David Chappell

ANALYTICAL SCENARIOS USING THE MICROSOFT DATA PLATFORM

A GUIDE FOR IT LEADERS

Sponsored by Microsoft Corporation

Contents

Microsoft's Data Platform: The Big Picture

We use data in many different ways, and the volume, variety, and velocity of that data increase every day. Because of this, organizations rely on lots of different data technologies. Taken as a group, these technologies make up a *data platform*.

One way to think about the technologies in a data platform is to divide them into three categories based on the kind of data they work with. Those categories are:

- *Operational data*, such as transactional data used by a banking system, an online retailer, or an ERP application. This data is typically both read and written by applications, commonly in response to user requests. A banking application might read your account balance, for instance, then write a new value to reflect a deposit you make. And while operational data was once almost entirely relational, the increasing volume and variety of data have changed this. Today, working with unstructured operational data can be just as important.

- *Analytical data*, such as the information kept in a data warehouse. This data is typically read-only, and it usually includes historical information extracted over time from other data sources, such as operational databases. Analytical data is commonly used for things such as business intelligence and machine learning, and like operational data, it can be either relational or unstructured.

- *Streaming data*, such as data produced by sensors. The defining characteristic of streaming data is velocity; if the data isn't processed quickly, it can lose a large share of its value. Many streaming scenarios today relate to the Internet of Things (IoT), where the focus is on interacting with data provided by lots of devices. Streaming data is also used in other situations, such as analyzing financial transactions as they happen. In both cases, the challenge is to work effectively with large amounts of data being produced in real time.

The Microsoft data platform provides technologies for all three categories, along with connections among the three. Figure 1 summarizes the platform's offerings in each area.

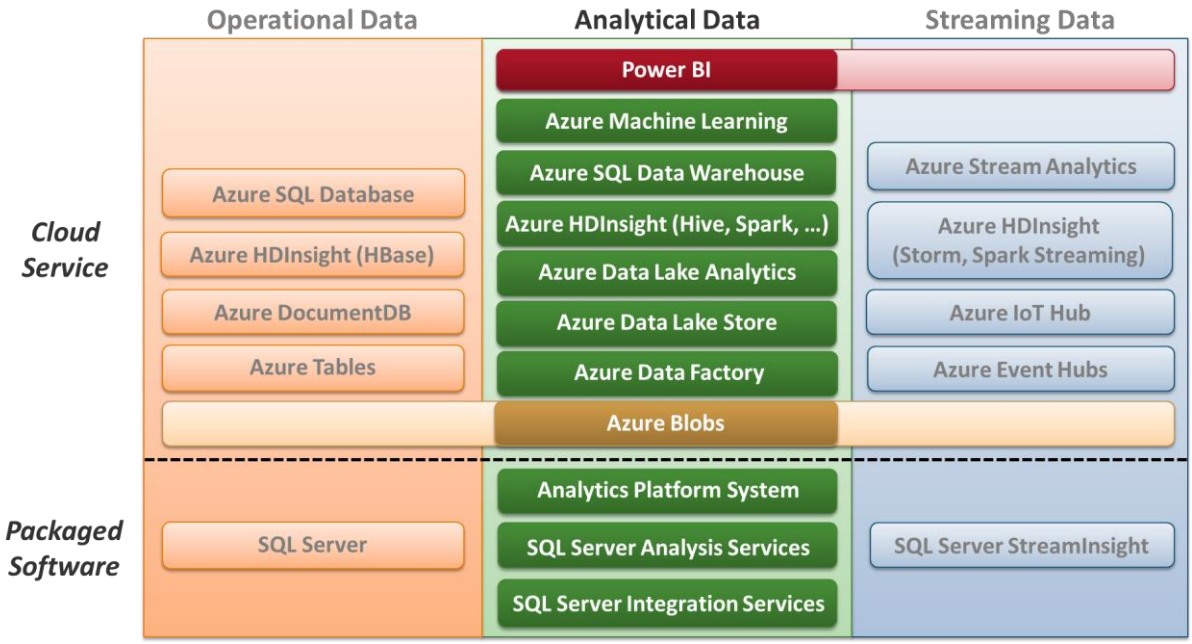

Figure 1: The Microsoft data platform includes cloud services and packaged software for working with operational data, analytical data, and streaming data.

This paper focuses on the middle column in the figure, Microsoft's offerings for working with analytical data. (For more on the other two categories, see the companion papers *Operational Data Scenarios Using the Microsoft Data Platform* and *Streaming Data Scenarios Using the Microsoft Data Platform*.) And don't be confused by the diagram: These technologies aren't layered in the sense that each one depends on the others shown below it. Instead, think of each column as a group of technologies for working with data in a particular way. Also, realize that the lines between the columns are permeable—these technologies can be used together in various combinations. For example, the analytical technologies in the center column are often used together with both the operational technologies in the left column and the streaming technologies in the right column.

The easiest way to understand these technologies—and more important, to understand how they can help your organization—is to walk through scenarios that use them. And given how important cloud computing is to IT leaders today, those scenarios should all use the cloud in some way. This paper looks at five analytical data challenges organizations often face, describing how the Microsoft data platform addresses each one. The scenarios are the following:

- Providing a common user interface for diverse data.
- Analyzing large amounts of relational data.
- Analyzing large amounts of unstructured data.
- Using large amounts of data to make better predictions.
- Using historical data to anticipate customer behavior.

Along the way, we'll take a brief look at each of the analytical data technologies shown in Figure 1. The goal is to provide a big-picture view of how the Microsoft data platform addresses the challenges of working with analytical data.

Scenario: Providing a Common User Interface for Diverse Data

Analysis—looking for patterns, relationships, and insights—is among the most important ways we use data. It's also among the most diverse, since there are many different analysis technologies. Whatever technologies you choose, though, the goal is often to display the results to people.

But having a separate user interface (UI) for each analysis technology makes life complicated for users. Why not instead provide a common way to present and work with data from many different sources? And why not make this UI widely accessible by running it in the cloud? This is exactly what Microsoft does with Power BI.

