Starburst

presto

The Presto Cost-Based Optimizer for interactive SQL on anything

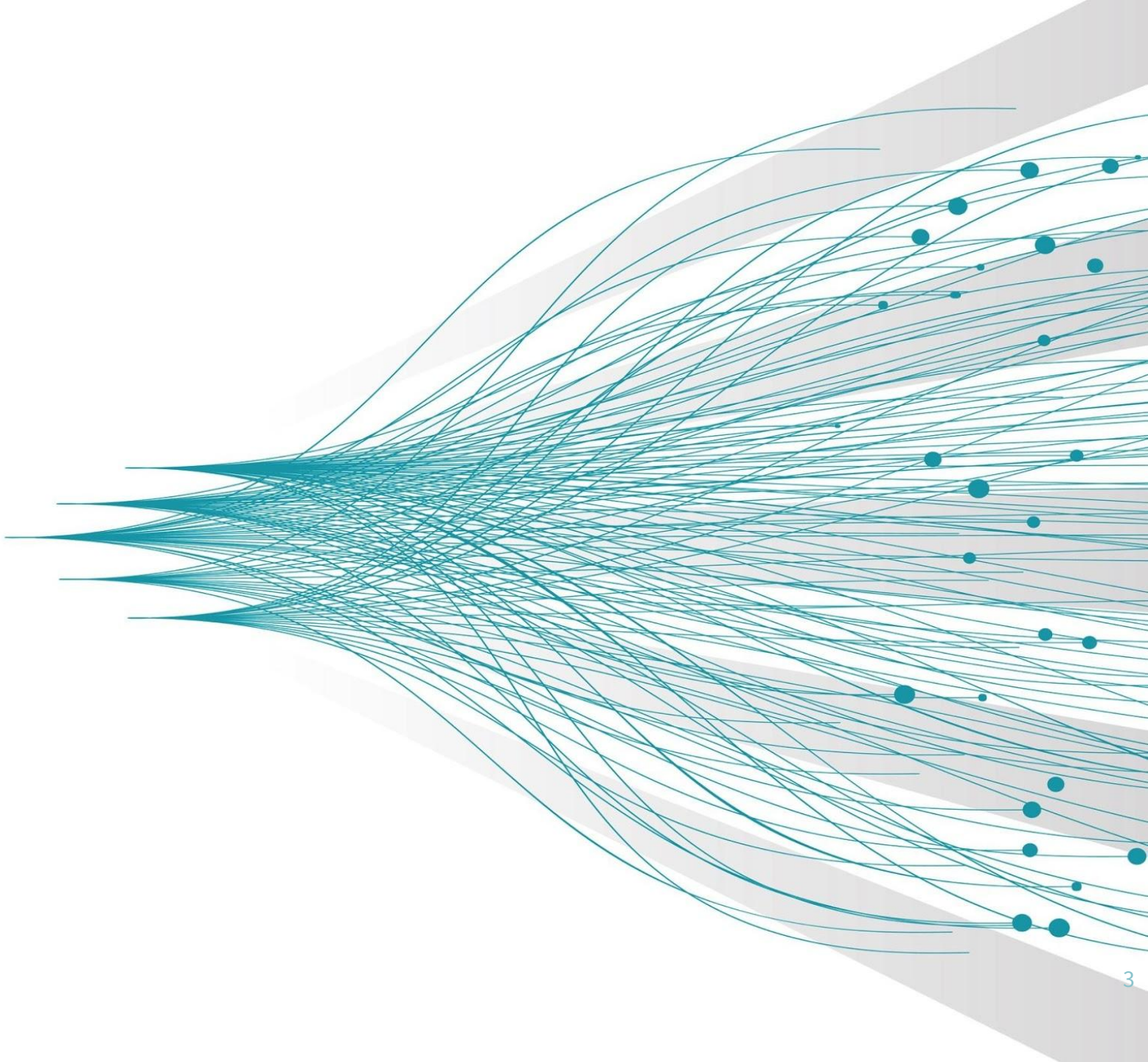
Strata Data Conference - London
May 1, 2019

Wojciech.Biela@starburstdata.com
Piotr.Findeisen@starburstdata.com

Agenda

- Who we are?
- Presto - quick recap
- Presto’s CBO launch
- Recent CBO enhancements
- CBO Roadmap

Starburst



Starburst Data

Founded by Presto committers:

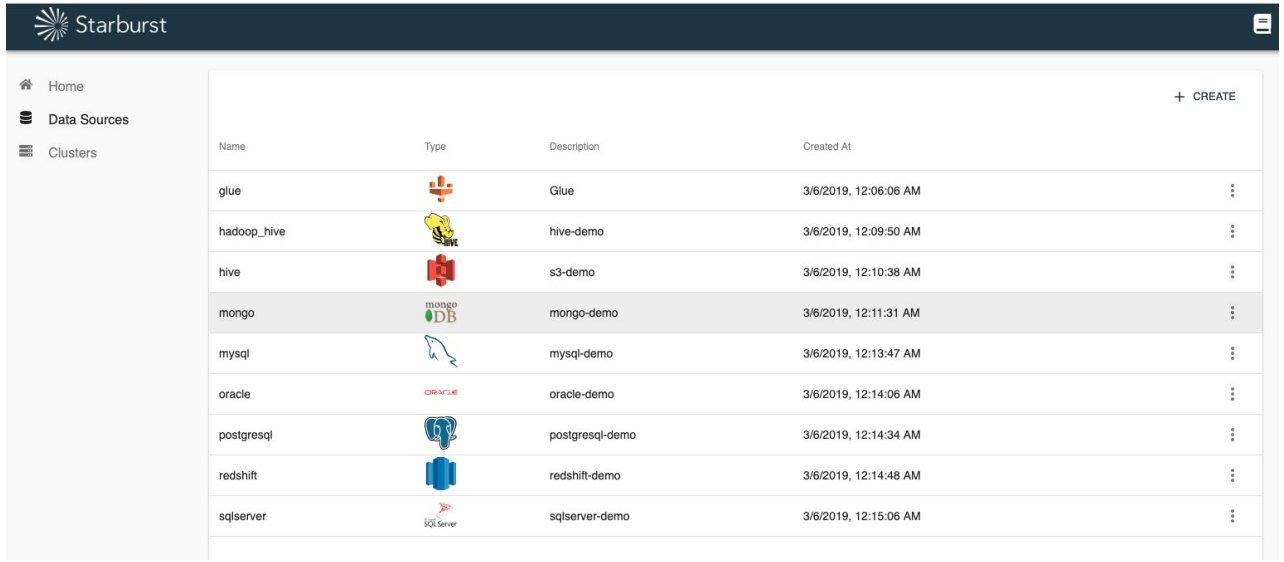
- Over 4 years of contributions to Presto
- Presto distro for on-prem and cloud env
- Supporting large customers in production
- Enterprise subscription add-ons

Notable features contributed:

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

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

Starburst Presto & Cloud

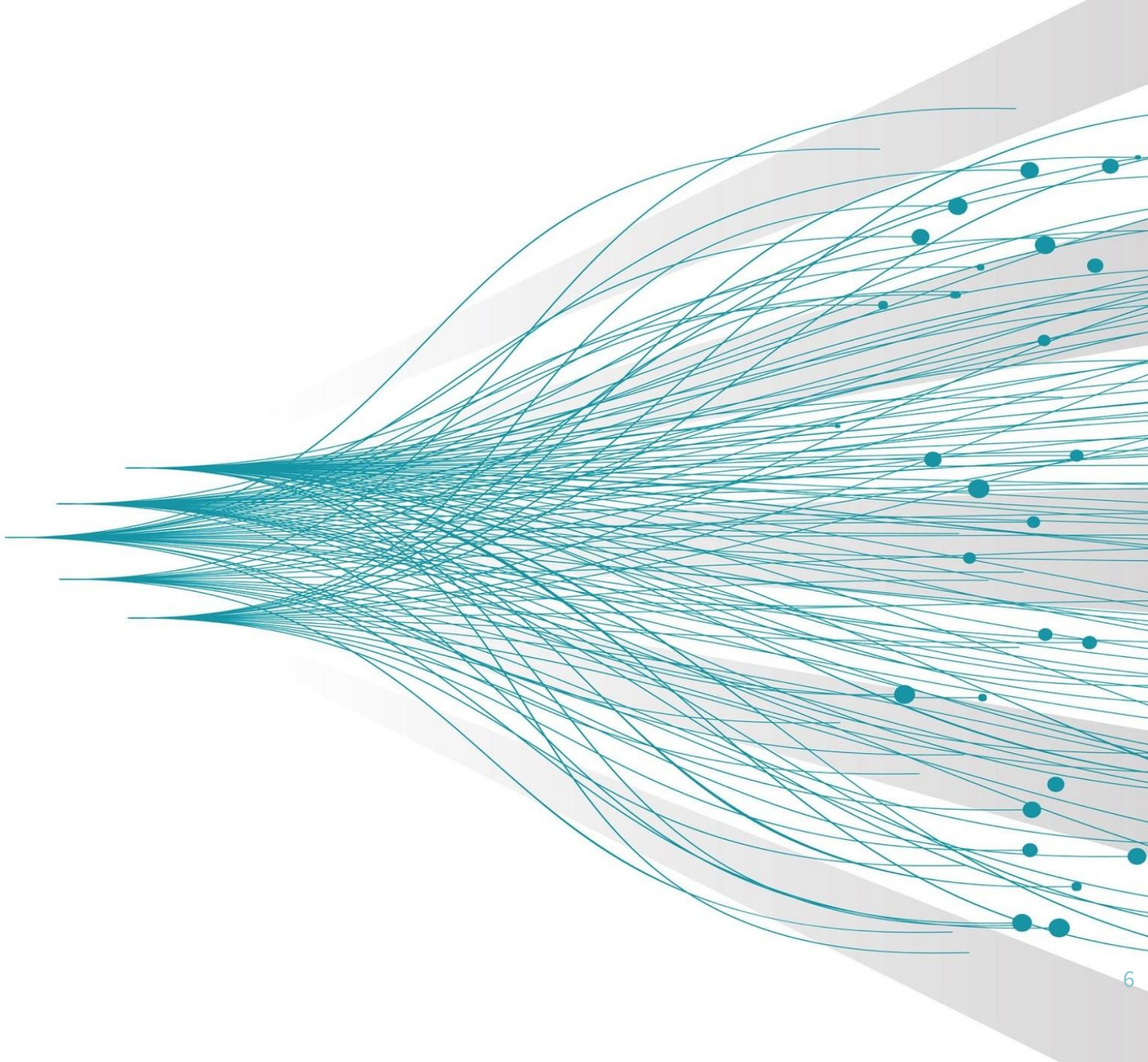


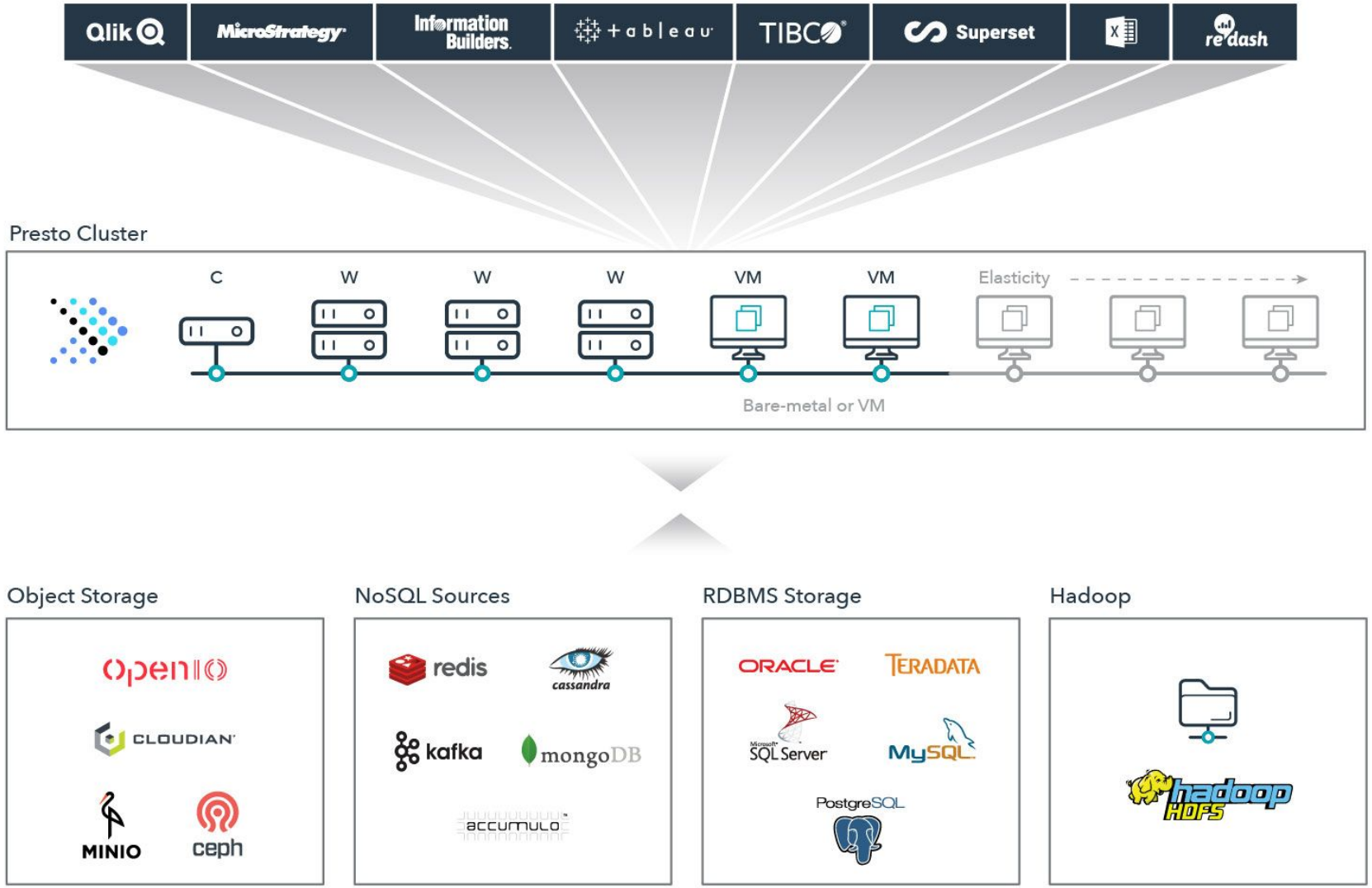
Azure HDInsight

Starburst Presto

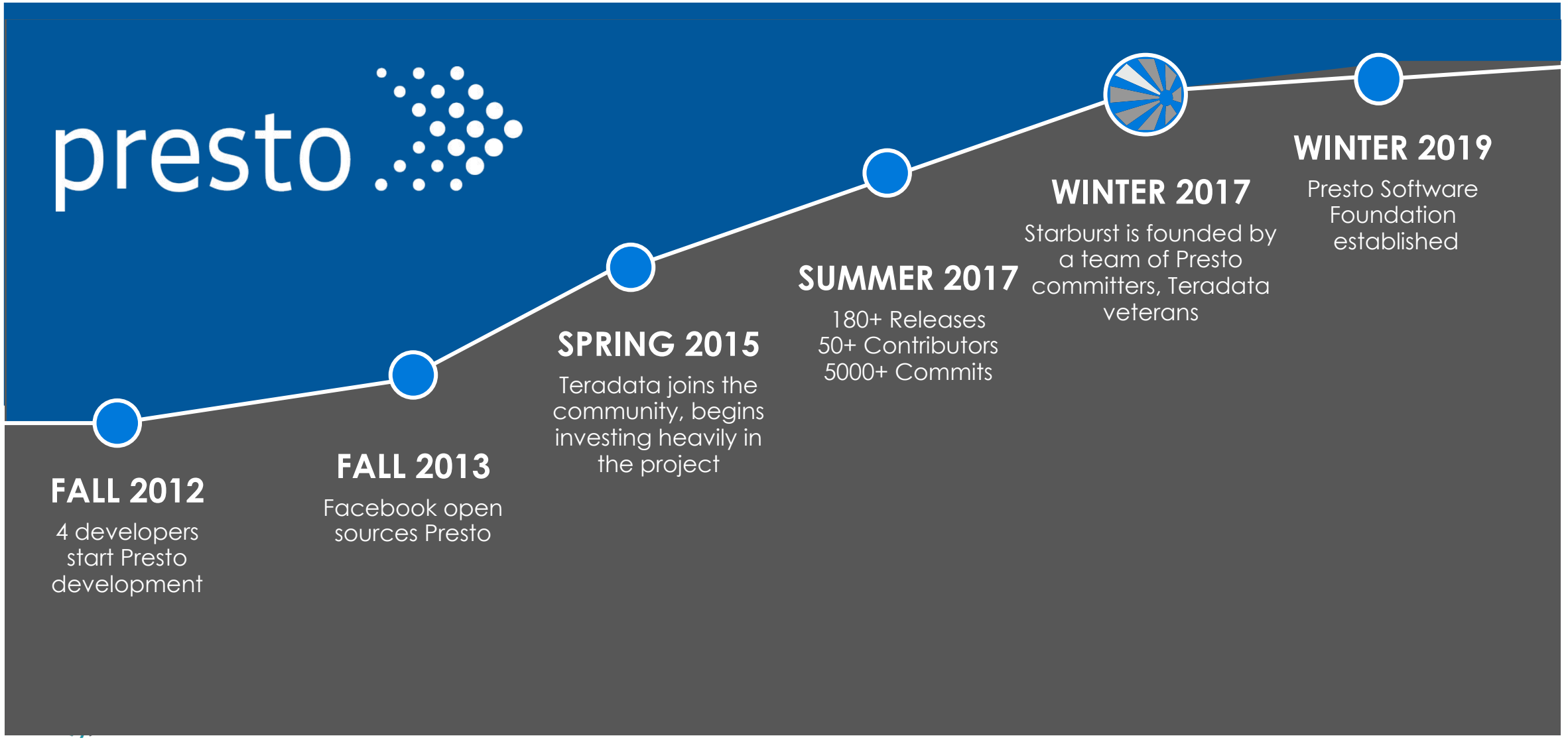
https://www.starburstdata.com/technical-blog/announcing-starburst-enterprise-302e-with-mission-control/

Presto




Qlik
MicroStrategy
Information Builders
+ableau
TIBCO
Superset
re:dash
Presto Cluster
C
W
W
W
VM
VM
Elasticity
Bare-metal or VM
Object Storage
Open IO
CLOUDIAN
MINIO
ceph
NoSQL Sources
redis
cassandra
kafka
mongoDB
accumulo
RDBMS Storage
ORACLE
TERADATA
Microsoft SQL Server
