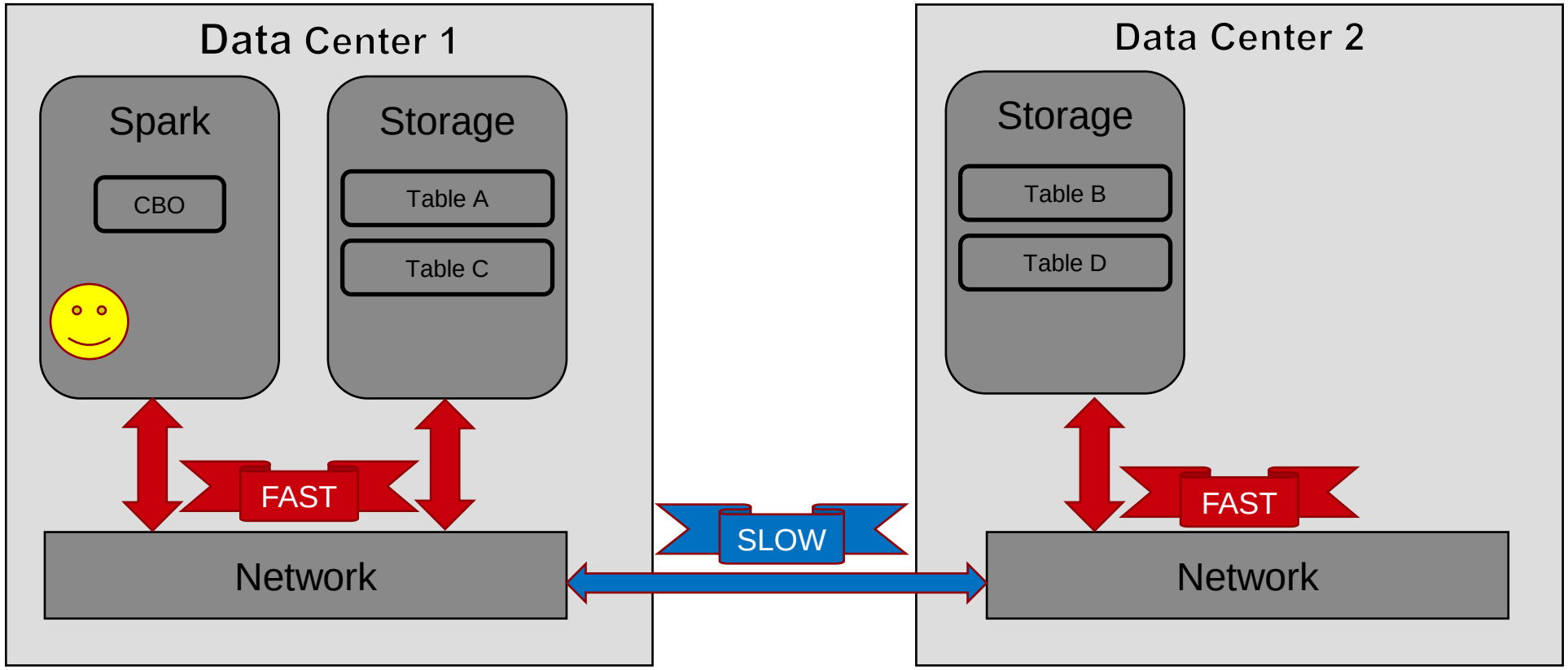
Baseline Investigations

- Goals:
  - Provide the environment and setup for Distributed Datacenters testing.
  - Generate dataset for TPC-DS benchmark
  - Gather results for:
    - TPC-DS benchmark with data located in one Datacenter.
    - TPC-DS benchmark with data distributed across two Datacenters.
- Values:
  - Clearly demonstrate the problem we are trying to solve.
  - Produce measurable performance and data transfer analyses.
  - Full featured simulation of two Distributed Datacenters running on commodity hardware.
  - Simulation can be used to analyze and debug customer use cases.
  - Simulation can be used for customer demo.
  - Provide augmented representation of data location to Spark via Spark Catalog connected to QFlock customized Metastore.

