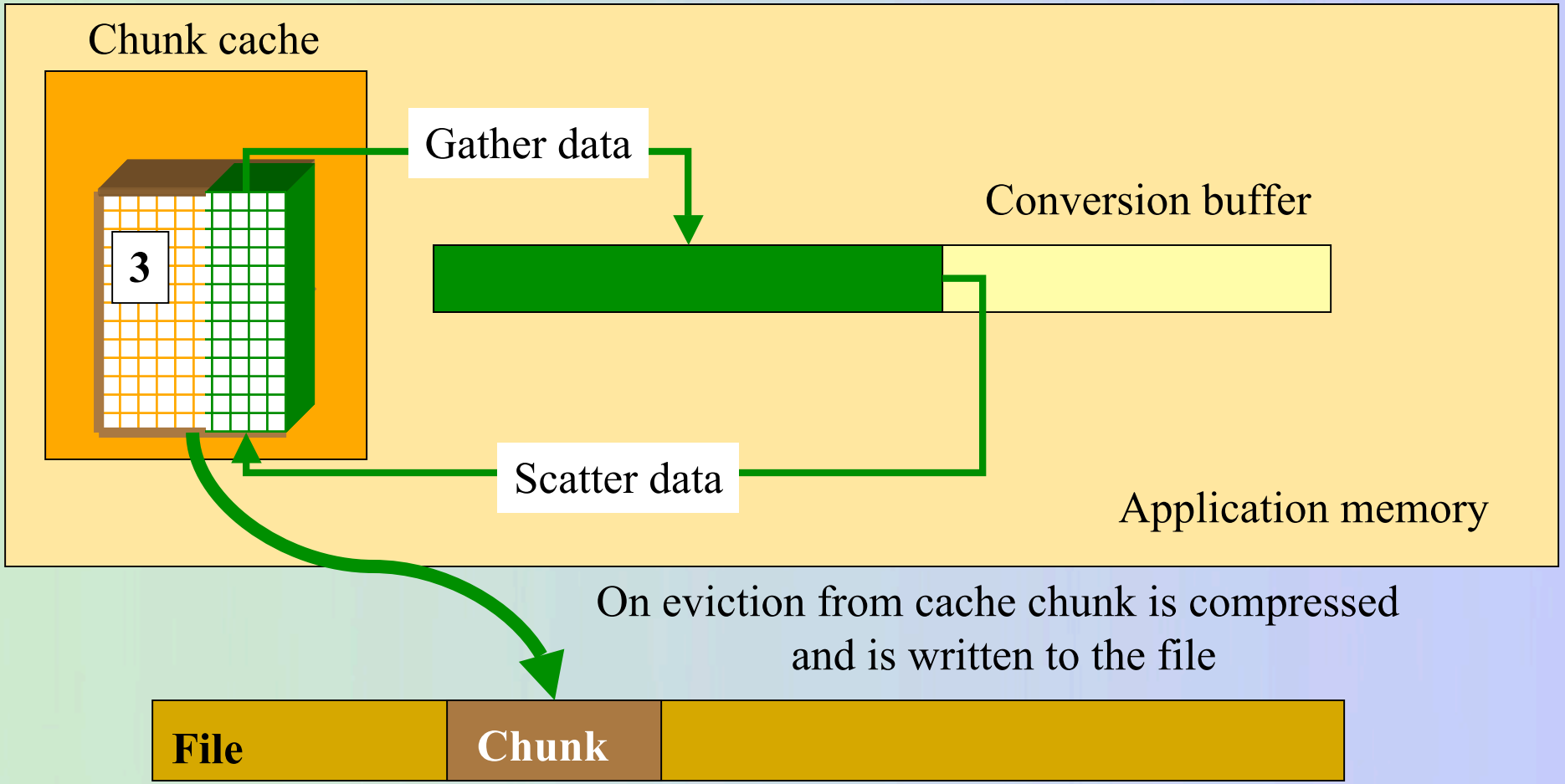
- Build list of chunks and loop through the list
- For each chunk:
- Bring chunk into memory
- Map selection in memory to selection in file
- Gather elements into conversion buffer and perform conversion
- Scatter elements back to the chunk
- Apply filters (compression) when chunk is flushed from chunk cache