Technology Snapshot: Power BI

Power BI is a cloud-based service that lets users access diverse data from anywhere. It can present up-to-the-minute views of data from many different sources, then make those views accessible on desktops and mobile devices, including iOS and Android phones. The sources of data can include on-premises analysis technologies, analysis services that run in the cloud, and cloud applications from Microsoft and other vendors. Figure 2 shows an example of a Power BI interface.

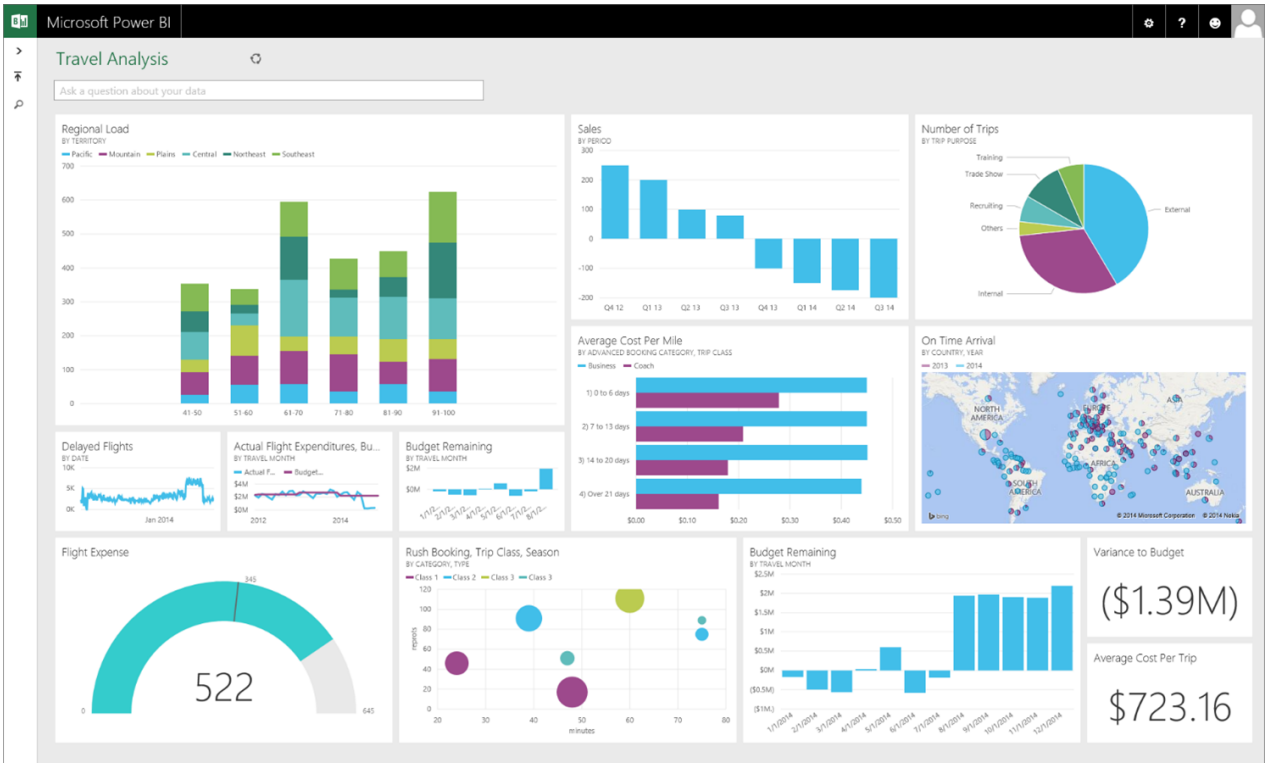

Figure 2: Power BI is a cloud-based service that provides a common user interface to data from many sources, including analytical data.

As this example suggests, Power BI can display information from many different sources in a unified way. Just as important, business users can use Power BI to define these interfaces and reports themselves—they don't need to rely on developers. Power BI also provides pre-built dashboards and reports for Office 365, Salesforce.com CRM, and other cloud applications. The tool supports natural language query as well, letting you ask questions such as "What are total sales by hour for diapers as a line chart?", then get back a graphical answer. All of these things have a common goal: providing a modern UI for accessing diverse data from anywhere.

Technology Snapshot: SQL Server Analysis Services

If you're like most IT leaders, you know that cloud computing will play a bigger role in your organization's future. But you also know that on-premises technologies will be important for many years to come.

Data analysis technologies provide a good example of this. Today, many organizations store periodic snapshots of operational data in on-premises data warehouses, then create business intelligence (BI) applications to analyze this data. In the Microsoft data platform, the fundamental technology for doing this is SQL Server. This relational database lets its users create data warehouses, then analyze the data they contain using SQL Server Analysis Services (SSAS). SSAS is a mature offering—it was first released in 1998—and it supports online analytical processing (OLAP), data mining, and more.

Describing the Scenario

Suppose that your organization uses SSAS to help your sales professionals monitor and understand current sales trends. The people who use this service might be at your headquarters, at some other location, or in the field. Along with this, they also need access to data provided by Office 365 and to customer information in Salesforce.com CRM. Your organization can use Power BI to provide a common UI to all of these, as Figure 3 shows.

