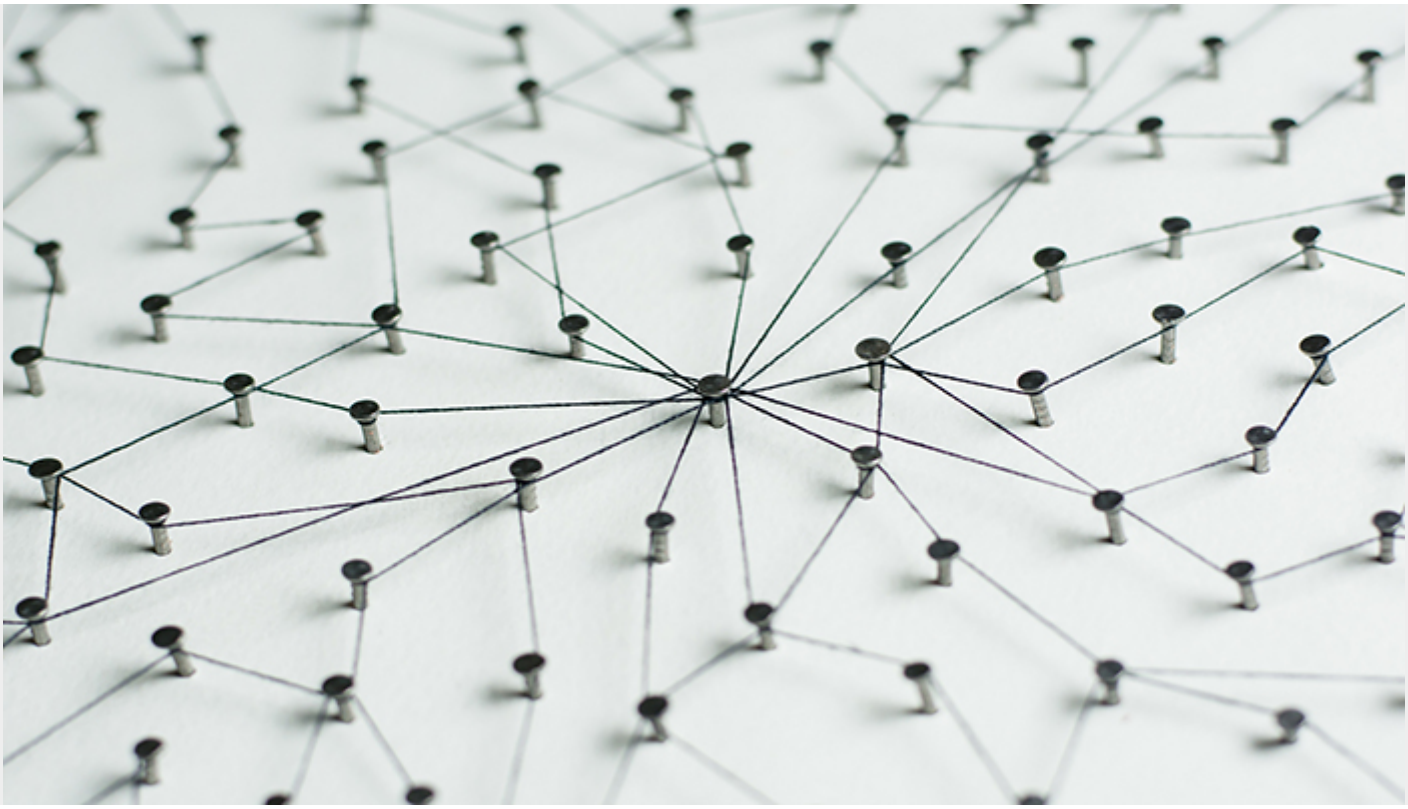
MODERN BATCH PROCESSING: A THING OF THE PAST OR ESSENTIAL DISCIPLINE?



What is Batch?

Applications operate in only a few modes.

One is interactive or real-time. In this mode, the application is communicating with another application or a human. Real-time means the response or process must produce a result immediately (or quickly enough to seem immediate) to satisfy the waiting second party. This is also the meaning of interactive, where the application is interacting (or conversing) with that same second party.

Another, relatively new mode is streaming. Data is arriving or flowing constantly and the expectation is that something is receiving the data and taking some action.

Everything else is unattended, asynchronous, background or “batch.” This is arguably the mode where computers are most effective because they can operate at “machine speed” without having to wait for slow humans, slow devices delivering data or slow networks to enable communication with another application or computer.

A common misconception is that batch equals long-running, overnight, low priority or deferred (as in I don’t need to do that now so I’ll run it in batch later) processes.

Although some batch processing may have those characteristics, there are also micro batch and "batch transactions" that are high-speed, iterative and efficient.

And let's not get too enamored with the term "instant," even though we live in an instant gratification society, because what is instant to humans is an eon to a computer. I've heard statements like, "Batch is no longer acceptable because we cannot wait overnight." That's a statement displaying a shocking misperception not only of what batch is, but also of how computers operate. But let's look at what batch is good for.

Typically, batch, as the word implies, relates to processing a collection and it has evolved to also mean performing multiple steps in some sequence. Adding scheduling to the mix adds the requirement for some event to occur before processing is initiated where the event may be the occurrence of a specific time, completion of another batch, creation of a file, the execution of a transaction, publishing a message on a queue and other similar events.

Traditional batch, like inventory processing, warehouse management, payroll and customer billing is still very much a major activity in almost every business computing environment. The big question for batch practitioners is whether new business services can or should use a batch approach in their implementations.

What is "Modern Batch Processing"?

Since the inception of commercial data processing, batch has accounted for a significant share of all useful work done by computers.

As noted above, a lot of people still think batch is that "old" overnight processing. I believe this thinking can be traced to the modern application development lexicon that favors the notions and terminologies of 'real-timeness'. This is because of the overwhelming attention paid to the 'real-time' nature of everything we do, especially in the responsiveness required from modern digital transformation applications. I believe it is time to consider defining a new "Modern Batch" and give the term meaning in the DevOps lexicon. If you are familiar with the Wizard of Oz, the Scarecrow, Lion, and Tin Man just needed some external validation to confirm to the world what was already true but misunderstood. So batch "just" needs the right term to reveal its value to the world and help it claim its rightful place as not only relevant, but an essential discipline in a modern architecture.

"Application Workflow Orchestration" may be that term. To be sure, it's a mouthful so let's unpack and examine this closely. Orchestration means automated coordination and management. Application is added because many areas in technology, such as configuration and infrastructure, release, change, etc. utilize orchestration. Workflow connotes a series of steps comprising a business service or function delivered by applications rather than managing the creation or deployment of the application components.

Now that we have a modern name, let's delve deeper into our topic.

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

- Part 2: Modern Batch: To Batch or Not to Batch? There's no Question!
- Part 3: Modern Batch: Managing the Madness?
- Part 4: Modern Batch: The Ops in DevOps fully evolved – "Jobs-as-Code!"