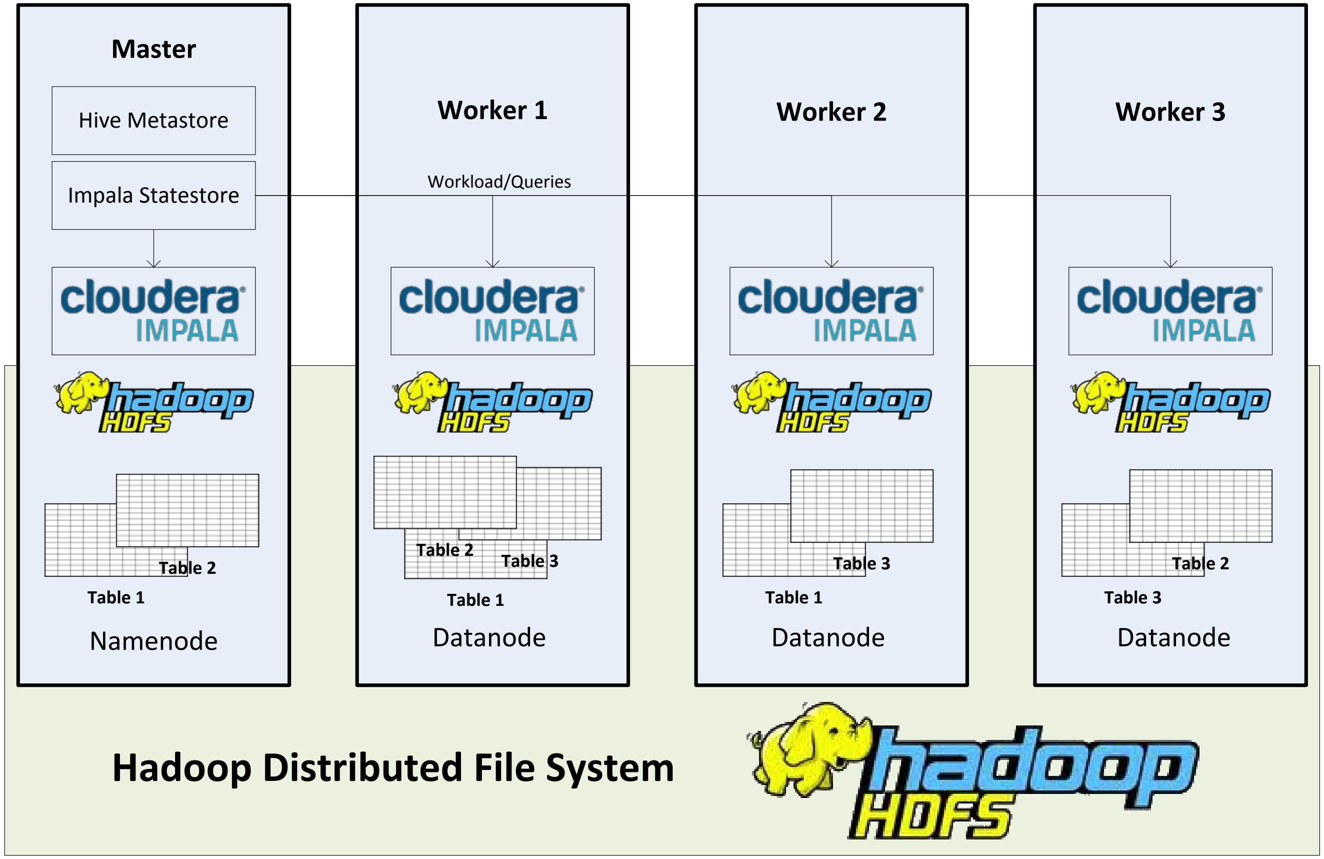
Big Data Query Processing with Mixed Workloads

