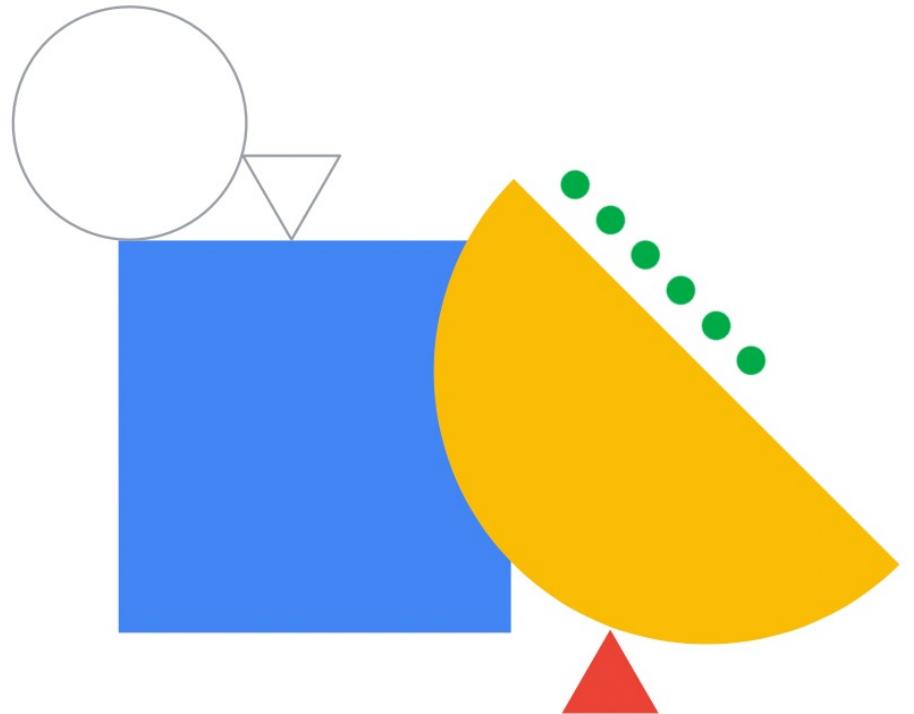


High-Throughput BigQuery and Bigtable Streaming Features



Module agenda

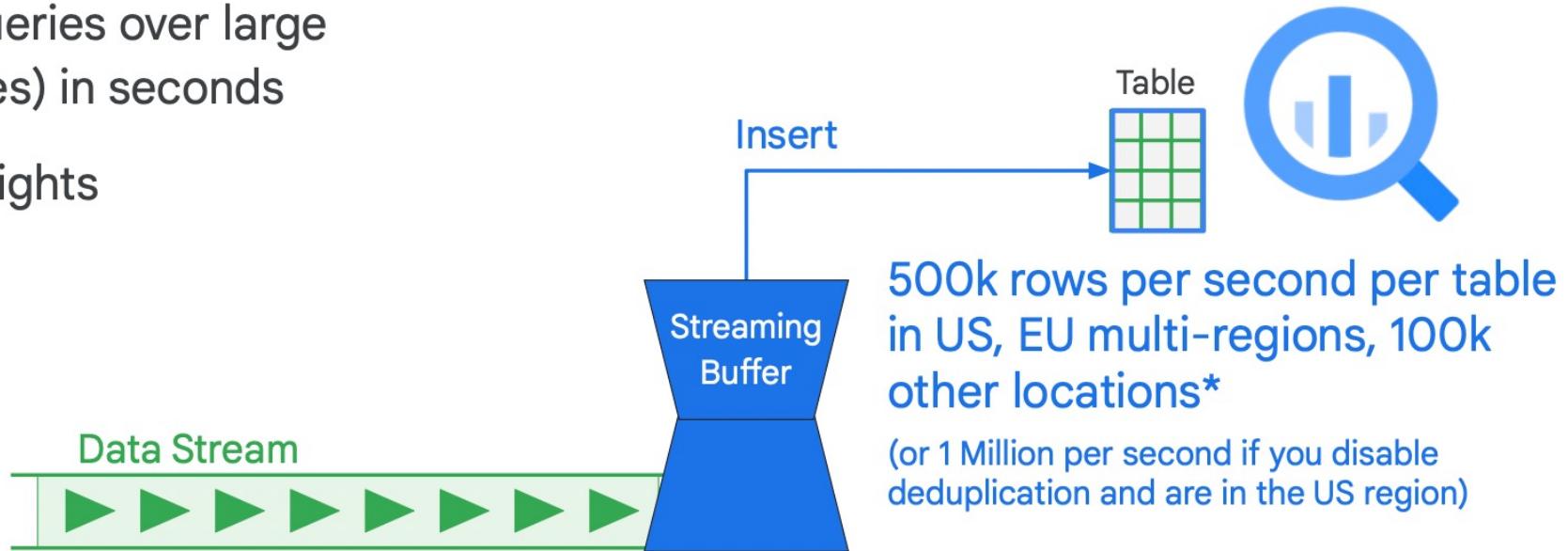
- 01** Streaming into BigQuery and Visualizing Results
 - 02** High-Throughput Streaming with Cloud Bigtable
 - 03** Optimizing Cloud Bigtable Performance
- 



Streaming into BigQuery and Visualizing Results

BigQuery allows you to stream records into a table; query results incorporate latest data

- Interactive SQL Queries over large datasets (petabytes) in seconds
- Near-real-time insights



Note:

Unlike load jobs, there is a cost for streaming inserts (see [quota and limits](#))

Insert streaming data into a BigQuery table

```
export GOOGLE_APPLICATION_CREDENTIALS="/home/user/Downloads/[FILE_NAME].json"
```

```
pip install google-cloud-bigquery
```

Install API

Credentials

The service must have Cloud IAM permissions set in the Web UI

```
from google.cloud import bigquery
client = bigquery.Client(project='PROJECT_ID')

dataset_ref = bigquery_client.dataset('my_dataset_id')
table_ref = dataset_ref.table('my_table_id')
table = bigquery_client.get_table(table_ref) ----- Get table access from API

# read data from Cloud Pub/Sub and place into row format
# static example customer orders in units:
rows_to_insert =
    [(u'customer 1', 5),
     (u'customer 2', 17),
    ]
errors = bigquery_client.insert_rows(table, rows_to_insert)
```

Python

Create a client

Access dataset and table

Perform insert

Review streaming data in BigQuery

Query editor

```
1 select * from cloud-training-demos.demos.current_conditions;
```

Run Save query Save view Schedule query More

Query results [SAVE RESULTS](#) [EXPLORE DATA](#)

Query complete (1.2 sec elapsed, 14.3 MB processed)

Job information [Results](#) [JSON](#) [Execution details](#)

Row	timestamp	latitude	longitude	highway	direction	lane	speed	sensorid
1	2008-11-01 11:55:00 UTC	33.191415	-117.363042	5	N	4	10.5	33.191415,-117.363042,5,N,4
2	2008-11-01 11:55:00 UTC	33.191415	-117.363042	5	N	4	10.5	33.191415,-117.363042,5,N,4
3	2008-11-01 09:35:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4
4	2008-11-01 12:30:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4
5	2008-11-01 09:40:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4
6	2008-11-01 10:55:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4

Want to visualize insights? Explore Google Data Studio insights right from within BigQuery

The screenshot shows the BigQuery Query editor interface. At the top, there's a "Query editor" header and a code editor containing the following SQL query:

```
1 select * from cloud-training-demos.demos.current_conditions;
```

Below the code editor are several buttons: "Run" (highlighted in blue), "Save query", "Save view", "Schedule query", and "More".

The main area is titled "Query results" and contains a "SAVE RESULTS" button. To its right is a "EXPLORE DATA" button, which is highlighted with a yellow box.

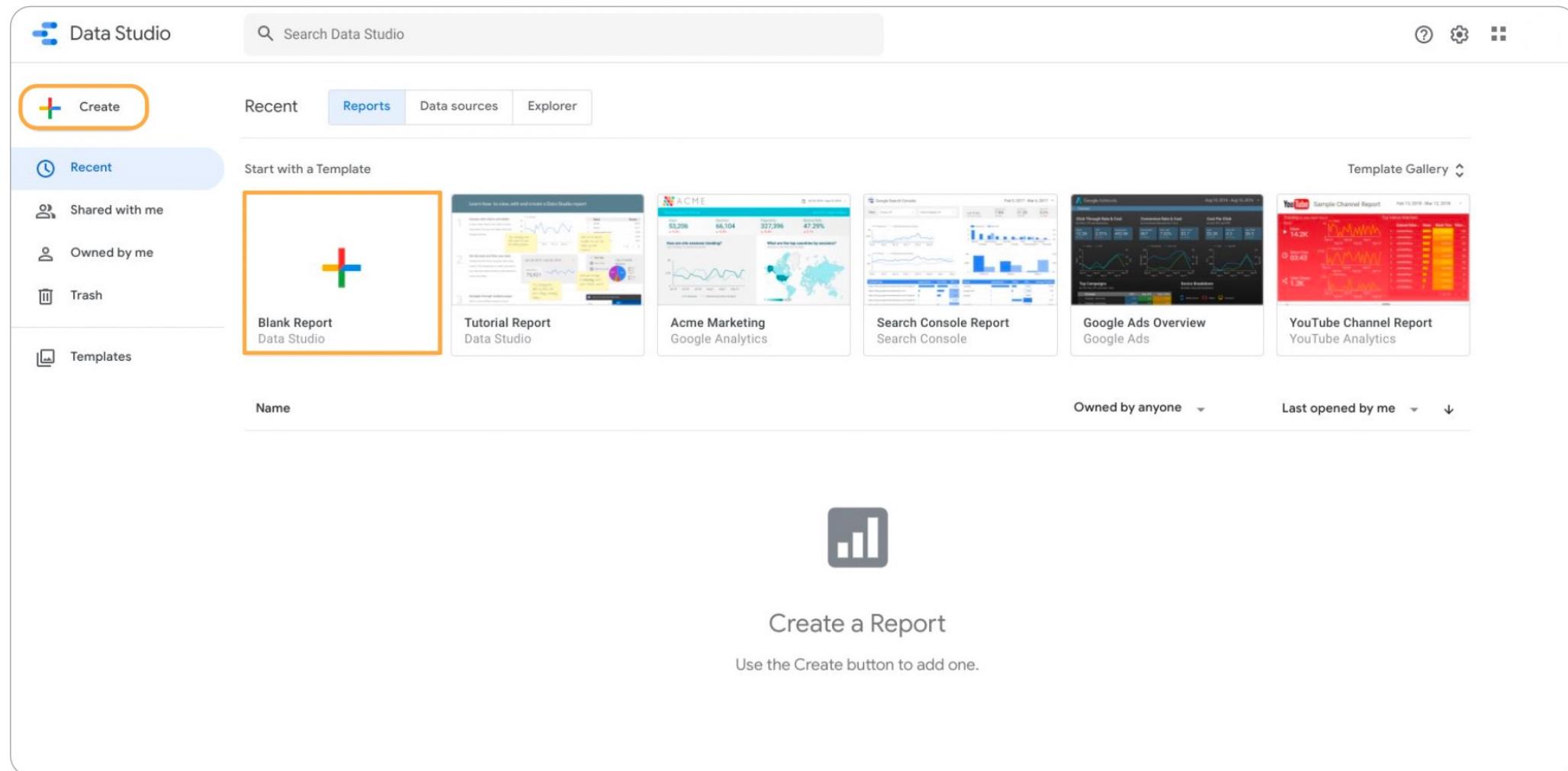
Below the results area, a message says "Query complete (1.2 sec elapsed, 14.3 MB processed)".

