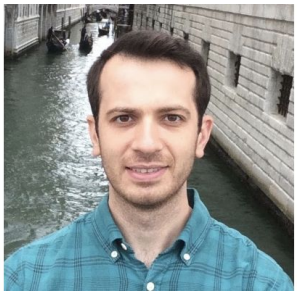
Dynamic Partition Pruning in Apache Spark

Bogdan Ghit and Juliusz Sompolski

Spark + AI Summit, Amsterdam

databricks

About Us



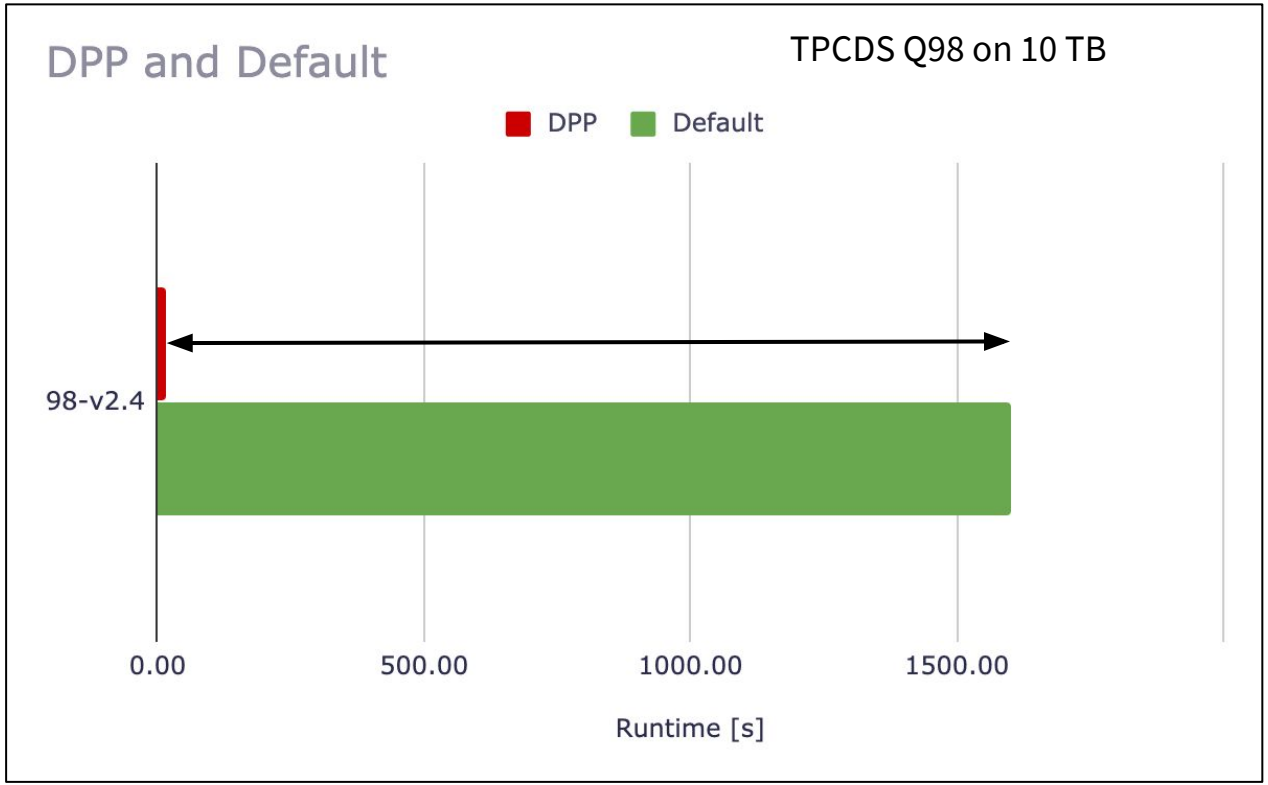
Bogdan Ghit

Juliusz Sompolski

BI Experience team in the
Databricks Amsterdam European Development Centre

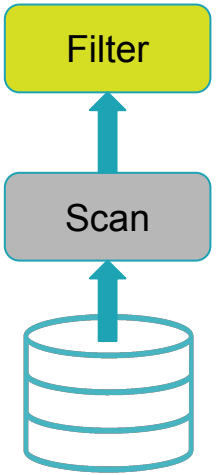
- Working on improving the experience and performance of Business Intelligence / SQL analytics workloads using Databricks
  - JDBC / ODBC connectivity to Databricks clusters
  - Integrations with BI tools such as Tableau
  - But also: core performance improvements in Apache Spark for common SQL analytics query patterns

How to Make a Query 100x Faster?

