Module 13

Optimizing for Performance

In this module we will:

- **Avoid BigQuery Performance Pitfalls**
- Prevent Hotspots in your Data
- Diagnose Performance Issues with the Query Explanation map

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Four Key Elements of Work

- **I/O** — How many bytes did you read?
- **Shuffle** — How many bytes did you pass to the next stage?
  - Grouping — How many bytes do you pass to each group?
- **Materialization** — How many bytes did you write to storage?
- **CPU work** — User-defined functions (UDFs), functions

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Avoid **Input / Output Wastefulness**

- Do not SELECT *, use only the columns you need
- Filter using WHERE as early as possible in your queries
- Do not use ORDER BY without a LIMIT

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Google Cloud

Module 13

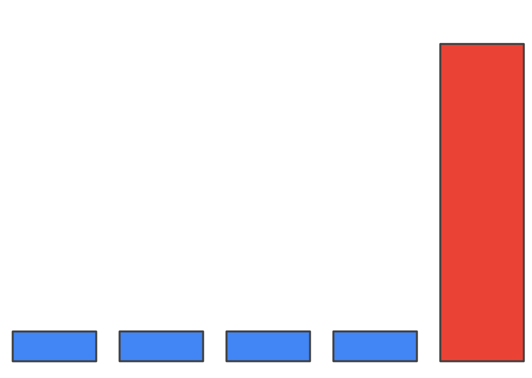
Optimizing for Performance

In this module we will:

- Avoid BigQuery Performance Pitfalls
- **Prevent Hotspots in your Data**
- Diagnose Performance Issues with the Query Explanation map

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Shuffle Wisely: **Be Aware of Data Skew** in your Dataset



Skewed Data creates an imbalance between BigQuery worker slots (uneven data partition sizes)

- **Filter your dataset** as early as possible (this avoids overloading workers on JOINs)
- Hint: Use the Query Explanation map and compare the Max vs the Avg times to highlight skew
- BigQuery will automatically attempt to reshuffle workers that are overloaded with data

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Careful use of GROUP BY

- Best when the number of distinct groups is small (fewer shuffles of data).
- Grouping by a high-cardinality unique ID is a bad idea.

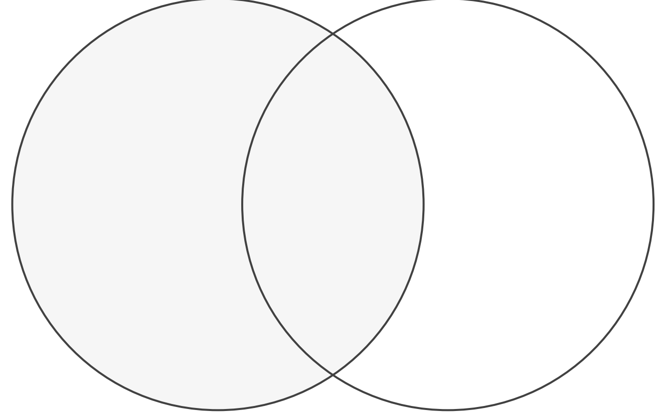
Row	contributor_id	LogEdits
1	2221364	4
2	104574	4
3	73576	4
4	311307	4
5	291919	4
6	140178	4
7	181636	4
8	3661553	4
9	3600820	4
10	4737290	4
11	938404	4
12	295955	4
13	183812	4
14	1811786	4
15	8918196	4
16	561624	4
17	5338406	4

← Do not Group on an ID

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Joins and Unions

- Know your join keys and if they're unique -- no accidental cross joins
- LIMIT Wildcard UNIONS with _TABLE_SUFFIX filter
- Do not use self-joins (consider window functions instead)



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Limit UDFs to Reduce Computational Load

- Use native SQL functions whenever possible

- Concurrent rate limits:
  - for non-UDF queries: 50
  - for UDF-queries: **6**

```
CREATE TEMP FUNCTION SumFieldsNamedFoo(json_row STRING)
  RETURNS FLOAT64
  LANGUAGE js AS """
function SumFoo(obj) {
  var sum = 0;
  for (var field in obj) {
    if (obj.hasOwnProperty(field) && obj[field] != null) {
      if (typeof obj[field] == "object") {
        sum += SumFoo(obj[field]);
      } else if (field == "foo") {
        sum += obj[field];
      }
    }
  }
  return sum;
}
var row = JSON.parse(json_row);
return SumFoo(row);
""";
```