Baseline Investigations

- Spark Investigations
  - Spark/Hadoop has an issue generating queries across a single file/table split across two Datacenters.
  - Our solution will split half of the files/tables (approximately half of the data) between two Datacenters.
- Baseline Configuration
  - Data center 1 (dc1)
    - Contains spark-dc1 and storage-dc1 (hdfs).
  - Data center 2 (dc2)
    - Contains storage-dc2 (hdfs).
  - Baseline test 1 (All data is located on dc1)
    - User uses Spark to issue TPC-DS benchmark using tables all stores in hdfs on dc1.
  - Baseline test 2 (Data is split between dc1 and dc2)
    - User uses Spark to issue TPC-DS benchmark with half of the tables on dc1 and the other half on dc2.

Performance test setup



Size Estimator

- Goal: estimate amount of data transferred.
- Value: Short term this allows us to estimate the benefits that our Federated Query solution will provide
- Value: Longer term this can be used as the basis for a query time estimator, also very valuable in a Federated Query environment.
- Deliverables
  - Investigated Spark Explain
    - Found that the estimated size is not useful for predicting I/O transfer bytes. Designed to measure Spark Object memory sizes.
  - Implemented custom stats in QFlock Metastore.
    - Stats will be generated by QFlock Metastore and retrieved by our estimator.
  - Implemented a QFlock Estimator
    - Using QFlock Spark Custom Extension
    - Using our QFlock Metastore generating custom statistics
    - Producing expected transfer size.
    - Considers the entirety of the query (filter, project), and considers which data needs to be transferred for a columnar file format.
    - Takes compression into account (parquet is compressed).
- Actual performance
  - We measured the QFlock Estimator accuracy with project operations using TPC-DS data set.
  - QFlock estimated transfer bytes were within 1% of the measured network transfer bytes.

QFlock Size Estimator Results

- Estimated sizes for project on average are within 1% of the measured network transfer bytes

Columns	QFlock Estimated Bytes	Network Bytes	Difference
1	1094411	1203476	9.06%
2	2419224	2490973	2.88%
3	7891279	7971850	1.01%
4	9561695	9665867	1.08%
5	12124920	12233986	0.89%
6	13248131	13449807	1.50%
7	14601744	15026898	2.83%
8	15264150	15408871	0.94%
9	18633783	18751671	0.63%
10	20621003	20701509	0.39%
11	23299429	23411943	0.48%
12	28425880	28549775	0.43%
13	33926735	34035714	0.32%
14	39398790	39509316	0.28%
15	44035636	44113367	0.18%
16	58378180	58451868	0.13%
17	72922326	72992092	0.10%
18	87812075	87875258	0.07%
19	94119338	94169128	0.05%
20	98756184	98835872	0.08%
21	113069928	113103855	0.03%
22	127700475	127949965	0.19%

More Details

Size Estimator Tasks

Spark Default Explain Estimates

Spark supports the ability to view the expected size of any given query. This is supported through the “explain” functionality. For example, if you issue the SQL command: EXPLAIN COST <query>, where <query> is any valid query for a table with statistics, then Spark will produce the Logical Plan for the query along with estimates for information such as number of bytes and rows.

Unfortunately, this estimated size is not very useful. This size is valid in some situations if we consider the uncompressed in memory size of the Java/Scala objects. The size also only considers the size of each individual object in memory and has no relation to the total amount of data that needs to be transferred.

QFlock Size Estimator

We have created a custom Spark extension which estimates the total cumulative size in bytes to be transferred by this plan. The estimates take into consideration a variety of factors:

Since our files are compressed (Parquet), we take into consideration the compressed size of the data. This compressed size for a query is calculated using additional metadata that our metadata server calculates. Our custom Spark extension gathers this information from our metadata server in order to estimate each query.
The estimator considers the entirety of the query, meaning that it estimates exactly which data needs to be transferred considering project and filter operations. For example, a filter operation needs to transfer all the rows for any columns that reside in the query. Whereas a project above a filter only transfers the rows which match the given filter.
We estimate both the default number of bytes Spark needs to transfer and the number of bytes that need to be transferred with pushdown enabled. This allows us to estimate the savings of our solution which will utilize pushdown to a remote data center via a remote Spark cluster.

