Exponam & Apache Spark

Exponam's direct integration with Apache Spark, including Databrick's commercial offering, improves the time-to-value of quantitative, analytic, and machine learning results available with Spark. Exponam's integration achieves results with these advantages:

- **A native data source for loading and saving Exponam .BIG files**

  The native data source is built with Exponam's powerful core technology, which dynamically tunes itself to your enterprise Spark clusters' runtime capabilities. This ensures lean execution, high performance, and brisk throughput to and from Spark's internal RDD (resilient distributed data) structures. With Exponam, you can ingest large datasets into Spark out of highly compressed import files without wasting space and time. And you are able to egress Spark data into a format that is orders of magnitude more compressed than standard delimited formats, allowing much larger datasets to be faithfully preserved for audit and archival needs.

- **Frictionless access with Spark DataFrames**

  Exponam data load and save operations are available using standard DataFrame syntax that data scientists use every day, whether with Scala, Python, or Spark SQL. Exponam's default options can be trivially overridden using standard DataFrame options, unleashing the full power of Exponam's underlying technology: security, file optimization levels, story files, and application-defined supplemental metadata.

  An Exponam file can contain any number of tables, each with its own schema and row-level data. Each table can be loaded individually, allowing a single Exponam file to transport entire rich repositories of data into Spark. Exponam's schemas eliminate the potential ambiguity of inferred schemas, and mean that the native representation of objects in RDDs is always optimal and correct.

  Further, Exponam's save operation with Spark DataFrames allows the flexibility that DataFrame users demand. Save can be invoked in a cluster-aware fashion, with each node in the cluster generating an output file for its local data only, which can be advantageous for extremely large RDDs. Alternately, DataFrame results can be coalesced (or glom'ed) through the master node, and result in a single output file. The point is that Exponam allows you to use the pattern that best fits your cluster profile and data egress requirements.

- **Data lineage**

  Modern data architectures seek to preserve data lineage across disparate products and solutions, an almost insurmountable task when data is moved between traditional silos, compute grids, and data grids. With Exponam, the provenance of data is integral to the file itself. This allows solution architectures using Apache Spark to maintain data lineage from ingest through egress, so that the linkage to upstream systems is faithfully preserved.

- **Security**

  Standard data exchange formats for Spark require data that is unencrypted when at rest. Exponam, in contrast, is always encrypted at rest, even as it is being loaded into the cluster. The attack surface for potential data breach is demonstrably smaller with Exponam.

  Further, Exponam's default behavior on load operations is to first establish the integrity of the file. If the file has been tampered with, it will fail with a standard Spark exception, and absolutely no row-level data will be generated in Spark.