At the bottom, there are tabs for "Job information", "Results" (which is selected and underlined in blue), "JSON", and "Execution details".

The results table has the following columns: Row, timestamp, latitude, longitude, highway, direction, lane, speed, and sensorid. The data is as follows:

Row	timestamp	latitude	longitude	highway	direction	lane	speed	sensorid
1	2008-11-01 11:55:00 UTC	33.191415	-117.363042	5	N	4	10.5	33.191415,-117.363042,5,N,4
2	2008-11-01 11:55:00 UTC	33.191415	-117.363042	5	N	4	10.5	33.191415,-117.363042,5,N,4
3	2008-11-01 09:35:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4
4	2008-11-01 12:30:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4
5	2008-11-01 09:40:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4
6	2008-11-01 10:55:00 UTC	33.191415	-117.363042	5	N	4	11.0	33.191415,-117.363042,5,N,4

Create new reports in the Data Studio UI



The screenshot shows the Google Data Studio interface. At the top left is the 'Data Studio' logo. To its right is a search bar with the placeholder 'Search Data Studio'. On the far right of the header are three icons: a question mark, a gear, and a grid. Below the header, there are four tabs: 'Recent' (which is selected and highlighted in blue), 'Reports', 'Data sources', and 'Explorer'. A large orange box highlights the 'Create' button, which has a plus sign icon and the word 'Create' next to it. To the right of the 'Create' button are three more tabs: 'Recent', 'Reports', and 'Data sources'. Below these tabs is a section titled 'Start with a Template'. It contains a list of recent templates: 'Shared with me', 'Owned by me', and 'Trash'. Under 'Templates', there is a 'Blank Report Data Studio' template, which is also highlighted with an orange box. To the right of the 'Blank Report' are five other template cards: 'Tutorial Report Data Studio', 'Acme Marketing Google Analytics', 'Search Console Report Search Console', 'Google Ads Overview Google Ads', and 'YouTube Channel Report YouTube Analytics'. Below the template section are three filter dropdowns: 'Name', 'Owned by anyone', and 'Last opened by me'. In the center of the main content area is a large, semi-transparent icon of a bar chart. Below the icon is the text 'Create a Report' and the instruction 'Use the Create button to add one.'

Connect to multiple different types of data sources

The screenshot shows the Google Data Studio interface. At the top, there's a toolbar with options like File, View, Page, Help, Reset, Share, View, and a settings icon. Below the toolbar is a menu bar with Add page, Add data, Add a chart, Add a control, and Theme and layout. A large central area is a grid for adding data to a report, with a placeholder text 'Add data to report'. Below this is a modal window titled 'Add data to report'.

The modal has two tabs: 'Connect to data' (which is selected) and 'My data sources'. It includes a search bar labeled 'Search'. The main content area is titled 'Google Connectors (22)' and describes them as 'Connectors built and supported by Data Studio'. There are three rows of connectors:

- Google Analytics** By Google: Connect to Google Analytics.
- Google Ads** By Google: Connect to Google Ads performance report data.
- Google Sheets** By Google: Connect to Google Sheets.
- BigQuery** By Google: Connect to BigQuery tables and custom queries.

- File Upload** By Google: Connect to CSV (comma-separated values) files.
- Campaign Manager 360** By Google: Connect to Campaign Manager 360 data.
- Cloud Spanner** By Google: Connect to Google Cloud Spanner databases.
- Cloud SQL for MySQL** By Google: Connect to Google Cloud SQL for MySQL databases.

- Display & Video 360** By Google
- Extract Data** By Google
- Google Ad Manager 360** By Google
- Google Cloud Storage** By Google

Add the data source to your report

The screenshot shows a "Untitled Report" interface with a toolbar at the top. Below the toolbar is a main workspace with a grid background. A sidebar on the left lists various data sources, including "Google Sheets" by Google. A modal window is centered over the workspace, prompting the user to add data to the report. The modal title is "You are about to add data to this report". It displays the selected data source, "Natural_disasters_climate_change - Sheet1". A note states, "Note that Report Editors can create charts using the new data source(s), and can add dimensions and metrics not currently included in the report." There is a checkbox for "Don't show me this again" and two buttons at the bottom: "CANCEL" and "ADD TO REPORT", with "ADD TO REPORT" highlighted with a blue border.

Untitled Report

File View Page Help

Reset Share View

Add page Add data Add a chart Add a control Theme and layout

Add data to report Data credentials: Steve Leonard

Google Sheets By Google

The Google Sheets connector allows you to access data stored in a Google Sheets worksheet.

LEARN MORE REPORT AN ISSUE

ALL ITEMS Spreadsheet Worksheet

OWNED BY ME 32. From Data to Insights with Google Clou... Natural_disasters_climate_change

SHARED WITH ME 16. Data Engineering on Google Cloud Mai... Consolidated Control Log

STARRED 29. Managing Google Cloud's Apigee API P... T-GCPNET-1 Networking in Google Cloud v1...

URL Networking in Google Cloud v1.2.5 - Maint... 17. Networking in Google Cloud Maintenan...

OPEN FROM GOOGLE DRIVE Course Map - Security in Google Cloud v2... 24. Security in Google Cloud Maintenance ...

31. End-to-End ML with Tensorflow on Goo... 2. Analyzing and Visualizing Data in Looker...

10. Google Cloud Fundamentals for AWS a... Courses Monitored & Maintained by SureS...

You are about to add data to this report

Natural_disasters_climate_change - Sheet1

Note that Report Editors can create charts using the new data source(s), and can add dimensions and metrics not currently included in the report.

Don't show me this again

CANCEL ADD TO REPORT

Optional Range, e.g. A1:BS2

Cancel Add

Select your data fields to build your visualizations

Untitled Report

File Edit View Insert Page Arrange Resource Help

Reset Share View

Add page Add data Add a chart Add a control Theme and layout

	Year	Earthquake	Epidemic	Storm	Wildfire	Volcanic ...	Insect inf...	Extreme t...	Landslide	Mass mo...	Flood	Drought
1.	2018	20	15	94	10	7	1	26	13	1	127	14
2.	2017	22	27	130	15	2	1	10	25	1	126	9
3.	2016	30	25	86	10	null	1	12	13	null	159	14
4.	2015	23	16	121	12	6	1	12	20	1	162	28
5.	2014	26	21	99	4	6	1	17	15	null	135	18
6.	2013	29	23	105	10	3	1	14	11	1	149	9
7.	2012	27	25	91	6	1	1	51	13	1	136	21
8.	2011	30	27	84	8	6	1	16	17	null	156	17
											1 - 30 / 30	< >

Chart > Table

DATA **STYLE**

Data source: Natural_disasters...

Available Fields: Drought, Earthquake, Epidemic, Extreme temperature, Flood, Insect infestation, Landslide, Mass movement (dry), Record Count, Storm, Volcanic activity, Wildfire, Year

Date Range Dimension: Year

Dimension: Year

Metric: Earthquake, Epidemic, Storm, Wildfire, Volcanic activity, Insect infestation, Extreme temperature, Landslide, Mass movement (dry), Flood, Drought

Drill down: Off

Add dimension: Add dimension

Add metric: Add metric

01 Available Fields

02 Dimensions

03 Metrics

Edit your data source fields, if necessary

Screenshot of a data visualization tool interface showing a table of natural disaster data and its configuration panel.

Table Data:

	Year	Earthquake	Epidemic	Storm	Wildfire	Volcanic ...	Insect inf...	Extreme t...	Landslide	Mass mo...	Flood	Drought
1.	2018	20	15	94	10	7	0	26	13	1	127	14
2.	2017	22	27	130	15	2	0	10	25	1	126	9
3.	2016	30	25	86	10	null	0	12	13	null	159	14
4.	2015	23	16	121	12	6	0	12	20	1	162	28
5.	2014	26	21	99	4	6	0	17	15	null	135	18
6.	2013	29	23	105	10	3	0	14	11	1	149	9
7.	2012	27	25	91	6	1	0	51	13	1	136	21

Data Source Configuration:

- Data source:** Natural_disasters...
- Available Fields:**
 - Drought
 - Earthquake
 - Epidemic
 - Add dimension
 - Add a field
 - Add a parameter