In the near term, the size estimator allows modeling of the amount of data to be transferred by each query. This is very valuable since depending upon the data transfer rate, we will be able to estimate the transfer time for both local transfers within a data center and remote transfers which span data centers. This in turn will allow us to estimate the benefits of our Spark solutions which allow for Federated Queries across data centers.

Longer term we can envision the size estimator being used as the basis for a query time estimator, which will prove to be invaluable in a Federated Environment, where users will want to check the query time prior to executing long running Federated Queries.

Demonstration

We will now show an example. The query here is executed across the TPC-DS data set and the SQL Query is:

```
SELECT ss_list_price FROM store_sales WHERE ss_sales_price > 100
```

Spark Explain

The Spark version of explain shows the size of each individual object that makes up the Spark Logical Plan.

```
Project [ss_list_price#17, ss_quantity#15L], Statistics(sizeInBytes=4.3 MiB, rowCount=1.89E+5)

+- Filter (isnotnull(ss_sales_price#18) AND (ss_sales_price#18 > 100.0)), Statistics(sizeInBytes=34.5 MiB, rowCount=1.89E+5)

   +- Relation
tpcds.store_sales[ss_sold_date_sk#5L,ss_sold_time_sk#6L,ss_item_sk#7L,ss_customer_sk#8L,ss_cdemo_sk#9L,ss_hdemo_sk#10L,ss_addr_sk#11L,ss_store_sk#12L,ss_promo_sk#13L,ss_ticket_number#14L,ss_
quantity#15L,ss_wholesale_cost#16,ss_list_price#17,ss_sales_price#18,ss_ext_discount_amt#19,ss_ext_sales_price#20,ss_ext_wholesale_cost#21,ss_ext_list_price#22,ss_ext_tax#23,ss_coupon_amt#24
,ss_net_paid#25,ss_net_paid_inc_tax#26,ss_net_profit#27] parquet, Statistics(sizeInBytes=527.3 MiB, rowCount=2.88E+6)
```

