



International Journal of Marketing Management

ISSN 2454 - 5007



www.ijmm.net

Email ID: editor@ijmm.net , ijmm.editor9@gmail.com

Performance Evaluation and Optimization of Hybrid Indexes in Hive

1 G. Sravanthi, 2C.V.S. Satyamurthy

Abstract: Big data technologies decrease corporate risks and expenses by delivering crucial and accurate analysis that leads to concrete decision-making and improved operational efficiencies. You will need a system to manage and handle massive amounts of categorized and uncategorized data in real time, as well as to safeguard sensitive information, in order for this data to be useful. Operational Big Data and Analytical Big Data are two broad categories of big data products offered by many vendors, such as Amazon, IBM, and Microsoft. With Hive, an easy-to-use tool for query and analysis of data warehoused on Hadoop, we may recapitulate Big Data. Compression and Bitmap indexing are supported in order to increase query performance. Both indexes have been applied on the same partition in Hive and have seen significant speed improvements as a result. Analyzing Simultaneous Bitmap and Compact Indexing on a partition using text data has resulted in a significant speed improvement.

Keywords: Hive, Hadoop, Indexing, Compact Indexing, Hadoop infrastructure, Big Data, HiveQL.

1. INTRODUCTION

Data is being used by enterprise organizations to help them make better decisions and improve their performance. There has been a dramatic shift in the availability and efficacy of Organizational data over the past decade. As a result, data utilization has been transformed, and the notion of Big Data has emerged. [1]
BigData

Large amounts of data that are challenging to handle and extract from using typical storage

methods [2]. For these reasons, people have welcomed online businesses (Google, eBay, Facebook, LinkedIn etc).

Big Data for Small and Big Companies

It's easy to see why Big Data was first adopted by online businesses. These Enterprises and startups focus on leveraging rapidly challenging data and combining unstructured data with established approaches [3]. The following are some of the concerns or challenges that face Big-Data..

1,2Computer Science of Engineering, Information Technology

1,2CVRColegeofEngineering, Hyderabad, IndiaE-mail: sravanthi.guntakani15@gmail.com, E-mail:satyamurtycv@yahoo.co.in

Volume:Traditional databases couldn't handle the amount of data that was available, therefore a new approach was needed.

Variety: In comparison to earlier text and table format, current available versions are in the form of pictures, videos and tweets etc.

Velocity: Increased usage of Online Space and the data was available as rapidly changing and

available made available instantly and effectively [4].

Open Source Tool for Big Data Analytics

A distributed software solution Hadoop is a scalable default tolerance for data storage, process, and extracting the main uses.

i. HDFS (which is storage)

ii. MapReduce (Fig.1)