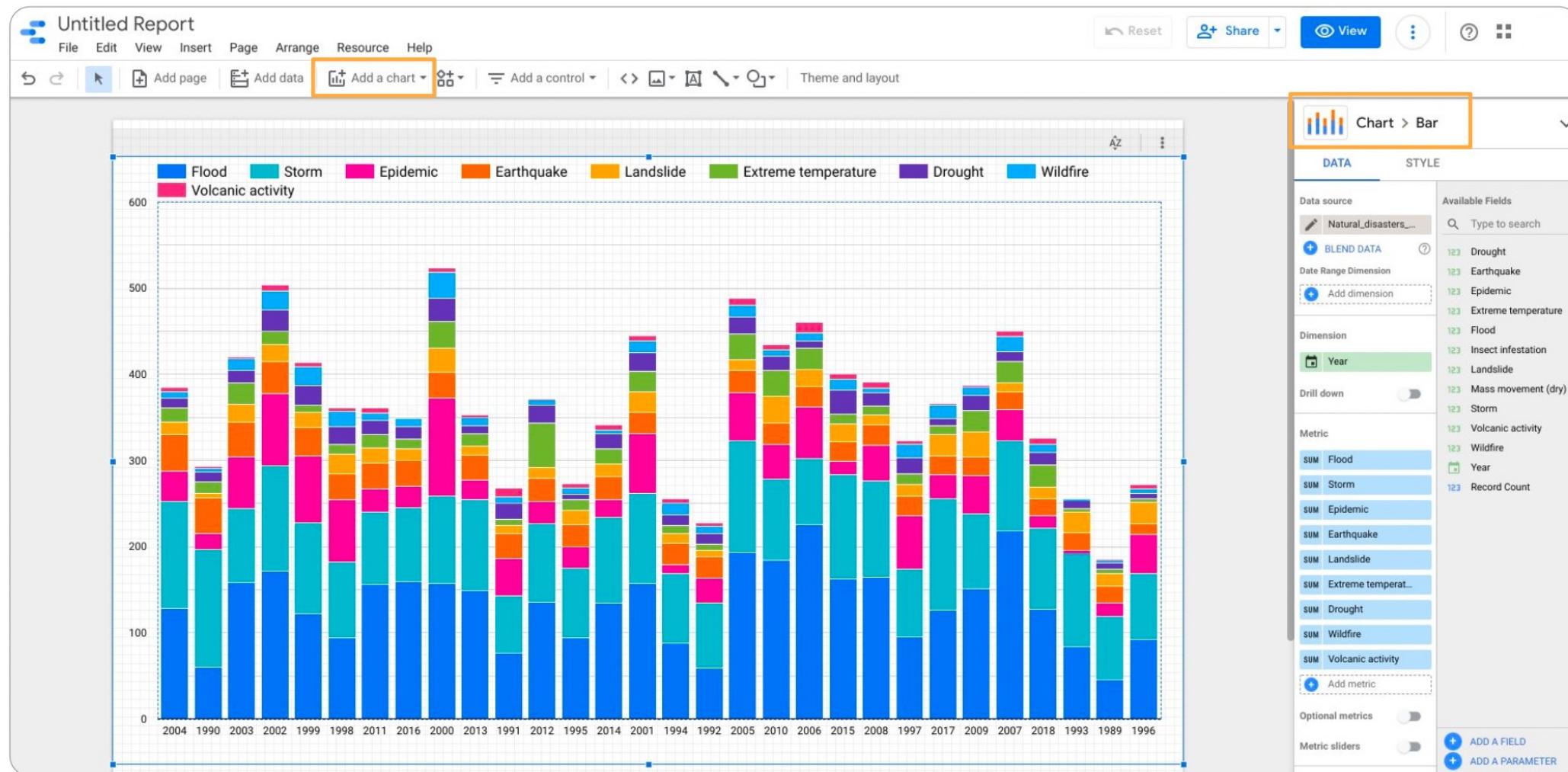
Fields Panel:

Field	Type	Default Aggregation	Description
Insect infestation	123 Number	Sum	
Landslide	123 Number	Sum	
Mass movement (dry)	123 Number	Sum	
Storm	123 Number	Sum	
Volcanic activity	123 Number	Sum	
Wildfire	123 Number	Sum	
Year	Year (YYYY)	None	

