Performance Comparison between MinIO and Amazon S3 for Apache Spark

JULY 2019

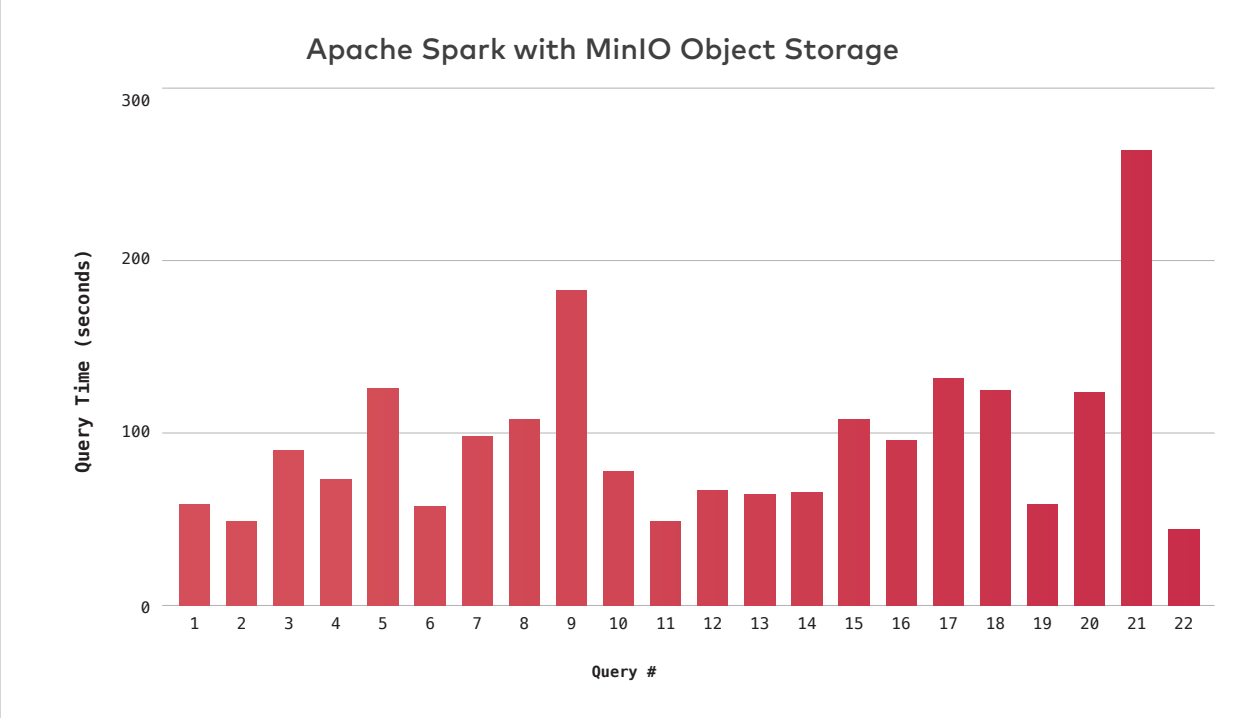
Performance comparison between MinIO and Amazon S3 for Apache Spark

MinIO is a high-performance, object storage server designed for AI and ML workloads. It is fully compatible with the Amazon S3 API.

Machine learning, big-data analytics, and other AI workloads have traditionally utilized the map-reduce model of computing where data is local to the compute jobs. Modern computing environments have adopted a cloud-native architecture where storage and compute are disaggregated. This enables computing to become stateless, elastic, and scalable independent of storage. Object storage has become the de-facto standard for this architecture.

Apache Spark is a unified analytics engine for big-data processing, with built-in modules for streaming, SQL, machine learning and graph processing. Enterprises such as HP, Shell, and Cisco utilize Spark to perform large scale analytics. At this scale, the performance of underlying storage directly is critically important.

This document describes the benchmarking tests and results measuring the performance of Apache Spark with MinIO Object Storage and Amazon S3.

