Splitting the load

How separating compute from storage can transform the flexibility, scalability and maintainability of big data analytics platforms

Contents

Executive summary

Hadoop is the dominant big data processing system in use today. It is, however, a technology that has been around for 10 years, and the world of big data has changed dramatically over that time.

Hadoop started with a specific focus – a bunch of engineers wanted a way to store and analyze copious amounts of web logs. They knew how to write Java and how to set up infrastructure, and they were hands-on with systems programming. All they really needed was a cost-effective file system (HDFS) and an execution paradigm (MapReduce) — the rest, they could code for themselves. Companies like Google, Facebook and Yahoo built many products and business models using just these two pieces of technology.

Today, however, we're seeing a big shift in the way big data applications are being programmed and deployed in production. Many different user personas, from data scientists and data engineers to business analysts and app developers need access to data. Each of these personas needs to access the data through a different tool and on a different schedule. Moreover, collaboration between these users is vital to making a business endeavor a success.

The traditional method of putting everything in a monolithic storage and compute cluster with a specific programming model does not scale well to handle such access patterns.

In this paper, we look at a different solution to the challenge of big data analytics: splitting the compute and storage environments between Hadoop and Spark clusters on the one hand, and object storage platforms on the other. This has significant benefits in terms of flexibility, scalability, maintainability and cost-efficiency.

IBM Analytics Engine supports this new type of architecture by bringing the two key capabilities, compute-as-a-service and object storage together in a single coherent solution. As well as providing Hadoop and Spark clusters on demand in minutes, the solution also acts as a high-performance big data processing engine for IBM® Watson Studio, empowering users of all types to deliver rapid insights to a business.

Challenges of Hadoop design

Hadoop was designed around using commodity x86 hardware with locally attached disks, typically inexpensive and high-capacity spinning disks.

The main components are a storage subsystem (HDFS), a cluster scheduler (YARN), compute engines (typically MapReduce and Spark), tools that generate code or jobs (for example, Hive and Pig) and utilities (for example, Oozie).

Clusters are divided into two types of nodes:

- **Management nodes,** which run the management plane, including the name node, yarn master, management UIs, and so on. Clusters typically have a small number (1 – 4+) of management nodes, with the total number depending on the overall size of the cluster and the level of high availability required.
- **Data nodes,** which provide the data processing and storage layers, and comprise a large part of the cluster. Typically, there will be a relatively large number (8 – 12) of spinning disks physically attached to each node.

When jobs are scheduled, YARN is aware of which data is located on which disk, so it can assign appropriate compute jobs to the nodes where the relevant data is situated. This system works very well and has been proven to scale effectively across thousands of nodes.

Hadoop clusters as described above are typically very long-lived, often with a lifetime measured in years. Clusters are often upgraded in-place due to their size and the hardware expense associated with running a secondary cluster in parallel.

Limitations of traditional Hadoop clusters

Traditional Hadoop clusters have several weaknesses that make it difficult to manage at scale:

1. **Compute is bound tightly to storage.** If your cluster runs many compute-intensive workloads on small data sets, you will need to purchase more data nodes — and each node will need to include disk storage, even if those disks aren't necessary for the size of your data.
2. **I/O-heavy clusters get very expensive.** If your cluster needs to work with very large data sets, you will need to purchase more data nodes to hold the data. Each node will need to include expensive RAM and CPU resources, even if the cluster's total compute needs are already met by existing nodes. Petabyte-scale clusters often have a lot of cold data but are expensive from a gigabyte-per-month perspective.
3. **Lack of elasticity.** Adding and removing data nodes based on real-time demand results in significant HDFS rebalancing work, diverting resources away from usable compute work.
4. **Security is costly.** Identity management with Apache Ranger cannot be easily integrated with other cloud security frameworks, increasing management overhead for access control.
5. **Single version lock-in.** Although it is technically possible to run multiple different Hadoop distributions or software versions on a single cluster, it is a lot of work, requires significant skill, and is error-prone. In addition, the dependency stack in Hadoop can be quite deep, so it is non-trivial to select compatible versions of software that is safe to install. As a result, most clusters tend to be locked in a single version of a single vendor's Hadoop distribution.
6. **Tuning for utilization.** Despite management APIs and UIs being readily available, managing Hadoop workloads is not simple. Tuning a cluster to drive maximum utilization requires time, effort and high technical skill.
7. **Updates are complex.** Clusters require a very diligent patching process, which can get challenging when the cluster is large and is constantly running jobs. Taking nodes offline to patch them has knock-on effects on performance. It often takes months to coordinate, test and execute an upgrade, because there can be dozens or hundreds of apps that depend on the shared infrastructure, all with different requirements and dependencies.
8. **Upgrades are expensive.** The risks associated with upgrading a fully distributed system that holds terabytes or petabytes of data are significant. Cluster upgrades have historically not been a smooth process, often requiring full-time migration professionals onsite to lead or assist with these delicate operations.