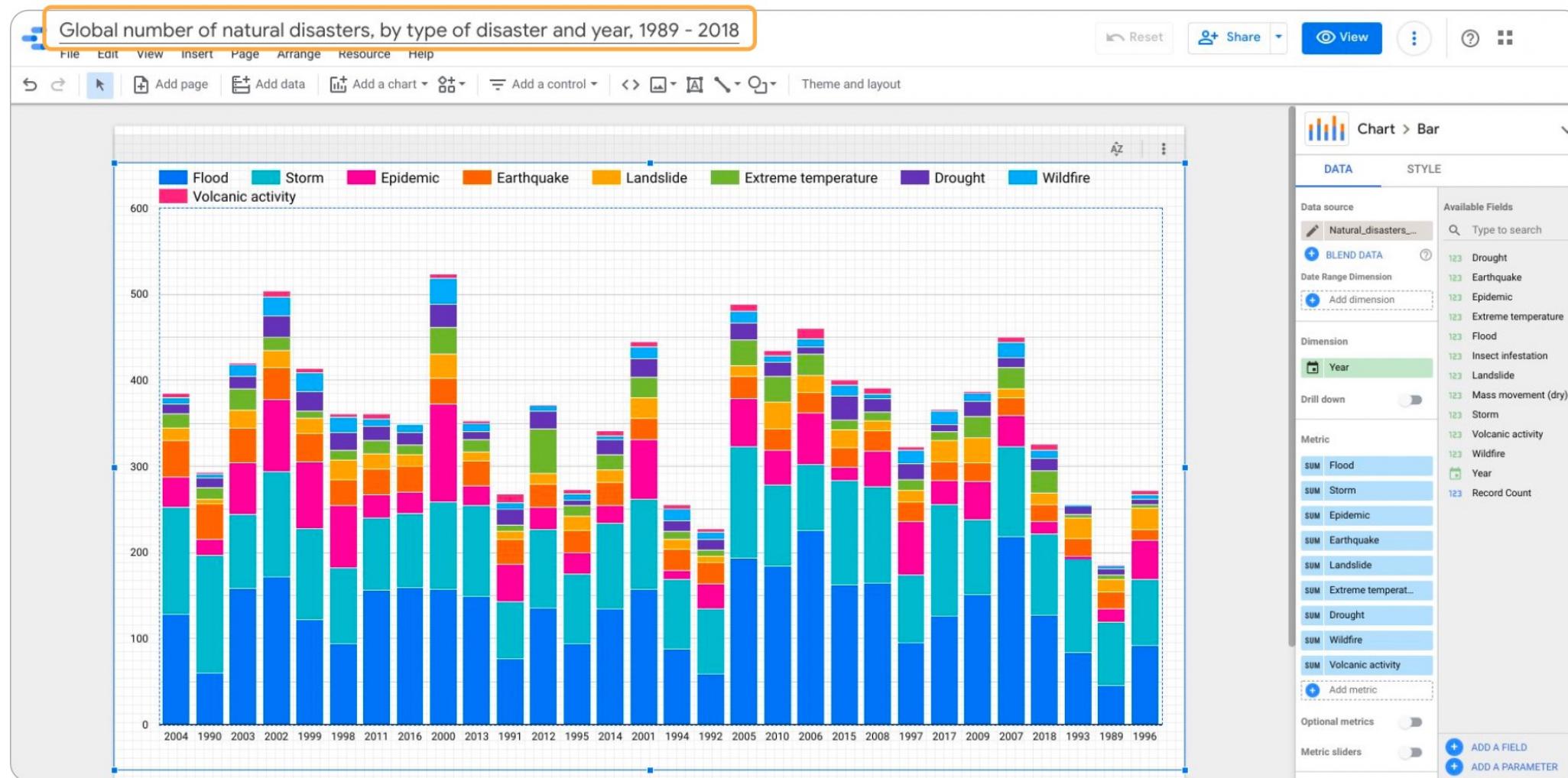
METRICS (1): Record Count

Bottom Navigation: REFRESH FIELDS, 13 / 13 Fields, DONE

Create charts to visualize data relationships



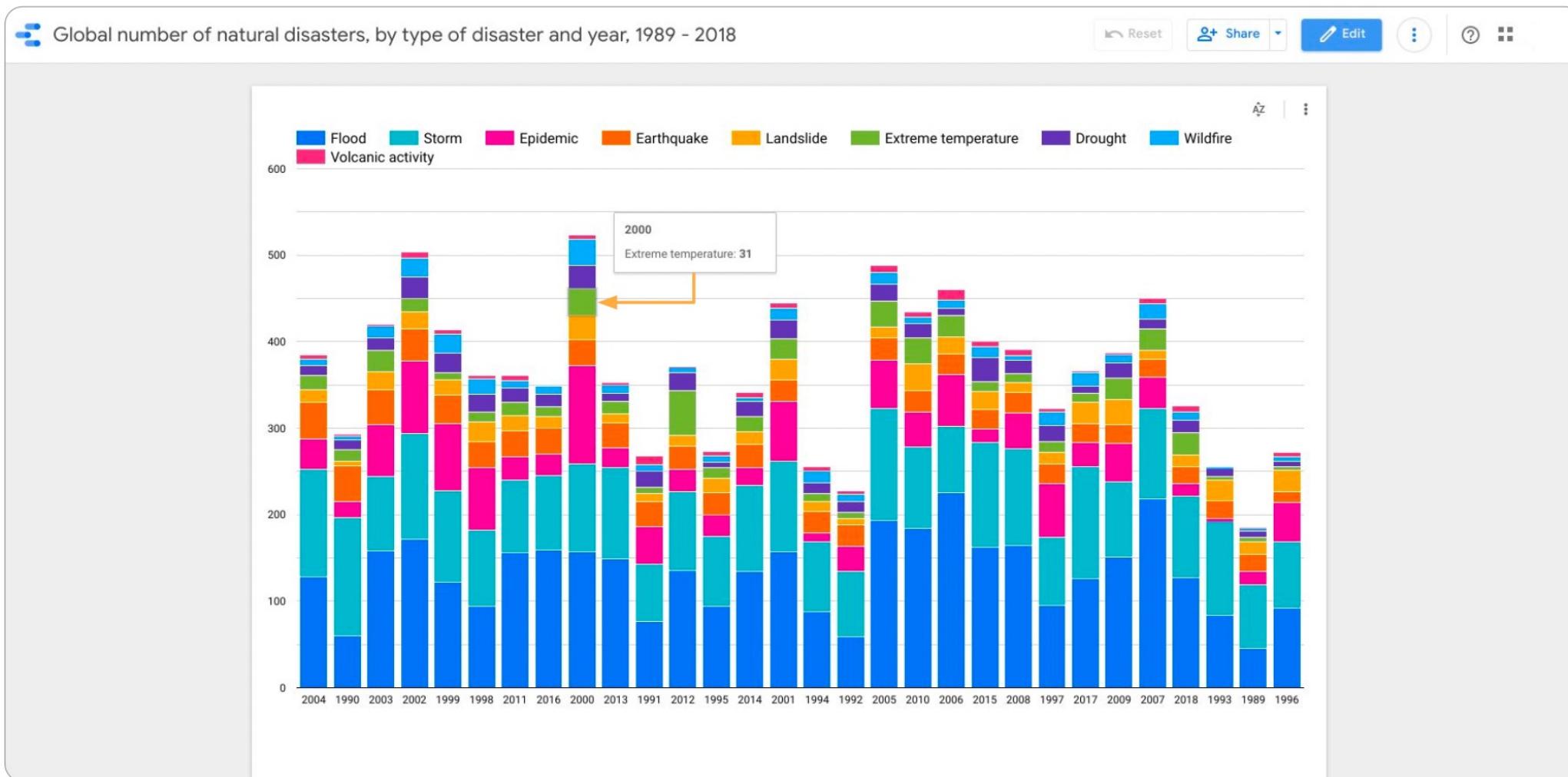
Add a descriptive name to your report



View the end-user version of the report

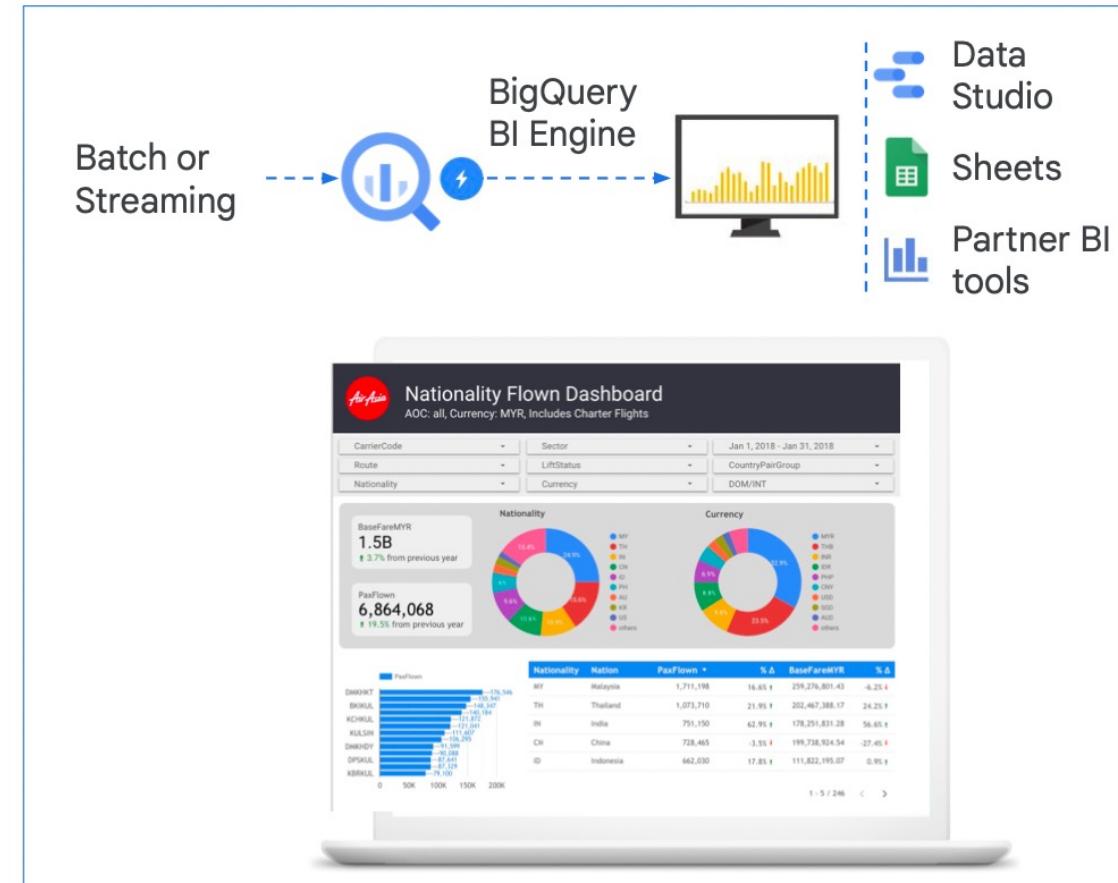


View your report as an end-user



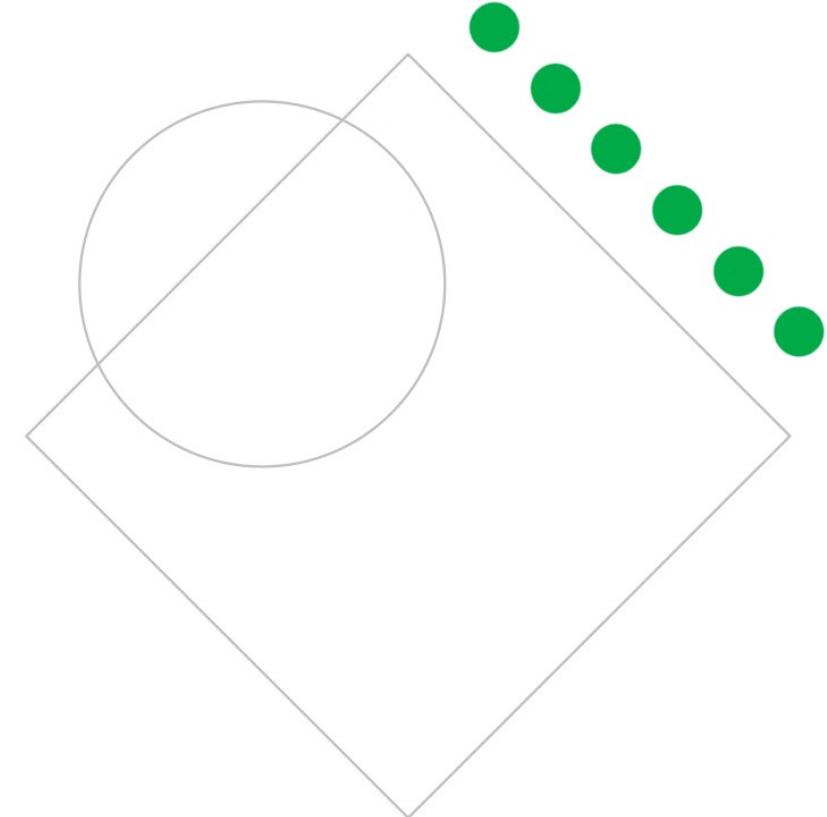
Add value: BI Engine for dashboard performance

- No need to manage OLAP cubes or separate BI servers for dashboard performance.
- Natively integrates with BigQuery streaming for real-time data refresh.
- Column oriented in-memory BI execution engine.



Lab Intro

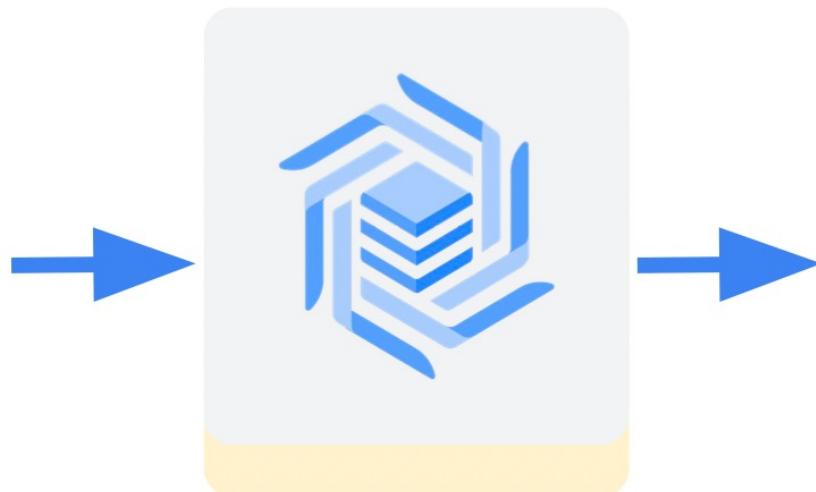
Streaming Data Processing:
Streaming Analytics and Dashboards





High-Throughput Streaming with Cloud Bigtable

Cloud Bigtable



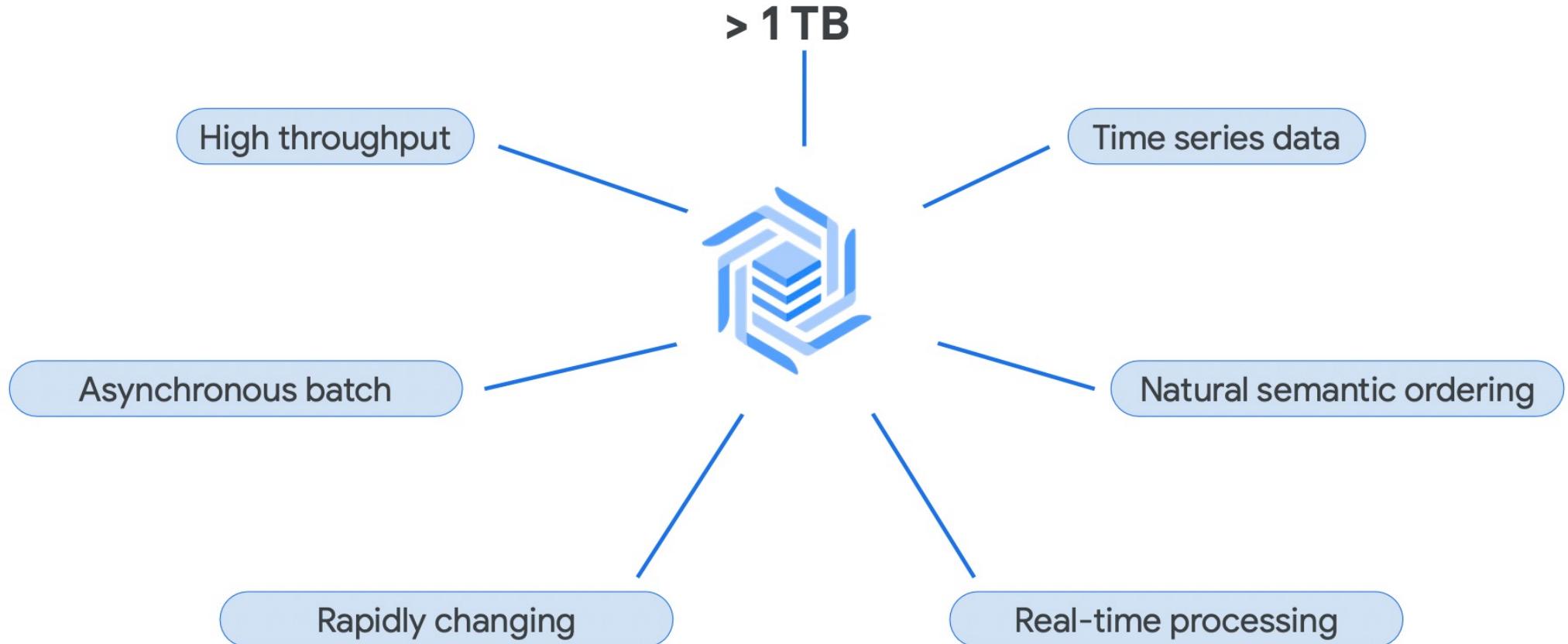
Qualities that Bigtable contributes to data engineering solutions:

- NoSQL Queries over large datasets (petabytes) in milliseconds
- Very fast for specific cases