Fig1: Hadoop HDFS, MapReduce Logical View

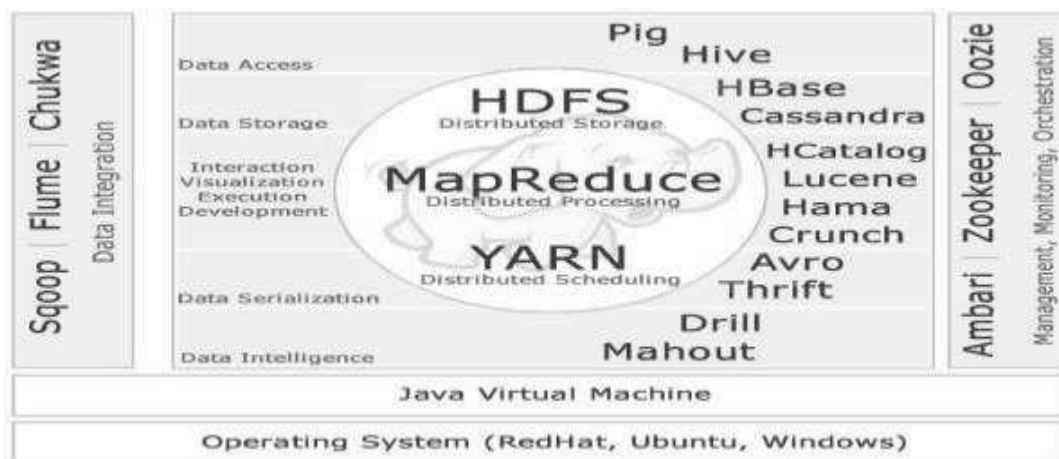
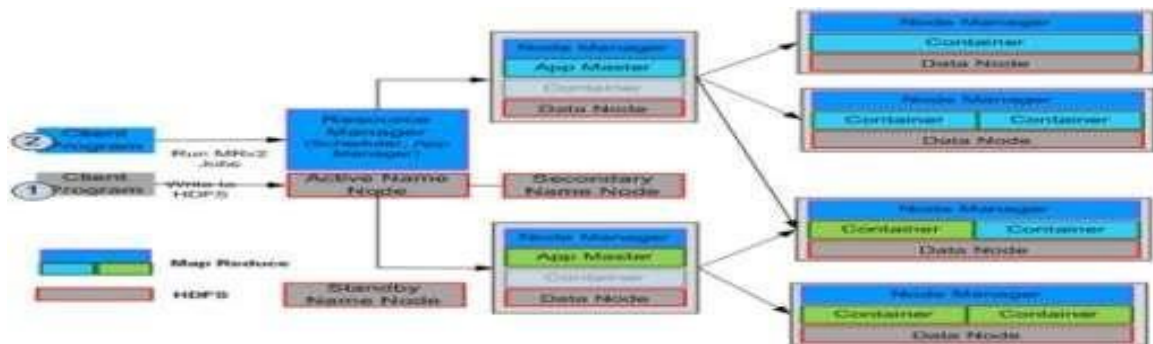


Fig2: The Hadoop technology stack

2. Related Word

Hive

An OLAP data warehouse that can handle and query large amounts of data in distributed storage is Hive. Huge amounts of data are reliably stored in the Hadoop Distributed File System (HDFS) [5] ecosystem, which is impervious to hardware failures.



Fig3: Layered Architecture of Big Data System

As a high-level programming paradigm, Hive, a SQL-like relational data warehouse technique, is built on top of Hadoop and MapReduce, a large computation in Hadoop that enables data streaming.

HiveQL It is a SQL-like query language, is used to query over Hive. Hive does not support SQL Commands (Update, Delete, and Insert at Row-Level) and transactions also. Hadoop Client Hive is to function with rare update and batch-mode data insertions are the characteristics of the Hive as an underlying system. Hive

an OLAP data for Online Transaction Processing (OLTP) features with big data, one should consider a NoSQL database [7].

Data in Hive Hive Data Types

Hive common relational databases data types as well as the three collection types: STRUCT, MAP, and ARRAY.

Hive File Formats

Hive can process data coming from different data sources using Extract, Transform and Load (ETL) tools for reading data from different

file formats and their customization.

Index Construction in Hive
`CREATE INDEX index_name`
`ON TABLE base_table_name (col_name,...) AS 'index.handler.class.name'`
`WITH DEFERRED REBUILD`
`[IDXPROPERTIES (property_name=property_value,...)]`
`[INTABLE index_table_name]`
`[PARTITIONED BY (col_name,...)]`
`[[[ROW FORMAT ...] STORED AS ...] STORED BY ...]`
`[LOCATION hdfs_path]`
`[TBLPROPERTIES (...)]`
`[COMMENT "index comment"]`

There are some parameters and keywords used in the command listing:

1. `index_name`: The index name given by the user used to access the index by the user itself. This name is used to refer to the index in `ALTER INDEX` and `DROP INDEX` commands.
2. `base_table_name (col_name,...)`: The table over which the index is to be created using the desired columns listed.
3. `'index.handler.class.name'`: This specifies the type of the index, which could be bit map or compact values.
4. `index_table_name`: The index name given by Hive (the default name) or the user, used to access the

index as a table by the user or Hive. For example, the user can see the content of an index using this parameter. Any other behavior of the index as a table can be addressed using this name.

