Self-Service Analytics

Why Self-Service Analytics Has Gone Backwards and What To Do About It

March 2018
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transitioning from the data warehouse</td>
<td>3</td>
</tr>
<tr>
<td>Six reasons why self-service analytics progress has stalled</td>
<td>3</td>
</tr>
<tr>
<td>The way ahead</td>
<td>5</td>
</tr>
</tbody>
</table>
We’re in an era of do-it-yourself — both in our lives generally and more specifically in business. Whether it’s YouTube videos that teach us how to change a tire or the way Amazon Web Services has transformed how developers and IT can launch applications, we are looking for ways to be more independent and self-sufficient.

The motivations for this trend in business are clear: less intermediation from IT means higher productivity and more innovation. Quality increases because those with the most knowledge about the specific business problem can tackle the issue themselves. In an era of do-it-yourself, less intermediation from IT means higher productivity and more innovation.

This move toward self-service is readily apparent in the realm of analytics. Decades ago, the data warehouse was designed to bring data together so that analysts could make sense of what was happening in the business. This was a tremendous achievement. But nothing in technology ever stays stagnant.

Since the inception of the data warehouse, the data challenges facing businesses have vastly increased. The data landscape has grown rapidly, both in the sheer amount of data now available, as well as in terms of the complexity and the speed at which it is arriving, beyond the capacity of the data warehouse to handle. Now companies have data everywhere: in SaaS applications, enterprise applications, cloud data stores, data warehouses, NoSQL databases, Hadoop, and more. The data you need is often scattered across sources in the cloud or on premises, in data lakes and in special repositories.

With tools like Tableau and Qlik, analysts could access and play with data in the data warehouse in new ways. Yet many users still preferred to use spreadsheets as the simplest way to do their work because they allowed people to extract and use data in a familiar way. With these tools, progress had clearly been made toward the ultimate goal of self-service analytics as well as toward freedom for analysts to work with data however and whenever they want, in a way that can be shared and viewed. But that progression was predicated on having the data warehouse at the center of the enterprise universe, and thus all the solutions for self-service analytics were dependent on the data warehouse architecture.

**Transitioning from the data warehouse**

Over the past decade, the assumption that the data warehouse must be the center of enterprise data started to break down in a variety of ways. As a result, instead of analytics becoming increasingly self-service oriented, for the first time, the world of analytics was actually going backward, away from the self-service ideal. The progress of previous efforts hasn’t been completely undone, but compared to where technology was a few years ago, self-service has become much more difficult to achieve. As a result, analysts are much more dependent on IT than ever before. It is worth considering why this backward trend has occurred so that enterprises can understand the best way to achieve self-service analytics going forward.
Six reasons why progress toward self-service analytics has stalled or regressed

The complexity of building, maintaining, and distributing extracts and data marts
Most of the self-service technology currently on the market creates datasets that are designed to be distributed to desktop tools. Those tools then use those datasets in a local computing environment such as a laptop or desktop or on a remote server. But this leads to data being divorced from the original source. Users are working with disconnected extracts of data, which creates a plethora of copies. To address this issue, companies had to find a distribution mechanism so that data in all these extracts could be refreshed at various intervals.

Many data prep products end up causing the same problem. Just like standard extract, transform, load (ETL) processes, data prep products create another copy of the data. It’s not feasible for IT to have every user create a copy of the data every time they need to analyze it. The result is that data prep tools end up far removed from being self-service. Data catalogs, BI acceleration tools, BI extracts, and ad hoc acceleration tools (such as MPP databases) all pose the same problem — they create more copies than it is feasible to deal with over the long term.

The need for management of the complexity and redundancy becomes overwhelming.

The performance of desktop tools vs. server-size datasets
Another reason for the backward trend in self-service analytics is that many self-service tools currently in use were built to run on desktops or laptops. The tools were not designed to connect to larger databases and then operate in the background. Therefore, when analysts need to run queries on large datasets, they have to let them run from their desktop for hours or submit them as a batch job.

This bogs down the entire process of exploration and analysis, which is inherently iterative. Analysts need a responsive, interactive experience, no matter how large the dataset is. Simply put, our previous generation of BI tools didn’t support that.

Data lineage and governance concerns
Data lineage is the ability to track data from its creation throughout its history, incorporating all the transformations it has undergone over its lifetime. Implementing comprehensive data lineage and chain of custody (which tracks who has touched the data at each point in time) have always been onerous data governance tasks, but that doesn’t mean they are any less of a requirement.

A few years ago, retaining information about data lineage was a concern. Now in many cases it has become a mandate. One of the main reasons it has become a requirement for businesses is because of increased regulations to protect user privacy — especially with the EU’s General Data Protection Regulation (GDPR), which seeks to enhance data security for its citizens. Such regulations have made data governance vitally important.