QFlock Explain

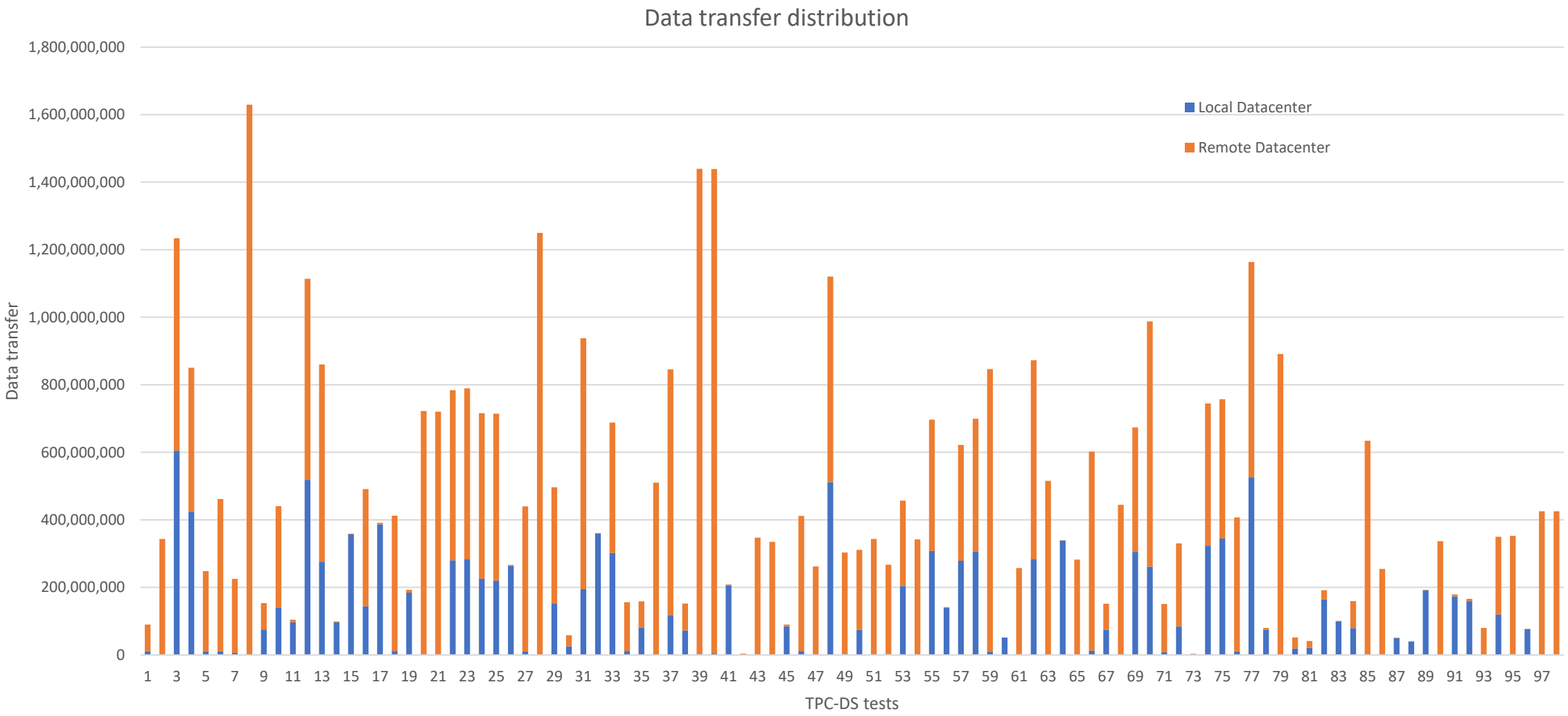
Our version of explain shows the overall size of transfer needed for parquet (compressed), as well as the expected size if pushdown is employed.

```
QflockLogicalRelation size:5969324 pushdownSize:526070  tpcds.store_sales[ss_list_price#17,ss_quantity#15L] QFlockRelation(2 ss_list_price,ss_quantity), Statistics(sizeInBytes=4.3 MiB,
rowCount=1.89E+5)
```