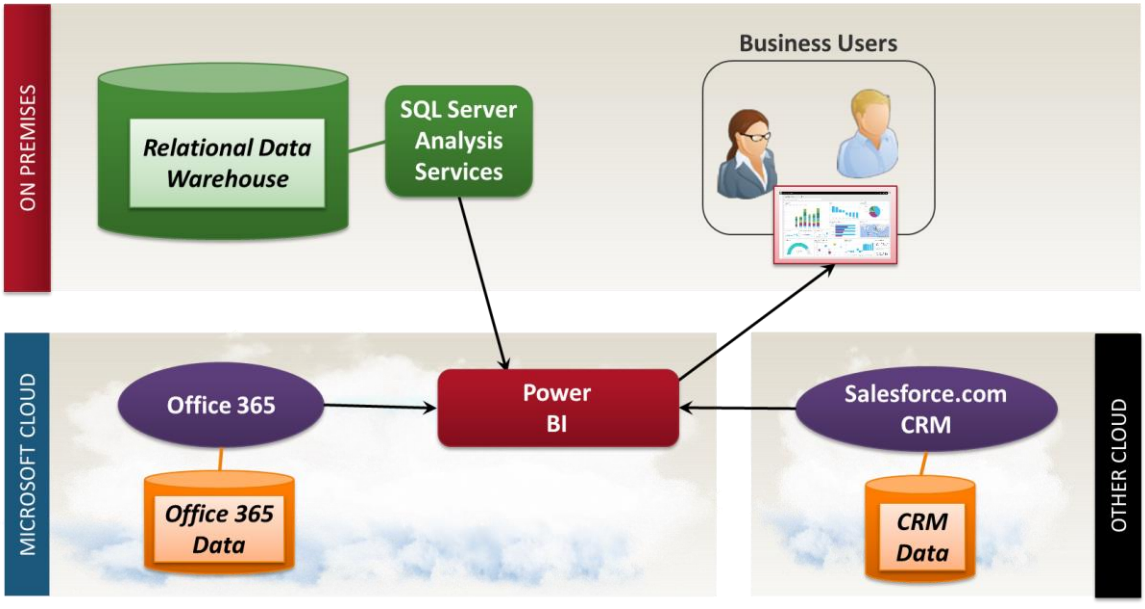

Figure 3: Power BI lets users access information from SSAS, Office 365, Salesforce.com, and many other data sources.

As this example shows, a Power BI dashboard can be connected directly to SSAS data. This lets a salesperson work with analytical data via this cloud-based service. He or she can also access information from applications running in the cloud or elsewhere through the same interface.

Understanding Your Options

IT leaders around the world face a common challenge: How should their organization adopt cloud technology? The cloud certainly has benefits, but it also has some clear risks.

Using Power BI can be a good place to start. The value of a common UI has obvious appeal—it can make your users happier. Starting here also lets you provide a widely accessible interface in the cloud while leaving critical data on premises, an approach that can minimize both regulatory concerns and your sense of risk.

Another way to get started with data analysis in the cloud is to do a new SSAS project on Microsoft Azure. Azure provides a technology called *infrastructure as a service (IaaS)* that lets you create virtual machines (VMs) on demand in Microsoft datacenters. It's possible to install SSAS and other software in these VMs, then run the environment much like your on-premises world. You might do this to save money, to get faster access to computing resources, or both.

Whatever approach you choose, one thing is clear: Microsoft's focus, in data analysis and other areas, is moving to the cloud. If you're a Microsoft customer, finding a way to adapt to this change should be a high priority for your IT organization.

Scenario: Analyzing Large Amounts of Relational Data

Data warehouses have always held large amounts of data. But in the last few years, that data has gotten even bigger. This implies more storage space, of course, but it also means that you need to process data faster to get quick results. One way to do this is to use massively parallel processing (MPP). Unlike the symmetric multiprocessing (SMP) typically used in traditional data analysis, MPP allows spreading a request across multiple machines, then executing that request in parallel. The result is significantly faster answers to many analytical queries.

Another change in data warehousing is that the information we need to store has gotten increasingly diverse. Along with relational data, warehouses today might also need to hold unstructured data. Modern applications create lots of this, including large log files, click trails from web applications, and more. As with relational data, there's business value in analyzing this information. And using a common technology to work with both unstructured and relational data can make life simpler.

The Microsoft data platform supports doing this in a couple of different ways. For on-premises data, the platform provides Analytics Platform System (APS). For data held in the cloud, there's a similar technology called Azure SQL Data Warehouse.

Technology Snapshot: Analytics Platform System

Many organizations today successfully use on-premises data warehouses created with SQL Server. But what if your situation requires handling many terabytes or even a few petabytes of relational data? APS is designed for scenarios like this.

APS is a dedicated hardware appliance that runs in your own datacenter, and it can handle petabytes of data. The appliance contains multiple physical servers, with the hardware supplied by Dell, HP, or another vendor. Applications running on APS use MPP, which lets them exploit the processing power of the appliance's multiple servers.

Yet in many organizations, the lion's share of their new data isn't relational—it's unstructured. For analyzing large amounts of unstructured data, the industry standard has become the Hadoop technology family. To let you work with both relational and unstructured data, APS also allows creating a Hadoop partition within the appliance.

Combining relational and unstructured data raises another question: How can an application issue a query against both? With APS, the answer is a technology called *PolyBase*. Using this technology, an application can issue standard T-SQL queries against relational data in APS, non-relational data in APS, or both, then let PolyBase handle the details of getting the result. Among other things, this lets users work with APS data from common tools such as Excel.

Technology Snapshot: Azure SQL Data Warehouse

APS lets you analyze large amounts of data in an on-premises appliance. But more and more of the data that you want to work with lives in the cloud. Maybe that data is created by a customer-facing web application running on Azure, for instance, or perhaps it's coming from devices in an IoT scenario that use Azure as a back end. Whatever the source, the problem is to store and analyze very large amounts of data in the cloud. To help you do this, the Microsoft data platform provides Azure SQL Data Warehouse.

To a great degree, SQL Data Warehouse replicates the functionality of APS in the cloud. Like APS, it can store large amounts of relational data, then let applications use MPP to execute high-performance queries across that data. It also supports PolyBase, letting you issue T-SQL queries across both relational and unstructured Hadoop data.

SQL Data Warehouse has an important difference from APS, however. APS is a physical appliance, which implies that you must choose the size you need when you buy the hardware. SQL Data Warehouse is a cloud service, so you can increase or decrease the processing resources you use as your needs change. And because it's a cloud service, you pay only for the resources you actually use.