Melanie Imhof, Jonas Looser, Thierry Musy, Kurt Stockinger

- Open-source Apache project for scalable, fault-tolerant and distributed software
  - Hadoop Common
    - Library for Hadoop modules
  - Hadoop Distributed File System (HDFS)
    - Distributed filesystem
  - Hadoop YARN
    - Job scheduling und cluster management
  - Hadoop MapReduce
    - YARN-based system to process large amounts of data in paralell

- unix-like file system interface
  - copying, deleting and creating files and directories
- internally the data is distributed and replicated

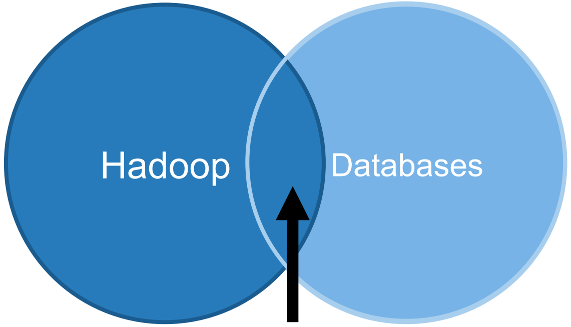

Master
Hive Metastore
Impala Statestore
Workload/Queries
Worker 1
Worker 2
Worker 3
cloudera IMPALA
hadoop HDFS
Table 2
Table 1
Namenode
Table 2
Table 3
Table 1
Datanode
Table 3
Table 1
Datanode
Table 2
Table 3
Datanode
Hadoop Distributed File System
hadoop HDFS

- interacts with the distributed data in the HDFS
- provides an SQL interface
- distributes the workload (execution of queries) to the nodes in the cluster
- master node gathers the computed data and sends back the query response to the user

Impala vs. RDBMS

- Impala
  - Scalable to “Big Data”, since based on Hadoop
  - Distributed data
  - Easy to use
    - SQL interface
    - Automatic generation of MapReduce code
  - Read-only data
  - Useable for real-time query processing

- RDBMS
  - Complex queries
  - Query optimization
  - Easy to use
    - SQL interface
    - Interface to Enterprise Tools (Visualization etc.)
  - Read-Write Data
  - Useable for real-time query processing



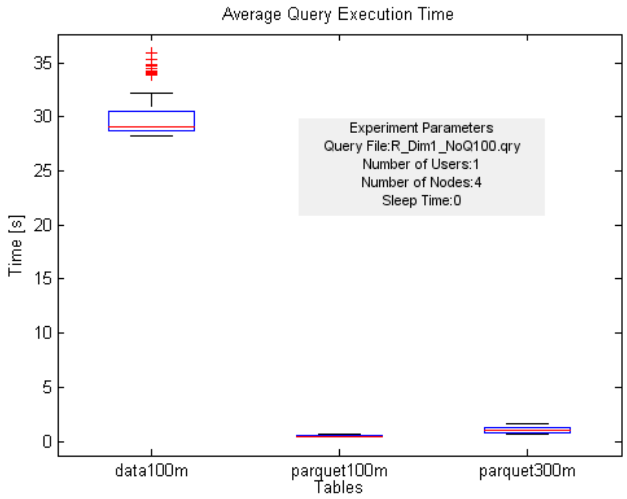
Impala

Test Data

Tables

File and Table Names	Number of Rows	Raw Data Size of CSV-File	Approx. Size in Impala	Storage Format in Impala	Time to Import CSV To HDFS [s]	Time to Create Parquet Table [s]
parquet100k	100‘000	33 MB	14.11 MB	PARQUET	2.55	3.87
parquet1m	1‘000‘000	332 MB	133.73 MB	PARQUET	15.61	7.19
parquet10m	10‘000‘000	3.36 GB	1.30 GB	PARQUET	165.62	37.55
parquet100m	100‘000‘000	33.9 GB	12.98 GB	PARQUET	1130.28	322.31
parquet300m	300‘000‘000	102 GB	38.95 GB	PARQUET	3178.91	825.76
parquet1g	1’000’000’000	339 GB	129.78 GB	PARQUET	-	-

All experiments conducted using the compressed Impala storage “Parquet”



