5 REASONS ETL IS THE WRONG APPROACH FOR MAINFRAME DATA MIGRATION

Change is good – a familiar mantra, but one not always easy to practice. When it comes to moving toward a new way of handling data, mainframe organizations, which have earned their keep by delivering the IT equivalent of corporate-wide insurance policies (rugged, reliable, and risk-averse), naturally look with caution on new concepts like extract, load, and transform (ELT).

Positioned as a lighter and faster alternative to more traditional data handling procedures such as extract, transform, and load (ETL), ELT definitely invites scrutiny. And that scrutiny can be worthwhile.

Definitions provided by SearchDataManagement.com say that ELT is "a data integration process for transferring raw data from a source server to a data system (such as a data warehouse or data lake) on a target server and then preparing the information for downstream uses." In contrast, another source defines ETL as "three database functions that are combined into one tool to pull data out of one database and place it into another database."

The crucial functional difference in these definitions is the exclusive focus on database-to-database transfer with ETL, while ELT is open-ended and flexible. To be sure, there are variations in ETL and ELT that might not fit those definitions, but the point is that in the mainframe world ETL is a tool with a more limited focus, while ELT is focused on jump-starting the future.

While each approach has its advantages and disadvantages, let's take a look as to why we think ETL is all-wrong for mainframe data migration.

ETL is too complex

ETL was not originally designed to handle all the tasks it is now being asked to do. In the early days, it was often applied to pull data from one relational structure and get it to fit into a different relational structure. This often included cleansing the data, too.

For example, a traditional relational database management system (RDBMS) can get befuddled by numeric data where it is expecting alpha data or by the presence of obsolete address abbreviations. So, ETL is optimized for that kind of painstaking, field-by-field data checking, "cleaning," and data movement, but not so much for feeding a hungry Hadoop database or modern data lake. In short, ETL wasn't invented to take advantage of all the ways data originates and all the ways it can be used in the 21st century.

ETL is labor-intensive

All that RDBMS-to-RDBMS movement takes supervision and even scripting. Skilled database administrators (DBAs) are in demand and may not last at your organization. So, keeping the human part of the equation going can be tricky. In many cases, someone will have to come along and recreate their hand-coding or replace it whenever something new is needed.

ETL is a bottleneck

Because the ETL process is built around transformation, everything is dependent on the timely completion of that transformation. However, with larger amounts of data in play (think, Big Data), this can make the needed transformation times inconvenient or impractical, turning ETL into a potential functional and computational bottleneck.

ETL demands structure

ETL is not really designed for unstructured data and can add complexity rather than value when asked to deal with such data. It is best for traditional databases but does not help much with the huge waves of unstructured data that companies need to process today.

ETL has high processing costs

ETL can be especially challenging for mainframes because they generally incur MSU processing charges and can burden systems when they need to be handling real-time challenges. This stands in contrast to ELT which can be accomplished using mostly the capabilities of built-in zIIP engines, which cuts MSU costs, with additional processing conducted in a chosen cloud destination. In response to those high costs, some customers have taken the transformation stage into the cloud to handle all kinds of data transformations, integrations, and preparations to support analytics and the creation of data lakes.

Moving forward

It obviously would be wrong to oversimplify a decision regarding the implementation of ETL or ELT—there are too many moving parts and too many decision points to weigh. However, what is crucial is understanding that rather than being focused on legacy practices and limitations, ELT speaks to most of the evolving IT paradigms.

ELT is ideal for moving massive amounts of data. Typically, the desired destination is the cloud and often a data lake, built to ingest just about any and all available data so that modern analytics can get to work. That is why ELT today is growing and why it is making inroads specifically in the mainframe environment. In particular, it represents perhaps the best way to accelerate the movement of data to the cloud and to do so at scale. That's why ELT is emerging as a key tool for IT organizations aiming at modernization and at maximizing the value of their existing investments.