MySQL
PostgreSQL
Hadoop
hadoop HDFS

Project History



The Presto fan club

See more at
https://github.com/prestosql/presto/wiki/Presto-Users

Presto in Production

- **Facebook**: 10,000+ of nodes, HDFS (ORC, RCFile), sharded MySQL, 1000s of users
- **Uber**: 2,000+ nodes (clusters on prem.) 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**: 400+ nodes in AWS, 100K+ queries daily, 20+ PBs in S3 (Parquet)
- **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

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

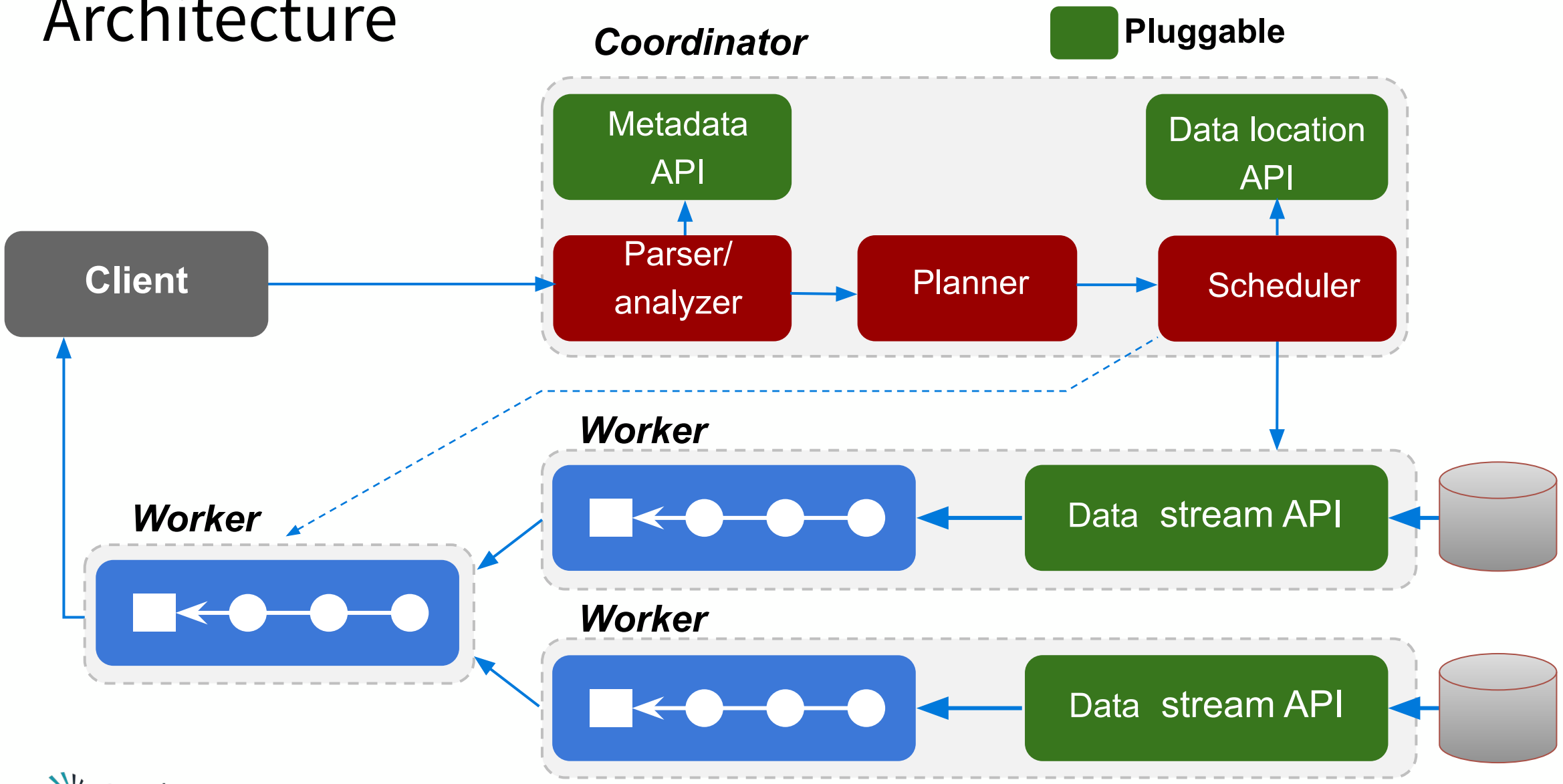
- No Hadoop distro vendor lock-in
- No storage engine vendor lock-in
- No cloud vendor lock-in

