MODERN BATCH. TO BATCH OR NOT TO BATCH? THERE'S NO QUESTION!

Batch is Dead, Long Live Batch

To borrow a line from the salute to new kings, as "old" batch applications may die, "modern" application workflow orchestration applications spring up to replace them. That is because a batch approach makes a whole lot of sense for many scenarios, no matter what you call it. Recall Shakespeare's line "A rose by any other name would smell as sweet". Below are a few examples.

Predictitve Maintenance

An energy company collects data from the hundreds of sensors that each oil well is instrumented with. This information provides indicators of oil well performance and operation. If any sensor is trending towards a failure indication, an engineer is dispatched to correct the problem before the well has to be shut down and stops producing oil.

But what if there are two wells to fix and only one engineer. Which one does he fix first? The simple answer is the one that produces more oil. The more complicated answer is the one that produces more oil but also can be fixed before the well fails and the engineer can reach and that the parts are available for.

Both the simple and complex answers require additional processing that has to examine production

trends over time, location relative to the availability of an engineer and the availability of parts. Those answers may require interaction with all sorts of operational back-end systems as well as sequencing and processing that most likely requires sophisticated orchestration.

Transactions

Let's consider an interactive application, like online purchasing of just about anything. The consumer expects to see a catalog of available goods, fill up a cart and check out. Below is a step-by-step review of the process:

- Purchase: Let's say one of the items purchased reduces inventory below a certain threshold. Let's also assume the goods have to be packaged and then shipped by a pre-determined courier or delivery company. Or perhaps the inventory is held by another supplier and that other company will package and ship the goods.
- Inventory Replenishment: Once the order has been confirmed, do you think that inventory is replenished immediately? Perhaps this purchase was made at 9:00 AM and hundreds or thousands of additional items will be purchased throughout the day. It seems reasonable to wait until some point, either a specific time or when some number of items have been purchased before placing an order to replenish the inventory. And once the replenishment is initiated, it's very likely that process consists of several steps occurring over days (or longer) which require coordination, monitoring and visibility into the progress of the entire process.
- Packaging for Shipment: What about the packaging? Does it make sense that the ordered items are picked immediately and boxed for shipment as soon as the transaction is completed or is it reasonable to wait again until a specific time or a specific number of certain goods can be picked up?
- Shipping: Finally, how about the shipping? Should we schedule a pickup for each order or only at the end of the day? And if we do request a shipment as soon as an order is placed, is it reasonable for a truck to come and pick up the single order and immediately begin the delivery process?

Streaming

Here's another example. Data is streaming from IoT devices into an application. Maybe we are collecting data from smart meters and watching for anomalies like device failures or from sensors watching for hazardous fume leakage. We definitely want to recognize problems immediately and take action. But what about the 99.999% of records that are recording just normal usage? We probably want to capture them and add the data to our billing and trend indicators showing customers how they use gas or electricity or whatever over a day, week, or month. Will we aggregate that data as it is streaming in and constantly update our information or will we defer that for a daily, weekly or monthly process?

Data Movement, Formatting and Storage

Data manipulation is another classic use for batch. Collections of records, what may be known as a file or a "data set," fit extremely well into the definition of a batch. In fact, almost any application that processes bulk data is really a batch application.

As traditional data management is being overhauled and disrupted with technologies like Hadoop

and Big Data, batch continues to be among the most common use cases in this sector for both traditional and modern applications.

Moving data is inherently a batch process because typically files, a synonym for a batch of data, are involved. Previously, I mentioned streaming. This is a relatively new term that is applied very broadly. It is assumed that because data is flowing in real time, each individual record correspondingly must be processed in real time. This is frequently not the case. In fact, one of the most popular new technologies in the Big Data world is Spark Streaming. It is a "micro-batch" implementation. Even when "data in motion" is being processed, it is also almost always stored for subsequent processing, usually in some batch mode.

Extract, Transfer, Load (ETL)

Data occurs in so many formats that it's almost always mandatory to reformat, edit, normalize or apply some other processing to make it useful. This collection of actions is described as ETL. An amusing play on words is a new variant called ELT used by Big dData aficionados to highlight the non-deterministic approach of Big Data that defers the transform phase to the last step so that potential insights are not biased by format. It is still largely a batch process.

Storage

All data that has any potential value beyond its immediate, real-time usage, must be stored somewhere and managed somehow. This turns out to be almost all data and one may be hard pressed to find any exceptions. Even intentionally temporal social media conversations that are designed to "self-destruct" within some short amount of time still must be managed for that duration. Even that destruction after the desired time has elapsed is a data management function and is simply a specific use-case of traditional records and retention management that demands destruction upon expiration.

This is the second of a 4-part blog series on "modern batch." You can read the other blogs here:

- Part 1: Modern Batch Processing: A Thing of the Past or Essential Discipline?
- Part 3: Modern Batch: Managing the Madness?
- Part 4: Modern Batch: The Ops in DevOps fully evolved – "Jobs-as-Code!"