How to choose between Bigtable and BigQuery



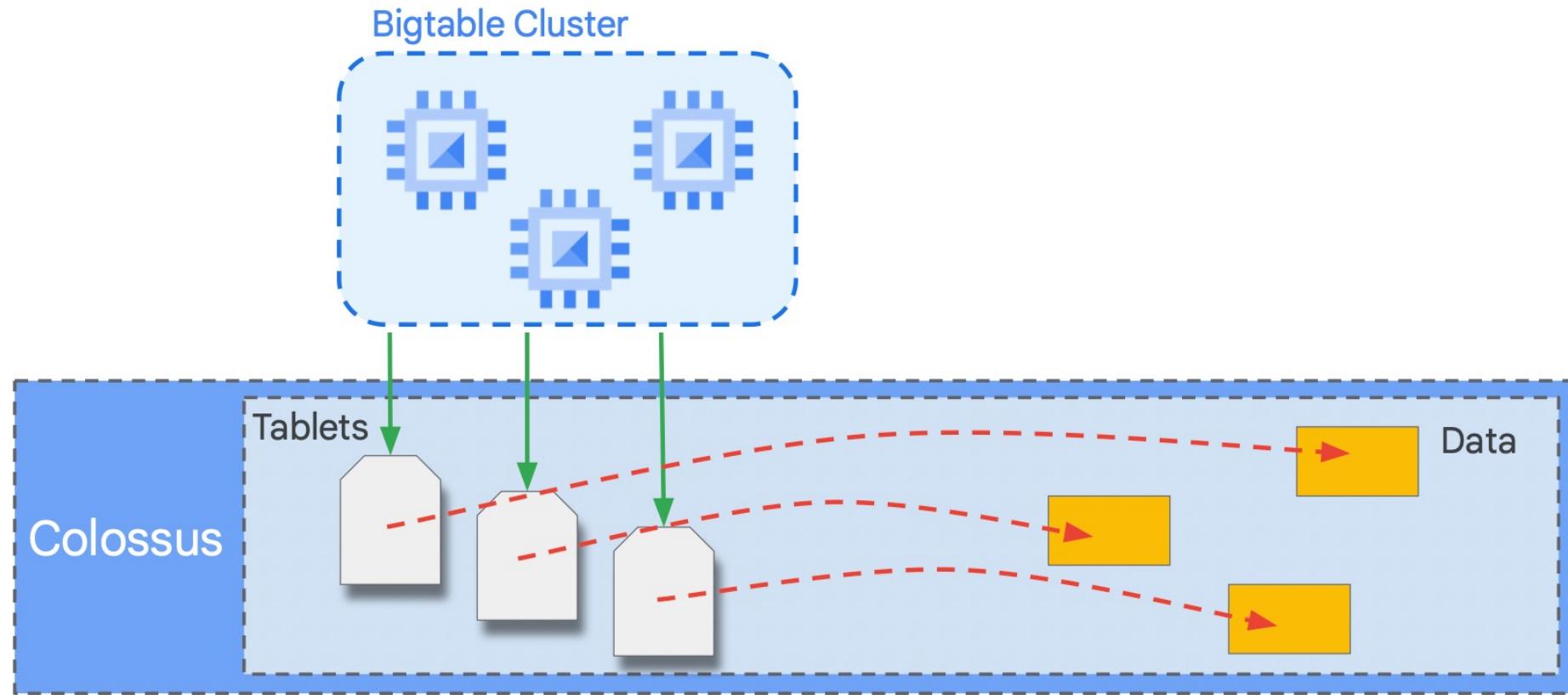
Consider Bigtable for these requirements



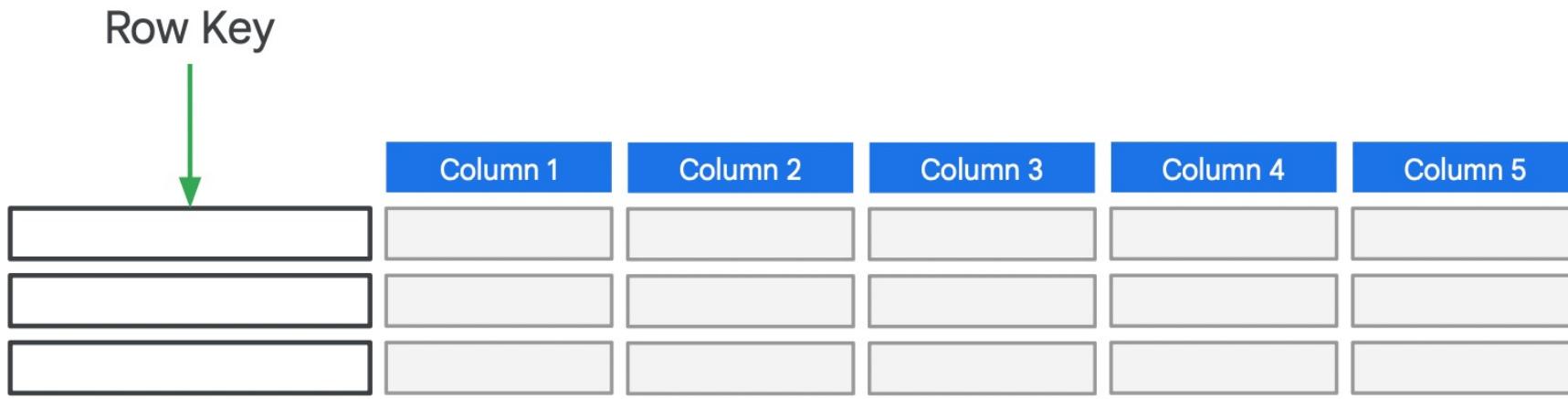
The most common use of Bigtable

Productionize a real-time lookup as part of an application, where speed and efficiency are desired beyond that of other databases.

How does Bigtable work?

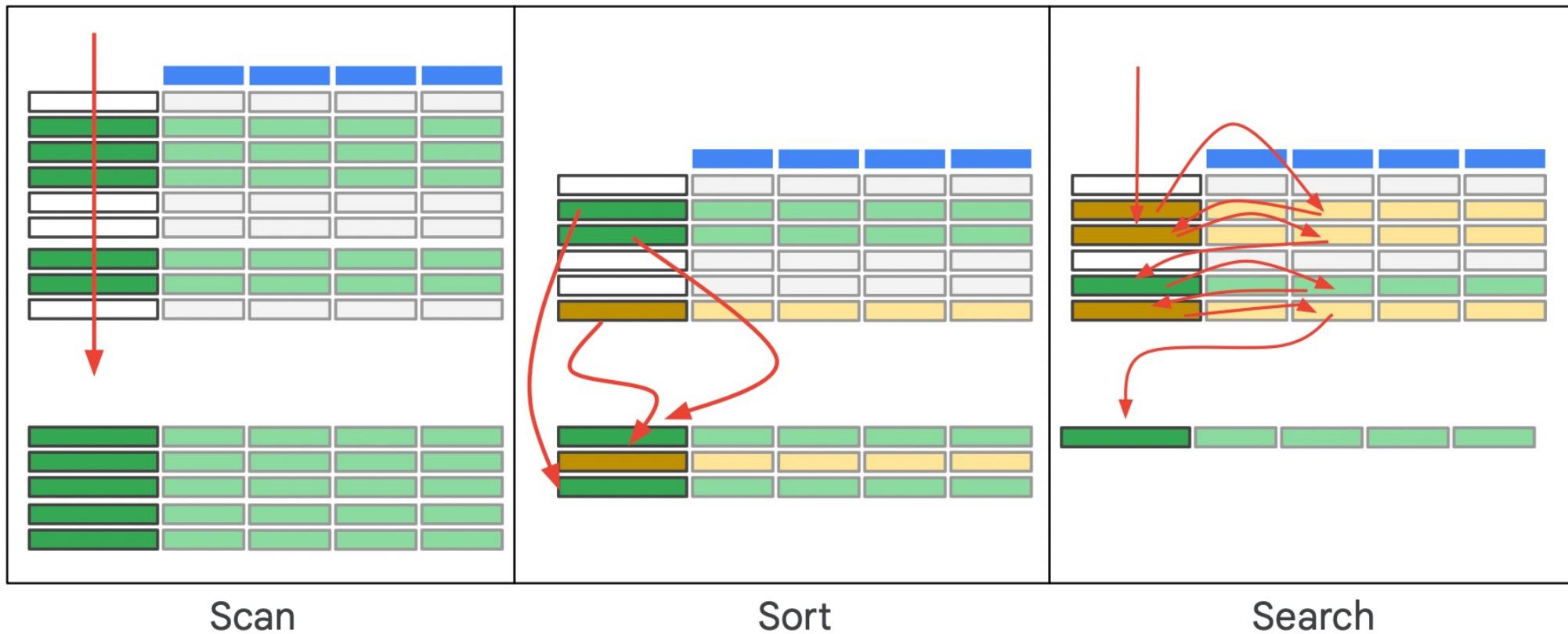


Bigtable design idea is "simplify for speed"



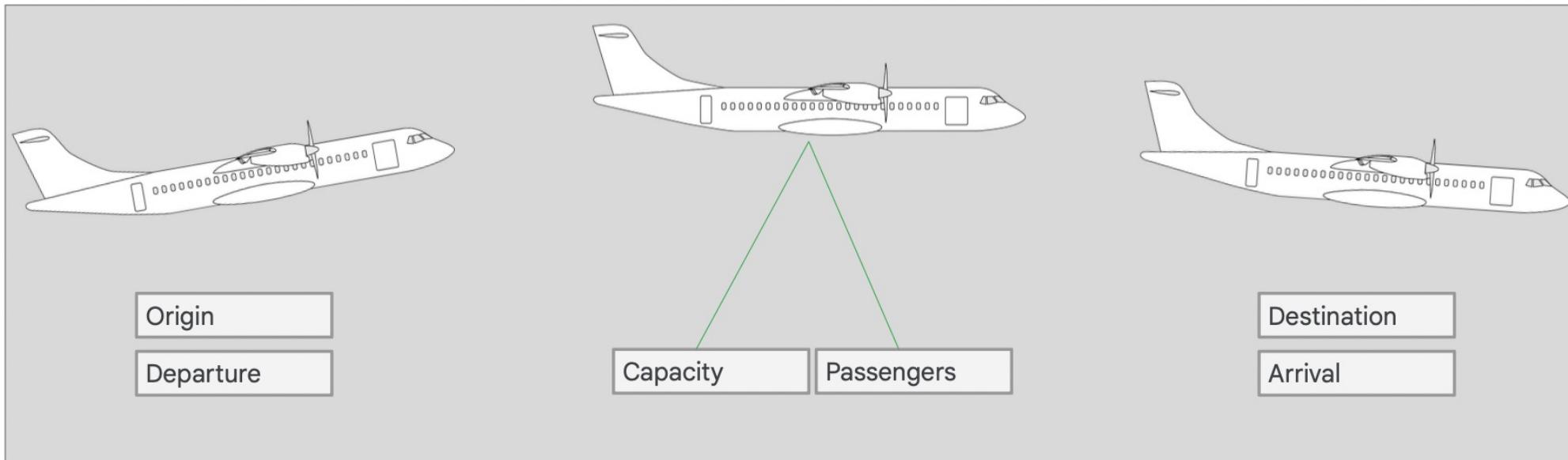
The Row Key is the index.
And you get only one.