Queries

Query Types	Description	Example

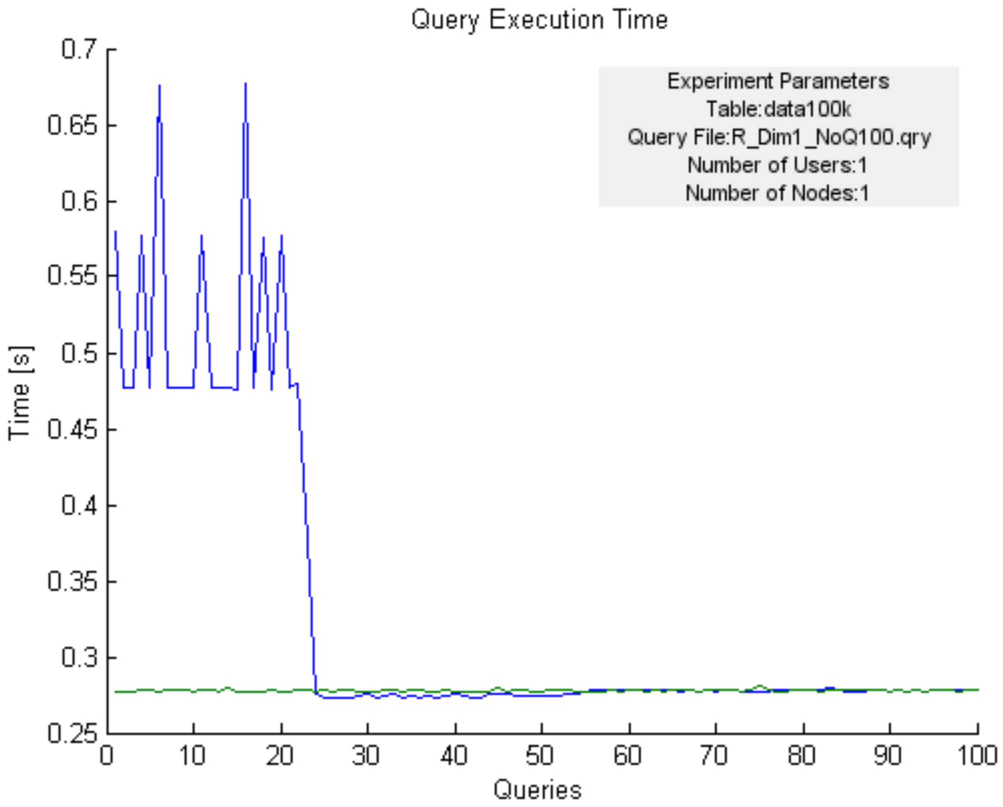
R	Integer and float range queries	SELECT count(*)
		FROM <tableName>
		WHERE A < 27

S	String queries	SELECT count(*)
		FROM <tableName>
		WHERE B LIKE '%ahx%'

G	Group by- queries	SELECT A, count(*)
		FROM <tableName>
		GROUP BY A

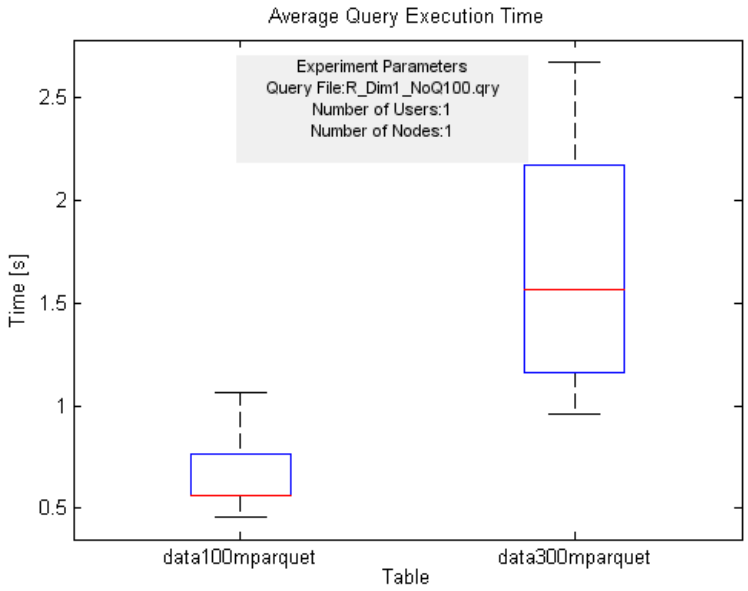
M	Mixed queries including R, S and G queries	SELECT *parse_url(C*, 'HOST') , count(*)
		FROM <tableName>
		WHERE A= 3 AND B LIKE '%86%'
		GROUP BY *parse_url(C*, 'HOST')

