EsgynDB – Enterprise-Class Database for Big Data

Enables an enterprise to use the most mature SQL Engine across transactional to analytical workloads, powered by Apache Trafodion

esgyn

Technical Overview White Paper

esgyn

The Genesis of EsgynDB

Database engines take a long time to mature. Maturity is the result of being exposed to a wide variety of customer workloads, each of them challenging the database in some way or other to improve. These improvements are in query *optimization technology*, in the *execution of transactions and queries*, in the *functions and features it supports*, and ultimately in its ability to meet the customers service level objectives of *scale, concurrency, and performance*. Meeting these objectives also entails *data and workload management*, *multi-tenancy*, and enforcement of *security, consistency, and availability*. Oracle, IBM DB2, Microsoft SQL Server, and Teradata have had decades to refine their database engines and even so have limitations in being able to meet all customer workload needs. It is the architecture that you start with, its flexibility and extensibility, which enables a database engine to ultimately meet the large variety of its customer's needs. It also takes patentable innovation and brilliant engineers with lifetime investments to build these powerful database engines.

Apache Trafodion has such a heritage. It started as NonStop SQL/MX at Tandem computers, an ANSI SQL database, built on a scalable Massively Parallel Processing (MPP) *architecture* that was designed for the ultimate in High Availability. MX stood for Massively Extensible because of its extensible architecture. It is still leveraged by HP Enterprise for linearly scalable OLTP workloads on NonStop servers. It was forked to tackle intense Enterprise Data Warehouse workloads on commodity MPP servers, which interestingly it was initially designed for. Thus, Apache Trafodion has seen workloads that span OLTP, operational data store (ODS), and business intelligence (BI) / Analytics applications.

In June of 2014, HP open sourced this 20+ year investment in database technology with hundreds of patents and over $300 million in investment just since 2006, as Trafodion. It became part of the Apache family of projects as Apache Trafodion in May of 2015. Esgyn spun off from HP, with many of the original engineers in July 2015, with a cumulative experience in database technology of 1000+ years, and built EsgynDB on top of Apache Trafodion, as a supported commercial database for enterprise deployments.

The NoSQL revolution and challenges

If the NoSQL revolution taught us anything, it brought to the forefront the following tenets in Big Data computing:

- Low cost of ownership: both in terms of open source software and commodity hardware
- Schema flexibility and schema on read: important for growing companies facing dynamic changes
- Elastic scaling: to meet the high volume and velocity of big data
- Semi-structured and unstructured data support: to handle the variety of data

- Flexibility of data models: to deal with the variety of workloads
- Parallel processing of user code: facilitated by MapReduce
- High Availability model: designed for failures inherent in commodity servers

This resulted in several NoSQL solutions, each with its own API, geared towards a specific subset of Big Data use cases. These use cases span the gamut from real-time streaming to analytical machine learning applications. A detailed discussion of these use cases is in a whitepaper "*The Future of Data Management*".

As the popularity of NoSQL and Hadoop grew, more applications began to move to these environments, with increasingly varied use cases. As web-scale startups matured, their operational workload needs increased, and classic RDBMS capabilities became more relevant. Additionally, large enterprises that had not faced the same challenges as the web-scale startups also saw a need to take advantage of this new technology, but wanted to use SQL. Here are some of their motivations for using SQL:

- It makes development easier because SQL skills are prevalent in enterprises
- There are existing tools and an application ecosystem around SQL
- Transaction support is useful in certain cases despite its overhead
- There is often the need to do joins, and a SQL engine can do them more efficiently
- There is a lot SQL can do that enterprise developers now must code in their application or MapReduce / Spark jobs
- There is merit in the rigor of predefining columns in many cases where that is in fact possible, with data type and check enforcements to maintain data quality, without compromising schema flexibility
- It promotes uniform metadata management and enforcement across applications