For each element 3 `memcopy` performed

Partial I/O for chunked dataset

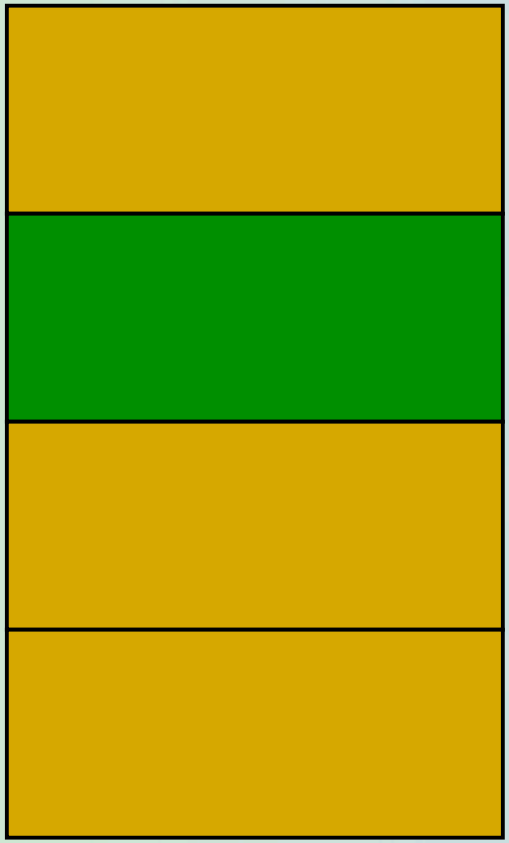


Partial I/O for chunked dataset



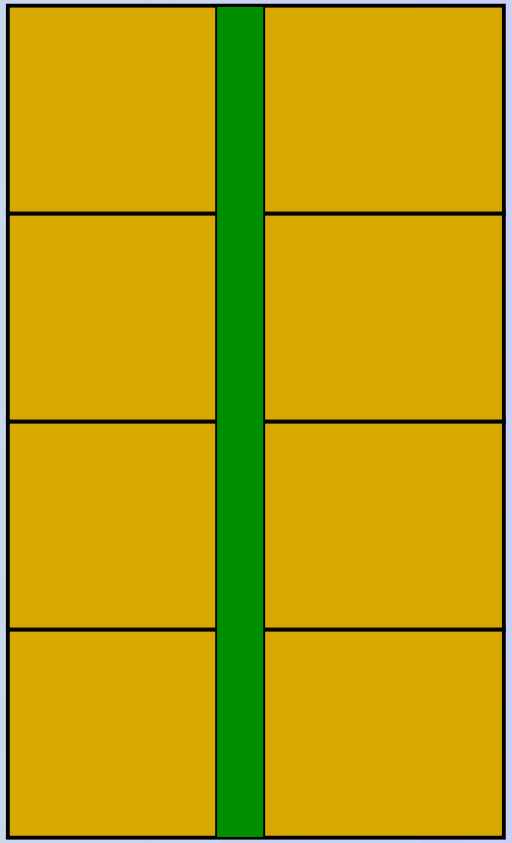
Chunking and selections

Great performance



Selection coincides with a chunk

Poor performance



Selection spans over all chunks

Things to remember about HDF5 chunking

- ✓ Use appropriate chunk sizes
- ✓ Make sure that cache is big enough to contain chunks for partial I/O
- ✓ Use hyperslab selections that are aligned with chunks
- ✓ Memory may grow when application opens and modifies a lot of chunked datasets

Variable length datasets and I/O

- Examples of variable-length data
  - String

    A[0] "*the first string we want to write*"

    ……

    A[N-1] *"the N-th string we want to write"*
  - Each element is a record of variable-length

    A[0] (1,1,0,0,0,5,6,7,8,9) length of the first record is 10

    A[1] (0,0,110,2005)

    ……..