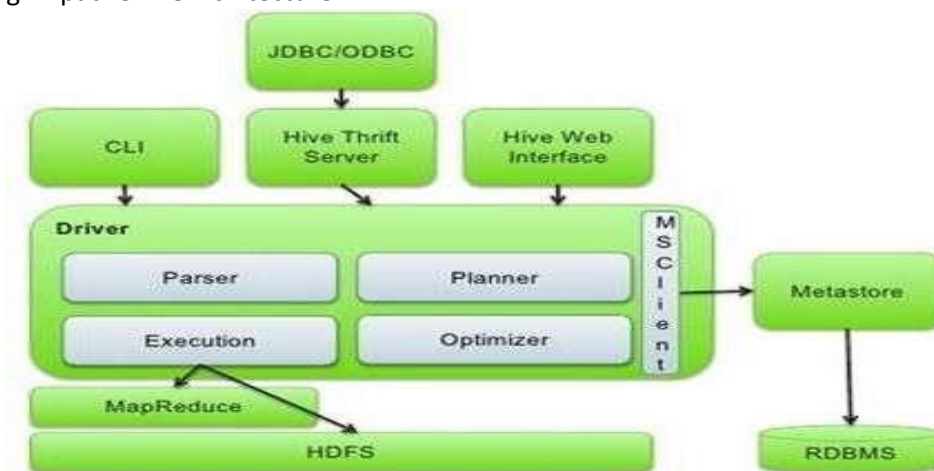
5. WITH DEFERRED REBUILD: This means the index starts empty. The index will be populated using: ALTER INDEX index_name ON table_name [PARTITION (...)] REBUILD

6. IDXPROPERTIES: This gives the properties of the index. For example: 'index_creator' = 'Mahsa' can be considered a property for an index.

7. INTABLE index_table_name: This is used when the user wishes to build an index in a different table from an already built index (if any), to keep them separate.

Hive Architecture

Fig4: Apache Hive Architecture



Major components of the Apache Hive are:

1. Metastore
2. Driver
3. Compiler
4. Optimizer
5. Executor CLI, UI, and Thrift Server

Compact Index, Aggregate Index, and Bitmap Index are all supported by Hive. In Hive, there are tables for each index dimension and the related index location arrays. Increase in the number of index dimensions results in an enormous table that takes up a lot of disk space.

The purpose of INDEX is to increase the speed at which data may be retrieved. There are two ways to accomplish the same thing: loading the complete table or partition to process records or loading only a portion of the file to process records with an index on column name.

Compact Index on Hive Table

Pair Indexed Column's value and its block id of Index

xColumn are stored in Compact indexing.

CREATE COMPACT INDEX:

```

CREATE INDEX index_name
ON TABLE base_table_name (col_name,...)
AS index_type
[WITH DEFERRED REBUILD]
[IDXPROPERTIES (property_name=property_value,...)]
[INTABLE index_table_name]
[[ROWFORMAT...]] STORED AS...
| STORED BY...
[LOCATION hdfs_path]
[TBLPROPERTIES (...)]
[COMMENT "index comment"];
  
```

Example

```

hive> CREATE INDEX index_students ON TABLE students (id)
  
```

```
>AS'org.apache.hadoop.hive ql.index.compact
.CompactIndexHandler'
>WITHDEFERREDREBUILD;OK
Timetaken:0.493seconds
```

ALTERCOMPACTINDEX

```
ALTERINDEXindex_nameONtable_name[PARTI
TIONpartition_spec]REBUILD;
```

Example

```
hive>ALTERINDEXindex_studentsONstudentsR
EBUILD;
```

```
QueryID=cloudera_20171111093030_6a37b9
2b-bae8-4fd1-91bb-d13e9d411513Total jobs
= 1
```

...

```
TotalMapReduceCPUTimeSpent:4seconds180
msecOK
```

DROPCOMPACTINDEX

```
DROPINDEX[IFEXISTS]index_nameONtable_na
me;
```

Example

```
hive>DROPINDEXIFEXISTSindex_studentsONst
udents;OK
```

```
Timetaken:0.27seconds
```

Bitmap IndexonHiveTable

Combiningindexedcolumnvalueandlistofrowsa
restoredasbitmapinBitmapindexing.