But speed depends on your data and Row Key



Flights of the world: Reviewing the data

Make
Model
Age



What is the best Row Key?

Query: All flights originating in Atlanta and arriving between March 21st and 29th

Origin	Arrival	Remaining columns...
ATL	20190321-1005	

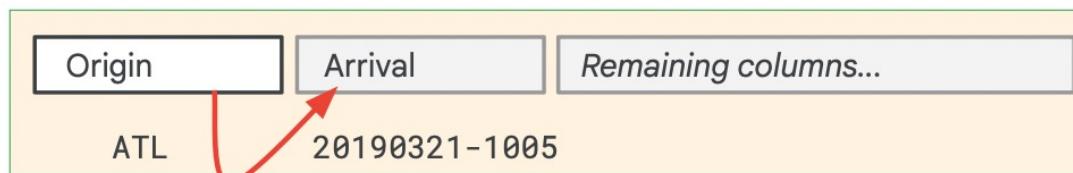
Sort or Search

Arrival	Origin	Remaining columns...
20190321-1005	ATL	

Sort or Search

What is the best Row Key?

Query: All flights originating in Atlanta and arriving between March 21st and 29th



Sort or Search

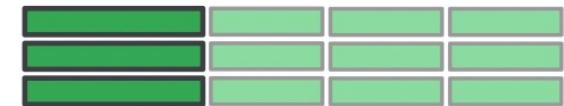
Sort or Search



ATL#arrival#20190321-1005

Constructed Row Key

Scan



Bigtable schema organization



Column Families

Row Key	Flight_Information					Aircraft_Information			
	Origin	Destination	Departure	Arrival	Passengers	Capacity	Make	Model	Age
ATL#arrival#20190321-1121	ATL	LON	20190321-0311	20190321-1121	158	162	B	737	18
ATL#arrival#20190321-1201	ATL	MEX	20190321-0821	20190321-1201	187	189	B	737	8
ATL#arrival#20190321-1716	ATL	YVR	20190321-1014	20190321-1716	201	259	B	757	23

Queries that use the row key, a row prefix, or a row range are the most efficient

Query: Current arrival delay for flights from Atlanta

1

row key based on atlanta arrivals
e.g. ORIGIN#arrival

(ATL#arrival#20190321-1005)

Puts latest flights at bottom
of table

2

reverse timestamp to the rowkey
e.g. ORIGIN#arrival#RTS

(ATL#arrival#560549313)

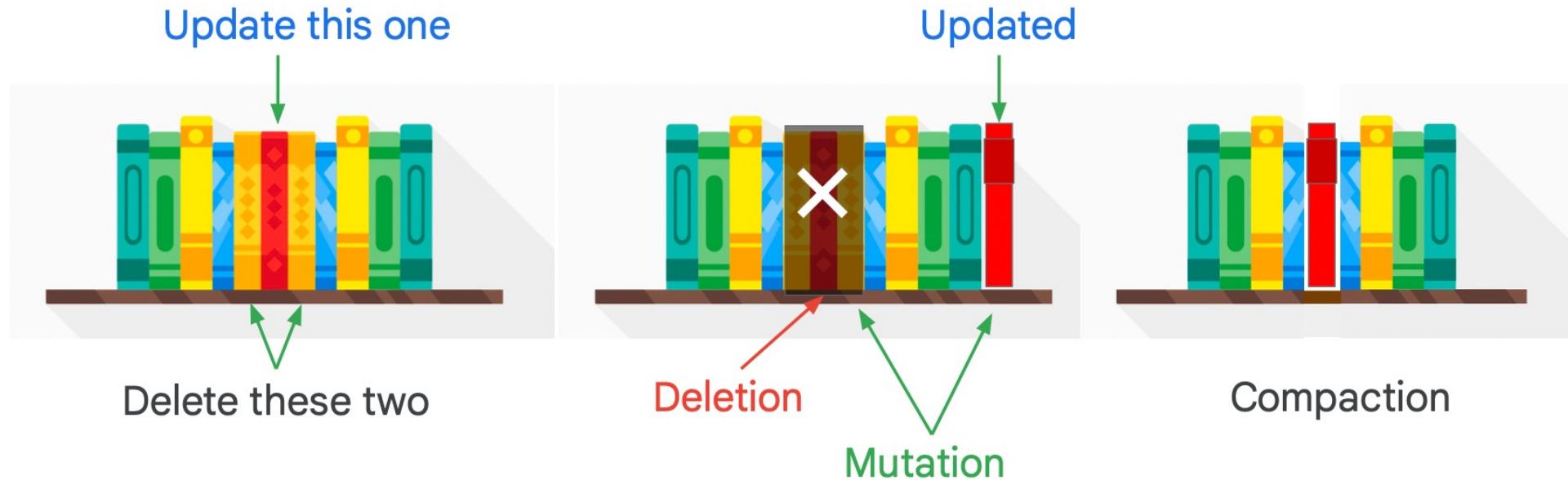
Puts latest flights at top
of table

Use reverse timestamps when your most common query is for the latest values

Query: Current arrival delay for flights from Atlanta

```
// key is ORIGIN#arrival#REVTS
String key = info.getORIGIN() //
+ "#arrival" //
+ "#" + (Long.MAX_VALUE - ts.getMillis()); // reverse timestamp
```

What happens when data in Bigtable is changed?



Optimizing data organization for performance



Group related data for more efficient reads

Example row key:

DehliIndia#2019031411841

Use column families



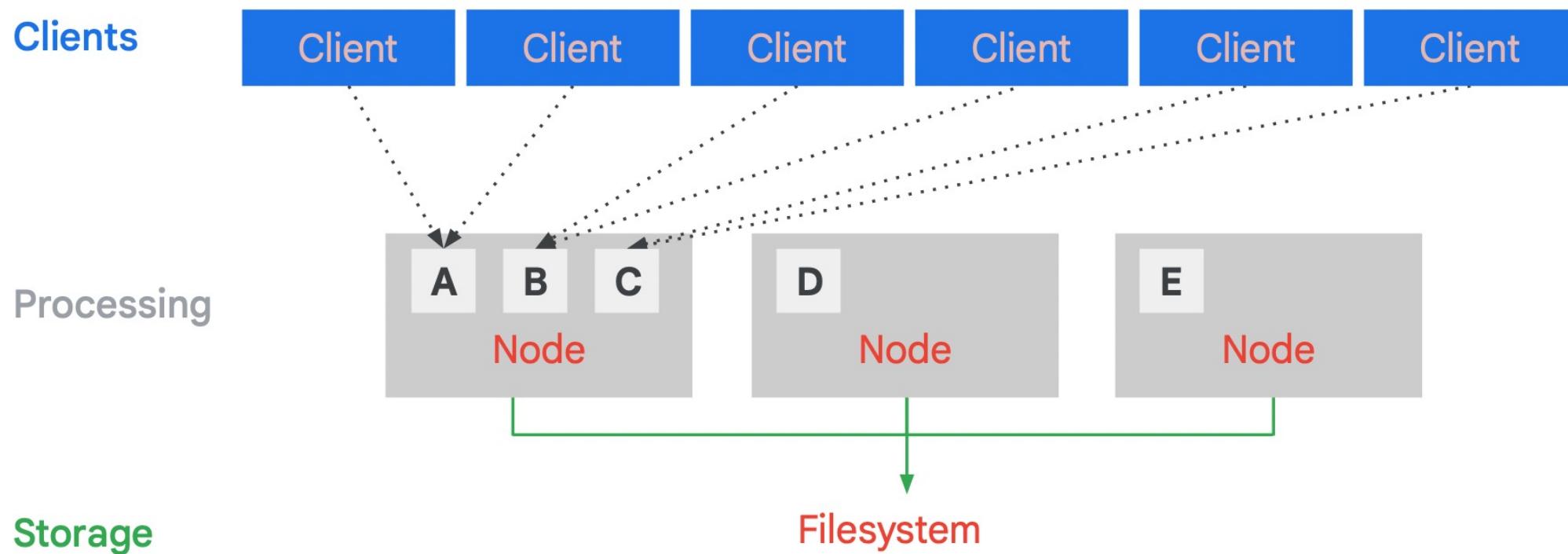
Distribute data evenly for more efficient writes



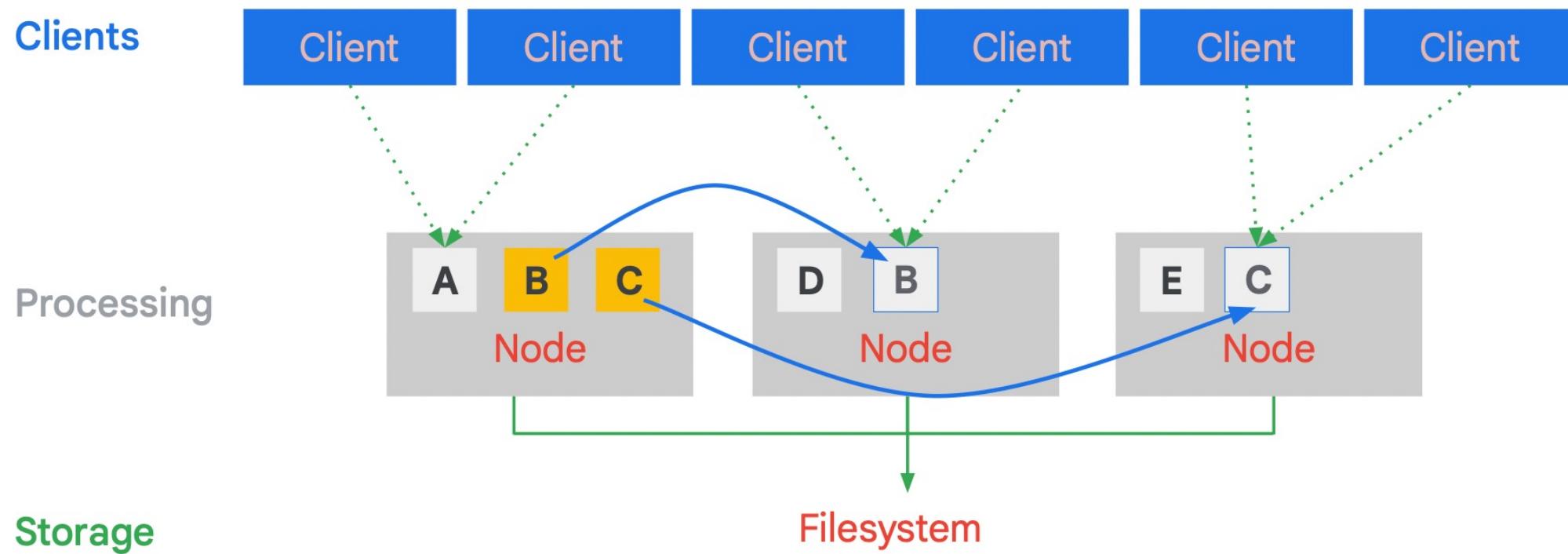
Place identical values in the same row or adjoining rows for more efficient compression

