Presto@Twitter

Journey to the Cloud and Federation

Maosong Fu, Presto Team@Twitter

October 2019

Outline

Overview

Presto on Google Cloud Platform (GCP)

Federated Presto

Presto @Twitter

- Clusters on-prem
    - ad-hoc cluster: ~2000 nodes
    - schedule cluster: ~500 nodes
    - clusters dedicated for heavy Presto customers
    - and more...
- Clusters on GCP
    - elastic; can scale from 50 to 800 nodes
    - deployed in DataProc

Presto Query @Twitter

- Ad-hoc interactive analysis only
- Data format: Parquet, lzo-thrift
- Daily queries: ~40K
- Daily processed data: ~50PB

Presto on GCP: Performance

Performance: Range Request in GCS connector

- Tested against dataset: ~15PB in parquet format; hourly partitioned with 3000 files each; 500 to 800 MB per file.

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

Performance: Range Request in GCS connector

- 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

Presto on GCP: 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 Presto worker via *`X-Presto-Extra-Credential`* header
- OAuth token is passed to connectors in *`ConnectorIdentity#extraCredentials`*

Token-based Authorization Made Possible

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

Even More Possibilities...

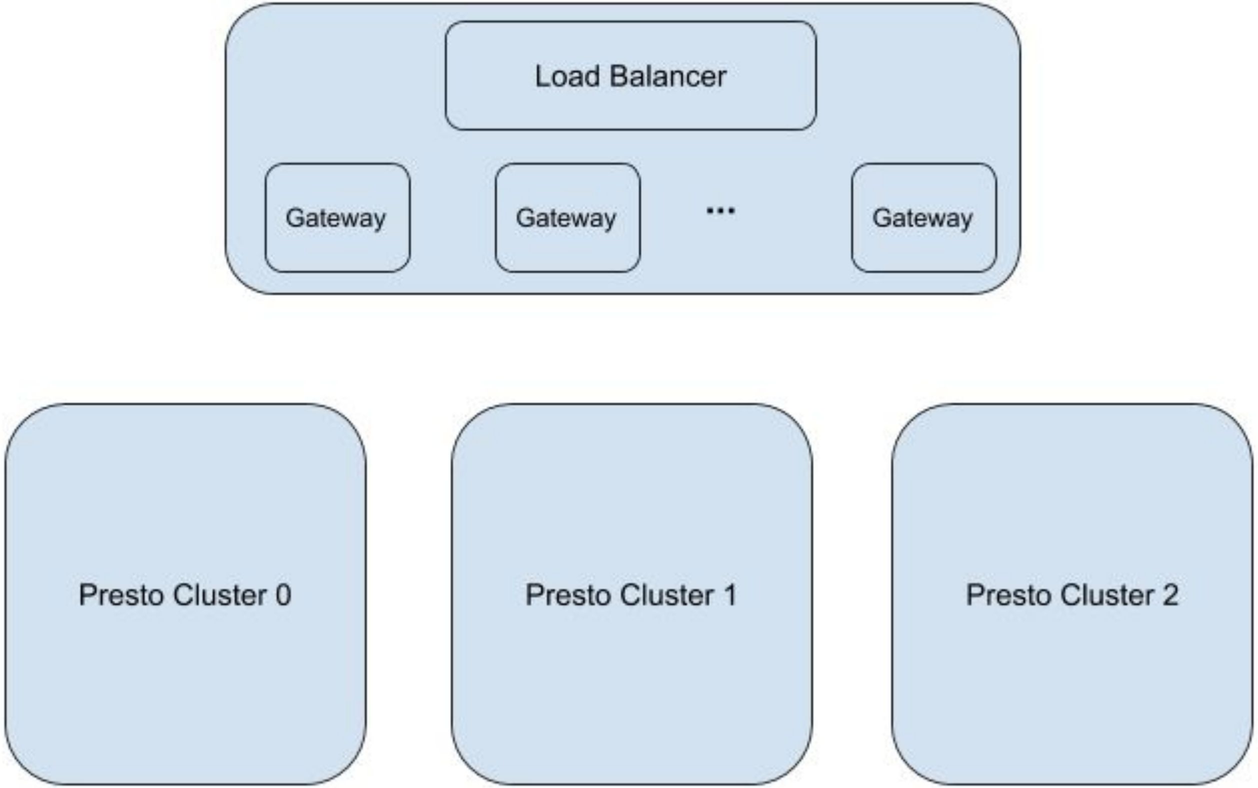
- We made the credential pass-through mechanism generic enough that can support lots of different use cases
  - it's implemented as a set of key-value pairs with no namespace
- Enable per-query authorization in JDBC based connector
  - user and password overridden by extra-credentials provided by the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