CREATEBITMAP INDEX:

```
CREATEINDEXolympic_index_bitmapONTABLE
olympic(age)
AS 'BITMAP'
```

```
WITHDEFERREDREBUILD;
```

```
ALTERINDEXolympic_index_bitm
```

aponolympicREBUILD;

IndexingAdvantages

1.

Indexesimprovethethequeryperformance

.

2.

MultipleIndexescanbecreatedonthesa
metable.

3.

Anytypeofindexingcanbecreatedonthe
data.

4.

DependingondataBitmapindexesarefa
sterthanCompactindexesandviceversa.

3.

PreliminariesandDefinitions

HDFS: HadoopDistributed File
SystemYARN:YetAnotherResourceNegotiatorD

N: Data Node

NN:NameNode

SN:SecondaryNameNodeRM: Resource
Manager NM:NodeManager

CLI:CommandLineInterfaceUI:UserInterface

OLTP:OnlineTransactionProcessingOLAP:Onlin
eAnalytical Processing

4. SystemModel

Ingeneraltheindexes(CompactIndexandBitmap
Index)areappliedasaseparateentityforperform
anceoptimization.Intheproposedmodelbothth
eindexesareappliedtogetherandtestedonnum
erousdatavalueswhichresultedinmoreperform
anceoptimization.

5.

PerformanceAnalysis

Table1:HiveQuerystatisticswithoutindexing(alltheresultsareinseconds)

Rows/Tables	Table1	Table2	Table3	Table4
50	0.628	0.1	0.085	0.167
100	0.14	0.17	0.087	0.17
150	0.137	0.07	0.0204	0.098
500	0.139	0.181	0.093	0.109
1000	0.251	0.074	0.159	0.108
5000	0.161	0.125	0.089	0.196

6000	0.143	0.225	0.089	0.117
10000	0.162	0.092	0.17	0.097
11000	0.145	0.104	0.085	0.07
11500	0.14	0.098	0.086	0.176

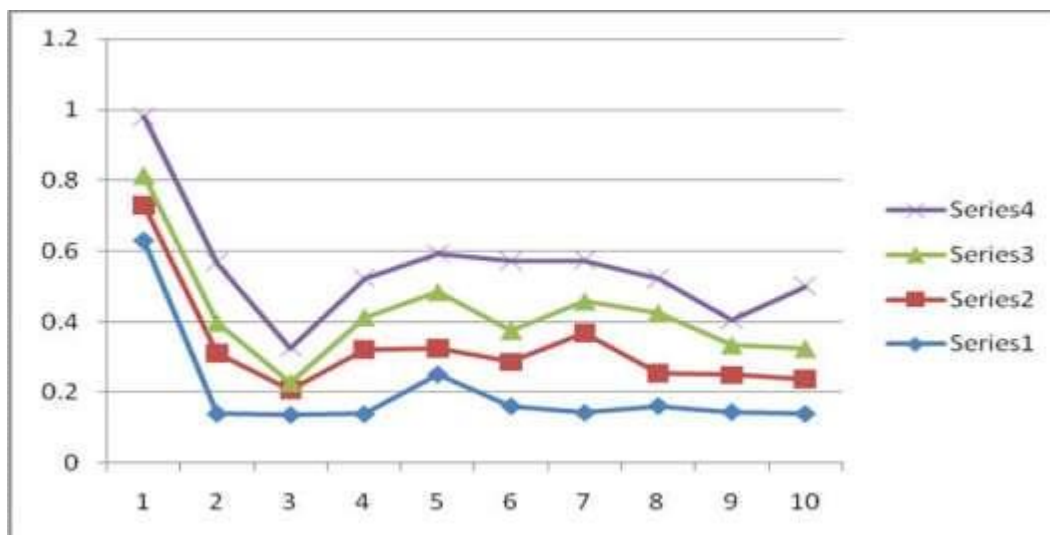


Chart1:HiveQuery statisticswithout indexing(alltheresultsareinseconds)