Built for Performance

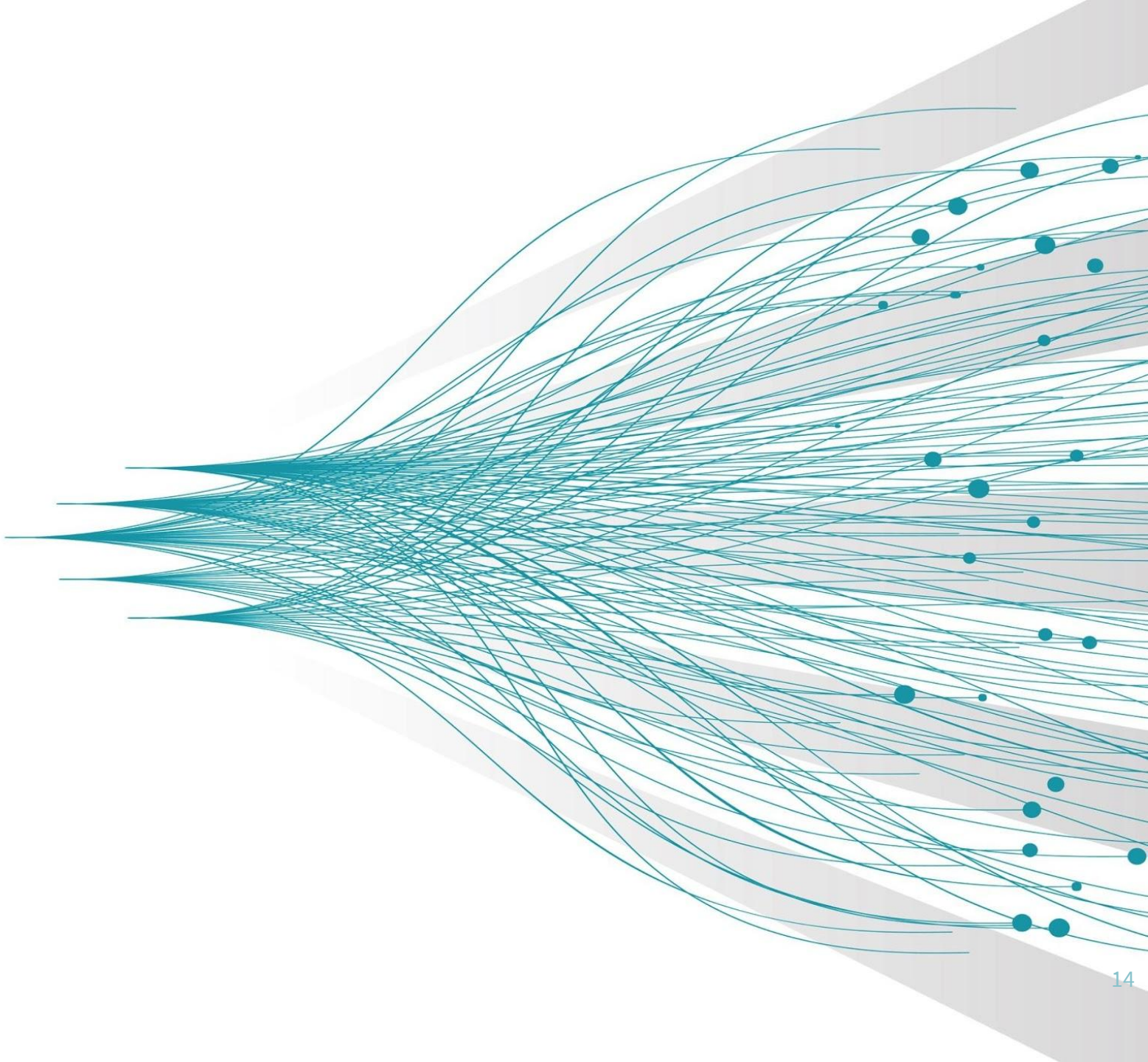
Query Execution Engine:

- MPP-style **pipelined** in-memory execution
- **Vectorized** data processing
- Runtime query **bytecode generation**
- 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**

Architecture



Presto CBO



Before we optimize ...

Join in Presto

- Hash Join
- Right table is in memory ("build table")
- Left table is streamed ("probe table")
- Can be broadcast or repartitioned

Before we optimize ...

Join in Presto

- Hash Join
- Right table is in memory ("build table")
- Left table is streamed ("probe table")
- Can be broadcast or repartitioned

- A join can be followed by a join, can be followed by a join…

CBO in a nutshell

Cost-Based Optimizer v1 includes:

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

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

Cost and Statistics

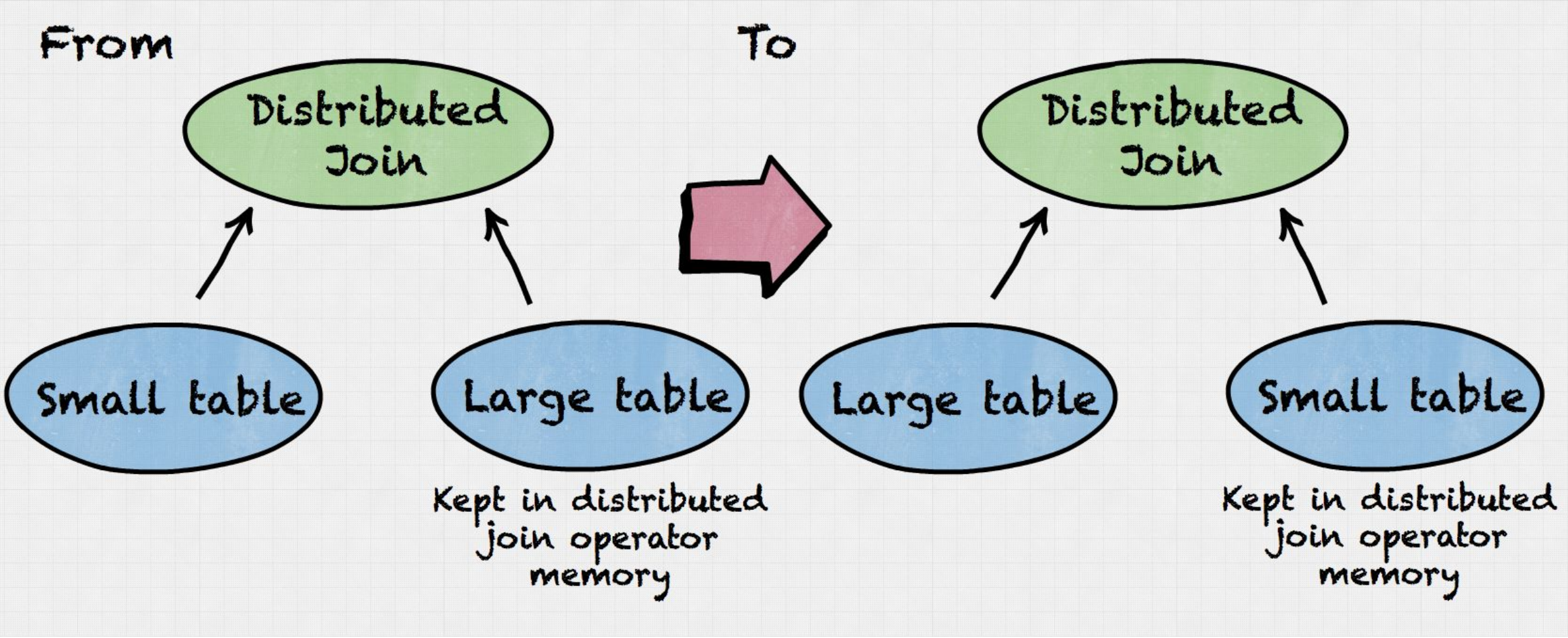
Cost calculation includes:

