CLOUD DAY 2020 29 OTTOBRE • #CLOUDDAY2020

BIGQUERY RESERVATIONS AND WORKLOAD MANAGEMENT

Valentino Miazzo Google Cloud





#CLOUDDAY2020

Kudos



managed/designs







Introduction



Agenda

Agenda What is BigQuery? **BigQuery pricing** Origin of a query Queries and pricing Reservation data model Workload management

2

3

4

5

6

7



What is BigQuery?



What is BigQuery?

- Multi-tenant, Exabyte scale DB
- Serverless
- HA and DR built-in
- Columnar OLAP engine
- Good for data lakes, data warehouses, data marts
- Less silos and pipes



10 years!





Spotify usage of BigQuery

>1200

Employees using BQ

>600

BigQuery projects

5000

BigQuery datasets >1m BigQuery

tables

400PB

Data processed monthly

>100PB

Data stored In BigQuery 6m Queries run monthly

500TB

Data loaded into BQ daily Some BigQuery Stats

10.5 Trillion Largest query (rows)

5 petabytes Largest query (data size)

250 petabytes Largest storage customer

4.5 million rows/sec Peak ingestion rate



BigQuery | Architecture

Decoupled storage and compute for maximum flexibility



Demo Query

1	SELECT
2	AVG(s.totalSale) analytic window function
3	OVER (PARTITION BY p.name ORDER BY orderDay
4	ROWS BETWEEN 7 PRECEDING AND CURRENT ROW) as runningAverage,
5	p.name as productName,
6	s.orderDay as orderDay
7	FROM (
8	SELECT
9	SUM(lines.totalSale) as totalSale,
10	DATE(orderDate) as orderDay,
11	lines.productKey as productKey
12	FROM `google.com:bigguery-petabyte.retail_petabyte.sales_partitioned_clustered`, 1PB, 1 trilion rows
13	UNNEST(salesOrderLines) as lines explode nested orders lines
14	GROUP BY orderDay, productKey
15) s
16	JOIN `google.com:bigquery-petabyte.retail_petabyte.products` p 0.5TB, 2 milions rows
17	ON s.productKey = p.productKey
18	WHERE
19	REGEXP_CONTAINS(s.productKey, r'^110453[0-9]{7}\-[0-9]{3}\$') complex condition
20	AND orderDay BETWEEN "2011-01-01" AND "2011-01-07" push down filter
21	ORDER BY orderDay
20	

BigQuery ML for predictive analytics

2

Execute ML initiatives without moving data from BigQuery

Iterate on models in SQL in BigQuery to increase development speed Automate common ML tasks, and hyperparameter tuning

3

LINEAR_REG, LOGISTIC_REG, K-MEANS, MATRIX_FACTORIZATION, TIME SERIES, BOOSTED TREE, DEEP NEURAL NETWORK, AUTO-ML, TENSORFLOW





BigQuery ML logistic regression example

Feed your model from BigQuery data and perform feature engineering in SQL

Tweak training using OPTIONS

CREATE OR REPLACE MODEL `us.sample_model`							
OPTIONS(model_type='logistic_reg') AS							
SELECT							
IF(totals.transactions IS NULL, 0, 1) AS label,							
<pre>IFNULL(device.operatingSystem, "") AS os,</pre>							
device.isMobile AS is_mobile,							
IFNULL(geoNetwork.country, "") AS country,							
IFNULL(totals.pageviews, 0) AS pageviews							
FROM							
`bigquery-public-data.google_analytics_sample.ga_sessions_*`							
WHERE							
_TABLE_SUFFIX BETWEEN '20160801' AND '20170630';							

Analyze GIS data in BigQuery with familiar SQL

Accurate spatial analyses with Geography data type over GeoJSON and WKT formats

Support for core **GIS functions** – measurements, transforms, constructors, etc... – **using familiar SQL**



Predicting hazardous driving: 7 weeks from "What?!" to "Wow!"



Tomorrow with <weather conditions>?

SELECT predicted_SeverityScore, Location FROM mI.PREDICT(MODEL HarshDrivingModels.Chicago, SELECT Location, TrafficVolume, Temperature, Visibility, Snow, Rain, Fog FROM WeatherForecast)





BigQuery pricing

BigQuery storage pricing

- Native Storage
 - Pay per GiB/mo (second resolution)
 - 7 days time travel included
 - Automatic 50% discount for tables not changed in the last 90 days
- Google Cloud Storage
 - \circ For further savings
 - Import/export
 - \circ Federation









A BigQuery slot is a unit of computational capacity required to execute SQL queries.