Table2:HiveQuerystatisticswithBitmapIndexing

Rows/Tables	Table1	Table2	Table3	Table4
50	0.129	0.935	0.085	0.086
100	0.157	0.104	0.095	0.182
150	0.104	0.09	0.181	0.096
500	0.139	0.098	0.154	0.084
1000	0.158	0.097	0.196	0.089
5000	0.123	0.175	0.173	0.096
6000	0.113	0.175	0.081	0.124
10000	0.144	0.108	0.082	0.08
11000	0.169	0.192	0.191	0.089
11500	0.121	0.096	0.095	0.185

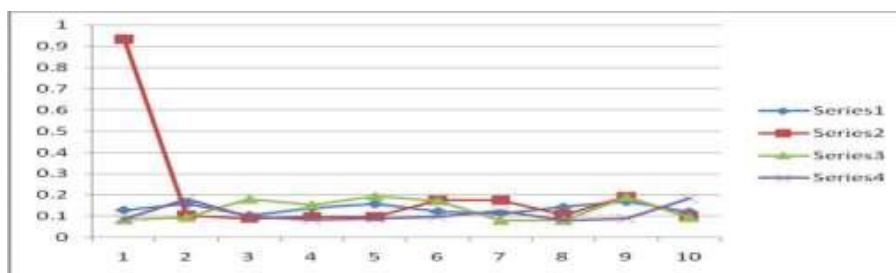
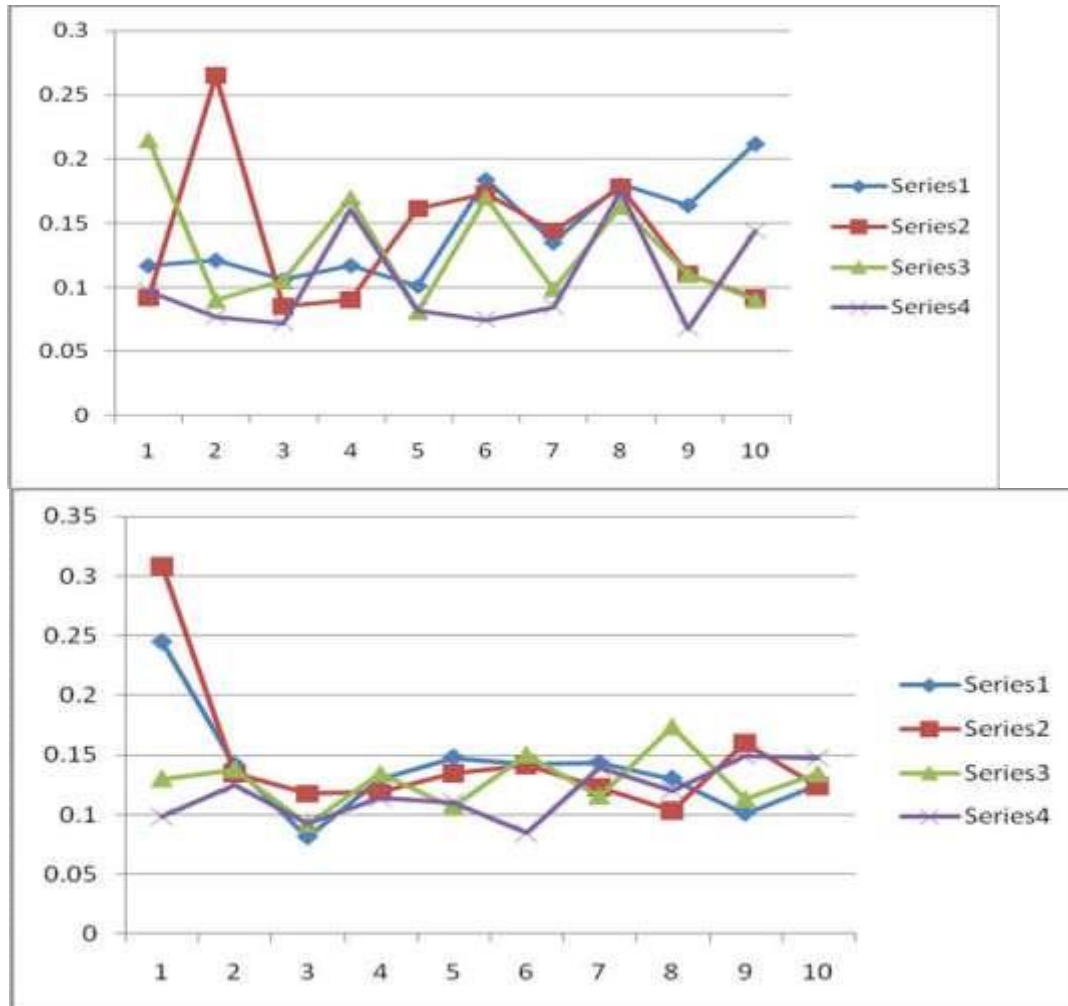


