

Pivotal HD: HAWQ

A True SQL Engine For Hadoop

Pivotal HD: HAWQ

A True SQL Engine For Hadoop

Today's most successful companies use data to their advantage. The data are no longer easily quantifiable facts, such as point of sale transaction data. Rather, these companies retain, explore, analyze, and manipulate all the available information in their purview.

Ultimately, they search for evidence of facts, insights that lead to new business opportunities or which leverage their existing strengths. This is the business value behind what is often referred to as Big Data.

Hadoop has proven to be a technology well suited to tackling Big Data problems. The fundamental principle of the Hadoop architecture is to move analysis to the data, rather than moving the data to a system that can analyze it. Ideally, Hadoop takes advantage of the advances in commodity hardware to scale in the way companies want.

There are two key technologies that allow users of Hadoop to successfully retain and analyze data: HDFS and MapReduce. HDFS is a simple but extremely powerful distributed file system. It is able to store data reliably at significant scale. HDFS deployments exist with thousands of nodes storing hundreds of petabytes of user data.

MapReduce is parallel programming framework that integrates with HDFS. It allows users to express data analysis algorithms in terms of a small number of functions and operators, chiefly, a map function and a reduce function.

The success of MapReduce is a testament to the robustness of HDFS — both as a system to restore and access data, and as an application programming interface (API) for Big Data analysis frameworks.

While MapReduce is convenient when performing scheduled analysis or manipulation of data stored on HDFS, it is not suitable for interactive use: it is too slow and lacks the expressive power required.

SQL

SQL is the language of data analysis. No language is more expressive, or more commonly used, for this task. Companies that bet big on Hadoop — such as Facebook, Twitter and especially Yahoo! — understood this immediately. Their solution was Hive, a SQL-like query engine which compiles a limited SQL dialect to MapReduce. While it addresses some of the expressivity shortcomings of MapReduce, it compounds the performance problem.

Many Hadoop users are disenfranchised from true SQL systems. Either, it is hard to get data into them from a Hadoop system; or, they cannot scale anywhere near as well as Hadoop; or, they become prohibitively expensive as they reach the scale of common Hadoop clusters.

HAWQ: THE MARRIAGE OF HADOOP AND PARALLEL SQL DATABASE TECHNOLOGY

HAWQ is a parallel SQL query engine that combines the key technological advantages of the industry-leading Pivotal Analytic Database with the scalability and convenience of Hadoop. HAWQ reads data from and writes data to HDFS natively. HAWQ delivers industry-leading performance and linear scalability. It provides users the tools to confidently and successfully interact with petabyte range data sets. HAWQ provides users with a complete, standards compliant SQL interface.

By using the proven parallel database technology of the Pivotal Analytic Database, HAWQ has been shown to be consistently tens to hundreds of times faster than all Hadoop query engines in the market today.

HAWQ ARCHITECTURE

HAWQ has been designed from the ground up to be a massively parallel SQL processing engine optimized specifically for analytics with full transaction support. HAWQ breaks complex queries into small tasks and distributes them to query processing units for execution. Combining a world-class cost based query optimizer, a leading edge network interconnect, a feature rich SQL and analytics interface, and a high performance execution run time with a transactional storage sub system, HAWQ is the only Hadoop query engine offering such technology.

HAWQ architecture is shown in [figure 1](#). HAWQ's basic unit of parallelism is the segment instance. Multiple segment instances work together on commodity servers to form a single parallel query processing system. When a query is submitted to the HAWQ master, it is optimized and broken into smaller components and dispatched to segments that work together to deliver a single result set. All operations—such as table scans, joins, aggregations, and sorts—execute in parallel across the segments simultaneously. Data from upstream components are transmitted to downstream components through the extremely scalable UDP interconnect.