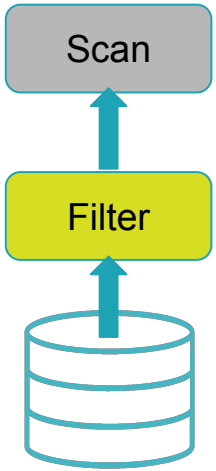


Static Partition Pruning

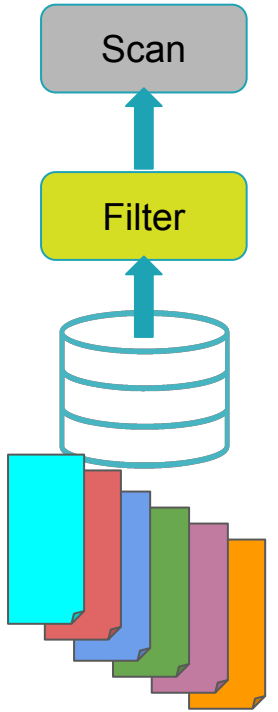
```
SELECT * FROM Sales WHERE day_of_week = 'Mon'
```



Basic data-flow



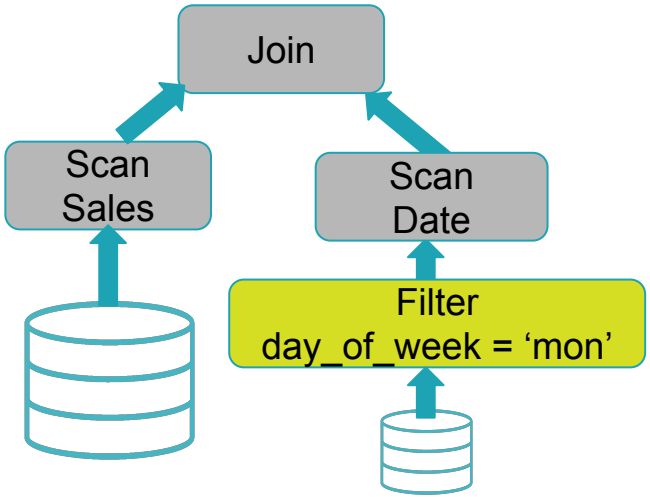
Filter Push-down



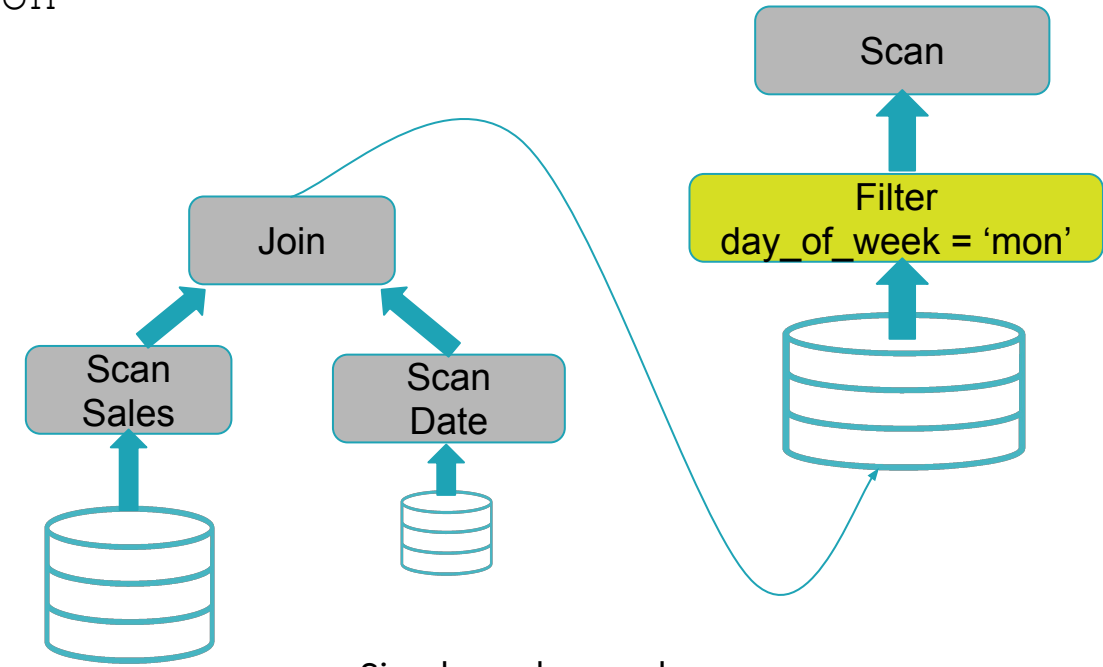
Partition files with multi-columnar data

Table Denormalization

```
SELECT * FROM Sales JOIN Date
WHERE Date.day_of_week = 'Mon'
```



