

Management of Meteorological Mass Data with MongoDB

Richard Lutz¹, Parinaz Ameri², Thomas Latzko³, Jörg Meyer⁴

Abstract

The remote sensing of atmospheric trace gases investigates dynamic, microphysical and chemical processes in the Earth's atmosphere, with the goal to understand, quantify and predict its natural variability and long-term changes. Accurate measurements of atmospheric trace gases from various observational platforms (ground-based stations, air craft, balloons, satellites) provide the data that are required for the modelling of atmospheric processes. The instrument GLORIA (Gimballed Limb Observer for Radiance Imaging of the Atmosphere), developed by KIT/IMK and FZ Jülich, an Infrared Spectrometer, which measures atmospheric emissions, was engaged in several measurement campaigns on board of HALO (High Altitude and Long Range Research Aircraft) and provided a large amount of data, which has to be managed efficiently for processing and visualisation. This paper describes the system background and the use of MongoDB for the provision of measured and processed mass data.

1. Introduction

The observation of the atmosphere is still one of the most important subject areas to obtain the necessary knowledge about meteorological and chemistry phenomena which influence climate change or greenhouse effect. Several remote sensing campaigns are performing worldwide and a huge amount of data is gathered and processed. In order to enable efficient processing and visualisation of the collected data, sophisticated and effective methods and tools are necessary. Although a lot of powerful databases and storage tools are available, which allow the management of big data, the best solution for this is to assemble best fitting tools.

This paper describes the system background and the use of MongoDB for the provision of measured and processed mass data collected by the remote sensing instrument GLORIA (Gimballed Limb Observer for Radiance Imaging of the Atmosphere) on board of the aircraft HALO (High Altitude and Long Range Research Aircraft). The processing of the data leads to chemical information (correlations and distributions) about trace gases in the atmosphere.

1.1. The instrument GLORIA

The Gimballed Limb Observer for Radiance Imaging of the Atmosphere (GLORIA) is a recently developed unique atmospheric remote sensing instrument that bridges the gap from scanning to imaging in the infrared spectral domain. This is realized by the combination of a classical Fourier transform spectrometer (FTS) with a 2-D detector array tailored to the FTS needs. Imaging improves the spatial sampling by an order of magnitude in comparison to state of the art limb scanning instruments. In addition to the limb mode, the instrument can also perform Nadir

¹ Karlsruhe Institute of Technology (KIT), 76344 Eggenstein-Leopoldshafen, Germany, Richard.Lutz@kit.edu, Institute for Applied Computer Science (IAI)

² Karlsruhe Institute of Technology (KIT), 76344 Eggenstein-Leopoldshafen, Germany, Parinaz.Ameri@kit.edu, Steinbuch Centre for Computing (SCC)

³ Karlsruhe Institute of Technology (KIT), 76344 Eggenstein-Leopoldshafen, Germany, Thomas.Latzko@kit.edu, Institute for Meteorology and Climate Research (IMK)

⁴ Karlsruhe Institute of Technology (KIT), 76344 Eggenstein-Leopoldshafen, Germany, Joerg.Meyer2@kit.edu, Steinbuch Centre for Computing (SCC)

measurements. GLORIA is designed to operate on the high altitude research platforms aircraft [6] and stratospheric balloons [4]. Figure 1 shows a graphic of GLORIA and an image of HALO which carries GLORIA as one of several instruments. GLORIA is attached underneath the fuselage of HALO (white arrow).

GLORIA is a joint development of the two national research centres of the Helmholtz Association Karlsruhe Institute of Technology (KIT) and Forschungszentrum Jülich (FZJ). Detailed information about the instrument is presented at [5].



Figure 1: The instrument GLORIA and the HALO aircraft (arrow: the location of GLORIA).

1.2. The GLORIA data structure

The GLORIA data consists of three major parts: the pre-processed measurement data, the processing configuration data, and the final result data (spectra, trace gas distributions, etc.). The first objective is to store the pre-processed data which is the base for further processing and visualisation.

Figure 2 shows the data structure of a pre-processed GLORIA dataset provided by the measurement campaigns on board of HALO [6]

A flight delivers a dataset and may take several hours. The files of a dataset are called "cubes" and contain a measurement sequence of about 3 to 12 seconds. A measurement sequence is a sequence of recorded "images" (slices). A single slice is delivered in a time period of 15ns-300ms by a detector field with 256x256 pixels. Depending on the scientific needs, only a subset of the full detector array is used for performing the measurements.