Increasing regulation will put further strictures on data access. Analysts often do not have the ability to use data within the regulatory environment they now face. While there are merits to
protecting data, for analysts, data is more difficult to access. If nothing changes, it will become even more difficult.

The rise of the data lake and use of big data
With the rise of big data, data lakes have become widespread. But big data and data lakes have caused some problems for analytics. Businesses have found data lakes challenging. In part it’s because of the scale of data they house, the fact that their performance isn’t interactive, and their lack of schema or schema variability.

Particularly onerous is the fact that data lakes are generally only accessible by IT, impeding self-service analytics. Data lakes store very large datasets. It is very difficult for existing technologies to interact with datasets this large. Often, these mammoth datasets are only accessible by extracting subsets into a SQL repository. The alternative is to try to run a SQL engine on the data lake, but that approach is too slow to support interactive analysis.

The result is a slow data lake that is impractical to use through BI tools. To work around the slowness, IT moves subsets of the data into SQL engines and cubes, which results in a complex landscape and an environment where you can only ask questions you planned for in advance. It offers no iterative exploration and analysis, a fundamental requirement for self-service analytics.

The rise of JSON as an important repository for business information
JSON is a document-oriented format with a variable structure, widely used in web applications, mobile apps, and data integration projects. However, JSON has made creating a self-service environment more difficult. Organizations have dealt with this by converting JSON to relational format or by having developers create custom dashboards using web frameworks instead of BI tools. Both strategies reduce self-service by relying on IT. JSON has therefore been a direct cause of the step back in self-service progress.

Tools that were designed for SQL simply don’t handle JSON well, and those that handle it at all do so only on a superficial level. Analysts also find that it is very challenging to blend JSON data with other data.

The spike in microservices
Microservices are an important trend in application development, breaking application code into discrete services. For example, microservices enable a site like Amazon to have dozens of different data elements on it, all of which can run concurrently. Microservices offer businesses the tremendous upside of providing users with a customized experience. When you load a site like Amazon, you might have one microservice handling navigation, another that loads product reviews, another that makes customized recommendations, others for product services... and the list can go on. On a truly complicated page, you could have 150 microservices running simultaneously.

This reliance on microservices is all well and good because of the customized user experience it can provide. But on the other end of microservices is the analyst, who faces a lot of hurdles. It’s a Humpty Dumpty like situation: Deconstructing data in smaller pieces means that eventually, someone has to put it back together for it to be useful or the business user can’t put it to work.
Integration becomes a crucial challenge. This has radically increased the number of repositories that analysts now have to access, and quite obviously, has made self-service analytics more difficult as data consumers have to wait on IT to put data together before it can be analyzed.

The way ahead

So given all of this complexity, how do we turn the tide toward self-service analytics?

Companies need a platform that allows them to integrate a variety of data sources, with data in multiple forms, and bring all of that information together in a way that accelerates analytics. This type of system also needs to avoid creating extracts of data so that copies don’t proliferate. And this process has to be operable by analysts themselves, rather than intermediated by IT, empowering analysts to explore and interact with all desired data and iterate on their findings.

Dremio is such a platform. Dremio is founded on the idea that the only way to solve the self-service analytics challenge is to simplify the entire data analytics stack. Dremio believes that a virtual view that allows data to be selected, combined, and transformed into the desired form is the right way to deliver self-service data access. And Dremio uses an in-memory columnar engine through the Apache Arrow project to deliver maximum query speed, memory, efficiency, and computing power. Dremio relies on SQL as the search language for queries, which also opens up the ability for analysts to interactively query the data themselves, using their favorite tools (for example, Tableau, Python, R).

Finally, Dremio is premised on the idea that a true self-service platform must be able to accelerate data and queries, and integrate all needed capabilities in a single solution. Once analysts are able to help themselves, IT can play a more strategic role in securing and governing the system that allows self-service by business users.

There’s no question that self-service analytics has gone backwards over the past decade. Dremio provides a comprehensive, open source approach to achieve self-service analytics.

This paper was created by CITO Research and sponsored by Dremio

About Dremio Corporation
Dremio is the Data-as-a-Service Platform. Created by veterans of open source and big data technologies, and the co-creators of Apache Arrow, Dremio is a fundamentally new approach to data analytics that helps companies get more value from their data, faster. Dremio makes data engineering teams more productive, and data consumers more self-sufficient. For more information, visit www.dremio.com.

Founded in 2015, Dremio is headquartered in Santa Clara, CA. Investors include Lightspeed Venture Partners, Redpoint, Norwest Venture Partners, and Cisco Investments. Connect with Dremio on GitHub, LinkedIn, Twitter, and Facebook.