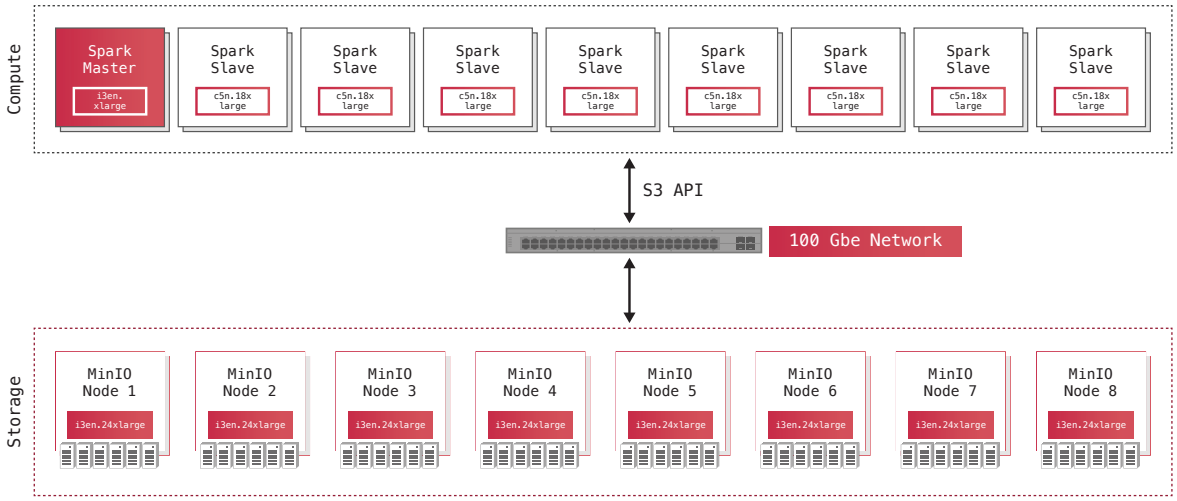


1. Benchmark Environment

1.1 Hardware

For the purpose of this benchmark, MinIO utilized AWS bare-metal, storage-optimized instances with local NVMe drives and 100 GbE networking. The nodes running Spark (c5n.18xlarge) have the highest available bandwidth to the S3 service.

Instance	# Nodes	AWS Instance type	CPU	MEM	Storage	Network
Spark Master	1	i3en.xlarge	4	32 GB	1 x 2500 GB	Upto 25 Gbps
Spark Slave	8	c5n.18xlarge	72	192 GB	EBS	100 Gbps
MinIO Server	8	i3en.24xlarge	96	768 GB	8 x 7500 GB	100 Gbps



1.2 Software

Property	Value
Apache Spark	Apache Spark 2.4.1 with Hadoop 3.1.2
MinIO	Minio-RELEASE.2019-06-15T23-07-18Z
Benchmark	TPC-H™ Benchmark Scaling Factor: 1000
Server OS	Ubuntu 18.04

TPC-H™ benchmark

TPC Benchmark™ H is comprised of a set of business queries designed to exercise system functionalities in a manner representative of complex business analysis applications. These queries portray the activity of a wholesale supplier and add necessary context to the components of the benchmark.

Dataset

TPC Benchmark™ H, provides its own dataset comprising of eight tables representing a complex business environment. The tables are interrelated to each other, facilitating complex queries across multiple tables.

The size of the dataset is variable and chosen based on the underlying storage system. The size of the dataset is determined based on a scaling factor. A scaling factor of one leads to a dataset approximately 1GB in size, scaling factor 100 generates a dataset approximately 100GB in size and so on. The scaling factor is plugged into a dataset generation tool to generate the data.

This benchmark used scaling factor 1000. A summary detailing the dataset is presented below:

Table	# Records	# Records (SF: 1000)
Customer	150,000 * SF	150,000,000
Orders	1,500,000 * SF	1,500,000,000
Lineitem	6,000,000 * SF	6,000,000,000
Supplier	10,000 * SF	10,000,000
Part	200,000 * SF	200,000,000
PartSupp	800,000 * SF	800,000,000
Region	5	5
Nation	25	25
Total		8.66 Billion

The data was formatted in ORC (Optimized Row Columnar) format, and stored in a MinIO bucket. Converting to this format automatically compresses the data, which shrunk the data size to 273 GB.

Apache Spark Performance Tuning

Apache Spark utilizes the Hadoop s3a connector to connect with MinIO and Amazon S3. The connector was tuned with the following configuration:

Parameter	Value
`spark.hadoop.fs.s3a.connection.maximum`	`4000`

The following parameters were configured while each query was submitted to Spark.

Parameter	Value
`num-executor-cores`	`2`
`driver-memory`	`24G`
`executor-memory`	`2G`

In addition to the above optimizations, the executors were configured to run only on the higher capacity worker nodes and never run on the master node.

