Presto

Past, Present and Future

Martin Traverso - September 23, 2020

What’s Presto

- Distributed analytics query engine
  - Scalable
  - Fast and efficient
- Open source, community-based
- ANSI SQL
- Separation of compute and storage
- Connectors

Brief history

2012
We created Presto at Facebook

2013
We open-sourced Presto

2014
Teradata establishes first Commercial Presto Offering

2017
Starburst is created as The Presto Company

2018
We left Facebook, established PrestoSQL

2019
We established Presto Software Foundation 1-31-19, joined Starburst September 2019

2020
20,000+ commits, 2,500 community members on PrestoSQL Slack


Martin Traverso


David Phillips


Dain Sundstrom


Eric Hwang


Justin Borgman


Matthew Fuller


Piotr Findeisen


Kamil Bajda-Pawlikowski

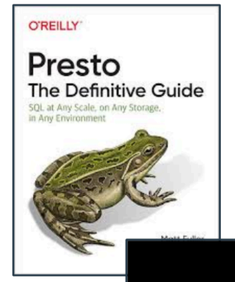

Karol Sobczak

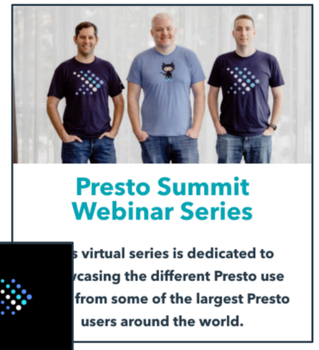

Grzegorz Kokosiński

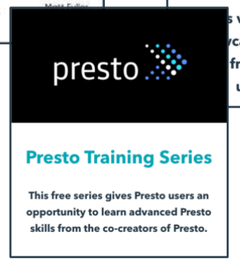

Martin Traverso


David Phillips


Dain Sundstrom







facebook

teradata.

Starburst
ANALYTICS ANYWHERE

Widely adopted by all major industries

Community

- 2750+ users and developers in Slack

  https://prestosql.io/slack.html

- Global community
  - In 2019, conferences in US, Japan, India and Israel
- 20,500+ commits (>25% in the last 18 months)

  https://github.com/prestosql/presto

In the beginning…

- “It is a good day when I can run 6 Hive queries” - a Facebook data scientist
- Peregrine

  https://xrds.acm.org/article.cfm?aid=2331056

… and Presto was born

- Initial goal: “make interactive analytics over Hive data better”
- Goals
  - Fast **AND** correct
  - Can run at Facebook scale and keep up
  - Standard SQL
  - Make it open source
  - For the long term (20+ years)

Early days

- First production version in ~6 months
  - Support for SELECT with JOINs and aggregations
  - We rewrote everything at least once in between
- Replaced Peregrine by July 2013

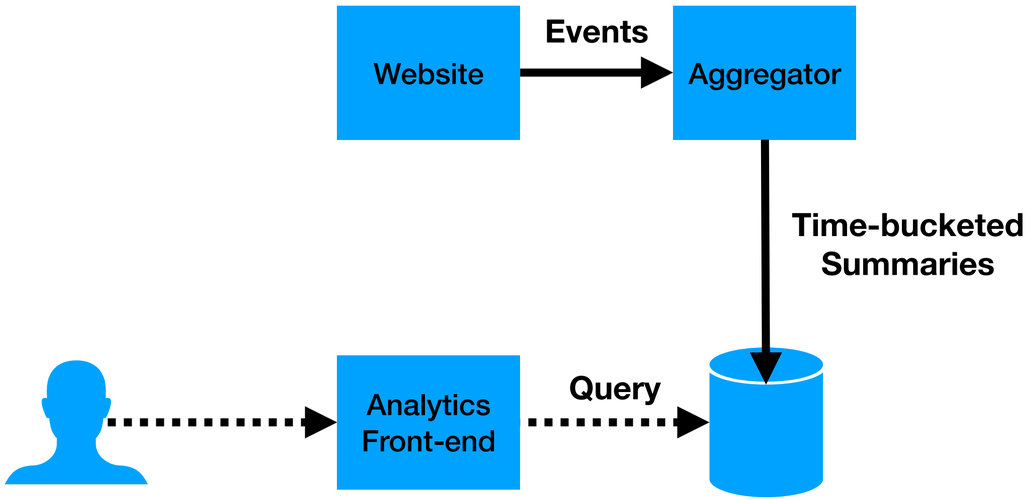
Some lessons learned

- Missteps
  - Modeling imports as materialized views, syntax and all
  - Approximate queries (a la BlinkDB)
- Successes
  - ANSI SQL
  - HTTP
  - Plugins

Plugins

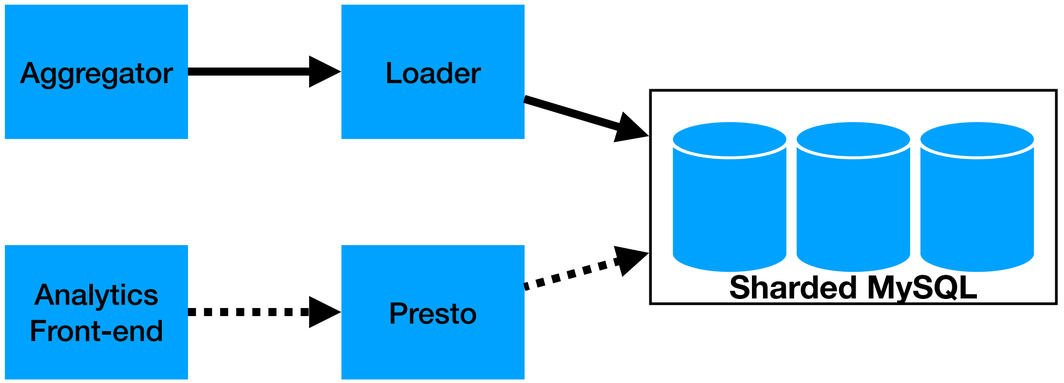
- Why?
  - Clean separation between engine and storage
  - FB was running forked version of Hive & HDFS
  - We wanted to open source Presto eventually
- BEST. DECISION. EVER

