

BEAM
SUMMIT

How to write an IO for Beam

John Casey

BEAM
SUMMIT

Beam IO: CDAP and SparkReceiver IO Connectors Overview

Alex Kosolapov
& Elizaveta Lomteva

Agenda

- Introduction
- Developing an IO
- CDAP IO Overview
- Streaming Source IO – SparkReceiver
- Testing IO
- Akvelon Data Analytics and ML Accelerators demo

AKVELON

1200+
technology experts

23+
years of expertise

150+
clients

15 offices
in 11 countries

24/7
operations support





AKVELON



Google Cloud
Partner

Developing Beam IO (Java)

- Starting point: [Developing a new I/O connector](#)
- Design:
 - Define the input/output format
 - Read – Splittable DoFn (SDF), Write – ParDo
 - Determine target pipeline configuration parameters
- Develop:
 - DoFn to process an element
 - Read/Write PTransforms
- Test IO:
 - Unit testing, Integration, Performance testing
- Release: IO Documentation and examples

CDAP cdap.io

An open-source platform for data applications in hybrid and multi-cloud environments

 Google Cloud Data Fusion

Visual point-and-click interface enabling code-free deployment of ETL/ELT data pipelines

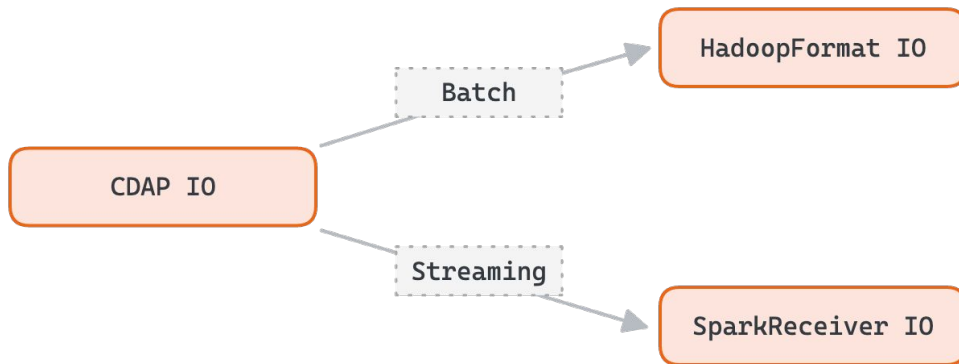
Ecosystem of plugins, including business applications connectors

CDAP IO

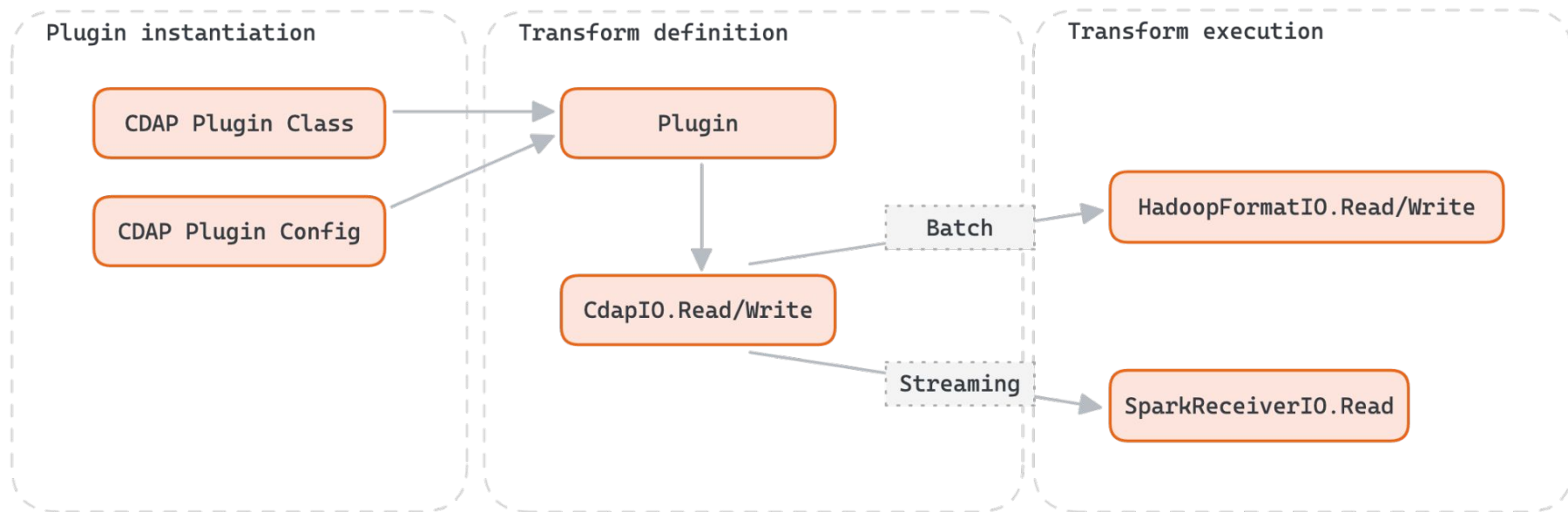
Provides transforms for reading and writing data via CDAP plugins

Connects Apache Beam with a variety of business applications like Salesforce, Hubspot, ServiceNow and Zendesk

Uses CDAP plugin definition



CDAP IO Workflow



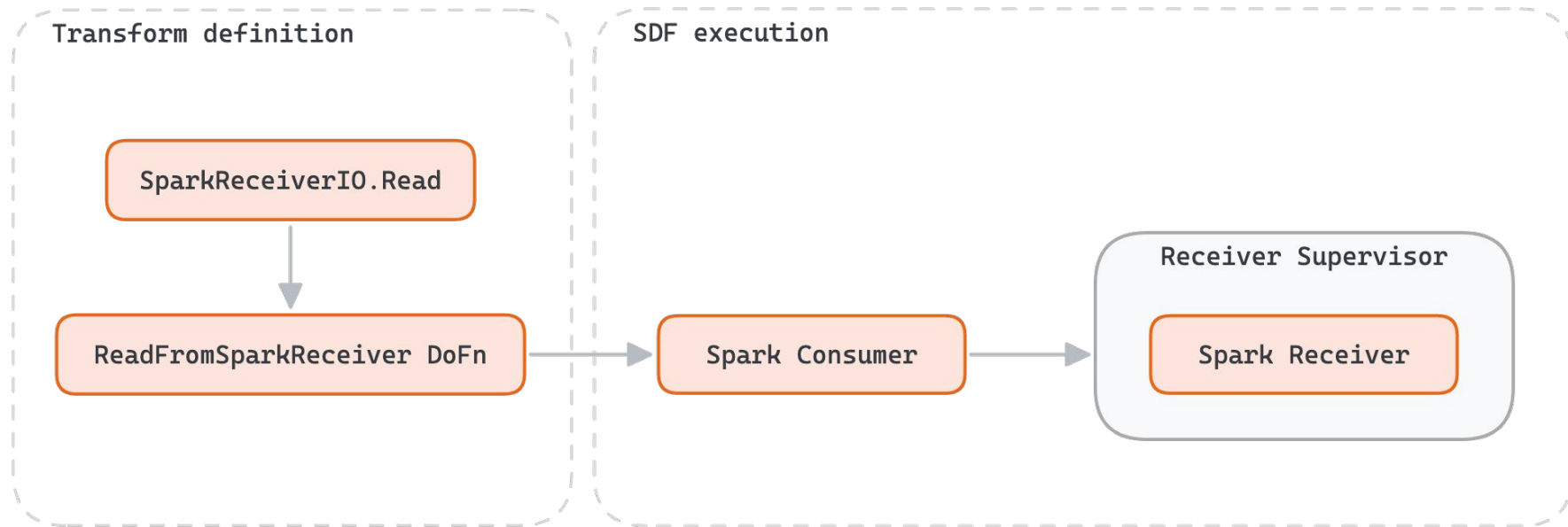
SparkReceiver IO

SparkReceiverIO provides transforms to read data via Apache Spark Receiver

Prerequisites:

- Spark Receiver provides HasOffset interface.
- Records have a numeric field that represents record offset.

SparkReceiver IO Workflow



Beam Parallelism & IO

Input parallelism – reading from bounded and unbounded sources, i.e. data source parallelism

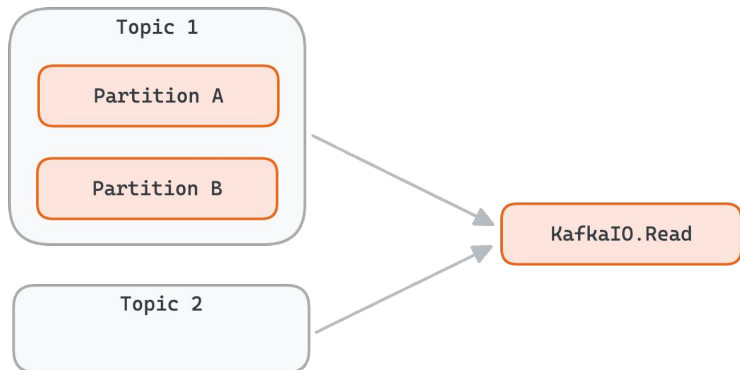
Inter-stage parallelism – splitting processing across workers, e.g. key-based data partitioning

Intra-stage parallelism – splitting element processing within transforms, e.g. Splittable DoFns, bundle processing

Data Source Parallelism

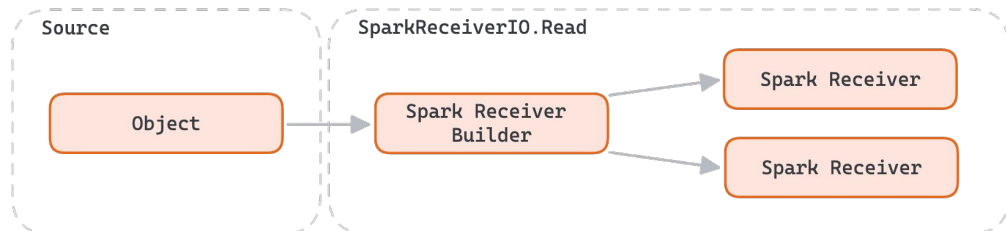
Refers to the parallelism achieved by reading data from multiple sources or partitions of a single source concurrently.

(E.g. Kafka topic partitions)



SparkReceiverIO

Each receiver builder can be associated with single source object and create multiple receivers during processing



Inter-stage parallelism

Refers to the parallelism between different transforms (or stages) within a Beam pipeline.

Achieved by runner implementation

(E.g. key-based operations in Beam)

SparkReceiverIO

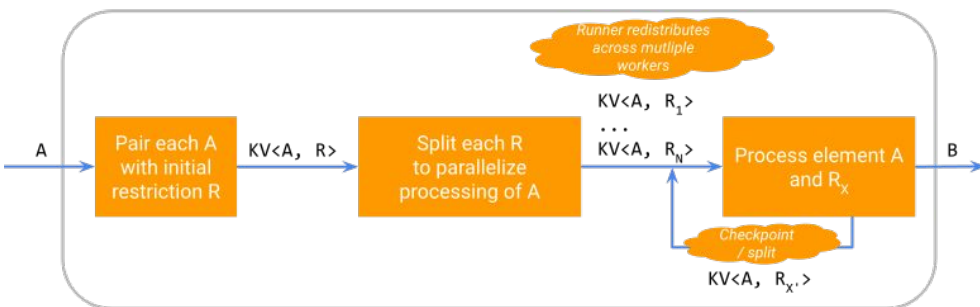
Achieved by supported runners – Direct runner and Dataflow runner v1 and v2



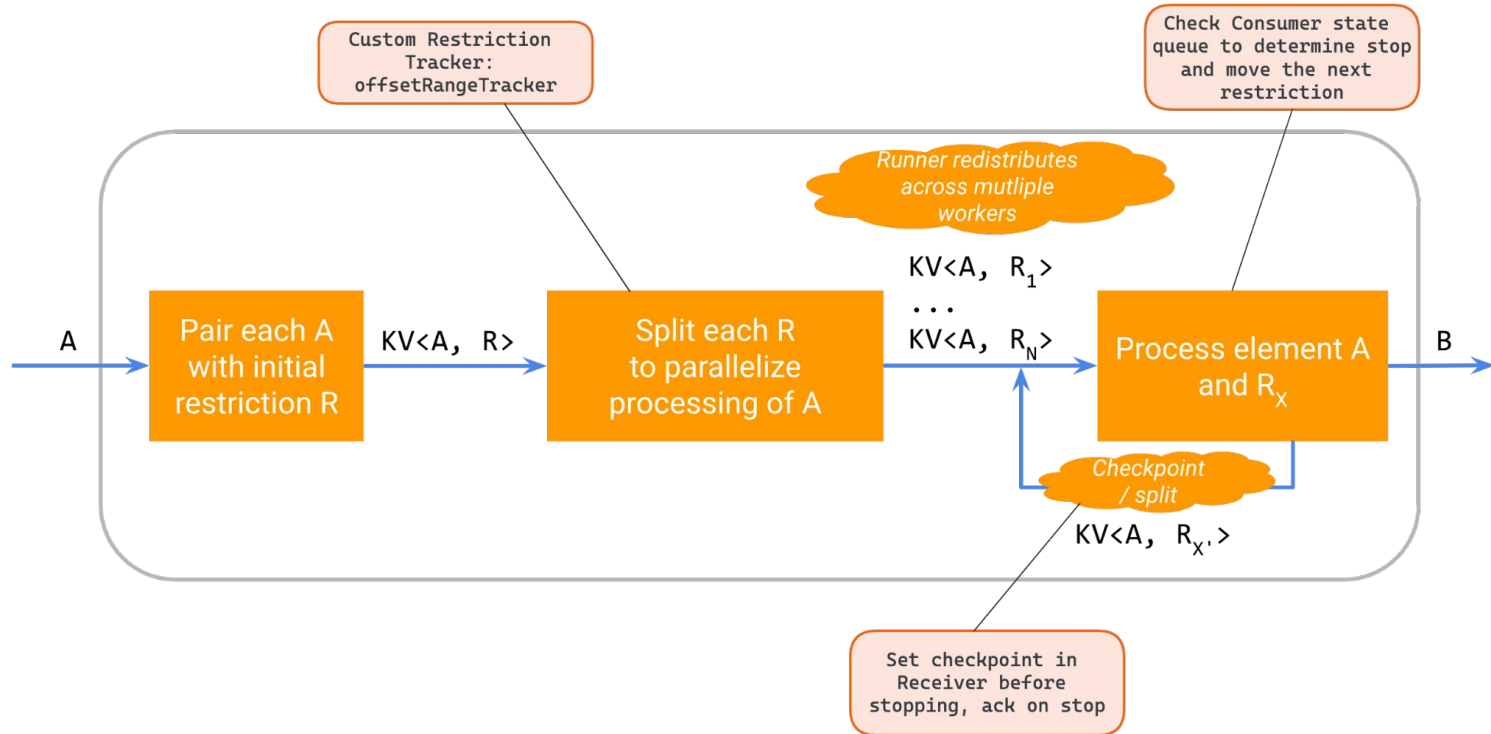
Intra-stage: Splittable DoFn (SDF)