- CPU
- Memory usage
- Network I/O

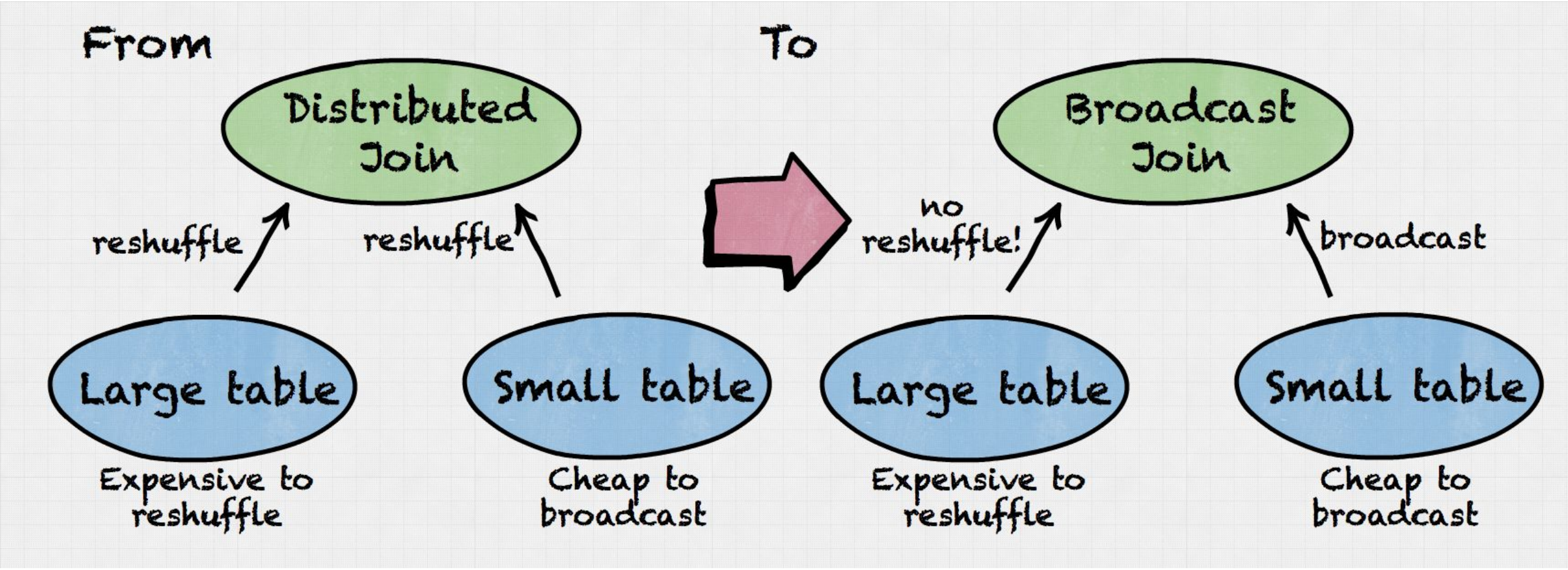
Hive Metastore statistics:

- 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

Join left/right side decision



Join type selection

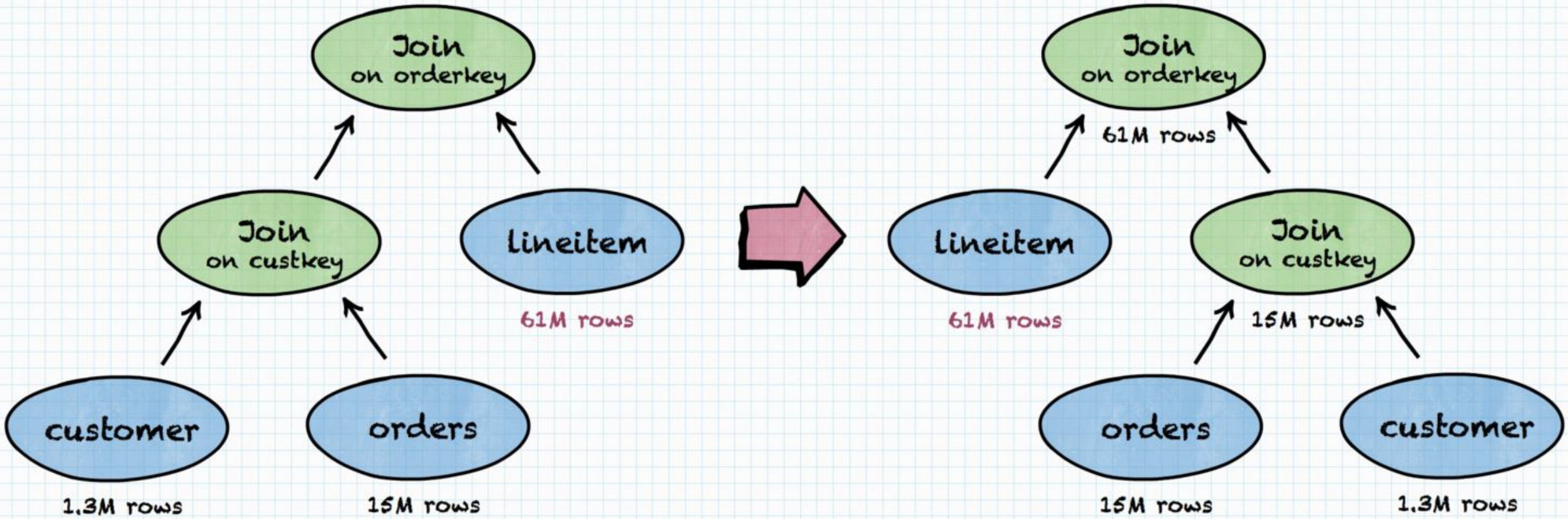


Join reordering

"Which customers are spending the most at our shop?"

```
SELECT c.custkey, sum(l.price)
FROM customer c
JOIN orders o ON c.custkey = o.custkey
JOIN lineitem l ON o.orderkey = l.orderkey
GROUP BY c.custkey
ORDER BY sum(l.price) DESC;
```

