ArchitectureAndPoc OfManagedDataHub

A Case Study

BigR.io

EMPOWERING DATA

ArchitectureAndPoc OfManagedDataHub

A Case Study

The Situation

A rapidly growing niche analytics firm has developed cutting-edge models helping retailers within their industry make improved product recommendations to customers. These recommendations not only match appropriate products to those customers, but also help configure the product to the customers' specific personalized needs. This requires no direct input from customers, and is inferred by analyzing sales and return history as well as specific product attributes, and differences across models and manufacturers.

As their data pools grew, they began to have performance, reliability, and management challenges related to the data infrastructure challenges. The data was stored in an overflowing RDBMS based data warehouse, with some limited complimentary tooling attempts, to keep cobbling together. Workarounds had hit the point of diminishing returns. Regular ingest (and occasional correction) of retailers' sales datasets took hours, sometimes failing silently. Data scientists spent considerable time working around data access and query limitations, often slicing data into much smaller segments than they would prefer and wasting valuable time strategizing the best way to massage specific data elements out of the system.

Objectives

BigR.io was engaged to help modernize and design the next generation data infrastructure, alleviating these problems and opening up future growth of both data volume and analytical approaches and tooling.

Specific objectives of this new architecture included:

- Improvement of data set ingests time to minutes.
- Ability to replay and correct past data ingests and reflect changes in "downstream" analyses.
- Interactive analysis and query across broader cross-sections of data.
- Support for a variety of analysis and reporting tools supporting the needs of both business users and data scientists.
- Strict schema enforcement.
- Managed infrastructure with bounded costs.

Results

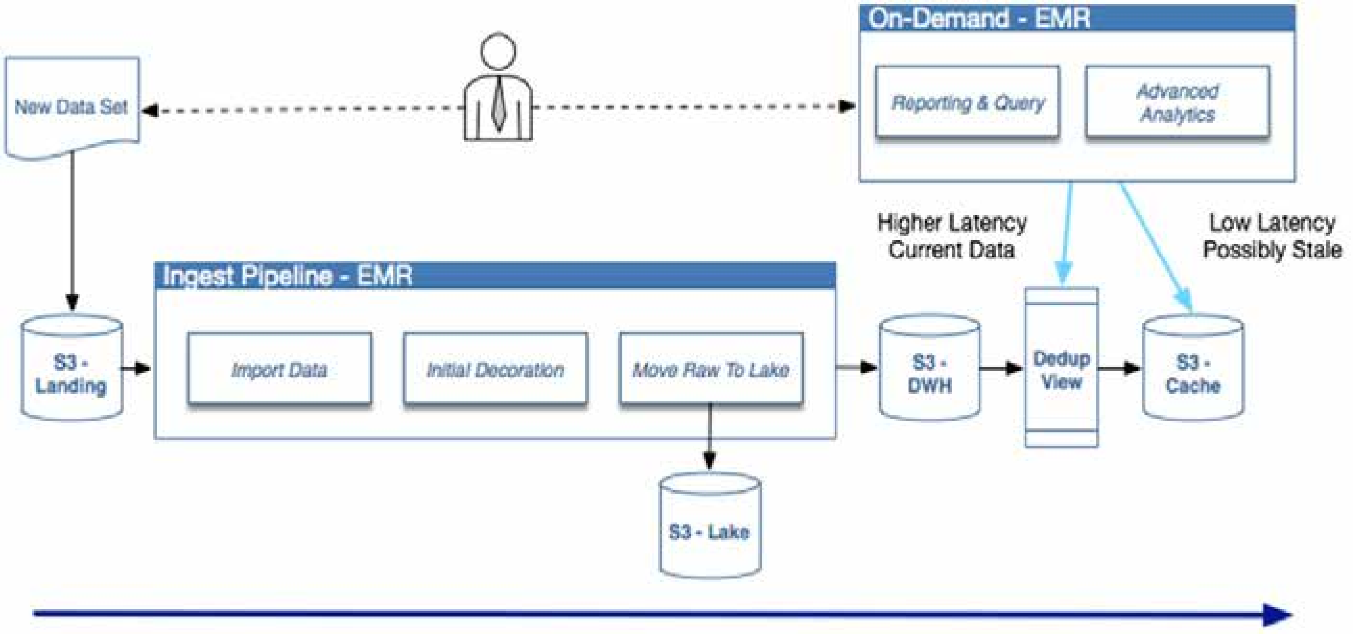
The architecture that was selected is based on Amazon Web Services (AWS) and has the following characteristics:

- Data at rest is stored in S3 using a tiered bucketing topology. Raw ingest data is cataloged in a “data lake” storage area, and stored original data and downstream analytics are structured in external tables managed by Hive/H Catalog.

- Processing of data is conducted on ephemeral Elastic Map Reduce (EMR) clusters accessing S3 via EMRFS; HDFS is utilized in-job where appropriate, but compute and storage are effectively decoupled.

- Business users are able to query data via Hive, Presto, and Spark SQL and to visualize using Tableau and other BI tools; developers and data scientists are able to use Apache Pig (allowing use of existing logic), Spark, and integrated Scala and Java logic.

- All data writes are immutable but partitioned on a per-ingest-job basis, allowing a last written strategy to support logical updates; a caching layer helps address speed and concurrent access concerns.



This new architecture resulted in “a huge win” and paved the way for long-term growth while eliminating the immediate pains:

- Queries that used to take hours (or that could not be run at all) can now be executed in minutes or seconds.

- Analyses that previously needed to be time-sliced into increments as small as one day of data can now be run over all time.

- All objectives were met, enabling our client to focus on their core business, refining the analytical models that are spearheading their industry instead of stumbling over infrastructure challenges.