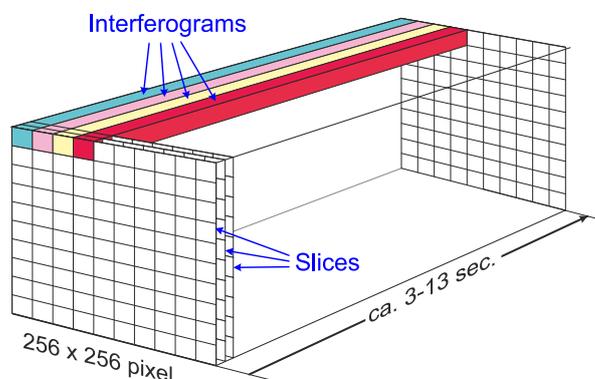


Figure 2: The structure of a GLORIA dataset (GLORIA cubes)

The chronological order of each value of a given slice pixel leads to an interferogram. The size of a dataset varies between 100 MB and 8 GB, depending on the GLORIA recording mode and the

measurement duration. A standard dataset of the last campaigns contains approximately 74.500 slices for a detector sub array of 128x48 pixels and has the size of 1 GB.

A HALO flight in 2012 took 12 hours and produced about 11.500 datasets (≈ 2.2 TB). Scientists are particularly interested in processing interferogram data. Since the GLORIA dataset files contain the ordered list of slices, the datasets must be converted to be stored in MongoDB based on the resulting interferograms.

1.3. The data management and processing environment

The Large Scale Data Facility (LSDF) at KIT is a data centre for large scientific data from several data intensive disciplines [3]. The LSDF provides petabytes of storage accessible via the protocols NFS, CIFS, and GridFTP. In addition, the LSDF consists of a Hadoop cluster including a Hadoop Distributed File System storage, and an OpenNebula cloud environment. LSDF serves as storage backend for multiple iRODS instances and for the NoSQL database MongoDB.

The MongoDB instance for GLORIA data runs on a dedicated hardware server with 12 physical CPU cores and 128 GB RAM. The cluster file system GPFS of LSDF is attached via a 10G ethernet connection. The operating system is a 64 bit Linux (CentOS 6). As of writing this paper, the version of the MongoDB server is 2.4.8.

In Figure 1, the setup of the database infrastructure is depicted. The MongoDB server provides data stored on LSDF to various clients (Processing, Loader and Visualization). So far, a single instance of MongoDB was sufficient to handle all data, but in order to scale out the infrastructure so-called sharding is being investigated. This allows splitting the data based on a shard key and distributing them to several instances. In addition, replica sets provide a mechanism to replicate data to many redundant instances in order to increase the reliability of the system.

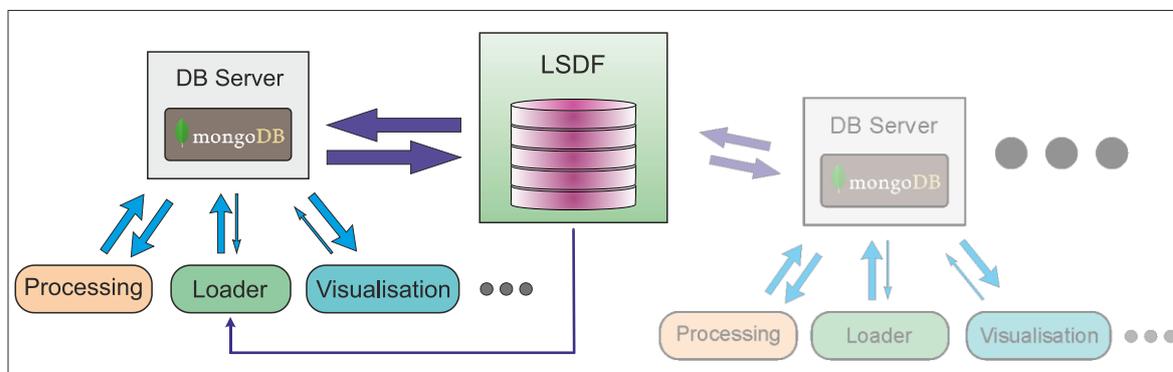


Figure 3: The database server MongoDB provides data to processing, loader, and visualization clients. The Large Scale Data Facility serves as storage backend. Additional instances of MongoDB are foreseen

