Top 5 Reasons for Choosing S3 over HDFS

Cost, elasticity, availability, durability, performance, and data integrity

May 31, 2017 (https://databricks.com/blog/2017/05/31/top-5-reasons-for-choosing-s3-over-hdfs.html) | by Reynold Xin (https://databricks.com/blog/author/rxin), Josh Rosen (https://databricks.com/blog/author/josh-rosen) and Kyle Pistor (https://databricks.com/blog/author/kyle-pistor) in **COMPANY BLOG (HTTPS://DATABRICKS.COM/BLOG/CATEGORY/COMPANY)**

At Databricks, our engineers guide thousands of organizations to define their big data and cloud strategies. When migrating big data workloads to the cloud, one of the most commonly asked questions is how to evaluate HDFS versus the storage systems provided by cloud providers, such as Amazon's S3 (https://aws.amazon.com/s3/), Microsoft's Azure Blob Storage (https://azure.microsoft.com/en-us/services/storage/blobs/), and Google's Cloud Storage (https://cloud.google.com/storage/). In this blog post, we share our thoughts on why cloud storage is the optimal choice for data storage.

In this discussion, we use Amazon S3 as an example, but the conclusions generalize to other cloud platforms. We compare S3 and HDFS along the following dimensions:

1. Cost
2. Elasticity
3. SLA (availability and durability)
4. Performance per dollar
5. Transactional writes and data integrity

Cost

Let's consider the total cost of storage, which is a combination of storage cost and human cost (to maintain them).

First, let's estimate the cost of storing 1 terabyte of data per month.

As of May 2017, S3's standard storage price for the first 1TB of data is $23/month. Note that depending on your usage pattern, S3 listing and file transfer might cost money. On the other hand, cold data using infrequent-access storage would cost only half, at $12.5/month. For the purpose of this discussion, let's use $23/month to approximate the cost. S3 does not come with compute capacity but it does give you the freedom to leverage ephemeral clusters and to select instance types best suited for a workload (e.g., compute intensive), rather than simply for what is the best from a storage perspective.

For HDFS, the most cost-efficient storage instances on EC2 is the d2 family. To be generous and work out the best case for HDFS, we use the following assumptions that are virtually impossible to achieve in practice:

- A crystal ball into the future to perfectly predict the storage requirements three years in advance, so we can use the maximum discount using 3-year reserved instances.
- Workloads are stable with a peak-to-trough ratio of 1.0. This means our storage system does not need to be elastic at all.
- Storage utilization is at 70%, and standard HDFS replication factor set at 3.

With the above assumptions, using d2.8xl instance types ($5.52/hr with 71% discount, 48TB HDD), it costs *5.52 x 0.29 x 24 x 30 / 48 x 3 / 0.7 = $103/month* for 1TB of data. (Note that with reserved instances, it is possible to achieve lower price on the d2 family.)

So in terms of storage cost alone, **S3 is 5X cheaper than HDFS.**

Based on our experience managing petabytes of data, S3's human cost is virtually zero, whereas it usually takes a team of Hadoop engineers or vendor support to maintain HDFS. Once we factor in human cost, **S3 is 10X cheaper than HDFS** clusters on EC2 with comparable capacity.

Elasticity

Capacity planning is tough to get right, and very few organizations can accurately estimate their resource requirements upfront. In the on-premise world, this leads to either massive pain in the post-hoc provisioning of more resources or huge waste due to low utilization from over-provisioning upfront.

One of the nicest benefits of S3, or cloud storage in general, is its elasticity and pay-as-you-go pricing model: you are only charged what you put in, and if you need to put more data in, just dump them there. Under the hood, the cloud provider automatically provisions resources on demand.

Simply put, **S3 is elastic, HDFS is not.**

SLA (Availability and Durability)

Based on our experience, S3's availability has been fantastic. Only twice in the last six years have we experienced S3 downtime and we have never experienced data loss from S3.

Amazon claims 99.999999999% durability and 99.99% availability. Note that this is higher than the vast majority of organizations' in-house services. The official SLA from Amazon can be found here: Service Level Agreement – Amazon Simple Storage Service (S3) (https://aws.amazon.com/s3/sla/).

For HDFS, in contrast, it is difficult to estimate availability and durability. One could theoretically compute the two SLA attributes based on EC2's mean time between failures (MTTF), plus upgrade and maintenance downtimes. In reality, those are difficult to quantify. Our understanding working with customers is that the majority of Hadoop clusters (https://databricks.com/glossary/hadoop-cluster) have availability lower than 99.9%, i.e. at least 9 hours of downtime per year.

With cross-AZ replication that automatically replicates across different data centers, **S3's availability and durability is far superior to HDFS'.**

Performance per Dollar

The main problem with S3 is that the consumers no longer have data locality and all reads need to transfer data across the network, and S3 performance tuning itself is a black box.