Chart2:HiveQuerystatisticswithBitmapIndexing

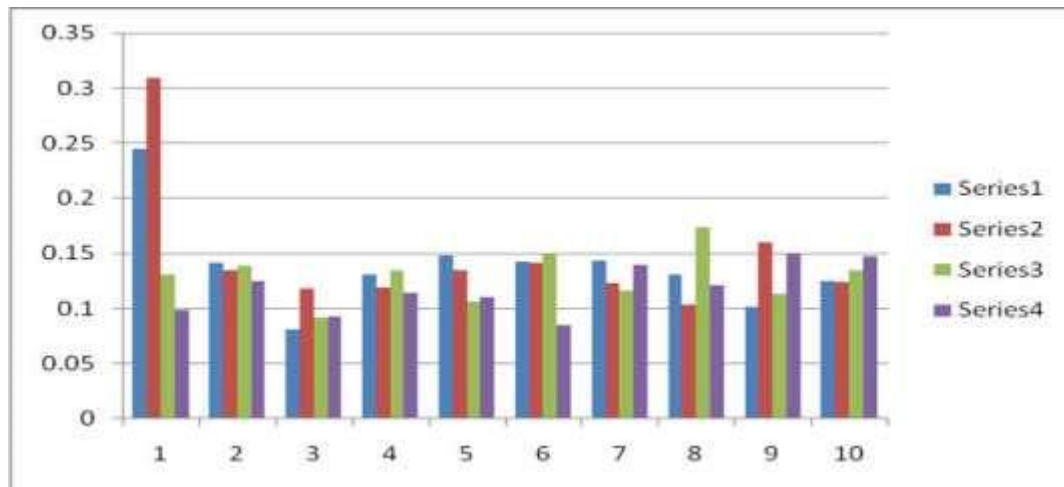
Rows/Tables	Table1	Table2	Table3	Table4
50	0.117	0.092	0.215	0.097
100	0.121	0.265	0.09	0.077
150	0.106	0.085	0.105	0.072
500	0.117	0.09	0.17	0.161
1000	0.101	0.161	0.081	0.082
5000	0.184	0.173	0.17	0.075
6000	0.135	0.144	0.099	0.085
10000	0.18	0.178	0.163	0.175
11000	0.164	0.11	0.11	0.068
11500	0.212	0.092	0.09	0.144