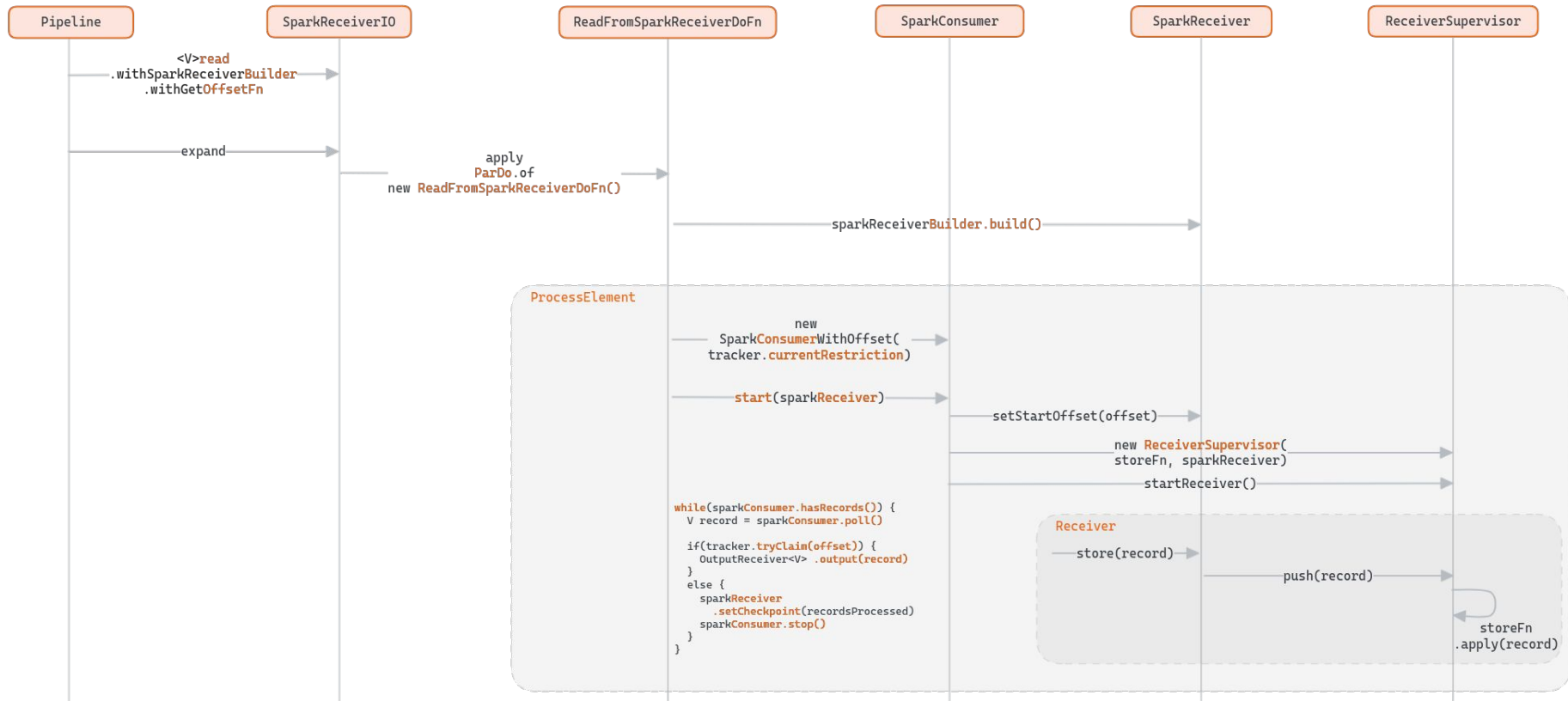
Executing an SDF follows the following steps:

1. Each **element** is paired with a **restriction** (e.g. filename is paired with offset range representing the whole file).
2. Each element and restriction pair is **split** (e.g. **offset** ranges are broken up into smaller pieces).
3. The runner redistributes the element and restriction pairs to several workers.
4. Element and restriction pairs are processed in parallel (e.g. the file is read). Within this last step, the element and restriction pair can pause its own processing and/or be split into further element and restriction pairs.



SparkReceiverIO





Agenda

- Introduction
- Developing an IO
- CDAP IO Overview
- Streaming Source IO – SparkReceiver
- **Testing IO**
- Akvelon Data Analytics and ML Accelerators demo

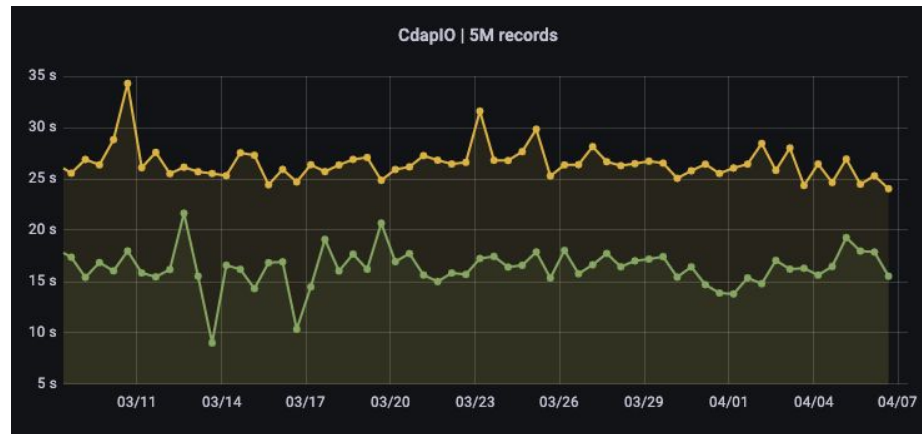
Testing IO and Release

IO Testing

- [testing guide](#), [IO transforms testing](#)
- Unit, integration and [performance test](#)
- Created RabbitMQ SparkReceiver on-demand source in Apache Beam that generates streaming data according to provided profile

Release

- Beam website [IO Connectors](#)
- Documentation & Readmes
- [Complete examples](#)



Documentation

Cdap IO

A `CdapIO` is a transform for reading data from source or writing data to sink CDAP plugin.

Batch plugins support

`CdapIO` currently supports the following CDAP Batch plugins by referencing `CdapIO.plugin` class name:

- [Hubspot Batch Source](#)
- [Hubspot Batch Sink](#)
- [Salesforce Batch Source](#)
- [Salesforce Batch Sink](#)
- [ServiceNow Batch Source](#)
- [Zendesk Batch Source](#)

Also, any other CDAP Batch plugin based on Hadoop's `InputFormat` or `OutputFormat` can be used. They can be easily added to the list of supported by class name plugins, for more details please see [CdapIO.readme](#).

Streaming plugins support

`CdapIO` currently supports CDAP Streaming plugins based on [Apache Spark Receiver](#).

Requirements for CDAP Streaming plugins:

- [ServiceNow](#). More info in the ServiceNow example [README](#).
- [Salesforce](#). More info in the Salesforce example [README](#).
- [Hubspot](#). More info in the Hubspot example [README](#).
- [Zendesk](#). More info in the Zendesk example [README](#).

Demo

AKVELON

Data and Analytics Accelerators

https://github.com/akvelon/DnA_accelerators

Akvelon Data and Analytics Accelerators

Akvelon is a digital product and software engineering company that empowers strategic advantage and accelerates your path to value in Data and Analytics, AI/ML, MLOps, Application development, and more with innovation and predictable delivery. Akvelon is providing this collection of accelerators as a reference and easy customizations for developers looking to build data, machine learning, and visualizations.

- [Get in touch about Data and Analytics and Data Migrations projects.](#)
- [Get in touch about ML projects.](#)
- [Get in touch about Google Cloud projects.](#)

Learn more about all our ML and software engineering services at our website akvelon.com.

Accelerators

ML, Streaming and Batch Data Processing

Apache Beam and Google Cloud Dataflow

[Apache Beam](#) provide unified streaming and batch processing to power ML and streaming analytics use cases. [Google Cloud Dataflow](#) is a managed to run Apache Beam in cloud with minimal latency and costs, and integrations with other Google Cloud products like [Vertex AI](#) and [Tensorflow TFX](#). Akvelon, a [Google Cloud Service Partner](#), and an active Apache Beam contributor and [Beam Summit](#) partner, presents several of our favorite accelerators for Dataflow.

- [Salesforce to Txt](#) - Flex templates for batch and streaming Salesforce data processing with Google Cloud Dataflow, using Apache Beam [CDAP IO](#).
- [Salesforce to BigQuery](#) - Flex templates for batch and streaming Salesforce data processing with Google Cloud Dataflow and BigQuery, using Apache Beam [CDAP IO](#). Flex templates provide a comprehensive example of using Machine Learning (ML) to process streaming data in Dataflow, using Java multilanguage pipeline with Python transforms to run custom TFX and PyTorch [ML models](#). This complete Flex template example also demonstrates creating and setting up [Expansion Service](#) in Dataflow to enable running custom Python transforms within a Java pipeline.
- [Tensorflow TFX model training with Apache Beam](#) - a Python notebook and Python Beam pipeline that demonstrates both Jupyter notebook to train a Tensorflow TFX ML model and the converted Python pipeline ready for Expansion Service use
- [PyTorch ML model training and Expansion Service for multilanguage pipelines with Apache Beam](#) - a complete example to train a PyTorch ML model using Apache Beam, convert the notebook to the Python pipeline, create custom Python Transforms and deploy as Apache Beam Expansion Service for Google Cloud Dataflow.

Custom Visualizations

Akvelon has accumulated vast experience with data analytics, custom visualizations, dashboards, and reports for a wide range of industries and use cases. Here are some of our favorite visualization accelerators.

Looker Visuals

github.com/akvelon/DnA_accelerators/tree/main/dataflow

main DnA_accelerators / dataflow /

Google Cloud Dataflow Accelerators

[Apache Beam](#) provides unified streaming and batch processing to power ML and streaming analytics use cases. [Google Cloud Dataflow](#) is managed to run Apache Beam in the cloud with minimal latency and costs, and integrates with other Google Cloud products like [Vertex AI](#) and [Tensorflow TFX](#). Akvelon, a [Google Cloud Service Partner](#), and an active Apache Beam contributor and [Beam Summit](#) partner, presents several of our favorite accelerators for Dataflow.

[Akvelon](#), a [Google Cloud Partner](#), is providing this open-source collection of Dataflow Flex templates as a reference and easy customizations for developers looking to build streaming, batch, multilanguage data pipelines with ML processing in Google Cloud Dataflow.


Flex Templates for Google Cloud Dataflow

Google Cloud Dataflow [Flex Templates](#) are a powerful way to build and run data pipelines on Google Cloud Platform. With [Flex Templates](#), you can package your pipeline code and dependencies as a Docker image, and then run it on Dataflow with just a few clicks. This makes it easy to build and deploy complex pipelines quickly and reliably.

- [Salesforce to Txt](#) - Flex templates for batch and streaming Salesforce data processing with Google Cloud Dataflow, using [CDAP IO](#).
- [Salesforce to BigQuery](#) - Flex templates for batch and streaming Salesforce data processing with Google Cloud Dataflow using [Apache Beam CDAP IO](#). Flex templates provide a comprehensive example of using Machine Learning (ML) to process data in Dataflow, using Java multilanguage pipeline with Python transforms to run custom TFX and PyTorch ML models. Flex template example also demonstrates creating and setting up [Expansion Service](#) in Dataflow to enable running custom transforms within a Java pipeline.

Machine Learning with Google Cloud Dataflow

- [Tensorflow TFX model training with Apache Beam](#) - a Python notebook and Python Beam pipeline that demonstrate both



Summary

Developing Beam IOs

Machine Learning

Multilanguage pipelines

https://github.com/akvelon/DnA_accelerators

AKVELON

https://github.com/akvelon/DnA_accelerators/

README.md

Akvelon Data and Analytics Accelerators

Akvelon is a digital product and software engineering company that empowers strategic advantage and accelerates your path to value in Data and Analytics, AI/ML, MLOps, Application development, and more with innovation and predictable delivery. Akvelon is providing this collection of accelerators as a reference and easy customizations for developers looking to build data, machine learning, and visualizations.

- [Get in touch about Data and Analytics and Data Migrations projects.](#)
- [Get in touch about ML projects.](#)
- [Get in touch about Google Cloud projects.](#)

Learn more about all our ML and software engineering services at our website akvelon.com.

Accelerators

ML, Streaming and Batch Data Processing

Apache Beam provide unified streaming and batch processing to power ML and streaming analytics use cases. Google Cloud Dataflow is a managed to run Apache Beam in cloud with minimal latency and costs, and integrations with other Google Cloud products like Vertex AI and Tensorflow TFX. Akvelon, a Google Cloud Service Partner, and an active Apache Beam contributor and Beam Summit partner, presents several of our favorite accelerators for Dataflow.

- [Salesforce to Txt](#) - Flex templates for batch and streaming Salesforce data processing with Google Cloud Dataflow, using Apache Beam CDAP IO.
- [Salesforce to BigQuery](#) - Flex templates for batch and streaming Salesforce data processing with Google Cloud Dataflow and BigQuery, using Apache Beam CDAP IO. Flex templates provide a comprehensive example of using Machine Learning (ML) to process streaming data in Dataflow, using Java multilanguage pipeline with Python transforms to run custom TFX and PyTorch ML models. This complete Flex template example also demonstrates creating and setting up [Expansion Service](#) in Dataflow to enable running custom Python transforms within a Java pipeline.
- [Tensorflow TFX model training with Apache Beam](#) - a Python notebook and Python Beam pipeline that demonstrates both Jupyter notebook to train a Tensorflow TFX ML model and the converted Python pipeline ready for Expansion Service use
- [PyTorch ML model training and Expansion Service for multilanguage pipelines with Apache Beam](#) - a complete example to train a PyTorch ML model using Apache Beam, convert the notebook to the Python pipeline, create custom Python Transforms and deploy as Apache Beam Expansion Service for Google Cloud Dataflow.

Custom Visualizations

Akvelon has accumulated vast experience with data analytics, custom visualizations, dashboards, and reports for a wide range of industries and use cases. Here are some of our favorite visualization accelerators.

Looker Visuals

AKVELON

https://github.com/akvelon/DnA_accelerators

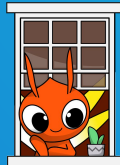
<https://akvelon.com>

Questions?

<https://www.linkedin.com/in/akosolapov>

<https://www.linkedin.com/in/elizaveta-lomteva>

BEAM
SUMMIT



BEAM
SUMMIT

Meeting Security Requirements for Apache Beam Pipelines on Google Cloud

Lorenzo Caggioni
Google

[linkedin.com/in/lcaggio/](https://www.linkedin.com/in/lcaggio/)

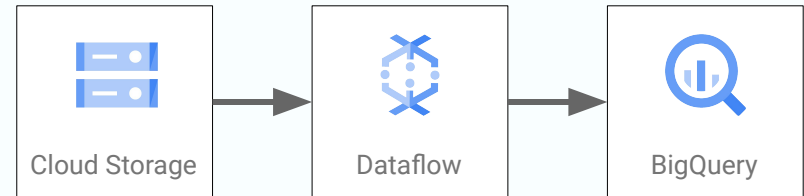
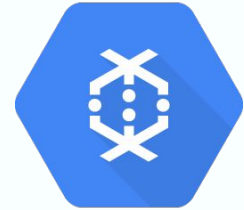


Securing a Beam Pipelines on Google Cloud

- Private resources
- Role separation and least privileges
- Data Encryption at rest

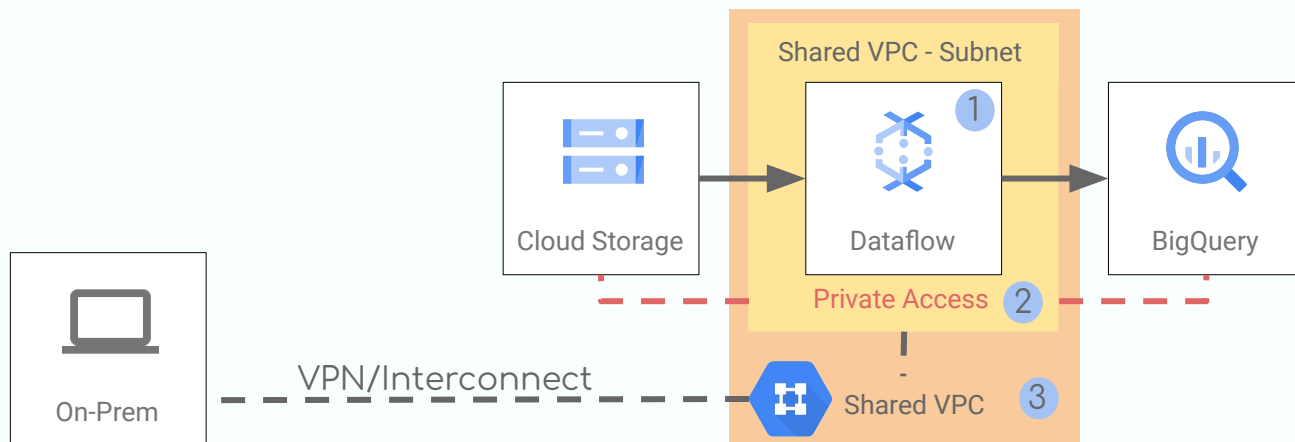
Customer requirements

1. Internal addressment of tenants must be private.
2. Every tenants must be isolated and dedicated to a specific system of services.
3. All data must have encryption at-rest with keys managed by ACME's security team.