BigQuery compute pricing

- Free: load/export jobs, tables re-clustering
- Option 1: on-demand
 - Volume based
 - Metric: bytes read
- Option 2: flat-rate
 - Power based
 - Metric: slot second
 - Longer the commitment, lower the price
 - Per minute (AKA flex slots)
 - Per month
 - Per year





BigQuery Blended Pricing Model

"Mix and Match" pricing models to get best value for

money

Customer

	С. Ві	iogle Clou (Query	d Platforr	eservations arra	+ BUY SLOTS	Re CREATE RESER	WATION
	Cap	pacity sur	nmary				
_	Total	sicts					
	Tetal 1000	slots)					
	Tetal 1000	alota))	10	RESERVATIONS Ø	ASSIGNMENTS	ø	
	Tetal 1000 SLOTO	alota 3 2000/WITMEDVT Iter table	3 0	RESERVATIONS Ø	ASSIGNMENTS	0	
\leq	SLOT C Dates	alices 3 COMMETMENT for table Slots	3 Ø	EESERAJIONS Ø	ASSIGNMENTS	@ Location	Slot comm
\langle	SLOT C	alices 3 3 3 3 3 3 3 3 3 3 3 3 1 4 3 1 4 3 1 4 3 1 4 3 1 3 3 3 3	S Ø	EESERWATIONS Commitment and time November 22, 2019 at 1.55	ASSIGNMENTS	© Location United States (US)	Slot comm 51922768

Customers can buy capacity in seconds, and share any unused capacity across the business.

Business Intelligence, Standard ETL, Scheduled Jobs to build Data Mart	Flat Rate, Yearly/Monthly
"Monday Morning" ETL, Monthly Financial reporting, Black Friday	Flat Rate, Minute
Urgent one time inventory analysis	On demand

Purchasing based on "knowability" of needs within each use case



Customer's doubts

"If I buy Slots, can I still use the on-demand pricing in selected case?"

"Can I switch between flat-rate and on-demand pricing? How?"



Origin of a query



Origin of a query

Every **query** has **an origin Project** that is independent of the **Projects of the queried Tables**.







How

By changing the origin Project of a query you can change the resources available to the query and the pricing model applied.

- A query with origin in a Project without Assignments will use the on-demand model.
- A query with origin in a Project with Assignments will use the flat-rate model and will have access to the Slots assigned.

This choice can be done ahead of time or **even dynamically**, just before running the query.



Example - Assign Slots to a Project





Example - query with flat rate pricing





Example - query with on demand pricing





Slots assignment

The big picture





Reservations data model



Reservations data model

UML diagram





Reservations data model

Example





Reservations data model

Assignments and Resource hierarchy





Workload management



Workload management

Flat-rate prioritized sharing

- Company Argo has a 2000 Slots yearly Commitment assigned to the organization.
- Some critical reports are ready after the not critical ones.
- Some "not approved" workloads are consuming lot of Slots
- Solution:
 - 3 compute Projects
 - "Critical", has 1450 Slots
 - "NotCritical", has 500 Slots
 - Idle Slots are freely consumed by the 2 above
 - "Quarantine" has 50 Slots and no idle Slots



Argo Reservation Ξ slots=1450 Folder 1 Commitment C Reservation slots=2000 plan=ANNUAL slots=500 Admin Project L----Reservation Project Project slots=50 Quarantine **NotCritical** ignore isle slots "Compute" hierarchy

Workload management

Flat-rate prioritized sharing

C Google Cloud

Policy inheritance

Workload management

Flat-rate manual scaling

- Company Neon has a 500 Slots yearly Commitment.
- Some critical weekly reports often don't finish in time.
- Upgrade to 1000 Slots is out of their budget.
- Solution:
 - Move critical reports on a dedicated compute Project P
 - Buy an additional Flex Commitment
 - Use it to accelerate the reports
 - Dispose it when the reports are completed



Workload management

Flat-rate manual scaling





Conclusion

- BigQuery is a serverless analytical DB
- Endlessly scalable
- Built-in streaming and ML
- Where your organization can collectively work
- Using the most suitable pricing model
- Allocating resources in a flexible and dynamic way





Trial slots promotion





Thank you!

Google Cloud