Use row keys to organize identical data

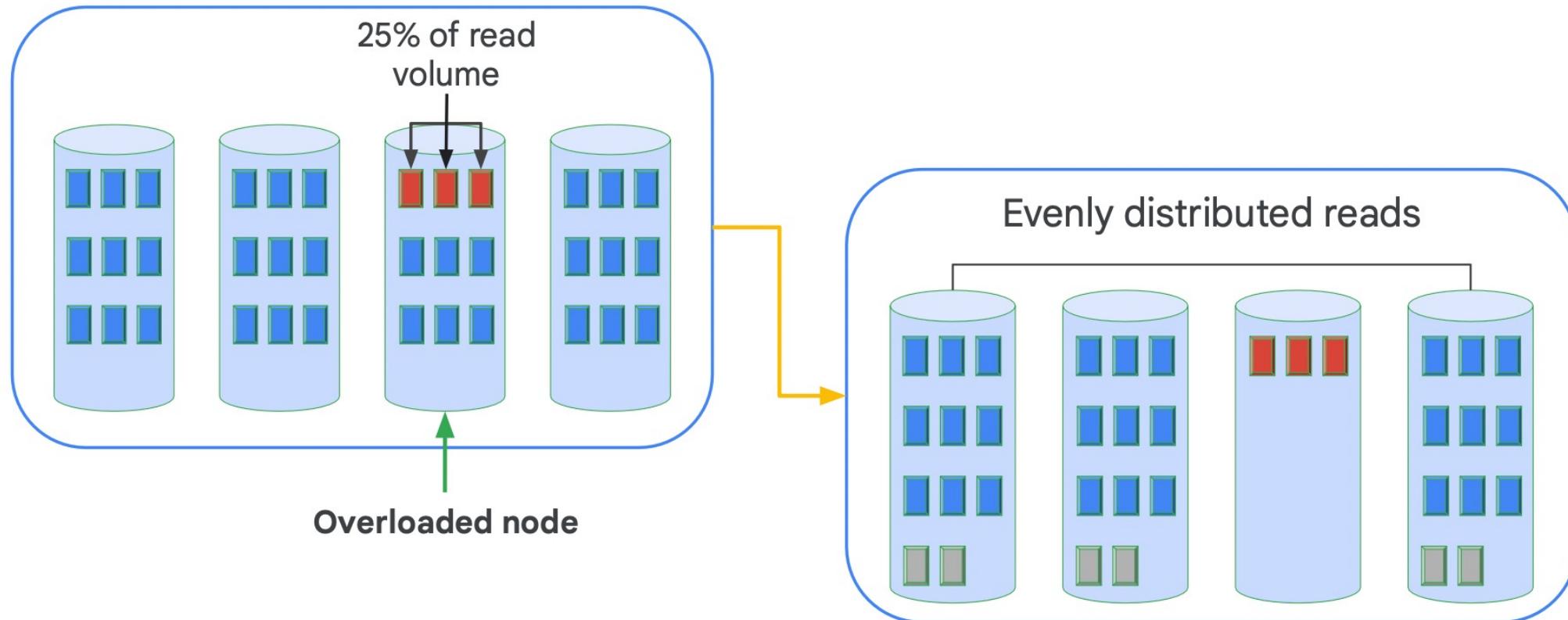
Bigtable self-improves by learning access patterns ...



...and rebalances data accordingly



Rebalance strategy: Distribute reads



Real world use case: Spotify



In 2019, Spotify ran the largest Dataflow job ever at the time with Bigtable "...used as a remediation tool between Dataflow jobs in order for them to process and store more data in a parallel way, rather than the need to always regroup the data"





Optimizing Cloud Bigtable Performance

Optimizing Bigtable performance

Tune the schema

Cloud Bigtable learning behavior

Tune the resources

Change schema to minimize data skew

Takes a while after scaling up nodes for performance improvement to be seen

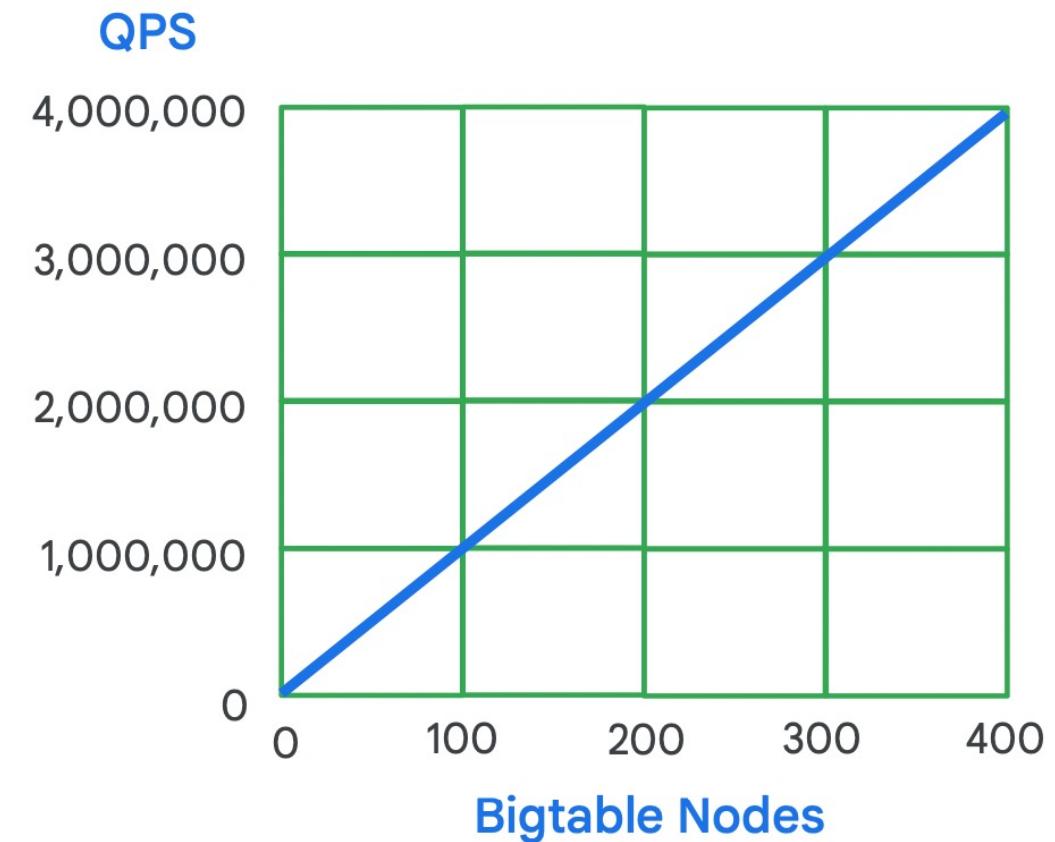
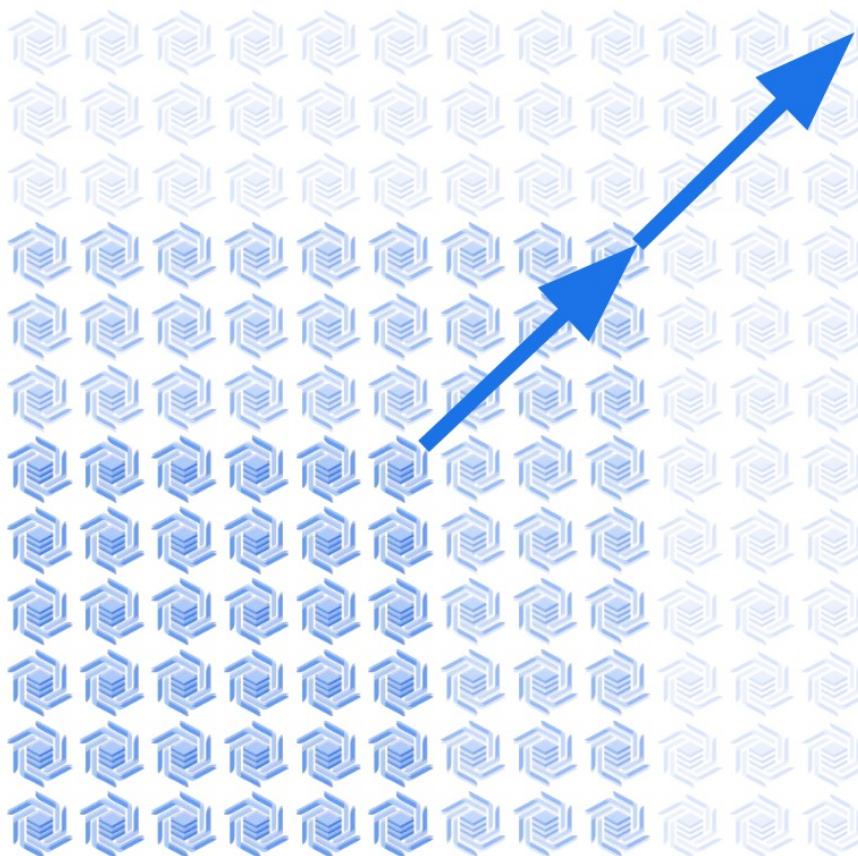
Test with > 300 GB and for minutes-to-hours to give time for Bigtable to balance and learn

