

SQL Server 2016 improves performance of Microsoft revenue system

Today's digital transformation, digital sales, and consumption-based revenue model in Microsoft—where customers pay based on service usage—have brought exciting innovation. Yet the amount of data in our company's revenue reporting platform—MS Sales—has exploded from the time it was built 20 years ago. The original architecture can't efficiently handle the growth in transactions, the speed at which business rules change, and other complexities. Without highly specialized and costly hardware, our options for scaling up the architecture are very limited.

MS Sales is pivotal to strategic business, management, and financial decisions. It helps employees do revenue reporting and analysis on time, accurately, and consistently. Because these activities are crucial to assessing and maintaining our company's performance and competitive position, the system must be performant, fast, and reliable.

To fully scale means moving to Microsoft Azure, which will take a few years. Until then, at Microsoft IT, we're partnering closely with the Azure Customer Advisory Team to improve performance by upgrading our MS Sales system to Microsoft SQL Server 2016. We see this upgrade as a step toward our ultimate goal of migrating to Azure.

So far, we've seen the following results:

- In our system integration testing (SIT) and user acceptance testing (UAT), we've had a 10 percent performance improvement across all MS Sales processes and 12 percent improvement for one of our most critical processes.
- In all our environments, we've had a 10 percent to 12 percent performance improvement of reporting queries.
- These performance gains increased delivery capabilities of our MS Sales platform for quicker insights, while keeping costs flat.

Need quick insights, performance, and speed

When we sold box products like Microsoft Office or SQL Server, we had far less transactional data than with a consumption-based model. Customers who bought a box product like Office every three years for \$400 to \$500 might now buy an Office subscription for \$30 a month. If there are 12 transactions in a year (x3 years), the amount of transactional data increases. Multiply this by the number of total customers, and the amount increases exponentially.

With a consumption model, the data in the reporting engine has a much higher level of detail. Despite and because of this, performance and scale are crucial for us. It's essential that we have:

- Faster decision making using our data.
- Faster insights on our new and emerging businesses, like cloud and devices—MS Sales publishes once a day and within that day, there's lots of revenue coming in.
- High availability—for example, we can't have downtime the day before an earnings release.
- Faster speed—even though we have monthly and quarterly reporting cycles, management decisions have to happen more frequently than that.
- More seamless connection of our insights to related data, and enabling more business scenarios and capabilities.

Big data versus big, important data

For MS Sales and financial data, despite the increased variety and scale of transactions today, there isn't as much volume, variety, and velocity as with big data. But even though we don't strictly have what we'd call big data, we do have *big, important data*.

Data ingestion, management, governance, and normalization aren't really part of big data, but these are crucial tasks for us. To handle these tasks, SQL Server 2016 is a great value proposition.

MS Sales overview

MS Sales gives a consistent view of Microsoft businesses and production revenue. Using it, employees can make better, faster strategic decisions. They can query data that's related to purchases, sales, budget, and forecasts, and drill down to get more details about transactions. The system gives us summaries and details of revenue data for business scenarios like:

- Line-item details of an individual licensing agreement with sales partners based on the country/region of use.
- Dell distribution of Microsoft Windows.
- Revenue in Germany across all channels.
- Industry or vertical slices around the world.

MS Sales publishes data that's aligned with the Microsoft financial calendar. The publishing processes include daily, restatement, weekly forecast, budget, and—the most critical—fiscal month end (FME). There's a lot of service level agreement (SLA) pressure related to FME because MS Sales data is growing so fast. And we need more processing to keep pace with an expanding number of revenue records and related details.

Table 1 is a list of MS Sales, number of data sources, transactions, and other statistics.

Table 1. MS Sales–related statistics

	Statistics
Data sources (external)	1,458 (for example, Walmart, Best Buy, and so on)
Data sources (internal)	65 (for example, SAP, Skype, Navision, and so on)
Transactions received	2.6 million per day (approximately, at peak)
Data breadth/depth	13.2 billion rows, more than 2,200 fields, and SQL databases with more than 35 TB of data
Compute	
• Scale	Current: Approximately 200,000 rows/second (SQL Server)
• Business rules	54; more than 3.8 million rule mappings (dynamic SQL Server for processing)
• Org mapping	More than 4.1 million in a recent 30-day cycle (changes to transactions)
Publishing	
• Published database	17 x 5-TB data mart
• Data coverage	21 years (current, 10 prior, and 10 future)
• Publishes	20 times monthly; expected: ability to publish in near real time
• Downstream feeds	More than 220 service accounts
• Users	More than 35,000 globally, 22,000 MSRA for ad-hoc reporting
• Queries	More than 3.3 million queries were run against the revenue data mart in FY16
Servers	57 (production environment); the largest are 72 x 2 TB with extreme input/output