Cold Cache vs. Warm Cache

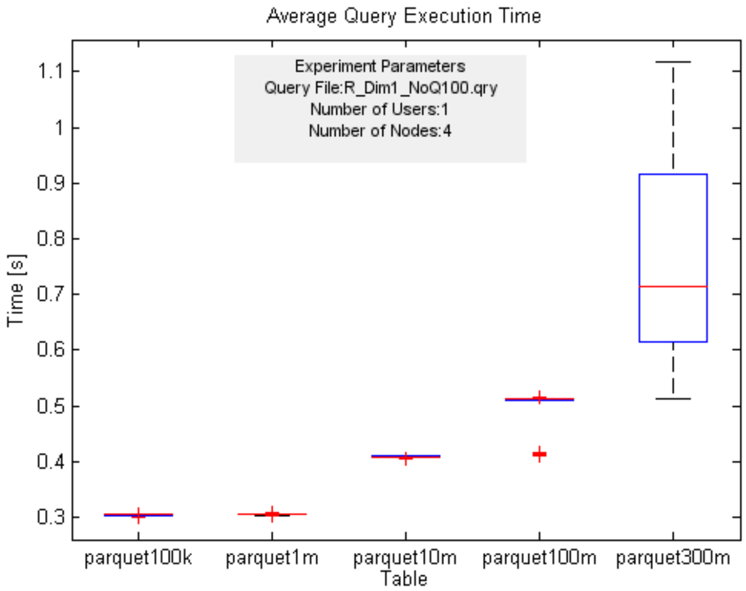


Data Size / Single vs. Multi Node

- Single Node

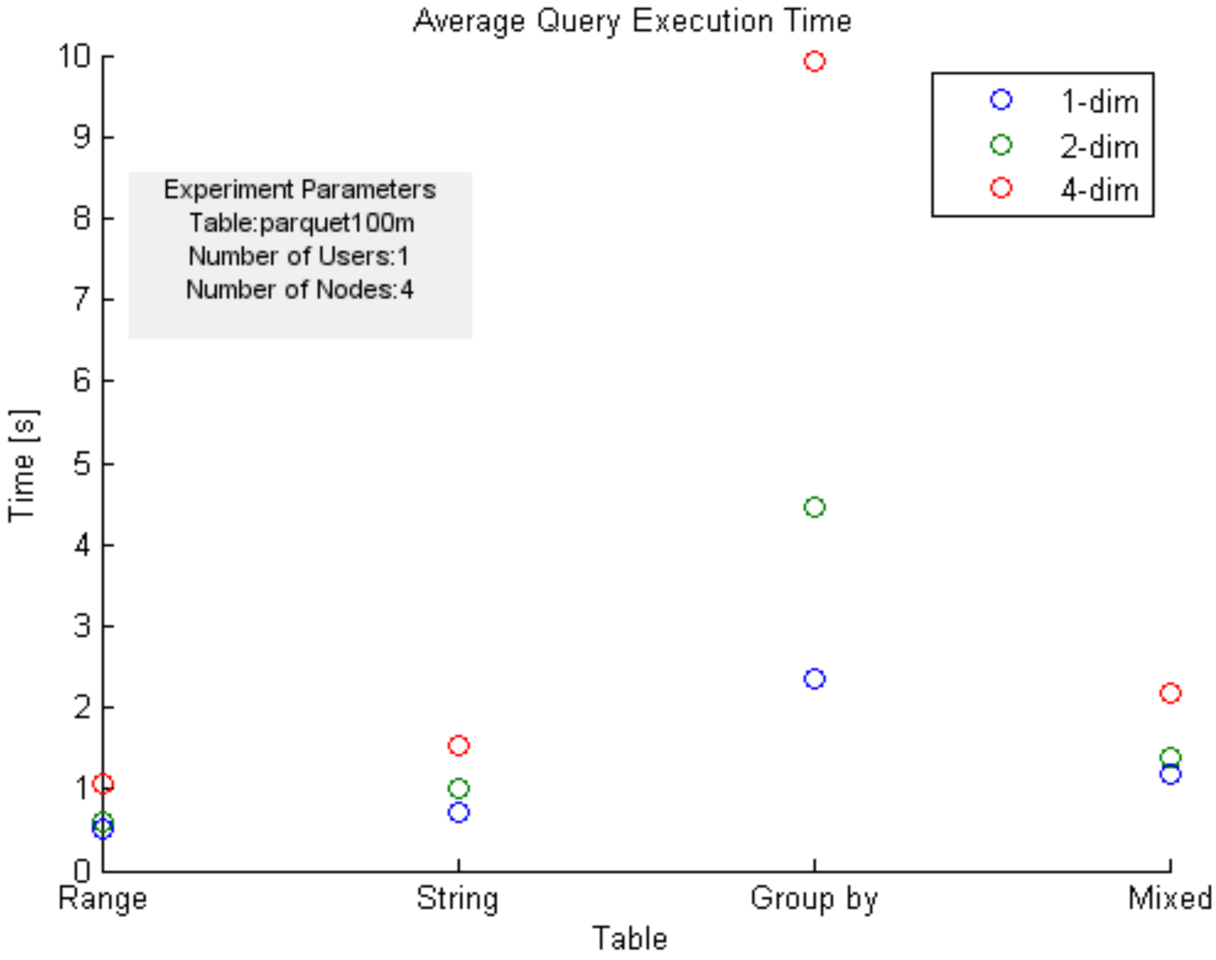


- Multi Node (4 Nodes)