The best of both worlds – EsgynDB

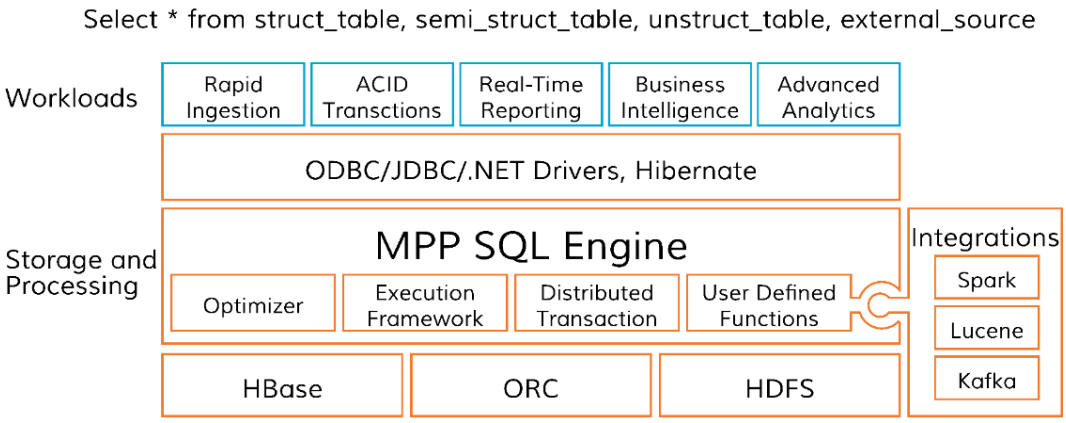
So, there was a resurgence of SQL and RDBMS capabilities, along with NoSQL capabilities, to offer the best of both the worlds. This led to the marriage of a powerful SQL database engine with the extensive NoSQL capabilities inherent in Apache HBase, Apache ORC and other Apache Hive file formats, called EsgynDB.

As is apparent from the NoSQL lessons, the direction for EsgynDB is to integrate with various storage engines to provide customers multiple data structure/model options to support their varied workloads and use cases. EsgynDB supports the structured relational abstraction on top of HBase, the Big Table data model with native HBase table support, key-value stores in Hive tables, and a columnar store with ORC tables.

Towards the end of supporting multiple data models, EsgynDB will continue integrating with storage engines to meet the varied use cases. These multi-model support opportunities will be in the areas of in-memory, text-search, JSON / document support, and graph

database capabilities. Some of this integration effort has already been initiated, such as integration with Ampool, that leverages Apache Geode, and Apache Spark for in-memory, Apache Lucene for text search, management of JSON documents in the database, and JanusGraph (previously Titan) for use cases needing extensive graph traversal.

Another part of the multi-model support story is having an extensible data-pipelining architecture that allows easy integration with other technologies in the eco-system. The *table user defined functions (Table UDFs)* infrastructure, allows such integrations, as with Kafka for streaming, machine learning algorithms such as Anaconda from Continuum, with JDBC data sources to pull in data residing in other databases, with Spark to leverage in-memory analytics, external web searches, integration with Lucene for search, etc.



What is Hybrid Transactional/Analytical Processing (HTAP)?

Along with this need for a single database engine to meet all the data model needs for their various Big Data use cases, customers are also looking for a single database engine to handle all their workloads, from transactional, operational, BI, to analytics. They would rather have all their data on a single platform instead of having separate specialized platforms, each using a different database for each of these workloads, as has been the case for far too long. They want this not only to reduce data movement and duplication, and the latency and operational cost that it introduces, but also since there is huge value in being able to leverage operational data along with semi-structured and unstructured data residing on the same platform. They want to use this data for reporting and real-time analytics as soon as it is captured, without any latency. They would like to seamlessly integrate and access operational, historical, and Big Data, with a single database engine.

Gartner defined the term HTAP in a January 28, 2014 report *Hybrid Transaction/Analytical Processing Will Foster Opportunities for Dramatic Business Innovation*. Wikipedia defines it as "the capability of a single database that can perform both online transaction processing (OLTP) and online analytical processing (OLAP) for real-time operational intelligence processing". EsgynDB not only supports Hybrid Transactional / Analytical Processing

(HTAP) workloads, but extends this processing against multiple data models needed to support the varied Big Data use cases. In essence, this is HTAP on Big Data.

On the HTAP front EsgynDB's march continues to provide full enterprise-class transactional support with multi-cluster / multi-datacenter, multi-master scaling and Disaster Recovery support, along with full point-in-time recovery, cloud elasticity, mixed workload management, multi-tenancy, a secured infrastructure, and *competitive performance on Big Data TPC-C and TPC-DS benchmarks*.

How EsgynDB meets those needs?

Streaming

The vision of a Zero Latency Enterprise is finally taking shape. Data is being captured at breakneck speeds and enterprises need to process this data in real-time. Latency introduced by batch processing of data has simply become unacceptable. Many applications now rely on a streaming paradigm of data ingest instead of a batch one. There are two aspects to this. One is the ingestion of this data to persist it. The second is to act upon it in real-time if the ingest rate is so high that persisting it before processing it is not an option.

HBase is often used to persist data at high rates of ingestion. EsgynDB's *operational performance* for ingestion matches that of HBase. So clearly, EsgynDB can meet the demands of high velocity ingestion of Big Data with full ACID transactional support when needed.

