June 2019

Into The Cloud: Twitter Presto's Journey to GCP

Outline

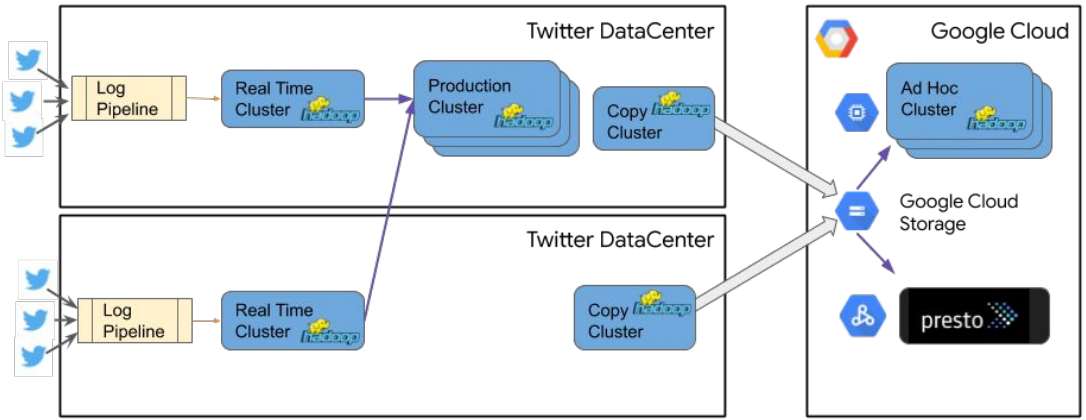
Overview

Performance

Authentication
Authorization
Auditing

Partly Cloudy

- Data replicated from Twitter data center to Google Cloud Storage (GCS) bucket
- Access permission for each bucket is managed by LDAP groups



Performance

Performance

- We observed significant read amplification using gcs-connector
  - Presto sees 70 GB/s
  - Google side reports 250 GB/s
  - ~4x read amplification

Performance

- The root cause ended up being the streaming range HTTP requests
  - read from the starting point till the end of the file
  - cancel the request when it moves to next range

	Before	After
Parquet Reader	`readFully(position, buffer, offset, length)`	`readFully(position, buffer, offset, length)`
GCS Connector	`GET https://www.googleapis.com/storage/v1/... RANGE=position-filesize`	`GET https://www.googleapis.com/storage/v1/... RANGE=position-{position+length}`
Read Amplification	~4x	~1x

Authentication
Authorization
Auditing

Authentication & Auditing

- Enabled HTTPS/TLS for client-coordinator communication
  - Internal communication via HTTP
- Integrated Kerberos / LDAP authentication
- Query audit log via Presto Event Listener
  - Audit logs are queryable in Presto

Authorization

- Storage-based security
  - Interrogate the storage (directory) permissions, instead of checking the Metastore for grants
- How it works on-prem with HDFS?
  - HDFS Impersonation
- How it works in the Google Cloud?
  - No fine-grained impersonation mechanism provided by cloud vendors
  - OAuth token based authorization

Token-based Authorization Made Possible

- Client provides its own OAuth token to access GCS buckets
- OAuth token is submitted to Presto coordinator via *`X-Presto-Extra-Credential`* header
- OAuth token is distributed to all workers in *`ConnectorIdentity#extraCredentials`*

Token-based Authorization Made Possible

- Hive Connector extracts the OAuth token from ConnectorIdentity
- HDFS client reads from GCS with `GcsAccessTokenProvider`

Even More Possibilities...

- We made the credential pass-through mechanism generic enough that can support lots of different use cases
  - it's implemented as a set of key-value pairs with no namespace
- Enable per-query authorization in JDBC based connector
  - user and password overridden by extra-credentials provided by the client

Q & A