    A[N] (1,2,3,4,5,6,7,8,9,10,11,12,….,M) length of the N+1 record is M

Variable length datasets and I/O

- Variable length description in HDF5 application

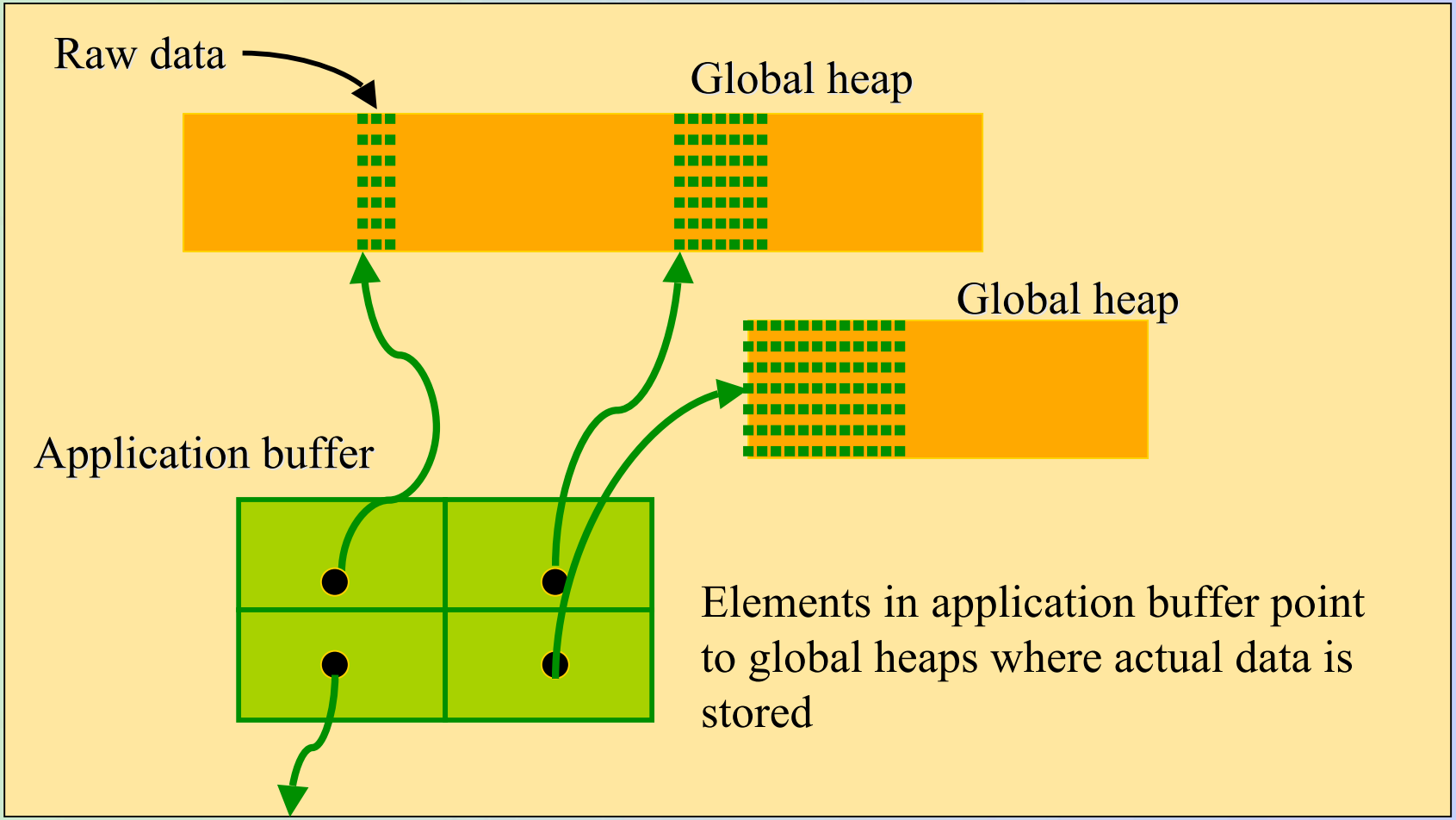
```
typedef struct {
        size_t length;
        void   *p;
}hvl_t;
```

