

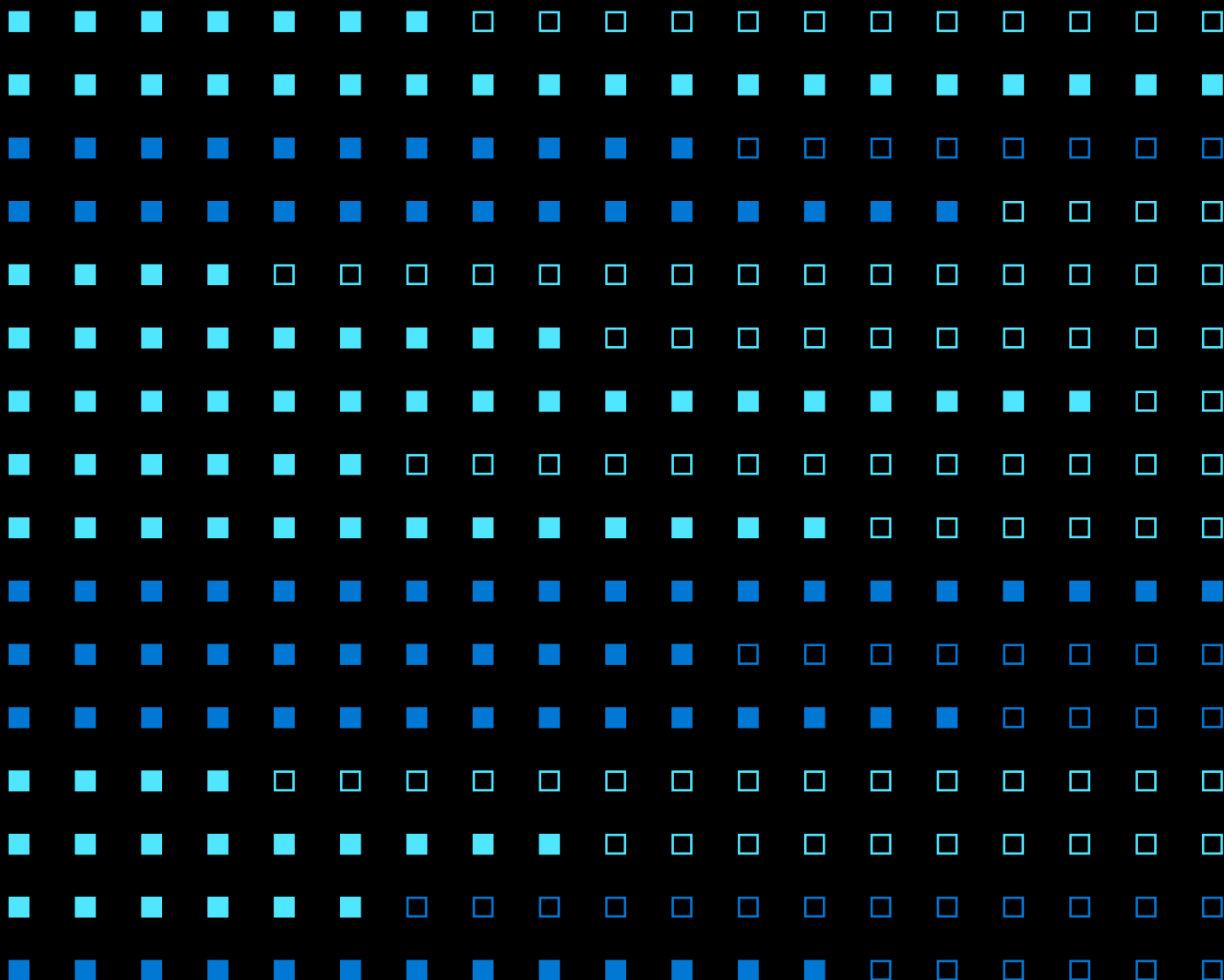
Microsoft SQL Server 2019

Big Data Clusters

Technical white paper

Published: September 2018

Applies to: Microsoft SQL Server 2019



Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. This content was developed prior to the product or service' release and as such, we cannot guarantee that all details included herein will be exactly as what is found in the shipping product. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. The information represents the product or service at the time this document was shared and should be used for planning purposes only.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Information subject to change at any time without prior notice.

Microsoft, Active Directory, Azure, Bing, Excel, Power BI, SharePoint, Silverlight, SQL Server, Visual Studio, Windows, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

© 2018 Microsoft Corporation. All rights reserved.