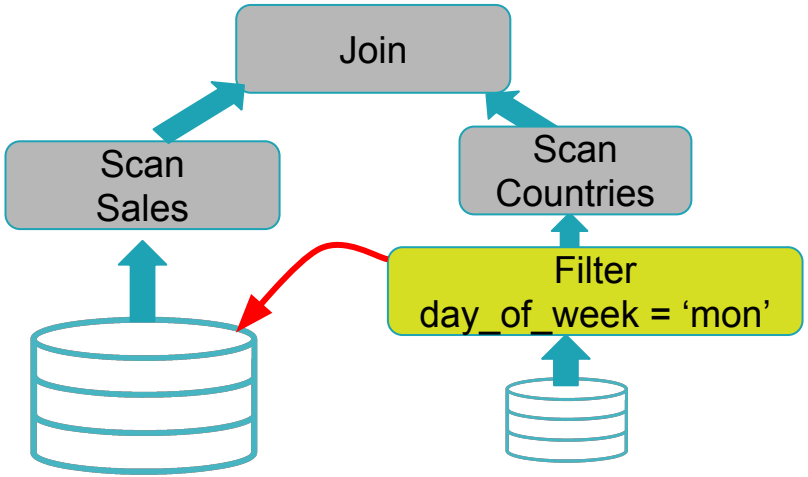
Static pruning not possible



Simple workaround

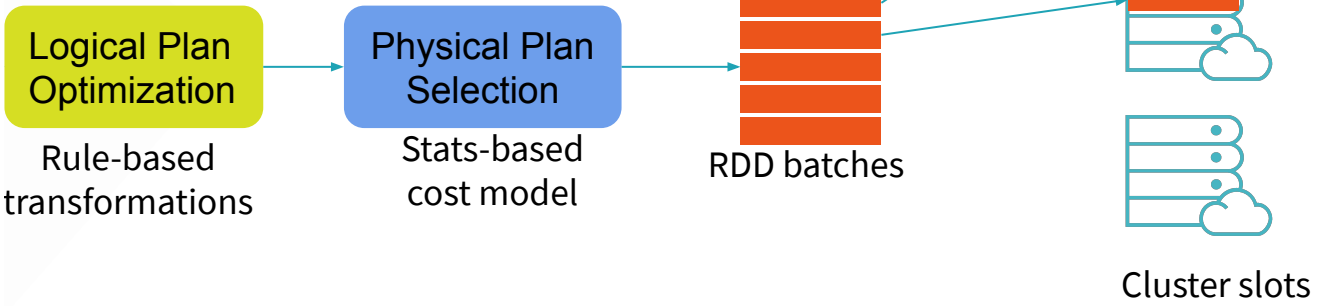
This Talk

```
SELECT * FROM Sales JOIN Date
WHERE Date.day_of_week = 'Mon'
```