Federated Presto

Motivation

- Better Scalability
- Better Resource management and isolation
- High Availability and Failure isolation
- Better maintainability: rolling upgrade, auto-scaling, etc

Architecture Overview



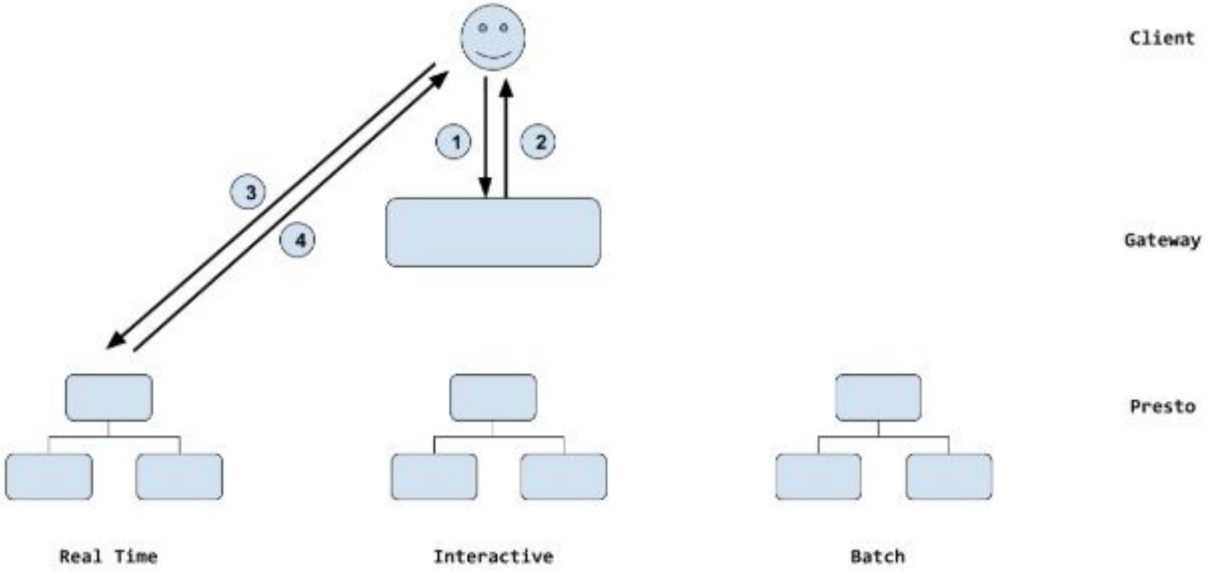
Federated Presto: Query Dispatching

Workload Characterization and Classification

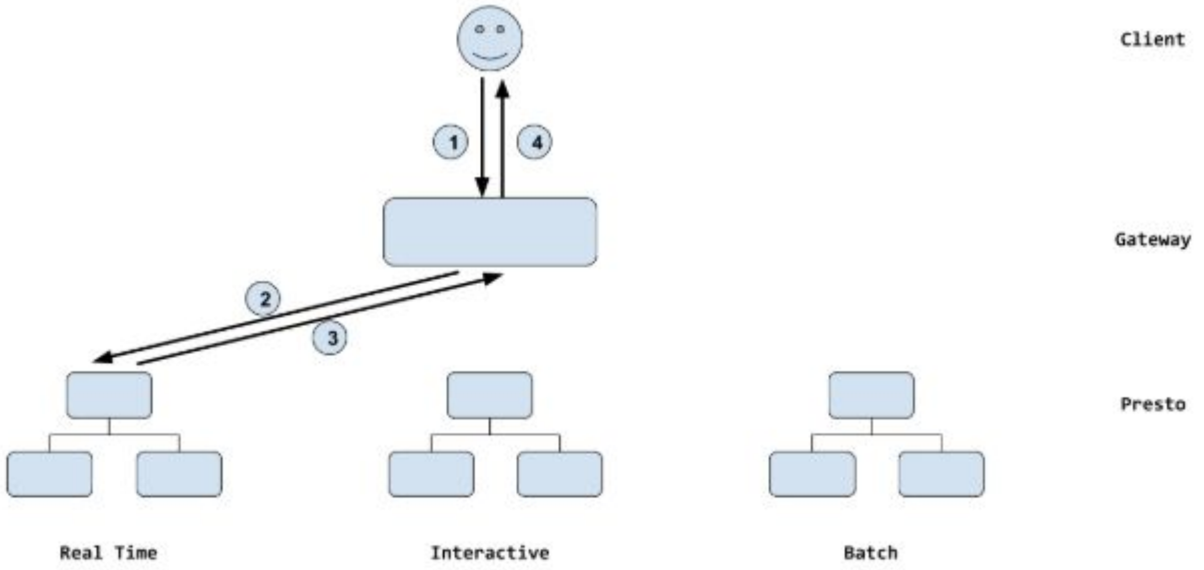
- Real-time
  - DATA_DEFINITION
  - DESCRIBE
  - EXPLAIN(analyze=false)
- Interactive
  - SELECT
- Batch
  - EXPLAIN(analyze=true)
  - ANALYZE
  - INSERT
  - DELETE
- Still rooms to improve...