Join reordering

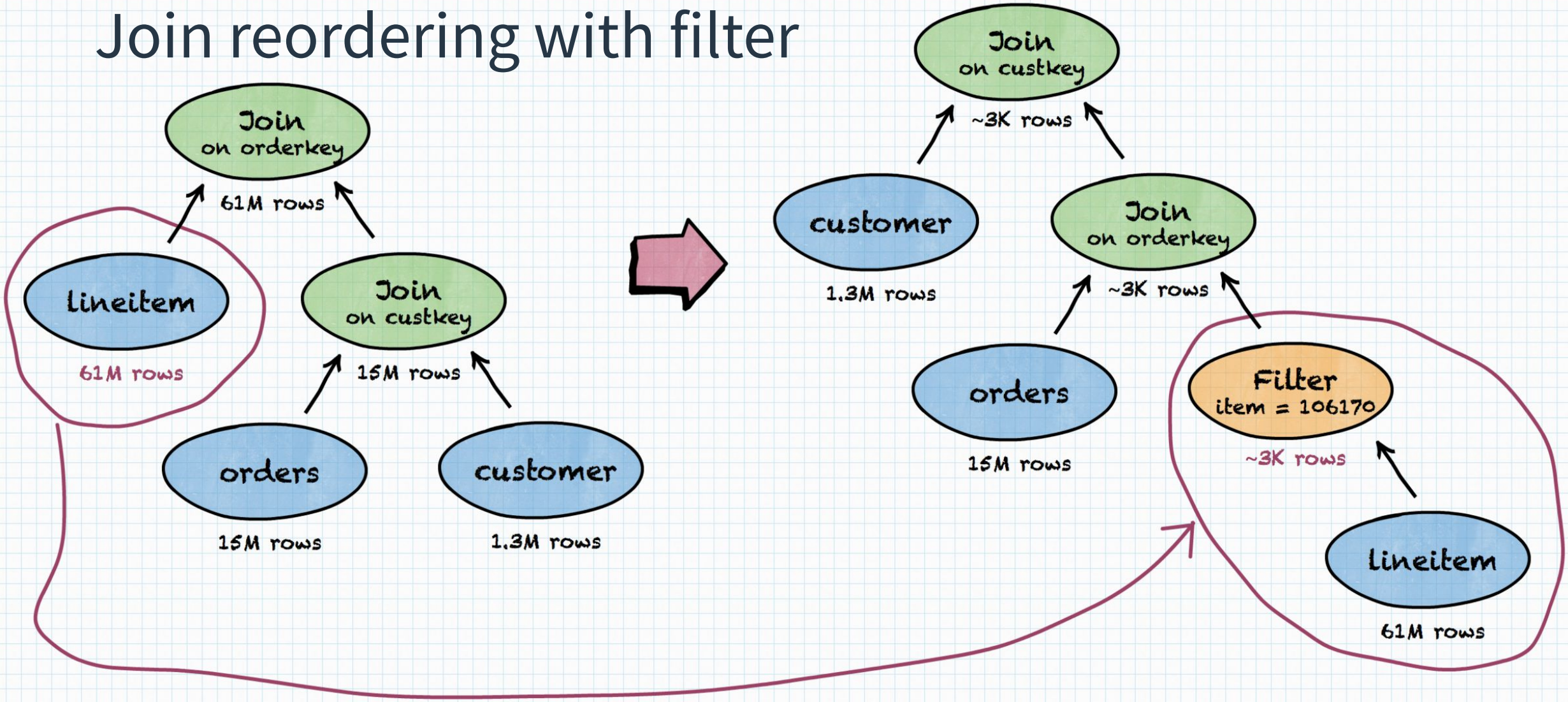


Join reordering with filter

"Which customers are spending the most on coffee?"

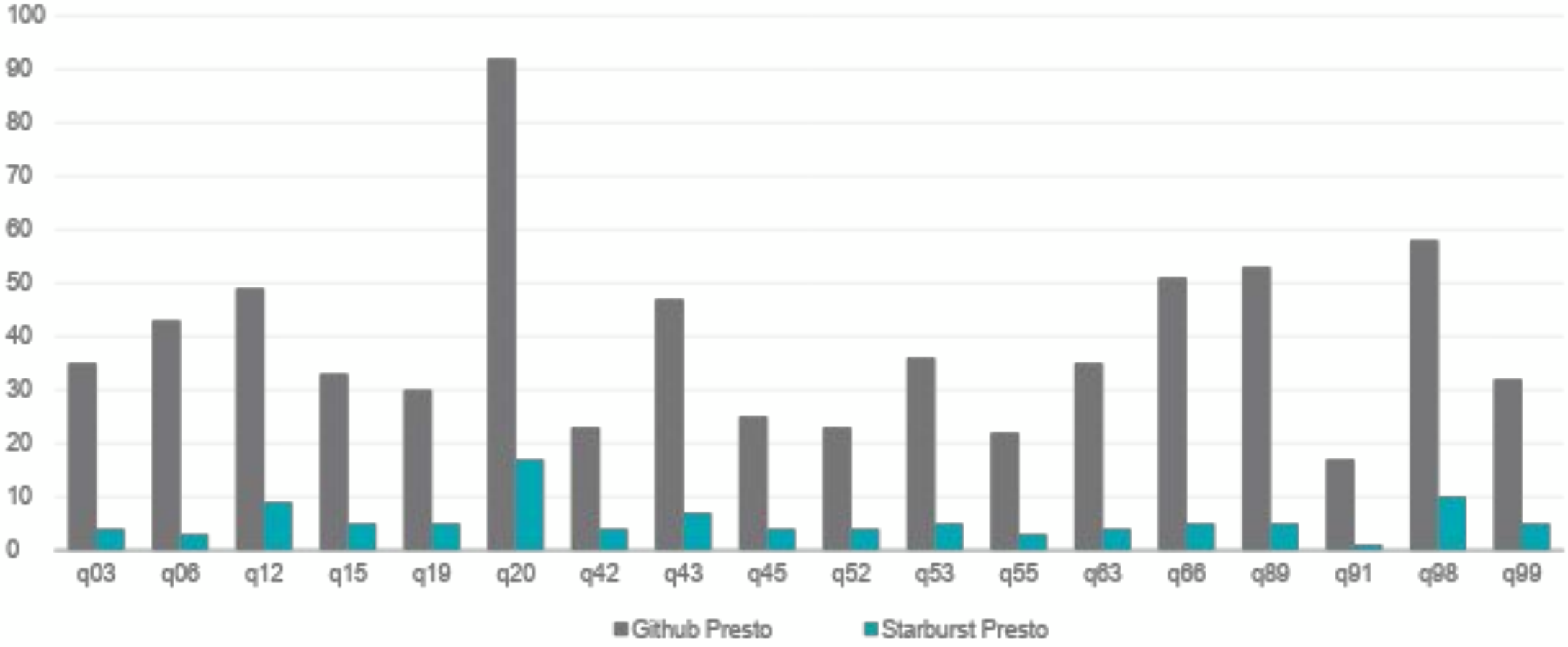
```
SELECT c.custkey, sum(l.price)
FROM customer c
JOIN orders o ON c.custkey = o.custkey
JOIN lineitem l ON o.orderkey = l.orderkey
WHERE l.item = 'coffee'
GROUP BY c.custkey
ORDER BY sum(l.price) DESC;
```

Join reordering with filter



Presto CBO Speedup

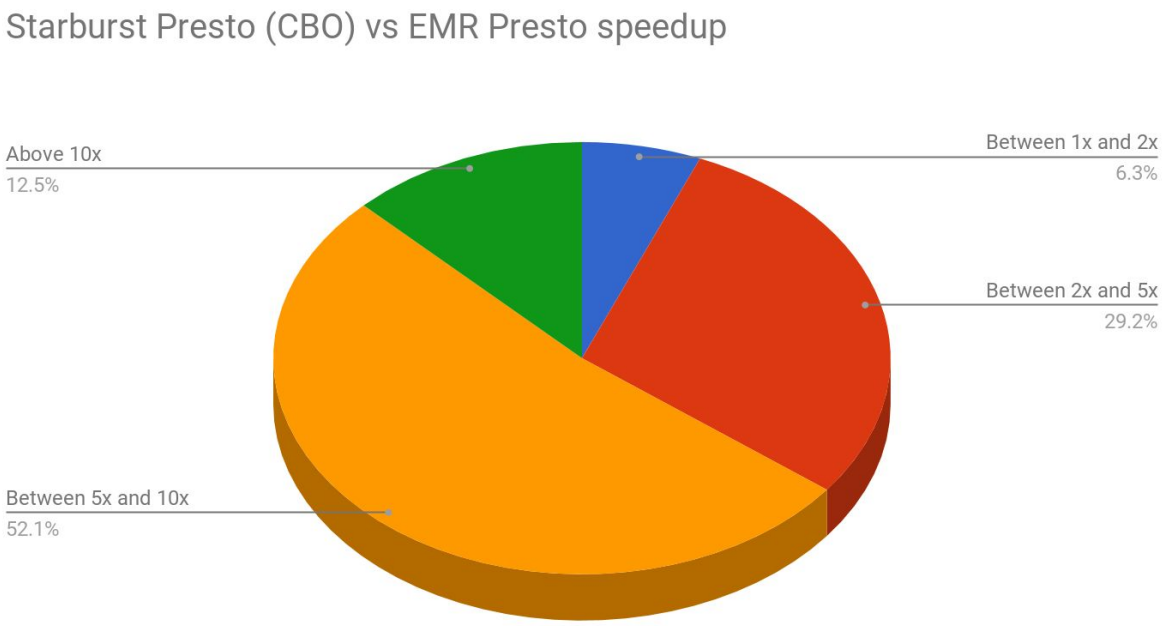
Duration of TPC-DS queries (lower is better)