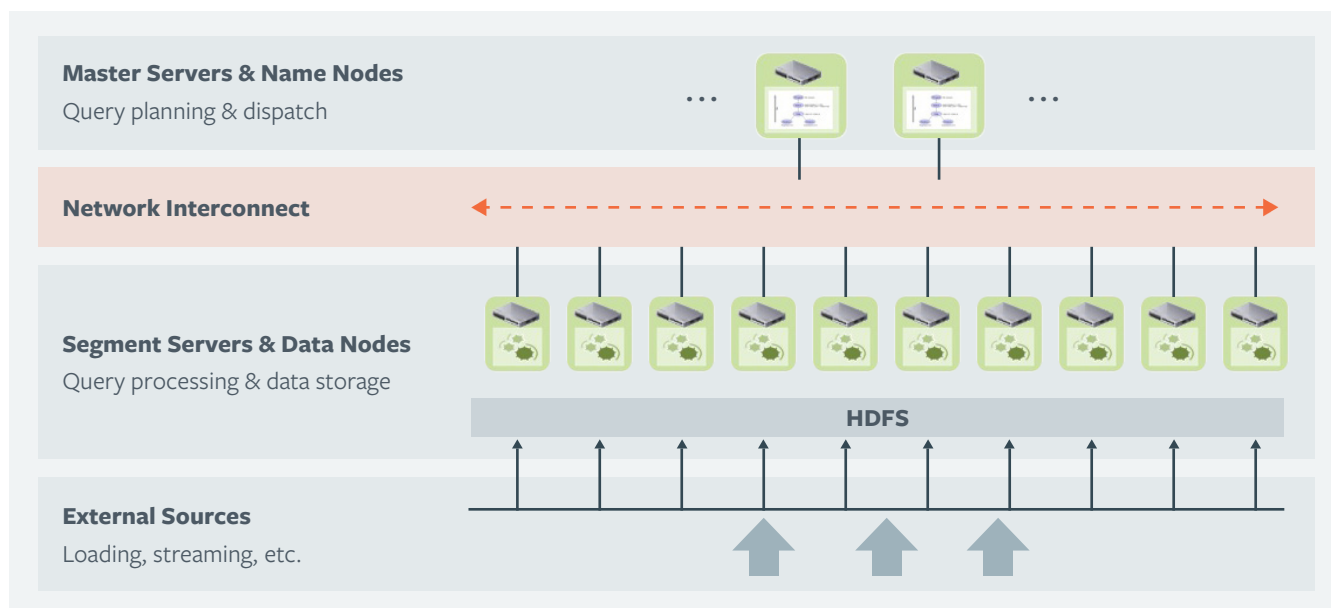


Figure 1: *HAWQ architecture*

HAWQ has been designed to be resilient and high performance, even when executing the most complex of queries. All operations, irrespective of the amount of data they operate upon, can succeed because of HAWQ's unique ability to operate upon streams of data through what we call dynamic pipelining. HAWQ is designed to have no single point of failure. User data is stored on HDFS, ensuring that it is replicated. HAWQ works with HDFS to ensure that recovery from hardware failures is automatic and online. Internally, system health is monitored continuously. When a server failure is detected, it is removed from the cluster dynamically and automatically while the system continues serving queries. Recovered servers can be added back to the system on the fly. At the master, HAWQ uses its own metadata replication system to ensure that metadata is highly available.

Comparison With Other Parallel Database Architectures

HAWQ's architecture is different from other parallel databases. HAWQ is metadata driven in that much of the behavior of the system is governed by data registered through HAWQ. This is transparent to the user. Metadata is managed by what is called a Universal Catalog Service. This is a metadata store that resides at the master. The Universal Catalog Service is replicated to ensure high availability of the data.

When a query plan is generated, HAWQ dispatches the plan to workers in the Hadoop cluster, along with a payload of metadata they need to execute the plan. This approach is optimal for any system seeking to scale to thousands of nodes.

Significantly, HAWQ's query optimizer generates the query plan at the master. Worker nodes in the cluster merely evaluate it. Many other parallel databases instead split queries into global operations and local operations, with the workers generating

optimal query plans for the local operations only. While this approach is technically easier, it misses many important query optimization opportunities. Much of HAWQ's amazing performance comes from its big picture view of query optimization.

HAWQ is also very resilient in the face of challenges common to Big Data. The query engine is able to intelligently buffer data, spilling intermediate results to disk when necessary. For performance, HAWQ delivers data to the local disk rather than HDFS. The result is that HAWQ is able to perform joins, sorts and OLAP operations on data well beyond the total size of volatile memory in the cluster.