Federated Presto: Protocol

Client Protocol



Alternatives considered: Proxy



Federated Presto: More

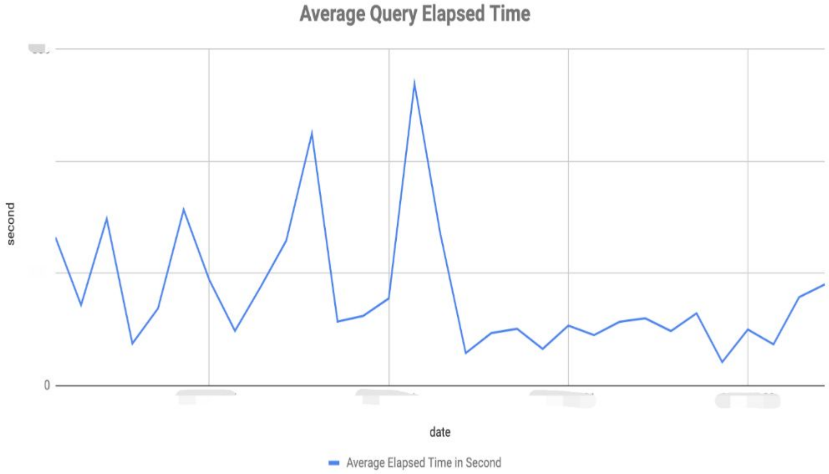
Rolling Upgrade/Auto-Scaling

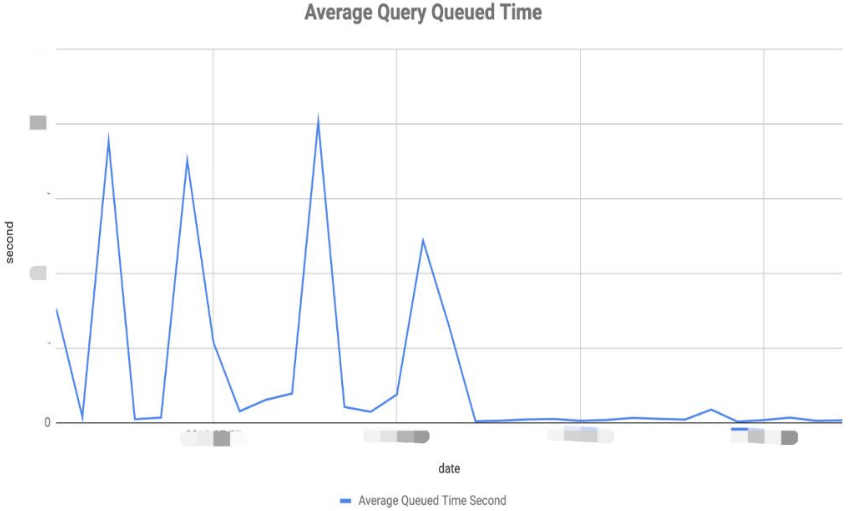
- Add a Presto cluster
  a. spin up the Presto cluster completely
  b. add the Presto cluster to the cluster manager in all gateway servers
- Remove a Presto cluster
  c. remove the Presto cluster from the cluster manager in all gateway servers
  d. shut down the Presto cluster

Federated Presto: Performance

Performance

- Scaled the Ad-hoc Cluster from ~500 nodes to ~2000 nodes
- Query Elapsed Time: Weekly Average reduced **~3x**; Weekly P99 reduced **~4x**.
- Query Queued Time: Weekly Average reduced **~10x**; Weekly P99 reduced





Thank you.

Q&A