When using HDFS and getting perfect data locality, it is possible to get ~3GB/node local read throughput on some of the instance types (e.g. i2.8xl, roughly 90MB/s per core). DBIO (https://databricks.com/blog/2017/05/24/databricks-runtime-3-0-beta-delivers-enterprise-grade-apache-spark.html), our cloud I/O optimization module, provides optimized connectors to S3 and can sustain ~600MB/s read throughput on i2.8xl (roughly 20MB/s per core).

That is to say, on a per node basis, HDFS can yield 6X higher read throughput than S3. Thus, **given that the S3 is 10x cheaper than HDFS, we find that S3 is almost 2x better compared to HDFS on performance per dollar.**

However, a big benefit with S3 is we can separate storage from compute, and as a result, we can just launch a larger cluster for a smaller period of time to increase throughput, up to allowable physical limits. This separation of compute and storage also allow for different Spark applications (https://databricks.com/glossary/what-are-spark-applications) (such as a data engineering ETL job and an ad-hoc data science model training cluster) to run on their own clusters, preventing concurrency issues that affect multi-user fixed-sized Hadoop clusters. This separation (and the flexible accommodation of disparate workloads) not only lowers cost but also improves the user experience.

One advantage HDFS has over S3 is metadata performance: it is relatively fast to list thousands of files against HDFS namenode but can take a long time for S3. However, the scalable partition handling feature (https://databricks.com/blog/2016/12/15/scalable-partition-handling-for-cloud-native-architecture-in-apache-spark-2-1.html) we implemented in Apache Spark 2.1 mitigates this issue with metadata performance in S3.

Stay tuned for announcements in the near future that completely eliminates this issue with DBIO.

Transactional Writes and Data Integrity

Most of the big data systems (e.g., Spark, Hive) rely on HDFS' atomic rename feature to support atomic writes: that is, the output of a job is observed by the readers in an "all or nothing" fashion. This is important for data integrity because when a job fails, no partial data should be written out to corrupt the dataset.

S3's lack of atomic directory renames has been a critical problem for guaranteeing data integrity. This has led to complicated application logic to guarantee data integrity, e.g. never append to an existing partition of data.

Today, we are happy to announce the support for transactional writes in our DBIO artifact, which features high-performance connectors to S3 (and in the future other cloud storage systems) with transactional write support for data integrity. See this blog post for more information (https://databricks.com/blog/2017/05/31/transactional-writes-cloud-storage.html).

Other Operational Concerns

So far, we have discussed durability, performance, and cost considerations, but there are several other areas where systems like S3 have lower operational costs and greater ease-of-use than HDFS:

- **Encryption, access control, and auditing:** S3 supports multiple types of encryption (https://aws.amazon.com/s3/details/#encryption), with both AWS- and customer-managed keys, and has easy-to-configure audit logging and access control capabilities. These features make it easy to meet regulatory compliance needs, such as PCI or HIPAA compliance (https://databricks.com/company/newsroom/press-releases/databricks-announces-hipaa-compliance-apache-spark-based-platform-achieves-aws-public-sector-partner-status).
- **Backups and disaster recovery:** S3's opt-in versioning (https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html) feature automatically maintains backups of modified or deleted files, making it easy to recover from accidental data deletion. Cross-region replication (https://docs.aws.amazon.com/AmazonS3/latest/dev/crr.html) can be used to enhance S3's already strong availability guarantees in order to withstand the complete outage of an AWS region.
- **Data lifecycle management:** S3 can be configured to automatically migrate objects to cold storage after a configurable time period. In many organizations, data is read

frequently when it is new and is read significantly less often over time. S3's lifecycle management policies can automatically perform migration of old objects to Infrequent Access storage in order to save on cost, or to Glacier to achieve even larger cost savings; the latter is useful for organizations where regulatory compliance mandates long-term storage of data.

Supporting these additional requirements on HDFS requires even more work on the part of system administrators and further increases operational cost and complexity.

Conclusion

In this blog post we used S3 as the example to compare cloud storage vs HDFS:

	S3	**HDFS**	**S3 vs HDFS**
Elasticity	Yes	No	S3 is more elastic
Cost/TB/month	$23	$206	10X
Availability	99.99%	99.9% (estimated)	10X
Durability	99.999999999%	99.9999% (estimated)	10X+
Transactional writes	Yes with DBIO	Yes	Comparable

To summarize, S3 and cloud storage provide elasticity, with an order of magnitude better availability and durability and 2X better performance, at 10X lower cost than traditional HDFS data storage clusters.

Hadoop and HDFS commoditized big data storage by making it cheap to store and distribute a large amount of data. However, in a cloud native architecture, the benefit of HDFS is minimal and not worth the operational complexity. That is why many organizations do not operate HDFS in the cloud, but instead use S3 as the storage backend.

With Databricks' DBIO, our customers can sit back and enjoy the merits of performant connectors to cloud storage without sacrificing data integrity.

TRY DATABRICKS FOR FREE.

GET STARTED TODAY (HTTPS://DATABRICKS.COM/TRY-DATABRICKS)

SEE ALL COMPANY BLOG POSTS (HTTPS://DATABRICKS.COM/BLOG/CATEGORY/COMPANY)