METADATASTORE: A PRIMARY DATASTORE FOR NSLS-2 BEAMLINES

A. Arkilic, L. Dalesio, D. Allan, W.K. Lewis, T. A. Caswell, NSLSII, Upton, NY, USA

Abstract

The beamlines at NSLS-II are among the highest instrumented, and controlled of any worldwide. Each beamline can produce unstructured data sets in various formats. This data should be made available for data analysis and processing for beamline scientists and users. Various data flow systems are in place in numerous synchrotrons, however these are very domain specific and cannot handle such unstructured data. We have developed a data flow service, metadatastore that manages experimental data in NSLS-II beamlines. This service enables data analysis and visualization clients to access this service either directly or via databroker API in a consistent and partition tolerant fashion, providing a reliable and easy to use interface to our state-of-the-art beamlines.

INTRODUCTION

Historically, the experimental data in beamline experiments are either hosted in purpose-specific relational database management systems or text files. Relational database systems generally provide robust and high performance solutions for most experiments. However, they can only be adopted by certain experiments since two experiments in the same experimental floor usually have very different data formats and indexing requirements. Using text and/or hierarchical file systems such as HDF5 addresses some of these needs. File based data storage has served the legacy system in NSLS-I for almost 3 decades but the dramatic increase in data rates of NSLS2 beamlines made file I/O a massive bottleneck. In order to address these issues and provide NSLS2 beamlines with a uniform data storage solution that is applicable to all our beamlines, we have developed metadatastore service using NoSQL backend MongoDB [1].

ARCHITECTURE

metadatastore is the primary source of storage for scans, sweeps, etc. metadatastore is implemented as a RESTful web service on Tornado with a NoSQL MongoDB backend. Since it is document based, MongoDB does not have the traditional notion of tables. Each entry to the database is a document that is analogous to an "entry" in relational database terms. MongoDB collections serve as the containers of documents, providing its clients with means to meaningfully group their documents [2]. The metadatastore backend consists of 4 major collections: RunStart, EventDescriptor, Event, RunStop. One can think of RunStart and RunStop as the head and tail of a series of events that happen throughout an experiment (Figure 1). This approach eliminates the need to update the header of the experiment, which eliminates the problems that could occur due to lack of transactions in MongoDB. EventDescriptor(s) contain information regarding the data acquisition Event(s). In other words, they contain information about what are the

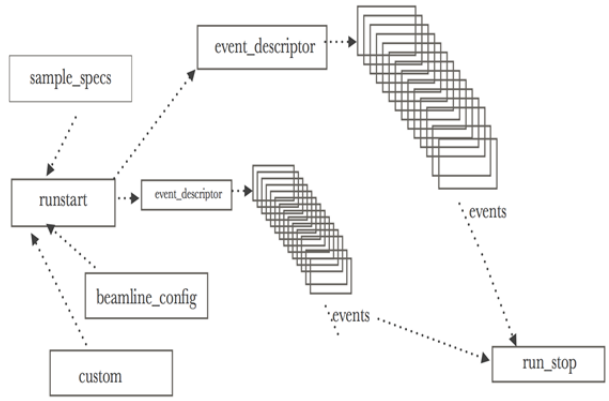

Figure 1: metadatastore Collection Relationship Overview.

"data_keys" that are being captured, what their shapes and data types are. This is immensely useful for data analysis toolkit in NSLS2 as this sort of header information is a major requirement to set up the data analysis environment prior to the arrival of actual data payload. In addition to the advantages provided so far, because of immense flexibility and performance of MongoDB, these RunStart, RunStop, and EventDescriptor documents support addition of any name-value pair or object in any hierarchy. Since Event(s) contain the actual data payload, they are defined as immutable documents. This semi-structured design pattern allows NSLS-II middle-layer to tackle many challenges including asynchronous scans, custom hardware brought from an external source, and varying data formats among experiments. For instance, a coherent X-ray diffraction and protein crystallography beamline utilize metadatastore in order to save their experiments despite their very different use-cases.

ISBN 978-3-95450-148-9

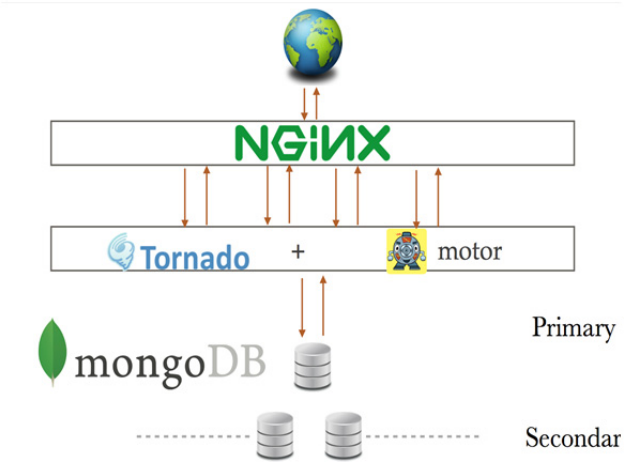

Figure 2: RESTful Web Service Overview.

The web service metadatastore is implemented in Python 2.7+ and Python 3. It includes some of the cutting edge developments in Python open-source community including asynchronous I/O. Historically, due to global interpreter lock, Python web applications has suffered from performance issues. This approach eliminates most of those problems, providing the scientific experiments with insert/query speeds far exceed their requirements (Figure 2). Tornado servers with 4 front-ends deployed with NGiNX load balancer on an Intel i7 processor can handle 8k+ requests per second.

FUTURE

Given the impact of metadatastore RESTful web service, there is an immense interest in developing GUIs and applications around metadatastore. JSON encoding might work for other industries that utilize MongoDB back-ends, however for scientific applications, they are not desirable due to performance and complexity of development as the GUIs need to have an understanding of data delivered for fast data processing pipelines. Due to its well-defined data layer pvData and high performance network protocol pvAccess, EPICS v4 is the best tool for this purpose. metadatastore documents can be served as NTTable or NTUnion over pvAccess to clients such as Control System Studio, which has support for EPICS v4. metadatastore utilizes the pipeline method that is custom built for streaming data in an extremely robust way. One of the major benefits of this protocol is the fact that all client side applications are essentially monitors. This allows extremely agile development, as one of the biggest over-head of development is to write layers that handle such handshake.

CONCLUSION

Metadatastore provides flexibl, high-performance, and rich semi-structured data storage for all beamline metadata. This service provides users with the ability to store asynchronous and hierarchical data of their choice, providing them with complex data regression capabilities. Compared to legacy approaches, due to its state of the art backend coupled with its narrow interface, metadatastore significantly reduces the time to locate any experimental run and its data points.

REFERENCES

[1] https://docs.mongodb.org/manual/core/data-model-design/

[2] NSLS-II High Level Application Infrastructure and Client API Design, G. Shen, L. Yang, K. Shroff Presented at the 2011 Particle Accelerator Conference

ISBN 978-3-95450-148-9