Strata Data 2019
San Francisco, CA

Tuning Performance for SQL-on-Anything Analytics

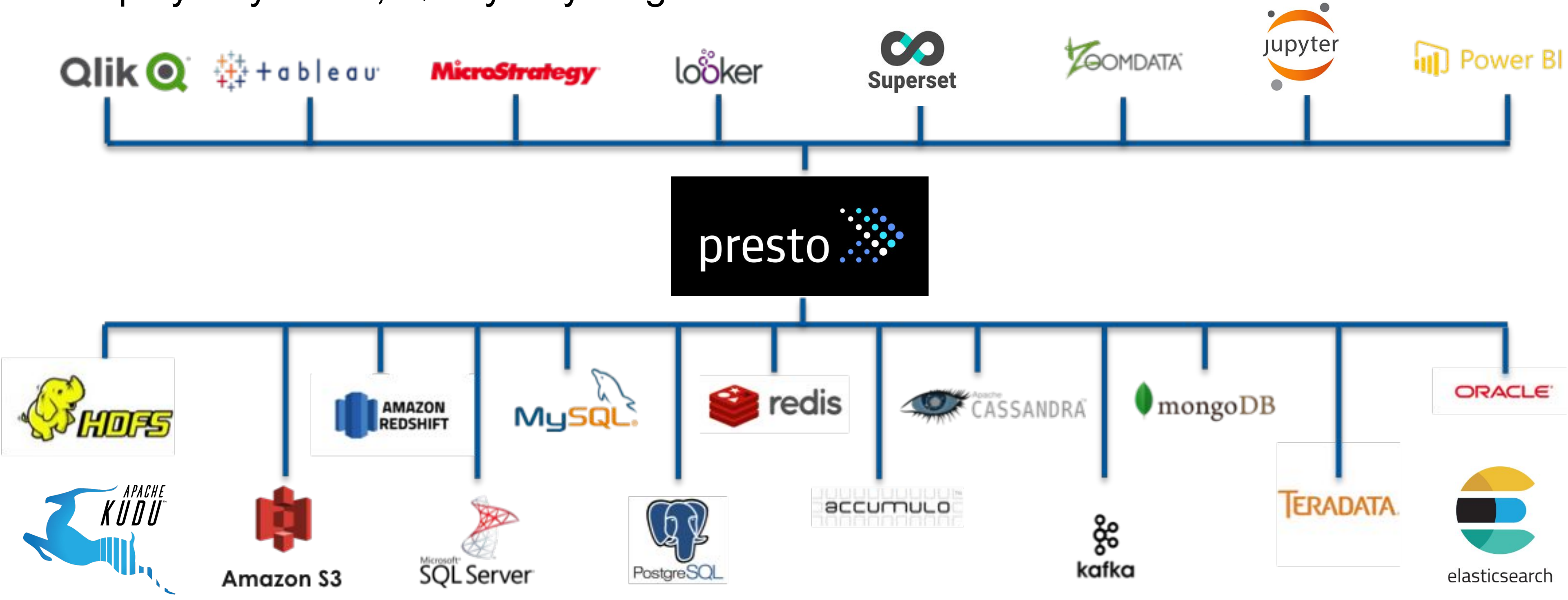
@prestosql @starburstdata

Martin Traverso, Co-creator of Presto

Kamil Bajda-Pawlikowski, CTO Starburst

Presto: SQL-on-Anything

Deploy Anywhere, Query Anything



Why Presto?

Community-driven
open source project

High performance ANSI SQL engine

- New Cost-Based Query Optimizer
- Proven scalability
- High concurrency

Separation of compute and storage

- Scale storage and compute independently
- No ETL or data integration necessary to get to insights
- SQL-on-anything

No vendor lock-in

- No Hadoop distro vendor lock-in
- No storage engine vendor lock-in
- No cloud vendor lock-in

Project History

presto

FALL 2012

4 developers start Presto development

FALL 2013

Facebook open sources Presto

SPRING 2015

Teradata joins the community, begins investing heavily in the project

SUMMER 2017

180+ Releases
50+ Contributors
5000+ Commits

WINTER 2017

Starburst is founded by a team of Presto committers, Teradata veterans

WINTER 2019

Presto Software Foundation established

Community

See more at our Wiki

Checkr amazon YAHOO! JAPAN JD.COM 京东 FreeWheel airbnb

mercado Libre.com Alibaba Group allegro jampp TiVo Marin SOFTWARE

ELE INTERNATIONAL SmartNews Eventbrite zomato TREASURE DATA

looker LinkedIn Supership zuora

WB Comcast. Starburst UBER glassdoor GROUPON

Walmart WiX FINRA

Bloomberg 去哪儿? Qunar.Com Grab Zillow GROUP lyft

twitter Qubole ELEMENTAL TO BIG DATA GREE cogo labs SCHIBSTED MEDIA GROUP

AdRoll trulia shopify Atlassian slack

SHAZAM Pinterest Akamai openspan

NASDAQ Dropbox CUEBIQ NETFLIX LINE

Presto in Production

Facebook: 10,000+ of nodes, HDFS (ORC, RCFile), sharded MySQL, 1000s of users

Uber: 2,000+ nodes (several clusters on premises) with 160K+ queries daily over HDFS (Parquet/ORC)

Twitter: 2,000+ nodes (several clusters on premises and GCP), 20K+ queries daily (Parquet)

LinkedIn: 500+ nodes, 200K+ queries daily over HDFS (ORC), and ~1000 users

Lyft: *------ redacted due to the quiet period for the IPO -----------*

Netflix: 300+ nodes in AWS, 100+ PB in S3 (Parquet)

Yahoo! Japan: 200+ nodes for HDFS (ORC), and ObjectStore

FINRA: 120+ nodes in AWS, 4PB in S3 (ORC), 200+ users

Starburst Data

The Presto Experts.