2. Benchmark Results

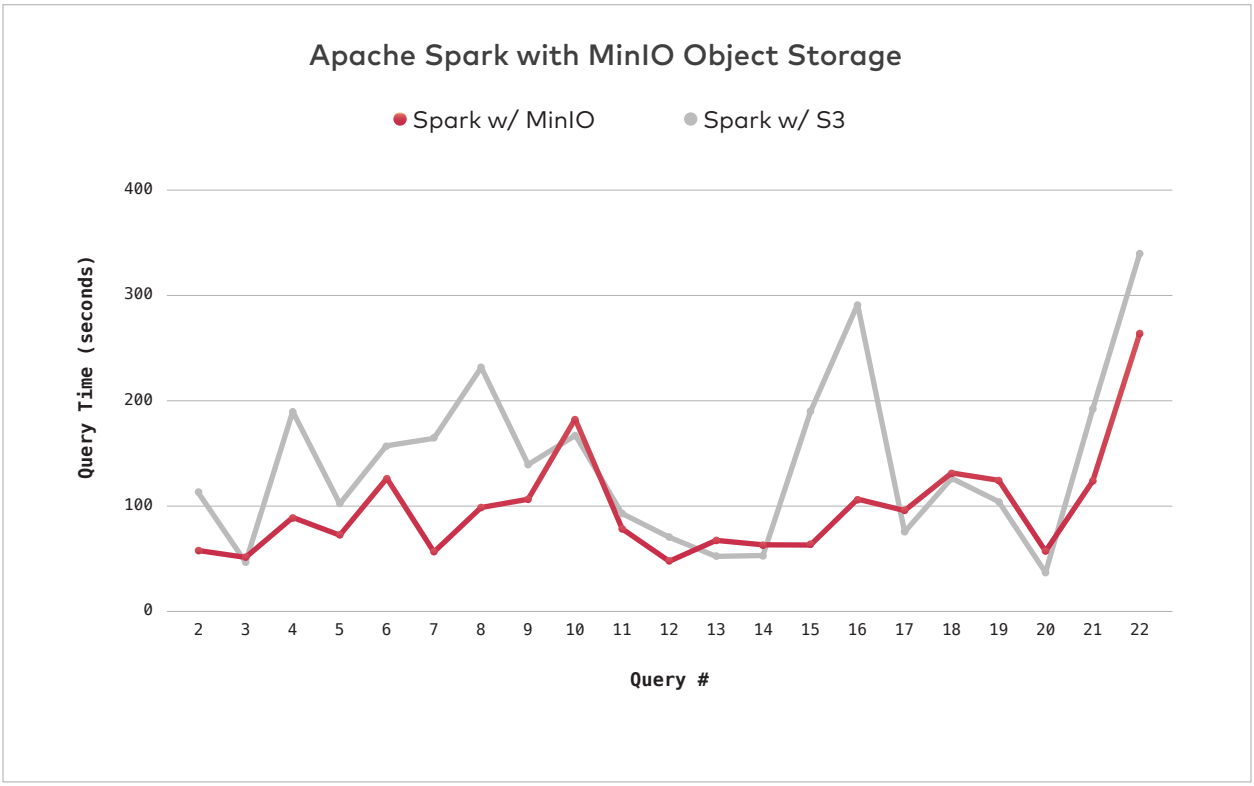
The time taken for each of the 22 TPC-H™ queries is presented below:

Node #	Query Execution Time (seconds)
	Spark w/ MinIO
1	58.14
2	48.37
3	89.25
4	72.72
5	125.28
6	57.00
7	98.47
8	106.97
9	182.19
10	77.49
11	48.64
12	66.59
13	63.96
14	64.89
15	106.98
16	95.46
17	131.54
18	124.73
19	57.93
20	122.91
21	264.01
22	44.36
Average	95.81

The values presented above serve to provide a reference point that will be used to benchmark future versions of MinIO and other applications serving similar use cases.

3. Comparing MinIO to Amazon S3

The same benchmark tests were run against data stored in Amazon S3 using the same hardware for Apache Spark. It should be noted that MinIO is strictly consistent, whereas Amazon S3 in only eventually consistent. The performance was largely the same with some queries slower than MinIO and others faster - and overall in favor of MinIO. A graph summarizing the query times comparing MinIO and S3 for Apache Spark workloads is presented below:



4. Results

The results show that the performance difference between running Apache Spark backed by S3, as compared to Apache Spark backed by MinIO is in favor of MinIO - making MinIO an attractive alternative for large scale, high performance, data-intensive workloads in private cloud environments.