How cloud has changed the game

Over the past ten years, advances in cloud computing have changed one of the most fundamental assumptions made by traditional Hadoop architectures. Cloud providers have improved network interconnect speeds to the point that speeds of 10 Gbps or more are normal, and 40 Gbps is increasingly common.

Today's networks are so fast that most public clouds no longer need to provide locally attached storage for compute. Data and compute are no longer physically co-located. Although it may look like you have local disks from the perspective of the operating system, the data is a network hop, held in object or block storage systems in a separate storage area network (SAN).

There are still a few exceptions to this rule. For example, some vendors offer special big data cloud servers with local disk storage, but they're typically designed for very specific use cases, and they come at a premium price compared to commodity cloud servers.

If data and compute no longer need to be co-located to provide adequate performance, then we can question the whole premise that each node on Hadoop clusters needs to handle both storage and compute. In fact, several other assumptions can be challenged too:

- Why not allow multiple clusters to read the same data set at once?
- Why not dynamically allocate compute for the duration of a single job?
- Why manage identity within clusters instead of using standard cloud-native identity and access management?
- Why be limited to a single version of a framework (like Apache Spark) for all jobs?

Introducing IBM Analytics Engine

IBM Analytics Engine provides an answer to these questions by providing an architecture for Hadoop clusters that completely decouples the compute and storage tiers. Instead of a permanent cluster formed of dual-purpose data nodes, it allows users to store data in an object storage layer such as IBM Cloud Object Storage and spins up clusters of compute nodes when users need them.

With this model, compute and storage can be scaled and paid for independently. Compute costs can even drop to zero when no analytics jobs need to be run.

IBM Analytics Engine allows users to configure and instantiate a Hortonworks Data Platform cluster in 30 to 50 minutes. After the cluster is created, users can submit and run their jobs, and then delete the cluster again when those jobs are finished. This flexible approach can help businesses deal more effectively with different Hadoop workload types.

In the IBM Analytics Engine model, each cluster has a single Hadoop user. Access control is performed at the cloud level rather than within the cluster, using cloud-wide access control lists (ACLs). Thus, data remains secure, without being locked into a single set of tools on a single compute cluster.

Overcoming the limitations

Let's revisit the limitations list and explore how IBM Analytics Engine addresses each of them:

1. **Compute is no longer bound tightly to storage.** You can spin up compute-only clusters on demand.
2. **I/O-heavy clusters are more cost-effective.** You can provision more Cloud Object Storage (or other data stores) on demand with no extra costs for compute cycles you won't use.
3. **Clusters are more elastic.** Adding and removing data nodes based on live demand is possible via REST APIs, and overhead is low because very little to no data is stored in the cluster. Moreover, since clusters are typically created for the duration of a specific job, it is less of a problem if any individual cluster is poorly sized: it can simply be deleted and recreated with a more appropriate configuration.
4. **Security is more cost-effective.** Security can now be handled at several levels; by using the IBM Cloud IAM service, which controls which users are authorized to spin up a cluster; the IBM Cloud Object Storage ACLs, which protect individual buckets of storage; and Watson Knowledge Catalog, which can enforce governance policies around the data assets themselves. This multilayered approach significantly simplifies the individual cluster security implementation, while enabling access management at a highly granular level.
5. **Vendor lock-in is avoided.** You can spin up your clusters to meet the needs of your job rather than forcing the jobs to conform to a single software package version. This means that if you need to run multiple different versions of software packages in different clusters, this is not an issue.
6. **Tuning for utilization is easier.** As jobs run on clusters that are sized for the job, utilization can be better tracked and managed. There is no need to determine a one-size-fits-all cluster approach.
7. **Updates are simpler.** With on-demand creation of temporary clusters, there is no need to interrupt ongoing jobs or schedule specific downtime for security patching; patches are automatically applied whenever a new cluster is created. You can automatically spin up new clusters with the right version of the software on a granularity you choose.
8. **Upgrades are unnecessary.** Because the data is not stored in the cluster itself, clusters never actually need to be upgraded: the existing cluster can simply be discarded, and a new one created with the new version of the software.