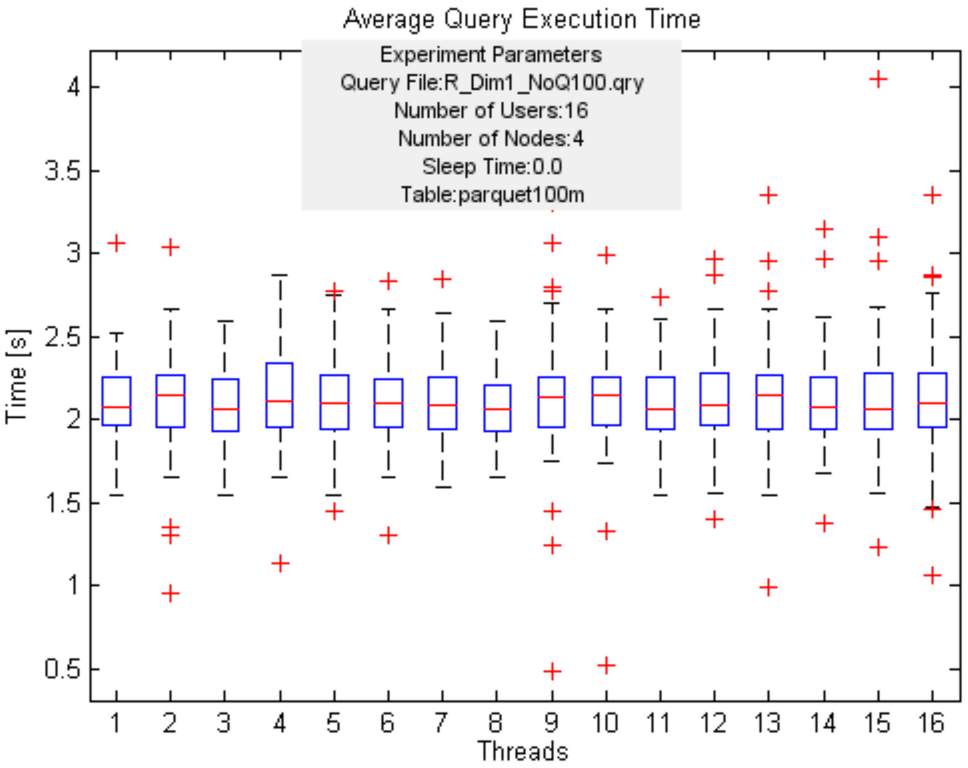


Data Size/Table	Single Node Avg Query Time[s]	Multi Node Avg Query Time[s]
parquet100k	-	0.30
parquet1m	-	0.31
parquet10m	-	0.41
parquet100m	0.67	0.49
parquet300m	1.66	0.75
parquet1g	-	1.79

