The NetCDF file format (CDF-1, CDF-2) provides a simple yet quite capable file format. The format lends itself quite well to MPI-IO optimizations. We would like to retain the simplicity of the file format while also allowing for arbitrarily large variable and record sizes.

CDF "Two and a Half"

Russ Rew made an interesting observation that the variable size field in the CDF-2 file format is redundant: you can compute it by taking the product of the dimensions. By ignoring this field, variables in a CDF-2 formatted file can have a quite large size: they are just restricted to 2^{31} elements in a dimension. We've run into a few groups that do want to put, for example, 5 billion elements in one dimension, but this approach has some appeal for its simplicity, and could buy some time while we work on a real 64 bit file format.

CDF-5

We must make several changes to CDF-2 if we want 64 bit dimensions:

- Some fields on-disk must be 64 bit.
- many platforms have 32 bit integers, so an array of ints will not be adequate to address variables with 64 bit dimensions. (start, count, stride).
    - In Pnetcdf we have use MPI_Offset for all these fields. MPI_Offset will be big enough to address large files (and in practice, that means 64 bits)
    - We might be able to define a new set of 64-bit functions. We'd have to implement all of these in serial netcdf, but in Pnetcdf they could be macros to the (already 64 bit clean) existing API.
- We have to ensure the header processing understands both the new format and the CDF-1 and CDF-2
- Then we have to get serial-netcdf to incorporate the changes.

Northwestern Meeting

- test suite for pnetcdf (could mimic what romio does)
- switch to using vectors instead of subarray (could keep subarray in smaller dimension cases)
- re-learn how to write out / read back header in the CDF-5 case (we do need CDF-1 -2 and -5 support)

Milestones

- Define new on-disk format: NewFileFormatDefinition
- Read new file format
- co-exist with CDF-1, CDF-2
- Modify serial NetCDF code to handle 64 bit addressing
- testcases
    - pass nc_test
    - pass FLASH-I/O

Friendly User Testing

The quick rundown on how to try this stuff out: NewFileFormatCode