KEY FEATURES AND BENEFITS OF HAWQ

EXTREME SCALE AND ONE STORAGE SYSTEM

- With HAWQ, SQL can scale on Hadoop. HAWQ is designed for petabyte range data sets.
- Data is stored directly on HDFS, providing all the convenience of Hadoop.

INDUSTRY LEADING PERFORMANCE WITH DYNAMIC PIPELINING

- HAWQ's cutting edge query optimizer is able to produce execution plans which it believes will optimally use the cluster's resources, irrespective of the complexity of the query or size of the data
- HAWQ uses dynamic pipelining technology to orchestrate execution of a query
- Dynamic pipelining is a parallel data flow framework which uniquely combines:
 - An adaptive, high speed UDP interconnect, developed for and used on Pivotal's 1000 server Analytics Work Bench
 - A run time execution environment, tuned for Big Data work loads, which implements the operations which underlie all SQL queries
 - A run time resource management layer, which ensures that queries complete, even in the presence of very demanding queries on heavily utilized clusters
 - A seamless data partitioning mechanism which groups together the parts of a data set which are often used in any given query
- Extensive performance studies show that HAWQ is orders of magnitude faster than existing Hadoop query engines for common Hadoop, analytics and data warehousing workloads

ELASTIC FAULT TOLERANCE AND TRANSACTION SUPPORT

- HAWQ's fault tolerance, reliability and high availability features tolerate disk level and node level failures.
- HAWQ supports transactions, a first for Hadoop. Transactions allow users to isolate concurrent activity on Hadoop and to rollback modifications when a fault occurs.

A TOOLKIT FOR DATA MANAGEMENT AND ANALYSIS

- HAWQ can coexist with MapReduce, HBase and other database technologies common in Hadoop environments.
- HAWQ supports traditional online analytic processing (OLAP) as well as advanced machine learning capabilities, such as supervised and unsupervised learning, inference and regression.
- HAWQ supports GPXF, a unique extension framework which allows users to easily interface HAWQ custom formats and other parts of the Hadoop ecosystem.

TRUE SQL CAPABILITIES

- HAWQ's cost based query optimizer can effortlessly find optimal query plans for the most demanding of queries, such as queries with more than thirty joins
- HAWQ is SQL standards compliant, including features such as correlated sub queries, window functions, rollups and cubes, a broad range of scalar and aggregate functions and much more
- HAWQ is ACID compliant
- Users can connect to HAWQ via the most popular programming languages, and it also supports ODBC and JDBC. This means that most business intelligence, data analysis and data visualization tools work with HAWQ out of the box.

PERFORMANCE RESULTS

As part of the development of HAWQ, we selected customer workloads which reflected both real world and aspirational use of Hadoop, whether via MapReduce, Hive or other technologies. We used these workloads to validate the performance and scalability of HAWQ.

Performance studies in isolation have little real world meaning, so HAWQ was benchmarked against Hive and a new parallel SQL engine for Hadoop, called Impala¹. So as to ensure a fair comparison, all systems were deployed on the Pivotal Analytics Workbench² (AWB). The AWB is configured to realize the best possible performance from Hadoop³. Of the workloads collected during research and development, five queries and data sets were used for the performance comparison. Two criteria were used: the queries must reflect different uses for Hadoop today and the queries must complete on Hive, Impala and HAWQ⁴.result is that HAWQ is able to perform joins, sorts and OLAP operations on data well beyond the total size of volatile memory in the cluster.

QUERY TYPE	HAWQ (SECS)	HIVE	SPEEDUP	IMPALA	SPEEDUP
User intelligence	4.2	198	47x	37	9x
sales analysis	8.7	161	19x	596	69x
click analysis	2.0	415	208x	50	25x
data exploration	2.7	1285	476x	55	20x
bi drill down	2.8	1815	648x	59	21x

Table 1: Performance results for five real world queries on HAWQ, hive and Impala

The performance experiment took place on a 60 node deployment with the AWB. This size deployment was selected as it was the point at which we saw most stability of performance from Impala. HAWQ has been verified for stable performance well beyond 200 nodes.