Contents

| | |
|---|----|
| Introduction..... | 4 |
| Data trends | 5 |
| Data virtualization..... | 5 |
| Big data and analytics..... | 5 |
| PolyBase - query over any type of database..... | 6 |
| Scalable, shared, and integrated compute and storage..... | 7 |
| Scale-out data marts | 8 |
| A shared data lake: read and write files in HDFS with SQL Server or Spark | 8 |
| A complete platform for AI and machine learning..... | 10 |
| Hybrid Deployment and Management Services Built In | 12 |
| Cluster administration portal | 12 |
| Controller service | 12 |
| Hybrid deployment and management..... | 12 |
| Conclusion..... | 13 |
| Calls to action..... | 13 |

Introduction

For 25 years, Microsoft SQL Server has been powering data-driven organizations. Relational databases have long been the norm and still hold a prominent place in any organization's data architecture. As the variety of types of data and the volume of that data has risen, the number of types of databases has risen dramatically. Today there are NoSQL databases, graph databases, in-memory databases, and big data provided by an ever-growing number of software vendors.

Over the years, SQL Server has kept pace by adding support for XML, JSON, in-memory, and graph data in the database. It has become a do-it-all, flexible database engine that enterprises can count on for industry-leading performance, high availability, and security. However, with its roots in a relational engine, a single instance of SQL Server was never designed or built to be a database engine for analytics on the scale of petabytes or exabytes. It also was not designed for scale-out compute for data processing or machine learning, nor for storing and analyzing data in unstructured formats, such as media files.

SQL Server 2019 extends its unified data platform to embrace big and unstructured data by integrating Spark and HDFS into a "big data cluster".

The SQL Server 2019 relational database engine in a big data cluster leverages an elastically scalable storage layer that integrates SQL Server and the Hadoop Distributed File System (HDFS) to scale to petabytes of data storage. The Spark engine that is now integrated into SQL Server enables data engineers and data scientists to harness the power of open source data preparation and query programming libraries to process and analyze high-volume data in a scalable, distributed, in-memory compute layer.

SQL Server 2019 big data clusters with enhancements to PolyBase act as a data hub to integrate structured and unstructured data from across the entire data estate—SQL Server, Oracle, Teradata, MongoDB, HDFS, and more—using familiar programming frameworks and data analysis tools.

When SQL Server 2017 added support to run on Linux, it opened the possibility of deeply integrating SQL Server with Spark, HDFS, and other big data components that are primarily Linux-based. SQL Server 2019 big data clusters take that to the next step by fully embracing the modern architecture of deploying applications—even stateful ones like a database—as containers on Kubernetes. Deploying SQL Server 2019 big data clusters on Kubernetes ensures a predictable, fast, and elastically-scalable deployment, regardless of where it is deployed. Big data clusters can be deployed in any cloud where there is a managed Kubernetes service, such as Azure Kubernetes Service (AKS), or in on-premises Kubernetes clusters, such as AKS on Azure Stack. Built-in management services provide log analytics, monitoring, backup, and high availability through an administrator portal, ensuring a consistent management experience wherever a big data cluster is deployed.

SQL Server 2019 big data clusters provide a complete AI platform. Data can be easily ingested via Spark Streaming or traditional SQL inserts and stored in HDFS, relational tables, graph, or JSON/XML. Data can be prepared by using either Spark jobs or Transact-SQL (T-SQL) queries and fed into machine learning model training routines in either Spark or the SQL Server master instance using a variety of programming languages, including Java, Python, R, and Scala. The resulting models can then be operationalized in batch scoring jobs in Spark, in T-SQL stored procedures for real-time scoring, or encapsulated in REST API containers hosted in the big data cluster.

Data trends

Data virtualization

Recognizing that different storage technologies are more appropriate for different types of data, an organization is likely to have data stored in a mixture of relational and non-relational data stores—often from several different vendors. A challenge for developers, data scientists, and analysts is that to extract value from this data, they typically need to combine data from these disparate sources. This problem is normally addressed by moving or copying all the relevant data from the source systems together on a single platform.

In traditional business intelligence systems, copies of data are created and loaded into a reporting platform with extract-transform-load (ETL) processes. Reporting and analysis are then carried out on these copies. While enabling enterprises to extract business value from their data, ETL processes have several issues:

- **Development, maintenance, and support challenges:** If they are to be repeatable and robust, ETL pipelines require effort to create, effort to keep them up to date, and effort to keep them running, especially as data models change over time.
- **Data latency:** ETL pipelines introduce an inherent delay. A recent study¹ found that more than 80% of data sets delivered by ETL pipelines is between 2 and 7 days old by the time it reaches an analytical system. 75% of businesses reported that delays in data processing had inhibited business opportunities.
- **Increased vulnerability:** Each copy of a data set must be secured against unauthorized access. It is difficult to consistently secure multiple copies of the data in disparate data storage and access systems.
- **Increased storage costs:** Each copy of a data set requires disk space to store, and these costs can be substantial if a data set is very large or is copied many times.
- **Compliance challenges:** As data is copied or moved around, the complexities of the governance of that data is multiplied, especially for organizations that are subject to strict data regulations such as GDPR.