1. Internal addressment of tenants must be private.

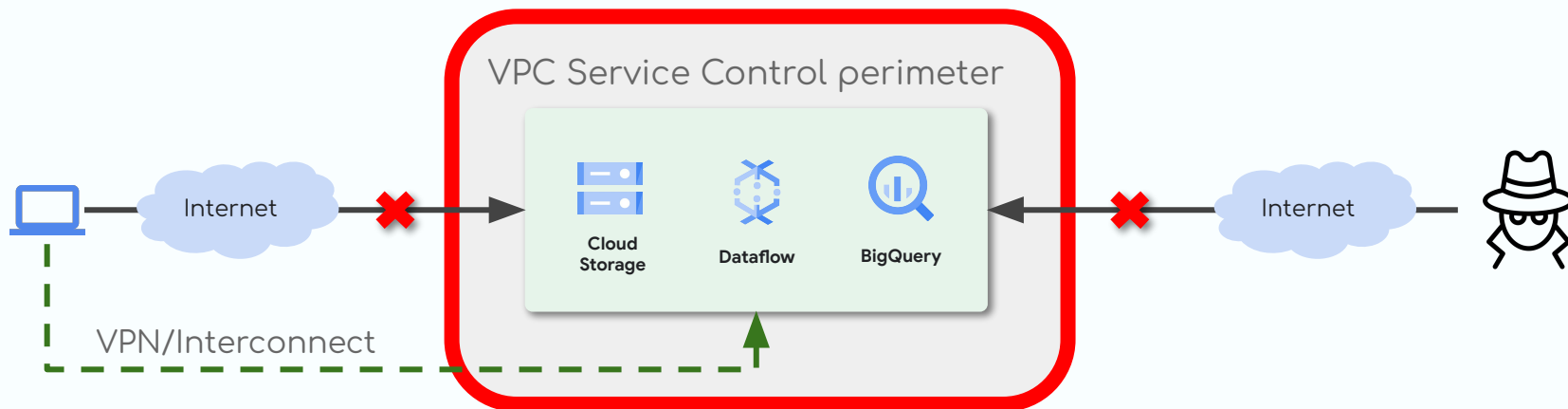
1. Set `disable-public-ips` when deploying the pipeline
2. Enable `Private Access` on the subnet to access Google APIs
3. Network: shared-VPC



1. Mitigate Data Exfiltration

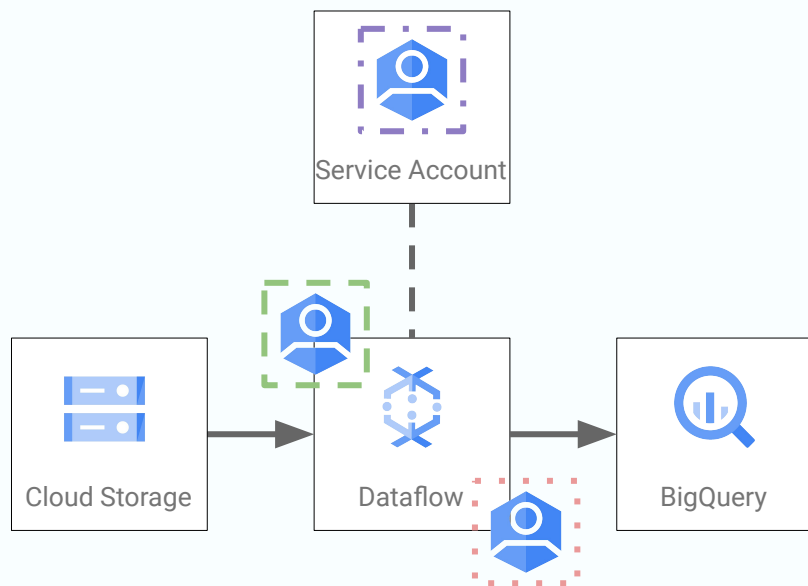
VPC Service Controls helps preventing data exfiltration and controlling access to Google APIs.

Isolate resources of multi-tenant Google Cloud services to mitigate data exfiltration risks.



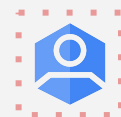
2. Tenants must be isolated

IAM and Service Accounts



Dataflow Service Agent

roles/dataflow.serviceAgent
roles/compute.networkUser



Worker Service Account

roles/storage.objectAdmin
roles/dataflow.worker
roles/bigquery.dataEditor

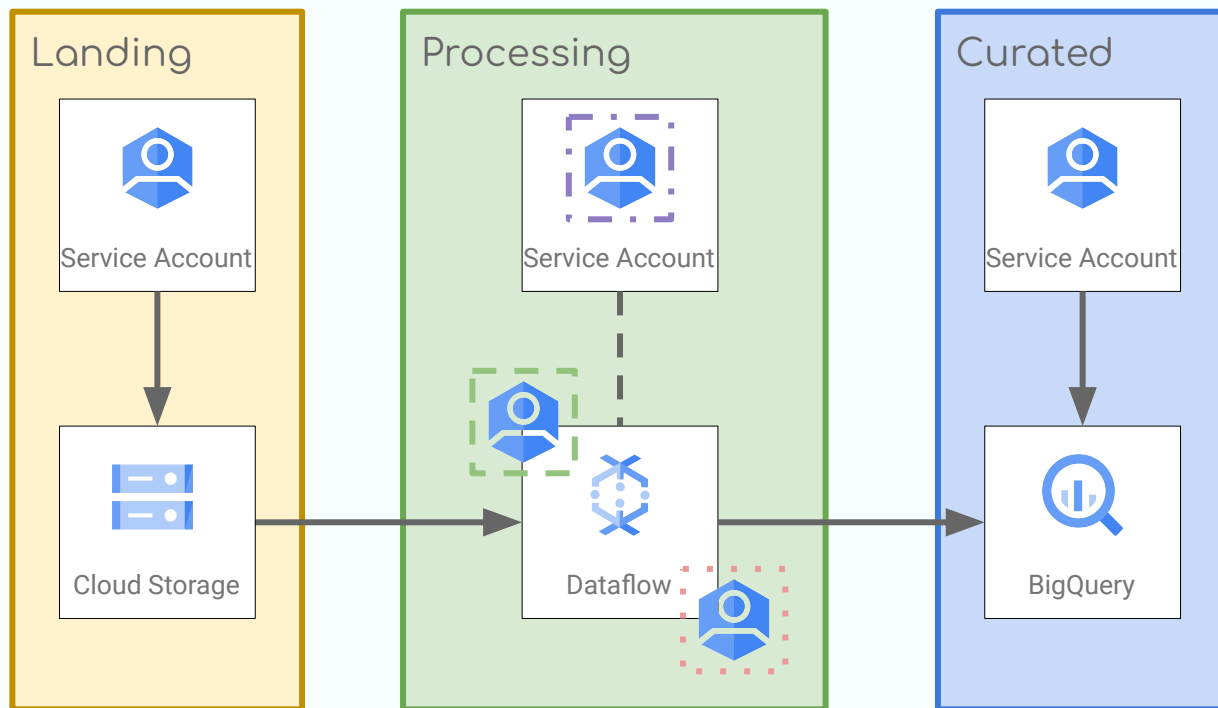


Job orchestrator

role/iam.serviceAccountUser
role/dataflow.admin

2. Tenants must be isolated

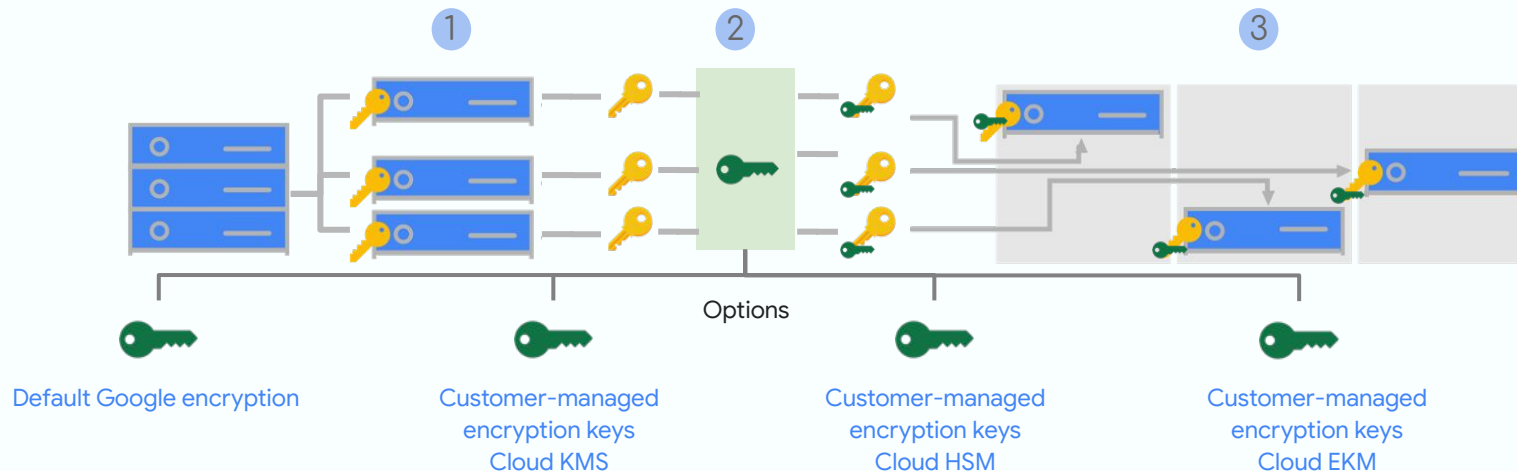
Project separation



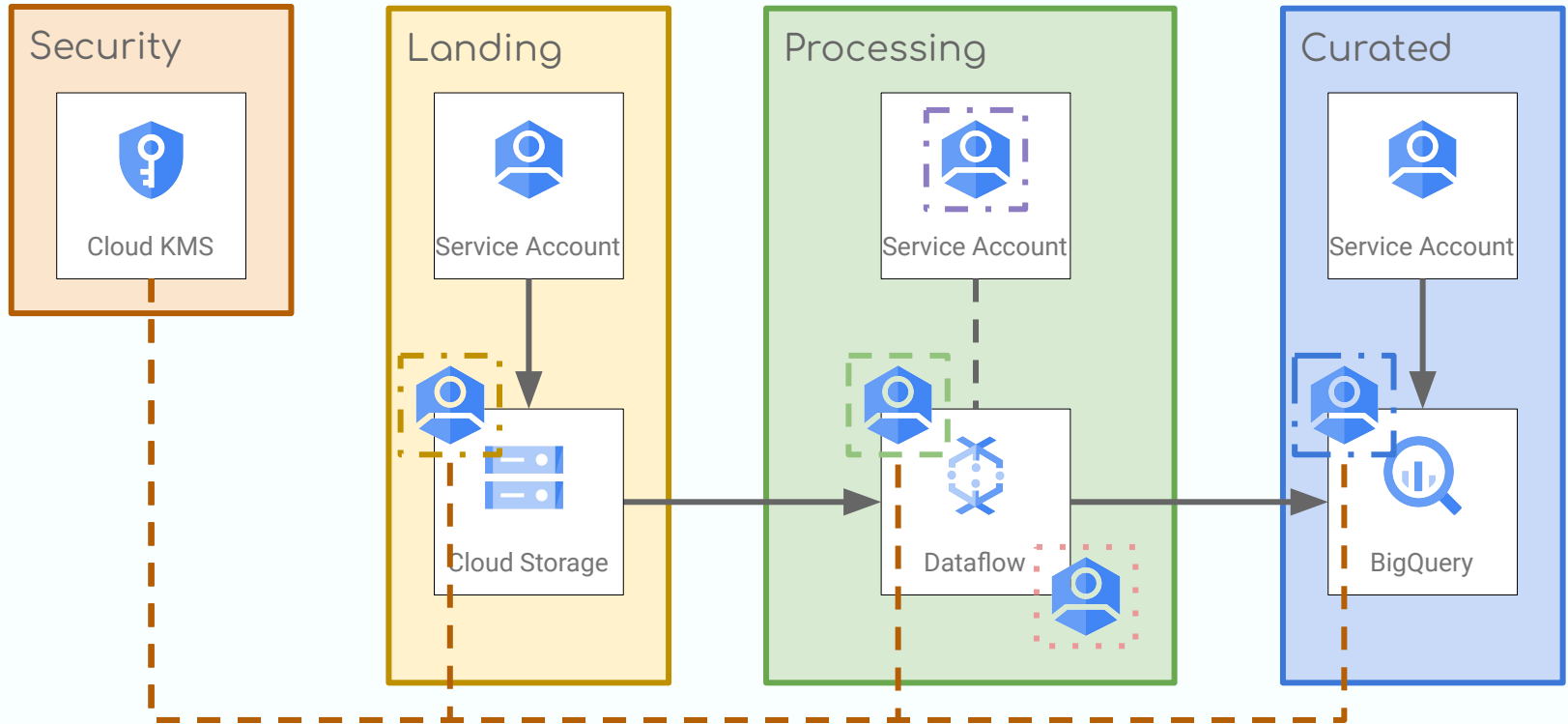
3. At rest encryption

Data at rest are encrypted on GCP:

1. Data split in chunk and encrypted with a key: Data Encryption Key (DEK)
2. DEK encrypted with Key Encryption Key (KEK)
3. Chunk stored with encrypted DEK



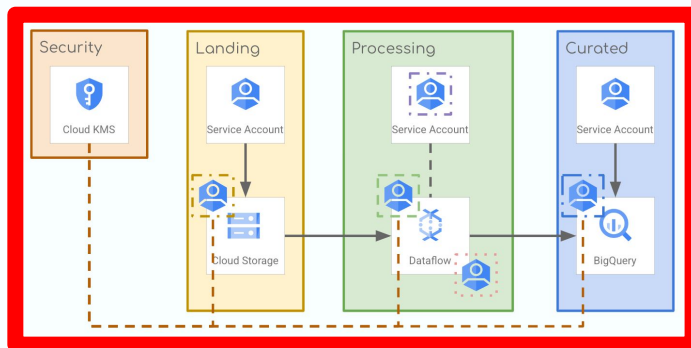
3. At rest encryption



`roles/cloudkms.cryptoKeyEncrypterDecrypter`

Recap

- ✓ Every tenants must be isolated and dedicated to a specific system of services.
- ✓ Internal addressment of tenants must be private.
- ✓ All data must have encryption at-rest with keys managed by ACEME's security team.