Founded by Presto committers:

- Over 4 years of contributions to Presto
- Presto distro for on-prem and cloud env
- Supporting large customers in production
- Enterprise subscription add-ons (ODBC, Ranger, Sentry, Oracle, Teradata)

Notable features contributed:

- ANSI SQL syntax enhancements
- Execution engine improvements
- Security integrations
- Spill to disk
- Cost-Based Optimizer

https://www.starburstdata.com/presto-enterprise/

Performance

Built for Performance

Query Execution Engine:

- MPP-style **pipelined** in-memory execution
- **Columnar** and **vectorized** data processing
- Runtime query **bytecode compilation**
- Memory **efficient** data structures
- Multi-threaded multi-core execution
- Optimized readers for **columnar formats** (ORC and Parquet)
- Predicate and column projection **pushdown**
- Now also **Cost-Based Optimizer**

CBO in a nutshell

Presto Cost-Based Optimizer includes:

- support for **statistics** stored in Hive Metastore
- **join reordering** based on selectivity estimates and cost
- automatic **join type** selection (repartitioned vs broadcast)
- automatic left/right side selection for joined tables

https://www.starburstdata.com/technical-blog/

Statistics & Cost

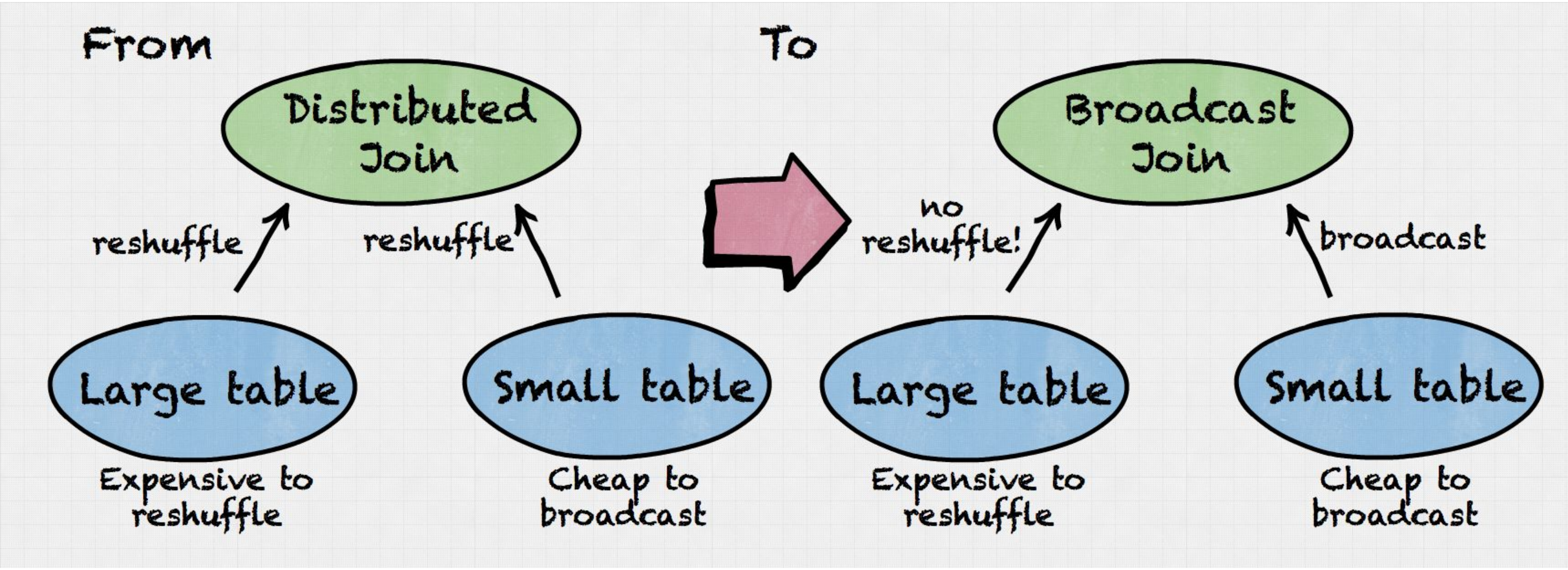
Hive Metastore statistics:

- number of rows in a table
- number of distinct values in a column
- fraction of NULL values in a column
- minimum/maximum value in a column
- average data size for a column

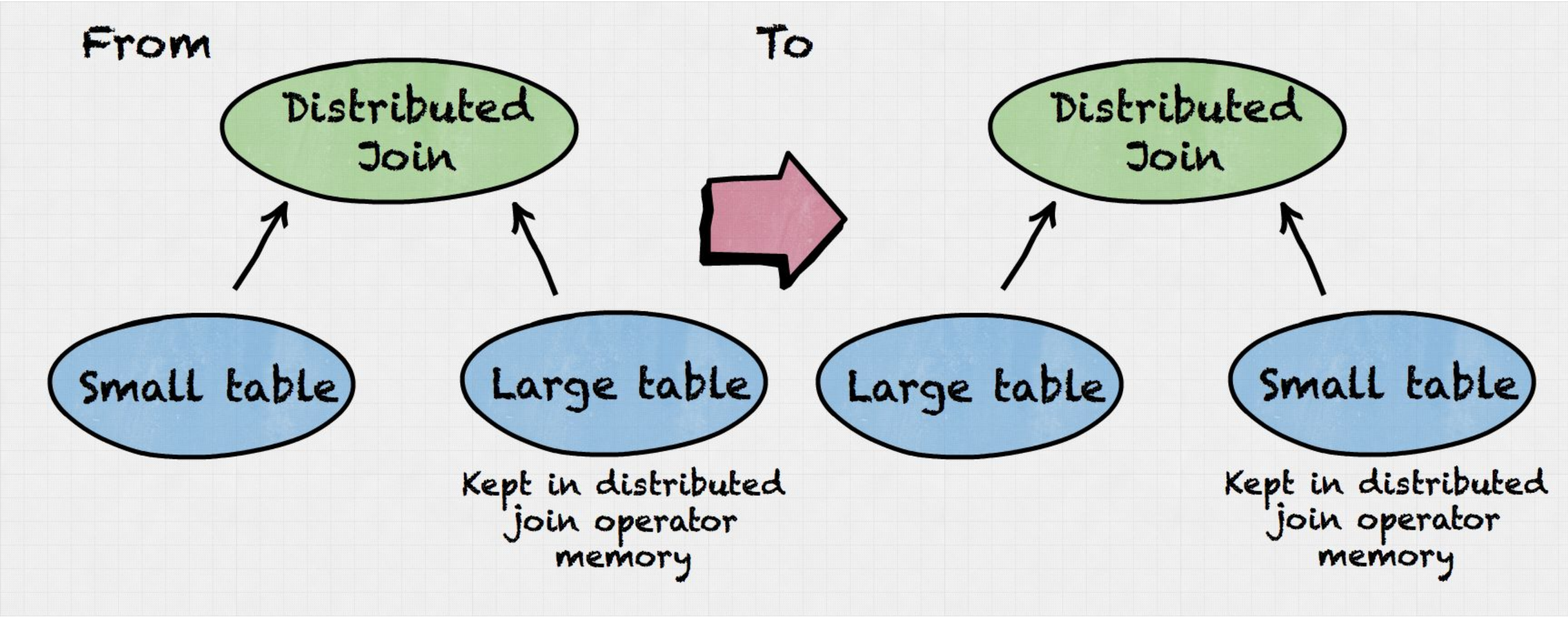
Cost calculation includes:

