
HDF5-iRODS Project

Final report

Peter Cao
The HDF Group
1901 S. First Street, Suite C-2
Champaign, IL 61820
xcao@hdfgroup.org

Mike Wan
San Diego Supercomputer Center
University of California
La Jolla, CA 92093
mwan@sdsc.edu

August 20, 2008

Acknowledgements

This project was sponsored by CIP/NLADR, an NSF PACI Project in Support of NCSA-SDSC Collaboration. The project was managed by the CyberInfrastructure Partnership (CIP), a joint effort led by the National Center for Supercomputing Applications (NCSA) and the San Diego Supercomputer Center (SDSC). The work was carried out by The HDF Group and the SDSC SRB team. The ASC/Alliance Center for Astrophysical Thermonuclear Flashes at the University of Chicago provided FLASH simulation data (HDF5 files) and other help for this project. The islice tool was built based on the extract_slice_from_chkpt tool developed by Paul Ricker (UIUC/NCSA). Ruth Aydt (The HDF Group) also contributed to the project.

Executive Summary

Numerous scientific teams use the HDF5 format to store very large datasets. Efficient use of this data in a distributed environment depends on client applications being able to read any subset of the data without transferring the entire file to the local machine. The goal of the HDF5-iRODS Project was to develop an HDF5-iRODS module for the iRODS datagrid server that supported this capability, and to apply the technology to an NCSA/SDSC Strategic Applications Program (SAP) project, FLASH.

A joint team from The HDF Group (representing NCSA) and the SDSC SRB group collaborated to accomplish the project goal. The team implemented five HDF5 microservices functions on the iRODS server, and developed an iRODS FLASH slice client application. The client implementation also includes a JNI interface that allows HDFView, a standard tool for browsing HDF5 files, to access HDF5 files stored remotely in iRODS. Finally, three new collection client/server calls were added to the iRODS APIs, making it easier for users to query the content of an iRODS collection.

Introduction

Simple data reduction operations and filters are commonly applied to large datasets. The resulting datasets are often quite small relative to the size of the original datasets. The time required to transfer a large file to a local machine can be very long when the file is stored remotely. A great deal of time can often be saved by performing the data reduction and filtering operations at the remote site and only transferring the results to the local machine.

An application developed previously, *HDF5-SRB*, enabled a client to access objects in an HDF5 file stored remotely in a Storage Resource Broker (SRB) without transferring the entire file. The success of the HDF5-SRB project prompted NCSA and SDSC to undertake the current project which implemented similar capabilities for the new iRODS technology, and provided support for an NCSA/SDSC Strategic Applications Programs (SAP) project. FLASH was the SAP project selected for the initial implementation.

Since the fall of 2004, the SDSC SRB team has been working on the next generation data management cyber-infrastructure, *i* Rule Oriented Data Systems (iRODS). Unlike SRB, where the policies used to manage the data at the server level are hard-coded, the iRODS architecture provides the means for encoding customized data management functionalities in an easy and declarative fashion using the Rule Oriented Programming (ROP) paradigm. Also, iRODS is open source. Because of the advantages of iRODS over SRB, NCSA, SDSC, and The HDF Group agreed to move to iRODS technology. A major task for this project was the transfer of previously-developed HDF5-SRB technology to HDF5-iRODS technology. Once this work was complete, its usefulness was verified for the FLASH SAP project.

FLASH Cosmology is a set of FLASH modules for cosmological applications, particularly the simulation of large-scale structures. FLASH simulations produce a variety of output data, require very fast I/O, and must run on high-performance parallel machines. Because of these characteristics, FLASH simulation output data is stored in HDF5 files.



Accessing FLASH simulation data stored remotely presents major challenges. Due to the large scale structures, FLASH simulations generate terabytes of data. While HDF5 supports high-performance access to subsets of the data in a local file, it can take hours to transfer the entire file over the network. Furthermore, having multiple copies of the entire file greatly increases overall storage capacity requirements. A main goal of the HDF5-iRODS project was to address these issues. The HDF5-iRODS module offers the following major benefits to the FLASH community:

1. **Reduces the need for large storage on local machine.** Terabytes of data reside remotely, only small subsets are staged locally.
2. **Supports fast browsing of data objects and metadata.** Users can browse data objects by examining the structure of a file without loading the data content. This capability can make an enormous difference for files with large numbers of data objects, or with very large objects.
3. **Provides rapid access to selected data and metadata.** By transferring only the selected data/metadata to the client, instead of the entire multi-terabyte file, transfer time is reduced tremendously, especially when wide-area networks are involved.
4. **Facilitates data sharing among scientists.** An iRODS server provides a common lookup for distributed data, making it easily shared by many scientists without requiring them to have multiple copies of the same file on their local machines. Furthermore, everyone will have access to updated data after a new simulation run.