1.4. The GLORIA data processing

One central issue is the processing workflow of GLORIA's data. The process chain consumes the measured data and metadata to generate the calibrated spectra (Figure 4). Due to the detector properties, each pixel has different sensitivity and nonlinearity behaviours. Experiments under laboratory conditions are performed to determine a set of parameters describing the nonlinearity. The first step is to use this set of parameters to perform a nonlinearity correction. The frames (slices) in the cube are measured at an equidistant time grid. The spectrum is based on an equidistant spatial grid.

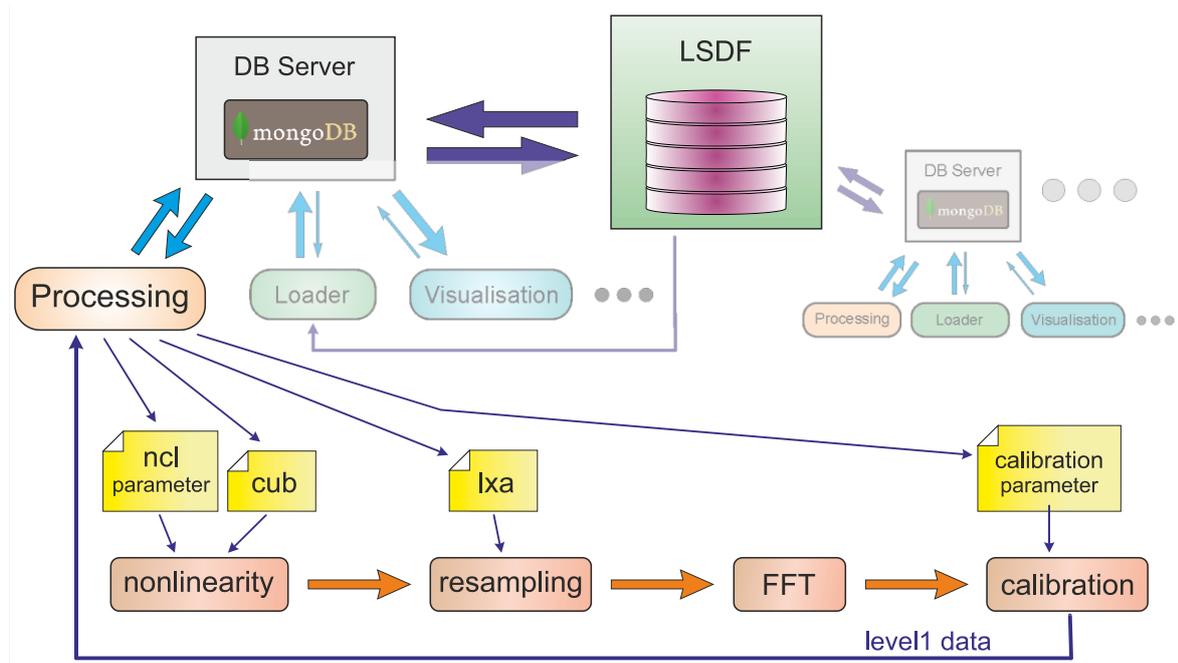


Figure 4: The Data Processing Environment

During the measurements, an equidistant spatial grid is defined by using a reference laser with a known laser wavelength. At every zero-crossing, an event sets a timestamp. With this timestamps, a tuple of time and space points are created. Performing a Whittaker-Shannon interpolation

$$I(x) = \sum_n I_n \text{sinc}\left(\frac{x - n\Delta T}{\Delta T}\right)$$

where x is the target point in space, I_n is the n -th measurement and T is the time difference, the measurement is resampled from a time grid to a equidistant space grid. After the Fourier transform, the spectrum needs to be calibrated. Before the actual processing, the calibration parameters for each sequence are generated and stored in the Database. The data for the calibrations consists of blackbody measurements where the radiance is known and also measurements looking in the deep space where a minimum of trace gases are expected. The computation of this parameter sets requires the combination of different measurements at different times to increase the signal-to-noise ratio on the one hand and to generate a continuous set of parameters for every measurement on the other hand. In order to access various combinations of measurement data, the used database must provide facilities for complex queries. After the computation with the loaded parameter set, the calibrated spectra (*level1 data*) are stored in the database with an easy to access interface for further analysis.

