Does Your SQL-on-Hadoop Solution Make The Grade?

Finding the right SQL Engine for your Hadoop initiatives is not an easy task. With a plethora of options and similar capabilities at the outset, you need a better way to look under the hood.

Read this guide to know what to ask your SQL-on-Hadoop vendor.
Does Your SQL-on-Hadoop Solution Make The Grade?

There are many SQL-on-Hadoop solutions in the marketplace. You are bombarded with confusing marketing messages, where everyone is claiming the same thing. You need to look carefully under the hood and understand the engine that powers these solutions. Are you being take for a ride? Will you be driving a lemon or a sports car?

When you grade a SQL-on-Hadoop solution, you need to evaluate:

- Parallelism
- Optimization Capabilities
- Skew Handling
- Extensibility
- Data Ingestion
- Mixed Workload

EsgynDB makes the grade

Esgyn is the leader in Converged Big Data solutions that empower global enterprises to realize the potential of Big Data. With the industry’s most mature, scalable and adaptive SQL Database for Big Data, Esgyn is leading the way enterprises cope with ever increasing data management needs in the cloud or on-premises. Esgyn is a key contributor to Apache Trafodion (incubating) project and has roots in Tandem NonStop SQL and HP Enterprise Data Warehouse products. EsgynDB is the only SQL Database for Big Data that provides a pluggable data management framework for disparate data sources to handle mixed workloads (reads while writing at real-time) without data duplication.

Use these questions to grade your SQL-on-Hadoop choices.

Parallelism

Does your SQL-on-Hadoop solution fully take advantage of parallelism?

Unlike SQL query engines based on single node SMP architectures that have been retrofitted to run on MPP architectures, the EsgynDB SQL engine has been designed from inception to run on scale-out, massively parallel clusters and has been proven on clusters of over 500 nodes. EsgynDB also differs from SQL query engines like Apache Impala and Apache Drill that are trying to build a SQL database engine from the ground up.
EsgynDB builds from a mature SQL engine and simply replaces the storage layer with HDFS. EsgynDB employs parallelism three ways.

- The same query operation is done in parallel on different table partitions across the cluster.
- At different levels of the query tree, results from lower level query operators are streamed in parallel up to higher level query operators. This streaming is done in memory whenever possible, avoiding the costly I/O of writing intermediate results to disk. And before writing intermediate results to disks, EsgynDB checks to see if additional memory is available for the query, and dynamically expands the memory if possible.
- Binary operators, such as two separate and distinct JOIN operators, can independently run in parallel.

In SQL query engines designed to run in scale-out MPP environments, often a query either runs on a single node (no parallelism) or on all the nodes in a cluster. But based on the number of rows to be processed and returned, characteristics of the data such as cardinality and distribution, and the messaging overhead, it can be more efficient to run a query on a subset of the nodes. Using unique technology known as adaptive segmentation to evaluate a query, EsgynDB can run a query on ½, ¼, or some other percentage of cluster nodes. By dynamically choosing the degree of node parallelism, EsgynDB avoids over-utilization of memory, CPU, and networking, and free resources to service other queries. Using this SQL engine, UK retailer Sainsbury’s ran a mix of short-running, small result set operational reports simultaneously with long-running, large result set quarterly reporting. This dynamic degree of node parallelism proved invaluable for running a mixed set of queries efficiently and quickly.

**Optimization Capabilities**

**How smart is your SQL-on-Hadoop solution’s optimizer?**

A key requirement for any SQL engine is to reduce the search space for a query, that is, the amount of data that must be scanned to determine the results for a given query. EsgynDB’s sophisticated cost-based and rules-driven optimizer was designed for extensibility so that the optimizer can be adapted to support varied and changing workloads. Over the years, this extensibility has been used enhance EsgynDB to run operational, analytic, and transactional workloads across industries. For example, EsgynDB can detect common database design patterns, such as star and snowflake schemes, and guide the optimizations appropriately. In addition, the EsgynDB optimizer has been calibrated for specific storage engines and files formats: the HBase storage engine, HDFS files with Hive metadata, ORC files, and EsgynDB tables.

**Does your SQL-on-Hadoop solution support a wide variety of join operators, optimized for analytic to transactional workloads?**
A join combines records from two or more tables by using values common to the tables. Unlike most other SQL-on-Hadoop solutions, which typically support 3-5 join operators, EsgynDB supports a rich set of 13 adaptive and parallel joins, including: nested join, nested cache join, merge join, matching partition join, repartitioned hash join, replication by broadcast hash join, inner / outer join broadcast, dimensional schema star join, inner join, left join, right join, full outer join and self join. EsgynDB leverages this rich set of join strategies, based on query workload, to improve the response times and throughput. While having a rich set of join operators can improve performance, picking the wrong join operator can also significantly hurt performance of a query. To avoid picking the wrong operator, EsgynDB employs risk-based heuristics along with cost-based analysis before selecting the join operator.