While ETL has its use cases, an alternative to ETL is data virtualization, which integrates data from disparate sources, locations, and formats, without replicating or moving the data, to create a single "virtual" data layer. Data virtualization enables unified data services to support multiple applications and users. The virtual data layer—sometimes referred to as a *data hub*—allows users to query data from many sources through a single, unified interface. Access to sensitive data sets can be controlled from a single location. The delays inherent to ETL need not apply; data can always be up to date. Storage costs and data governance complexity are minimized.

Big data and analytics

With the onset of the Internet of Things (IoT), connected devices and people are generating volumes of data that exceed the storage capacity of any traditional database system. By some estimates², 90% of all the data in the world was generated in the last 2 years at a rate of 2.5×10^{18} bytes of data *per day*. This new type of data is often in formats that are not suitable for storing in relational database tables or for querying using relational query semantics.

The Apache Hadoop project introduced new technologies—storage in the Hadoop Distributed File System (HDFS), resource scheduling in YARN, and parallel compute query engine in MapReduce—to help store and analyze this new type of data known as “big data”. The big data ecosystem has evolved over time and now there are many

¹ 3rd Platform Information Management Requirements Survey, IDC, October, 2016, n=502

² “How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read,” 2018, Forbes.com

other components of a big data architecture that play a role in some aspect of a big data cluster, such as Knox or Ranger for security, Hive for providing structure around the data and enabling SQL queries over HDFS data, and many more.

The Apache Spark project added a powerful, new analytics engine to the big data ecosystem. Spark enables distributed, scalable compute to execute Java, Python, Scala, or R code on data stored in HDFS or other data sources. Data in Spark is cached in memory for super-fast data access. Spark is up to 100x faster than a Hadoop MapReduce query. Data scientists and engineers can use a rich and ever-expanding library of Python, R, Java, and Scala open-source code libraries to ingest and manipulate data, query it, and build and operationalize machine learning models.

PolyBase - query over any type of database

First added to the SQL Server database engine in SQL Server 2016, PolyBase enables applications and users to query big data stored in HDFS-compatible Hadoop distributions and file systems such as HortonWorks, Cloudera, and Azure Blob Storage by using T-SQL to define an external table to represent HDFS data in SQL Server. Users or applications can run T-SQL queries that reference the external table as if it were a normal SQL Server table. When the query is executed, data from the external data source is retrieved and returned to the user, but it is not stored in persistent storage in SQL Server.

Using this approach of querying data from the source ensures that there are no data latencies, duplicated storage costs, or data-quality issues introduced by ETL pipelines. Once you have created external tables in SQL Server, you can control access to data sources by granting access to external tables to Active Directory users and groups, thus centralizing the data access policies to a single location.

SQL Server 2019 extends the capabilities of PolyBase with new connectors to create external tables that link to a variety of data stores, including SQL Server, Azure SQL DB, Azure SQL DW, Oracle, Teradata, MongoDB, Azure CosmosDB, or any ODBC-compliant data source via a generic ODBC driver.