Technology Snapshot: SQL Server Integration Services

To create and maintain a data warehouse, organizations regularly pull data into the warehouse from operational databases. The warehouse can be built using SQL Server or APS or SQL Data Warehouse or many other technologies, and the operational databases that provide the source data might use SQL Server, Oracle, a NoSQL technology, or something else. Whatever the specifics, the process is commonly called *extract, transform, and load (ETL)*, and it's usually automated.

SQL Server Integration Services (SSIS) is a technology for doing ETL and more. It can be used with many different data technologies, including those just listed, and it provides a drag-and-drop interface for defining data workflows. Like SSAS, SSIS is included with SQL Server, and it's become a widely used tool for data integration.

Describing the Scenario

APS and Azure SQL Data Warehouse offer similar services. Either one can potentially be used by applications running on premises, in the cloud, or both. Figure 4 shows a scenario that uses both technologies.

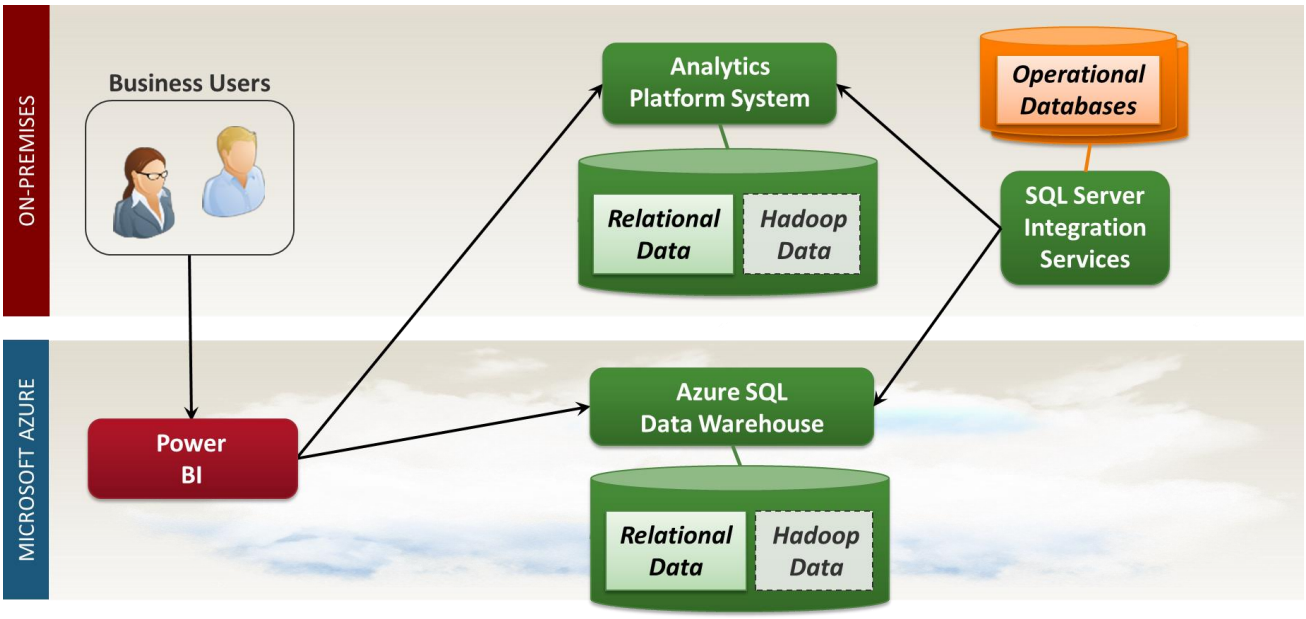

Figure 4: Analytics Platform System and Azure SQL Data Warehouse allow working with large amounts of data on premises and in the cloud, respectively.

In this scenario, SSIS is used to load data into both APS and Azure SQL Data Warehouse. This data might be identical, with a copy both on-premises and in the cloud, or it might be different. If some of the data you're storing is prohibited by law from living in the cloud, for example, SSIS could route this data exclusively to APS. Whatever the situation, data stored in both places can be accessed and analyzed using Power BI.

Understanding Your Options

Using either APS or Azure SQL Data Warehouse can make sense whenever using MPP makes sense. There are also cases where using the two together can be helpful, such as these:

- Aging data from on-premises storage to cloud storage. Suppose you have an on-premises application that needs the scale and MPP power of APS. Over time, the amount of data you need to store might outstrip even the capacity of this appliance. In a situation like this, you might choose to move older data that's accessed less frequently to Azure SQL Data Warehouse. This would likely make storage cheaper while still letting you get at this aged data from your existing MPP applications.

- Application development and testing in the cloud. While there are some differences between APS and Azure SQL Data Warehouse, the two provide similar services. Because of this, it's possible to create new MPP applications in the cloud, then run them on premises. This avoids the risk of development projects interfering with a production APS environment. It can also give development groups more control over the world they work in, since the team can create and use its own instance of Azure SQL Data Warehouse.

- Doing disaster recovery in the cloud. Suppose your organization has created one or more mission-critical applications using APS. In cases like this, having a disaster recovery solution is essential. What if your on-premises datacenter goes down because of a flood or an earthquake or human error? Azure SQL Data Warehouse can help solve this problem. Because this cloud technology is so much like APS, your on-premises

applications can potentially run in the cloud when they need to, such as when the on-premises appliance is unavailable.

Scenario: Analyzing Large Amounts of Diverse Data

For the last couple of decades, most data warehouses have focused on relational data. This shouldn't be surprising. The applications whose operational data fed these warehouses were using relational data, so that's what data analysts had to work with. Yet today's applications often create large amounts of unstructured data. Traditional technologies for analyzing relational data weren't appropriate for this new world. Something new was needed.

A number of technologies have been created to address this problem of analyzing large amounts of more diverse data. One of the first, created by the open source community, was the Hadoop technology family. Microsoft provides Hadoop (and more) as a service with Azure HDInsight.

Technology Snapshot: Azure HDInsight

While you're free to install and manage a Hadoop cluster yourself in Azure IaaS VMs, doing this isn't especially simple. To make using Hadoop easier, HDInsight implements this technology family as a cloud service. The goal is to make it faster and cheaper to run your own Hadoop cluster.