Make sure clients and Cloud Bigtable are in **same zone**

Disk speed on VMs in the cluster: SSD is faster than HDD

Performance increases linearly with **number of nodes**

Throughput can be controlled by node count

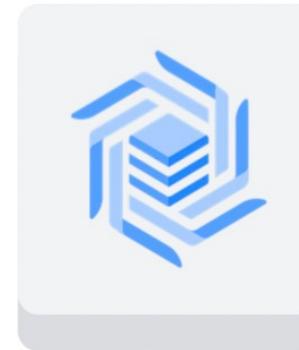


Features for Bigtable streaming

Incoming streaming
data is independent



Writing

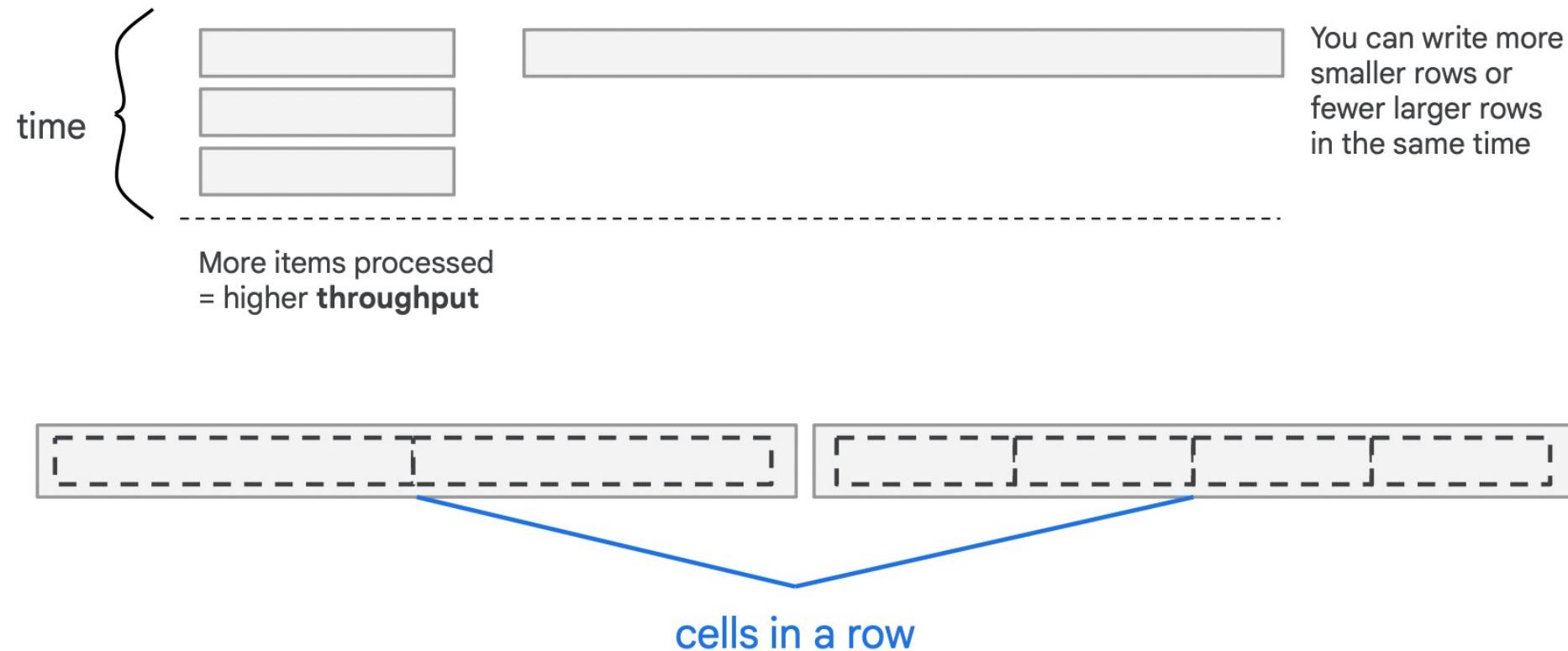


Application reading of
data is controllable



Reading

Schema design is the primary control for streaming



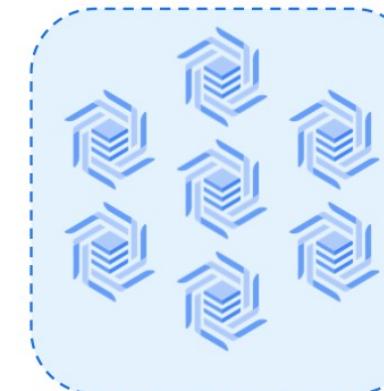
Use Bigtable replications to improve availability

Why perform replication?

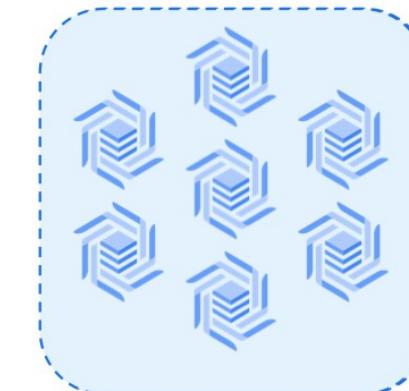
- Isolate serving applications from batch reads
- Improve availability
- Provide near-real-time backup
- Ensure your data has a global presence

```
gcloud bigtable clusters create CLUSTER_ID \
    --instance=INSTANCE_ID \
    --zone=ZONE \
    [--num-nodes=NUM_NODES] \
    [--storage-type=STORAGE_TYPE]
```

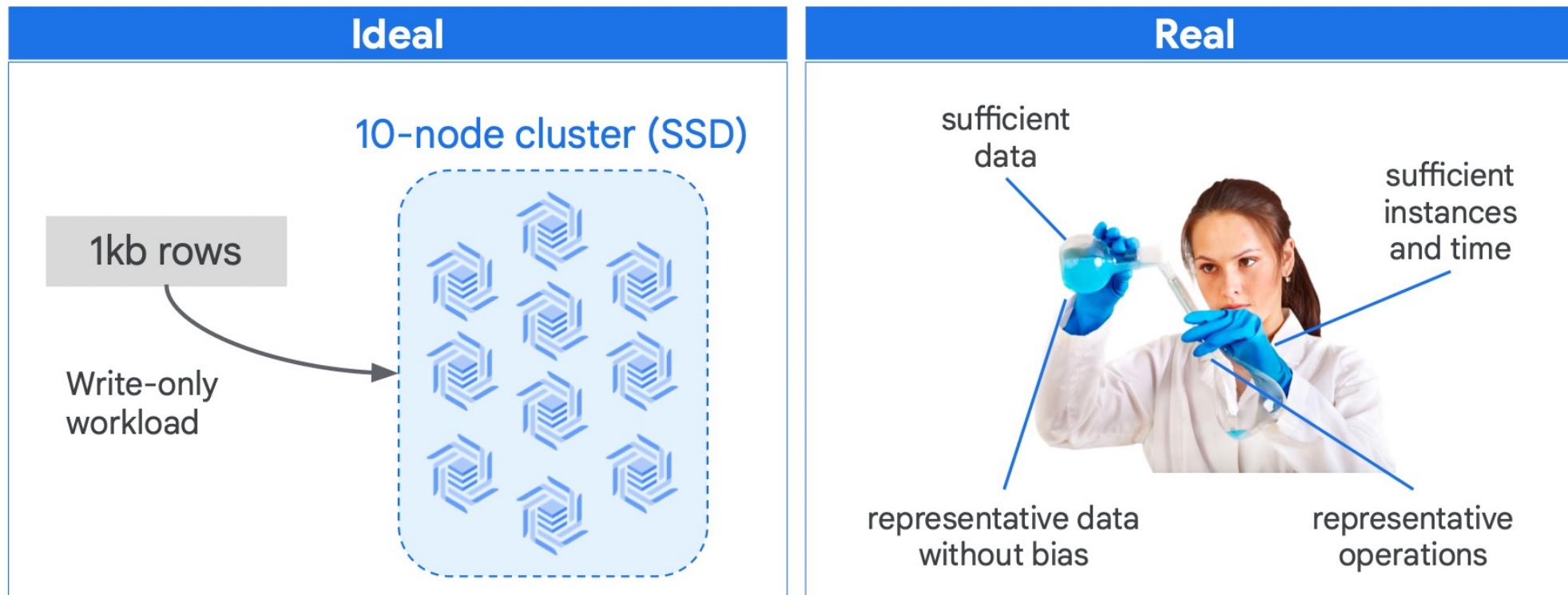
Batch analytic
read-only Cluster



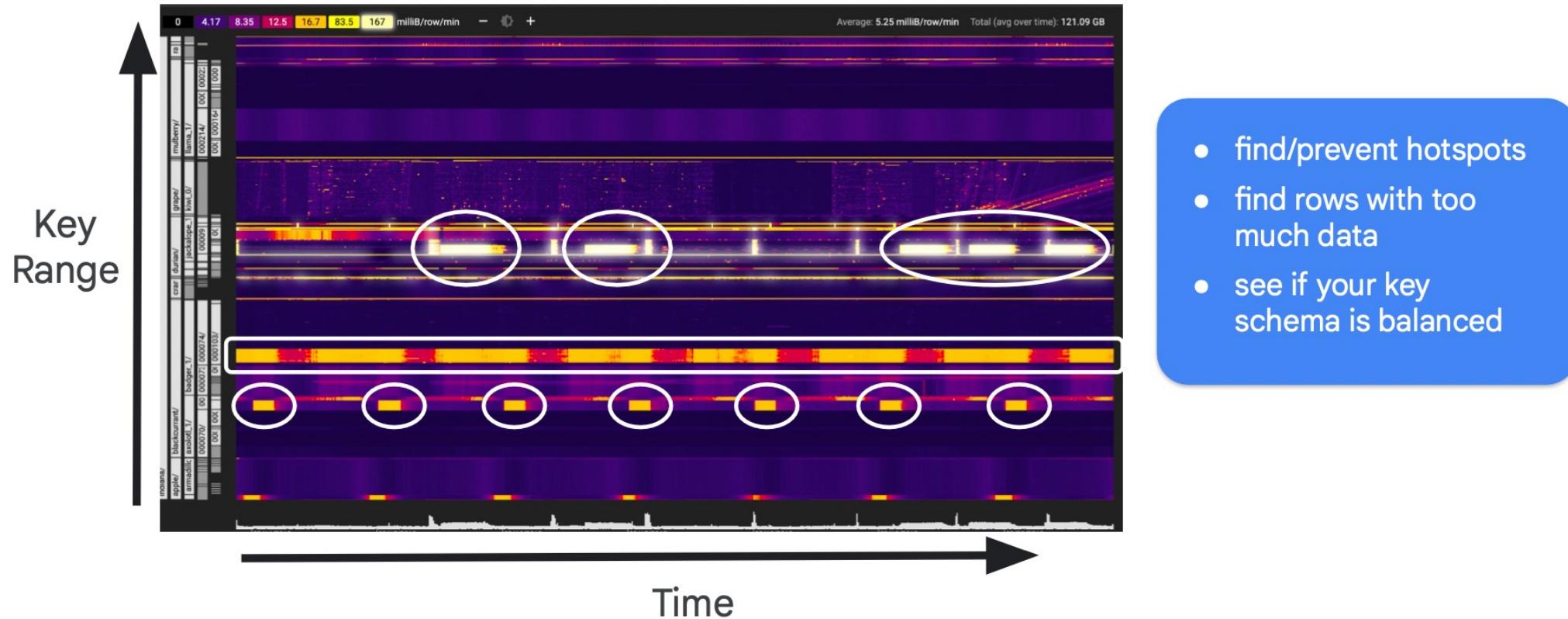
App Traffic Cluster



Run performance tests carefully for Bigtable streaming



Key Visualizer exposes read/write access patterns over time and key space



Lab Intro

Streaming Data Processing: Streaming
Data Pipelines into Bigtable