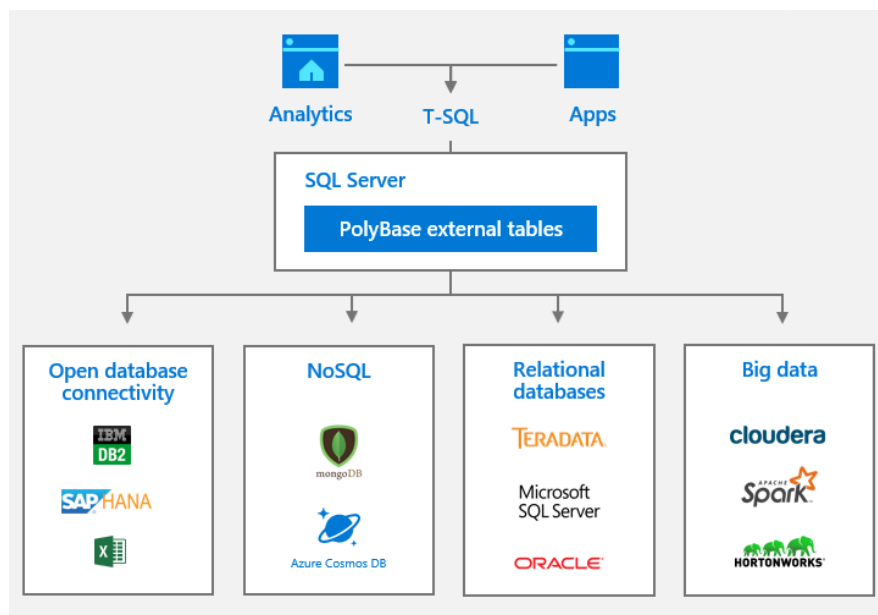


Figure 1: Data sources that can be integrated by PolyBase in SQL Server 2019

In some data query scenarios, data virtualization across multiple data sources is inherently slower than reading the data from a single system. To alleviate the performance impact of virtualizing the data, PolyBase and SQL Server 2019 big data clusters employ a variety of technologies and techniques to parallelize and scale-out compute and cache data.

PolyBase optimizes performance by using push-down computation. Operations such as projections, predicates, aggregates, limits, and homogeneous joins are pushed to the source system to take advantage of the query optimizer in each of the source systems. Only the filtered results of these operations are returned to SQL Server, which improves performance by reducing the amount of data to transfer.

In SQL Server 2019 big data clusters, the SQL Server engine has gained the ability to natively read HDFS files, such as CSV and parquet files, by using SQL Server instances collocated on each of the HDFS data nodes to filter and aggregate data locally in parallel across all of the HDFS data nodes.

Performance of PolyBase queries in SQL Server 2019 big data clusters can be boosted further by distributing the cross-partition aggregation and shuffling of the filtered query results to “compute pools” comprised of multiple SQL Server instances that work together. Big data cluster administrators can quickly create or remove compute pools with command-line tools, the administrator portal, or APIs, giving them the flexibility to elastically scale the compute resources up or down depending on demand. Resource governance policies can be applied to compute pools to control who can use the compute resources. Unlike the scale-out groups of PolyBase in a SQL Server 2017 instance, SQL Server 2019 big data clusters can have any number of compute pools with any number of SQL Server instances in each of them. Scale-out nodes in a SQL Server 2017 PolyBase scale-out group must be individually installed and configured, but compute pools with many SQL Server instances can be provisioned as containers on a Kubernetes cluster with a single command or API call in seconds.

Scalable, shared, and integrated compute and storage

SQL Server, as with other relational databases, has historically been a *scale-up* system. To handle more data or get better performance, a bigger or faster server is required. Although the computing power and storage capacity of today’s servers is staggering, there are practical and physical limits to the size of a single server. SQL Server 2019 big data clusters deeply integrate the SQL Server engine with leading big data technologies in a single package supported by Microsoft that gives SQL Server new abilities to scale-out compute and storage independently.

SQL Server 2019 big data clusters take the form of SQL Server and big data services running as pods of containers in different types of “pools”:

- A **compute pool** is a group of SQL Server pods used for parallel ingestion of data from an external source—such as Oracle, HDFS, or another SQL Server—and for cross-partition aggregation and shuffling of the data as part of a query.
- A **storage pool** is a group of pods containing SQL Server engine, HDFS data node, and Spark containers. This provides the scalable storage tier along with the collocated compute for SQL Server and Spark right next to the data.
- A **data pool** is a group of SQL Server engine pods that is used either to cache data from an external source or to store an incoming stream of append-only data. In either case, the data is partitioned and distributed across all of the SQL Server instances in the pool.
- The **master pool** is a special, singleton pool of SQL Server pods that can be either a singleton SQL Server or a SQL Server deployed as multiple instances in an Always On Availability Group for high availability and read

scale out. This SQL Server instance is where read-write OLTP or dimensional data is stored in a big data cluster.

Each of these pools can be independently scaled up and down to have more or fewer pods as demand changes. In the case of compute pools and data pools, multiple pools can be provisioned each with multiple pods.

Scale-out data marts

When you combine the enhanced PolyBase connectors with SQL Server 2019 big data clusters data pools, data from external data sources can be partitioned and cached across all the SQL Server instances in a data pool, creating a “scale-out data mart”. There can be more than one scale-out data mart in a given data pool, and a data mart can combine data from multiple external data sources and tables, making it easy to integrate and cache combined data sets from multiple external sources.

