

Study of NoSQL technologies for the CMS Conditions Data

Extended abstract

Roland Sipos

For the CMS Collaboration
Roland.Sipos@cern.ch

1. Introduction

CMS (Compact Muon Solenoid) is an experiment at the LHC at CERN, with the goal to answer the most fundamental questions about the universe, with the essential help of several fields of computer science. The CMS Physics, Software & Computing Group, Computing Operations section (CMG-CO) plays a main role in the development and maintenance of the complex infrastructure for managing the detector's Alignment and Calibration constants („conditions”). This data is essential for the analysis and reconstruction of the recorded data.

1.1. Conditions Data

”Conditions” are non-event data, varying with time. These are critical for the data-flow and need to be properly re-synchronized during the data processing. For this, the Conditions Database is a dedicated infrastructure that includes applications and services for both online and offline environments. The data model consists of the following elements:

- **Payload:** Data structure for a given task, detector.
- **IOV (Interval of Validity):** Array of intervals.
- **Tag:** A label for categorizing a specific IOV.
- **Global Tag:** A consistent set of Tags, dedicated for a given work flow.

Conditions are *written once* and *never deleted*.

2. NoSQL - One size does not fit all

Currently the Conditions Databases uses a central, CERN provided database service that uses ORACLE servers. As the CMS Conditions' Data Model is relatively simple, and transactions are not necessary, the strict relational model may be needless for such application. *NoSQL* (Not only SQL) databases are following alternative solutions for storing data.

- **Models:** NoSQL databases may use non-conventional data models. Key-Value pair storage, Column Oriented BigTable approaches, or document stores.
- **CAP Theorem:** Also known as Brewer's Theorem states, that it is impossible to provide full consistency, availability and partition tolerance in the same time.
- **ACID versus BASE:** Compared to atomicity, consistency, isolation and durability, NoSQL databases provide basic availability, soft state, and most importantly eventual (tuneable) consistency.
- **Sharding:** A special way of horizontal partitioning, where the splitting of the dataset is done on potentially multiple instances of the schema.

The CMS Experiment is interested in these alternative technologies of storing large binary data (BLOBs), especially if the gain is significant.

3. Performance evaluation

The empirical evaluation of proposed NoSQL database candidates is necessary in order to check if a given prototype meets the performance criteria of the original CMS Conditions Database. For this I used a professional tool, called *JMeter*, thanks to it's great extendability and the mature support for creating really complex work flows.

3.1. Testing framework

A JMeter extension (called CustomSamplers) was implemented with CMS specific needs, in order to measure different scenarios of the Condition Database. It has support for a wide range of relational and non- relational databases. For each database the extension has:

- Deployers - for building up the data model for the different databases,
- QueryHandlers - for simulating the actual work flow that is used in the original database,

- ConfigElements - for configuring and storing persistence objects in the JMeter handled thread contexts,
- Samplers - for summarizing the results and reporting them for the listener elements,
- Utilities - in order to generalize frequently repeated tasks during the testing phase.

With this, and basic JMeter features, very complex CMS related work flows and test scenarios can be made that helps the deep performance evaluation of each candidate as a proposed alternative of the current databases.

3.2. Deployment

The administration of databases for each test scenario is a complex and time consuming task if it's not automated. In order to make the deployment environment independent, Puppet was used. With CMS specific, and already existing scripts, the most complicated configurations are set up in minutes on the deployment machines, that are provided by the *CERN Virtual Machine* and *OpenStack* virtualization services.

4. Outlook

As the project currently is in development, further work in the project involves:

- proper documentation and evaluation of test results for several databases (*MongoDB*, *CouchDB*, *Cassandra*, *RIAK*, *HBase*, *Hypertable*, etc),
- prototype creation for the promising NoSQL candidates,
- evaluate necessary code changes in the CMS Software Framework's Conditions software.

Acknowledgments. I would like to express my gratitude to Andreas Pfeiffer and András Benczúr, my supervisors for their help and support in the project.

References

- [1] A. PFEIFFER ET AL., CMS experience with online and offline databases
Journal of Physics: Conference Series (JPCS), 396 (2012), 052059.