In a nutshell, MS Sales:

- Has more than 35,000 users at Microsoft (finance, sales, marketing, and executives), and gets data from more than 1,500 data sources (external and internal combined).
- Has about 22,000 users with advanced access who create ad-hoc reports with Microsoft Reporting Analytics (MSRA), a Microsoft Excel add-in.

- Is supported by 57 servers and about 2 terabytes of space. The system publishes data, but it also processes it in a way that supports load balancing.
- Has 21 years of data (the past 10 years, the current year, and 10 years into the future).

When we get inbound transactions, we have to understand what a particular customer is trying to achieve in our channels, and then tailor how we engage with them based on that. For example, org mapping helps us see the different interactions that customers have with Microsoft via products and services like Hotmail, Skype for Business, or Windows Phone. Another part of our evolving and complex business is product bundling—for example, phones that are sold with Skype and Office 365. Also, there can be different interaction points that a customer can have with Microsoft, like partner, direct sales, or retail environment. MS Sales has to make sense of all this information in a landscape of ever-increasing numbers of transactions, users, downstream feeds, queries, and business rules.

How we define success for MS Sales

SQL Server 2016 is a great way to help us increase our delivery capabilities within MS Sales, while keeping costs flat. We know that we've been successful when we've done the following:

- At a minimum, MS Sales must help employees have on-time, accurate, and available reporting. SQL Server 2016 helps ensure that the data is right, that we publish within our SLAs, and that we don't have service outages.
- After the minimum bar, we need rapid data publishing. Via customer surveys, we learned that 80 percent of our customers want faster data delivery. We need agility to adapt to new business scenarios and deployment velocity.
- We need to provide insights beyond revenue transactions. For example, what insights can we get on the new Azure offer that we rolled out?
- Our system must handle and adapt to the evolving needs of our security, privacy, and compliance mandates.

Effort required vs. benefits of upgrading thus far

We've upgraded all 50+ servers across 25 development, SIT, and UAT environments of MS Sales to SQL Server 2016. We're moving our entire production infrastructure to SQL Server 2016.

For our early-stage technology adoption effort, we had one person for two to three months who upgraded and performed regression testing. We also had two software engineer resources for two months who tested the entire MS Sales platform for compatibility and regression.

Without a large time and resource investment, we've already made valuable performance and process improvements. Plus—after we finish upgrading to SQL Server 2016—there are more benefits to come, as we explore using new SQL Server 2016 capabilities. For example, we'll explore using SQL Server Always On instead of database mirroring. And, to improve the performance of transaction processing, data ingestion, and data load, we'll look into using in-memory online transaction processing (OLTP) in SQL Server 2016.

Performance and process improvements

Here are performance-related improvements based on runtime statistics as data goes through the publishing process:

- In our SIT and UAT environments—just from using SQL Server 2016 out of the box—we've observed a 10 percent performance improvement across MS Sales processes.
- We've seen a 12 percent improvement for one of our most critical processes—the factory process.
- In all environments, we've seen a 10 percent to 12 percent performance improvement in reporting queries.

Technical implementation and architecture

Moving to SQL Server 2016 is the runway for the next version of our revenue reporting system—in the cloud at scale.

Technologies involved

In our upgrade from SQL Server 2014 to SQL Server 2016, we used the following technologies:

- [SQL Server 2016 Enterprise core-based licensing](#)
- Windows Server 2016
- SQL Server Analysis Services on our cube servers (for mining and making sense of the data)
- SQL Server Integration Services (for data integration and data extraction, transformation, and loading)
- SQL Server Database Engine (for storing, processing, and securing data and for fast transaction processing)
- Database mirroring (for high-availability of SQL Server databases)
- Log shipping (for automating database backups and transaction log files on a primary production database server, and then restoring them onto a standby server)
- Columnstore indexes (for quicker processing time of data warehousing queries)