SCAN ME

End to end example

Lorenzo Caggioni

QUESTIONS?

Contact info

<https://twitter.com/lcaggio>

<https://www.linkedin.com/in/lcaggio>

<https://github.com/lcaggio>

BEAM
SUMMIT

Simplifying Speech-to-Text Processing with Apache Beam and Redis

Pramod Rao
& Prateek Sheel

Simplifying Speech-to-Text Processing with Apache Beam and Redis



Pramod Rao

Cloud Data Engineer

Google Cloud Consulting



Prateek Sheel

Data & Analytics Consultant

Google Cloud Consulting

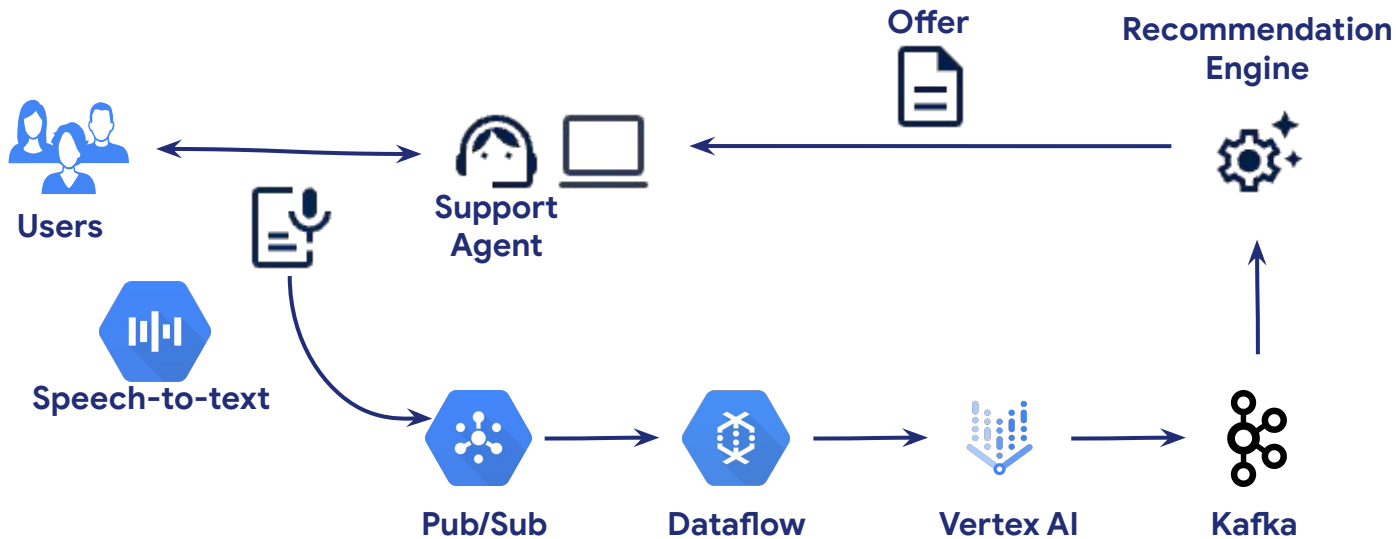
Overview	01
Design Journey	02
Lessons Learned	03

01

Overview



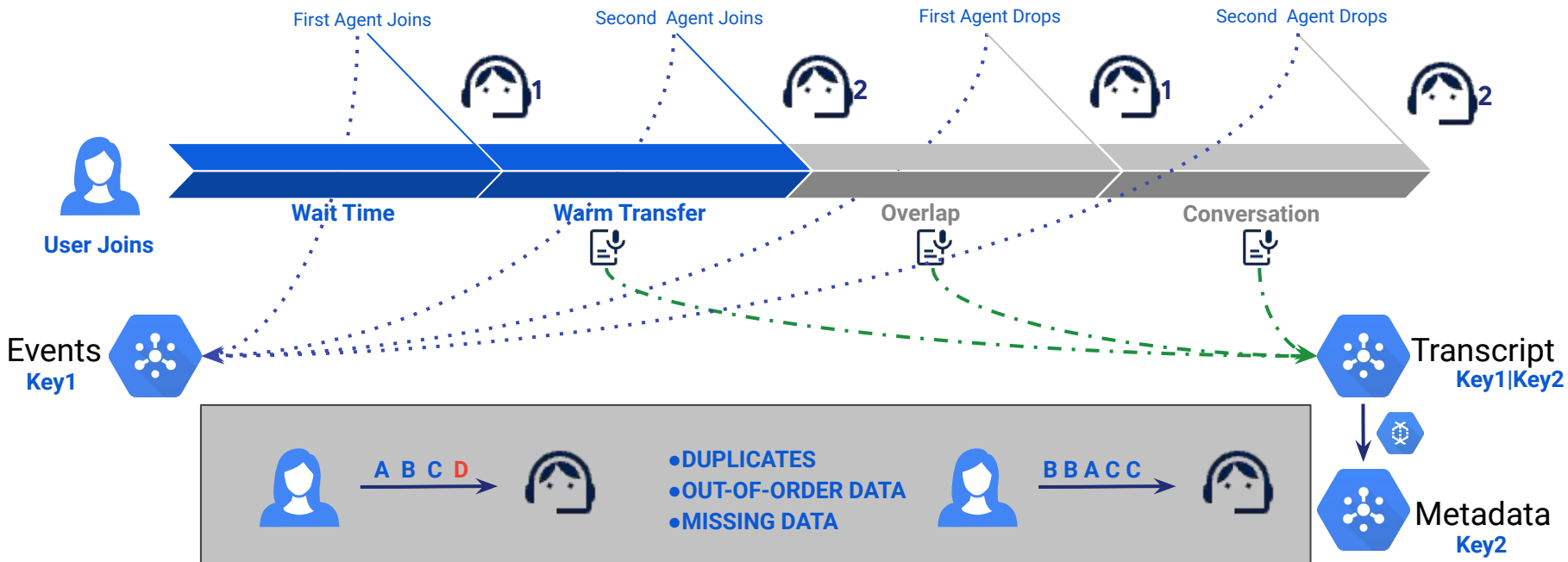
Business Process



So, what's the problem?

Multiple Call-Transfer Scenarios

Plus, additional business rules

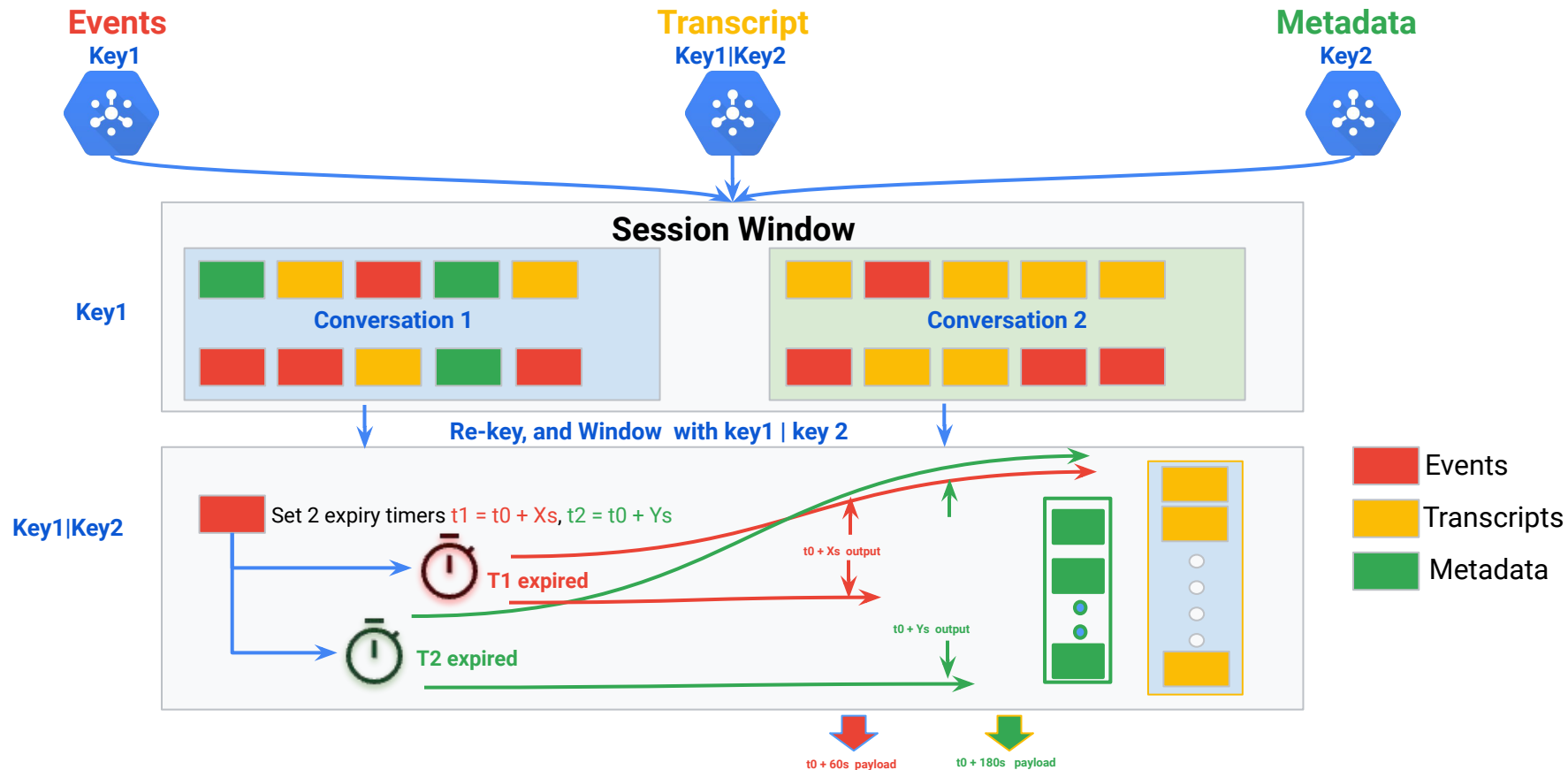


02

Design Journey



Design Approach # 1



Design 1 Trade Offs

Dependencies

No state external to Dataflow. No external service dependencies.

Latency

Need to wait for the session to end and the timers to expire before the output payloads can be produced. Not ideal based on the business SLO.

Completeness

In some cases all of the information required to creating the output payloads may not be available when the timers expire. This is due to the uncertain **ordering** of events.

Code Complexity

Windowing allows for **relatively** simpler business logic implementation for creating the output payloads since re-keying produces outputs at the required **granularity**

Design Approach # 2

Events

Key1



Transcript

Key1|Key2

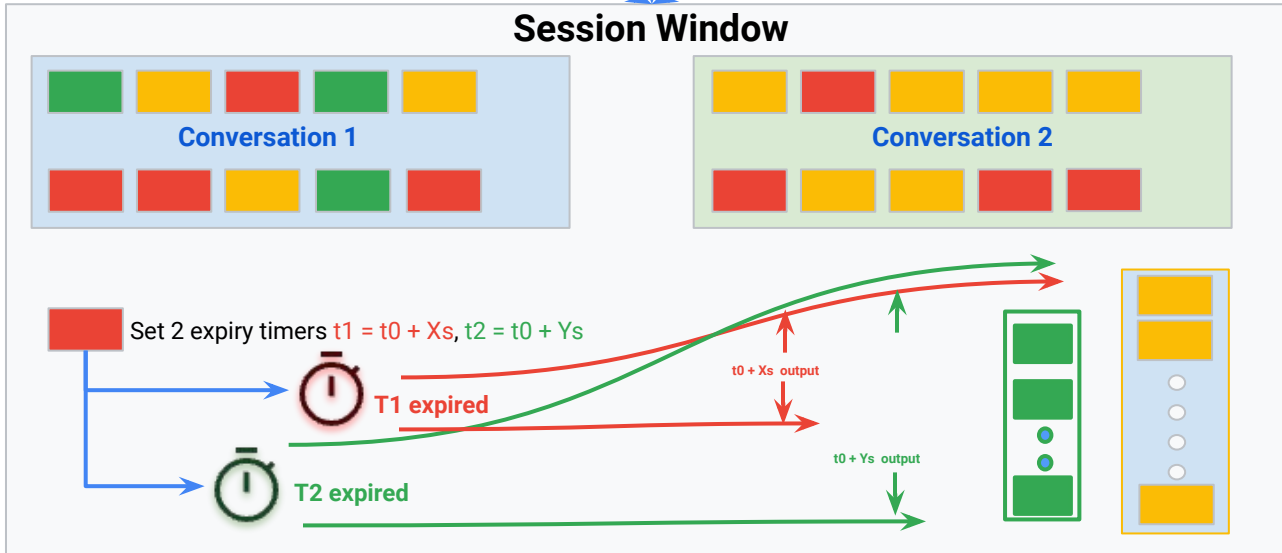


Metadata

Key2



Key1



- Events
- Transcripts
- Metadata

t0 + 60s payload

t0 + 180s payload

Design 2 Trade Offs

Dependencies

No state external to Dataflow. No external service dependencies.

Latency

Need to wait for the session to end and the timers to expire before the output payloads can be produced. Not ideal based on the business SLO.

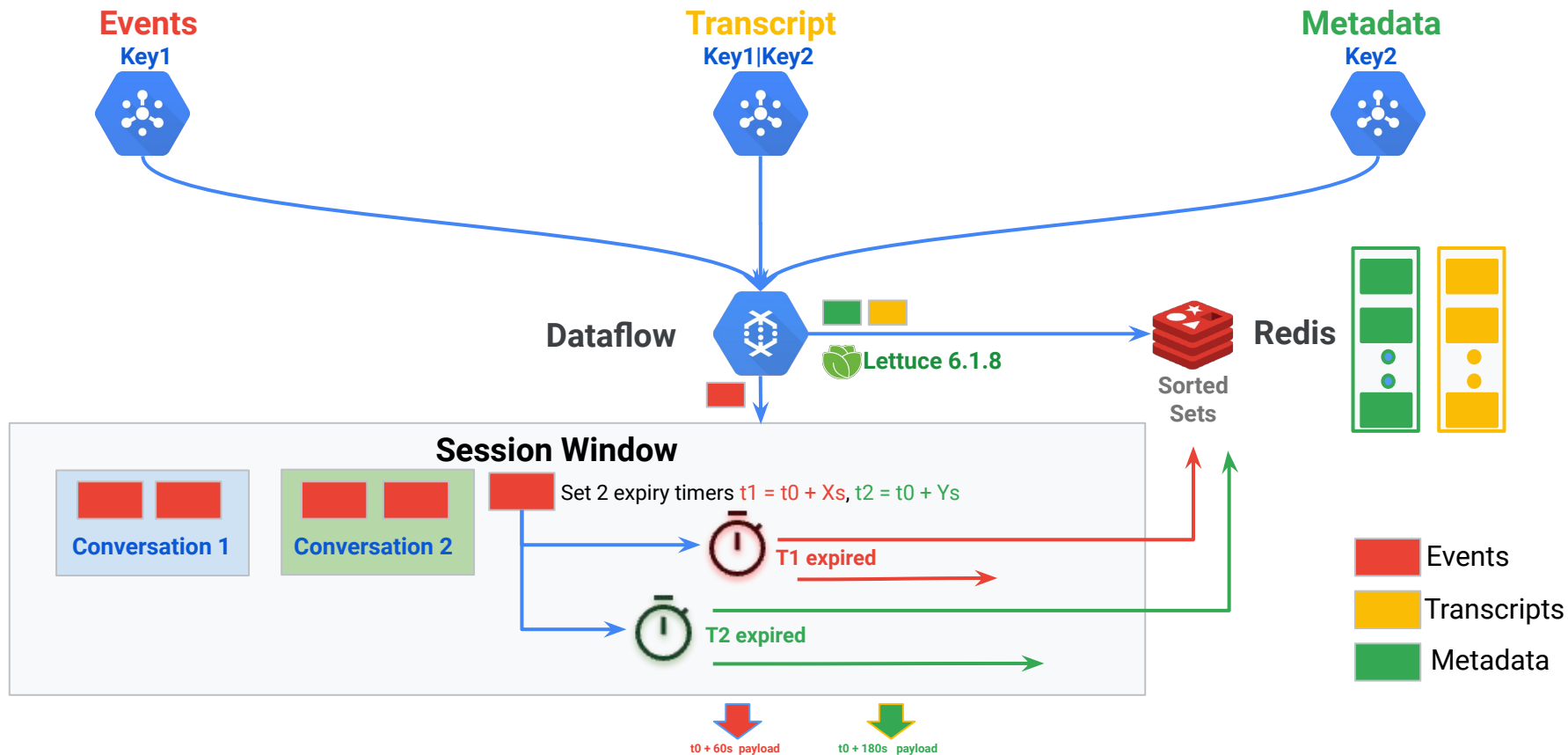
Completeness

In some cases all of the information required to creating the output payloads may not be available when the timers expire. This is due to the uncertain **ordering** of events.