Query Types

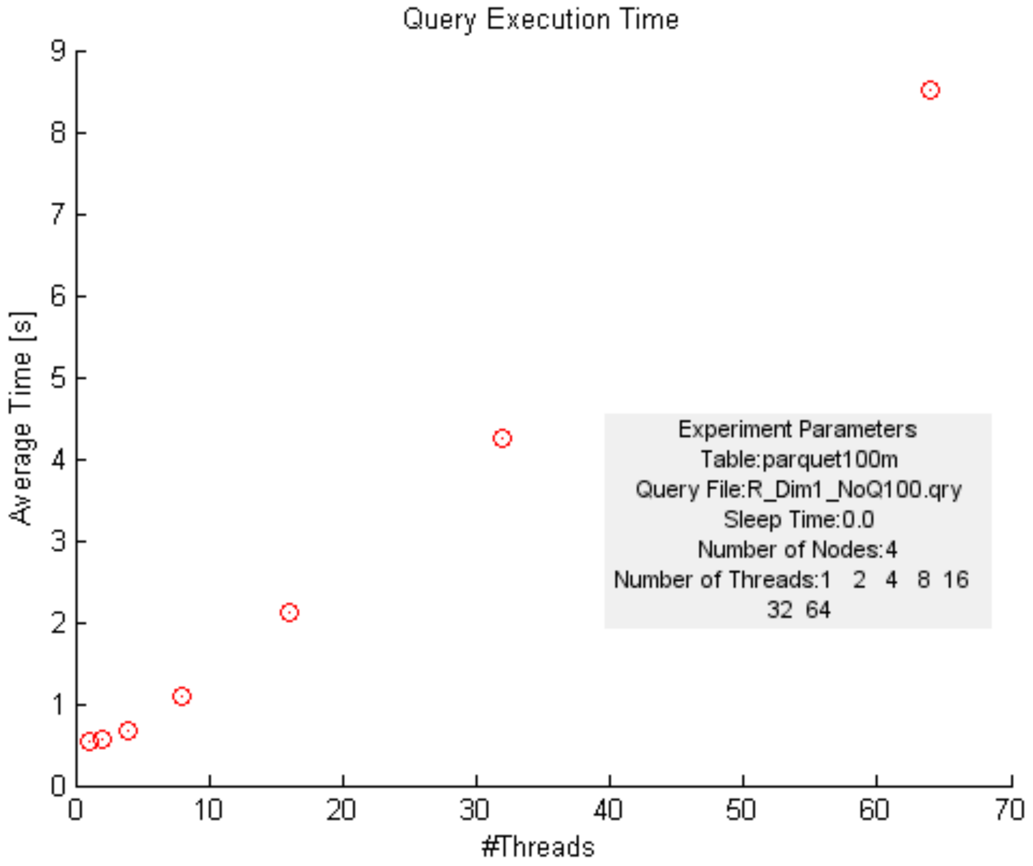

Average Query Execution Time
Time [s]
0
1
2
3
4
5
6
7
8
9
10
Range
String
Group by
Mixed
Table
Experiment Parameters
Table:parquet100m
Number of Users:1
Number of Nodes:4
1-dim
2-dim
4-dim