Big Data solutions only amplify the need for rich, efficient join capabilities. With Big Data solutions, you are combining real-time and historical data. Your data includes structured transactional data and often external data in semi-structured and unstructured formats. You are performing predictive and prescriptive analytics that require more and more context and more and more joining of varied data sets to identify trends and anomalies and gain insight. The EsgynDB optimizer has supported over 300-way joins in customer proof-of-concept testing and is ready for your complex analytic workloads.

**Does your SQL-on-Hadoop solution cache query plans?**

When a typical SQL query engine receives a new query, it compiles the query into a run-time plan and then optimizes that plan. For operational and transactional workloads, but also BI workloads with short response times, it is common to see essentially identical queries. These queries are run by the same user or a group of users, with minor variants in the query selection based on time and other variables, but the same optimized run-time plan applies. EsgynDB has query plan caching, where frequently run queries plans are stored in memory, eliminating the need to recompile and optimize such queries. Not only does this save processing resources, it speeds the return of query results.

**Does your SQL-on-Hadoop solution enable DBAs to fine tune your queries?**

No DBA wants to tune queries by hand, but the reality is that temporary workarounds for expensive queries, where missing statistics or other factors result in poor performance, are sometimes required.

In many SQL-on-Hadoop solutions, a DBA has some limited capabilities to provide a recommendation or hint to the SQL engine, which may or may not be followed. With the EsgynDB SQL Engine, DBAs have extensive control and any hint to tune and force the shape of the query plan will be followed.
Does your SQL-on-Hadoop solution requires lots of indexes for performance?

Many SQL-on-Hadoop solutions need lots of indexes to access data efficiently, especially for operational queries. This results in a proliferation of indexes to manage and slows data ingestion, update and bulk load performance. EsgynDB implements Multi-Dimensional Access Method (MDAM) technology to use efficiently an index when leading columns of the index do not have predicates on them.

Skew Handling

Does your SQL-on-Hadoop solution handle data skew?

A common problem in many data warehousing / business intelligence applications is data skew, where one column has a disproportionately high number of rows with the same value. For example, HP did not register customer numbers for many sales transactions in some product lines. Instead, the number 2 was used. A simple hash join on two tables, partitioned across the cluster based on customer number, resulted in one node processing all the data where customer number was 2. Not only does this make the query slow by having one node perform most of the processing, this resulted in all queries on the system being slowed if any of them required resources on the overloaded node. In essence, one node is busy and the rest of the nodes are underutilized. EsgynDB is the only SQL-on-Hadoop solution with Skew Buster, a technology that uses the rich statistics that EsgynDB collects on the data to detect data skew in a query and proactive address it. When Skew Buster detects skew in a query, the values from one table are broadcast to all the other nodes in the cluster, distributing the processing workload for this query. In the previously discussed example, HP measured 40x improvement in query performance with Skew Buster, bringing a query down from 80 minutes to 2 minutes. With the era of Big Data and queries mixing structured and semi-structured data, the challenges of data skew only increases.
Extensibility

Does your SQL-on-Hadoop solution enable you to create your own functions? How extensible is your SQL-on-Hadoop solution?

UDFs enable you to create and integrate directly into the database functions that are not supported in SQL. That means all the users can take advantage of that function once it has been built. It also means that if that code needs to change, it only needs to change in one place, and all applications benefit from it. EsgynDB supports three types of User Defined Functions (UDFs): Scalar UDFs, Table-value UDFs, and Table-mapping UDFs. You have the choice of writing UDFs in C++ or Java. Scalar UDFs and Table-value UDFs enable you to perform mathematical calculations or string functions on a set of queried values and use the results later in the same query.

Table-mapping UDFs are similar to Hadoop Map / Reduce operators, providing even more flexibility to embed custom code within a SQL query. And unique amongst UDF implementations, EsgynDB enables you to optimize the performance of your custom code and how it runs with the SQL engine. These optimization capabilities can greatly reduce data movement, reduce resource utilization, and deliver faster running queries.

UDFs enable you to:

- Ingest and enrich data as it is being inserted into the database from streaming frameworks like Apache Kafka
- Integrate with machine learning algorithms such as Anaconda
- Connect to other JDBC data sources and pull data into EsgynDB

Data Ingestion

Does your SQL-on-Hadoop solution support high-speed, bulk-loading of data?