Exploring the IBM Analytics Engine architecture

1. Connecting to object storage with Stocator

In theory, connecting Hadoop to object storage systems has been possible for some years. Hadoop offers various object storage connectors that allow it to treat common object storage systems as though they are file systems.

However, this approach is highly inefficient, requiring hundreds of application programming interface (API) calls to complete even the simplest write operations. The algorithms that Hadoop uses to persist distributed data sets are not optimized for object stores, and the need to support file system operations comes at a significant cost to performance.

A team from IBM Research realized the huge potential of solving this challenge and began working on Stocator, a next-generation open source connector. Stocator leverages the unique semantics of object storage to greatly reduce the number of API calls. For write-intensive workloads, Apache Spark running with Stocator can perform 32 times fewer operations on object storage than the legacy Hadoop connectors, and half the operations for read-only workloads[1]. This dramatic decrease in operations on object storage reduces costs for the user as well as increasing performance. Thus, Stocator unlocks the full potential of object storage as a persistence layer for Hadoop and Spark clusters.

2. Leveraging container technology

IBM Analytics Engine clusters are built on Hortonworks Data Platform, an ODPi-compliant, 100-percent open-source Hadoop and Spark distribution that maximizes versatility and minimizes lock-in.

Instead of running on bare-metal servers, IBM Analytics Engine harnesses modern virtualization and container technology. The Hadoop or Spark distribution that runs on each node of the cluster is encapsulated in a Docker container, which can be initialized or shut down in the order of minutes.

3. Focusing on usability

Users can interact with IBM Analytics Engine via an intuitive web interface that makes it easy to spin up, scale and delete clusters, monitor and manage them, and change their settings by uploading configuration scripts. The solution also provides both a REST API and a command line interface for programmatic access, helping developers build apps that integrate real-time Spark analytics.

IBM Analytics Engine is also part of IBM Watson Studio, which means it integrates seamlessly with other services such as:

- Watson Knowledge Catalog, for metadata management and cataloging
- IBM Data Refinery, for integrating, cleaning, enriching and refining large data sets
- IBM Watson Machine Learning, for training and testing predictive models and neural networks

Instead of each data science team using different tools, Watson Studio provides a set of best-of-breed solutions to support every step in the data science value chain.

Moreover, as each of the Watson Studio services is built on a shared architecture of microservices, integration between the different components is seamless and consistent. This eliminates the need to write custom scripts to link tools together and makes the flow of data more reliable and easier to maintain.

Use cases for IBM Analytics Engine

IBM Analytics Engine is designed as a multi-purpose big data processing engine that can be leveraged by other solutions and tools in many different use cases. Here are just a few examples of how IBM Analytics Engine can help solve real-world big data challenges.

1. Optimizing processing for different types of workloads

Many companies have a "batch at night, interactive during the day" usage pattern in their Hadoop clusters. There's always a balance between the needs of these two different types of workload, and it's difficult to achieve a perfect balance within a single permanent cluster.

IBM Analytics Engine provides a more versatile approach to cluster management that helps optimize the use of available resources to meet the needs of these workloads.

For example, it is possible to spin up a fresh cluster at the beginning of the nightly batch runs, perform the work there, and when finished, to delete the cluster. An administrator can create the cluster to meet the required service levels, using scripts to customize the cluster using the right software versions. An administrator can even spin up multiple independent batch clusters if necessary, depending on how the runs need to be isolated.

In the morning, a new cluster (or clusters) can be spun up for interactive use, using a different set of configuration scripts to install the right software components to meet the needs of the day time users.

Instead of fighting with Hadoop identity management to allocate different parts of a permanent cluster to different workload profiles and user types, users can have their own clusters, and access to each cluster can be controlled using the organization's standard cloud identity and access management framework.

Additionally, IBM Cloud enables fine-grained billing at a per-instance level which allows spreading costs fairly between the different teams using the resources. This is difficult to achieve in traditional monolithic Hadoop clusters.

2. Simplifying data governance

Traditional data lakes built on Hadoop often suffer from data governance issues. To restrict access to sensitive data, the Hadoop cluster needs to be segregated into different zones for different sets of users, which requires a large amount of configuration and maintenance.

This problem can be solved by combining IBM Analytics Engine with IBM Cloud Object Storage and IBM Knowledge Catalog. Instead of building the data store as a single, persistent Hadoop cluster that must handle all the data, governance configurations and computation work, you separate these three concerns.

All data assets are stored permanently in IBM Cloud Object Storage, and users access them via IBM Knowledge Catalog. The catalog contains metadata about each of the data assets, such as a profile of the contents and the data governance policies that apply. This helps users find relevant assets and understand which data sets they can use for which purposes.