¹ <https://github.com/cloudera/impala/>, in beta at the time of writing

² <http://www.analyticsworkbench.com/>

³ http://www.greenplum.com/sites/default/files/Greenplum-Analytics-Workbench-Whitepaper_o.pdf

⁴ Many queries in the aspirational class used in excess of eight joins, but these queries did not complete on Hive or Impala

Results of the performance comparison are contained in table 1. It is clear that HAWQ's performance is remarkable when compared to other Hadoop options currently available. One of the most important aspects of any technology running on top of Hadoop is its ability to scale. Absolute performance alone is considered secondary. As such, an experiment was undertaken to verify the performance at 15, 30, and 60 nodes.

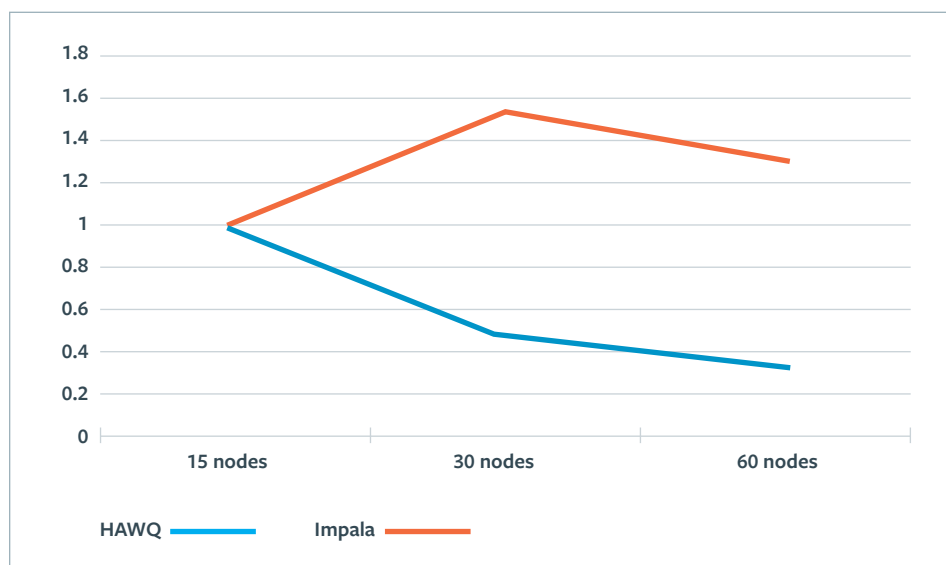


Table 2: *Relative scalability of HAWQ and Impala*

Table 2 shows the relative scalability for HAWQ and Impala. For a data set of a consistent size, the run time of queries detailed in table 1 practically halved each time the number of servers is doubled. HAWQ has been shown to continue to scale like this beyond 200 nodes. Surprisingly, Impala's performance is better with 15 nodes than 30 or 60 nodes: when the same amount of data is given four times the amount of computing resources, Impala spent 30% longer returning results.

CONCLUSION

SQL is an extremely powerful way to manipulate and understand data. Until now, Hadoop's SQL capability has been limited and impractical for many users. HAWQ is the new benchmark for SQL on Hadoop — the most functionally rich, mature, and robust SQL offering available. Through Dynamic Pipelining and the significant innovations within the Greenplum Analytic Database, it provides performance previously unimaginable. HAWQ changes everything.

LEARN MORE

To learn more about our products, services and solutions, visit us at pivotal.io

Pivotal offers a modern approach to technology that organizations need to thrive in a new era of business innovation. Our solutions intersect cloud, big data and agile development, creating a framework that increases data leverage, accelerates application delivery, and decreases costs, while providing enterprises the speed and scale they need to compete.

Pivotal 3495 Deer Creek Road Palo Alto, CA 94304 pivotal.io

Pivotal, Pivotal CF, and Cloud Foundry are trademarks and/or registered trademarks of Pivotal Software, Inc. in the United States and/or other Countries. All other trademarks used herein are the property of their respective owners. © Copyright 2015 Pivotal Software, Inc. All rights reserved. Published in the USA. PVTL-WP-02/15