- CPU
- Memory
- Network I/O

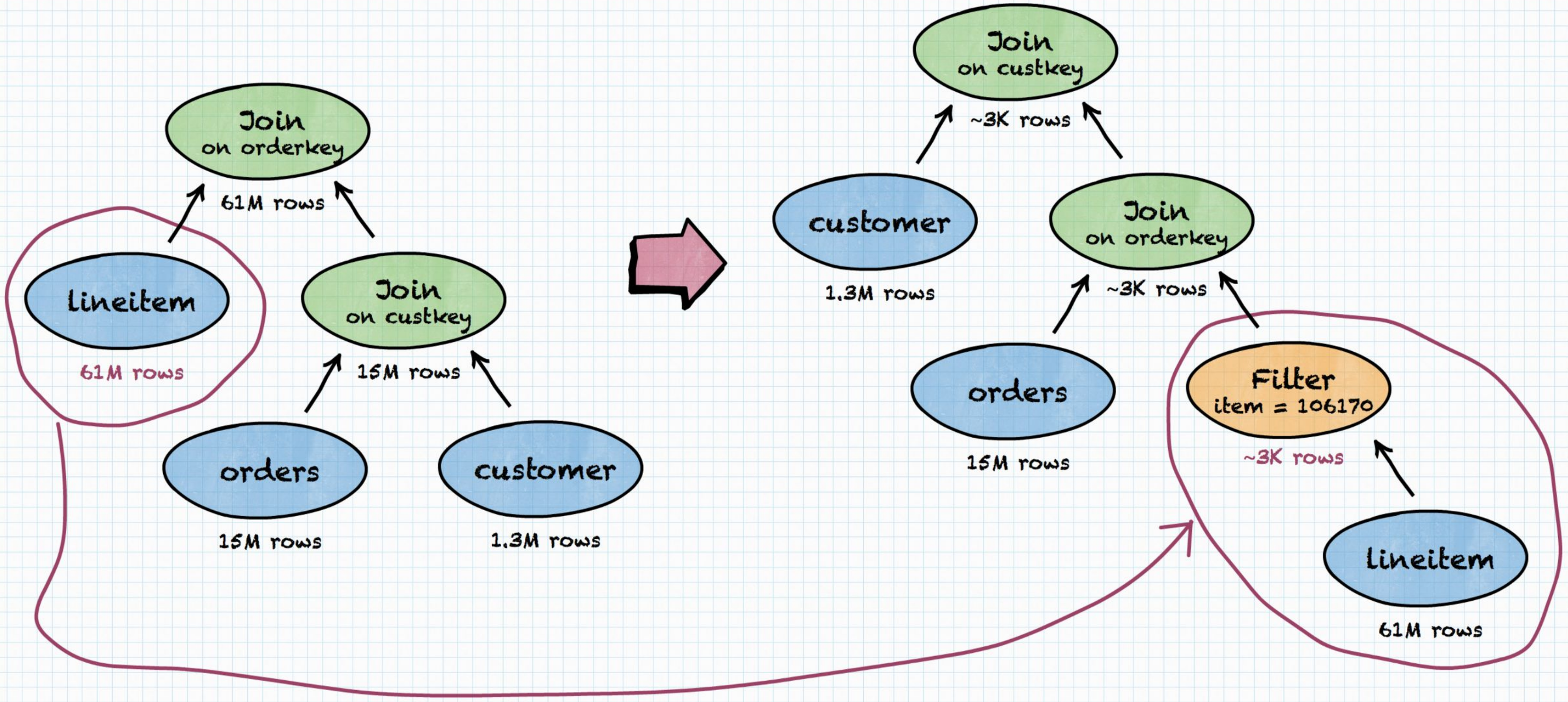
Join type selection



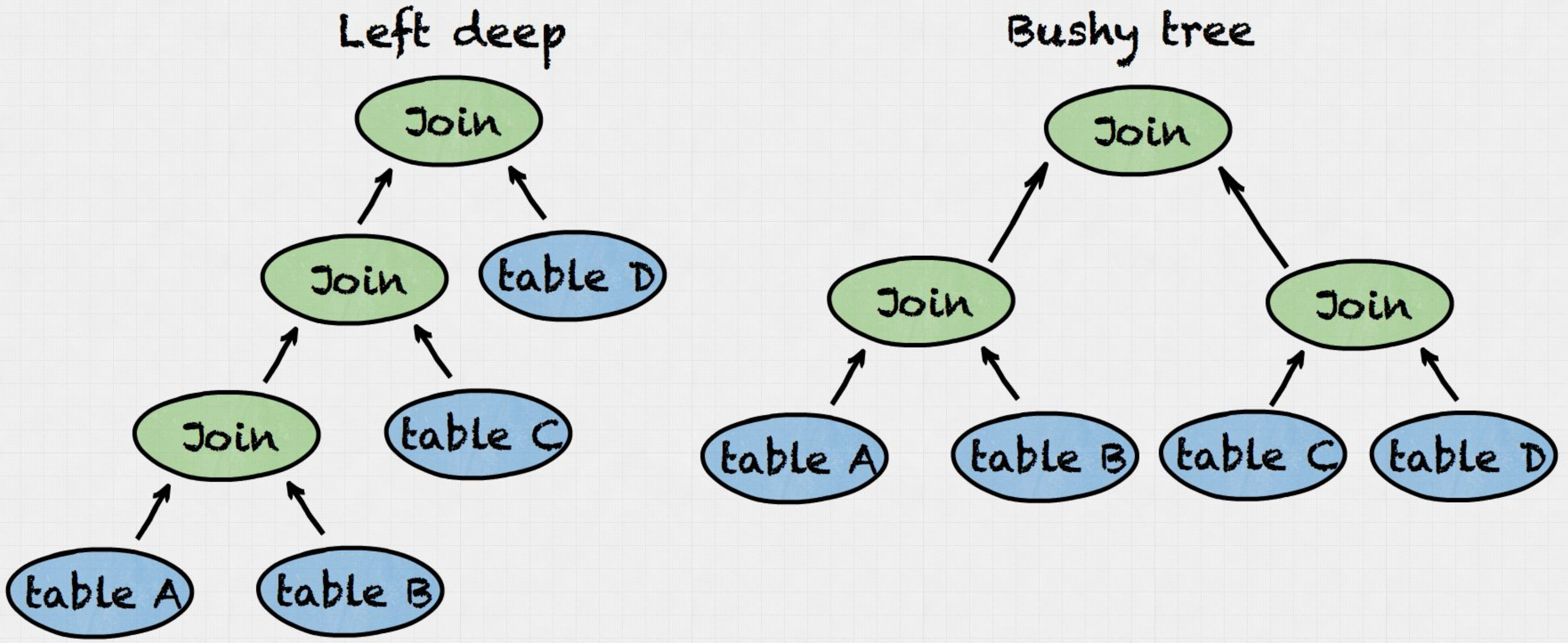
Join left/right side decision



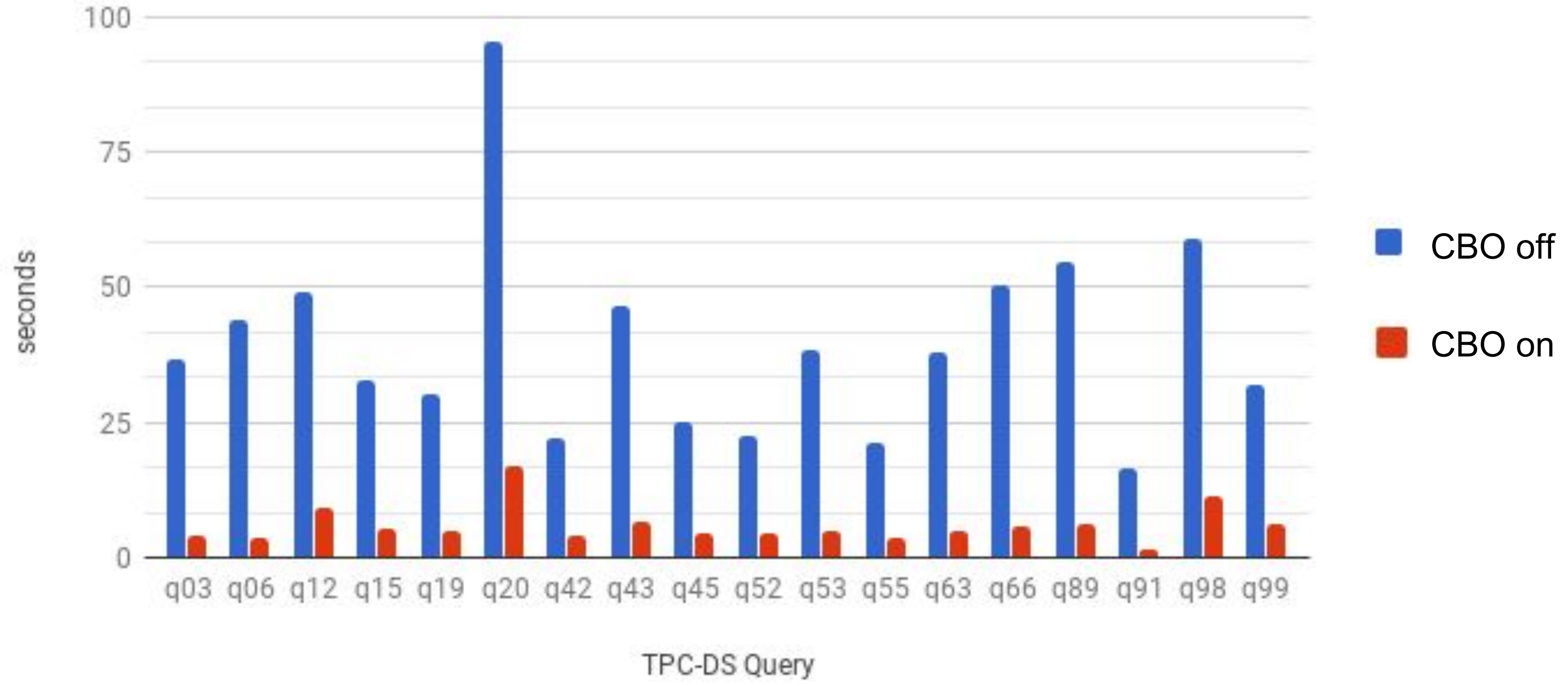
Join reordering with filter



Join tree shapes



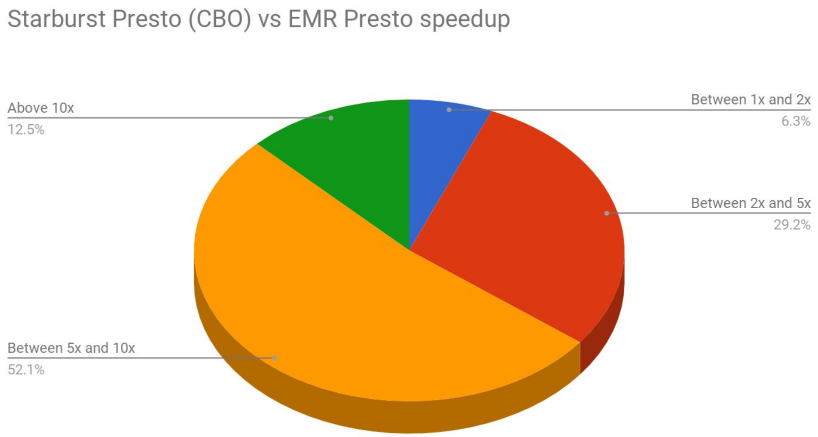
Benchmark results

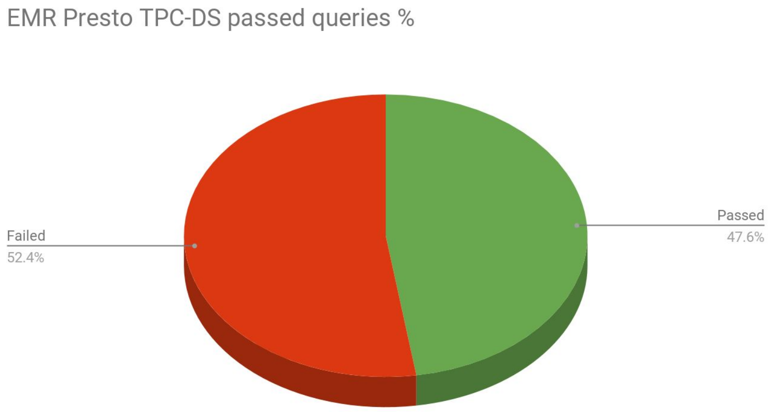


https://www.starburstdata.com/presto-benchmarks/

Benchmark results

- on average 7x improvement vs EMR Presto
- EMR Presto cannot execute many TPC-DS queries
- All TPC-DS queries pass on Starburst Presto





https://www.starburstdata.com/presto-aws/

Recent CBO enhancements