2. Why MongoDB?

MongoDB is an open-source database used by companies of all sizes, across all industries and for a wide variety of applications. It is an agile database that allows schemas to change quickly as applications evolve, while still providing the functionality that developers expect from traditional databases, such as secondary indexes, a full query language and strict consistency [9].

The first approach to manage the GLORIA data was reuse of the software environment of its predecessor instrument MIPAS which is available on aircraft, balloon and for ground-based measurements. Furthermore, MIPAS provided a huge amount of data as one instrument aboard the European environmental satellite ENVISAT.

The software environment for the MIPAS aircraft and balloon experiments is based on Firebird, a free parallel development of Borland's™ relational database Interbase. The database manages the metadata of the measurements with links to the data files which reside on the file system.

The data generated by GLORIA is approximately hundred times the data produced by MIPAS. For this reason, the data management has to be reconfigured. Additionally, several fundamental new requirements exist for the data management which must be fulfilled by the new environment. The most important requirements are:

- Access to data and metadata within one single database query
- Usage of a free database which can be managed easily
- The data schema should be open for enhancements and modifications
- Fast access to large quantities of data
- Horizontal scalability
- Provision of powerful interfaces (drivers) for Java, Python, C++, etc.
- Possibility of array-oriented storage of datasets

2.1. Access to data and metadata within one single database query

The first important requirement for the new environment was access to data and metadata within one single database query, i.e. one query should provide the metadata as well as the underlying measurement data. Therefore, it is possible to get complete big datasets via the internet without multiple accesses to the database. Only few relational databases provide such an access mode. For instance Oracle™ offers its BFiles which are external BLOBs stored in files on the file system. The BFiles can be read similar to standard column types.

2.2. Usage of a free database which can be managed easily

In order to perform experiments and various different performance and access tests with large data sets, it is necessary to install and manage the database without big effort. The creation, deletion and modification of databases, database users and access permissions are mostly performed automatically by use of script files or via interfaces.

2.3. Free data schema for enhancements and modifications

On demand frequent changes is part of nature of scientific software development. These changes are applied based on experiences gathered during the development or based on technical innovations and enhancements of the measurement system. Therefore, changes on the data schema are often necessary. A very easy way to modify the structure of data storage is the use of schema-free NoSQL databases, a relatively young database type [10]. The most popular representatives of about 150 NoSQL databases are HBase, CouchDB, and MongoDB. While HBase is a so-called Wide Column Store database, CouchDB and MongoDB are document-based, i.e. the entries of a document collection are documents, which represent a list of key-value pairs in any order. Search indexes can be created for singular and combined keys. Several performance tests concerning the type and amount of data as well as the usability of desired interfaces showed that MongoDB would meet our requirements better than other NoSQL databases.

Figure 2 shows the basic difference between the relational database and a schema-free document database structure. While the tables in a relational database have a fixed format and fixed column order, a MongoDB collection can contain entities of different types in any order. The entities are grouped in a “document” which resembles a relational table row but with variable length. The

element *dbRef* allows the creation of an explicit reference to another document in the same database or in another database on another server.

