Real-Time Data Analytics

Data Analytics is becoming a key discipline for competitive advantage in many enterprises, and use cases for real-time analytics with Big Data are continuously surfacing with compelling value propositions. The legacy approaches to data preparation and delivery clearly are not appropriate for real-time data analytics.

Typically, for Business Analytics, data is fed periodically into a data warehouse using an ETL (Extract/Transform and Load) process, which brings dissimilar data to a meaningful and useful aggregation that can be queried. Unfortunately, the nature of data warehouses almost ensures that the data is not fresh, and certainly not real-time. With the explosion of data, and moving toward real-time data analytics, the data warehouse is simply not equipped to deliver live data. If we look at the pathway from sources to analytics, it becomes clear that most data preparation liberally dishes up latency at every step.

Real-time analytics means data is delivered to the analytics tool or dashboard in a timeframe from milliseconds to a few seconds, so it is clear that another paradigm is in order. This paradigm is a combination of live Data Federation and Data Virtualization, which are the underpinnings of Enterprise Enabler® ("EE"), which dramatically reduces custom programming, designing and populating databases, and the unnecessary reduction of latency on those data sets.

Real-time data feeds can be accessed and connected for display and analysis, but many times that information is useful only in the context of data that is more static. Suppose you have a real-time feed that shows the current location of an airplane, another that shows altitude, and perhaps other live data feeds. As the analytics tool performs algorithms against this data live, there may be additional information that is included in the analysis, but is either not available real time, or does not change often. Perhaps that data is accessed every minute or hour, as appropriate, or on and event trigger or a web service call. The end user will want to see other "contextual" information about the plane itself, perhaps the pilot, his history with that plane, number of gates available at the destination, number of passengers and other related information.

Along with real-time data, Enterprise Enabler offers this contextual data, pulling it directly from multiple dissimilar sources, and aligning it as a virtual data model that can be queried for display by the analytics tool as if it were a database.

Data Preparation

While data preparation is generally considered a one-time configuration to feed data to the tool, the definition often needs to be modified over time in order to reflect the changing needs and new understanding of the target objectives. Most integration platforms are heavy and unable to prepare and configure data quickly, nor to apply changes easily. Enterprise Enabler is an agile tool that eliminates nearly all of the classic data preparation time. This lack of heavy overhead is reflected not only in the ease of the data preparation step, but also in the speed of run-time delivery.

Consider the steps for just adding a new set of data to the data warehouse.

1. Determine what data is needed
2. Get in the queue with IT to get it in the data warehouse
3. Perform considerable custom programming to get the data aligned and meaningful
4. Schedule periodic ETL updates
5. Test, and debug.

That's essentially the same process that is used any time a staging database is involved. The bottom line is that it is awkward, if not impossible, to provide context data in the classic ETL manner. Instead, use of the data virtualization paradigm, which delivers the latest data with minimal latency to the analytics tool, offers speed and agility and eliminates data staging.

Real-time Data

With the understanding that real-time data feeds nearly always require timely related contextual information in order to make it meaningful, let's move the focus to "just-in-time" data delivery. Some data is clearly useful only if it is available "real time," however there seems to be a propensity to use real-time data whenever it is available, regardless of whether it improves the outcome compared to data with latency. The correct identification and use of data should always be determined before initiating a project.

With the definition of real-time data being from milliseconds to a few seconds, much data can be presented in that time frame without special technologies. For example, resolving a complex query across six different sources can be easily accomplished by Enterprise Enabler in what we are defining as "real time."

Latency

Real-time analytics demands near-zero latency. This means that data access, manipulation, and delivery must leverage clean, efficient architectures and eliminate custom coding wherever possible. Enterprise Enabler is the most effective platform on which to implement real time as well as "just-in time" data flows. When applying the "Just-in-time" concept to data latency, and dashboard design, it becomes obvious that not all the data needs to be, or should be, real-time.

What do you need, and how do you need to get it?

When considering "real-time" data feeds for you dashboard, there are many different kinds of data feeds available. Some are "streaming," some are "on demand," which are initiated upon refresh, an event, or a call from a logical workflow.

Do you need a "snapshot" of current data? Do you need it to be streaming? What about historical data for specific time frames? Do you need raw data from a feed?

Are you doing RTSA (Real Time Streaming Analytics?) How do you need/want the data to be delivered… RSS, HTTP, XML?

Enterprise Enabler and Actionable Dashboards

Most dashboards and analytics displays are just that … they display information in tables, graphs, maps, and other graphics. Suppose you perform some analytics and have an "aha!" moment? Enterprise Enabler, as the underlying integration platform can respond to your decisions and immediately write back to applications and databases or trigger complex processes based on your conclusions.