Multi-User

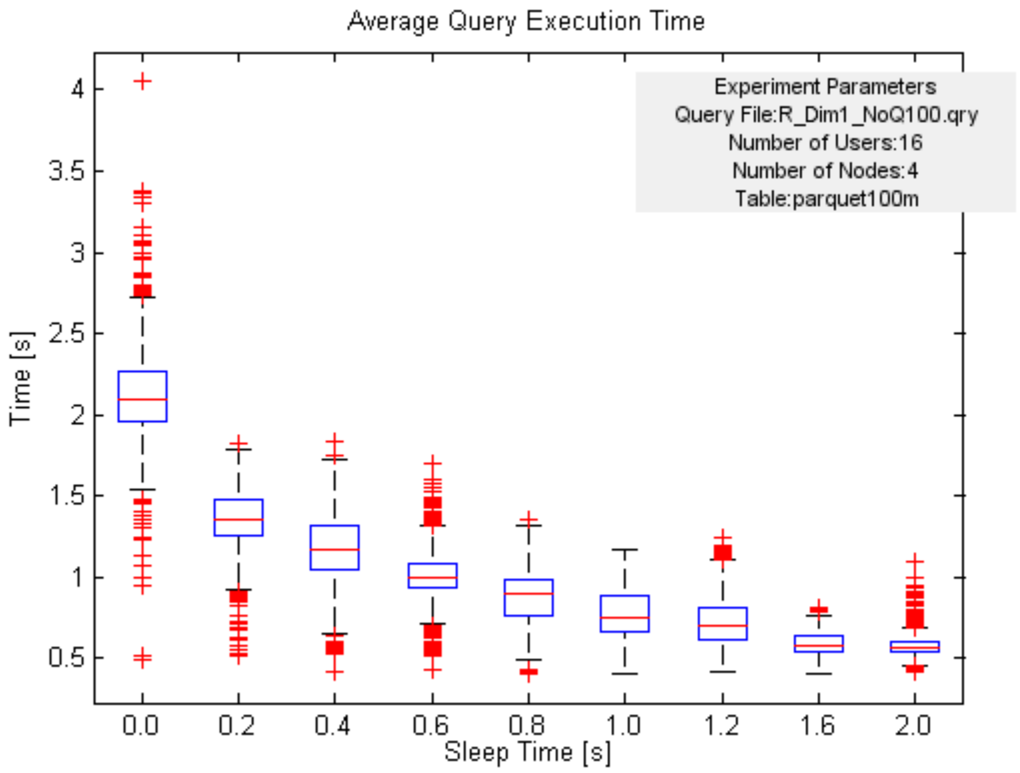


No user is discriminated.
Approximately the same response time (~2 sec) for all users.

Multi User


Query Execution Time
Average Time [s]
#Threads
0
1
2
3
4
5
6
7
8
9
0
10
20
30
40
50
60
70
Experiment Parameters
Table:parquet100m
Query File:R_Dim1_NoQ100.qry
Sleep Time:0.0
Number of Nodes:4
Number of Threads:1 2 4 8 16
32 64

Multi-User (Sleep Time)



Conclusions

- Easily scales up to 300’000’000 rows (~1sec per query)
- Impala supports range, string, group by, order by and paging queries
- Lower bound on query response time ~0.3sec
- Multi-node: 4 nodes are ~2 times faster than 1 node
- Use impala built-in functions with caution
    - Think about splitting strings into separate columns
- Multi-user: No user starving
- Multi-user: Maximal query throughput (8 queries per sec)

- Further information:
  - http://blog.zhaw.ch/datascience/big-data-query-processing-with-mixed-workloads/