Migration process and status

Right now, our top priority is to upgrade our production environment to SQL Server 2016. We're also experimenting with features in SQL Server 2016 to see how we can take full advantage of them, but we'll focus more on that after the migration. Here are some of the in-progress and completed activities in our overall migration process:

- We've upgraded more than 50 MS Sales servers to SQL Server 2016 in our development, SIT, and UAT environments. We plan to migrate all production servers to SQL Server 2016.
- To speed up the processing time of common data warehousing queries, we used columnstore indexes.
- We're looking at in-memory processing for uploading and managing some of our data.
- We plan to use Always-On Availability Groups to replace the database mirroring on our middle-tier servers.
- We did regression testing on MS Sales tools like MSRA to make sure that they operate well and are performant on SQL Server 2016.

This is the high-level, step-by-step approach we take for our upgrade:

1. We use architecture diagrams, and start with migrating just a few reporting servers to SQL Server 2016 rather than migrating them all at once.
2. We migrate component by component, and capture performance metrics to prove that it's technically feasible.
3. For each migrated component, we communicate our results, and show that it's a very solid return on investment.

Architectural overview of MS Sales

The MS Sales publishing schedule undergirds the MS Sales system and data flow. Given the data volume and up to 21 years' worth of data, not all transactions can be updated with each publishing event. So, MS Sales has several factory processes that publish specific sets of data on a fixed schedule.

For critical processes like FME reporting, the processing time can take 11 hours. When FME reporting data is published, the months of the current fiscal year are updated, in addition to transactions in the previous 24 months. A restatement process publishes changes to historic, current, and future billings, based on the most recent changes to business rules, revenue reporting standards, and organization structures in Microsoft.

We adhere to information security/compliance standards, including:

- The Sarbanes-Oxley Act.
- All rules and regulations for the Securities and Exchange Commission.
- All laws related to medium business impact and personally identifiable information.

Figure 1 gives a high-level overview of the hardware setup for ingestion, processing, and distribution of MS Sales data. It also shows the servers that we're upgrading to SQL Server 2016 and overall data flow.

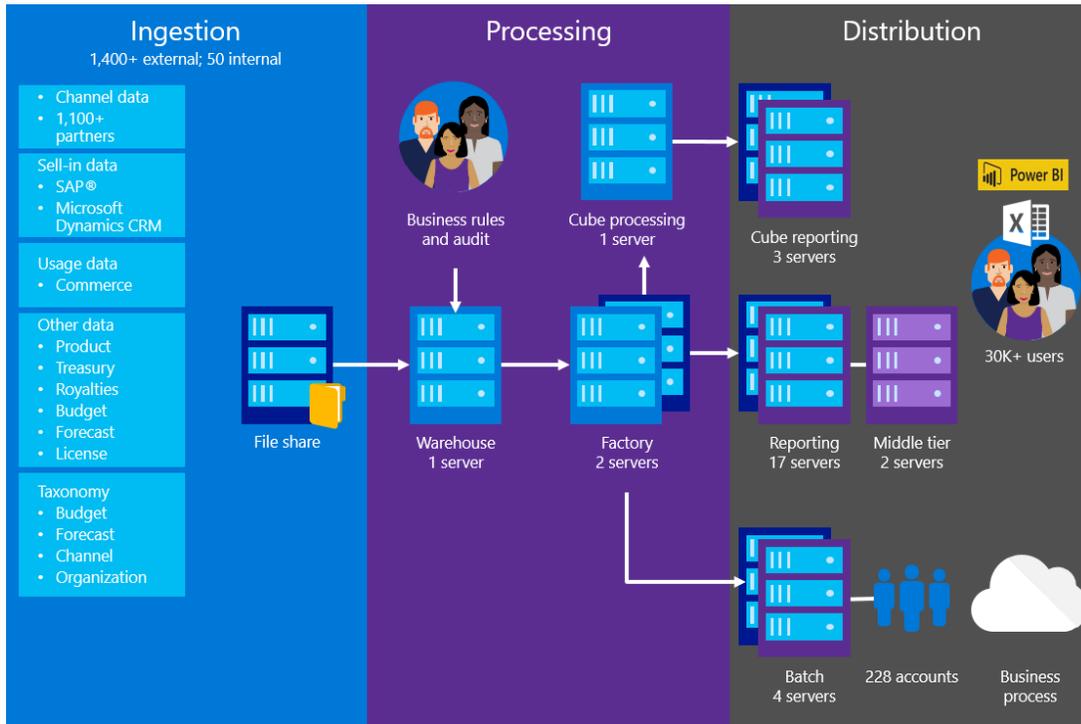


Figure 1. MS Sales architecture

Figure 1 has three main components:

- **Ingestion.** MS Sales has about 1,450 partner and internal data sources. Data goes from the file share into the warehouse server.
- **Processing.** We have a warehouse server for our business rules and audit-related data, two factory servers for factory jobs, and a cube processing server.
- **Distribution.** There are 3 cube reporting servers, 17 reporting servers, and 30,000 users who query these. MS Sales data is consumed via Standard Reporting, MSRA, and downstream apps. Microsoft Power BI is used for data visualization. There are four batch servers for business continuity.

Table 2 summarizes the hardware configuration, logical processors, memory, and storage of the servers.

Table 2. Hardware configuration

Server classification	Processor count	Memory	Storage
Warehouse	48	1 TB	20 TB
Factory	72	2 TB	25 TB
Middle-tier server	24	192 GB	2 TB
MSRA reporting server	48	1 TB	7 TB
Direct reporting server	48	1 TB	7 TB
Analysis server	48	1 TB	7 TB
Pump server	24	192 GB	7 TB

Technical limitations

MS Sales has a component that involves Azure SQL Database. We'd like to move data faster from on-premises to Azure because we regularly sync data between them in this hybrid scenario. We still use Microsoft Sync Framework and some other earlier technologies, but we hope to use SQL Server Stretch Database to move our data more quickly. Instead of scaling expensive on-premises storage, with SQL Server Stretch Database you extend or "stretch" data to the cloud—for much less cost and with easy online access to databases. Although it wasn't an exact match for our scenarios at the time, we plan to explore it again later, after we move all our environments to SQL Server 2016.

Risk mitigation and business continuity

As part of mitigating risk, we need to ensure that we have a predictable publishing cadence, good data quality, and a reliable infrastructure for those who use our tools.

From a business continuity standpoint, we have backup servers in a different datacenter and do periodic failovers. We have a continuous backup and recovery process for our failover servers. We're doing business continuity in phases, with a final business continuity exercise that will test our expected outcomes.

Expected outcomes of business continuity include being able to:

- Fail over to the business continuity environment within business time requirements.
- Run a daily factory job and publish to the business continuity environment.
- After we fail over, pull data from the MSRA reporting server and direct reporting server.
- Replicate the data back to the production datacenter from the business continuity datacenter.

Lessons learned

From our experience so far, here's what we've learned:

- If your servers have more than 40 logical cores, consider using SQL Server 2016 Enterprise core-based licensing to get the most out of the servers' processing power. When we began setting up the infrastructure, we used SQL Server 2016 Enterprise edition rather than SQL Server 2016 Enterprise core-based licensing. SQL Server 2016 Enterprise limits the logical CPUs that a SQL Server instance can use to 40. To maximize the potential of our servers and processing power, we decided to use the core-based licensing.
- Be aware of your trace flags. Trace flags temporarily set specific server characteristics or switch off a particular behavior. They also help diagnose performance issues and debug. Some flags are version-specific. So, a flag from an earlier SQL Server version could be obsolete or cause a different behavior in SQL Server 2016, and there could be performance issues. As we migrate our environments to SQL Server 2016, we look closely at the trace flags.
- You might notice performance degradation when you use Change Data Capture to capture changes on certain large tables that are more than 10 GB.

Looking ahead

After we completely migrate to SQL Server 2016—and outside the longer-term goal of moving to Azure—one key goal is to explore new SQL Server 2016 capabilities. Here's what our roadmap looks like:

- Continuously drive performance improvements.
- For high availability and business continuity, use SQL Server Always On instead of database mirroring.
- To improve the performance of transaction processing, data ingestion, and data load, take advantage of the in-memory OLTP capabilities in SQL Server 2016.
- To reduce maintenance and storage costs of on-premises data, explore using SQL Server Stretch Database.
- Possibly increase the cadence of our annual forecasting and budgeting cycles, from annual to weekly or monthly.

For more information

Microsoft IT

microsoft.com/ITShowcase

[SQL Server 2016](#)

[SQL Server 2016 Licensing](#)

[IT Showcase data analytics content](#)

© 2017 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.