Among the technologies that HDInsight implements are the following:

- *Hadoop Distributed File System (HDFS)*, which provides a way to store very large files containing unstructured data, relational data, semi-structured data (such as JSON documents), or anything else. HDInsight supports the HDFS API over Azure Blobs (described later), which lets Hadoop applications access blob data in a familiar way.
- *MapReduce*, a low-level programming framework for applications that process HDFS data in parallel. Because these applications run across many machines at once, they can analyze large amounts of data more quickly than an application running on just one server. (The idea is analogous to the MPP approach used by APS and Azure SQL Data Warehouse.) HDInsight also supports newer Hadoop programming frameworks for building parallel data applications, such as Tez.
- *Hive*, which provides a higher-level approach for creating parallel data analysis applications. Hive includes HiveQL, a SQL-like language that can be used to express queries on unstructured data. These queries in turn create applications built on a lower-level framework such as MapReduce or Tez. HDInsight also supports other high-level Hadoop languages, such as Pig.
- *Spark,* which provides an alternative programming framework for running parallel data jobs on a Hadoop cluster. Unlike the traditional MapReduce approach, however, which writes intermediate data to disk, Spark keeps this information in memory. The result is much better performance for some kinds of applications. Spark also includes several other technologies, such as support for machine learning and for analyzing streaming data. One advantage of the Spark family over Hadoop is that Spark's main technologies were created to work together smoothly. With Hadoop, different technologies were built by different people at different times, so using them together can require extra effort.

HDInsight also includes other Hadoop technologies that fit in other categories—they're not designed to do data analysis. For example, HDInsight HBase is a NoSQL database for working with operational data, while HDInisight Storm is designed for working with streaming data.

Technology Snapshot: Azure Data Lake Analytics

Hadoop—and HDInsight—require creating a cluster. But suppose you just want to run parallel data analysis applications without worrying about clusters—what then? Azure Data Lake Analytics addresses this situation. Like HDInsight, this technology lets its users create queries that run in parallel across many different servers. Rather than explicitly requiring a cluster, though, an Azure Data Lake Analytics user can just specify how many servers a query should use. The system automatically allocates those resources to this query, then gives them up when the query is complete.

Azure Data Lake Analytics lets queries access diverse data from various sources, including Azure Data Lake Store, SQL Database, Blobs, SQL Data Warehouse, and more. Wherever the data comes from, the queries can be written in a common language called U-SQL. Derived from both SQL and C#, U-SQL can operate on both structured and relational data. And while a U-SQL query runs in parallel, this parallelism is handled automatically—developers needn't worry about the details.

Technology Snapshot: Azure Data Lake Store

HDFS is becoming today's most common approach for storing large amounts of diverse data. While HDInsight supports the HDFS API over Azure Blobs, this implementation is mainly intended to be used by HDInsight applications. To make HDFS more generally available, the Microsoft data platform includes Azure Data Lake Store.

Azure Data Lake Store provides HDFS as a cloud service. Unlike relational data warehouses, which commonly fit data into a strict schema, HDFS—and Azure Data Lake Store—can hold pretty much any kind of data, whether or not it has a defined schema. Applications running on HDInsight can read and write data stored in this service, as can U-SQL queries running in Azure Data Lake Analytics. Azure also supports popular Hadoop distributions running in IaaS VMs, including Hortonworks, Cloudera, and MapR, and applications running in these environments can use Azure Data Lake Store as well. Even non-Hadoop applications can use this storage service, since access is via the industry-standard RESTful interface defined by WebHDFS.

Technology Snapshot: Azure Blobs

The term "blob" is an acronym for Binary Large OBject, and that's exactly what Azure Blobs stores: raw binary data, Blob storage is quite scalable—a single blob can hold up to 200 gigabytes—and relatively inexpensive at just a few cents per gigabyte per month. An application might use it to store log data, images displayed to users, or pretty much anything else.

Describing the Scenario

Suppose a development team in your organization creates a new e-commerce application. The team might choose to use a NoSQL database, such as Azure's DocumentDB, for the application's operational data. Since this operational data isn't relational, putting the application's historical data in a traditional relational data warehouse won't do—something else is required. And suppose the application also generates unstructured data in other ways, such as by tracking the clickstream each user creates. Tracking and analyzing this data is important, since it can help you understand customer behavior and improve the application over time.

In situations like this, HDInsight, Data Lake Analytics, Data Lake Store, and Blobs can help. Figure 5 shows how things might look.