You need efficient, flexible, bulk data loading capabilities as part of your standard operations and maintenance toolkit to:

- Copy test workloads between clusters
- Migrate data from an old to a new production cluster
- Migrate data from another database
- Load transactional data as part of a deployed solution

EsgynDB includes odb, a platform-independent, multi-threaded, ODBC command-line tool for parallel data extract and load. Like EsgynDB, odb was designed from the ground up for parallelism and efficiency. It supports parallel extract and load streams and avoids multiple ODBC buffer packing and unpacking with a single pack on the extract side and a single unpack on the load side.
Does your SQL-on-Hadoop solution support streaming of data efficiently into the database?

Many Big Data applications, such as Internet of Things, rely on message systems to stream data for processing. Apache Kafka is a high-throughput, distributed messaging system that can be used to create a near-real-time stream-processing workflow with EsgynDB. Kafka maintains feeds of messages in categories called topics. Kafka producers are the processes that publish messages to topics. Likewise, Kafka consumers are processes that subscribe to topics and consume messages. EsgynDB can be inserted into a stream-processing workflow by using a Table-mapping UDF. EsgynDB has a library that understands the hash partitioning and salting of the data that can be used by the Kafka producers. That library lets you ensure that messages that will be processed by a consumer go directly to the consumer on the node where the data will be stored, reducing network traffic and speeding performance.

This model of connecting EsgynDB to Apache Kafka can also be applied to other Hadoop ecosystem messaging and streaming frameworks, such as Apache Storm, Apache Apex, and Apache Flink. No other SQL-on-Hadoop solution has such rich, integrated capabilities to support streaming data.

Mixed Workloads

Does your SQL-on-Hadoop solution have the features to handle analytical to operational to transactional workloads?

Many SQL-on-Hadoop solutions have features for basic analytics, but lack the features required for operational and transactional workloads such as referential integrity, stored procedures, and a rich set of OLAP function that SQL developers expect when building applications.

Does your SQL-on-Hadoop solution take advantage of the strengths of the storage engines and file formats?

Most SQL-on-Hadoop solutions do not leverage the strengths of storage engines and file formats to provide better performance and functionality because of the deep integration and customization required. EsgynDB exploits the strengths of the HBase storage engine and ORC file format, by understanding the distribution of data to select the best query plan to access the data and pushing down operations to the storage engine. These efforts come into play when you have queries that federate data from multiple sources, such as EsgynDB tables, HBase tables, and ORC files. In these scenarios, you need a SQL engine that understands and integrates to get optimal performance.
Does your SQL-on-Hadoop provide scalable transactional consistency?

Few SQL-on-Hadoop solutions provide the transactional support you need for a federated Big Data solution. You need more than just simple transaction support for a single row – you need complex multi-row, multi-table and multi-statement transaction support that scales. EsgynDB has a completely distributed architecture, with a transaction manager per node. EsgynDB deeply integrates with the HBase storage engine through HBase coprocessors in each region server. This distributed architecture and deep integration in unique to EsgynDB and enables EsgynDB to deliver highly-scalable, low-latency ACID transactional consistency at very high levels of concurrency. For example, EsgynDB handle update conflicts between two transactions on the same piece of data locally on that node, instead of requiring inefficient messaging between transaction managers and nodes like other SQL-on-Hadoop solutions. Of those solutions that offer transactional support such as Apache Phoenix, the architecture is not truly distributed, resulting in high resource consumption for message traffic, inability to handle high levels of concurrency, inability to handle many updates happening in one transaction, and performance bottlenecks as the cluster scales.

Does your SQL-on-Hadoop provide disaster recovery and scaling across clusters for operational and transactional workloads?

No other SQL-on-Hadoop solution ensures zero lost transactions in spite of a disaster. EsgynDB provides the ability to scale read and write workloads across clusters distributed over multiple data centers. And only EsgynDB support Active-Active configurations, where both clusters can actively be processing read and write transactions.
About Esgyn Corporation

Esgyn is the leader in Converged Big Data solutions that empower global enterprises to realize the potential of Big Data. With the industry’s most mature, scalable and adaptive SQL Database for Big Data, Esgyn is leading the way enterprises cope with ever increasing data management needs in the cloud or on-premises. Esgyn is a key contributor to Apache Trafodion (Incubating) project and has roots in Tandem NonStop SQL and HP Enterprise Datawarehouse Products. Please visit esgyn.com for more information on Esgyn and visit trafodion.apache.org for more information on Apache Trafodion.