

CarbonDataProposal

Apache CarbonData

Abstract

Apache CarbonData is a new Apache Hadoop native file format for faster interactive query using advanced columnar storage, index, compression and encoding techniques to improve computing efficiency, in turn it will help speedup queries an order of magnitude faster over [PetaBytes](#) of data.

CarbonData github address: <https://github.com/HuaweiBigData/carbondata>

Background

Huawei is an ICT solution provider, we are committed to enhancing customer experiences for telecom carriers, enterprises, and consumers on big data, In order to satisfy the following customer requirements, we created a new Hadoop native file format:

- Support interactive OLAP-style query over big data in seconds.
- Support fast query on individual record which require touching all fields.
- Fast data loading speed and support incremental load in period of minutes.
- Support HDFS so that customer can leverage existing Hadoop cluster.
- Support time based data retention.

Based on these requirements, we investigated existing file formats in the Hadoop eco-system, but we could not find a suitable solution that satisfying requirements all at the same time, so we start designing [CarbonData](#).

Rationale

CarbonData contains multiple modules, which are classified into two categories:

1. CarbonData File Format: which contains core implementation for file format such as columnar,index,dictionary,encoding+compression,API for reading/writing etc. 2. CarbonData integration with big data processing framework such as Apache Spark, Apache Hive etc. Apache Beam is also planned to abstract the execution runtime.

CarbonData File Format

CarbonData file format is a columnar store in HDFS, it has many features that a modern columnar format has, such as splittable, compression schema , complex data type etc. And [CarbonData](#) has following unique features:

Indexing

In order to support fast interactive query, CarbonData leverage indexing technology to reduce I/O scans. [CarbonData](#) files stores data along with index, the index is not stored separately but the [CarbonData](#) file itself contains the index. In current implementation, CarbonData supports 3 types of indexing:

1. Multi-dimensional Key (B+ Tree index)
The Data block are written in sequence to the disk and within each data blocks each column block is written in sequence. Finally, the metadata block for the file is written with information about byte positions of each block in the file, Min-Max statistics index and the start and end MDK of each data block. Since, the entire data in the file is in sorted order, the start and end MDK of each data block can be used to construct a B+Tree and the file can be logically represented as a B+Tree with the data blocks as leaf nodes (on disk) and the remaining non-leaf nodes in memory.

2. Inverted index

Inverted index is widely used in search engine. By using this index, it helps processing/query engine to do filtering inside one HDFS block. Furthermore, query acceleration for count distinct like operation is made possible when combining bitmap and inverted index in query time. 3. MinMax index
For all columns, minmax index is created so that processing/query engine can skip scan that is not required.

Global Dictionary

Besides I/O reduction, CarbonData accelerates computation by using global dictionary, which enables processing/query engines to perform all processing on encoded data without having to convert the data (Late Materialization). We have observed dramatic performance improvement for OLAP analytic scenario where table contains many columns in string data type. The data is converted back to the user readable form just before processing/query engine returning results to user.

Column Group

Sometimes users want to perform processing/query on multi-columns in one table, for example, performing scan for individual record in troubleshooting scenario. In this case, row format is more efficient than columnar format since all columns will be touched by the workload. To accelerate this, [CarbonData](#) supports storing a group of column in row format, so data in column group is stored together and enable fast retrieval.

Optimized for multiple use cases

[CarbonData](#) indices and dictionary is highly configurable. To make storage optimized for different use cases, user can configure what to index, so user can decide and tune the format before loading data into [CarbonData](#).

For example

Use Case	Supporting Features
Interactive OLAP query	Columnar format, Multi-dimensional Key (B+ Tree index), Minmax index, Inverted index
High throughput scan	Global dictionary, Minmax index
Low latency point query	Multi-dimensional Key (B+ Tree index), Partitioning
Individual record query	Column group, Global dictionary

BigData Processing Framework Integration

- [CarbonData](#) provides [InputFormat](#) [OutputFormat](#) interfaces for Reading/Writing data from the [CarbonData](#) files and at the same time provides abstract API for processing data stored as Carbondata format with data processing framework.
- [CarbonData](#) provides deep integration with Apache Spark including predicate push down, column pruning, aggregation push down etc. So users can use Spark SQL to connect and query from [CarbonData](#).
- [CarbonData](#) can integrate with various big data Query/Processing framework on Hadoop eco-system such as Apache Spark, Apache Hive etc.