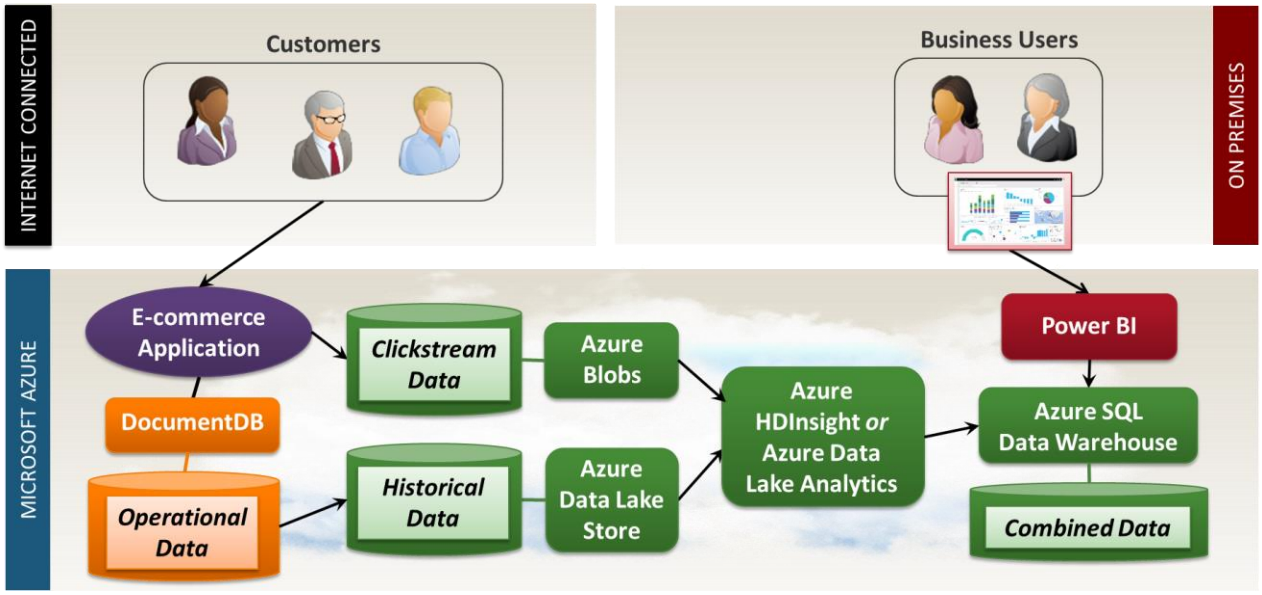

Figure 5: Azure Blobs, Azure Data Lake Store, Azure HDInsight, and Azure Data Lake Analytics can be used together to store and analyze large amounts of diverse data.

In this example, the application writes clickstream data directly into Azure Blobs. This data is voluminous, and it doesn't need much structure, so low-cost blob storage is a good option. For historical data, snapshots of operational data are moved periodically into Azure Data Lake Store, allowing it to be accessed by any application that can use HDFS.

Both kinds of data can be analyzed using either HDInsight or Azure Data Lake Analytics. An analysis application might issue HiveQL queries, for instance, each of which creates a MapReduce job that processes the relevant data in parallel. Alternatively, this application might be written as a U-SQL query run on Azure Data Lake Analytics. In either case, the result can be written to Azure SQL Data Warehouse, where it can be accessed by business users through Power BI. For instance, a Power BI dashboard might show things such as the number of current customers, what countries they're coming from, how many shopping carts were abandoned without a purchase in the last hour, and any other aggregate information that's important for this business. Users might also user Power BI to explore this combined data to look for trends and other useful information.

Understanding Your Options

As just described, the Microsoft data platform provides a variety of technologies for working with large amounts of diverse data. When you're making a decision about when and how to use the technologies this section describes, there are several things to keep in mind. Among the most important are the following:

- Hadoop is an ecosystem of related technologies. If you choose one Hadoop technology, it often makes sense to choose others from this family. For example, while the scenario in Figure 5 uses DocumentDB as its NoSQL store for operational data, the developers could have chosen HDInsight HBase instead. If you know you'll be

using Hive to analyze data, it might make sense to use HBase rather than DocumentDB. Because HBase runs on an HDInsight cluster, management and billing is likely to be simpler than if you choose another NoSQL database.

- It's not always obvious whether HDInsight or Azure Data Lake Analytics is the best choice for a particular situation. One way to think about the difference between these two technologies is to realize that what HDInsight really provides is clusters as a service. It lets you easily create a Hadoop cluster, then run applications using Hive or Spark or something else on that cluster. You pay for the VMs in that cluster as long as they're running. Azure Data Lake Analytics, however, provides parallel queries as a service. You can create a query, specify the number of nodes you'd like that query to execute on (that is, how parallel the query should be), then run the query. You're charged only for the compute time the query uses, and there's no need to set up your own cluster. Depending on what you're doing, the price difference these two options might be substantial.

- Both Azure Data Lake and Azure Blobs store unstructured data, and HDInsight applications can access both. Which one should you choose? The answer depends on the specifics of your situation. Applications access Blobs through an Azure-specific interface, for example, while Data Lake supports the industry-standard HDFS interface. Also, a single file in Data Lake can hold a petabyte of data, much more than a single blob. A simple way to think about it is that because Data Lake is designed to be accessed by parallel applications, it's a good choice for data that will be analyzed in this way—you'll see better performance. Blobs are a more attractive solution in simpler situations, such as holding images that an application displays to users.

- As described earlier, both Analytics Platform System and Azure SQL Data Warehouse can store and analyze unstructured data. But doesn't this sound a lot like what HDInsight is for? It certainly does, and so rather than reinvent the wheel, these technologies actually reuse what's already there. APS uses HortonWorks to create a Hadoop region in the appliance, while Azure SQL Data Warehouse relies directly on HDInsight to store and process Hadoop data in the cloud.

Scenario: Using Large Amounts of Data to Make Better Predictions

Data often contains patterns, especially if you have lots of it. But those patterns are frequently too hard for people to identify. Instead, the technology of *machine learning* can be used to find them, then to recognize the same patterns when they appear in new data. One way to think about this is to view machine learning as *predictive* analytics, a way to more accurately determine what's likely to happen.

Machine learning can be applied in many different areas, including fraud detection, recommending new movies that customers might like, predicting which customers are likely to switch to a competitor, and more. The challenge is to acquire enough data to find repeated patterns, then effectively use machine learning to make correct predictions based on that data. To help your organization do this, the Microsoft data platform provides Azure Machine Learning (ML).

Technology Snapshot: Azure Machine Learning

As its name suggests, Azure Machine Learning is a cloud service for doing predictive analytics. Starting with raw data, Azure ML provides tools for cleansing that data (e.g., removing duplicate records), then running different learning algorithms on the data to search for patterns. Azure ML can then generate a *model*, which is software that

an application calls to detect pattern matches in new data. The model can return a probability indicating how strong the match is. This lets the application make better decisions about what to do.

For example, Azure ML might read lots of data about credit card transactions, then work out a pattern for determining whether a new transaction is fraudulent. Once it's done this, it can create a model that implements that knowledge. Other applications can call this model to determine whether a new credit card transaction is likely to be fraudulent. To make all of this easier to do, Azure ML provides ML Studio, a graphical tool that lets its users work with data and run experiments to create the best possible model.

