5 data integration trends that will define the future of ETL in 2018

Abhishek Tiwari

Citation: *A. Tiwari*, ”5 data integration trends that will define the future of ETL in 2018”, Abhishek Tiwari, 2017. doi:10.59350/7e1xt-b6v23

Published on: December 28, 2017

ETL refers to extract, transform, load and it is generally used for data warehousing and data integration. ETL is a product of the relational database era and it has not evolved much in last decade. With the arrival of new cloud-native tools and platform, ETL is becoming obsolete. There are several emerging data trends that will define the future of ETL in 2018. A common theme across all these trends is to remove the complexity by simplifying data management as a whole. In 2018, we anticipate that ETL will either lose relevance or the ETL process will disintegrate and be consumed by new data architectures.

Unified data management architecture

A unified data management (UDM) system combines the best of data warehouses, data lakes, and streaming without expensive and error-prone ETL. It offers reliability and performance of a data warehouse, real-time and low-latency characteristics of a streaming system, and scale and cost-efficiency of a data lake. More importantly, UDM utilizes a single storage backend with benefits of multiple storage systems which avoids moving data across systems hence data duplication, and data consistency issues. Overall, less complexity to deal with.

Databricks Delta is a perfect example of this class. Delta implements the unified data management layer by extending the Amazon S3 object storage for ACID transactions and automatic data indexing. A solution like Delta makes ETL unnecessary for the data warehousing.

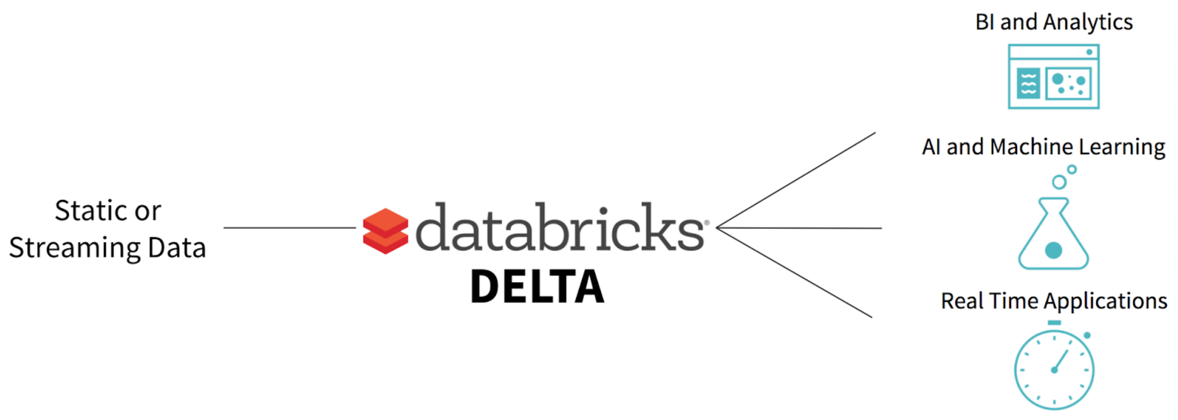

Figure 1: A Unified Data Management System for Real-time Big Data. Image credits Databricks

Common in-memory data interfaces

In 2018, we will see new data integration patterns those rely either on a shared high-performance distributed storage interface (Alluxio) or a common data format (Apache Arrow) sitting between compute and storage. Both Alluxio and Apache Arrow are designed to enable the data interoperability between existing big data frameworks by being a common interface for high-performance in-memory access however the approach is different. Nonetheless, they can complement each other. For instance, Alluxio, originally known as Tachyon, can potentially use Arrow as its in-memory data structure.

Apache Arrow is a standard columnar in-memory data interchange format which supports zero-copy reads for lightning-fast data access without any CPU overhead. At this stage, Apache Arrow has support for 13 major big data frameworks including Calcite, Cassandra, Drill, Hadoop, HBase, Ibis, Impala, Kudu, Pandas, Parquet, Phoenix, Spark, and Storm. Because supported big data frameworks and applications can utilize the same internal memory format, they can avoid data serialization and deserialization to convert data between various formats. Apache Arrow's in-memory columnar layout is specifically optimized for data locality for better performance on modern hardware like CPUs and GPUs.