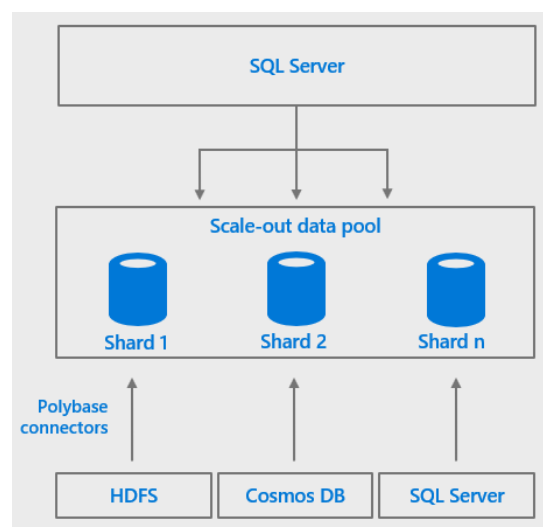


Figure 2: Using a scale-out data pool to cache data from external data sources for better performance

A shared data lake: read and write files in HDFS with SQL Server or Spark

For persistence in a SQL Server big data cluster, SQL Server instances in the data pool can read from and write data to parquet and CSV files in HDFS to use it as a persisted store. SQL Server and Spark can both read and write these files, creating a shared data lake. Applications and analytics querying the data from the SQL Server master instance and Spark jobs can all simultaneously query the data and get the same result. This creates a data lake that can be accessed by many different types of users, tools, and systems without having to export data out of SQL Server to the data lake or rolling up the data from the data lake and putting it in SQL Server.

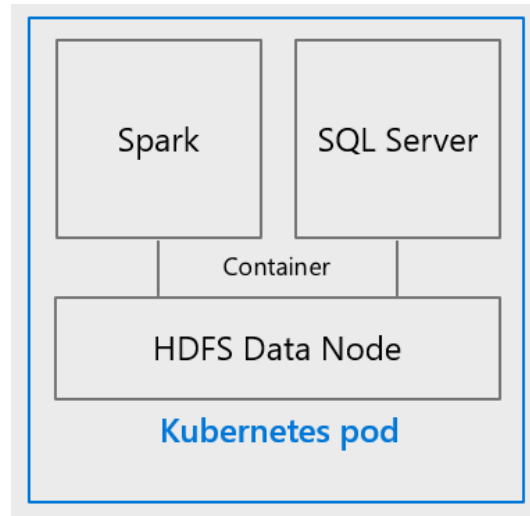


Figure 3: SQL Server and Spark are deployed together with HDFS creating a shared data lake

The highest value enterprise data has long been stored in a relational database like SQL Server, but some interesting new types of data are being primarily stored in HDFS—for example, data from Internet of Things (IoT) devices. The true value of that data is locked up in big data systems and can realistically only be analyzed by big data engineers and data scientists. To get the value out of big data, data scientists typically export high-*value* data out of the relational database and import it into Hadoop so that they can join it with the new high-*volume* data. Without the context of the dimensional high-value data in the enterprise database, high-volume big data doesn't really have much value; big data streams need to be joined to things like customers, accounts, parts, products, marketing campaigns, and other entities.

SQL Server 2019 big data clusters make it easier for big data sets to be joined to the dimensional data typically stored in the enterprise database, enabling people and apps that use SQL Server to query big data more easily. The value of the big data greatly increases when it is not just in the hands of the data scientists and big data engineers but is also included in reports, dashboards, and applications. At the same time, the data scientists can continue to use big data ecosystem tools while also utilizing easy, real-time access to the high-value data in SQL Server because it is all part of one integrated, complete system.