Code Complexity

Granularity of outputs doesn't match the inputs thereby increasing the business logic **complexity** required to produce the output payloads

Design Approach # 3



Redis

Latency

Low latency data store that dovetails well with streaming use cases

Order

We rely on Redis sorted sets for accumulating the speech transcripts, we are able to maintain the **order** of the conversation as well as **deduplicating** the transcripts **automagically**

Data Lifecycle

Redis offers a simple approach to manage **cleanup** of stale data

Design 3 Trade Offs

Dependencies

Dependency on a managed Redis instance. This also results in additional **costs** to host a Redis instance in the Cloud environment.

Latency

No need for any additional wait time over and above the required timers.

Subsecond end-to-end latency for ML predictions.

Completeness

Least chance of incomplete outputs due to the **ordering** provided by Redis

Code Complexity

Much **simpler processing** because complicated scenarios related to cross-referencing the three data sources are eliminated. Only need to “act” on events.

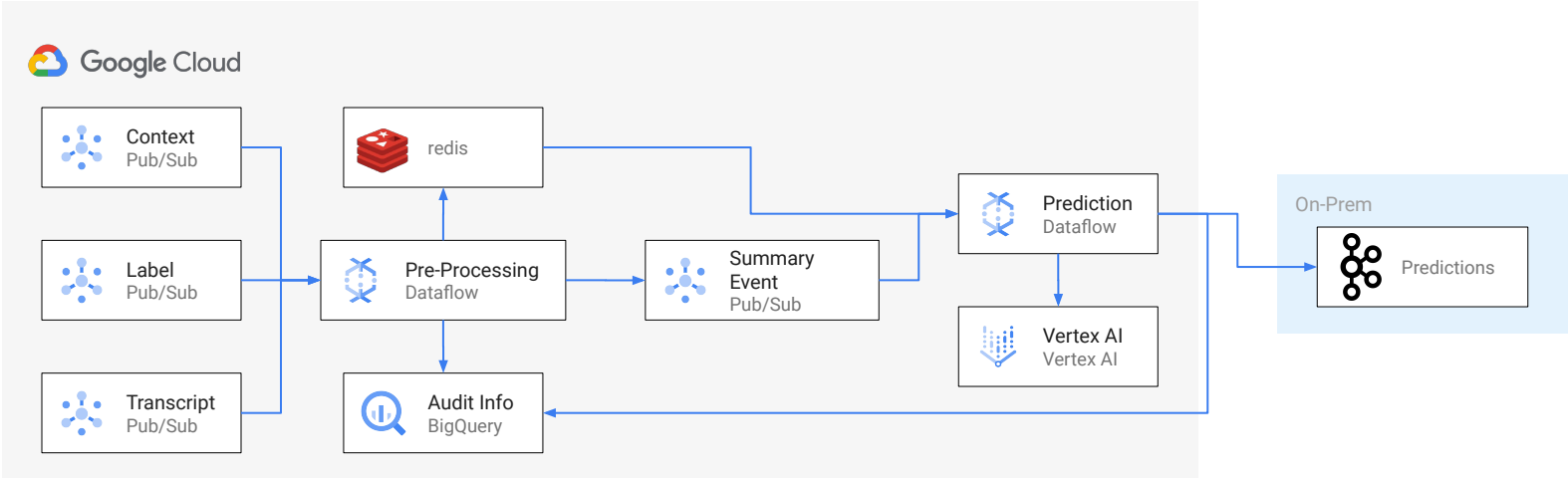
Latency Metrics*

Dataflow	PreProcessing	Redis	Predictions	End-To-End
Machine Type	Avg. (ms)	Avg. (ms)	Avg. (ms)	Avg. (ms)
n1-standard-2 t0+60s	1210.90	20.84	204.83	1441.75
n1-standard-2 t0+180s	1155.52	18.62	260.33	1441.72
n2d-standard-4 t0+60s	580.38	9.84	198.68	796.10
n2d-standard-4 t0+180s	596.54	9.98	260.54	874.35

*Excluding the wait time to accumulate data for each event type

Final Solution

Speech-to-text Processing with Apache Beam and Redis



03

Lessons Learned



Lessons Learned

Functional



Order of data

Real world scenarios include out-of-order data, duplicates, and missing elements



Granularity of inputs

Business logic is greatly simplified if all inputs are at the same level of "granularity"



Latency

Latency requirements dictate the nature of the final solution

Operational



Observability

Non functional requirements such as operational metrics and dead-letter queues are essential to gain insights into the processing state at any time



Configurability

Levers should be provided to change the processing characteristics without changing any code



Representative test data

"Good" test data is imperative to shorten the development lifecycle and can be tricky to generate or acquire



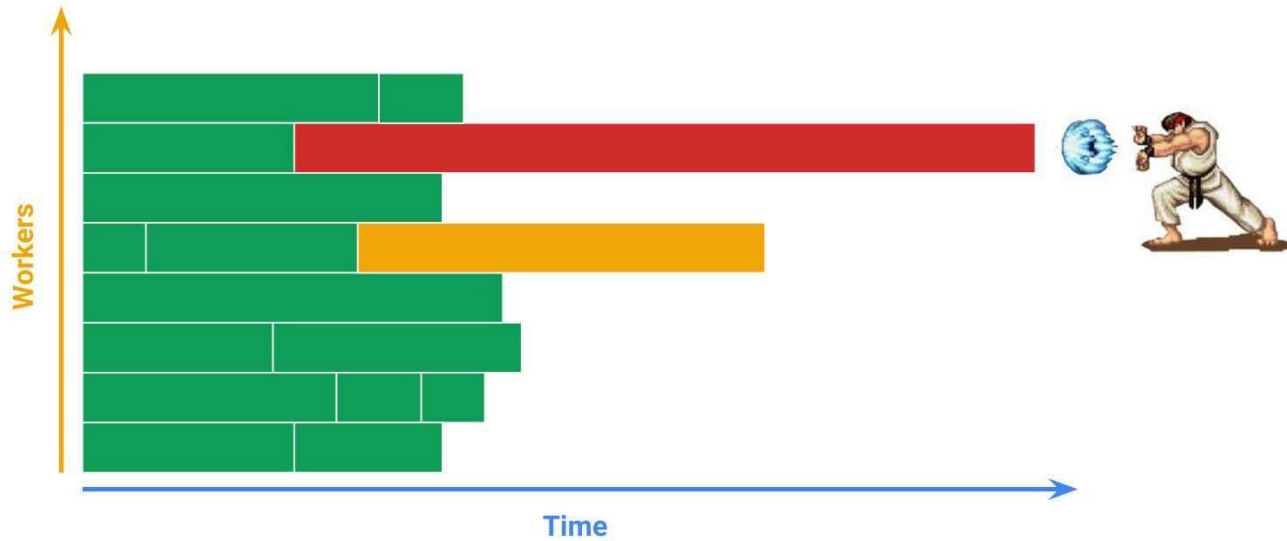
Thank you!

<https://cloud.google.com/consulting>

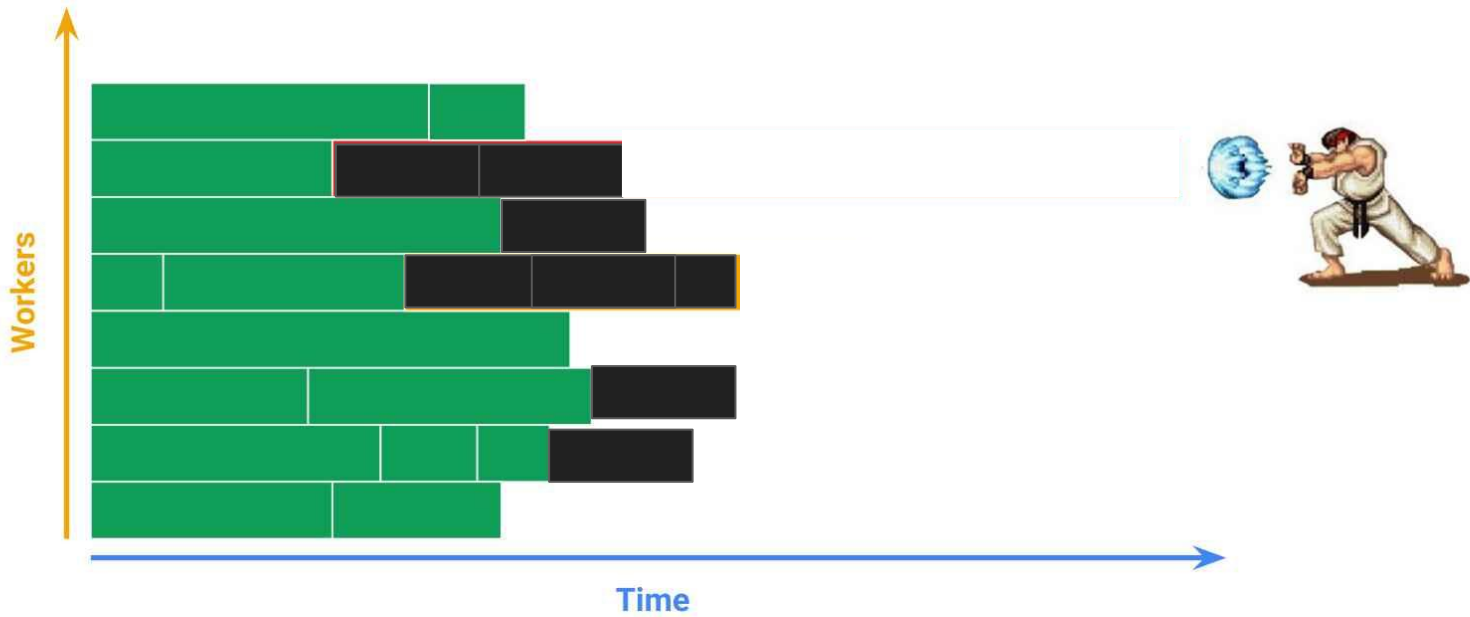
BEAM
SUMMIT

Hot Key Detection and Handling in Apache Beam Pipelines

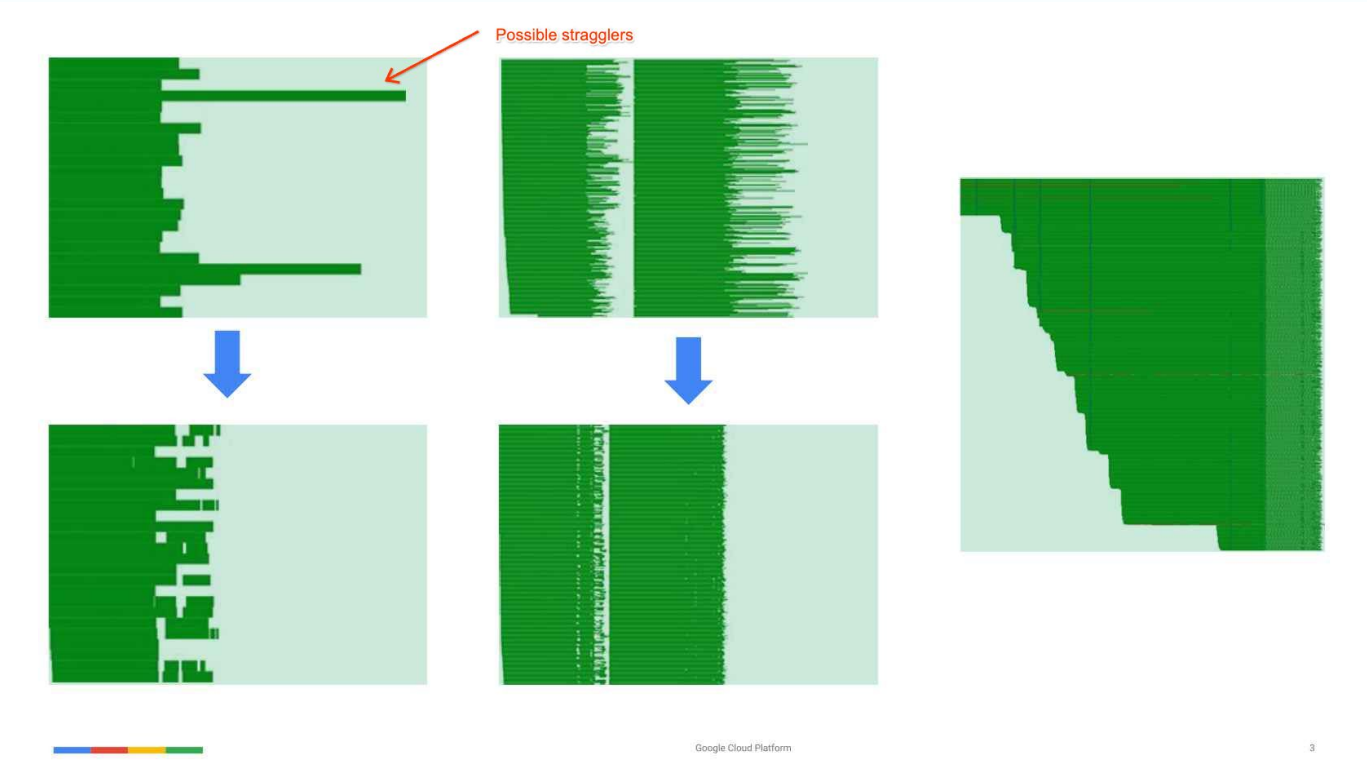
Shafiqa Iqbal
& Ikenna Okolo



To this



How stragglers can look like



WordCount

```
Pipeline p = Pipeline.create(options);
p.apply(TextIO.Read.from("gs://dataflow-samples/shakespeare/*"))
  .apply(FlatMapElements.via(
    word → Arrays.asList(word.split("[^a-zA-Z']+")))
  .apply(Filter.byPredicate(word → !word.isEmpty()))
  .apply(Count.perElement())
  .apply(MapElements.via(
    count → count.getKey() + ": " + count.getValue())
  .apply(TextIO.Write.to("gs://.../..."));
p.run();
```