Dynamic pruning

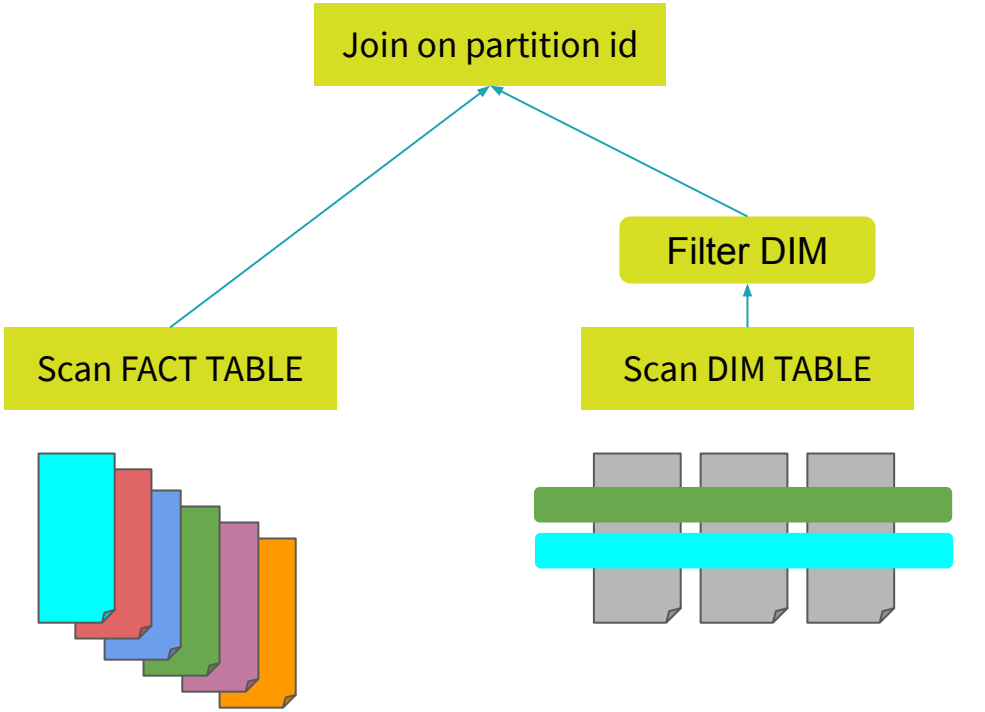
Spark In a Nutshell



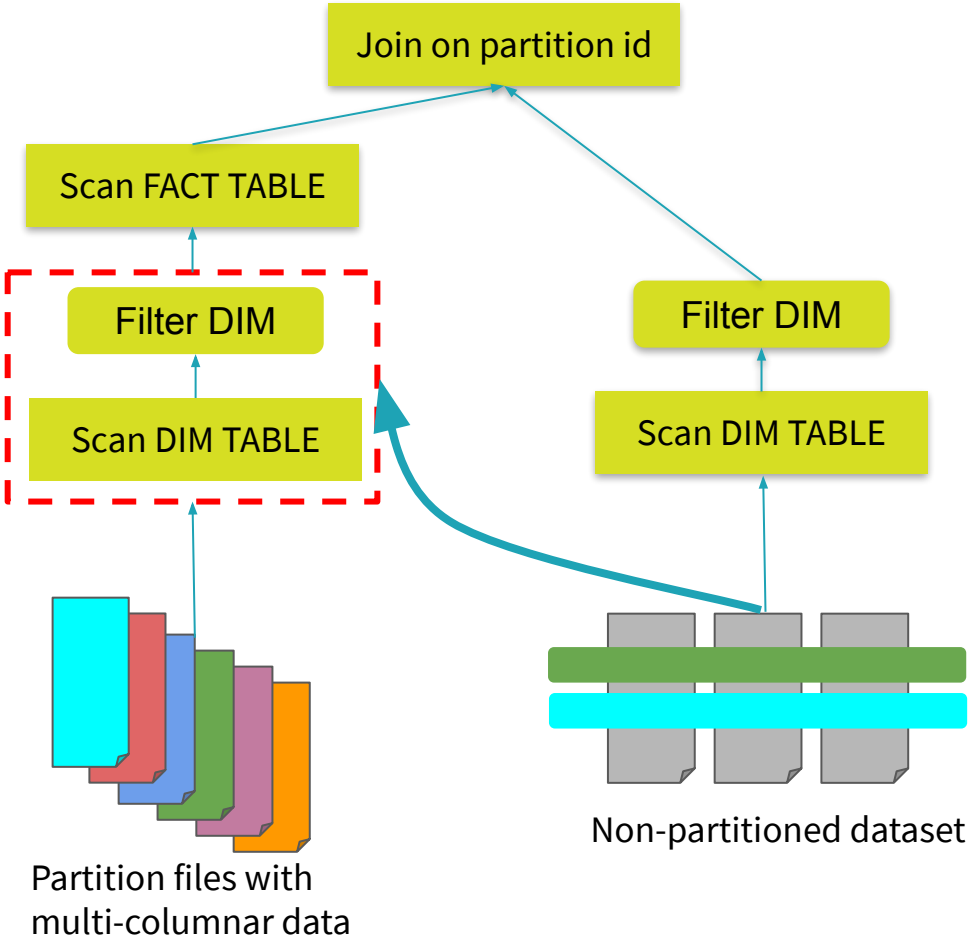
Optimization Opportunities

Query Shape

Data Layout



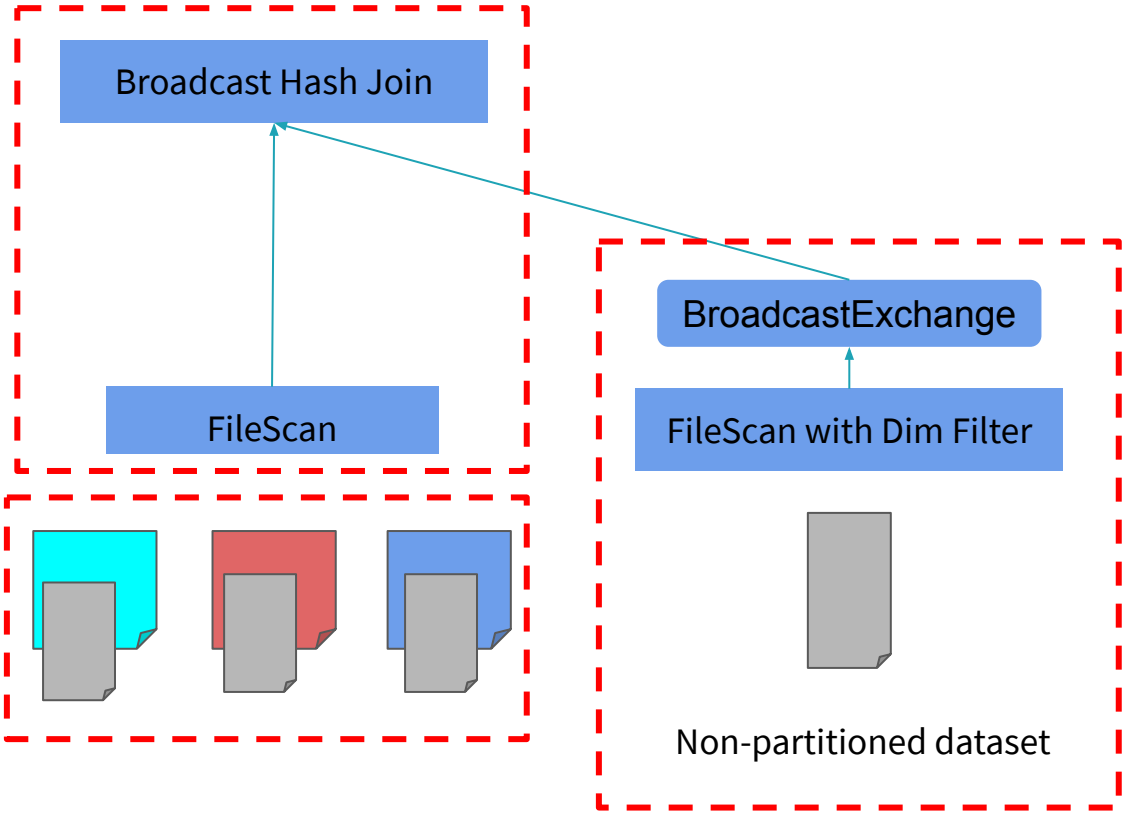