- Deciding on semi-join distribution type based on cost
- Support for outer joins
- Capping a broadcasted table size
- Various minor fixes in cardinality estimation
- ANALYZE table (native in Presto)
- Stats for AWS Glue Catalog (exclusive from Starburst)

Current and Future work

What’s next for Optimizer

- Stats support
    - Improved stats for Hive
    - Stats for DBMS connectors and NoSQL connectors
    - Tolerate missing / incomplete stats
- Core CBO enhancements
    - Cost more operators
    - Adjust cost model weights based on the hardware
    - Adaptive optimizations
    - Introduce Traits
- Involve connectors in optimizations

Involving Connectors in Optimization

History and Current State

- Original motivation: partition pruning for queries over Hive tables
- Simple range predicates and nullability checks passed to connectors. Modeled as `TupleDomain`

```
((col0 BETWEEN ? AND ?) **OR** (col0 BETWEEN ? and ?) **OR** …))
                          AND
((col1 BETWEEN ? AND ?) **OR** (col1 BETWEEN ? and ?) **OR** …))
                          AND
                          ...
```

History and Current State

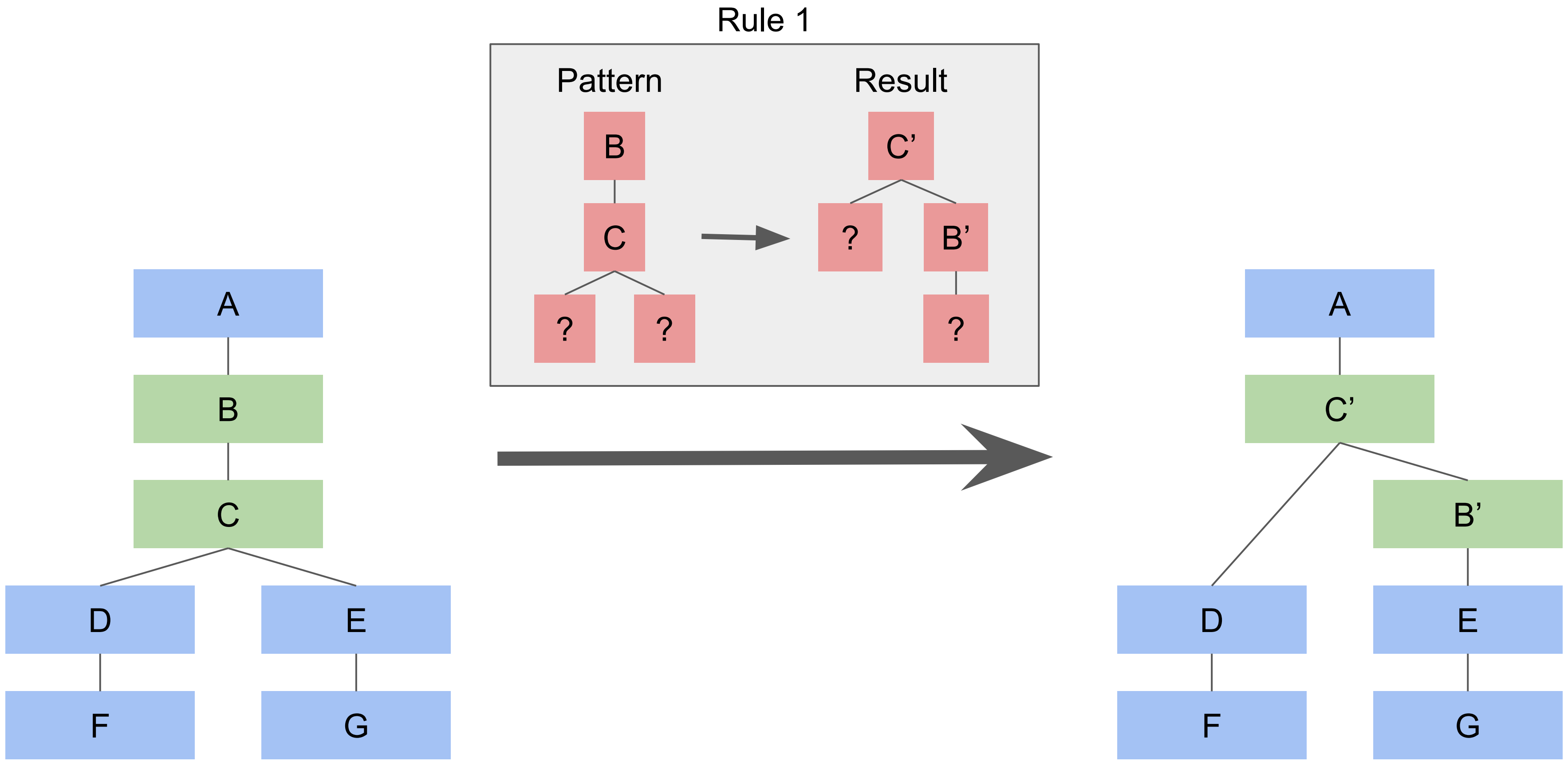
- Partial evaluation of non-trivial expressions
  - Bind only known variables
  - Result in "true/false/null" or "can't tell”. E.g.,