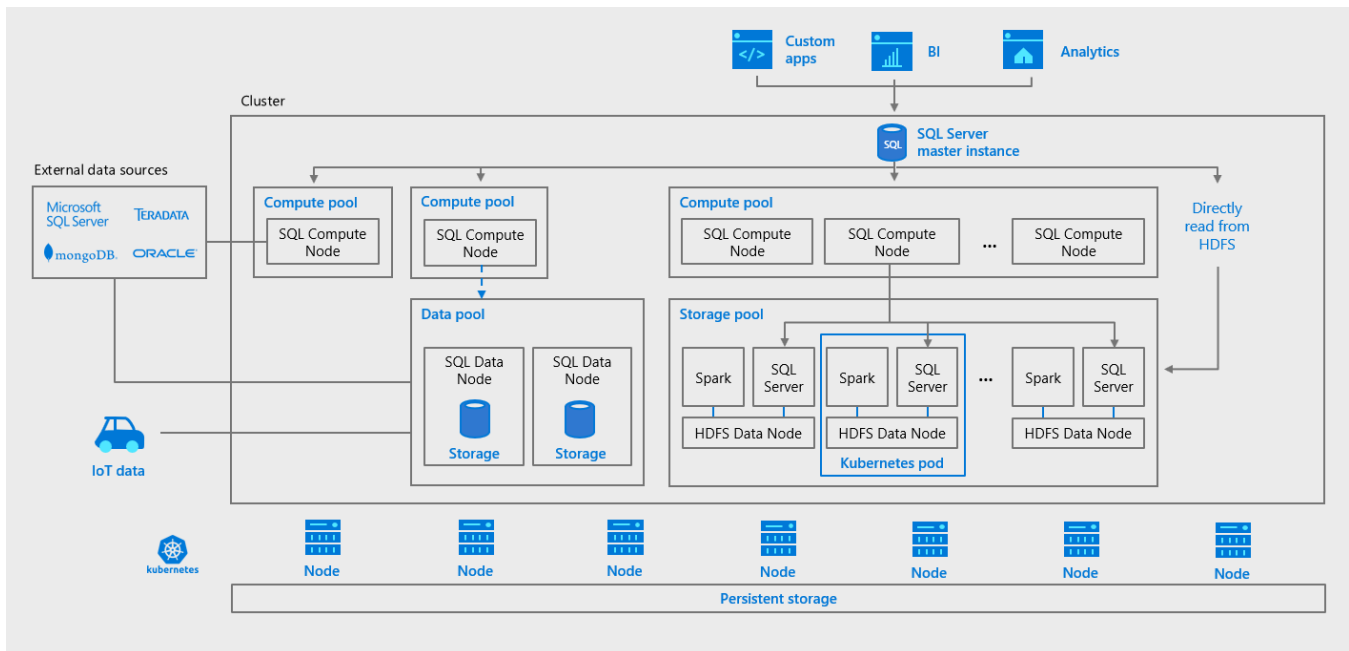


Figure 4: A scalable compute and storage architecture in SQL Server 2019 big data cluster

A complete platform for AI and machine learning

Artificial intelligence and machine learning are frequent headlines in today's technology news. Organizations in every industry are rapidly hiring data scientists and setting up systems to collect data and feed it to machine learning model training. The outcomes from predictive analytics is a function of the "Four V's of Big Data":

- **Volume:** The greater the volume of data processed by a machine learning algorithm, the more accurate the predictions will be.
- **Variety:** The greater the variety of different sources of data, the more accurate the predictions will be.
- **Veracity:** Accurate machine learning is, not surprisingly, dependent on the quality of the data going into the model training.
- **Velocity:** Real-time predictions depend on up-to-date data flowing quickly through the data processing pipelines.

SQL Server big data clusters make machine learning easier and more accurate by improving each of the "Four V's of Big Data":

- **Volume**—increases the volume of data available for AI by capturing data in scalable, inexpensive big data storage in HDFS and by integrating data from multiple sources using the PolyBase connectors.
- **Variety**—increases the variety of data available for AI by integrating multiple data sources through the PolyBase connectors.
- **Velocity**—increases the velocity of data available for AI by using elastic compute and caching to speed up queries.
- **Veracity**—increases the veracity of data available for AI by sharing data without the copying or moving of data, which introduces data latency and data quality issues; SQL Server and Spark can both read and write into the same data files in HDFS.

SQL Server big data clusters provide all the tools and systems to ingest, store, and prepare data for analysis as well as to train the machine learning models, store the models, and operationalize them.

Data can be ingested using Spark Streaming, by inserting data directly to HDFS through the HDFS API, or by inserting data into SQL Server through standard T-SQL insert queries. The data can be stored in files in HDFS, or partitioned and stored in data pools, or stored in the SQL Server master instance in tables, graph, or JSON/XML. Either T-SQL or Spark can be used to prepare data by running batch jobs to transform the data, aggregate it, or perform other data wrangling tasks. Data scientists can choose either to use SQL Server Machine Learning Services in the master instance to run R, Python, or Java model training scripts or to use Spark. In either case, the full library of open-source machine learning libraries, such as TensorFlow or Caffe, can be used to train models. Lastly, once the models are trained, they can be operationalized in the SQL Server master instance using real-time, native scoring via the PREDICT function in a stored procedure in the SQL Server master instance; or you can use batch scoring over the data in HDFS with Spark. Alternatively, using tools provided with the big data cluster, data engineers can easily wrap the model in a REST API and provision the API + model as a container on the big data cluster as a scoring microservice for easy integration into any application.

Importantly, this entire pipeline all happens in the context of a SQL Server big data cluster. The data never leaves the security and compliance boundary to go to an external machine learning server or a data scientist’s laptop. The full power of the hardware underlying the big data cluster is available to process the data, and the compute resources can be elastically scaled up and down as needed.

