Azure HDInsight improves performance of big data tax system and cuts costs

Microsoft Finance, which handles the company's international taxes, faces rapidly changing tax laws and data volumes that are expected to grow from 6 to 100 terabytes within a few years. This growth can affect core tax functions—like preparing data subsets to report sales or income taxes, and to comply with audits. In Core Services Engineering (formerly Microsoft IT), to improve global scalability and response times, we've transformed the group's infrastructure from a traditional data warehouse into a big data solution. Microsoft Finance uses the Microsoft Azure platform's elastic scaling and distributed computing and storage capabilities to:

- Significantly cut its response time to international tax audit requests.
- More efficiently process and report on large volumes (hundreds of terabytes) of data.
- Provide better support for business growth worldwide.
- Get insights for quicker, more proactive decisions.

Optimizing tax processes at Microsoft

We want to provide an optimal process for the Microsoft tax business to file returns, quickly respond to tax audit queries, and make proactive tax decisions. And we want to make it quicker and easier to do self-service reporting on aggregated data. As part of the improvement process, our technical requirements include:

- Creating a scalable platform that can load terabytes of data on demand.
- Optimizing and autoscaling Azure HDInsight cluster resources.

OneTax speeds up data loading and response time

The work we've been doing delivers on all these requirements. Last year—in Microsoft IT builds a big data tax solution for Finance with Azure—we described the big data solution we designed to tackle massive data growth, changing tax laws, and tax management in a services and subscriptions world of increased transactions.

OneTax, as the solution is known internally, aggregates raw data from multiple sources (including SAP) into an online data warehouse based on HDInsight, Azure Data Factory, and Azure SQL Database. OneTax was first implemented in the United States, and we'll gradually roll it out worldwide. Multiple billing systems, diverse tax laws, and manual processes had slowed reporting, but that's changing with the new platform. Data loading from upstream systems has gone from weeks to hours. When it comes to reporting, data extracts that took between 16 to 24 hours now take one hour. And queries that took six hours or more now take 10 minutes—an 800 percent productivity gain.

Big improvements since last year

Since last year, we've built on previous wins with OneTax and have made even bigger gains, including:

- **Faster data loading and response times without investing more in infrastructure.** By using Apache Spark instead of Apache Hive to improve our data loading from upstream systems, we've seen four times the performance. For the type of extraction, loading, and transformation (ELT) data loading scenarios that Microsoft Finance encounters, we've found that Spark has worked well.
- **Cost savings.** As part of our initial setup with Spark, we need fewer HDInsight clusters than with Hive and reduced our cluster size from 40 to 20 nodes. This smaller size lowers our infrastructure costs. In one case, we save $535 by using Spark for data processing of a 20-GB file. Multiply $535 by the number of files of this size, and the cost savings quickly add up. Also, to lower the cost of our Azure resources on an ongoing basis, we

autoscale clusters by creating a runbook in Azure—scaling up and down as needed. We've saved $1,000 a month by automatically deleting clusters when we don't need them.

- **Smaller cluster size suffices for processing, even large files.** In addition to cutting costs by having a smaller cluster, with Spark we can process even large files with fewer nodes than with a Hive cluster configuration.

Technologies used

We use the following technologies for the OneTax solution:

- Azure SQL Database—where we store UI-related metadata and master data lookup information for OneTax
- Azure Data Factory—cloud-based data orchestration service
- Azure Blob storage—where we compress and store our data
- Azure HDInsight:
  - Spark, for fast data loading—comes with HDInsight
  - Hive, for fast reporting and data extracts—comes with HDInsight
- Microsoft SQL Server virtual machines
- Common Hosted ETL Framework (CHEF)—the tool that we created for developers to automatically build pipelines, instead of spending six months manually transforming ELT packages
- Our internal reporting analytics tool, for our tax business users—a Microsoft Excel add-in available via Visual Studio Tools for Office (VSTO)
- Microsoft Power BI—for self-service analytics

What's Spark?