A Simple Approach



Work duplication may be expensive

Heuristics based on inaccurate stats

Broadcast Hash Join



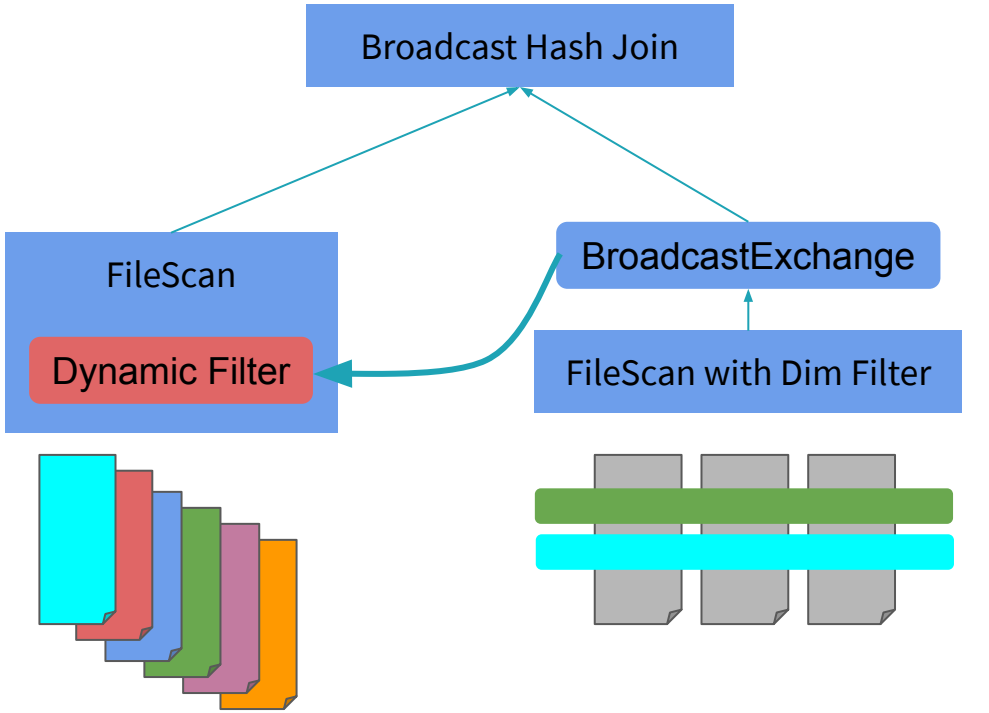
Execute the join locally without a shuffle