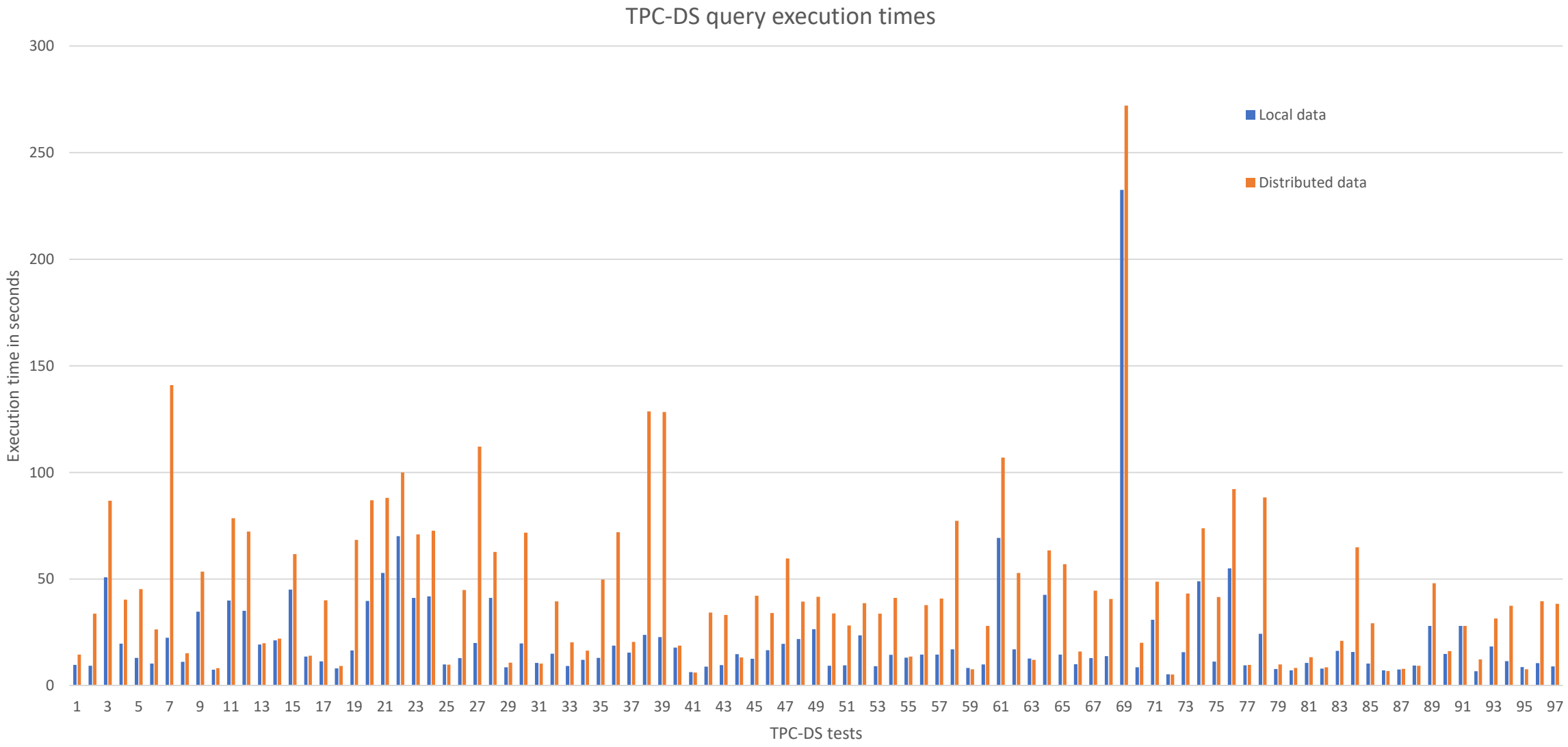
Metastore and Spark Catalog

- Spark can’t use Metastore @Catalog
  - Spark version of Metastore connector 2.3.9 does not have Catalog support at API level
  - QFlock will provide custom solution for Catalog
- QFlock Metastore proxy
  - Thrift location: https://github.com/apache/hive/blob/master/standalone-metastore/metastore-common/src/main/thrift/hive_metastore.thrift
  - Clear way to augment Spark access to Metastore
- NDP style statistics calculation
  - File size property added to Metastore Table
  - Column specific compression ratio property added to Storage Descriptor property of a Table
- Augmented data location
  - Ability to point Spark to another storage location