Describing the Scenario

Suppose a credit card company has collected large amounts of data about previous transactions, storing this data in Azure Blobs. For each transaction, the data contains many things: the customer's age, gender, and nationality, where and when the transaction took place, and lots more. The record for each transaction also indicates whether it turned out to be fraudulent.

Azure ML can examine this data looking for patterns. For example, maybe American men between 25 and 29 years old using a credit card after midnight at a casino in Monaco had especially high rates of fraud. If so, Azure ML can generate a model that recognizes this pattern in new data. Figure 6 shows how this might look.

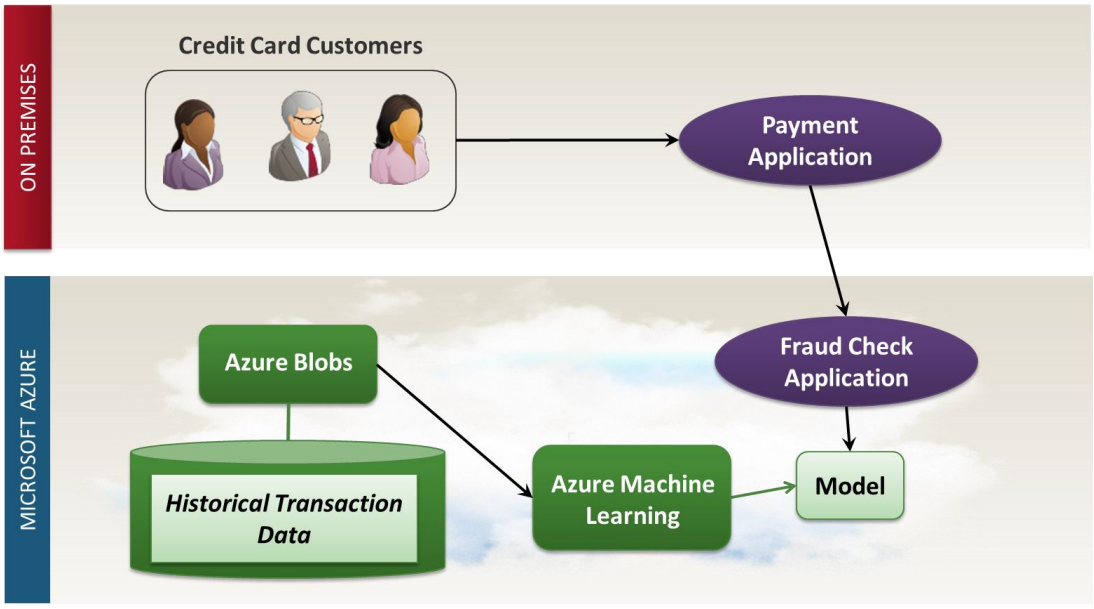

Figure 6: Azure ML can use historical transaction data to help predict whether a current transaction is fraudulent.

When a payment application is presented with a new credit card transaction, it can call a fraud check application that in turn invokes an Azure ML model. Based on what it's learned from the large set of historical transaction data, the model will then return the probability that this transaction is false. The payment application uses this probability to decide whether it should accept or reject this new transaction.

It's important to realize that the choice to accept or reject a transaction is a business decision. In some cases, an organization might opt to accept even a high-risk transaction, if only to avoid offending a good customer. Azure ML can help you recognize patterns, but it doesn't tell you what to do when it finds one—that's up to you.

Understanding Your Options

Like everything else in the Microsoft data platform, you have choices to make when you use Azure ML. Here are some of the most significant:

- The scenario illustrated in Figure 6 shows Azure ML reading from Azure Blobs, which is a popular option. Azure ML can also read data from other sources, including HDInsight, operational data technologies such as Azure SQL Database, and more. It's also possible to upload files from your on-premises disk directly into Azure ML.
- Azure ML provides several different modules for data cleansing, and it also implements a number of machine learning algorithms to find patterns in the data. If you need to go beyond these, the service allows writing custom modules in R, the lingua franca of data science. The Microsoft data platform also includes R Server, providing a parallel implementation of R and more.

Cortana Analytics Suite

The Microsoft data platform includes a variety of different analysis technologies, each addressing a specific area. But creating a complete solution to many problems requires using these technologies together, as in the scenarios described here. To make this easier to do, Microsoft offers a unified approach with the Cortana Analytics Suite.

As its name suggests, this set of offerings lets you access the results of your data analysis through Cortana, Microsoft's digital personal assistant. Combining Cortana's speech recognition with the natural language query ability of Power BI lets you speak questions to your analytics solution and get back answers. The suite also includes many other components, including Azure ML, HDInsight, SQL Data Warehouse, Data Lake, and more. The goal is to give you a single way to pay for and use this group of data analysis technologies together.

Scenario: Using Historical Data to Anticipate Customer Behavior

Each of the components in the Microsoft data platform can provide value on its own. But as with other technologies, they can be even more valuable when they're used together. Combining, say, Azure ML with Power BI and HDInsight and Azure Blobs can let you do things such as determine whether a customer is likely to switch to one of your competitors.

Doing this well requires a way to automate interactions among the technologies you're using. In the Microsoft data platform, the cloud service that does this is Azure Data Factory.

Technology Snapshot: Azure Data Factory

Analytical data scenarios often have many moving parts. For example, building and using a data warehouse typically means pulling data out of one or more operational databases, inserting it into the warehouse, then running an analysis application on the warehouse's data. Similarly, using machine learning might require aggregating very large amounts of raw data into manageable chunks, invoking machine learning technology on that aggregated data, then displaying the result to users. It's possible to do these things manually, but if they're done regularly, it makes more sense to automate them.

The Microsoft data platform provides Azure Data Factory to address problems like these. This cloud service lets you define and run automated processes that move and analyze data. The data it works with can come from various places: Azure data stores such as SQL Database or Blobs, databases in your own datacenter, or somewhere else. The service also allows monitoring data pipelines to make sure they're functioning properly. Rather than require people to execute these processes by hand, Data Factory lets a cloud service handle the task.