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Module 13

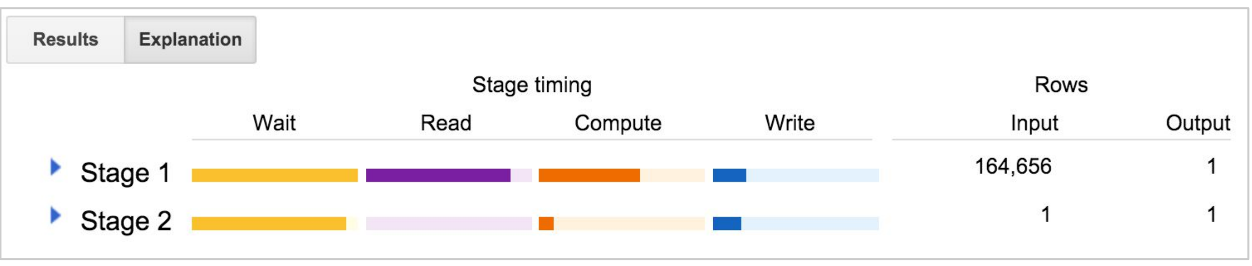
Optimizing for Performance

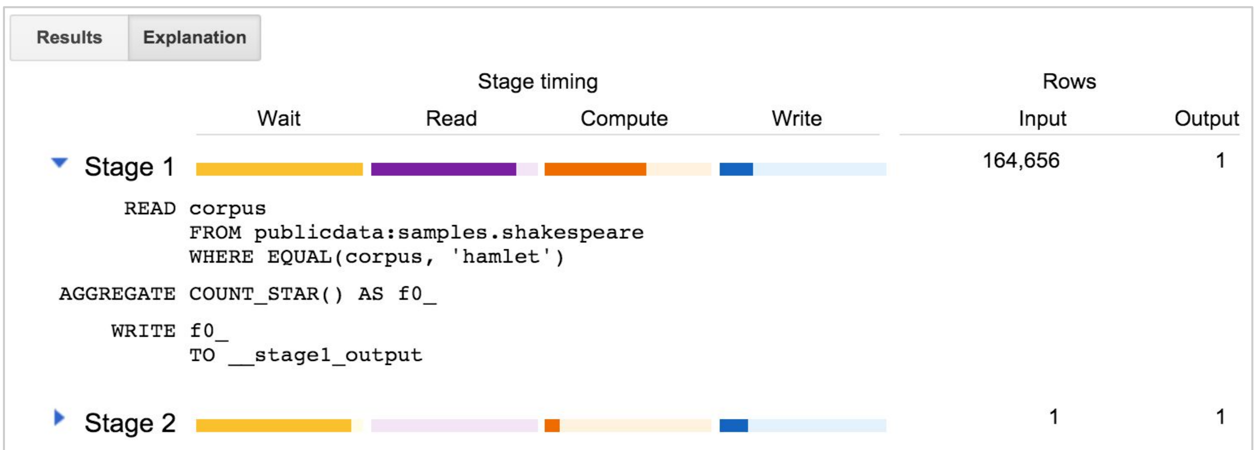
In this module we will:

- Avoid BigQuery Performance Pitfalls
- Prevent Hotspots in your Data
- **Diagnose Performance Issues with the Query Explanation map**

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Diagnose Performance Issues with the **Query Explanation** Map





© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Diagnose Performance Issues with the **Query Explanation** Map

The following ratios are also available for each stage in the query plan.

API JSON Name	Web UI*	Ratio Numerator **
`waitRatioAvg`	AVG	Time the average worker spent waiting to be scheduled.
`waitRatioMax`	MAX	Time the slowest worker spent waiting to be scheduled.
`readRatioAvg`	AVG	Time the average worker spent reading input data.
`readRatioMax`	MAX	Time the slowest worker spent reading input data.
`computeRatioAvg`	AVG	Time the average worker spent CPU-bound.
`computeRatioMax`	MAX	Time the slowest worker spent CPU-bound.
`writeRatioAvg`	AVG	Time the average worker spent writing output data.
`writeRatioMax`	MAX	Time the slowest worker spent writing output data.

* The labels 'AVG' and 'MAX' are for illustration only and do not appear in the web UI.

** All of the ratios share a common denominator that represents the longest time spent by any worker in any segment.

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Example: Large GROUP BY

Large GROUP BY query

```
SELECT
  LogEdits, COUNT(contributor_id) AS Contributors
FROM (
  SELECT
    contributor_id,
    CAST(LOG10(COUNT(*)) AS INT64) LogEdits # Buckets user edits
  FROM `publicdata.samples.wikipedia`
  GROUP BY contributor_id)
GROUP BY LogEdits
ORDER BY LogEdits DESC
```

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Large GROUP BY Means **Many Forced Shuffles**

Stage 4

Stage 3

Shuffling can occur

Stage 2

Stage 1

Master

Shard

Shard

Shard

Shard

Shard

Shard

Network Reads

Distributed storage (Google File System)

```
READ Contributors, LogEdits FROM stage3 output AS
publicdata:samples.wikipedia
SORT LogEdits DESC
WRITE Contributors, LogEdits to output
```

```
READ Contributors, LogEdits from stage2 output AS
publicdata:samples.wikipedia
AGGREGATE SUM_OF_COUNTS(Contributors) AS Contributors
GROUP BY LogEdits
WRITE Contributors, LogEdits to stage3 output
```

```
READ contributor_id, f0_ FROM stage1 output AS
publicdata:samples.wikipedia
AGGREGATE SUM_OF_COUNTS(f0_) AS f0_ GROUP BY contributor_id
COMPUTE INTEGER(LOG10(f0_))
AGGREGATE COUNT(contributor_id) AS Contributors GROUP BY
LogEdits
WRITE Contributors, LogEdits to stage2_output BY
HASH(LogEdits)
```