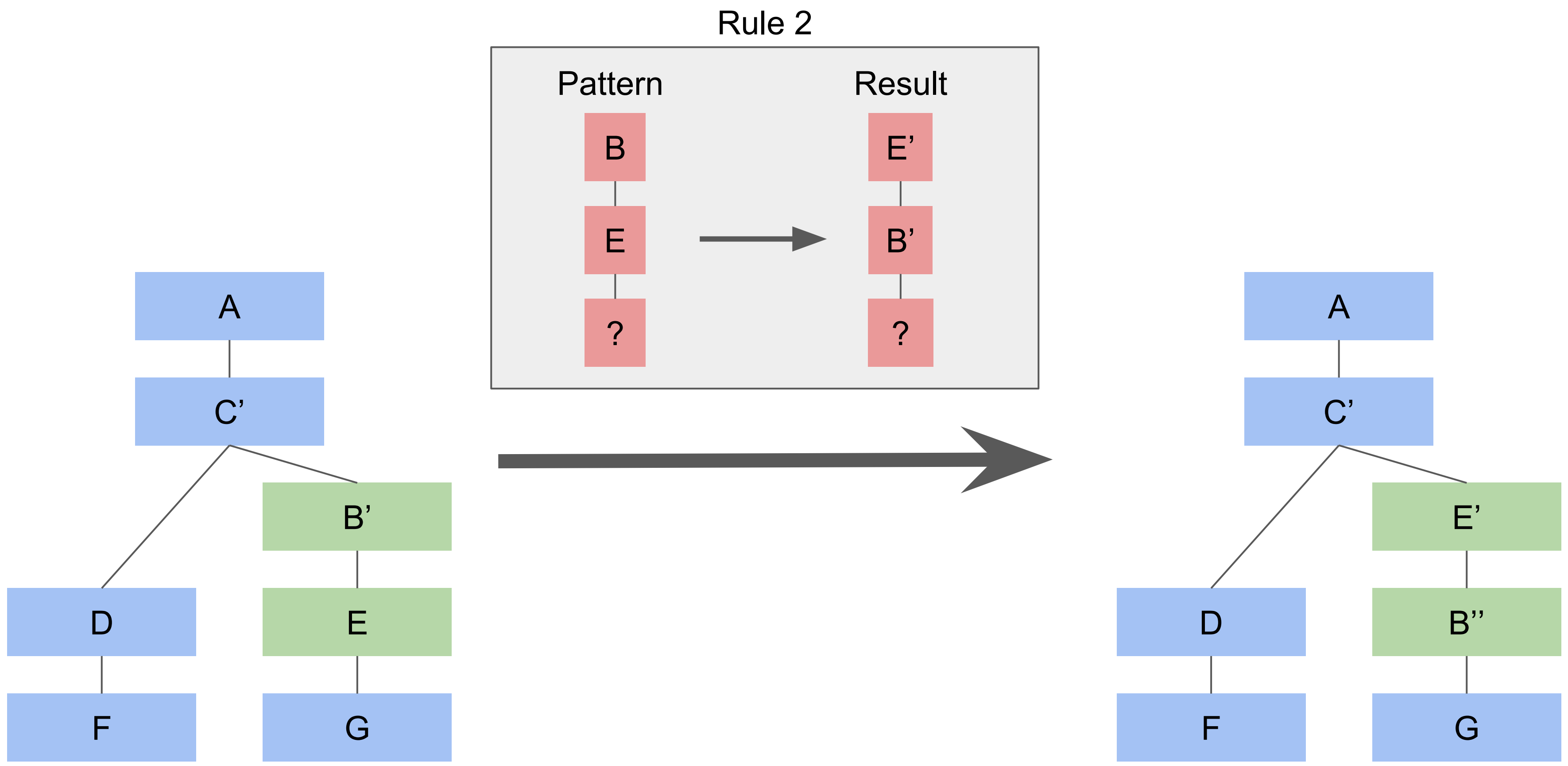
```
f(a, b) := lower(a) LIKE ‘john%’ AND b = 1

f(‘Mary’,   ?) ⟶ false ⟶ can prune
f(‘John S’, ?) ⟶ b = 1 ⟶ ¯\_(ツ)_/¯
```

Beyond Simple Filter Pushdown...

