kognitio

Whitepaper

SQL on Apache® Hadoop® benchmarks using the TPC-DS query set

As SQL on Hadoop moves from evaluation into production, many organizations have issues using the tools in the standard Hadoop distributions to support enterprise level SQL on data in Hadoop.

This is caused by a number of issues including:

- **SQL maturity** – some products cannot handle all the SQL generated by developers and/or third party tools. They either do not support the SQL, or produce very poor query plans
- **Query performance** – queries that are supported perform poorly even under single user workload
- **Concurrency** – products cannot handle concurrency well in terms of performance and give errors when under load

These issues along with the mixed workloads required to support enterprise BI and complex analytics are investigated in this benchmarking paper.

The TPC-DS benchmark is a well-respected, widely used query set that is representative of the type of queries that seem to be the most problematic.

The TPC framework is also designed for benchmarking concurrent workloads.

The TPC-DS query generator was used in the benchmark to randomize each query (via parameter selection) and randomize the query submission order in each concurrent stream.

The benchmark can be interpreted as posing three distinct performance questions:

- **Can the platform run the queries?**
  = functional testing over 1GB data
- **Can the platform perform at scale?**
  = single stream over 1TB data
- **How does the platform perform under load?**
  = concurrent multiple streams over 1TB

The platforms included in this benchmark are:

- Apache Impala (version 2.6.0)
- Kognitio (version 8.1.50)
- Apache Spark™ (version 2.0 beta)

Each platform utilized the same 12 node infrastructure running Cloudera CDH 5.8.2.

Can the platform run the queries?

One of the major pain points in SQL on Hadoop adoption is the need to migrate existing workloads to run over data in Hadoop.

The breadth of SQL supported by each platform was investigated. Each of the 99 TPC-DS queries was qualified as one of the following:

- Runs ‘out of the box’ (no changes needed)
- Minor syntax changes – such as removing reserved words or ‘grammatical’ changes
- Long running – SQL compiles but query doesn’t come back within 1 hour
- No support – syntax not currently supported

Platform	Impala	Kognitio	Spark
Out of box	55	76	72
Minor changes	18	23	27
Long running	2		
No support	24		

The table above shows that for functional testing (over 1GB of data) both Kognitio and Spark can execute all 99 TPC-DS queries. This is a big improvement for Spark from version 1.6 where it could only execute 51 out of the 99 queries. Impala has some way to go with SQL support: OLAP grouping sets, some sub-query functionality and set functions are still lacking.

Can the platform perform at scale?

Running a single query stream at a 1TB scale is a starting point for assessing platform performance.

For Impala and Spark the 1TB data set was held in Apache Hive™ tables using parquet as the storage format, with larger tables partitioned on columns most frequently used in joins within the queries. For Kognitio the data was held in view images, with large data hashed on the main join columns.

For a single query stream Kognitio outperforms both Impala and Spark as can be seen from the overview table below and the 3 speed comparisons in figure 1.

