Presto: Interacting with petabytes of data at Facebook

By Lydia Chan on Wednesday, November 6, 2013 at 10:01am

By Martin Traverso

Background

Facebook is a data-driven company. Data processing and analytics are at the heart of building and delivering products for the 1 billion+active users of Facebook. We have one of the largest data warehouses in the world, storing more than 300 petabytes. The data is used for a wide range of applications, from traditional batch processing to graph analytics [1], machine learning, and real-time interactive analytics.

For the analysts, data scientists, and engineers who crunch data, derive insights, and work to continuously improve our products, the performance of queries against our data warehouse is important. Being able to run more queries and get results faster improves their productivity.

Facebook's warehouse data is stored in a few large Hadoop/HDFS-based clusters. Hadoop MapReduce [2] and Hive are designed for large-scale, reliable computation, and are optimized for overall system throughput. But as our warehouse grew to petabyte scale and our needs evolved, it became clear that we needed an interactive system optimized for low query latency.

In Fall 2012, a small team in the Facebook Data Infrastructure group set out to solve this problem for our warehouse users. We evaluated a few external projects, but they were either too nascent or did not meet our requirements for flexibility and scale. So we decided to build Presto, a new interactive query system that could operate fast at petabyte scale.

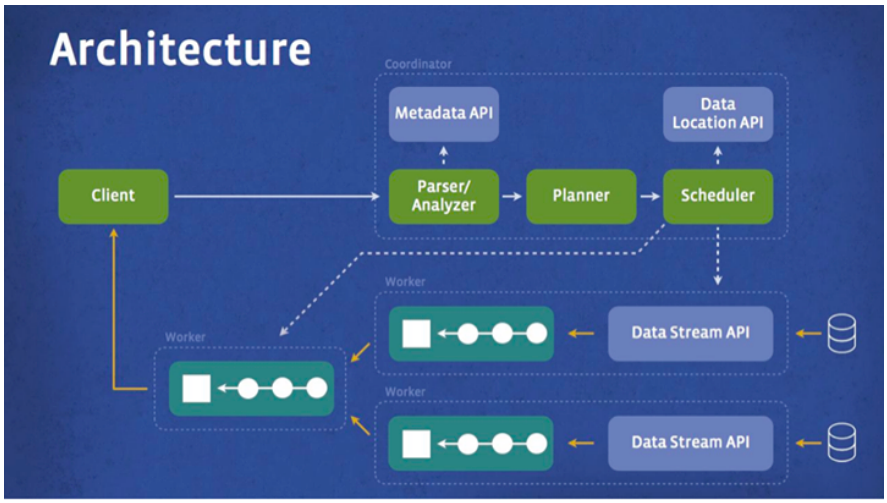
In this post, we will briefly describe the architecture of Presto, its current status, and future roadmap.

Architecture

Presto is a distributed SQL query engine optimized for ad-hoc analysis at interactive speed. It supports standard ANSI SQL, including complex queries, aggregations, joins, and window functions.

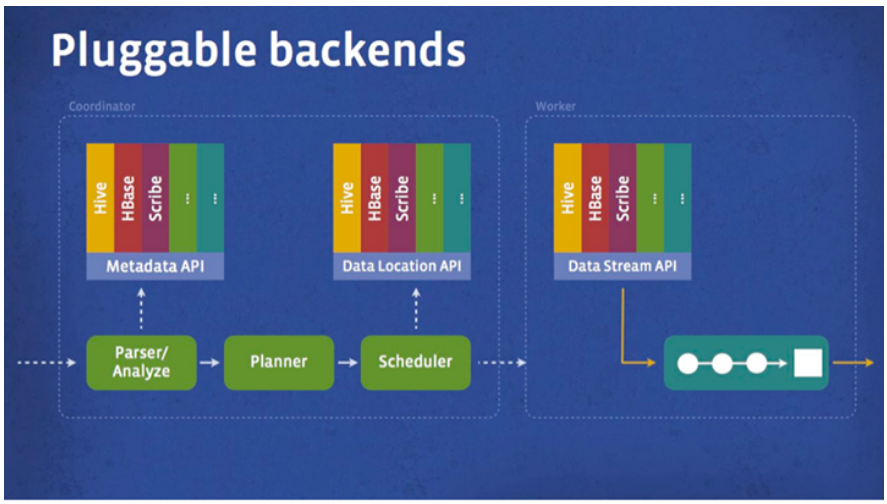
The diagram below shows the simplified system architecture of Presto. The client sends SQL to the Presto coordinator. The coordinator parses, analyzes, and plans the query execution. The scheduler wires together the execution pipeline, assigns work to nodes closest to the data, and monitors progress. The client pulls data from output stage, which in turn pulls data from underlying stages.