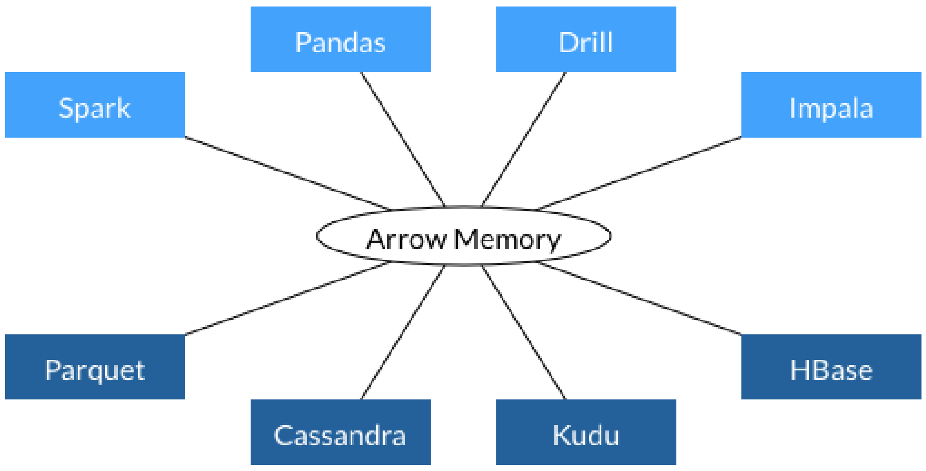

Figure 2: A common data layer sitting between compute and storage. Image credits Apache Arrow

In contrast, Alluxio a middleware for data access - think Alluxio storage layer as fast cache. It generally improves performance by placing frequently accessed data in memory. Internally Alluxio implements a tiered storage, which enables Alluxio to use other storage types such as SSD and HDD for less frequently accessed data in addition to memory. Based on data access pattern i.e. hot, warm and cold

Alluxio makes. Alluxio facilitates data sharing across big data framework and locality between jobs irrespective frameworks.

Machine learning meets data integration

Data solution vendors like SnapLogic and Informatica are already developing machine learning and artificial intelligence (AI) based smart data integration assistants. These assistants can recommend next-best-action or suggest datasets, transforms, and rules to a data engineer working on a data integration project. Similarly, they can help data scientists by recommending which data sets to use for their next project or suggestions on additional data sets that may complement their existing data sets.

In addition, Machine learning techniques have been also applied to automate the entity relationship modeling. There are a few vendors like Panoply, Informatica, and Tamr offering automated data modeling capability as part of their data integration kit. Vendors report that they have achieved up to 70-90% accuracy with rest of manual adjustments and customizations were performed by a human expert in the loop.

Obviously, this has a clear impact on traditional ETL process which provides a fixed set of views. With smart data assistant and automated data modeling going mainstream in 2018, there will be sufficient arguments for embracing a No-ETL approach which supports structure as well as unstructured data. More details on this approach.

Event-driven data flow architecture

More and more organization are jumping on the bandwagon of event-driven architecture with the view that it can provide actionable insights in real-time. To implement event-driven architecture, organizations are now thinking events as first-class citizens and data integration in terms of event data flows rather than data that is just processed and landed in a database. To achieve this, organizations are utilizing distributed messaging system such as Apache Kafka. On-top, they are implementing concepts such events, topics, event producers, and event consumers. Producer applications publish events to one or more topics which are then processed by consumer applications. This leads us to a data flow pattern where data is read from a given topic stream, transformed and written back to another topic stream.

A key aspect of event-driven data flow architecture is support for microservices architecture and more specifically database per service pattern. Most of the organizations struggle with monolithic application architecture and microservice are enabling organizations to break these monoliths into manageable chunks or components.

Leveraging the recent hardware advances

Vendors are already working to apply recent hardware advances such as Graphics processing unit (GPU), Tensor processing unit (TPU), and Single instruction multiple data (SIMD) to create data warehouse solutions which will be up to 100X faster than a traditional data warehouse.

GPUs and TPUs are particularly interesting one due to current machine learning trends. A recent research from Google demonstrated that neural network based learned index can outperform the cache-optimized B-Tree index by up to 70% in speed while saving an order-of-magnitude in memory. These learned indexes can immediately replace index structures currently used by data warehouses. At least 10X performance boost is anticipated if learned indexes are used with GPUs/TPUs.

On top of that, SIMD instructions such as FAST can exploit modern processor architectures achieve 6X performance improvement over traditional data warehouse indexes. SIMD instructions are already utilized by Apache Arrow - mentioned in the previous section - for native vectorized optimization of analytical data processing.