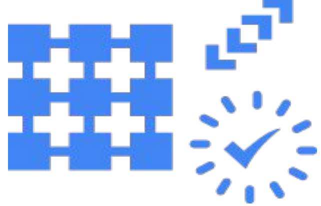
```
READ contributor_id FROM publicdata:samples.wikipedia
AGGREGATE COUNT_STAR() AS f0_ GROUP BY contributor_id
WRITE contributor_id, f0_ to stage1 output BY
HASH(contributor_id)
```

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Table Sharding - Then and Now

Traditional databases get performance boost by partitioning very large tables

Usually requires an administrator to pre-allocate space, define partitions, and maintain them



Manual **Table sharding** divides big table into smaller tables with new suffix of YYYYMMDD

Queries use Table Wildcard functions

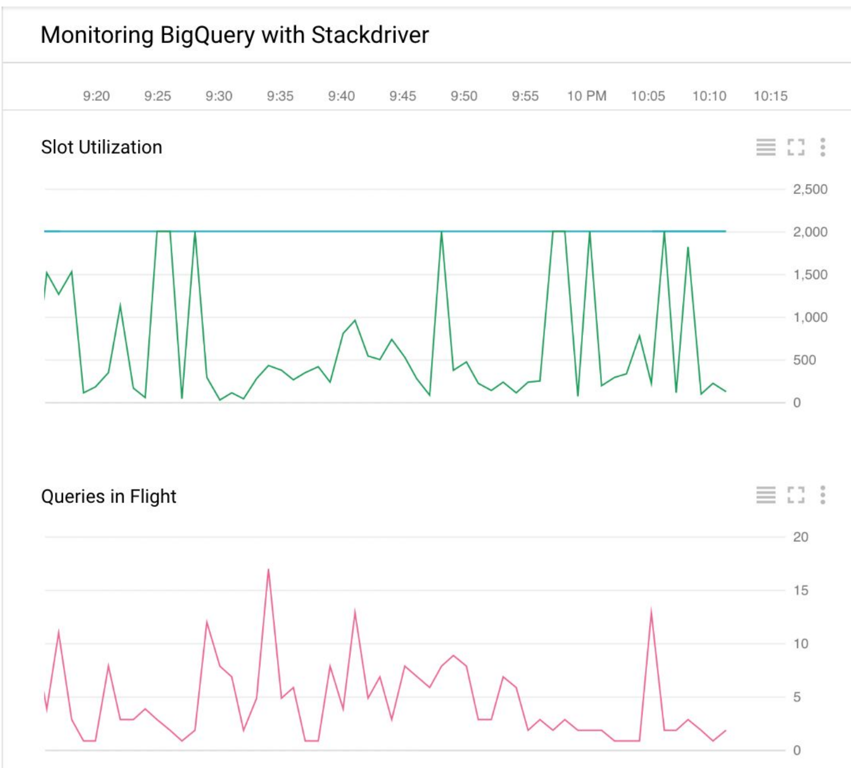
2017
2016
2015

Date Partition a single table based on specified DAY or a Date Column

Creating Partitioned Tables

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Monitor Performance with Stackdriver

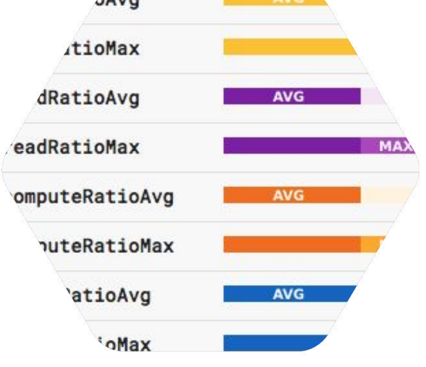


- Available for all BigQuery customers
- Fully interactive GUI. Customers can create custom dashboards displaying up to 13 BigQuery metrics, including:
  - Slots Utilization
  - Queries in Flight
  - Uploaded Bytes (not shown)
  - Stored Bytes (not shown)

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Google Cloud

Summary: Query and data model design has a significant impact on performance



Query only the rows and columns you need to reduce bytes processed

Investigate the query explanation map to see if data skew is bottlenecking your query

Avoid SQL anti-patterns like ORDER BY without a LIMIT or a GROUP BY on high-cardinality fields

Use table partitioning to reduce the volume of data scanned

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Lab 11
Optimizing and Troubleshooting Query Performance

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Optimizing and Troubleshooting Query Performance

In this lab, you will fix and troubleshoot SQL queries for performance improvements.

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Google Cloud