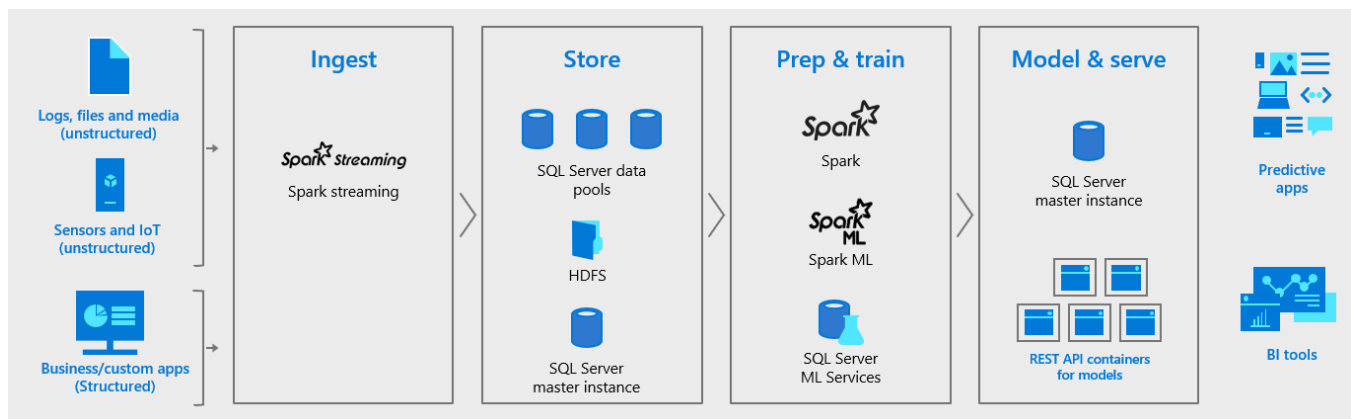


Figure 5: A complete AI platform: SQL Server 2019 big data cluster

Azure Data Studio is an open-source, multi-purpose data management and analytics tool for DBAs, data scientists, and data engineers. New extensions for Azure Data Studio integrate the user experience for working with relational data in SQL Server with big data. The new HDFS browser lets analysts, data scientists, and data engineers easily view the HDFS files and directories in the big data cluster, upload/download files, open them, and delete them if needed. The new built-in notebooks in Azure Data Studio are built on Jupyter, enabling data scientists and engineers to write Python, R, or Scala code with Intellisense and syntax highlighting before submitting the code as Spark jobs and viewing the results inline. Notebooks facilitate collaboration between teammates working on a data analysis project together. Lastly, the External Table Wizard simplifies the process of creating external data sources and tables, including column mappings.

The included [PROSE SDK](#) from Microsoft Research makes it easy to normalize the chaos in your data by analyzing your raw data and producing the *code* that needs to be executed on it to convert it to a normalized and cleaned-

up version of the data. For example, the PROSE-generated code can be used in Spark jobs to prep data for ingestion into SQL Server or to feed into model training or scoring.

Hybrid Deployment and Management Services Built In

Cluster administration portal

SQL Server 2019 big data clusters include management and monitoring out of the box, using open-source components where possible. All the nodes in the cluster run `collectd` for OS and application performance data collection and `fluentd` for log collection. Logs are collected to ElasticSearch monitoring data to InfluxDB.

The big data cluster administration portal exposes monitoring and management views over the entire cluster from Spark to HDFS to the SQL Server master instance. Data collected into ElasticSearch is exposed through Kibana for powerful log analytics and the monitoring data collected into InfluxDB is rendered in the provided dashboards in Grafana.

Standard open-source big data administration tools are also provided out of the box. The YARN UI and Spark UI render detailed views into the health and performance of YARN, Spark, and the jobs running in the big data cluster. SQL Server 2019 also includes some unique extensions to the Spark UI that graphically visualize the job steps including *detailed* resource consumption and timing metrics as well as a capture of the files that were accessed by a given job.

Controller service

The SQL Server 2019 big data cluster controller service coordinates the behavior of the cluster. The controller service is responsible for orchestrating the provisioning and deprovisioning pods/pools through calls to the Kubernetes APIs and managing the allocation of resources within the cluster. The controller service is responsible for many aspects of the cluster lifecycle, including:

- Initializing the compute, data, and storage pools when the cluster is first installed.
- Maintaining and distributing configuration throughout the cluster.

In future releases, the controller will also be responsible for:

- Adding and removing additional compute pools and data pools.
- Adding and removing capacity to the cluster by adding or removing pods from a pool.
- Configuring high availability through SQL Server availability groups.
- Managing software updates —upgrades, downgrades, and rollbacks.
- Configuration management

The controller service is also responsible for security, including cluster authentication, cluster authorization, and rotation of the certificates used to allow nodes to communicate securely within the cluster.