  - Base type can be any HDF5 type

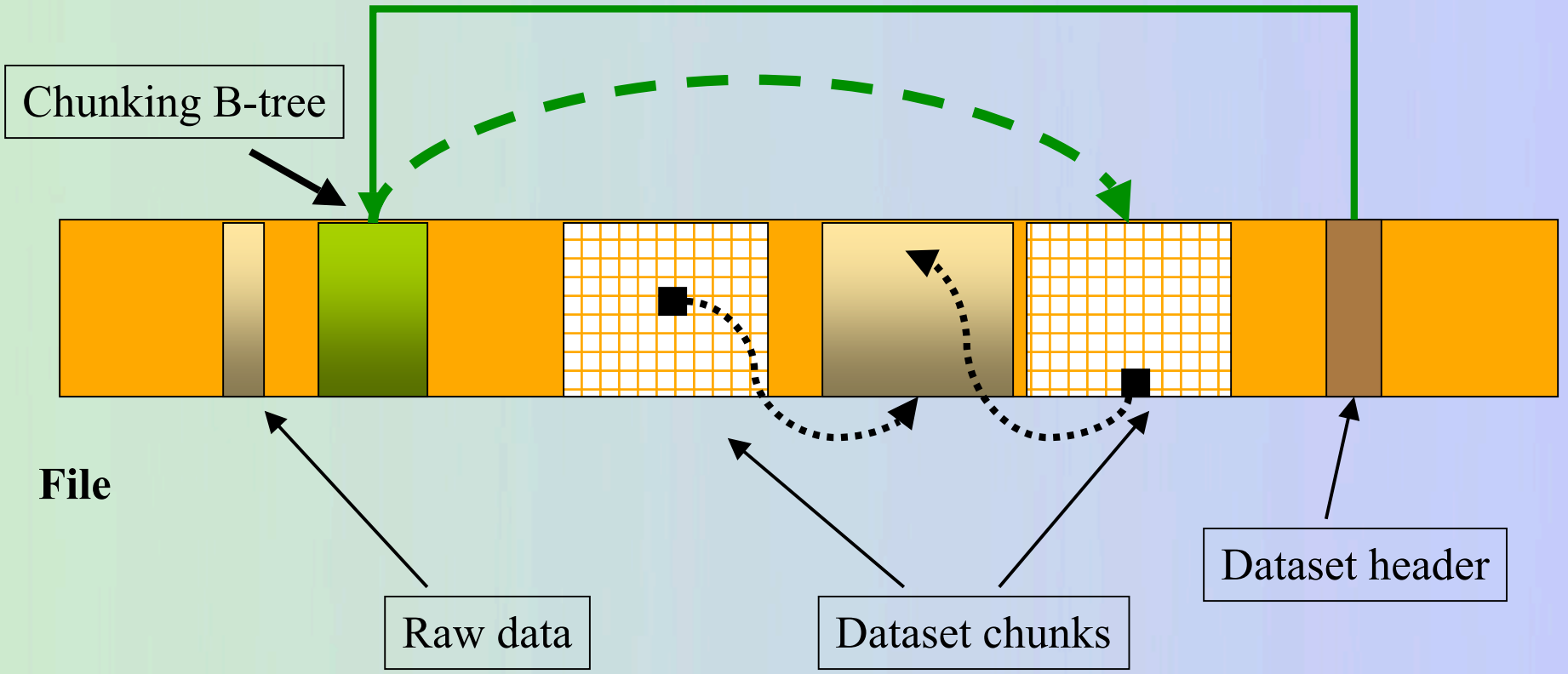
```
H5Tvlen_create(base_type)
```

  - ~ 20 bytes overhead for each element
  - Raw data cannot be compressed

Variable length datasets and I/O



VL chunked dataset in a file



Variable length datasets and I/O

- Hints
  - Avoid closing/opening a file while writing VL datasets
    - global heap information is lost
    - global heaps may have unused space
  - Avoid writing VL datasets interchangeably
    - data from different datasets will is written to the same heap
  - If maximum length of the record is known, use fixed-length records and compression

Crash-proofing

Why crash proofing?

- HDF5 applications tend to run long times (sometimes until system crashes)
- Application crash may leave HDF5 file in a corrupted state
- Currently there is no way to recover data
- One of the main obstacles for productions codes that use NetCDF-3 to move to NetCDF-4
- Funded by ASC project
- Prototype release is scheduled for the end of 2007

HDF5 Solution

- Journaling
  - Modifications to HDF5 metadata are stored in an external journal file
  - HDF5 will be using asynchronous writes to the journal file for efficiency
- Recovering after crash
  - HDF5 recovery tool will replay the journal and apply all metadata writes bringing HDF5 file to a consistent state
  - Raw data will consist of data that made to disk
- Solution will be applicable for both sequential and parallel modes

Thank you!

Questions ?