Currently RDBMS (Oracle or MySQL) is used as the storage backend. To provide better performance and scalability the data stored in relational storage is partitioned into actual (live) and archive (historical) parts. But even in this scheme, as the archived data volume grows, the underlying software and hardware stack encounters certain limits that negatively affect processing speed and the possibilities of metadata analysis. Another problem of using the Oracle database is highly fragmented tables because the PANDA=>PANDAARCH data move was done from the PanDA server with conventional row insertion into the archive data and deletion from the operational data.

We aim to develop a NoSQL-based data storage for archived Panda data. In the validation stage, it would continuously function in parallel with Oracle Database.

Apache Cassandra, the open source distributed NoSQL database, which is capable of handling all of the big data challenges that might arise: massive scalability, an always on architecture, high performance, strong security, and ease of management, to name a few. Apache Cassandra is the technology of choice for such data-driven organizations as *Netflix, eBay,Constant Contact, Comcast, Barracuda Networks* and scores of others. And it furnishes the core functionality responsible for the superior real-time big data capabilities in DataStax Enterprise.

The official information provides the next information about Cassandra advantages which are crucial for us:

1) **No single point of failure, elastic scalability:** Rather than using a legacy master-slave or a manual and difficult-to-maintain sharded design, Cassandra is a peer-topeer distributed ring architecture that is much more elegant and easy to setup and maintain. It allows you to easily add capacity online to accommodate more customers and more data whenever you need. In Cassandra, all nodes are the same and there is no concept of a master node with all nodes communicating with each other via a *gossip* protocol. Cassandra s built-for-scale architecture means that is capable of handling petabytes of information and thousands of concurrent users/operations per second across one to many data centers as easily as it can manage much smaller amounts of data and user traffic. It means that also unlike other masterslave or sharded systems, Cassandra system has no single point of failure system and therefore is capable of offering true continuous availability.

2) **Transaction support** Delivers the AID in ACID compliance through its use of a commit log to capture all writes and built-in redundancies that ensure data durability in the event of hardware failures, as well as transaction isolation, atomicity, with consistency being tunable.

3) **Flexible data storage** Easily accommodates the full range of data formats structured, semi-structured and unstructured that run through today s modern applications. Also dynamically accommodates changes to your data structures as your data needs evolve.

4) **Data optimization in the background** - Data in Cassandra is sequentially appended, not placed in pre-set locations. It s continuously appends, merging, and compacting when needed.

5) **Support for data analytic functions with Hadoop -** Cassandra data model doesn t support JOINs or subqueries, except for batch analysis through Hadoop or Spark. To simplify the operational overhead of Hadoop by removing the single points of failure in the Hadoop NameNode, and to offer easy Hadoop integration for Cassandra users, the Cassandra File System (CFS) was designed. The CFS is an HDFS compatible filesystem, which was designed by DataStax Corporation to easily run analytics on Cassandra data. Now it is implemented as part of DataStax Enterprise, which combines Apache Cassandra and Solr together into a unified big data platform, CFS provides the storage foundation that makes running Hadoop-styled analytics on Cassandra data hassle-free. The main design goals for the Cassandra File System were to first, simplify operational overhead of Hadoop by removing the single points of failure in the Hadoop NameNode. And second, to offer easy Hadoop integration for Cassandra users (one distributed system is enough).

-- MariaGrigorieva - 2014-11-14

This topic: PanDA > NoSQLPandaArchive
Topic revision: r1 - 2014-11-14 - MariaGrigorieva