

# Advanced Message- Passing Programming

---

Alternative Parallel IO Libraries

# ARCHER Training Courses

---

Sponsors



# Reusing this material



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

# Overview

- Issues with MPI-IO
- HDF5
- NetCDF
- Availability on ARCHER
- Summary

# MPI-IO Issues

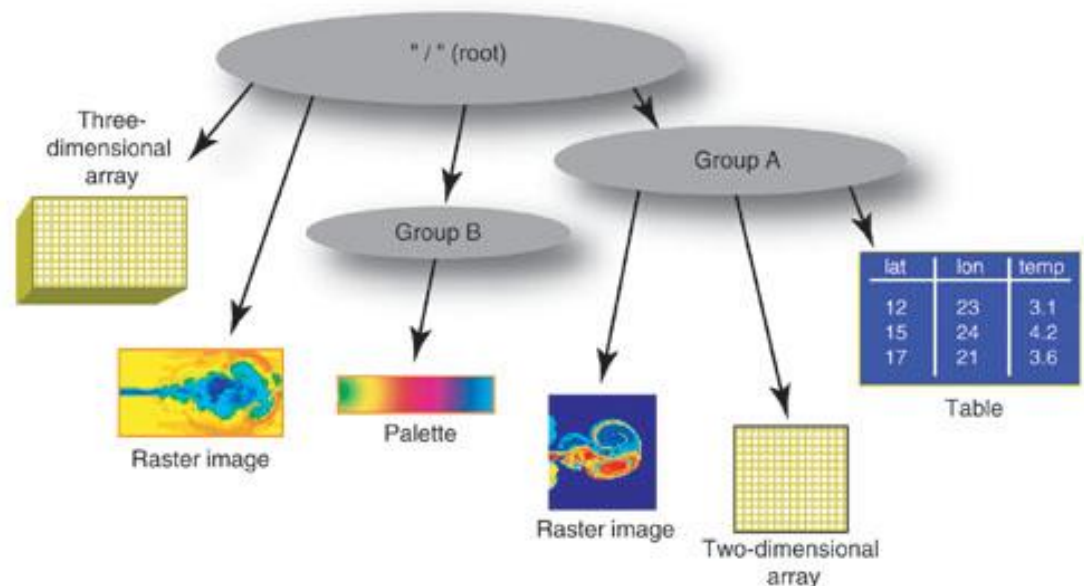
- Files are raw bytes
  - no header information
  - storage is architecture-specific (e.g. big / little-endian floating-point)
- Difficult to cope with in other codes downstream
  - user must write their own post-processing tools
  - c.f. cioview / fioview with “metadata” encoded in file name!
- But ...
  - it can be very fast!

# Solution

- For functionality
  - define higher-level formats
  - include metadata, e.g. “this is a 4x5x7 array of doubles”
  - enables standard data converters, browsers, viewers etc.
- For performance
  - layer on top of MPI-IO
- Many real applications use higher-level formats
  - understanding MPI-IO will enable you to get performance as well

# HDF5

- “**Hierarchical Data Format (HDF)** is a set of file formats (**HDF4, HDF5**) designed to store and organize large amounts of data.” (Wikipedia)
  - data arranged like a Unix file system
  - self-describing
  - hierarchical
  - can use MPI-IO



# Parallel HDF5 (Fortran)

- Approach much like MPI-IO

- describe global dataset

**MPI\_ORDER\_**  
**FORTTRAN**

... describes its local portion(s) of the g

**global data,**  
**encodes sizes**

```
CALL h5sselect_hyperslab_f(filespace, &  
    H5S_SELECT_SET_F, offset, &  
    count, error)
```

**starts**

- Then call collective write

- hyperslabs can be merged to create global file
  - actual file IO done through MPI-IO
  - important to choose collective IO

**subsizes**



# NetCDF: Network Common Data Form

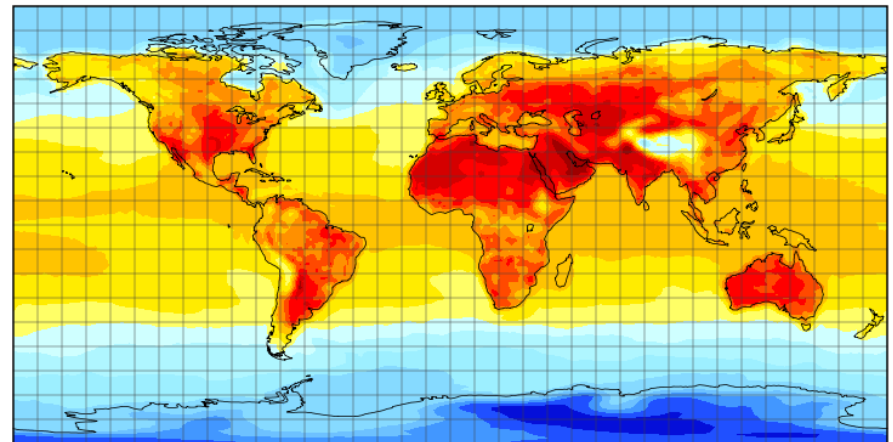
- “a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data..” (Wikipedia)

- more restricted than HDF5
- common in certain communities
  - climate research
  - oceanography
  - GIS ...

- Rich set of tools
  - data manipulation
  - visualisation
  - ...

txxETCCDI\_yr\_MIROC5\_historical\_r2i1p1\_1850-2012.nc

Annual Maximum of Daily Maximum Temperature



Annual Maximum of Daily Maximum Temperature (degrees\_C)  
-2.0E+01 -4.9E+00 1.0E+01 2.5E+01 4.0E+01 5.5E+01  
Data Min = -2.0E+01, Max = 5.5E+01

image taken from <http://live.osgeo.org>

# Parallel NetCDF (Fortran)

file identifier

sizes

```
nf90_def_var(ncid, "data", NF90_DOUBLE, dimids,  
varid) )
```

...

```
nf90_var_par_access(ncid, varid, nf90_collective)
```

...

```
nf90_put_var(ncid, varid, buf, start, count)
```

Write\_all()

starts

subsizes

10

# ARCHER details

- HDF5

- `user@archer:~> module load cray-hdf5-parallel`
- interfaces to Cray MPI-IO

- NetCDF

- `user@archer:~> module load cray-netcdf-hdf5parallel`
- interfaces to HDF5 ...
- ... which interfaces to Cray MPI-IO

# Summary

- MPI-IO may seem a little low-level
  - but is building block of parallel IO on ARCHER
- Higher-level formats layer on top of MPI-IO
  - to benefit from performance work by Cray, Lustre etc.
- Common formats are HDF5 and NetCDF
  - both supported on ARCHER
- Understanding MPI-IO performance is key to getting good performance for HDF5 and NetCDF