Primitives to keep in mind

ParDo



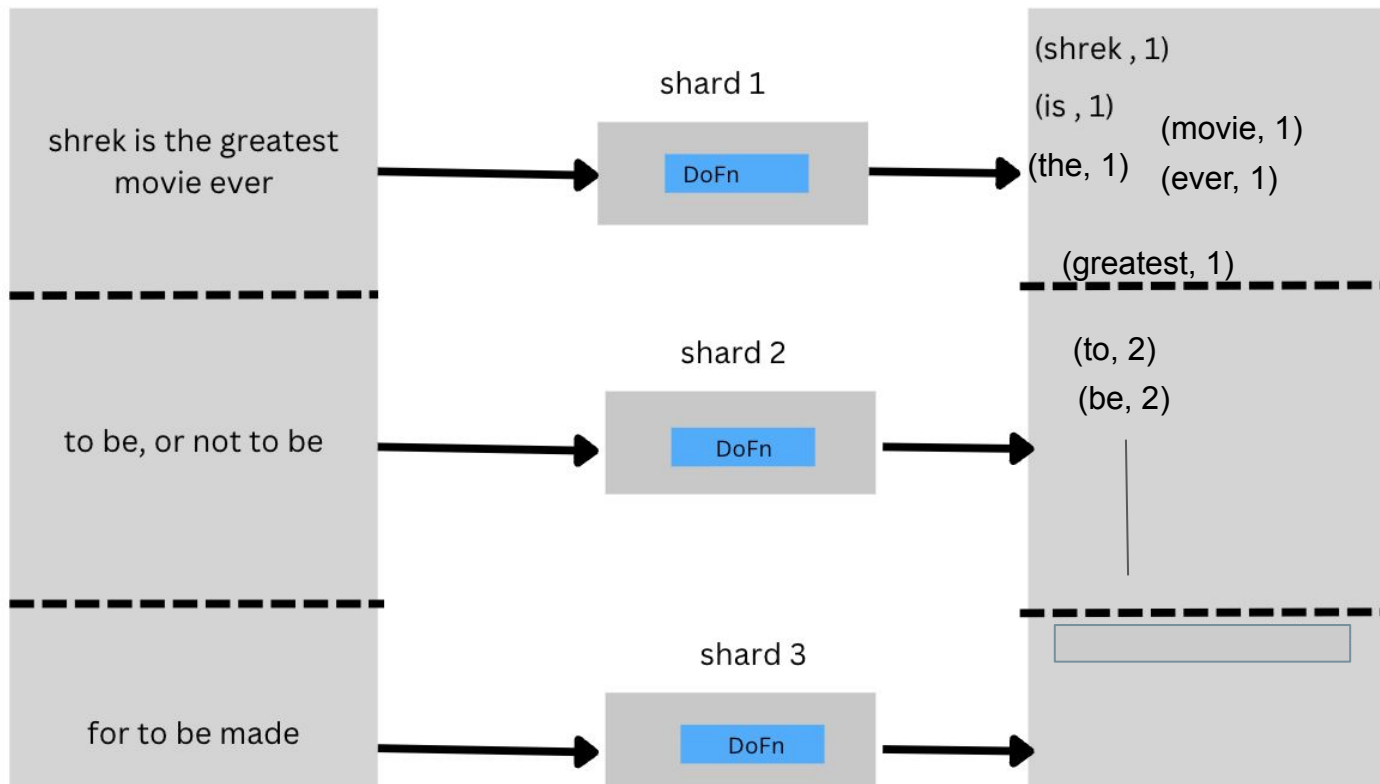
GroupByKey



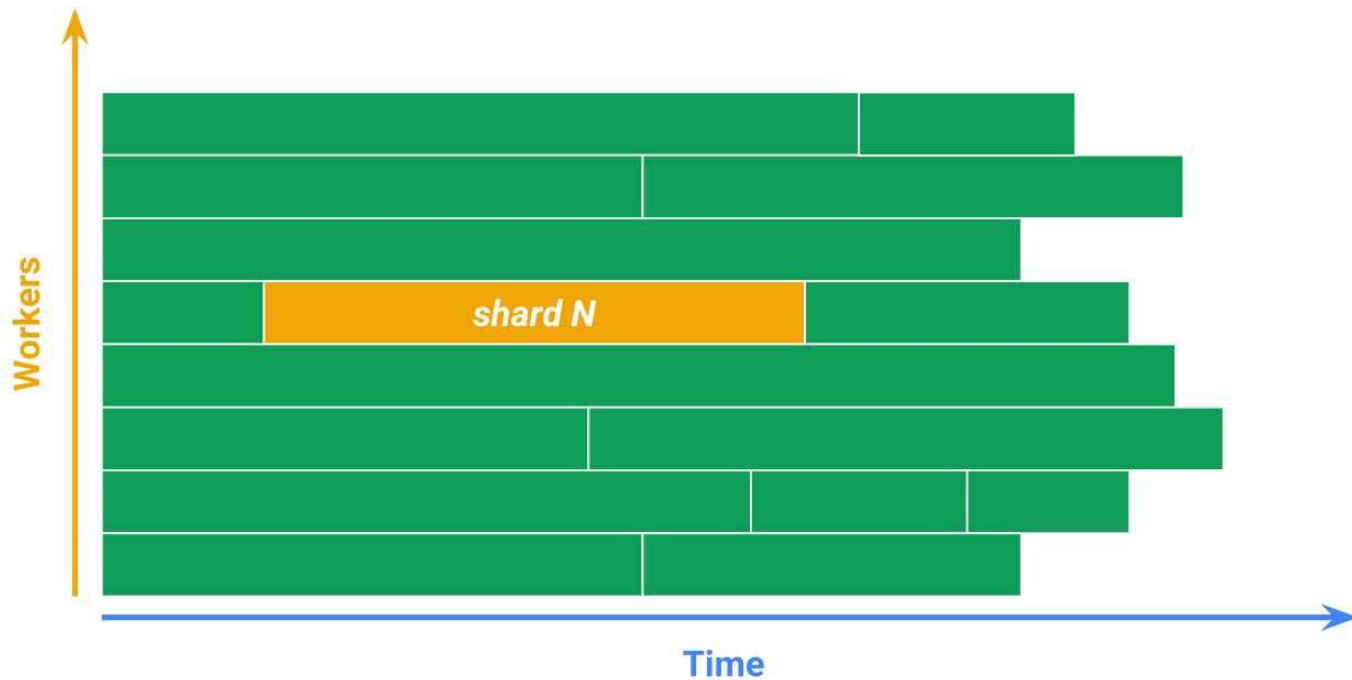
MapReduce = ParDo + GroupByKey + ParDo



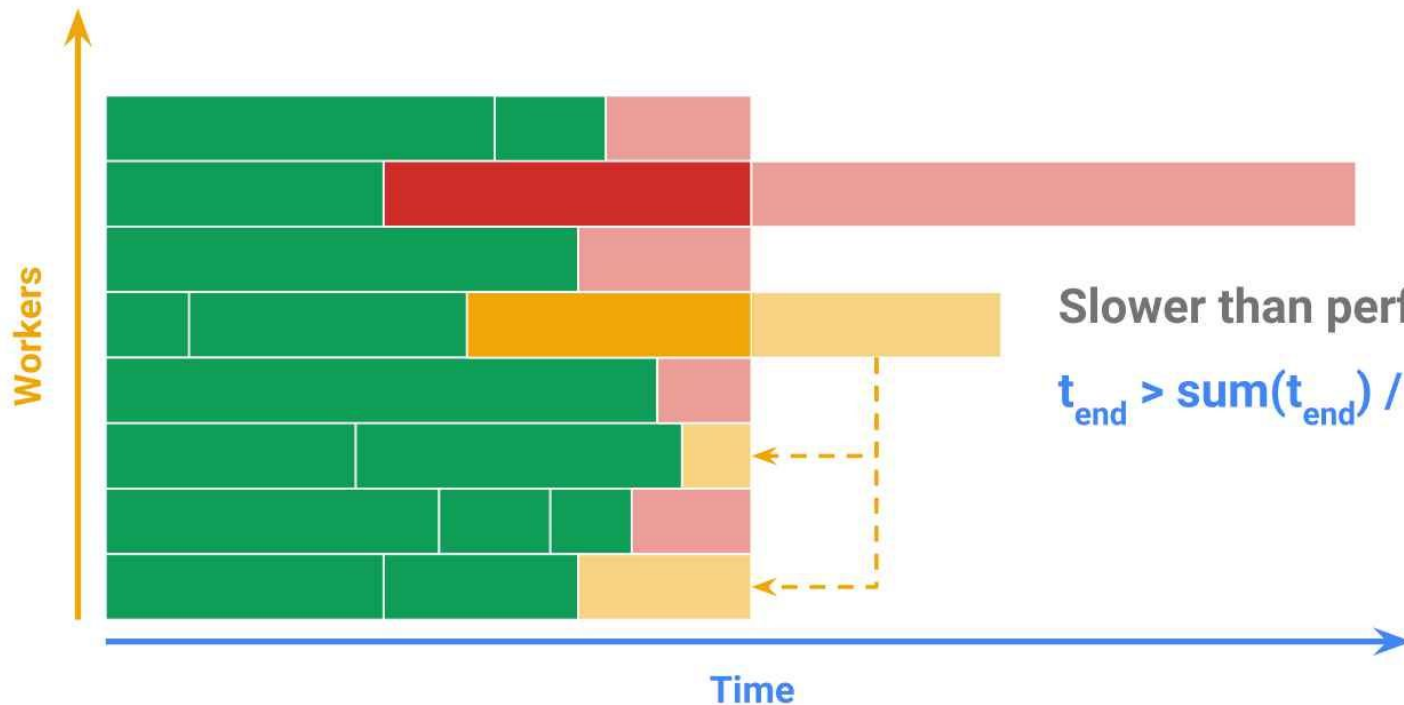
How a ParDo would work



Gantt charts



What is a straggler, really?



Slower than perfectly-parallel:

$$t_{\text{end}} > \text{sum}(t_{\text{end}}) / N$$

Amdahl's law: it gets worse at scale

$$\textit{Speedup} = \frac{N}{1 + (N-1) \cdot S}$$

#workers

serial fraction

Higher scale \Rightarrow More bottlenecked by serial parts.

Reasons for Stragglers

Uneven partitioning

- Process dictionary in parallel by first letter
-> 6x speedup only by ahmdahl's law

Uneven Complexity

- Join keys with some external input values

Uneven resources

- Bad machines, network or resource contention

Bugs

- Slow RPCs or bugs

Reasons for Stragglers

Hot keys

Uneven partitioning

- Process dictionary in parallel by first letter
-> 6x speedup only by ahmdahl's law

Uneven Complexity

- Join keys with some external input values

Uneven resources

- Bad machines, network or resource contention

Bugs

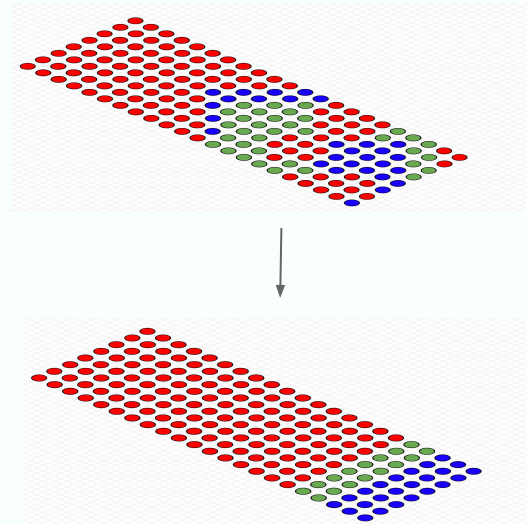
- Slow RPCs or bugs

What are hotkeys

A hot key is a key with enough elements to negatively impact pipeline performance. These keys limit a Pipeline's ability to process elements in parallel, which increases execution time.

Think about hotkeys in this way. Let's imagine there's a room filled with 150 Red, 30 Blue and 20 Green unsorted plates and there are 3 students who are to arrange those plates in sorted orders (as seen here to the right).

Let's assume that student 1 will sort the Red plates, student 2 will sort the blue and the last student will sort the green plates.



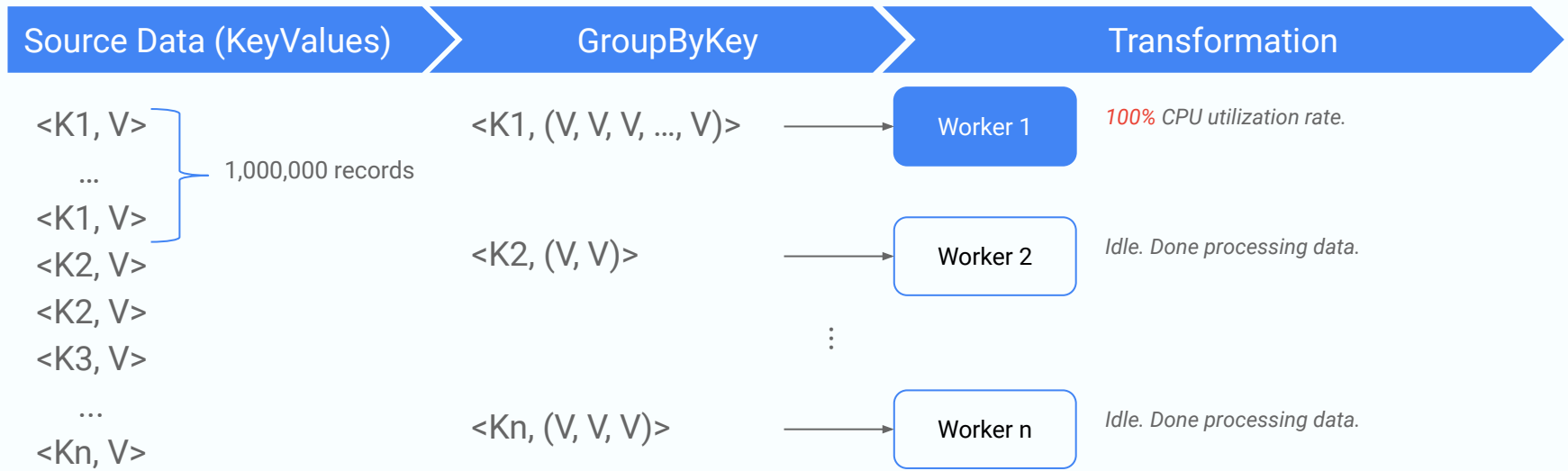
What are hotkeys contd...

From the illustration in the previous slide, students 2 and 3 will finish before student 1. Though the second and third students had already completed sorting their respective colored plates, they have to wait for the first student to complete theirs before the task can be termed as completed. This delay by student 1 is due to the larger number of plates they need to sort. In parallel processing, this is referred to as hotkeys.

If we replace the students with workers and the unsorted-plates with work-items to be processed, we can apply the same thinking to Dataflow pipelines. If the work-items are not evenly distributed, then there's bound to be an issue of hotkeys which obviously would impact the performance of the Pipeline.

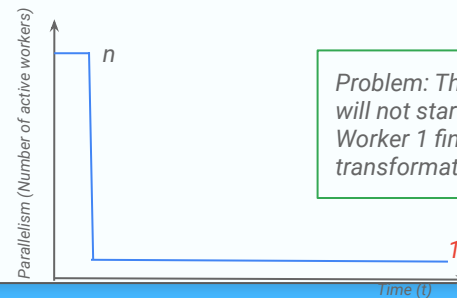
In subsequent slides, we will explain this using a Key Value pair to represent individual work-items.