- Dereference expressions. E.g., `x.a > 5`
- Array/map subscript. E.g., `a[‘key’] = 10`
- Complex filters and projections
- Aggregations
- Joins
- Limit: ~~https://github.com/prestosql/presto/pull/421~~
- Sampling
- Others…

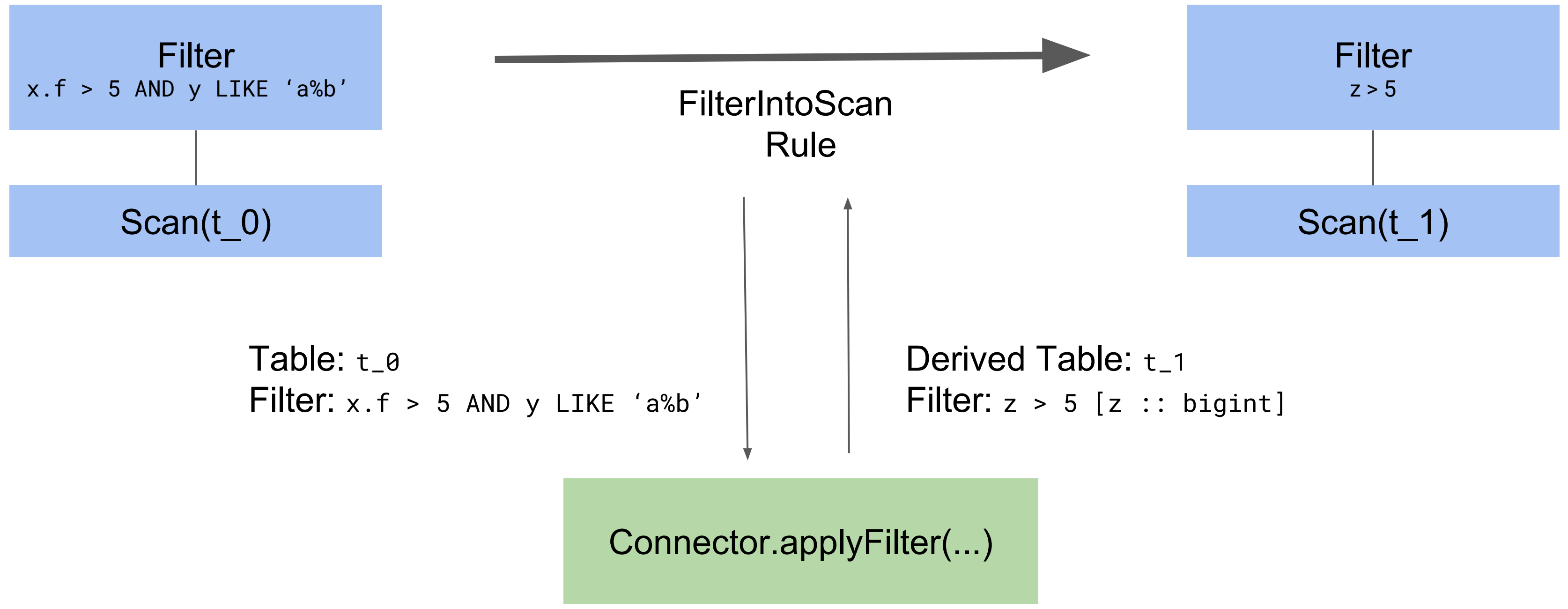
https://github.com/prestosql/presto/issues/18


Rule 1
Pattern
Result
B
C
?
?
C'
?
B'
?
A
B
C
D
E
F
G
A
C'
B'
D
E
F
G


Rule 2
Pattern
Result
B
E
?
E'
B'
?
A
C'
B'
D
E
F
G
A
C'
E'
D
B''
F
G

```
SELECT count(*)
FROM t
WHERE x.f > 5 AND y LIKE ‘a%b’
```

```
Table t
   x :: row(f bigint, g bigint)
   y :: varchar(10)
```



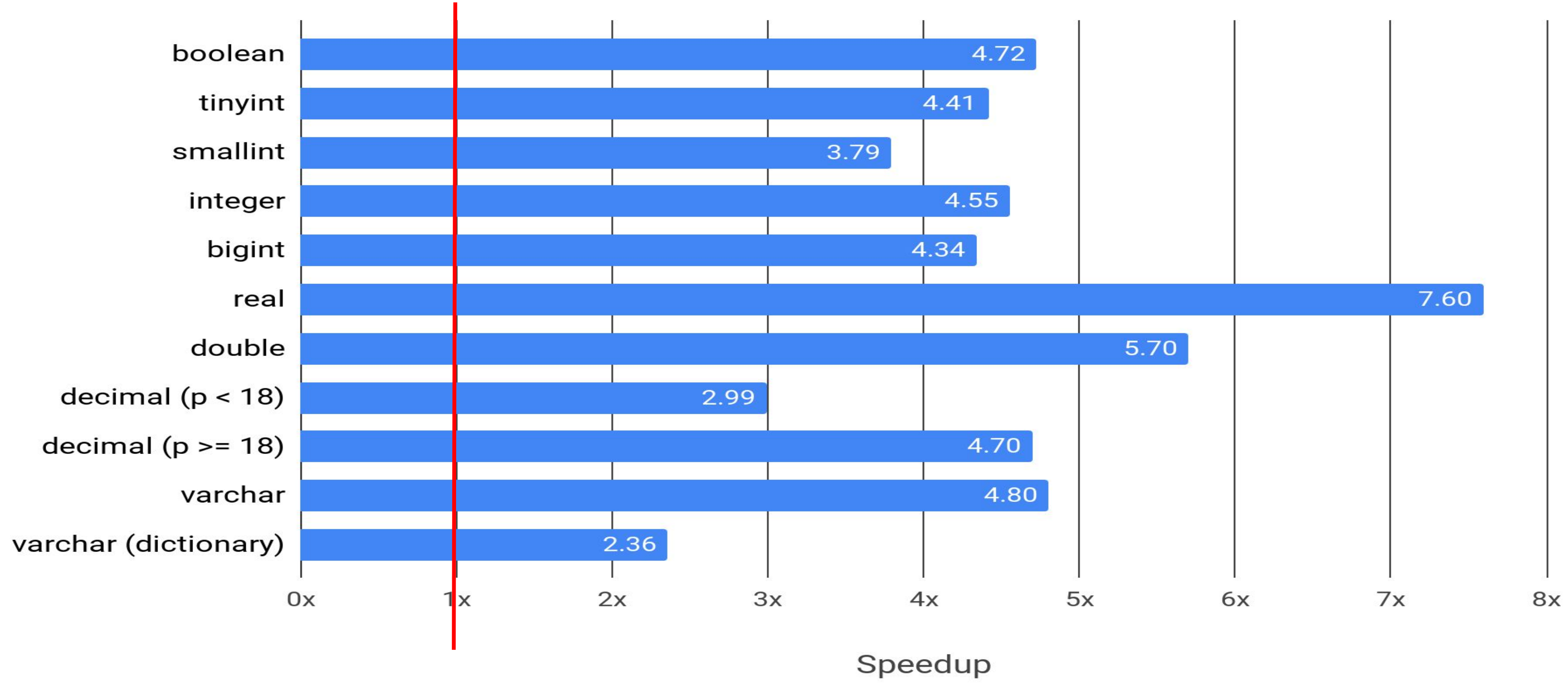
New Connector APIs

```
applyFilter(ConnectorTableHandle table, Expression filter)

applyLimit(ConnectorTableHandle table, long limit)

applyAggregation(ConnectorTableHandle table, List<Aggregation> aggregates)

applySampling(ConnectorTableHandle table, double samplingRate)

...
```