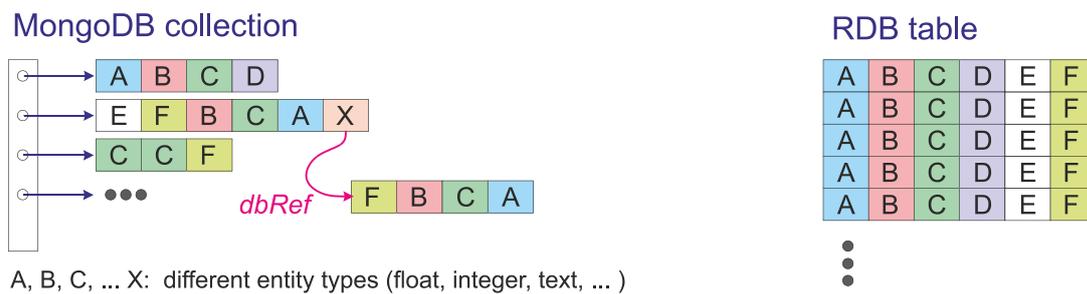


Figure 2: The basic data structure of MongoDB and a Relational Database

2.4. Fast access to large quantities of data

The fixed table structure of relational databases facilitates the recognition of the stored element types. This advantage is compensated by the ability of MongoDB to allow additions to collections and documents without restructuring the schema. A fast access to desired documents must be realized via indexes on the identifying key names. Several tests showed that the search for documents for given keys in large databases must be based on explicit indexes for the desired keys to find the data in a fast way. Without indexes the search took an unexpectedly long time.

Based on current experience and several performance tests with about 57 million documents, MongoDB is able to fulfil the access requirements for large quantities of data. Powerful caching and indexing mechanisms support the search in big data stores. A typical query scenario needs about 1-2 seconds (see subsections 3.3 and 3.4).

2.5. Horizontal scalability

This requirement is based on the importance to access big datasets from different locations. The GLORIA project covers a lot of scientists at different scientific organisations which must have a fast access to the data. MongoDB is built for scalability, performance and high availability, it is scaling from single server deployments to large and complex multi-site architectures. MongoDB's native replication and automated failover enable enterprise-grade reliability and operational flexibility. MongoDB features like auto-sharding and replication support the realisation of this requirement. There are some tests scenarios still in progress to check the availability and performance of a sharded system.

2.6. provision of powerful drivers

The GLORIA software environment uses different approaches to compute the given measurement data. Results are interferograms, spectra and trace gas distributions. To address these challenges, different tools based on different programming languages are used. Like other databases, MongoDB offers a lot of drivers. For the most important languages in the GLORIA project, C++, Java, and Python, powerful drivers are available.

2.7. Possibility of array-oriented storage of datasets

The GLORIA data is mostly structured as value arrays (see subsection 1.2). Experiences with big satellite data showed that databases offering array elements are particularly suitable for this use case. Only a few databases provide this kind of element type (e.g. Oracle™, Firebird, MongoDB) whereas a powerful support for array elements is given with Oracle and MongoDB. While Oracle allows the definition and management of nearly every primitive (short, int, float, etc.) or complex

element type, MongoDB offers arrays only for its standard BSON types [1, 9]. Both databases provide convenient methods to access the arrays within database queries.

3. Experiences with GLORIA data on MongoDB

Most of the tests have been performed to examine the usability and reliability of MongoDB for the given requirements. The major challenge is to test with big data, because the cube data has to be converted before being stored (see section 1.2).

For the test scenario, loading, converting and storing the data of a selected HALO/GLORIA flight took about 34 hours for 2 TB. Therefore, some important configuration and structural decisions have to be taken before loading the complete data. The following subsections describe some essential aspects regarding the usability of MongoDB for the given test scenario.

3.1. Installation and maintenance

The download and installation of a MongoDB server is very easy to perform. Like other databases, MongoDB provides a shell environment to access the database. MongoDB offers an interactive JavaScript shell. The creation of users, databases, collections, indexes and access permissions is done very easily and quickly. Simple and complex queries can be performed easily, too.

For each database, MongoDB creates an “unlimited” number of 2 GB database files and creates a journaling directory. The maximum document size is 16 MB, larger documents can be stored in so-called GridFS files, a kind of BLOB file which is referenced by database objects.

3.2. MongoDB BSON types and conversions

MongoDB stores documents on disk in the BSON serialization format [1]. BSON is a binary representation of JavaScript Object Notation (JSON) documents, though it contains more data types than JSON [7]. The 'smallest' BSON type is a 32 bit integer, but majority of GLORIA data is of type short (16 bit interferogram values). Therefore, if the data would store without conversion, the storage of this data using 32 bit integer arrays would duplicate the storage size unnecessarily. Therefore the *short* arrays are converted to binary arrays before storing them into MongoDB.