Spark is an open source processing framework that runs large-scale data analytics applications. Built on an in-memory computing engine, Spark enables high-performance queries on big data. It uses a parallel data processing framework that retains data in memory and on disk if needed. As a result, Spark delivers 100 times faster speed for tasks like ELT, batch processing, and interactive querying on data in Hadoop Distributed File System (HDFS). Azure makes Spark easy and cost-effective to deploy—there's no hardware to buy and no software to configure.

Why Spark?

There are many reasons we've been using Spark for our data loading/ELT scenarios:

- It improves data loading performance, especially from smaller to larger files.
- It efficiently handles in-memory operations, with fewer read/write operations to HDFS.
- It has more analytics capabilities than Hive.
- There's less code generated for data processing steps compared to Hive query language.
- We can add Application Insights telemetry while running rules.

Where we are now

Right now, we're using Spark in pilot mode. We have approximately 60 rules that address two to three upstream systems. These rules are going to user acceptance testing in March. By April, we should start going to production and onboarding some upstream systems to Spark. By June, we anticipate that we'll be completely on board in production with Spark for our data loading.

OneTax architecture

Since last year, in addition to moving from Hive to Spark, we've also somewhat changed our architecture. Figure 1 shows the current data flow into the OneTax app.

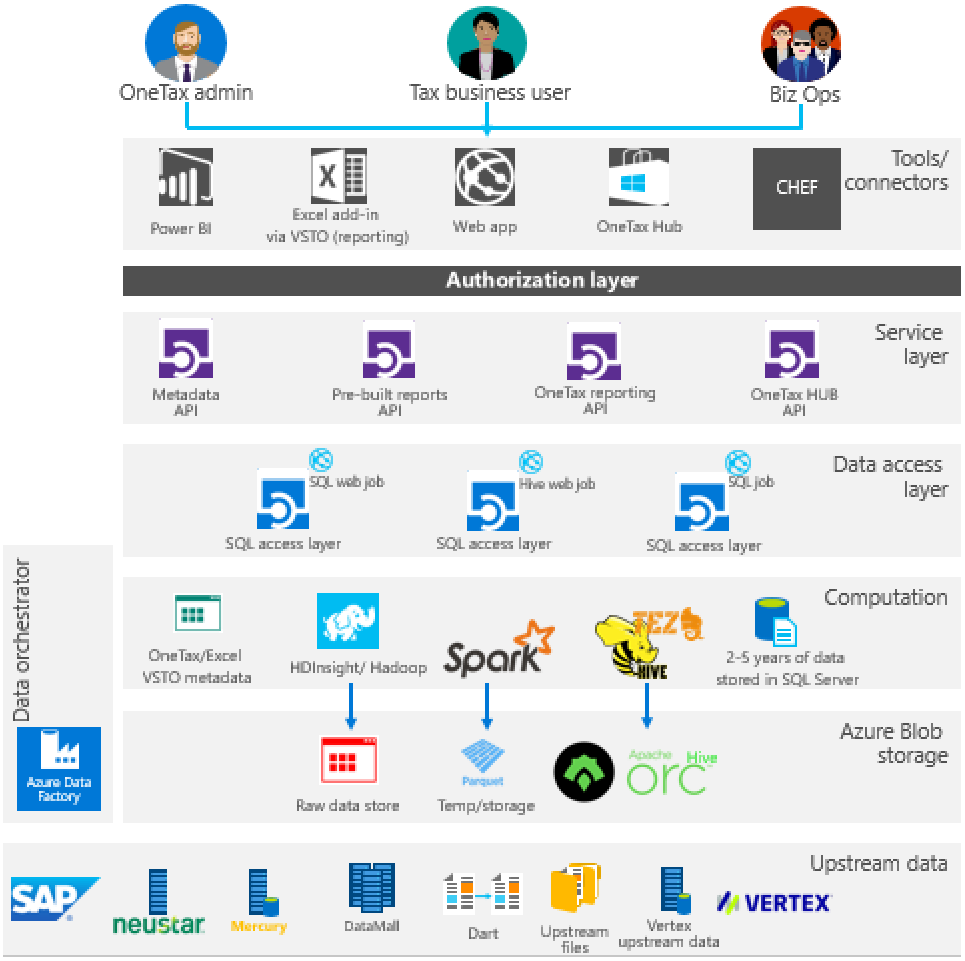


Figure 1. Data flow into our OneTax app

With this architecture, data is quickly loaded into the system. In turn, Finance employees can quickly generate reports.