Work Performed

- The iRODS micro-services are small, well-defined procedures/functions that perform certain tasks within the iRODS infrastructure. In order to support HDF5 with iRODS, five new micro-services were implemented on the server side.
 - msiH5File_open
 - msiH5File_close
 - msiH5Dataset_read
 - msiH5Dataset_read_attribute
 - msiH5Group_read_attribute
- To make these micro-services easier to use, five client functions were also created.
 - clH5File_open
 - clH5File_close
 - clH5Dataset_read
 - clH5Dataset_read_attribute
 - clH5Group_read_attribute

- Three new collection client/server calls were added to make it easier for users to query the contents of a collection registered with an iRODS server.
 - rcOpenCollection
 - rcReadCollection
 - rcCloseCollection
- *islice*, a command line tool for use with FLASH data was developed. The tool extracts a slice, perpendicular to one of the coordinate axes, from a FLASH output file stored on an iRODS server. The selection of a slice is based on the *extract_slice_from_chkpt* tool, developed by Paul Ricker (UIUC/NCSA). The extracted slice is saved to raw floats and jpeg files on the local machine. The *islice* tool has been tested with FLASH simulation data provided by the ASC/Alliance Center for Astrophysical Thermonuclear Flashes at the University of Chicago.
- iRODS is written in C. An HDF5-iRODS JNI was implemented to interoperate between Java applications and the HDF5-iRODS module. This JNI allowed a Java application to be built that could talk to an iRODS server and display HDF5 data residing on the server.
- To support iRODS connectivity from HDFView, a set of iRODS HDF5 Java objects were added to the HDF-Java products.
 - H5SrbFile.java
 - H5SrbGroup.java
 - H5SrbScalarDS.java
 - H5SrbCompoundDS.java
 - H5SrbDatatype.java
- The HDFView GUI was improved to support iRODS. The improved GUI does not depend on the jargon package, but instead uses the new collection client/server calls.
- The HDF5-iRODS module, the FLASH slice tool, and the improved HDFView were tested using an iRODS server run by The HDF Group.
- iRODS 1.1, which includes the HDF5-iRODS module, has been released.
- The HDF5-iRODS User's Guide, final project report, and distribution README files have been written. The User's Guide is undergoing a final editorial pass prior to publication.

Remaining Tasks

The following tasks remain to be completed. The HDF Group and the SDSC SRB team will continue to work on these tasks over the next few months.

- Deploy HDF5-iRODS to evaluation server at NCSA or SDSC (TBD)
- Release *islice*, the FLASH slice tool (Nov. 10, 2008)
- Release HDFView with HDF5-iRODS support (Oct. 10, 2008)

- Publish the User's Guide (Dec. 10, 2008)
- Present the work at the HDF and HDF-EOS Workshop XII (Oct. 15-17, 2008, Aurora, CO)
- Present a poster at the eScience 2008 conference, pending acceptance by the program committee (Dec. 7-12, 2008, Indianapolis, IN)

Proposed Next Steps

The current project supported the development of the HDF5-iRODS technology and the demonstration of its usefulness for a particular high-performance science application. In order for the work to have a broader and sustained impact on the science community, further action is needed. Some proposed work that could be undertaken with additional support is outlined here. If funding is possible, a more detailed proposal can be written.

- Perform updates and tests of the HDF5-iRODS module as HDF5, iRODS, and underlying system technologies continue to evolve.
- Support the staff responsible for running iRODS servers and installing the HDF5-iRODS software. Or, alternatively, perform those activities directly.
- Support science communities as they become familiar with HDF5-iRODS.
- Add new functions and capabilities requested by users.
- Support additional FLASH operations such as statistics, data projections, etc.
- Build a general framework that will allow the HDF-iRODS module to be applied to other SAP applications.
- Implement general command line tools to access HDF5 files stored on iRODS servers, such as *ih5ls* (list data objects in an HDF5 file) and *ih5read* (read a specific dataset or subset of a dataset).