Broadcast the build side results

Execute the build side of the join

Place the result in a broadcast variable

Reusing Broadcast Results



Experimental Setup

Workload Selection

- TPC-DS scale factors 1-10 TB

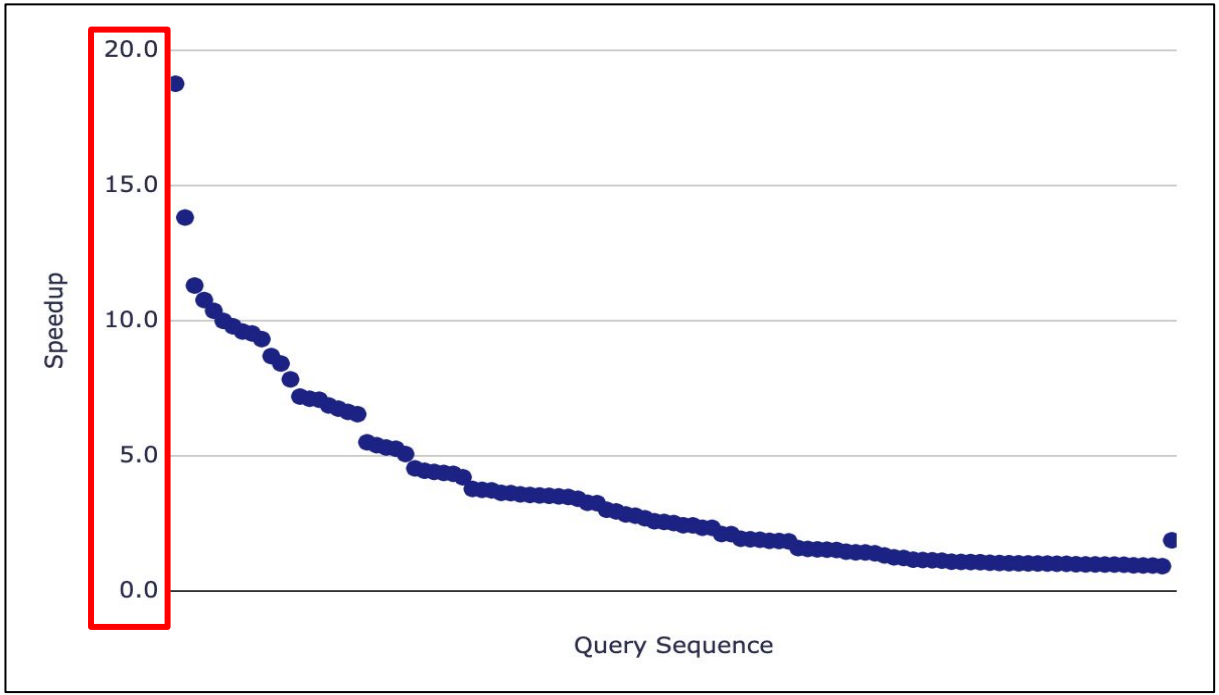
Cluster Configuration

- 10 i3.xlarge machines