Kafka is often used as a buffer to avoid situations where ingest into the Big Data system may temporarily overwhelm the processing of the data ingested, either for real-time analytics, or to persist into a data store. EsgynDB integrates with Kafka to facilitate ingestion for persistence. Since this integration is facilitated via a *table UDF*, it allows for processing the stream for sophisticated action or analytics, while also leveraging the full SQL capabilities of the engine.

Ingestion rates for EsgynDB are around 74 microseconds (µs) per row of data, since the inserts are into HBase memstore, or memory. After that, any query needing access to that data can immediately access it. So only if the application needs to process data within about less than 100µs or 0.1ms (millisecond) would processing of the data before persisting it be necessary. Just to provide a perspective, Netflix captures 700 billion messages via 24 clusters of 3000+ brokers in a day. Or 370µs per message. LinkedIn does 13 million messages per second using 1100 brokers, or 85 µs per message. We are assuming that a *broker* equates to a *node*. Facebook Messenger and WhatsApp process 60 billion messages a day.

So, unless your requirements are more stringent than they are for Netflix, LinkedIn, or Facebook, EsgynDB can meet your high velocity streaming ingest and persistence needs

even without Kafka. Of course, EsgynDB supports transactional inserts and updates that Kafka does not with the data ready to use with SQL. It also supports *bulk loads* for more batch or mini-batch load scenarios which have much higher ingest rates.

As a general guideline for EsgynDB, you can assume a streaming ingest rate of around 50,000 rows per second per node, and a bulk load ingest rate of around 100,000 rows per second per node.

Operational workloads

Traditionally operational workloads have been relegated to the domain of relational databases. But there is growing interest and pressure to embrace these workloads in Hadoop due to its benefits of significantly reduced costs, reduced vendor lock-in, and its ability to seamlessly scale to larger workloads and data.

Transactional protected operational workloads are typically deemed mission critical in nature because they help companies make money, touch their customers or prospects, or help them run and operate their business. Typically, they have very stringent requirements in terms of response time expectations (sub-second), transactional data integrity, number of users, concurrency, availability, and data volumes. With the advent of the "growing internet of things", the number and types of devices has driven tremendous transaction and data growth, and changes in the type of data that needs to be captured and utilized as part of these transactions. Next generation operational applications often require multi-structured data types, because operational data is evolving rapidly to include a variety of data formats and types of data, for example transactional structured data combined with semi-structured log or Internet of Things (IoT) data from devices, text messages and review comments for sentiment and other analysis, documents or textual information that needs to be searched, visual images associated with the structured data / metadata, etc.

Combined, these requirements can expose Hadoop limitations in terms of transaction support, zero lost transactions even in the face of disasters, bulletproof data integrity, sub-second response times, operational query optimization, and managing workloads comprised of a complex mix of concurrently executing transactions all with varying priorities. EsgynDB addresses each of these limitations. As a result, it provides a differentiated DBMS capable of hosting these applications and their data. It thereby also reduces latency and duplication of data across proprietary operational deployments and workloads on Hadoop that need that data

EsgynDB delivers comprehensive and full-functioned SQL *database features* which allow companies to reuse and leverage existing SQL skills to improve developer productivity for all workloads. Then there are specific capabilities in EsgynDB that are important for these operational workloads, such as:

- *Deep integration with HBase* to leverage it to the fullest
- Optimizations for low-latency read and write transactions in support of the high concurrency and fast response time requirements of transactional SQL workloads
- A Type 2 JDBC driver for direct JNI client access to HBase services to minimize service times
- Caching of compiled SQL plans thereby eliminating unnecessary recompilation overhead. This caching is done at various stages of the compilation process and is not simply a text matching capability. Different plans are cached for similar queries if cardinalities of the values in their predicates differ substantially resulting in different execution plans.
- A patented technology called *Multi-dimensional Access Method* (MDAM) that
  - Efficiently uses index even when leading columns of the index don't have predicates on them
  - Substantially reduces the need for secondary indexes
  - Facilitates *salting* & computed / *divisioning* columns to cluster data
- Extensions to Hadoop® HBase by adding support for ACID (atomic, consistent, isolated and durable) transaction protection that guarantees data consistency across multiple rows, tables, SQL statements using a completely *distributed transaction management* architecture, deeply integrated with HBase, to deliver highly scalable, very efficient, transactional consistency, at very high levels of concurrency
- *High Availability and data integrity features* critical for operational applications
- Full *active-active distributed transactions support* across data centers to scale read/write workloads with zero lost transactions for disaster recovery.
- *Point-in-time recovery* from operational errors, such as a dropped table or erroneous transactional updates
- *Elastic scalability* so that nodes and storage across the cluster can be dynamically increased or decreased with no downtime, while transactions and queries are being processed. The very next transaction or query will leverage the reconfigured compute or storage resources.