The execution model of Presto is fundamentally different from Hive/MapReduce. Hive translates queries into multiple stages of MapReduce tasks that execute one after another. Each task reads inputs from disk and writes intermediate output back to disk. In contrast, the Presto engine does not use MapReduce. It employs a custom query and execution engine with operators designed to support SQL semantics. In addition to improved scheduling, all processing is in memory and pipelined across the network between stages. This avoids unnecessary I/O and associated latency overhead. The pipelined execution model runs multiple stages at once, and streams data from one stage to the next as it becomes available. This significantly reduces end-to-end latency for many types of queries.



The Presto system is implemented in Java because it's fast to develop, has a great ecosystem, and is easy to integrate with the rest of the data infrastructure components at Facebook that are primarily built in Java. Presto dynamically compiles certain portions of the query plan down to byte code which lets the JVM optimize and generate native machine code. Through careful use of memory and data structures, Presto avoids typical issues of Java code related to memory allocation and garbage collection. (In a later post, we will share some tips and tricks for writing high-performance Java system code and the lessons learned while building Presto.)

Extensibility is another key design point for Presto. During the initial phase of the project, we realized that large data sets were being stored in many other systems in addition to HDFS. Some data stores are well-known systems such as HBase, but others are custom systems such as the Facebook News Feed backend. Presto was designed with a simple storage abstraction that makes it easy to provide SQL query capability against these disparate data sources. Storage plugins (called connectors) only need to provide interfaces for fetching metadata, getting data locations, and accessing the data itself. In addition to the primary Hive/HDFS backend, we have built Presto connectors to several other systems, including HBase, Scribe, and other custom systems.



Current status

As mentioned above, development on Presto started in Fall 2012. We had our first production system up and running in early 2013. It was fully rolled out to the entire company by Spring 2013. Since then, Presto has become a major interactive system for the company's data warehouse. It is deployed in multiple geographical regions and we have successfully scaled a single cluster to 1,000 nodes. The system is actively used by over a thousand employees, who run more than 30,000 queries processing one petabyte daily.

Presto is 10x better than Hive/MapReduce in terms of CPU efficiency and latency for most queries at Facebook. It currently supports a large subset of ANSI SQL, including joins,

left/right outer joins, subqueries, and most of the common aggregate and scalar functions, including approximate distinct counts (using HyperLogLog) and approximate percentiles (based on quantile digest). The main restrictions at this stage are a size limitation on the join tables and cardinality of unique keys/groups. The system also lacks the ability to write output data back to tables (currently query results are streamed to the client).

Roadmap

We are actively working on extending Presto functionality and improving performance. In the next few months, we will remove restrictions on join and aggregation sizes and introduce the ability to write output tables. We are also working on a query "accelerator" by designing a new data format that is optimized for query processing and avoids unnecessary transformations. This feature will allow hot subsets of data to be cached from backend data store, and the system will transparently use cached data to "accelerate" queries. We are also working on a high performance HBase connector.

Open source

After our initial Presto announcement at the Analytics @ WebScale conference in June 2013 [3], there has been a lot of interest from the external community. In the last couple of months, we have released Presto code and binaries to a small number of external companies. They have successfully deployed and tested it within their environments and given us great feedback.

Today we are very happy to announce that we are open-sourcing Presto. You can check out the code and documentation on the site below. We look forward to hearing about your use cases and how Presto can help with your interactive analysis.

http://prestodb.io/
https://github.com/facebook/presto

The Presto team within Facebook Data Infrastructure consists of Martin Traverso, Dain Sundstrom, David Phillips, Eric Hwang, Nileema Shingte and Ravi Murthy.

Links

[1] Scaling Apache Giraph to a trillion edges. https://www.facebook.com/notes/facebook-engineering/scaling-apache-giraph-to-a-trillion-edges/10151617006153920
[2] Under the hood: Scheduling MapReduce jobs more efficiently with Corona https://www.facebook.com/notes/facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920
[3] Video of Presto talk at Analytics@Webscale conference, June 2013 https://www.facebook.com/photo.php?v=10202463462128185


Chuka Uchenna Ikokwu This is maaadd! Great job FB! Thanks for making my life 10x easier!
May 6 at 3:03pm


Troy Blake Interesting information. I like the comparison between Hive and Presto.
1 · November 11, 2013 at 10:35am

John Getson So what is reason behind the issue where only the header information is loaded... no variable or fluid info is being displayed at all?
November 8, 2013 at 8:15am


Adam Bunch Is there any documentation on the hardware you used, i have yet to find any, but may be missing it.
November 8, 2013 at 7:00am


