Grab

Presto @ Grab

About the Speaker

Edwin Law - Senior Data Engineering Manager

Been with Grab for 6 years, since August 2013

Managing ~50 Engineers in 4 Data Engineering teams

Reach me at:

email - edwin.law@grab.com

Linkedin - linkedin.com/in/edwinlawhh

Grab Today

DRIVING SOUTHEAST ASIA FORWARD

Grab aims to serve its communities by bringing Southeast Asians closer to what matters to them

Over **163 Million** mobile downloads

Over **9 Million** micro-entrepreneurs across our network

339 CITIES* 8 COUNTRIES

*Includes Indonesian cities and regencies, aligning with the government's definition of second-level administrative subdivision under the provinces

VIETNAM
MYANMAR
THAILAND
PHILIPPINES
CAMBODIA
MALAYSIA
SINGAPORE
INDONESIA

Provide the everyday services that matter the most to our users

Making transportation safe, accessible and affordable to all

› Seven R&D centres located in Singapore, Beijing, Seattle, Bangalore, Ho Chi Minh City, Kuala Lumpur and Jakarta

Grab Today

SERVICES — GRAB OFFERS PAYMENTS AND A RANGE OF CONSUMER SERVICES THROUGH ONE MOBILE APP

MOBILITY

GET A RIDE

GrabTaxi
› E-hailing solves safety and price certainty issues associated with traditional taxis in the region.

GrabBike
› Fastest growing transport service. Most popular option in Greater Jakarta, where population is 60% larger than Beijing.

GrabCar
› Economy and premium options.

GrabTukTuk / GrabRemorque / GrabThoneBane
› Book an affordable ride via our three-wheeler service in the Philippines, Cambodia and Myanmar.

GrabWheels
› A shared active mobility service available in Singapore.

GET IT FASTER

JustGrab
› Automatically assigns a vehicle from both Grab's taxi and car fleets at fixed fares for a faster ride.

SHARE A RIDE

GrabShare
› Commercial on-demand carpooling service for passengers to share their ride.

GrabShuttle
› Pre-book a shuttle bus seat for an affordable, comfortable commute.

FINTECH

GrabFinancial
› Micro-lending services to millions of unbanked consumers and micro-entrepreneurs across Southeast Asia.
› Micro-insurance products for driver-partners to protect their vehicles and livelihoods.

GrabPay
› The only digital payments provider in SEA with access to e-money licenses in the six major ASEAN economies.
› P2P fund transfers using GrabPay mobile wallet.
› QR-code enabled payments with merchants in restaurants and shops.

GrabRewards
› Southeast Asia's largest loyalty programme, with over 500 merchants across the region.
› Users can earn points while using Grab's services and redeem discounts through the GrabRewards catalogue.

LIFESTYLE

GrabExpress
› On-demand delivery of parcels in Indonesia, Malaysia, Singapore, Thailand and Vietnam.

GrabFood
› An on-demand food delivery platform available in Indonesia, Thailand, Vietnam, Malaysia, Singapore and Philippines.

GrabFresh
› On-demand grocery delivery service available in Indonesia and Thailand.

GrabAssistant
› A concierge service in Philippines that lets customers ask the riders to buy items or queue for them.

Before Presto, there was ...

- MPP Data Warehouse , heavily used in 2015 - 2017

- Pros

  - **Access control** features out of the box (Postgres like)

  - **Simple architecture** - one piece of infra to manage

  - storage & compute combined

- Cons

  - **Resource contention** - everyone fighting for same resources

  - Hard (slow) to **scale out** (vastly improved today)

 - **Concurrency limits** (~15 back then, much higher now)

presto

- Our Chosen SQL Query Engine to access our Data Lake on S3
- Pros
  - Enabled **Multi Cluster Architecture** (Decoupled storage and compute)
  - Very fast scale in/out, up/down
  - Great Open Source community
- Cons
  - Complex architecture with many interdependent moving parts
  - Works great but missing many of the database features out of the box
  - Had to build / integrate access control, monitoring, etc.
  - Needs a larger team to support the ecosystem

