HDF5 File Format Meeting

Location & Date:

- University of Amsterdam, The Netherlands
- September 9-10, 2010

Overview:

- Quincey Koziol from the HDF5 Group visited the LOFAR crew at the UvA on Thursday-Friday (Sept. 9-10th). The agenda is listed below. We gave Quincey an overview of LOFAR along with an introduction to the LOFAR data format/structures from the Interface Control Documents (ICDs). All presentations are attached below. Quincey's HDF5 presentations can also be found below.

Participants:

- HDF5 Group Member - Quincey Koziol
- Attendees - Anastasia Alexov, Ken Anderson, John Swinbank, Lars Baehren, Micheal Wise, Adriaan Renting, Ger van Diepen, Tom Bennett (South Africa via EVO)

Agenda

Thursday 10AM-lunch:

- LOFAR Overview (15-30 mins) - Swinbank, An Introduction to LOFAR
- LOFAR Image cubes (15-30 mins) - Anderson, LOFAR Sky Image Cubes
- (coffee break @11AM)
- LOFAR Beam-formed data (15-30 mins) - Alexov, LOFAR Beam-Formed Data & Pipeline Overview
- LOFAR Coordinates (15-30 mins) - Baehren, The Data Access Library ([[engineering:software:tools:DAL]])
- (lunch)

Thursday 2PM-5PM:

- HDF5 talk/tutorial (60 min) - Quincey Koziol
- Discussion on questions/concerns listed below - All

Friday 10AM-lunch:

- Performance questions (15-30 mins) - van Diepen, HDF5 and casacore
- Open discussion on performance
- (coffee break @11AM)
- HDF5/LOFAR-related questions - All
- (lunch)

Friday 2PM-5PM:

- Open discussion - All

HDF5 Presentations

- Introduction to HDF5
- Introduction to HDF5 - Session Two - Data Model Comparison; HDF5 File Format
- Introduction to HDF5 - Session Three - HDF5 Software Overview
- Introduction to HDF5 - Session Four - Java Products
- Introduction to HDF5 - Session Five - Reading & Writing Raw Data Values
- Introduction to HDF5 - Session Seven - Datatypes
- Introduction to HDF5 - High Performance I/O
- Introduction to HDF5 - Introduction to Mathematical Concepts
- Data Storage and I/O in HDF5
- HDF5 Update

QUESTION/TOPIC for DISCUSSION:

- performance benchmarks
- access patterns
- intelligent filtering and slicing
- efficient data storage / chunking
- parallel I/O
- distributed storage
- tools which exist for HDF5 (other than HDFView, h5py, pytables)

CONCERNS:

- Speed when using smallish hyperslabs
- Robustness. E.g. What happens if a system crashes after writing some hours of data? What are the chances that all data are lost? What impact has robustness on performance?
- The future of the C++ interface.
- Is HDF5 always backward compatible? I.e. can a file created with HDF5 1.8 always be read with newer versions, even a hypothetical version 2.1 (AFAIK HDF5 1.8 is not backward compatible with 1.6). The same for building the software.
- Is it more efficient to store data in:

1. One table per subband (N x 1-dimensional tables in one Group)
2. One table for all subbands (1 x N-dimensional table in one Group)
3. 1 Array per Subband (N x 1-dimensional array in one Group)
4. 1 Array for all Subbands (1 x N-dimensional array in one Group)

LOFAR ICD (Interface Control Documents):

LOFAR Interface Control Documents (ICDs):
http://usg.lofar.org/wiki/doku.php?id=documents:lofar_data_products

From:
https://www.astron.nl/lofarwiki/ - **LOFAR *Wiki***

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:hdf5

Last update: **2017-03-08 15:27**