Describing the Scenario

Suppose a mobile phone company wishes to know which customers are likely to switch to another carrier, that is, to churn. This is useful information, and different people within the company will want different things. The firm's leaders might want a regularly updated list of the biggest at-risk customer organizations, letting them know where to focus their sales resources. Call center operators, by contrast, need to know the propensity to churn for each incoming caller. Maybe the operators are empowered to make different offers to different customers, for instance, based on each customer's likelihood of leaving.

By using several parts of the Microsoft data platform together, plus the right customer data, the mobile phone company can solve this problem. Figure 7 illustrates the solution's components and how they fit together.

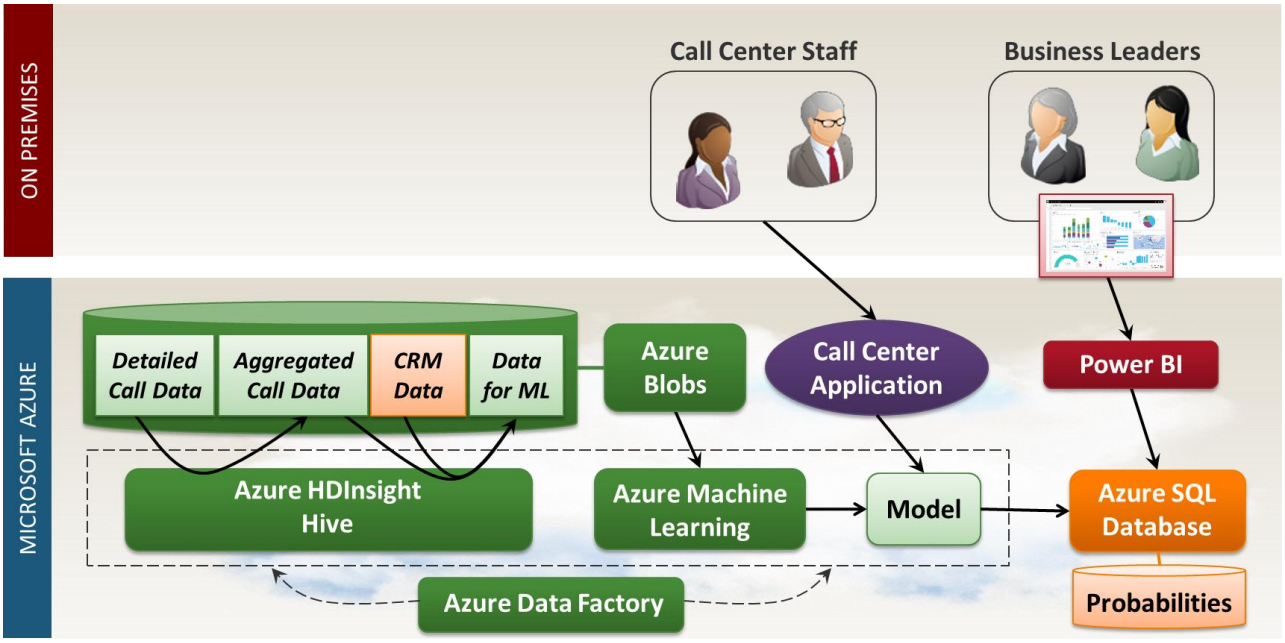

Figure 7: Multiple data platform technologies can be combined to predict how likely a customer is to switch to a competitor.

The phone company uses Azure Blobs to store detailed data about every phone call its customers make. Blobs are cheap and scalable, which makes them a good choice for storing the very large amounts of data this policy generates. The company can then use machine learning to scan the call data for patterns that indicate whether a customer is likely to switch to another carrier. Yet because Azure ML isn't designed to work with this much data, the phone company uses a Hive application running in HDInsight to aggregate the detailed call data into larger chunks. Azure ML then can then read this data to look for patterns.

But Azure ML doesn't rely solely on data about phone calls to make its predictions. In this scenario, another Hive job combines the aggregated call data with data from the firm's CRM system. Using CRM data along with the call data lets the solution identify and learn about customers, which helps Azure ML generate a better model.

The generated model is used in two different ways. First, as the figure shows, it's accessed by a call center application. When a customer calls into the call center, this application queries the model to learn the probability that he or she will switch to a competitor. The application then displays this information on the user interface used by the call center staff. Each of them now has up-to-date knowledge of their caller's propensity to churn and can act accordingly.

The Azure ML model is also used by Power BI to create a dashboard for the firm's business leaders. They don't need real-time information about potential customer churn, however, and so predictions are requested periodically for a specific group of customers. (Azure ML provides a batch-oriented API alongside its interactive API expressly for situations like this.) Each batch of predictions is stored in Azure SQL Database, where Power BI can access and display each batch whenever it needs to.

Over time, the model produced by Azure ML is likely to get out of date. To avoid this, the organization runs the steps within the dotted lines once a week to generate a new model. The firm's IT department uses Azure Data Factory to automate this entire process.

Understanding Your Options

This scenario isn't especially simple, but neither is the problem it solves. Many organizations won't start here—it might seem too complex. You might instead begin with some of the simpler scenarios shown earlier, minimizing your risk while you familiarize yourself with these new technologies.

Still, it's important to recognize that one of the main goals of the Microsoft data platform is to provide components that fit together well. To get the full benefit of the platform, expect to use multiple technologies in various combinations.

Conclusion

Doing data analysis well is hard. You're usually working with lots of data, whether it's relational, unstructured, or both. You're also trying to please a variety of stakeholders, each with different needs. Choosing an effective data platform can make this task easier. The platform needs to provide the right components for the problems you face, along with effective connections among those components.

Microsoft's goal is to provide a diverse and integrated data platform, both on premises and in the cloud. Whether you're working with analytical data, operational data, or streaming data, the Microsoft data platform provides interconnected technologies designed to work together. The result is a powerful set of products and services that address a broad set of data needs.

About the Author

David Chappell is Principal of Chappell & Associates (http://www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technologies.