fig 1a - **Kognitio vs Impala**

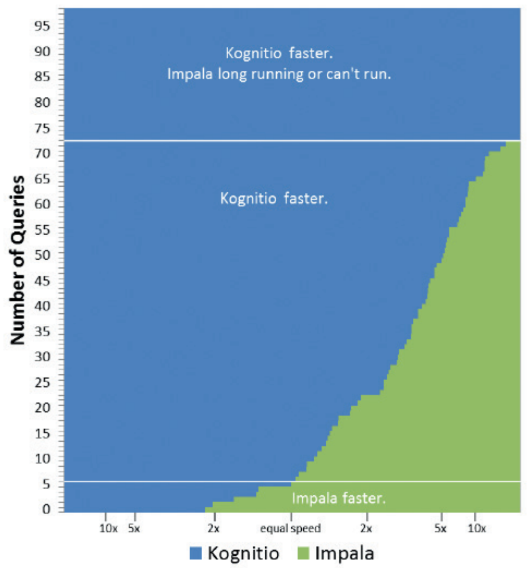


fig 1b - **Kognitio vs Spark**

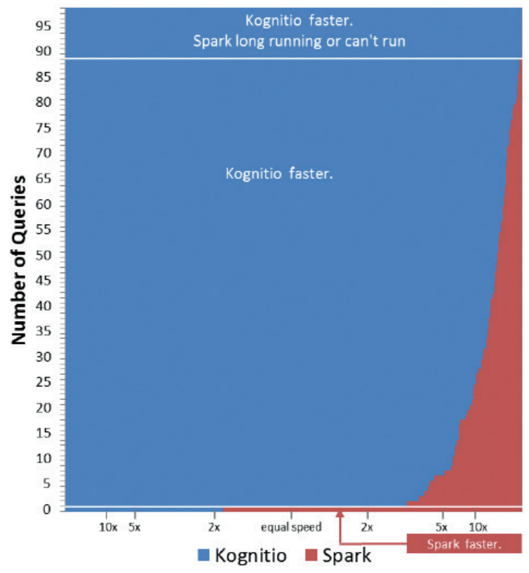
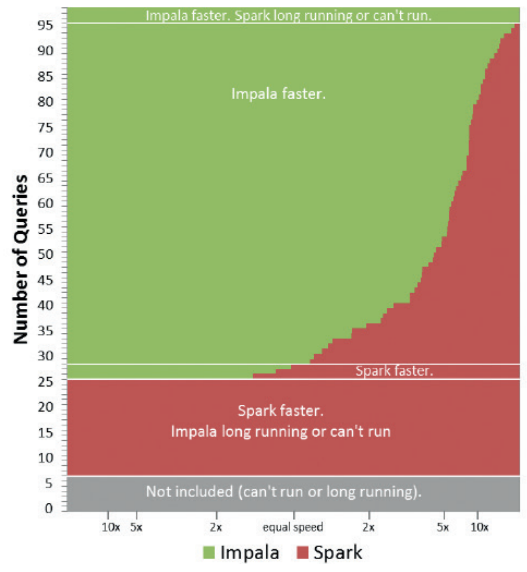


fig 1c - **Impala vs Spark**



Platform	Impala	Kognitio	Spark
Queries run	73	99	89
Long running	2		10
No support	24		
Fastest query count	6	92	1

Query overview - single stream at 1TB

Kognitio runs all 99 queries without issue at this scale and is the fastest platform for 92 out of 99 queries.

Each plot in figure 1 represents the relative speed between two of the platforms; Kognitio (blue), Impala (green) and Spark (red). Each query is represented by a horizontal block.

The faster platform for a given query gets the largest proportion of the block. Therefore, overall, the more a colour dominates the better that platform performs.

The solid blue block at the top of figure 1(a) represents the 26 queries that Kognitio can run but Impala does not support, or that are long running. Kognitio runs 67 out the remaining 73 queries faster than Impala with 7 of these over 10x times faster.

At the 1TB scale Spark required configuration changes in order for many of the queries in the benchmark to run. Despite these changes there were still 10 queries that Spark could not complete at the 1TB scale.

Kognitio is faster than Spark for 88 of the remaining 89 queries in the benchmark. There are 61 queries where Kognitio is over 10x faster than Spark and five queries where Kognitio is over 100x faster.

From the benchmark it is clear that Spark requires configuration depending on data sizes and workload. Spark is still some way off the ‘deploy and go’ experience you see with Kognitio which needs no configuration other than resource allocation via YARN and a one-off creation of memory images of the data.

In figure 1(c) comparing queries that both Impala and Spark could run; Impala is faster in 67 and Spark is faster in just 3 queries. Impala is over 10x faster than Spark in 15 queries.

Note the grey block at the bottom figure 1(c) represents the seven queries that neither Impala nor Spark could execute at this scale.

How does the platform perform under load?

To support enterprise level SQL requirements it is essential that SQL on Hadoop platforms can perform under mixed concurrent workloads.

The TPC-DS 1TB benchmark was run under increasing workloads up to 10 query streams as defined in the TPC-DS documentation.

An overview of results is given in the following table.

Platform	Impala	Kognitio	Spark
Queries Run in each stream	68	92	79
Long running	7	7	20
No support	24		
Fastest query count	12	80	0

Query overview - 10 streams at 1TB

In order to run this workload effectively seven of the longest running queries had to be removed. These dominated the concurrent performance and as such we consider them to be queries that would not typically be included in a high concurrency workload. They are represented by the grey block at the bottom of each graphic in figure 2, overleaf.

From the results table it is clear that Kognitio is the most performant, running 80 out of the 99 queries the fastest.

Figure 2 shows the speed comparisons between the three platforms for the 10 stream benchmark. Figure 2(a) shows that Kognitio is faster in 56 out of the 68 queries that Impala can run. Impala is faster in just 12 queries.

As well as the slow running queries removed from all benchmarks, Spark also required a further 5 queries to be removed as these queries became long running (over 1 hour) as system resources had to be shared as concurrency increased. This left a total of 79 queries in the Spark benchmark.