Data transfer distribution



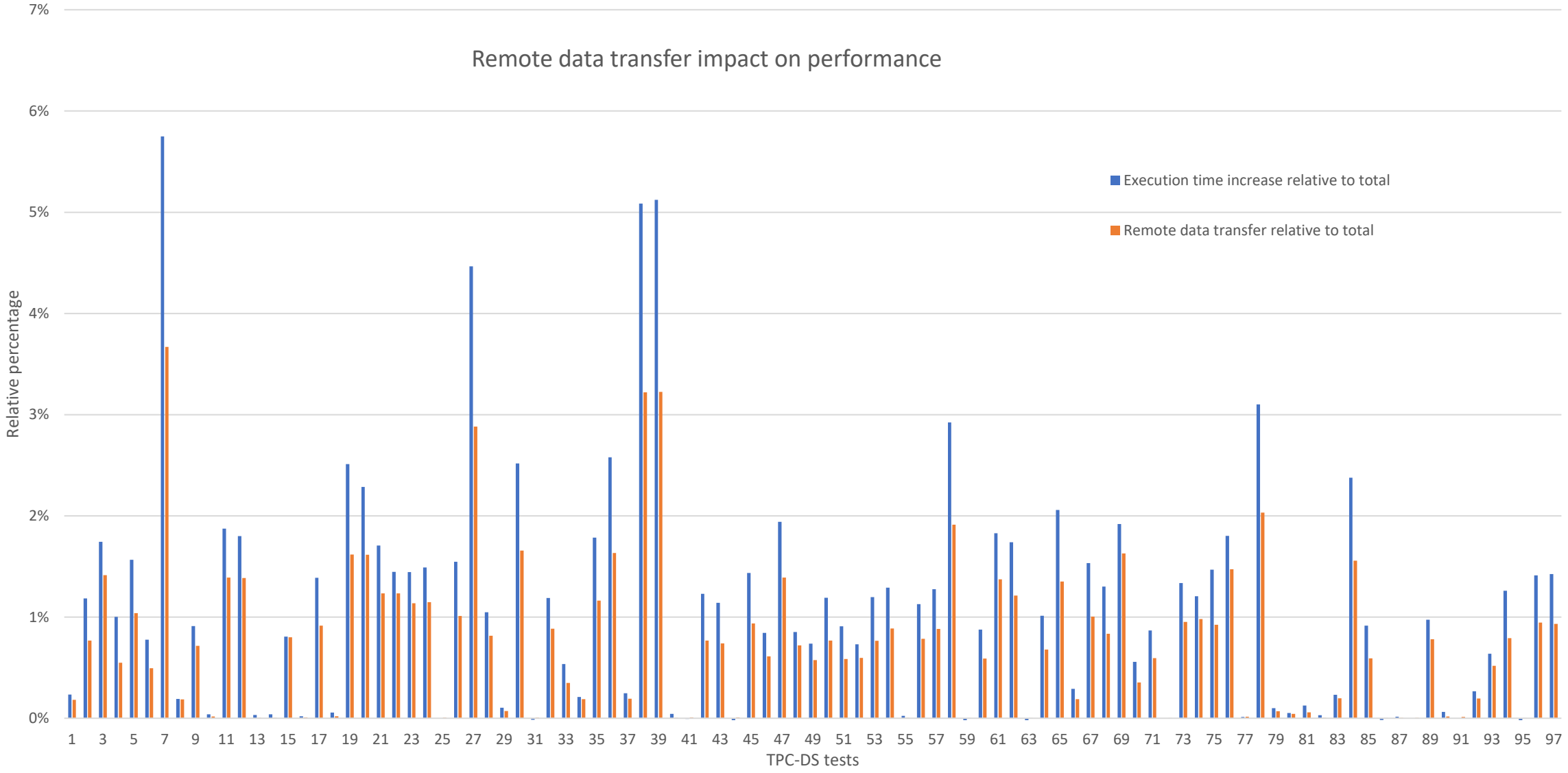
TPC-DS query execution times



Remote data transfer impact on performance

- In the next slide we will demonstrate correlation between amount of data accessed remotely and increase of execution time of a particular query.
- Relative execution time increase was computed as following:
  - ((Test time with remote data) **–** (test time with local data)) ***/*** (Total execution time of all tests)
- Remote data transfer was calculated as following:
  - (Test remote data transfer) / (Total data transfer for all tests)
- It is important to understand that amount of remote data may not always impact test execution time. There are other factors at play, such as computational bottlenecks, etc.

Remote data transfer impact on performance



References

- Spark issue to bring link support
  - [SPARK-32432] Add support for reading ORC/Parquet files with SymlinkTextInputFormat - ASF JIRA (apache.org)
- Github
  - https://github.com/open-infrastructure-labs/QFlock
- Metastore
  - https://github.com/apache/hive/blob/master/standalone-metastore/metastore-common/src/main/thrift/hive_metastore.thrift

Thank You!