Many of these capabilities can also be leveraged across BI / Analytics workloads as well.

Deep integration with HBase

When Trafodion was first open sourced, while there were SQL solutions on the Big Data Hadoop platform that were somewhat addressing the BI and analytical needs, there were none that offered what EsgynDB could for transactional to operational workloads. So the initial focus for EsgynDB was to leverage the incredible database engine developed over all those years, and *deeply integrate that with HBase* and provide a completely scalable *Distributed Transaction Management* infrastructure with minimal overhead, to address those workloads. HBase was chosen due to its capability to be able to handle low latency workloads, high ingestion rates, key structure that enabled operational queries effectively,

schema flexibility, along with all the benefits of NoSQL that it inherits from the Hadoop Distributed File System (HDFS). It also provides capabilities such as compression, encryption, replication for High Availability, elastic scalability, and a whole lot more.

HBase provides EsgynDB more than just a storage engine for a structured relational abstraction. It provides a Big Table data model (key-column-value, aka wide-column store) that customers can leverage for semi-structured data, which does not lend itself as easily to a relational model. It is a perfect union of SQL and NoSQL capabilities for operational workloads. EsgynDB can not only query Trafodion structured tables and native HBase tables in the same query, but can also update them in the same ACID transaction. HBase tables can be defined as EsgynDB external tables. Views can be created using these tables.

The choice of HBase also provides customers a very large eco-system to leverage for their varied needs, with a clear momentum for Hadoop in the Big Data space.

BI / Analytics workloads

Customers have been running Big Data workloads on Hadoop. As they have built these Data Lakes to service these workloads, they have also realized that they need to integrate structured historical data that they host in their Enterprise Data Warehouses (EDW) to run BI queries, along with this Big Data workload. This is to provide more analytics insight by integrating structured historical data with semi-structured and unstructured external Big Data, but also to offload processing that they have been running on expensive proprietary hardware to a substantially lower TCO open source platform. This includes offloading Extract Load Transform (ELT) processing from those platforms, as well as integrating and extending the reporting and analytics to integrate all enterprise data for better insights, eliminating the silos of separate EDW and Data Lake deployments.

EsgynDB is uniquely positioned to service such workloads as well, due to its extensive exposure to EDW ELT and BI reporting and analytics workloads. Its capabilities and maturity in this area far surpass other SQL-on-Hadoop engines servicing these workloads. While the data coming into an EDW may be transformed by ETL tools, HP used the technology behind EsgynDB extensively for massive transformations, i.e. ELT, for both transformation of the data sourced in from many operational systems across the world to be usable by enterprise BI users, and for rollups or aggregations of data into a reporting zone to facilitate servicing the huge volume of concurrent queries against the EDW at very high speeds. This is more pertinent in the world of Data Lakes, which not only source structured enterprise operational data, but also semi-structured and unstructured data that has not been pre-processed by any ETL tool, and could be argued needs more transformation and processing, for the data in these structures to be ready for complex BI and analytical query processing. EsgynDB with its vast experience, can handily deal with such workloads.

Also, EsgynDB is architected to be able to handle HTAP workloads very efficiently. This facilitates even better insight and closed-loop analytics, by reducing latency between operational data, captured on the same platform by EsgynDB via OLTP and streaming applications, and the ability to analyze it as soon as it is captured, and being able to deploy analytical models generated from that analysis, back into real-time operational applications, running on the same platform.

Similar to the specific capabilities to handle operational workload, there are specific capabilities in EsgynDB that are important for BI and Analytics workloads, such as:

- *Deep integration with ORC Files* to leverage it to the fullest
- A *sophisticated optimizer* that leverages statistics on the data to generate the most optimal plan for the query
- Patented technology to deal with skew called *skew buster*
- Patented technology called *adaptive segmentation* to utilize just the resources needed for a query, resulting in high levels of concurrency with minimal resource usage and enhanced resiliency
- A *data flow execution architecture* that leverages the utmost in parallelism and the efficient use of memory to deliver fast response times
- Sophisticated parallel database engine capabilities to support complex reporting queries at high concurrency and throughput.
- Multiple join strategies based on the query pattern. It uses them cautiously to deliver the best performance and concurrency, with efficient use of resources.
- Optimizable *table user defined function* capability to parallelize C++ or Java user functions, similar to MapReduce, to execute in parallel, and integrate with other data sources or storage engines.