Presto Basics
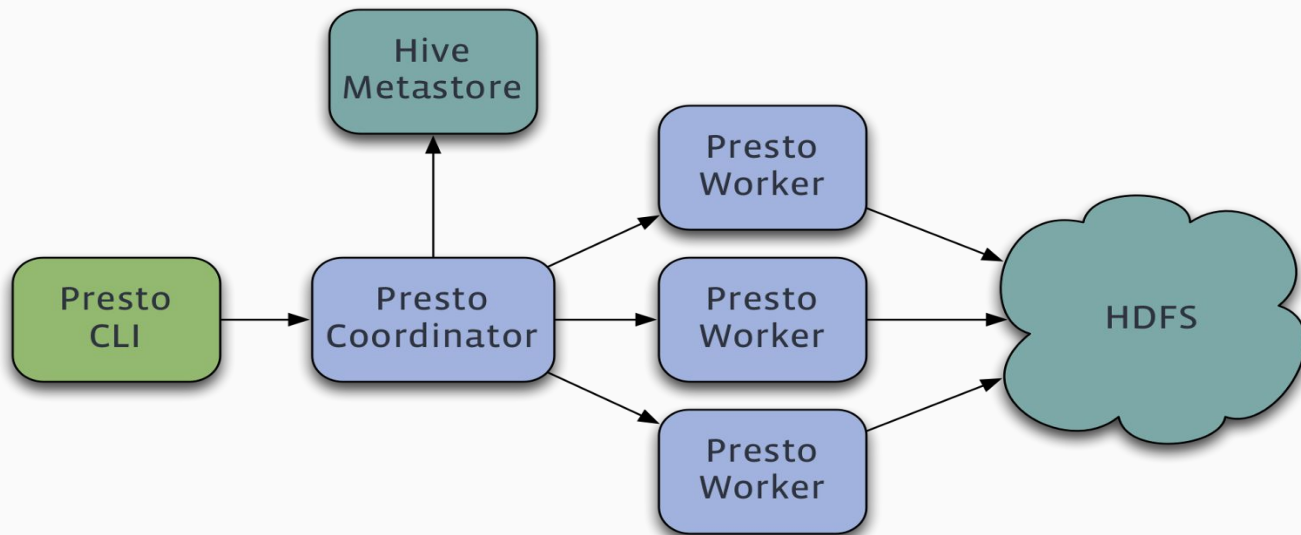
presto

Our Architecture
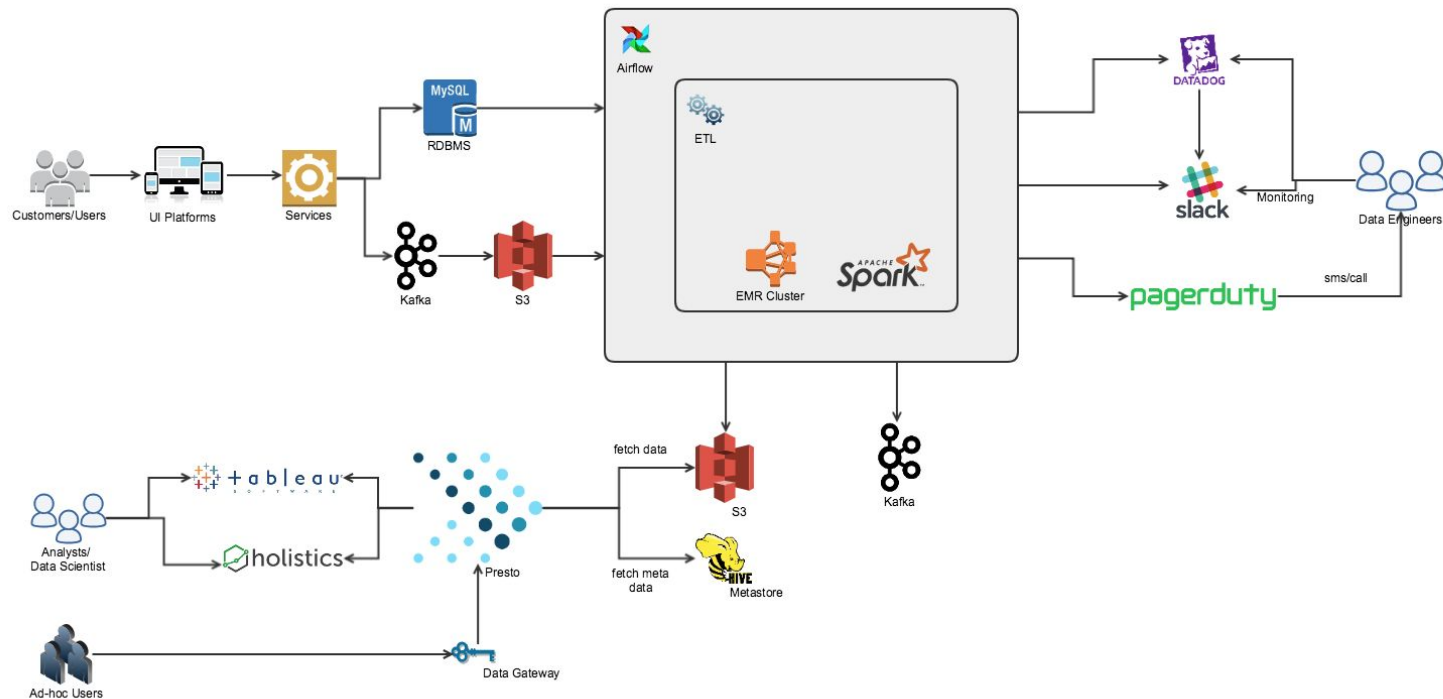
Data Gateway

- Built this early on to get access control for Presto
- Single entry point for all queries
- Authentication and Authorisation service
- Grant access to schemas, tables and clusters
- Integrated with API endpoints of Presto
- Users can route queries to multiple clusters

Presto on EMR

- Pros
  - We were already familiar with EMR, use it for our Spark workloads as well
  - Relatively easy to setup, out of the box compatible with our other infrastructure components on AWS
  - Reliable and Scalable
- Cons
  - Configuration Changes were not that easy to apply
  - Cluster Administration interface was clunky
  - No workload aware autoscaling (not simple to setup)
  - High cost due to always on clusters

Presto on Qubole

- Pros
    - Workload aware Autoscaling Presto Clusters
    - Simple cluster administration, allowed us to
    - Cost saving - 2x workloads at ⅓ the original cost on EMR
    - Cross cloud compatibility - AWS & Azure
- Cons
    - Still have workload isolation issues (too many queries from users on some clusters causing long wait times or poor performance)
    - Autoscaling + spot node consumption makes clusters seem less reliable compared to EMR clusters which were always at max size
    - We spun up too many clusters, it became challenging to manage

Current scale

- 40 workload aware autoscaling Qubole Presto Clusters
- More than 800 nodes at maximum scale
- Master Nodes : r5a.2xlarge (64GB)
- Worker Nodes : r4.8xlarge (> 80% spot nodes) (244GB)
- Serving Adhoc SQL queries and ETL jobs
- 200k+ Queries daily
- 600+ direct users, 3k+ users of BI tools connected to Presto

Extending the Data Gateway

- The Data Gateway is actually a proxy / query router
- We can make it do more than just access control