Moving data from source to destination

We've onboarded new data sources and upstream data into our system. We now load 200 to 300 gigabytes of data a month versus the 20 to 50 gigabytes that we used to load each month. To move data from source to destination, we use Azure Data Factory, a cloud-based data orchestration service.

Storage, data ingestion, and processing

Azure Data Factory loads the data from upstream systems into Azure Blob storage, where we compress and store the data with formats like Apache Parquet, a format for compressing and storing columns of data.

Computation

The Spark engine reads and massages the raw data. We used to write hundreds of lines of code just to do simple data manipulation. But with Spark, far fewer lines of code are required.

When we ran queries in Hive, the data went into HDFS. Now, data goes into memory, and all operations are done there. This makes it much faster to manipulate data than with the Hive execution engine.

After the data is massaged into the appropriate reporting format, we move and store it in reporting tables that are in Hive-optimized row columnar (ORC) format—a format that's available out of the box from HDInsight. Think of it like columnstore indexing in Microsoft SQL Server. If we load 20 GB of data into our system and put it into ORC format, it's loaded as 2 GB. That's 10 times the compression rate by storing our reporting data in Hive ORC format.

Generating and displaying reports

For fast report generation, we run Hive with the Tez execution engine. Reports/data extracts used to be generated and displayed in the OneTax web app. But now, Microsoft Finance and tax business users generate them with our internal reporting analytics tool—a Microsoft Excel add-in available via VSTO.

Self-service analytics

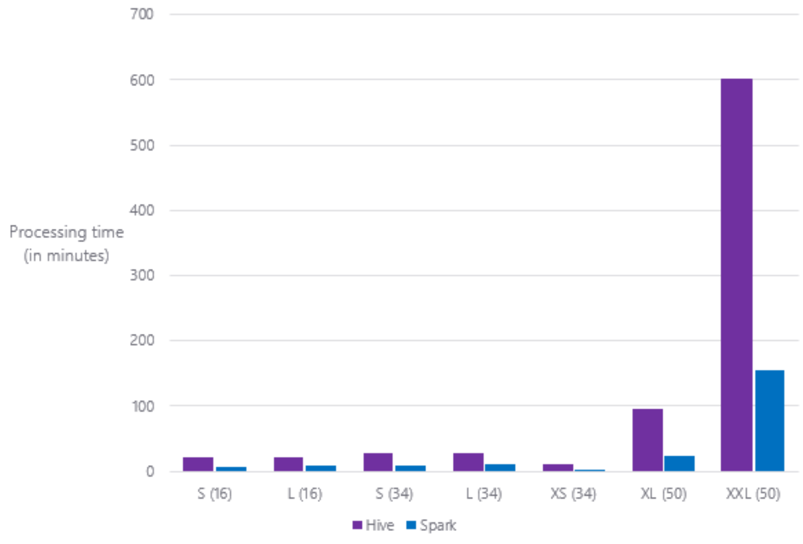
Employees can do their own analytics via Power BI.

Administrative activities

OneTax loads data from upstream systems into HDInsight, and all administrative activities are done through OneTax.

Performance gains

As we've seen from our system integration testing (SIT) data thus far, processing time and data loading are much quicker with Spark than with Hive. In fact, loading time has been reduced from weeks to hours. In our SIT environment, we tested Spark with different data sizes. Figure 2 compares Spark with Hive performance for small, large, extra-small, extra-large, and extra, extra-large files. The numbers in parentheses indicate the number of processing steps involved.