In the Spark benchmark the available YARN resources limit the number of concurrent Spark jobs. Clearly this impacts query times as concurrency increases. On the benchmarking system, typical Spark concurrency was between 7 and 8 jobs.

fig 2a - **Kognitio vs Impala**

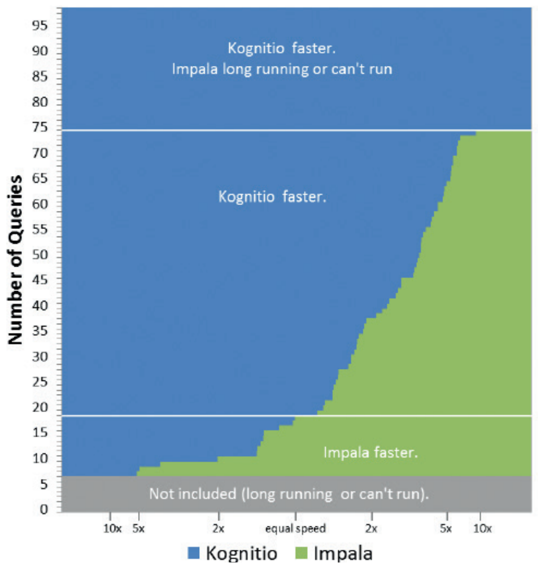


fig 2b - **Kognitio vs Spark**

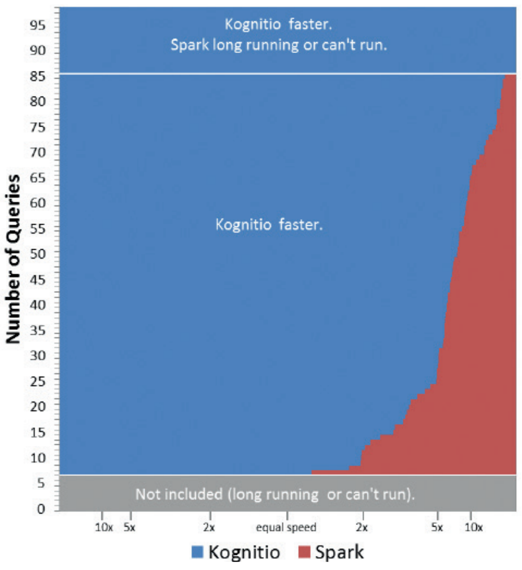
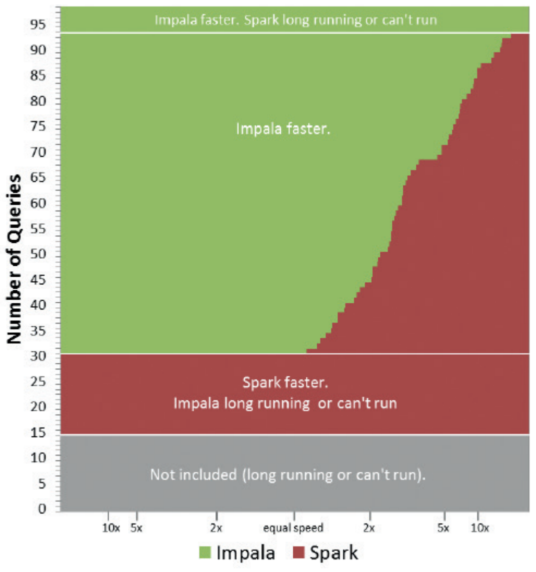


fig 2c - **Impala vs Spark**



Spark was slower than Impala and Kognitio for all comparable queries. Kognitio was over 10x faster in 18 queries and Impala was 10x faster than Spark in 6 queries.

During the Spark concurrency benchmarks it proved extremely difficult to work out which queries were causing issues. Different queries ran slowly or errored in separate benchmark runs. This suggests administration and troubleshooting performance in an enterprise environment would be resource intensive.

Technical Details

Details of the infrastructure utilised for this benchmark along with timings for individual query can be found at www.kognitio.com/downloads/benchmarking/tpcds-techinfo.pdf

Future Testing

Standard Hive was originally investigated as part of this benchmark but lack of SQL support and poor single thread performance meant it was removed from the benchmarks. In future testing, as well as updating the benchmarks to the latest releases for Impala, Kognitio and Spark, **Hive with LLAP will also be included**.

Higher concurrency will be investigated in two ways: a smaller data set (300GB) on the same system and the same 1TB data on a larger system.

The imminent release of **Kognitio version 8.2 will bring new features that will allow Kognitio to handle concurrency even more effectively on highly distributed systems such as Hadoop**. The main feature enhancement, known as asymmetric processing, allows Kognitio to make smarter decisions about where to run the query processing based on the size and location of the data used in each processing step. This will be particularly effective for workloads where many users are submitting lots of queries over different subsets of large data sets.

The configuration of Spark needs more work. A thorough investigation of data distributions is required. The use of a thrift server to access Spark will also allow multiple query streams to access a single context for better memory re-use.

To learn more visit www.kognitio.com.

kognitio