- 3 Layer Architecture (Query Router -> Compute Cluster -> Data)
- Extend it to route queries to different clusters based on query & cluster profiles
- Extend it to have it spin up Presto Clusters on K8s when queries come in
- Many other interesting things we can do on the query router layer

On Demand Presto on Kubernetes

- It takes ~30 seconds to create a new Presto Cluster on Kubernetes
- Every ETL query runs on it's own Presto Cluster
- For Ad-hoc queries, a cluster is created for every user
  - Spin up at first query and torn down when idle for some period

On Demand Presto on Kubernetes

- **Pros**:
  - Ultimate workload isolation
    - each user & each ETL job has dedicated resource
  - Best resource utilization through container packing in Kubernetes
  - Users don't need to think about resources available, it should always be available!
- **Cons**:
  - Added complexity, running Kubernetes is not trivial
  - Users might feel like they have less control, they no longer know where their query is running.
  - We need to figure out many things on our own, not many ppl doing it like this

On Demand Presto on Kubernetes

- Allow users to specify what presto version or configuration to use
- Add intelligence that learns based on historical query performance
  - Provision clusters that are just the right size to complete the query in reasonable amount of time
- Dynamically prioritize execution of queries based on query profile / user profile etc.

What else we're thinking about

- I believe the query router layer allows us to do interesting things.
- Query analysis engine:
  - advise users on how to rewrite their queries
  - Rewrites inefficient user queries before sending it to the compute cluster
- User Submit Standard ANSI SQL - we choose what engine to run it on
  - We could rewrite the query to execute in Presto / Spark / Hive etc.