Integration with Hive tables

EsgynDB also supports Hive tables in text and sequence file (key-value) formats to enable integration of data residing in HDFS in those formats in the same query, accessing these formats directly and efficiently in HDFS. It also leverages these formats for loading data into EsgynDB at bulk load speeds. Hive tables can also be defined as external tables, making it easier for the user to use them in queries.

Deep integration with ORC Files

Customers demanded more from EsgynDB, with the desire to run BI/Analytics workloads on the same data with minimal data movement and no data duplication. While EsgynDB supported such capabilities, it was clear that HBase had its limits when it came to such workloads. For scan-heavy reporting and analytics workloads columnar stores provide performance that HBase cannot. EsgynDB provides *deep integration with Apache ORC files, a columnar data store integrated with HDFS.* to address these workloads.

Meeting the demands of HTAP and beyond

We have discussed how EsgynDB addresses the varied HTAP workload needs, the increasing demand for streaming, while accommodating multiple Big Data models. Let us now look at other aspects of what is important for Enterprises deploying Big Data technologies.

- You are familiar with the four Vs of Big Data – Volume, Velocity, Variety, and Veracity:
  - **Volume:** EsgynDB runs on Hadoop and has been a massively parallel, substantially scalable database since its inception. Therefore, it is clearly geared towards handling large volumes of data. The *architecture* and *performance* discussions in the *Appendix* discuss these in more detail.
  - **Variety:** EsgynDB supports multiple data models and has initiatives in place to further diversify this support. It not only supports a variety of data structures and models, but it also supports a variety of workloads from OLTP, operational, to BI and analytics – that is, HTAP workloads.
  - **Veracity:** Veracity has a lot to do with data governance, transparency, lineage, also called data provenance. These are procedures and processes put in place that can be facilitated by EsgynDB. Also, the analysis of the data to ensure veracity of non-enterprise semi-structured and unstructured data can also be facilitated by EsgynDB. Finally, with its distributed ACID transactional support, solid Disaster Recovery and recovery capabilities, strong and enforced data types and check constraints, EsgynDB ensures data integrity and quality, playing an integral role in ensuring veracity.
  - The **Velocity** of data is the speed with which data is being consumed by the Big Data platform, for IoT related applications. One aspect of data velocity is the speed of ingestion, discussed earlier. The second is being able to handle HTAP workloads in real-time. Esgyn delivers on performance SLAs at very high concurrencies to meet this need.
- Drastic reduction in overall Total Cost of Ownership with:
  - Open source software and pricing model
  - Running on commodity hardware instead of specialized hardware
  - Leveraging existing SQL skills and tools in the organization
- Being able to run on the Cloud, such as AWS, with *Elastic scalability* and virtual provisioning to:
  - Reduce cost in the operational complexity of managing hardware and software resources
  - Provision resources on a need based – increasing resources during seasonal sales increases and reducing it thereafter, for example
  - Leverage software and hardware resources much more efficiently across multiple tenants may they be multiple hosted customers, multiple applications of an enterprise, multiple divisions or departments, each with varying needs for compute and data resources
  - Move towards a self-service paradigm

- Enterprise-class *manageability* and *operational infrastructure*: Enterprises have reached a maturity in how they manage and operate their application and data deployments. They expect the same caliber of support for their Big Data deployments. EsgynDB offers enterprise level *Manageability* and *Availability* capabilities to address these needs.

Conclusion

So EsgynDB is the only database engine that can run all your workloads from transactional, operational, BI, to analytics on a single platform, while facilitating the integration of structured, semi-structured, and unstructured data, blending NoSQL and SQL technologies, to provide you a platform for your next generation applications. And it can do this with *unparalleled performance at both ends of the spectrum*.

For detailed discussions on the capabilities mentioned in the sections above please follow the hyperlinks to the appropriate sections in the Appendix.

About Esgyn and EsgynDB

Esgyn Corporation's mission is to disrupt the Big Data database market by establishing EsgynDB as the most prevalent database engine, used for all workloads from OLTP to ODS to BI to Analytics (or HTAP), against streaming, in-memory, and persistent data, deeply integrated with storage engines that support multiple data models on a single system with open source economics.

Esgyn's premier offering is EsgynDB Enterprise, a hardened, secure, enterprise-class HTAP SQL database engine for Big Data built on Apache Trafodion. With offices in Silicon Valley and Shanghai, Esgyn offers support, services and training for EsgynDB that enterprises expect for their production environments.

For more information visit www.esgyn.com or email info@esgyn.com.