After a user has retrieved the data set they need, they can then use IBM Analytics Engine to spin up a temporary cluster and load the data into it for analysis. Once the analysis is complete, the results can be saved back into object storage and be added to the catalog for other users to find. Once the cluster has served its purpose, it can be deleted.

Throughout the process, governance rules are applied automatically, and there is no need for complex cluster configuration—saving time for data stewards and IT teams and making it easier for users to find the data they need.

3. Reducing the cost of disaster recovery

Disaster recovery has been a significant pain-point for traditional Hadoop-based data stores. In most cases, the only viable solution has been to set up a secondary cluster and implement expensive proprietary software to keep it in sync with the main cluster.

The combination of IBM Analytics Engine and IBM Cloud Object Storage makes disaster recovery both more convenient and less expensive. IBM Cloud Object Storage provides organizations with an innovative, security-rich and cost-effective way to store and manage data without creating multiple copies. IBM SecureSlice technology distributes data slices across storage systems for security and protection. This unique approach combines encryption, erasure coding and information dispersal of data to multiple locations for protection without complex or expensive copies. For even greater levels of protection, users can also choose to automatically replicate data across IBM Cloud regions.

The service is continuously available; it can tolerate even catastrophic regional outages without down times or the need for customer intervention. If, for example, a regional disaster takes out one storage system, then all a user needs to do is pass their configuration file to IBM Analytics Engine and spin up a new cluster, which can continue accessing data stored in the unaffected regions.

This simple architecture significantly reduces the amount of effort required to set up a disaster recovery environment and makes it quicker and easier to recover if a disaster occurs.

4. Streamlining data science and machine learning workflows

Data science workflows depend on the ability to explore large data sets and train and test predictive models and neural networks efficiently. However currently, data scientists spend much of their time writing scripts to move data between environments or configuring and maintaining Hadoop and Spark clusters.

Tools such as IBM Watson Studio and IBM Watson Machine Learning leverage IBM Analytics Engine to make these processes more seamless and automated. For example, a data scientist who wants to explore a large data set can simply open a notebook in IBM Watson Studio, associate an IBM Analytics Engine Spark cluster with it, and start looking for insights straight away.

Benefits of migrating to IBM Analytics Engine

For users who are currently using other cloud-based Spark or Hadoop services, such as IBM BigInsights® on Cloud or IBM Analytics for Apache Spark, the migration to IBM Analytics Engine can unlock significant benefits. For example:

- The containerized architecture enables much faster instantiation of new clusters and on-demand cluster scalability, without needing to wait for assistance from the support team.
- The REST API and command line interface provide more powerful options for developers to integrate clusters into their apps.
- Authentication is handled via the same identity and access management service as other Watson Studio services, providing a single, integrated, security-rich environment.
- Patches and upgrades can be incorporated into new clusters whenever it's convenient for the user, without disrupting running workloads.
- Configurations can easily be replicated across multiple clusters, instead of having to install third-party libraries and packages individually for each cluster.

Conclusion

IBM Analytics Engine is designed to resolve some of the biggest problems that organizations have encountered during their move to a data lake strategy.

By separating compute from storage, the architecture of IBM Analytics Engine allows companies to scale their big data landscape in line with business requirements and avoid investing in infrastructure that they don't need. The use of object storage helps to protect data and improve availability. Meanwhile, the fact that clusters can be spun up and spun down dynamically as needed helps to simplify maintenance and upgrades.

These architectural decisions also result in a more streamlined user experience. The solution makes it easier for data scientists, data engineers and developers to develop and deploy analytics applications, empowering them to spin up Hadoop and Spark clusters within minutes and manage them from a single point of control. Critically, users no longer need to worry about the underlying infrastructure, so they can focus on their core role of delivering insights to the business.

By taking advantage of IBM Analytics Engine's integration with other IBM Watson Studio services, you can open up new possibilities for end-to-end data science workflows, improving collaboration, increasing productivity and delivering actionable insights to your business faster than ever before.

About the Author

Rohan Vaidyanathan
Offering Manager, IBM Analytics Engine
IBM Watson Data Platform
rvaidya@us.ibm.com

For more information

To learn more about IBM Analytics Engine, contact your IBM representative or IBM Business Partner, or visit: **ibm.com**/analytics/us/en/watson-data-platform/analytics-engine/

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
January 2018

[1] As compared with the S3a object storage connector of Hadoop 2.7.3 run with its default parameters

ASW12451-USEN-00