S=small: 500 MB, L=large: 2 GB, XS=extra small: 20 MB, XL=extra large: 5 GB, XXL=extra large: 20 GB
Numbers in parentheses=number of data processing steps

Figure 2. SIT data that shows Spark versus Hive performance

For example, we took a small 500-MB file, and ran 16 data processing steps. With Hive, it took 25 minutes, but with Spark it takes 5 minutes. That's more than four times the performance when we load data with Spark.

Spark reduces the processing time, and it also significantly reduces the number of clusters that we need—even to process larger files. Our Hive cluster was typically 40 nodes. With Spark, we can reduce our cluster size from 40 to 20 nodes. With the smaller cluster, we can load more data into our system than with Hive clusters.

When we load an extra-large 5-GB file into Hive with 50 data processing steps, it would take about 98 minutes. With Spark, it takes 25 minutes. Also, in Hive, the cluster wasn't fully used when we ran a query. If we run this same query in the Spark cluster, with fewer nodes, the cluster is fully utilized.

We've also seen quicker report generation by running Hive with the Tez execution engine. Suppose a business user requests a data extract—such as tax returns from the last five years for the month of February. Data extracts like these used to take 16 to 24 hours. Now they take only one hour.

Cost savings

When we initially set up Spark, we need fewer HDInsight clusters than with Hive. We reduced our cluster size from 40 nodes to 20 nodes, which lowers our infrastructure costs. Also, to reduce the cost of our Azure resources on an ongoing basis, we autoscale HDInsight clusters. To do this, we created a runbook in Azure. We wrote Windows PowerShell scripts to automatically delete clusters at night and on weekends or to scale them down to one node, when nobody's using them for data loading or reporting. Then, when we need the clusters again, we run the script and automatically scale up. We saved almost $1,000 a month by autoscaling.

An example of cost savings

Table 1 shows how using Spark to process a 20-GB file takes less time, uses fewer nodes, and saves money, compared to processing of a file of this same size with Hive.

Table 1. Example of processing a 20-GB file with Hive versus Spark

Hive	Spark
For 20 nodes, Hive processing took about 600 minutes.	The same data processing for the same 20-GB file in Spark—using only four nodes of the same cluster type—took 150 minutes.
Cost of data processing on a file of this size is $552.	Cost of data processing on the file is $17.
The total costs that we save by using Spark for data processing only one file of this size is $535. When we multiply that by the number of 20-GB files we might have, the cost savings are further increased.	

Looking forward

As part of a longer-term strategy, we're evaluating whether to use interactive reporting and telemetry in our production environment.

Interactive reporting

Right now, when someone submits a reporting request, it's done in batch mode. After the request is submitted, there's a 5-minute to 10-minute wait for report data. For interactive reporting clusters, we're exploring whether to use the Hive Long Live and Process (LLAP) cluster—a new Hive cluster type in preview phase in HDInsight. With LLAP, you can do interactive querying via in-memory computing, which should make Hive queries faster. Even though we aren't using this in production, our exploration so far has shown that generating our reports is 25 times faster with LLAP than with traditional Hive. It has also improved concurrent query processing.

Telemetry

We're exploring how to provide telemetry insights in our data processing steps. When we load data from upstream systems into our system, there are many steps to transform the data. We hope to use telemetry from the data loading process, so that the business can disable or reduce the number of steps. For example, if certain data processing rules generate errors or aren't getting applied, we can provide insight to the business that these rules are no longer valid, and should be removed from the data transformation. This can improve the speed and performance of data loading.

Conclusion

Through ongoing improvements to the OneTax management solution at Microsoft, Microsoft Finance has the insight and agility it needs for faster, data-driven decisions that improve global business processes. Since last year, we've made gains in performance and cost savings. Loading data and generating reports are faster, which leaves more time for analysis. And business analysts have easy access to data while the data is being transformed. Big data insights lead to better, more proactive decisions.

For more information

Microsoft IT Showcase

microsoft.com/ITShowcase

Microsoft IT builds a big data tax solution for Finance with Azure

IT Showcase data analytics content