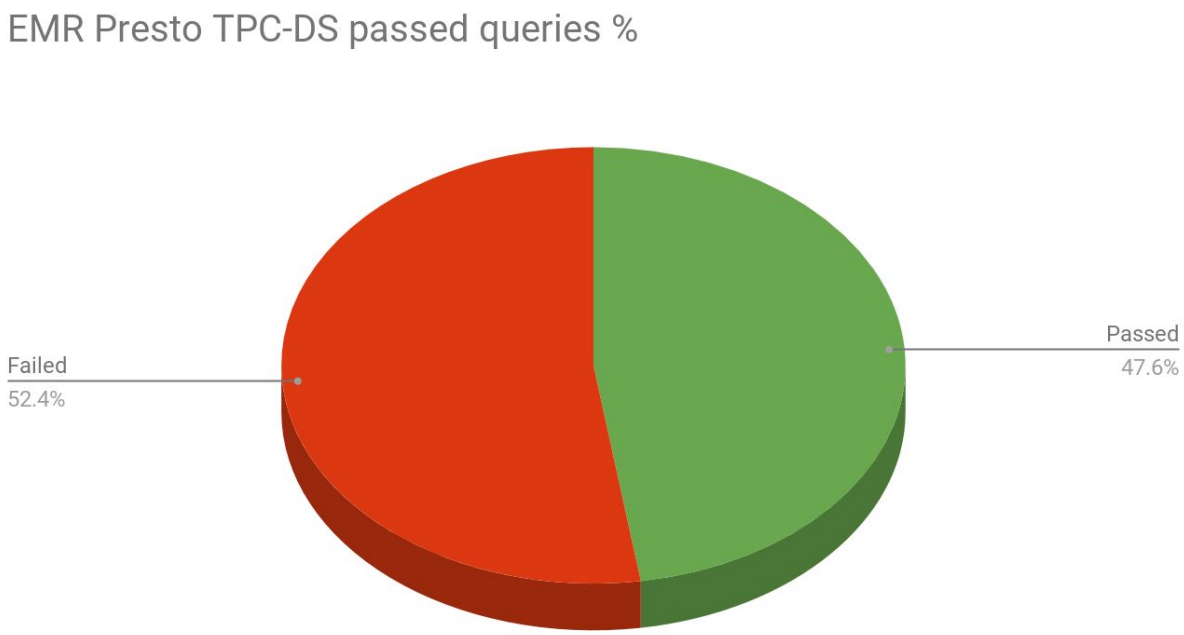


https://www.starburstdata.com/presto-benchmarks/

Cloud cost reduction

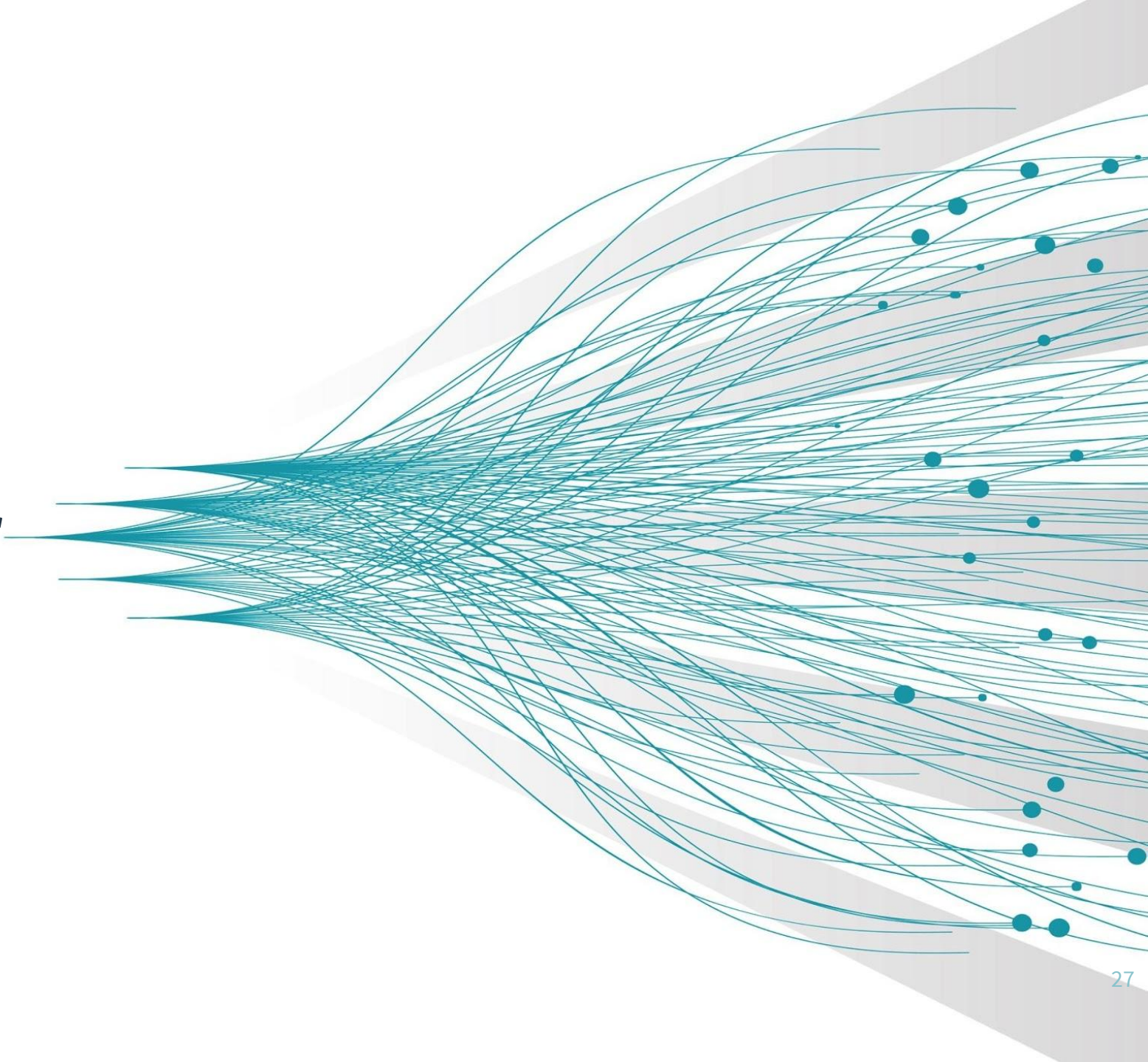
- on average 7x improvement vs EMR Presto (up to 18x faster!)
- EMR Presto cannot execute many TPC-DS queries (all pass on Starburst Presto)
- on average 4x improvement on simpler queries benchmark (TPC-H, 3rd party benchmark)