Hybrid deployment and management

Because SQL Server 2019 big data clusters are deployed on Kubernetes, administrators have flexible choices about where to deploy the cluster. Big data clusters can be deployed on-premises on a Kubernetes cluster deployed via a `kubeadm` or on Azure Kubernetes Service (AKS) on Azure Stack. Alternatively, if a hosted cloud is the preferred target, big data clusters can be deployed in VMs with Kubernetes in them or on a managed Kubernetes service,

such as Azure Kubernetes Service (AKS). In the future additional options such as OpenShift and OpenShift on Azure are planned to be supported deployment platforms as well.

Because the entire solution including the controller service and administration portal are deployed as containers on Kubernetes, SQL Server 2019 big data clusters are “self-managed” to some extent, meaning that administrators can reduce the operational overhead of a traditional big data cluster by relying on the automated services for provisioning, high availability, monitoring, and backup that are built into SQL Server 2019 big data clusters. A REST API and the mssqlctl command-line utility open up options for automation.

Conclusion

SQL Server 2019 big data clusters are a compelling new way to utilize SQL Server to bring high-value relational data and high-volume big data together on a unified, scalable data platform. Enterprises can leverage the power of PolyBase to virtualize their data stores, create data lakes, and create scalable data marts in a unified, secure environment without needing to implement slow, costly ETL pipelines. This makes data-driven applications and analysis more responsive and productive. SQL Server 2019 big data clusters provide a complete AI platform to deliver the intelligent applications that help make any organization more successful.

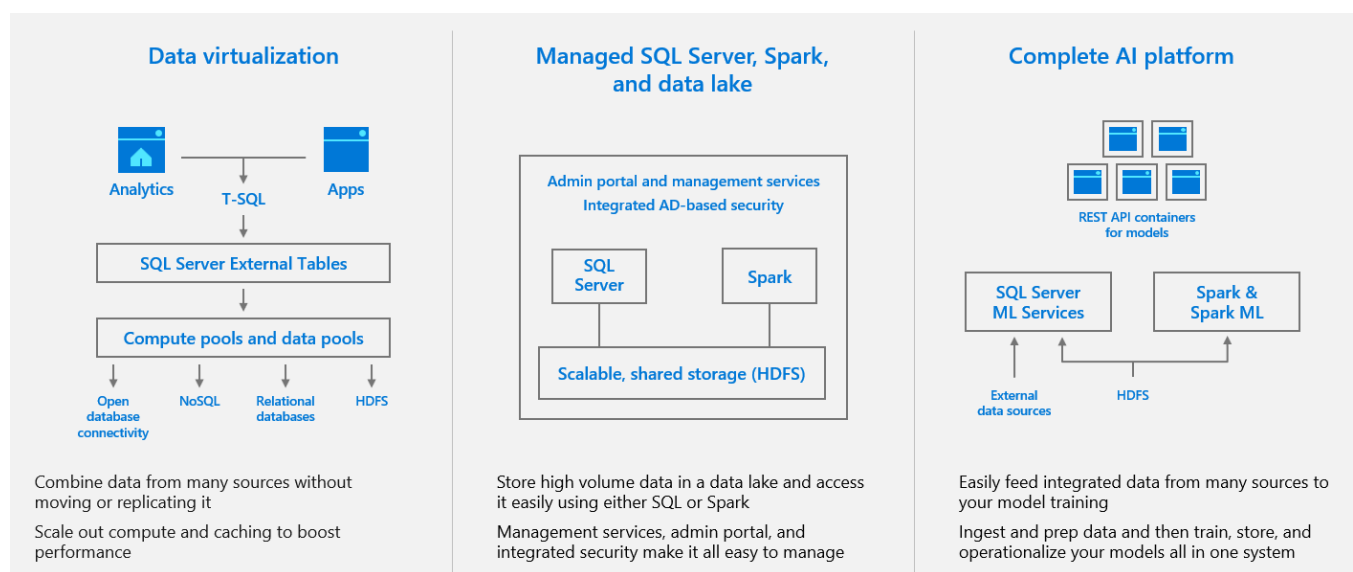


Figure 6: Summary of a SQL Server 2019 big data cluster

Calls to action

For more information about SQL Server 2019, please visit <https://aka.ms/ss19>

For more information about Azure Data Studio, please visit <https://aka.ms/azuredatstudio>

For SQL Server 2019 big data cluster documentation, please visit <https://aka.ms/sqlbigdatacluster>