How do Hotkeys cause problems?



Note that the dataset is heavily imbalanced.

K1 has broken the uniformity and thus is called the "hotkey."



Problem: The next job will not start until Worker 1 finishes its transformations.

How to identify hotkeys

CPU utilization (All Workers) ? Create alerting policy

Feb 20, 2020 3:04 PM

SELECTED LINES

Name	Job ID	CPU Utilization
	sit-793344-022-02201245-s-rbwz	100.2%
	bshe-harness-0c	100.2%
	sit-793344-022-02201245-s-rbwz	100.2%
	bshe-harness-0c	4.19%
	sit-793344-022-02201245-s-rgm8	4.19%
	bshe-harness-0r	3.63%
	sit-793344-022-02201245-s-dggv	3.63%
	bshe-harness-0x	3.63%
	sit-793344-022-02201245-s-rqgx	3.63%

One of the quickest ways to identify a Job that is impacted by hotkeys is by taking a quick look at the worker CPU utilisation. While some workers are maxing out at about 90% utilisation, some are idle at about <5% utilisation. This truly indicates that there is a possibility that the Job is stuck due to hotkeys.

Fxg-dna-gsi-edd-npe-dev-1, 2020-02-27_04_30_10-5380840952249221324

#

BEAM SUMMIT NYC 2023

What can you do?

Uneven partitioning

Uneven complexity

Uneven resources

Noise

Oversplit

Hand-tune

Use data statistics

Backups

Restarts

Predictive

Weak

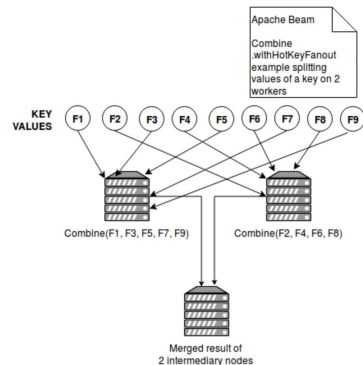
Data Monitoring, key partitioning,
iterative optimization

Using statistical analysis to pre-detect the hot key

How do you fix hotkeys?

To resolve this issue, you may have to check that your data is evenly distributed. If a key has disproportionately many values, consider the following courses of action:

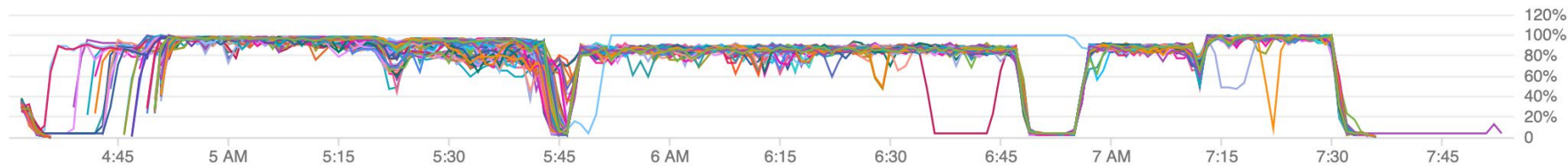
- Rekey their data. Apply a [ParDo](#) transform to output new key-value pairs.
- Autosharding
- `Combine.Globally #withFanout(int fanout)`
- Java jobs should consider using the [Combine.PerKey.withHotKeyFanout](#) transform.
- Python jobs should consider using the [CombinePerKey.with_hot_key_fanout](#) transform.
- Finally, consider enabling [Dataflow Shuffle](#) (if using dataflow).



Job not impacted by hotkeys anymore!

CPU utilization (All Workers) ▾ ?

[Create alerting policy](#)



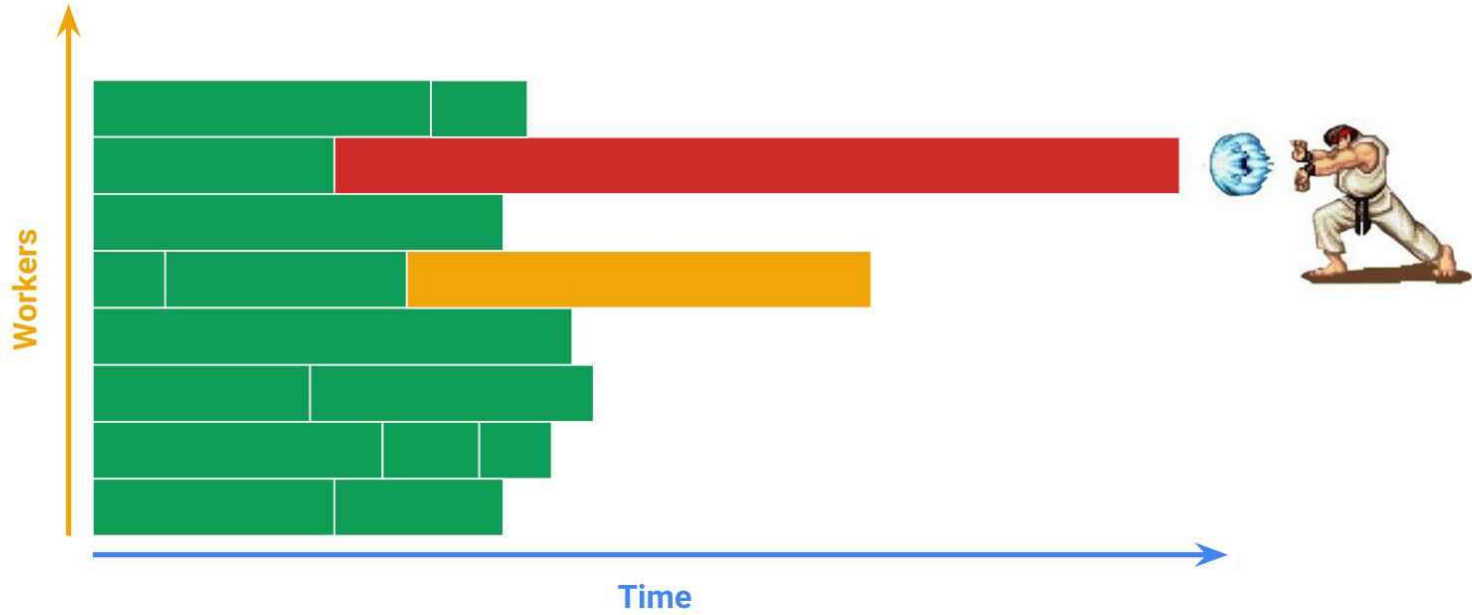
Name

Value

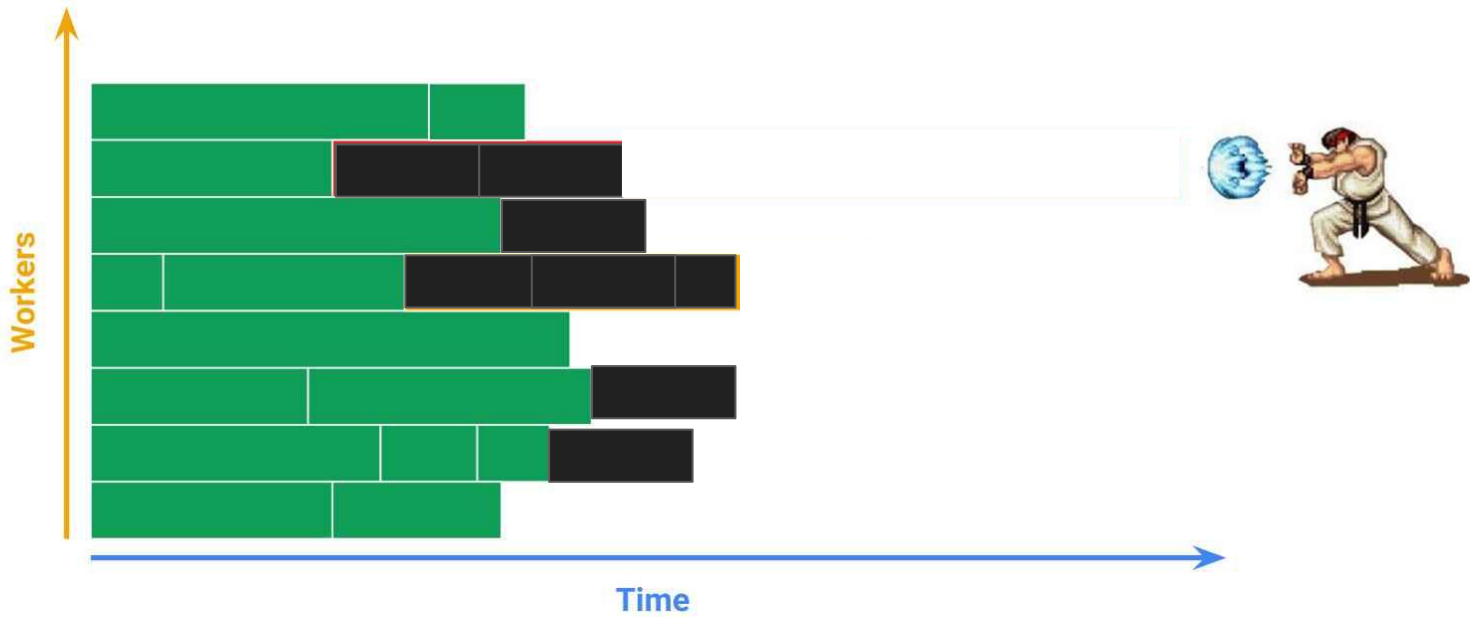


● dna-integration-gsi-edd-p-02270430-b79j-harness-0100	1.66%
● dna-integration-gsi-edd-p-02270430-b79j-harness-05z9	1.61%
● dna-integration-gsi-edd-p-02270430-b79j-harness-0g4x	2.08%
● dna-integration-gsi-edd-p-02270430-b79j-harness-0kjp	1.81%

From this



To this



Hotkeys FAQ

Can we assign a more powerful machine to the worker that is processing the hotkey (i.e. Worker 1)?

>> Unfortunately, you cannot. Dataflow, by design, assigns the same machine to *all* of its workers.

In that case, if all workers run with powerful machines, the pipeline will finish quicker. + It will be cheap, since most of them will be idle anyways.

>> This will not speed up the process. A powerful machine will still use up only one of its cores. Imagine a giant for-loop to better understand -- cores do not split the work of a for loop.

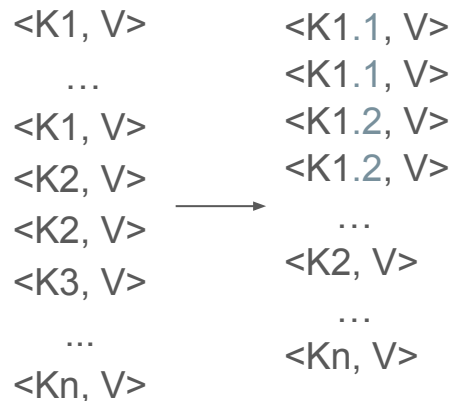
I enabled autoscale, but my job doesn't finish any faster. Why?

>> You will see in monitoring that the average CPU utilization rate is far below 20%; therefore, Dataflow will not bring in more workers. Even if it does, it won't help -- remember that you already have n-1 idle workers. Surely n idle workers won't make a difference.

Root cause: dataset is imbalanced.

Fix the root cause: balance the dataset.

Solution: Classify the imbalanced key and break them down into smaller pieces.



BEAM
SUMMIT

Troubleshooting Slow Running Beam Pipelines

By Mehak Gupta
Google Cloud, Canada

About Me



Hello!

I'm Mehak

Technical Solutions Specialist at Google Cloud

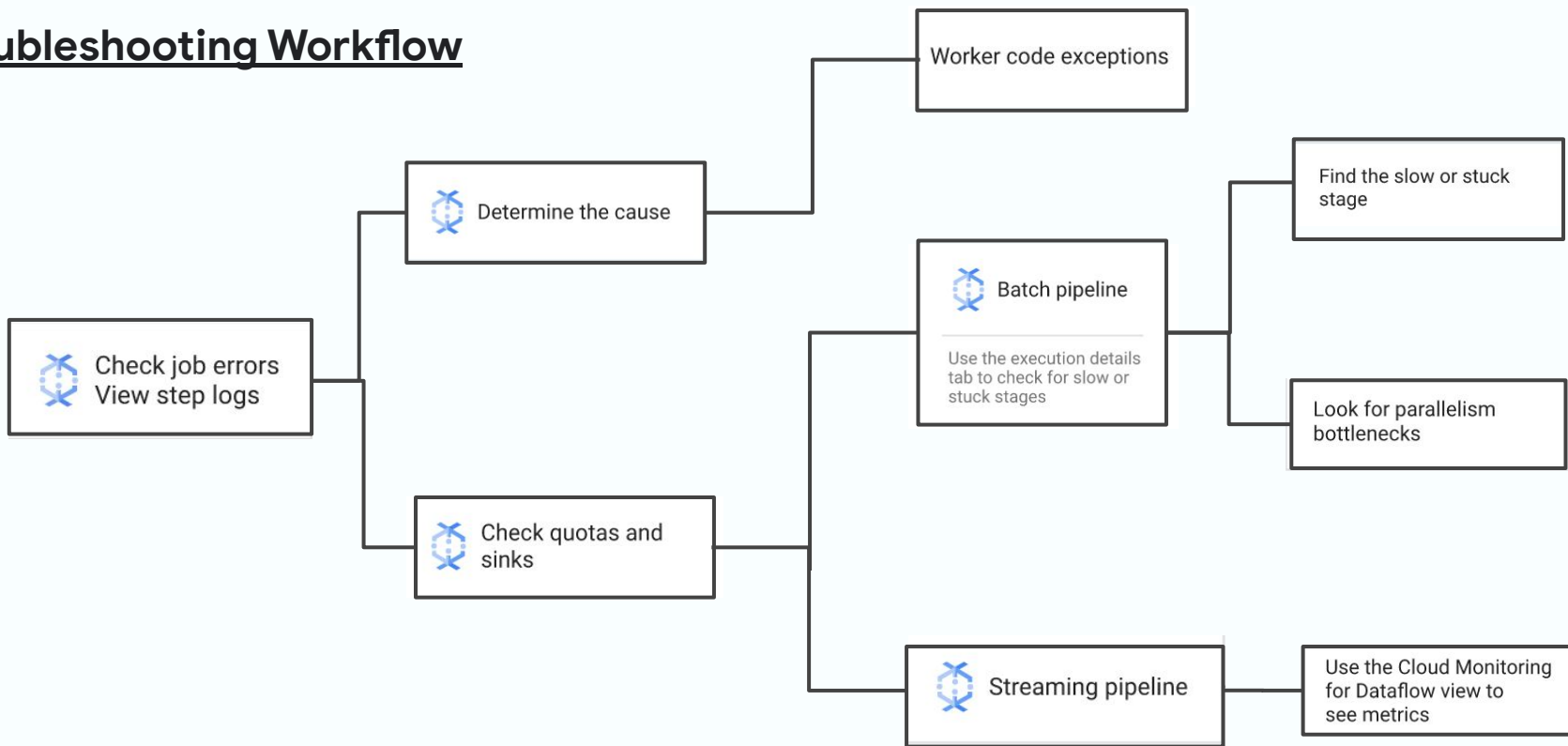


- Apache Beam pipeline troubleshooting techniques that would empower professionals to research and resolve Beam issues
- Self service skills would reduce MTTR (Mean Time To Recover) from a job failure significantly
- Share some tricks and samples of troubleshooting slow running beam pipelines using Dataflow as an example

How to identify if the beam pipeline is slow/stuck

- Pipeline is running from a long time without reporting results
- Increased data watermark or system latency
- Pipeline is not consuming input

Troubleshooting Workflow



Troubleshoot slow/stuck dataflow jobs

Dataflow

Jobs [+ CREATE JOB FROM TEMPLATE](#) [+ CREATE MANAGED DATA PIPELINE](#) [ENABLE SORTING](#) [REFRESH](#) [LEARN](#)

Running [Filter](#) Filter jobs ? ||

Name	Type	End time	Elapsed time	Start time	Status	SDK version	ID	Region	Insights ?
wordcount6	Streaming		48 days 19 hr	Apr 11, 2023, 3:08:03 PM	Running	i 2.43.0	2023-04-11 12 08 02-301837046300790666	us-east1	
wordcount5	Batch	May 28, 2023, 12:31:04 PM	21 hr 30 min	May 27, 2023, 3:00:29 PM	Succeeded	i 2.46.0	2023-05-27_12_00_28-10040782164894481412	us-east1	

Troubleshooting using Logs Explorer View

Troubleshoot slow/stuck dataflow jobs

Check logs here

The screenshot displays the Google Cloud Dataflow console interface. The top navigation bar includes 'Overview', 'JOB GRAPH', 'EXECUTION DETAILS', 'JOB METRICS' (selected), 'RECOMMENDATIONS', and 'AUTOSCALING'. The left sidebar lists 'Overview', 'Jobs', 'Pipelines', 'Workbench', 'Snapshots', and 'SQL Workspace'. The main content area shows 'Data freshness' metrics with a line chart. Below the chart, the 'Logs' section is highlighted with a red box, showing 'JOB LOGS' and 'WORKER LOGS'. A red arrow points from the text 'Check logs here' to the 'Logs' section. Another red arrow points to the 'Open in Logs Explorer' button at the bottom right of the log entry.