Example: <https://github.com/HuaweiBigData/carbondata/blob/master/examples/src/main/scala/org/carbondata/examples/CarbonExample.scala>

Initial Goals

Our initial goals are to bring [CarbonData](#) into the ASF, transition internal engineering processes into the open, and foster a collaborative development model according to the "Apache Way".

Current Status

CarbonData is production ready and already provide a large set of features. The current license is already Apache 2.0.

Meritocracy

We intend to radically expand the initial developer and user community by running the project in accordance with the "Apache Way". Users and new contributors will be treated with respect and welcomed. By participating in the community and providing quality patches/support that move the project forward, they will earn merit. They also will be encouraged to provide non-code contributions (documentation, events, community management, etc.) and will gain merit for doing so. Those with a proven support and quality track record will be encouraged to become committers.

Community

If [CarbonData](#) is accepted for incubation, the primary initial goal is to build a large community. We really trust that CarbonData will become a key project for big data column-like platforms, and so, we bet on a large community of users and developers.

Known Risks

Development has been sponsored mostly by a one company. For the project to fully transition to the Apache Way governance model, development must shift towards the meritocracy-centric model of growing a community of contributors balanced with the needs for extreme stability and core implementation coherency.

Orphaned products

Huawei is fully committed [CarbonData](#). Moreover, Huawei has a vested interest in making CarbonData succeed by driving its close integration with sister ASF projects. We expect this to further reduces the risk of orphaning the product.

Inexperience with Open Source

Huawei has been developing and using open source software since a long time. Additionally, several ASF veterans agreed to mentor the project and are listed in this proposal. The project will rely on their guidance and collective wisdom to quickly transition the entire team of initial committers towards practicing the Apache Way.

Reliance on Salaried Developers

Most of the contributors are paid to work in big data space. While they might wander from their current employers, they are unlikely to venture far from their core expertises and thus will continue to be engaged with the project regardless of their current employers.

An Excessive Fascination with the Apache Brand

While we intend to leverage the Apache 'branding' when talking to other projects as testament of our project's 'neutrality', we have no plans for making use of Apache brand in press releases nor posting billboards advertising acceptance of [CarbonData](#) into Apache Incubator.

Initial Source

<https://github.com/HuaweiBigData/carbondata.git>

External Dependencies

All external dependencies are licensed under an Apache 2.0 license or Apache-compatible license. As we grow the Carbondata community we will configure our build process to require and validate all contributions and dependencies are licensed under the Apache 2.0 license or are under an Apache-compatible license.

- Apache Spark
- Apache Hadoop
- Apache Maven
- Apache Commons
- Apache Log4j
- Apache Thrift
- Apache Zookeeper
- Scala
- Snappy
- Kettle (Pentaho)
- Eigenbase
- Fastutil
- GSON
- Jmockit
- Junit

Required Resources

Mailing lists

- private@carbondata.incubator.apache.org (moderated subscriptions)
- commits@carbondata.incubator.apache.org
- dev@carbondata.incubator.apache.org
- issues@carbondata.incubator.apache.org

Git Repository

- <https://git-wip-us.apache.org/repos/asf/incubator-carbondata.git>

Issue Tracking

- JIRA Project [CarbonData](#) (CarbonData)

Initial Committers

- Liang Chenliang
- Jean-Baptiste Onofré
- Henry Saputra
- Uma Maheswara Rao G
- Jenny MA
- Jacky Likun
- Vimal Das Kammath
- Jarray Qiuheeng

Affiliations

- Huawei: Liang Chenliang
- Talend: Jean-Baptiste Onofré
- Ebay: Henry Saputra
- Intel: Uma Maheswara Rao G

Sponsors

Champion

- Jean-Baptiste Onofré - Apache Member

Mentors

- Henry Saputra (eBay)
- Jean-Baptiste Onofré (Talend)
- Uma Maheswara Rao G (Intel)

Sponsoring Entity

The Apache Incubator