Data-Processing Framework

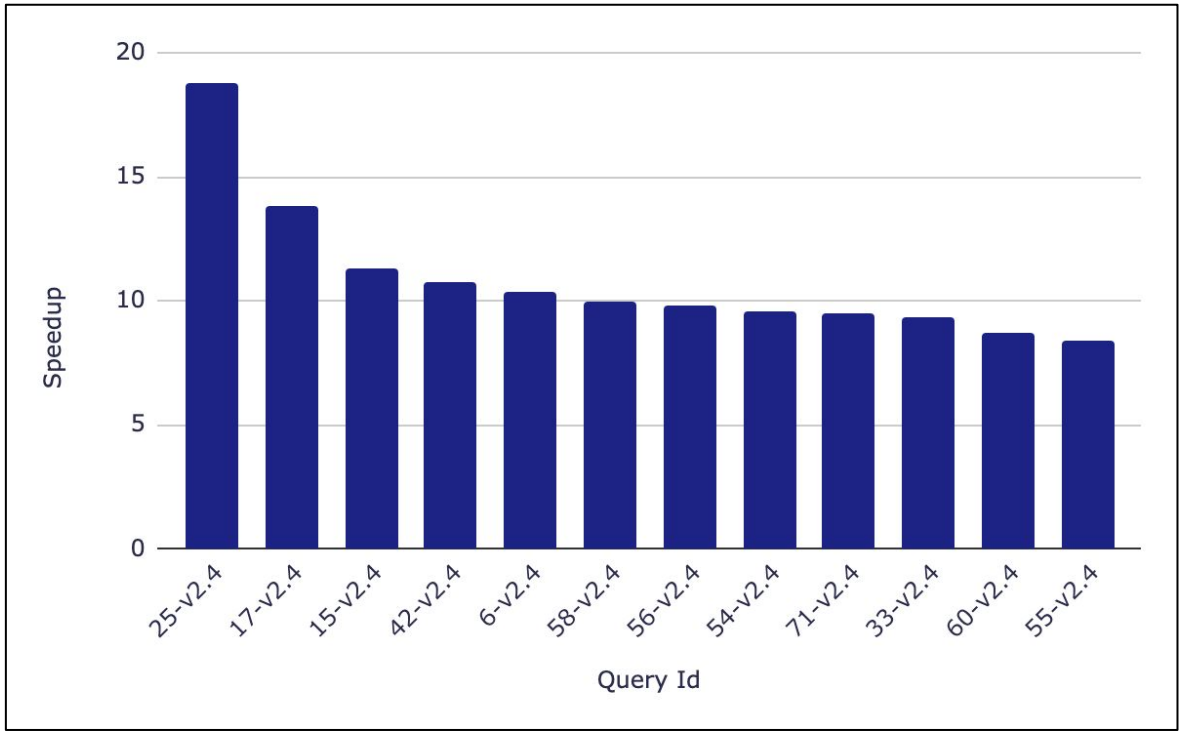
- Apache Spark 3.0

TPCDS 1 TB



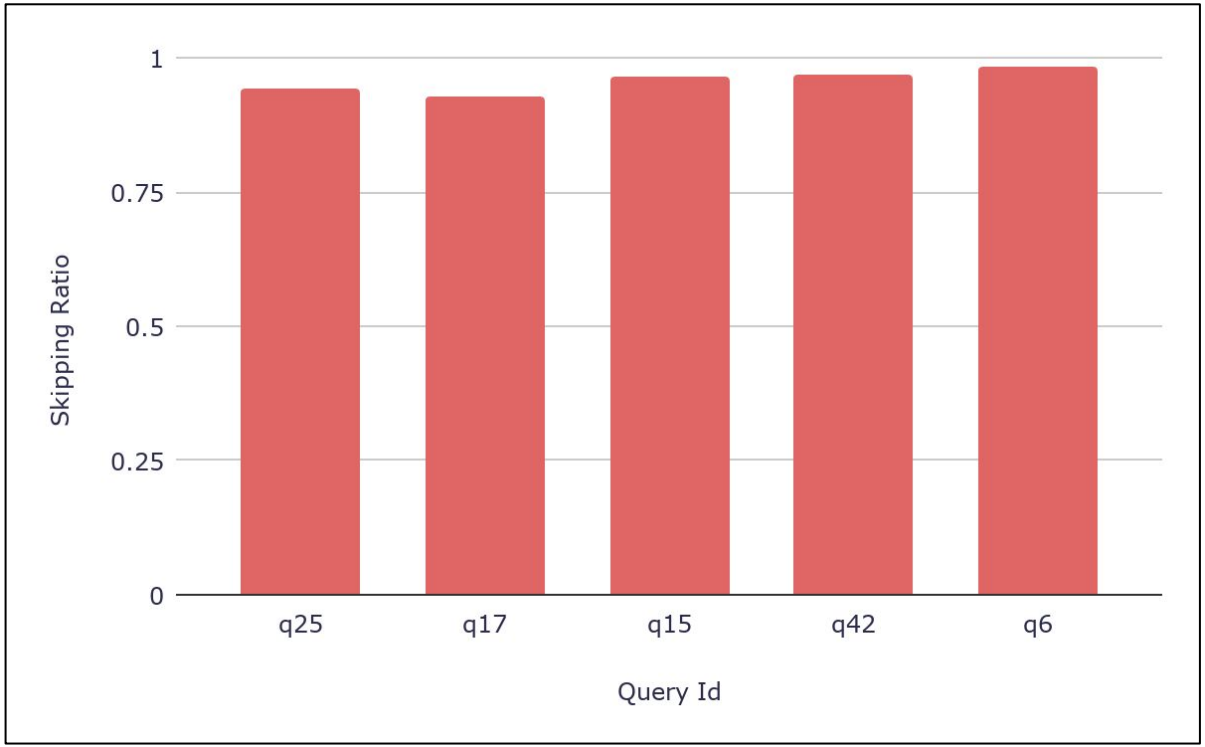
60 / 102 queries speedup between 2 and 18

Top Queries



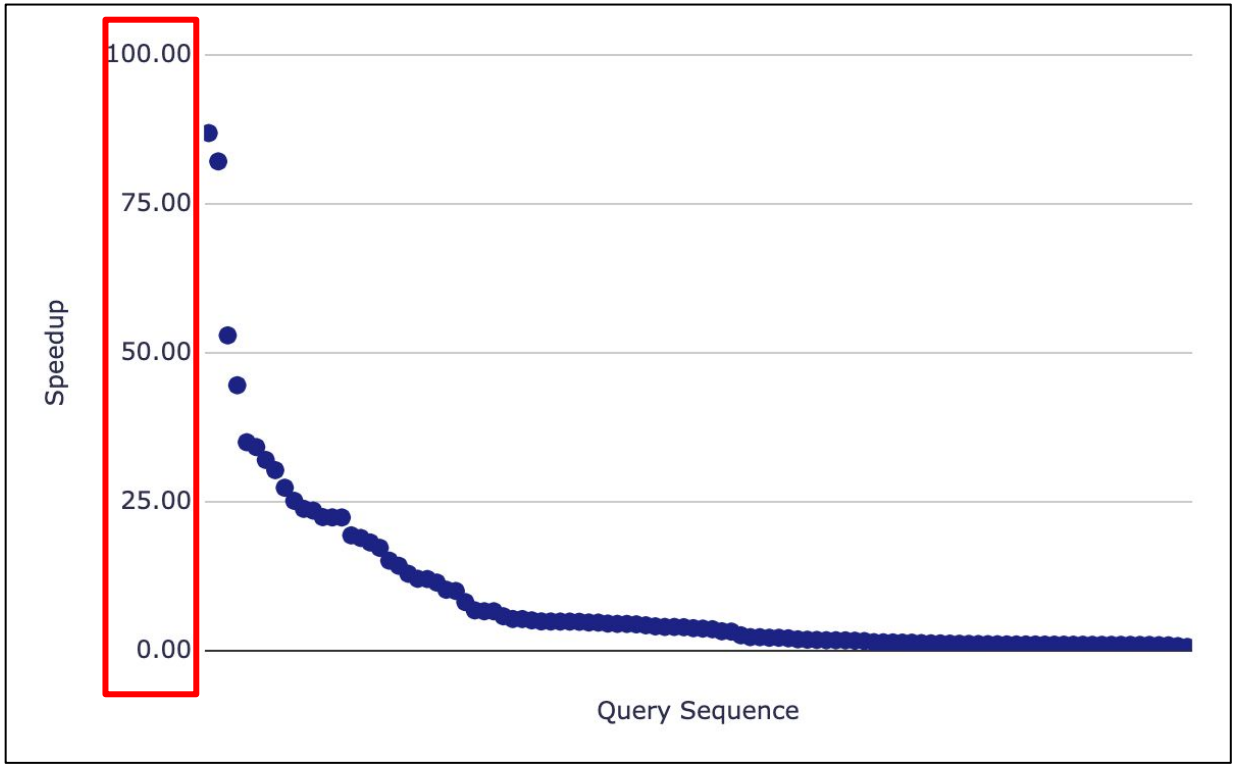
Very good speedups for top 10% of the queries