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

CBO Next Chapter



Enhancements to date

- Deciding on semi-join distribution type based on cost
  - "… `WHERE x IN (SELECT y FROM ...)`" queries
- Capping a broadcasted table size
- Various minor fixes in cardinality estimation
- Peak memory estimation (more meaningful memory cost metric)

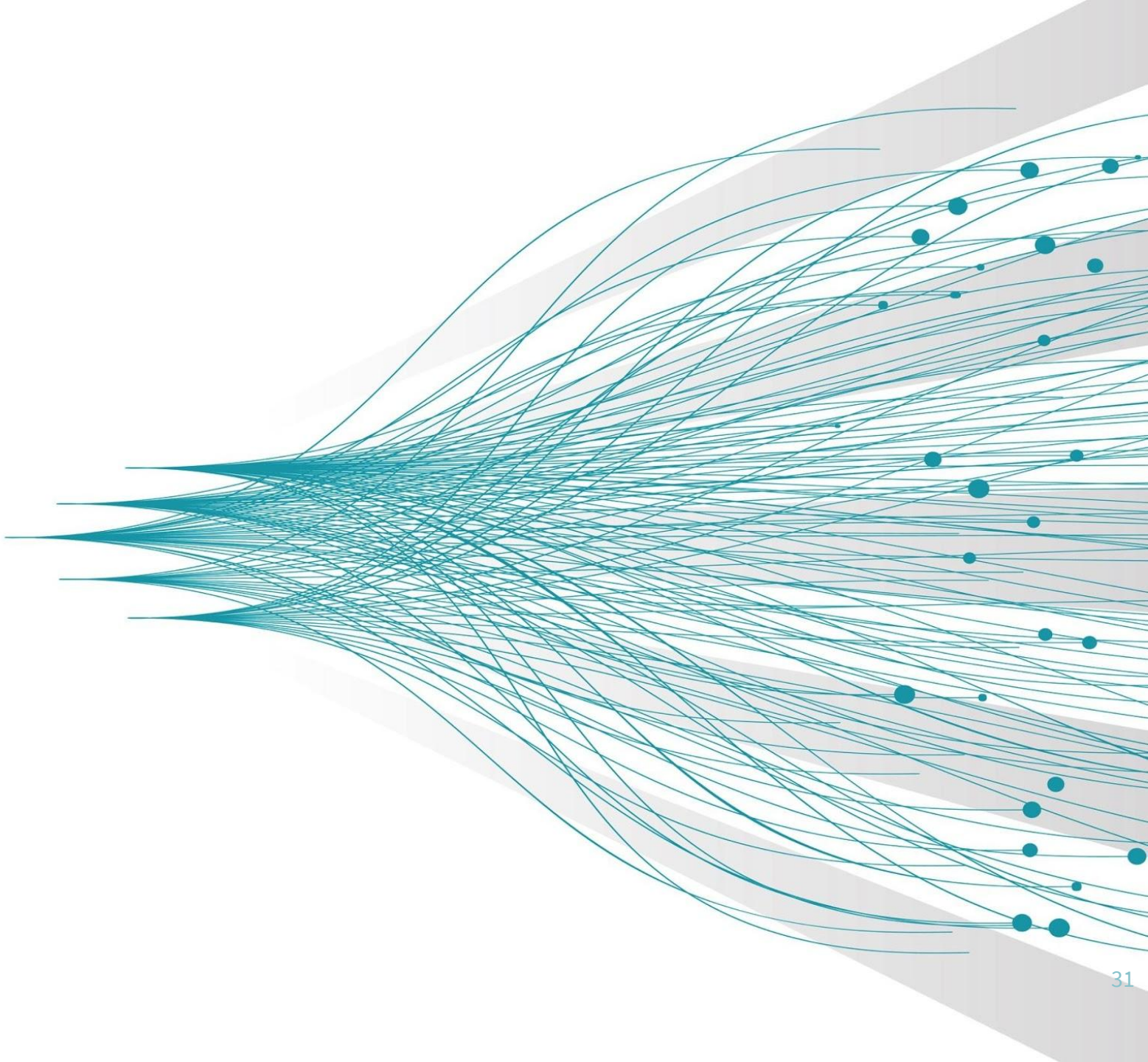
Collecting statistics

- ANALYZE table
  - native in Presto
  - Hive runtime no longer needed to collect stats
- Collecting table statistics on write
  - low overhead, the data is being processed in memory
- Stats for AWS Glue Catalog
  - Glue does not provide statistics support out of the box
  - exclusive from Starburst

Statistics in other connectors

- CBO initially supported with Hive connector only
- We added stats for RDBMS connectors (Starburst)
  - PostgreSQL
  - MySQL
  - SQL Server
  - Teradata
  - Oracle
- Enables CBO in federation use-cases

CBO Roadmap



What’s next

- Stats support
  - Improved stats for Hive and Glue, e.g. NDV estimates across partitions
  - Stats for NoSQL connectors
- Core CBO and engine enhancements
  - Involve connectors in optimizations (" *-pushdown ")
  - Peak memory-based decisions
  - Adjust cost model weights based on the hardware
  - Cost and stats for more operation types
  - Introduce Traits — consider alternative plans during optimizations
  - Late materialization
  - Adaptive optimizations

Further reading

https://prestosql.io/

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

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/

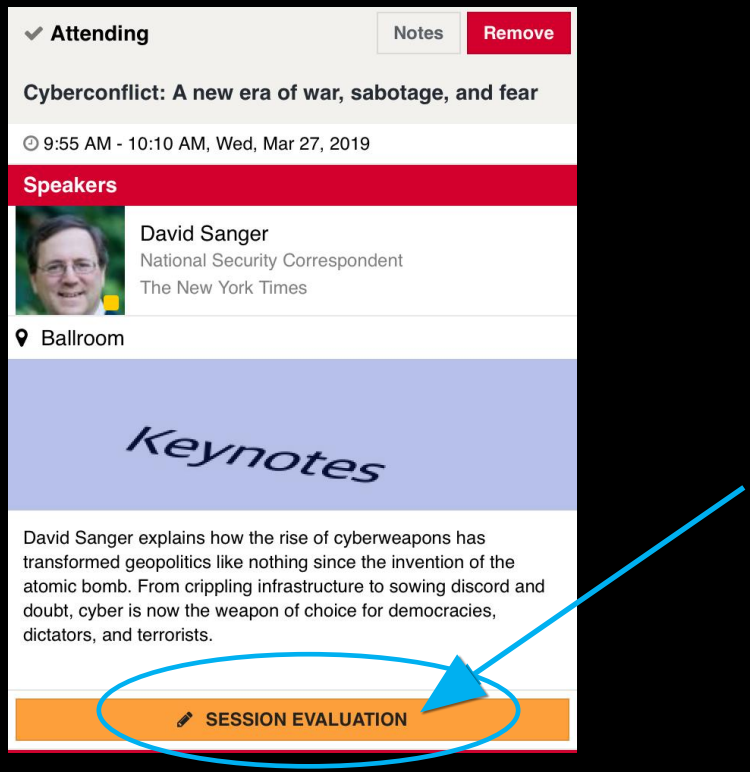
https://virtuslab.com/blog/benchmarking-spark-sql-presto-hive-bi-processing-google-cloud-dataproc/

Thank You!

Twitter: @starburstdata @prestosql
Blog: www.starburstdata.com/technical-blog/
Newsletter: www.starburstdata.com/newsletter

Rate today's session

Session page on conference website



O'Reilly Events App