Kangmo Kim very cool.
1 · November 7, 2013 at 4:04pm


JunYoung Kim It's amazing!!!! Congratulations! !!
1 · November 6, 2013 at 4:17pm


Ray Long Thank you for giving back to the Open Source community and releasing Presto!
1 · November 6, 2013 at 12:58pm


Aravind Rao curious, may i please know which SQL clients you guys are using for querying DW with Presto and any new plugins to be installed?
1 · November 6, 2013 at 10:52am


Ashutosh Misra It's amazing..
April 15, 2014 at 10:58pm


Sushant Wason Impressive stuff!
March 21, 2014 at 3:58pm


盖晓阳 amazing ..
March 13, 2014 at 1:12am


Brhane Hntsa turly greatpt and expected morefrom
March 5, 2014 at 3:33pm


Sb Gowtham facebook is always best for dealing with BigData
February 26, 2014 at 11:07pm


Aravindan Balan Great work ppl Way to go
January 19, 2014 at 12:48pm


Mathias Bogaert I've added this to my Hadoop Ansible playbook:
https://github.com/analytically/hadoop-ansible
January 10, 2014 at 2:24am


Tomáš Sako Thanks for this interesting post and for open-sourcing. Big data engineers usually do not share too much technical details However, from the benchmark point of view, to which version of Hive is Presto 10x better?
Wish you good luck.
January 9, 2014 at 9:58am


Andy Barr Beautiful! Looking forward to playing with this in my own project Thanks Facebook!
January 5, 2014 at 8:11pm


Chuck Connell Something that does not seem to be explained anywhere... I have a running Hadoop/Hive cluster (with Cloudera CDH4). Does Presto run directly on the same datanodes as Hadoop? Or does it install on its own set of compute nodes, and just read data from my... See More
December 26, 2013 at 12:14pm


Shyamala Lokre-pitre Nice post.
December 12, 2013 at 10:48pm


Victor F. E-b Checka Selene Castro...!!
December 11, 2013 at 5:59pm


Andalib Shadani You guys are really awsum..putting all your efforts to make work easier! A Big Thanks
December 1, 2013 at 12:04am


Joey Zhang luck,very good
November 28, 2013 at 12:25am


Neel Adhikary Thank you very much for open sourcing this great system.
November 19, 2013 at 10:05pm


Stelios Doulakis Great job..
November 19, 2013 at 4:03am


Ganesan Senthilvel Great product; Out of box thinking from mapreduce algorithm; Open source too..
November 16, 2013 at 9:58pm


Wenshuo Han Good work. Love to see a bit of graph comparision regarding the performance between Presto and Hive/MR
November 13, 2013 at 8:48pm

Andrew Aik-Xung Lee Crazy..... now I have a reason to build a Linux machine... Or is it time for me to get a build with Mac OS X?? Choices, choices
November 12, 2013 at 12:47pm

Peng Du Is this something like Impala of Cloudera?
November 11, 2013 at 5:32pm

Erik Nolke Thanks, I am excited to try it out!
November 11, 2013 at 10:35am

Jimmy Cage The Plant Manager is responsible for general supervision of all phases of plant operations including: production, quality control, maintenance, receiving, and shipping. Responsibilities also include recruiting, hiring and training personnel and facili... See More
November 11, 2013 at 6:57am

Akshat Thakar open source is way to build great softwares!
November 11, 2013 at 12:49am

Tarık Yılmaz Great work! also HBase connector woow
November 9, 2013 at 2:24am

Lionel Silberman Exciting. Wonder if multiple processor/memory machines per data node and they create a hierarchy that respects data locality? And, if the join doesn't fit in memory, does it spill to disk or fail?
November 8, 2013 at 10:25am

Seiji Kasuya Jesus Christ！！
November 8, 2013 at 5:48am

Senthil Ganesh Great show!
November 8, 2013 at 1:41am

Tousif Khazi very nice..
November 7, 2013 at 11:10pm

Gustavo De Mari nice job! thanks for open source
November 7, 2013 at 3:38pm

Alex Wiederkehr crazy
November 7, 2013 at 7:41am

Lio Sean Great job guys! Really looking forward to the open source
November 7, 2013 at 5:49am

Bhupesh Khanna Congratulations Martin Traverso and Team, Great Work !!!
November 7, 2013 at 3:53am

Ankit Dhingra Facebook Engineering Could you shed some light on the other projects that you evaluated before starting to build your own?
November 7, 2013 at 1:10am

Ámít Khíwál Insane! Keep up the good work guys!
November 6, 2013 at 11:58pm

View more comments