In the beginning of the project, there was a request that the interferogram-based datasets should be queried for selected slices (see subsection 1.2). MongoDB offers direct access to selected array elements, which must be BSON types. An option is the Binary type, an object that carries a byte array. Due to the fact that the storage and retrieval of Binary arrays is considerably time-consuming, the interferograms are stored as a pure byte arrays with the consequence to renounce the direct array access.

The small number of BSON types for small elements ((unsigned) int and short, float, etc.) turned out to be obstructive in some scenarios.

3.3. Indexing and searching for data

MongoDB offers a powerful indexing mechanism that allows the fast retrieval of data for a large amount of documents. If a document has to be found by given attributes, an index should be created for each desired attribute and/or combination of attributes. The interferogram collection contains about 57 millions of documents with the major attributes *cubeID* (e.g. “20120926_060047”) and their coordinates on the detector field. A repeated search for all interferograms of a selected cube needs less than one second to get the resulting database cursor to the data. Note: the time to get the cursor does not imply the download of the data but only the result of the search inside the index file (size of 9 GB for the test scenario). Arbitrary queries can be performed via the MongoDB shell or via a driver using a JavaScript syntax with MongoDB extensions (less or greater than,

contains, etc.). In contrast to SQL, the attributes, which has to be excluded within a query, must be listed explicitly.

3.4. Loading and retrieving data - caching mechanisms (observations)

During the tests it turned out that the retrieval of data stored in MongoDB does not only depend on common caching mechanism (whether the data was already accessed before), but also on the order the data was inserted into the database. A complete cube data is read much faster than the data resulting from a query for all interferograms of a given detector field coordinate of the whole flight. Although the data size is comparable, the download time differs considerably.

4. Conclusions

The storage of GLORIA data for the given requirements is a big challenge to any database. The underlying test scenario comprises a GLORIA/HALO flight with about 11.500 cubes and 57 million of interferograms and a resulting data size of 2,2 TB. With MongoDB a suitable way is found to manage the data. The utilisation of MongoDB for the storage of GLORIA data shows the following advantages and some disadvantages during the performed test:

- Easy and fast installation of MongoDB, simple creation of users, databases, and collections
- Fast access, the attributes together with the interferogram data can be retrieved within one single query
- All required drivers with rich functionality are available for the integration in applications
- Rich query features are available using JavaScript via a command shell and via drivers
- Open-source application
- Good indexing and caching mechanisms are provided
- The lack of small primitive BSON data types is negative for the scenario

Up to now, the basic tests are completed. The next step will be the conversion and storage of all other available GLORIA data which will lead to a resulting estimated size of 30 TB. Further tests have to be performed exploring the performance difference when using several instances or one single database instance for all flights.

References

- [1] Internet site of BSON Specification, <http://bsonspec.org/>, last visited 05/2014.
- [2] Friedl-Vallon, F. et al, "Instrument concept of the imaging Fourier transform spectrometer GLORIA", *Atmos. Meas. Tech. Discuss.*, 7, 2301-2337, doi:10.5194/amtd-7-2301-2014, 2014.
- [3] Garcia, A. O. et al, "Data-intensive analysis for scientific experiments at the Large Scale Data Facility", *Proceedings on Symposium on Large Data Analysis and Visualization (LDAV 2011)*, IEEE Computer Society Press, pp. 125-126, DOI: 10.1109/LDAV.2011.6092331, 2011.
- [4] GLORIA Internet site of Helmholtz Association, <http://gloria.helmholtz.de/>, last visited 05/2014.
- [5] GLORIA Internet site of Earth Observation Portal, <https://directory.eoportal.org/web/eoportal/airborne-sensors/gloria> , last visited 05/2014.
- [6] Internet site of DLR/HALO, <http://www.halo.dlr.de/>, last visited 05/2014.
- [7] Internet site of JSON Specification, <http://json.org/>, last visited 05/2014.
- [8] Kleinert, A. et al, "Level 0 to 1 processing of the imaging Fourier transform spectrometer GLORIA: Generation of radiometrically and spectrally calibrated spectra", *Atmos. Meas. Tech. Discuss.*, 7, 2827-2878, doi:10.5194/amtd-7-2827-2014, 2014.
- [9] Internet site of MongoDB, <http://www.mongodb.org/>, last visited 05/2014.
- [10] NoSQL Internet site, <http://nosql-database.org/>, last visited 05/2014.