Presto for user-facing apps



- “UPSERT” semantics
- Normalized data set
  - 15-way joins
- Large data set, but very selective queries
- Interactive latencies (< 5s)
- 24/7 availability

Presto for user-facing apps



- Data organized to match expected queries
- Lookup Joins
- Latency improvements
  - avoid check-sleep loops
- JDBC-based connectors
- Regional clusters with global query router

Presto for A/B testing framework

- Replace analytics backend of A/B testing framework
- Requirements
  - Reliable data loads
  - 5-10 minute load latency
  - Consistent performance
- Seconds to minutes
- Very large data sets

Presto for A/B testing framework

- Insert
- Delete
- Co-located joins
- Intra-node parallelism

Presto for batch workloads

Challenges

- Capacity and expectations
- Per cluster scalability limits
- Deploying risky changes
- UDFs
- Long running queries vs failures
- High memory queries

Presto Gained

- Resource groups
- Local scheduling improvements
- Server-controlled session properties
- Lambda expressions
- Grouped execution

Presto for batch workloads

- By end of 2018…
  - > 50% of workload running on Presto
  - > 85% of new jobs written for Presto
  - Largest deployment of Presto at FB

Recent improvements

- Dynamic filtering and partition pruning

  https://prestosql.io/blog/2019/06/30/dynamic-filtering.html

  https://prestosql.io/blog/2020/06/14/dynamic-partition-pruning.html
- Storage Caching
- Web UI security, security in general
- Connectors
  - Apache Iceberg, Apache Druid, Apache Pinot, BigQuery, Google Sheets, Oracle, etc.
- ARM64

  https://prestosql.io/blog/2019/12/23/Presto-Experiment-with-Graivton-Processor.html
- Variable precision date times (`timestamp(p)`, `time(p)`, etc.)
- Much, much more:

  https://prestosql.io/blog/2020/01/01/2019-summary.html

  https://prestosql.io/blog/2020/05/15/state-of-presto.html

Roadmap

- Materialized Views
- Dynamically resolved functions
- Hive View support
- Optimized Parquet reader

Getting involved

- Slack
  - https://prestosql.io/slack.html
  - #troubleshooting channel
- Contribute code:
  - https://prestosql.io/development
  - “good first issue” tag on github
- File issues/bugs:
  - https://github.com/prestosql/presto
- Write blog posts:
  - https://prestosql.io/blog

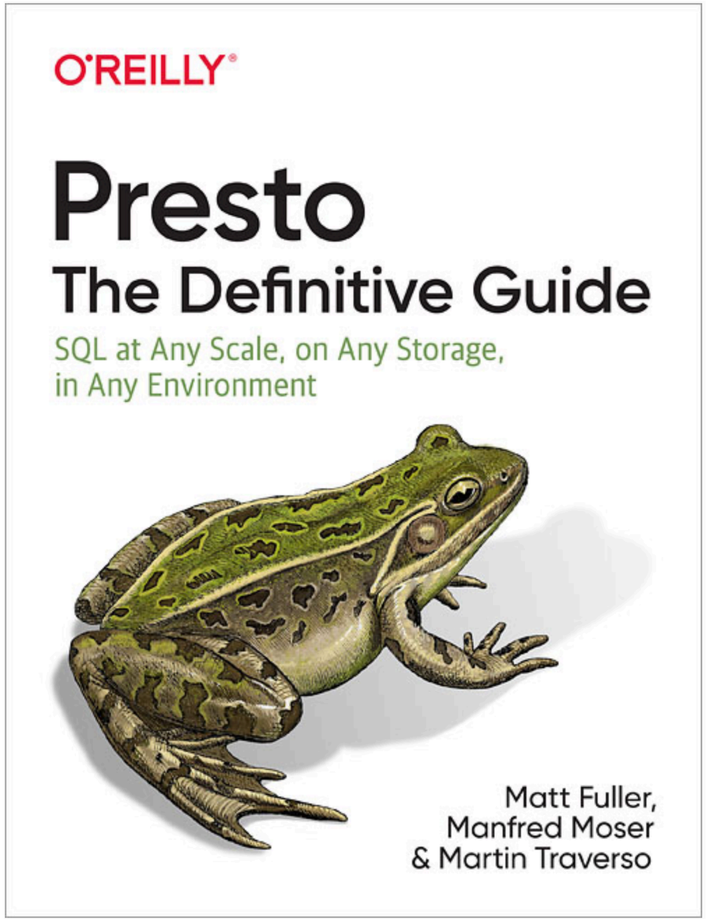
Resources

- Presto - The Definitive Guide. Free download!

  https://www.starburstdata.com/oreilly-presto-guide-download

- Presto: SQL on Everything - IEEE International Conference on Data Engineering

  https://prestosql.io/paper



Questions?