2009 LCI Conference Tutorial

HDF5 and netCDF-4: Two Solutions for Data Management Problems Based on One File Format

Elena Pourmal, Albert Cheng, Ruth Aydt (The HDF Group)
Ed Hartnett (Unidata/UCAR)

Overview

Over the past 20 years, the HDF (www.hdfgroup.org) and netCDF (www.unidata.ucar.edu) file formats have become de facto standards for storing, managing, and exchanging data in science and engineering communities. Petabytes of data have been written in both formats and used in many endeavors, including climate change modeling, weather prediction, nuclear fusion simulation, non-destructive material testing, bioinformatics, and high-resolution imaging.

Advances in high-performance computing have made it possible to model and study very complex phenomena in a wide range of scientific fields while producing, accessing, and analyzing gigabytes of complex and diverse data. Efficient data management, including seamless data interoperability, is a critical part of the scientific discovery process that presents new challenges to the users and maintainers of scientific data formats.

This tutorial introduces HDF5 and netCDF-4, a new version of netCDF built on top of HDF5. HDF5 and netCDF-4 were created to address data management needs in today's heterogeneous and quickly evolving high-performance computational environments. Both software packages provide efficient and scalable access to data by taking advantage of underlying file system capabilities and I/O libraries. Based on the same file format, netCDF-4 and HDF5 provide different views of data, with netCDF-4 focusing on data simplicity, and HDF5 focusing on data complexity and heterogeneity. The last two sections of this document provide additional information about HDF5 and netCDF-4.

Tutorial Outline

This full-day tutorial will provide participants with the background they need to use HDF5 and netCDF-4 effectively on high-performance Linux clusters. The general outline for the topics to be covered will progress from basic to advanced, with a mixture of case studies, presentations, and demos designed to keep the participants actively engaged throughout the day. Participants will be encouraged to ask questions throughout presentation, with time allowed at the end of the day for more in-depth discussions.

1. Introduction to HDF5

The tutorial will explain the HDF5 data model and show how applications can take advantage of the model to represent their data structures. The data model discussion will include an overview of HDF5 abstractions such as datasets, groups, attributes, and datatypes. Simple C and

Fortran examples will cover programming model and API design, and will help new users navigate through the rich collection of HDF5 interfaces. HDF5 tools and online utilities for creating, managing, and browsing data stored in the HDF5 files will be demonstrated.

2. Advanced HDF5 features

To achieve good performance with HDF5 and netCDF-4, applications developers need to understand HDF5's advanced optimization features including partial I/O, chunking, compression, and metadata cache management. It is important to use these features appropriately to achieve good performance and efficient storage**.** A substantial amount of time will be spent on these features in recognition of their critical importance to developers of high-performance applications. The tutorial will explain how HDF5 handles application data, and discuss how to use HDF5's performance tuning capabilities to improve sequential I/O, to handle large numbers of objects in HDF5 files, and to match data layouts to application access patterns.

3. Parallel HDF5

This part of the tutorial is designed for users who have had exposure to MPI I/O and who would like to learn about the parallel HDF5 library. It will cover parallel HDF5 design and programming models and APIs. C and Fortran examples will be used to demonstrate the capabilities of the HDF5 parallel library. The tutorial will discuss the performance of the parallel HDF5 library and its tuning capabilities to improve parallel I/O. The h5perf tool, which comes with the parallel HDF5 library, will be used to compare the performance of parallel HDF5, MPI I/O, and POSIX I/O for different access patterns and storage layouts. HDF5 parallel applications developers can use the tool to evaluate the performance of each layer on their HPC systems and tune their applications.

4. NetCDF-4

Using the example of netCDF-4, the tutorial will show how common data models and their implementations can take advantage of access and space optimization features in HDF5 to achieve scalable I/O. Both the classic and enhanced netCDF data models and APIs will be introduced, and performance results for netCDF-4 will be shown. Examples will be presented in both C and Fortran. The tutorial will also cover parallel features of netCDF-4 and demonstrate how to move existing netCDF applications to use parallel I/O.

HDF5 Features

HDF5 was designed to store, access, manage, exchange, and archive diverse, complex data. It can handle all types of data suitable for digital storage, regardless of the data's origin or size. For example, petabytes of remote sensing data received from satellites, terabytes of computational results from weather and nuclear testing models, and megabytes of high-resolution MRI brain scans are stored in HDF5 files, along with the additional information necessary for efficient data exchange, processing, visualization, and archiving. HDF5 has a rich and sophisticated set of features for optimizing storage space and access time, including compression, chunking, metadata caching, and an extensible set of I/O drivers.

In recent years, the number of applications that successfully use HDF5 in fields other than physical sciences and engineering has increased. HDF5 was employed in the production of visual effects for the "Lord of the Ring" sequel. Many applications in bioinformatics use HDF5 to manage an avalanche of DNA sequencing data. Other applications use HDF5 as a container for heterogeneous data, for example, for storing audio and video streams along with analysis data and visualization results. One of the more unorthodox examples is an application in the field of Behavioral Neurobiology that uses HDF5 to study animal vocal behavior.

The robustness of HDF5, and the availability of open source and commercial tools for analysis and visualization of data stored in the HDF5 format, has made HDF5 an attractive standard data format for companies and government organizations concerned with reducing data management costs. In June 2008, NASA endorsed HDF5 as a data standard for Earth Science Data Systems.

HDF5 runs on a variety of platforms from Windows desktops to high-performance Linux clusters. The HDF5 library comes with C, C++, Fortran, and Java programming interfaces. It is developed and supported by The HDF Group, a non-profit corporation with a mission to ensure the long-term accessibility of HDF data (www.hdfgroup.org).

NetCDF-4 Features

Developed at the Unidata Program at UCAR, netCDF is widely used in atmospheric and oceanographic sciences. Programming interfaces to access data stored in the netCDF files are available in C, C++, Fortran77, Fortran90, Java, Ruby, Python and many other languages.

NetCDF-4 is a new version of netCDF. Built on top of netCDF-3 and HDF5, it empowers and extends the netCDF data model with HDF5 features including a grouping mechanism and a rich collection of datatypes, while preserving the simplicity of the original netCDF data model. NetCDF-4 takes advantage of the efficient I/O and storage capabilities provided by the HDF5 library. New features of netCDF-4 enabled by HDF5 include large file support, multiple unlimited dimensions, parallel I/O, and data compression.

NetCDF-4 is API backward and file format compatible. Applications that use netCDF-3 can be re-linked with netCDF-4 and store data in the original netCDF format, or they can be easily modified to take advantage of the new features in the HPC environment.