Table3:HiveQuerywithCompactIndexing

z



AverageHiveQuerywithnoindex,Bitmap,CompactandBothBitmapandCompactIndexingtogether



Bar chart) Average Hive Query with no index, Bitmap, Compact and Both Bitmap and Compact Index in go to get her

Note (Bar Chart Legend):

Series 1: No

Index Series 2: Bitmap Index Series 3: Compact Index Series 4: Both Index X – Count of Values Y – Axis Time in Seconds

6. Conclusion and Future Scope

We conclude that Individually Compact and Bitmap indexes work well when it comes to data analysis. It has been shown that using both indexes in Hive Partition for single-dimensional data analysis is the most efficient method. Apache Spark must be used in order to analyze various data sets in the form of audio, video, and image data.

REFERENCES

- [1] The importance of 'big data': A definition, M. A. Beyer and D. Laney, Gartner Technical Report, 2012.
- [2] There is a lot of data out there that can be mined with the help of big data mining, and this is what we're going to focus on in this section of the IEEE Transactions on Knowledge and Data Engineering (TKDE).
- [3] "Mining of huge datasets" by Rajaraman and J. D. Ullman, Cambridge

University Press, 2012.

- [4] Z. Zheng, J. Zhu, and M. R. Lyu [3] are the authors. An Overview of Service-Generated Big Data and Big Data-as-a-Service," in IEEE BigData, p. 403-410, October 2013. To compare the effectiveness of social, collaborative filtering and hybrid recommenders, ACM Transactions on Intelligent Systems and Technology published

an article by Abellogin, Cantador, Dez et al in volume 4, issue 1, January 2013 (pp. 1-37).

[5] In "Can Dissimilar Users Contribute to Accuracy and Diversity of Personalized Recommendation?" by W. Zeng, M. S. Shang, and Q. M. Zhang et al., International Journal of Modern Physics C, vol. 21, no. 10, June 2010, pages 1217-1227.

[6] [5] Apache Hadoop [Online]. [6] Hadoop is currently available at <http://hadoop.apache.org/>.

[7] This paper is entitled "The Hadoop Distributed File System," and it was published in the Proceedings of the IEEE Conference on Mass Storage Systems and Technologies (MSST) in Incline Village, NV, in 2010.

[8] "[Online] Apache Hbase" (version 7) You can access it at <http://hbase.apache.org/>.

[9] (nine) Abouzeid, Bajda-Piwlikowski, D. Abadi, Silberschatz, and Rasin, all of them are from the University of Pennsylvania. Analytical workloads can benefit from Hadoopdb, an architectural combination of mapreduce and dbms technologies. IEE

Transactions on Knowledge and Data Engineering, 2(1):926–933, 2009.

[10][10]

[11] Allan Jindal, Yuri Kargin, Vijay Setty, and J. Schad are listed as co-authors on a study published in the Journal of Clinical Pathology (JCP) in 2009. It's possible to speed up a yellow elephant with Hadoop++ (without it even noticing). 3(1-2):515–529. 2010 VLDB Endowment Proceedings.

[12] [20] <http://dwgeek.com/hive-different-file-formats-text-sequence-rc-avro-orc-parquet-file.html/>
<https://snippetessay.wordpress.com/2015/07/25/hive-optimizations-with-indexes-bloom-filters-and-statistics/>

[13][13]

<http://web.cse.ohiostate.edu/hpcs/WW/HTML/publications/papers/TR-11-4.pdf>

[14]

<https://www.semantiko.com/blog/orc-intelligent-big-data-file-format-hadoop-hive/>

[15] U. F. Minhas, F. Ozcan, and A. Floratou. It's a return to shared-nothing database architectures with SQL on Hadoop. 7(12), 2014, Proceedings of the VLDB Endowment

[16] AshwiniSomnath, A. Thusoo, J. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy are the authors. Hadoop-based data warehouse on the order of a petabyte in size. IEEE 26th International Conference on Data Engineering (ICDE), pages 996–1005. IEEE, 2010.