Fusion Engineering and Design 85 (2010) 583–590

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Commonalities and differences between MDSplus and HDF5 data systems

G. Manduchi*

Consorzio RFX, Euratom-ENEA Association, Corso Stati Uniti 4, Padova 35127, Italy

ARTICLE INFO

Article history:
Available online 24 April 2010

Keywords:
Data systems
Databases
Data acquisition

ABSTRACT

MDSplus is a data acquisition system widely used in nuclear fusion experiments. It defines a file format for pulse files and provides a set of tools for data acquisition and management. The whole MDSplus package is used in several fusion experiments to set up and supervise the data acquisition process. Other experiments use only the data management layer of MDSplus to provide a common format for data exchange between plasma fusion laboratories.

HDF5 is a file format and a data access library used by a larger community, mainly outside fusion. HDF5 is used, for example, in earth science research, defence applications and weather services. HDF5 allows managing large and complex data sets and provides a common data format among heterogeneous applications.

Both MDSplus and HDF5 support a rich set of data types and a hierarchical data organization, as well as multi-language data access libraries. There are however several significant differences between the two system architectures, making each system better suited in different application contexts.

The paper provides a brief overview of the data architectures of MDSplus and HDF5 and analyzes in detail the peculiar aspects of the two systems.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

MDSplus is a package for data acquisition used in many fusion devices [1]. Among other tools, MDSplus defines a data format in order to store in a file structure, called Pulse File, all the data related to a given experiment, such as configuration data, acquired signals and the results of on-line elaboration. Data are organized as a tree in order to reflect the hierarchical organization of the underlying experiment. Many data types are supported, ranging from scalars to multidimensional arrays. More specialized data types are used to describe accessory information such as the operations performed during the experiment sequence. A multi-language data access library allows users reading and writing data during the experiment sequence and afterwards. More recently, an Object-Oriented (OO) data access interface has been developed. This interface presents the same OO framework in several languages such as C++, Java, and Python. Other languages, although not natively Object Oriented are being integrated, such as MATLAB and IDL, and an F90 interface is foreseen.

The Hierarchical Data Format (HDF5) [2] implements a model for storing and managing data. HDF5 is intended to provide a common data format for heterogeneous applications in fields such as earth sciences and meteorology. In HDF5 data are stored in a hierarchical format and the data model supports a rich variety of data types and data space organizations. The HDF5 conceptual model of data is called the Abstract Data Model, and is independent of storage medium or programming environment. Data objects from the Abstract Data Model are manipulated via a Programming Model, that is a model of the computing environment. The concrete implementation of the Programming Model is the library defining the HDF5 Application Programming Interface (API). HDF5 provides Java, C and FORTRAN data access libraries.

It is worth noting that neither MDSplus nor HDF5 are relational database systems. The structure of data items is not organized in tables, and the database represents instead a picture of a set of data related to a single event. Even if such event is normally represented by a physics experiment for MDSplus, or some kind of computation for HDF5, both systems can be used for storing computation results as well as acquired data. Therefore MDSplus and HDF5 do not present sophisticated data search capabilities as those provided by the select operation in a relational database.

The MDSplus and HDF5 data systems may at a first glance appear quite similar – both refer to a hierarchical database, support many data types and provide a multi-language data access library – but there are several important differences which can make either system preferable, based on the requirements of the target application. The rest of the paper will highlight such differences and their impact on functionality and performance.

* Tel.: +39 049 829 5039; fax: +39 049 870 0718.
E-mail address: gabriele.manduchi@igi.cnr.it.

0920-3796/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.fusengdes.2010.03.055

2. Data hierarchy

The hierarchical organization of data items is similar in MDSplus and HDF5. A data item is referred to as a tree node in MDSplus and a dataset in HDF5. Before accessing data, the database must be open in both systems. The hierarchical organization of HDF5 is similar to that of the UNIX file system, defining directories (data groups) which may contain datasets or other data groups. Data objects (datasets and groups) can be renamed, and more flexible hierarchies can be defined by using the link operation, which allows a given data instance to be declared as belonging to different data groups. HDF5 documentation, however, warns against an excessive usage of renaming and linking, which could lead to situations in which sets of data objects are no more reachable. The syntax used to specify data paths is the same as used in UNIX file systems, e.g. ***/group1/group2/data1***.

In MDSplus the hierarchical organization is achieved via tree nodes and members, and all data descend from a root node named TOP. Data can be associated only with members, and nodes are used to group members and other nodes. The syntax for defining path names distinguishes between member and nodes. For example, path ***TOP.NODE1:MEMBER1*** indicates the data item named ***MEMBER1***, which is a child of node ***NODE1***, which in turn is a descendant of the root node ***TOP***. Renaming is allowed, but no explicit link operation is provided in MDSplus.

MDSplus is more restrictive than HDF5 as regards node and member names, which are limited to 12 chars in length. MDSplus, however, provides an alternative naming mechanism, that is the possibility of associating a unique tag name with members and nodes. Tag names are unique within a given subtree (subtrees will be explained later), and more than one tagname can be associated with each data item. Tagnames are very often used in place of the actual path name.

Note that MDSplus tagnames and HDF5 links provide a similar functionality and are useful for identifying a given data item under different perspectives.

3. Data types

Both systems support a variety of scalars and multidimensional arrays. The way data are defined is however very different. The definition of data is more straightforward in MDSplus: a new data item is created by instantiating the corresponding class. In HDF5, instead, data items are represented by datasets, and each dataset defines a *datatype* and a *dataspace* component. Datatypes define the type of the atomic data, whereas dataspaces define the number of dimensions and, for each dimension, the maximum and the current size. There are also several differences in the way data items are overwritten in a database. In MDSplus there is no restriction on the type and shape of the data which are being written, provided the declared usage of the target tree member is compatible. Usages are defined in MDSplus to differentiate between numerical and string data types, and from the data types which describe the configuration of the experiment.

In HDF5, once a dataset has been written in the database, it can be overwritten only with another compatible dataset, i.e. for which a pre-defined type conversion is available and whose shape has the same size. Otherwise, the dataset has to be explicitly deleted before writing a new one.

The difference in the programming interface is illustrated by the following example, where a 3×2 integer array is created and written in the database named MY_DATA (with shot 100 for MDSplus). The path name of the data item is ***/group1/data1***. The HDF5 examples are written in C, while MDSplus examples are in C++. A C++ interface exists in HDF5, but it is only a wrapper to the C interface. For simplicity, in the following examples, no error checking is carried out.

Download English Version:

https://daneshyari.com/en/article/272374

Download Persian Version:

https://daneshyari.com/article/272374

Daneshyari.com