Data Skipped



Very effective in skipping data

TPCDS 10 TB



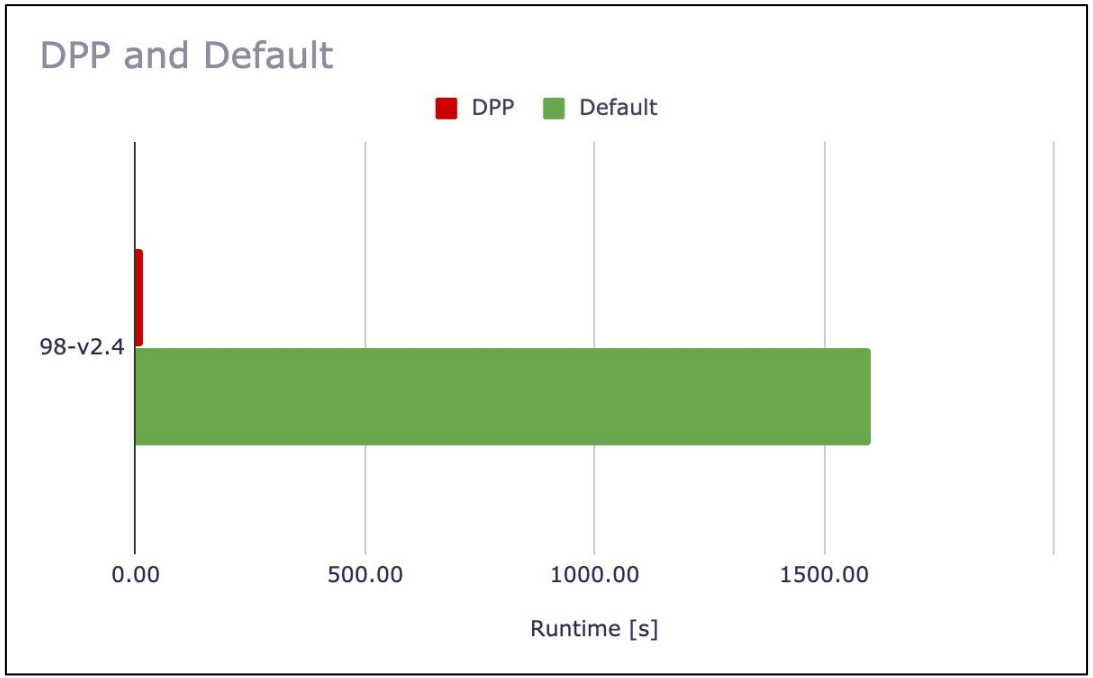
Even better speedups at 10x the scale

Query 98

```
SELECT i_item_desc, i_category, i_class, i_current_price,
       sum(ss_ext_sales_price) as itemrevenue,
       sum(ss_ext_sales_price)*100/sum(sum(ss_ext_sales_price)) over
         (partition by i_class) as revenueratio
FROM
   store_sales, item, date_dim
WHERE
  ss_item_sk = i_item_sk
  and i_category in ('Sports', 'Books', 'Home')
  and ss_sold_date_sk = d_date_sk
  and cast(d_date as date) between cast('1999-02-22' as date)
            and (cast('1999-02-22' as date) + interval '30' day)
GROUP BY
  i_item_id, i_item_desc, i_category, i_class, i_current_price

ORDER BY
  i_category, i_class, i_item_id, i_item_desc, revenueratio
```

TPCDS 10 TB



Highly selective dimension filter that retains only one month out of 5 years of data

Conclusion

Apache Spark 3.0 introduces Dynamic Partition Pruning

- Strawman approach at logical planning time
- Optimized approach during execution time

Significant speedup, exhibited in many TPC-DS queries

With this optimization Spark may now work good with star-schema queries, making it unnecessary to ETL denormalized tables.

Thanks!

Bogdan Ghit - linkedin.com/in/bogdanghit

Juliusz Sompolski - linkedin.com/in/juliuszsompolski

databricks