Performance Benefits (?)

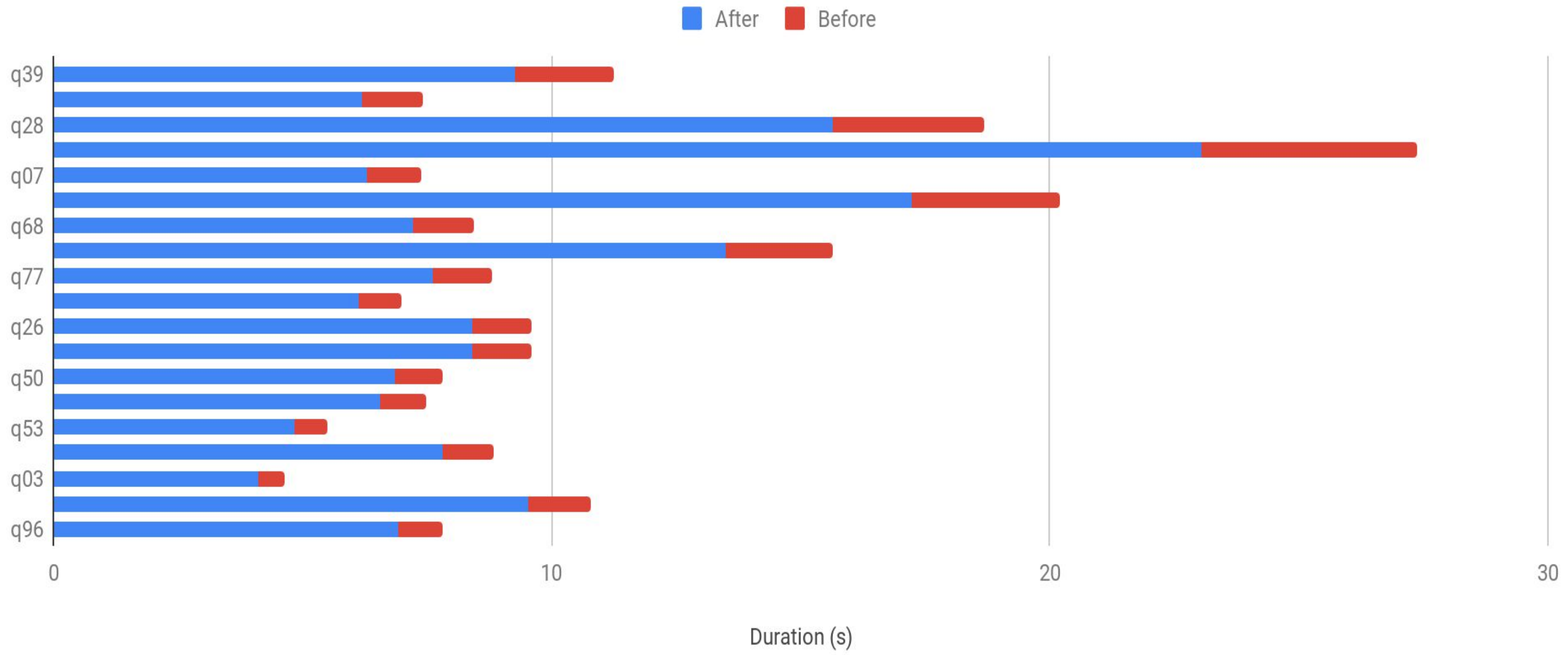
- Better support for sophisticated backend systems
  - Druid, Pinot, ElasticSearch
  - SQL databases
- Improved performance for columnar data formats (Parquet, ORC)

ORC Performance Improvements



https://github.com/prestosql/presto/pull/555

ORC Performance Improvements - TPC-DS



Project Roadmap

- Coordinator HA
- Kubernetes
- Dynamic filtering
- Connectors
    - Phoenix
    - Iceberg
    - Druid
- TIMESTAMP semantics
- And more… https://github.com/prestosql/presto/labels/roadmap

Getting Involved

- Join us on Slack
  - Invite link: https://prestosql.io/community.html
- Github: https://github.io/prestosql/presto
- Website: https://prestosql.io

Further reading

https://www.starburstdata.com/presto-newsletter/

https://fivetran.com/blog/warehouse-benchmark

https://www.concurrencylabs.com/blog/starburst-presto-vs-aws-emr-sql/

http://bytes.schibsted.com/bigdata-sql-query-engine-benchmark/

https://virtuslab.com/blog/benchmarking-spark-sql-presto-hive-bi-processing-googles-cloud-dataproc/

Thank You!

www.prestosql.io

www.starburstdata.com

@prestosql @starburstdata