Overview

Jobs

Pipelines

Workbench

Snapshots

SQL Workspace

JOB GRAPH EXECUTION DETAILS **JOB METRICS** RECOMMENDATIONS AUTOSCALING

▼ MORE HISTORY

Metrics < Data freshness [SAVE AS DASHBOARD](#)

▼ OVERVIEW METRICS

Data freshness

System latency

Throughput

Errors

▼ STREAMING METRICS

Backlog

Processing

Parallelism

Logs [HIDE](#) 1

JOB LOGS [WORKER LOGS](#) [DIAGNOSTICS](#)

Severity Info Filter Search all fields and values

SEVERITY	TIMESTAMP	SUMMARY
> 1	2023-02-28 23:20:18.161 EST	Operation ongoing in step ArchiveAndDeserializeToCdpMessage/Archiving/BigQueryIO.Write/StorageApiLoads/StorageApiWriteInconsistent/Write Records for at least 50m0s without outpu...
> 1	2023-02-28 23:20:18.163 EST	Operation ongoing in step ArchiveAndDeserializeToCdpMessage/Archiving/BigQueryIO.Write/StorageApiLoads/StorageApiWriteInconsistent/Write Records for at least 50m0s without outpu...
> 1	2023-02-28 23:20:18.165 EST	Operation ongoing in step ArchiveAndDeserializeToCdpMessage/Archiving/BigQueryIO.Write/StorageApiLoads/StorageApiWriteInconsistent/Write Records for at least 50m0s without outputting or completing in state finish at java.base@11.0.9/jdk.internal.misc.Unsafe.park(Native Method) at java.base@11.0.9/java.util.concurrent.locks.LockSupport.park(LockSupport.java:194) at java.base@11.0.9/java.util.concurrent.locks.AbstractQueuedSynchronizer.parkAndCheckInterrupt(AbstractQueuedSynchronizer.java:885) at java.base@11.0.9/java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedInterruptibly(AbstractQueuedSynchronizer.java:1039) at java.base@11.0.9/java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireSharedInterruptibly(AbstractQueuedSynchronizer.java:1345) at java.base@11.0.9/java.util.concurrent.CountDownLatch.await(CountDownLatch.java:232) at app/org.apache.beam.sdk.io.gcp.bigquery.RetryManager\$Callback.await(RetryManager.java:156) at app/org.apache.beam.sdk.io.gcp.bigquery.RetryManager\$Operation.await(RetryManager.java:139) at app/org.apache.beam.sdk.io.gcp.bigquery.RetryManager.await(RetryManager.java:258) at app/org.apache.beam.sdk.io.gcp.bigquery.StorageApiWriteUnshardedRecords\$WriteRecordsDoFn.flushAll(StorageApiWriteUnshardedRecords.java:664) at app/org.apache.beam.sdk.io.gcp.bigquery.StorageApiWriteUnshardedRecords\$WriteRecordsDoFn.finishBundle(StorageApiWriteUnshardedRecords.java:744) at app/org.apache.beam.sdk.io.gcp.bigquery.StorageApiWriteUnshardedRecords\$WriteRecordsDoFnDoFnInvoker.invokeFinishBundle(Unknown Source)

insertId: "2242392282109665217:154912:0:14886181"

[Open in Logs Explorer](#)

Troubleshoot slow/stuck dataflow jobs

🕒 Last 30 days 🔍 Search all fields

Dataflow Step | Log name | Severity | +1 filter | Show query

```
1 resource.type="dataflow_step"
2 resource.labels.job_id="2023-04-11_08_02-301837046300790666"
3
```

Log fields Histogram

Create metric | Create alert | Jump to now | More actions

Log fields <> Histogram

Search fields and values

RESOURCE TYPE

- Dataflow Step Clear x

SEVERITY

- Info 16
- Warning 1

LOG NAME

dataflow.googleapis.com/job-message 17

PROJECT ID

605955549251 17

JOB ID

2023-04-11_12_08_02-3018370463007 17

STEP ID

Value not present 17

JOB NAME

Query results 17 log entries

Find in results | Correlate by | Download

SEVERITY	TIMESTAMP ↑	EDT ↓	SUMMARY ✎ EDIT
🕒 This query has been updated. Run it to view matching entries. Run query			
> i	2023-05-10 00:59:40.173	EDT	Worker configuration: n1-standard-4 in us-central1-c.
> !	2023-05-10 11:05:58.046	EDT	Internal Issue (a115679b95e60023): 63963027:24112
> i	2023-05-13 06:30:12.417	EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-13 06:30:25.223	EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-22 23:18:24.798	EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-22 23:18:42.714	EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-23 03:47:00.560	EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-23 03:47:24.437	EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-26 17:25:06.517	EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-26 17:25:21.435	EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-26 22:03:39.756	EDT	Worker configuration: n1-standard-4 in us-central1-c.

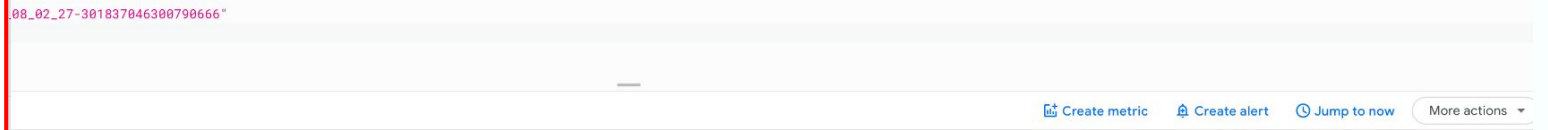
Troubleshoot slow/stuck dataflow jobs

⌂ Last 30 days 🔍 Search all fields

Relative time (Ex: 15m, 1h, 1d, 1w)

- Last 15 seconds 15s
- Last 30 seconds 30s
- Last 1 minute 1m
- Last 5 minutes 5m
- Last 10 minutes 10m
- Last 15 minutes 15m
- Last 30 minutes 30m
- Last 45 minutes 45m
- Last 1 hour 1h
- Last 3 hours 3h
- Last 6 hours 6h
- Last 12 hours 12h
- Last 1 day 1d
- Last 2 days 2d
- Last 7 days 7d
- Last 14 days 14d
- Last 30 days 30d**
- 📅 Start and end times >
- 🕒 Around a time >
- 🌐 Time zone: EDT (UTC-4) >

Dataflow Step Log name Severity +1 filter Show query



SEVERITY	TIMESTAMP ↑	EDT ▼	SUMMARY ✎ EDIT
🕒	This query has been updated. Run it to view matching entries. Run query		
> ⓘ	2023-05-10 00:59:40.173 EDT		Worker configuration: n1-standard-4 in us-central1-c.
> ⚠️	2023-05-10 11:05:50.046 EDT		Internal Issue (a115679b95e60023): 63963027:24112
> ⓘ	2023-05-13 06:30:12.417 EDT		Worker configuration: n1-standard-4 in us-central1-c.
> ⓘ	2023-05-13 06:30:25.223 EDT		Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> ⓘ	2023-05-22 23:18:24.798 EDT		Worker configuration: n1-standard-4 in us-central1-c.
> ⓘ	2023-05-22 23:18:42.714 EDT		Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> ⓘ	2023-05-23 03:47:00.560 EDT		Worker configuration: n1-standard-4 in us-central1-c.
> ⓘ	2023-05-23 03:47:24.437 EDT		Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> ⓘ	2023-05-26 17:25:06.517 EDT		Worker configuration: n1-standard-4 in us-central1-c.
> ⓘ	2023-05-26 17:25:21.435 EDT		Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> ⓘ	2023-05-26 22:03:39.756 EDT		Worker configuration: n1-standard-4 in us-central1-c.

Troubleshoot slow/stuck dataflow jobs

☰ Last 30 days 🔍 Search all fields Dataflow Step Log name Severity +1 filter Show query

```
1 resource.type="dataflow_step"
2 resource.labels.job_id="2023-04-11_08_02_27-301837046300790666"
3
```

Log fields Histogram

Log fields: Search fields and values

RESOURCE TYPE: Dataflow Step (Clear x)

SEVERITY: Info (16), Warning (1)

LOG NAME: dataflow.googleapis.com/job-message (17)

PROJECT ID: 605955549251 (17)

JOB ID: 2023-04-11_12_08_02_3018370463007

STEP ID: Value not present (17)

JOB NAME

Query results: 17 log entries

Find in results Correlate by Download

This query has been updated. Run it to view matching entries. Run query

SEVERITY	TIMESTAMP	SUMMARY
Info	2023-05-10 00:59:40.173 EDT	Worker configuration: n1-standard-4 in us-central1-c.
Warning	2023-05-10 11:05:50.046 EDT	Internal Issue (a115679b95e60023): 63963027:24112
Info	2023-05-13 06:30:12.417 EDT	Worker configuration: n1-standard-4 in us-central1-c.
Info	2023-05-13 06:30:25.223 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
Info	2023-05-22 23:18:24.798 EDT	Worker configuration: n1-standard-4 in us-central1-c.
Info	2023-05-22 23:18:42.714 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
Info	2023-05-23 03:47:00.560 EDT	Worker configuration: n1-standard-4 in us-central1-c.
Info	2023-05-23 03:47:24.437 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
Info	2023-05-26 17:25:06.517 EDT	Worker configuration: n1-standard-4 in us-central1-c.
Info	2023-05-26 17:25:21.435 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
Info	2023-05-26 22:03:39.756 EDT	Worker configuration: n1-standard-4 in us-central1-c.

Troubleshoot slow/stuck dataflow jobs

Last 30 days Search all fields Dataflow Step Log name Severity +1 filter Show query

```
1 resource.type="dataflow_step"  
2 resource.labels.job_id="2023-04-11_08_02_27-301837046300790666"  
3
```

Log fields Histogram

Log fields

Search fields and values

RESOURCE TYPE

- Dataflow Step Clear x

SEVERITY

- Info 16
- Warning 1

LOG NAME

- dataflow.googleapis.com/job-message 17

PROJECT ID

- 605955549251 17

JOB ID

- 2023-04-11_12_08_02-3018370463007

STEP ID

- Value not present 17

JOB NAME

Query results 17 log entries

Find in results Correlate by Download

This query has been updated. Run it to view matching entries. Run query

SEVERITY	TIMESTAMP	SUMMARY
> i	2023-05-10 00:59:40.173 EDT	Worker configuration: n1-standard-4 in us-central1-c.
> !	2023-05-10 11:05:50.046 EDT	Internal Issue (a115679b95e60023): 63963027:24112
> i	2023-05-13 06:30:12.417 EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-13 06:30:25.223 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-22 23:18:24.798 EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-22 23:18:42.714 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-23 03:47:00.560 EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-23 03:47:24.437 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-26 17:25:06.517 EDT	Worker configuration: n1-standard-4 in us-central1-c.
> i	2023-05-26 17:25:21.435 EDT	Your project already contains 100 Dataflow-created metric descriptors, so new user metrics of the form custom.googleapis.com/* will not be created. However, all user metri-
> i	2023-05-26 22:03:39.756 EDT	Worker configuration: n1-standard-4 in us-central1-c.

Troubleshoot slow/stuck dataflow jobs

Logs Explorer REFINE SCOPE Project SHARE LINK

You are missing one or more permissions required to use the query library. [Learn more](#)

Query Saved (0) Suggested (3) Library Clear query Stream logs

5/27/23, 3:23 PM – 5/30/23, 11:46 AM Dataflow Step Log name Severity +1 filter

```
1 resource.type="dataflow_step"
2 resource.labels.job_id="2023-04-11_08_02_27-30183704300790666"
3
```

Log fields Histogram Create

Log fields

RESOURCE TYPE Clear x

- Dataflow Step

SEVERITY

- Debug 52
- Info 10
- Error 2

Histogram 60 0 May 27, 3:00 PM May 28

Query results 65 log entries

SEVERITY	TIMESTAMP	SUMMARY	EDIT
Info		This query has been updated. Run it to view matching entries.	Run
Info		To view older entries: Extend time by: 1 day Edit time	
	2023-05-27 15:23:47.727 EDT	Autoscaling is enabled for	

Find in file

Select which logs you want to view from here:

- worker-startup
- worker
- docker & kubelet
- shuffler

Select log names Clear X

- system dataflow.googleapis.com/system
- vm-health dataflow.googleapis.com/vm-health
- vm-monitor dataflow.googleapis.com/vm-monitor
- worker dataflow.googleapis.com/worker
- worker-startup dataflow.googleapis.com/worker-startup
- CLOUD MONITORING API
- ViolationAutoResolveEvent1 monitoring.googleapis.com/ViolationAutoR...
- ViolationOpenEvent1 monitoring.googleapis.com/ViolationOpenE...

Cancel Apply

Troubleshoot slow/stuck dataflow jobs

Logs Explorer

REFINE SCOPE Project

SHARE LINK

You are missing one or more permissions required to use the query library. [Learn more](#)

Query Saved (0) Suggested (3) Library

Clear query Stream logs

5/27/23, 3:23 PM - 5/30/23, 11:46 AM Search all fields

Dataflow Step

Log name

Severity

+1 filter

```
1 resource.type="dataflow_step"
2 resource.labels.job_id="2023-04-11_08_02_27-30183704300790666"
3
```

Log fields Histogram

Log fields

Search fields and values

RESOURCE TYPE

Dataflow Step

SEVERITY

Debug	52
Info	10
Error	2

Histogram



Query results 65 log entries

SEVERITY TIMESTAMP EDT SUMMARY EDIT

This query has been updated. Run it to view matching entries.

Run query

To view older entries: Extend time by: 1 day Edit time

2023-05-27 15:23:47.727 EDT Autoscaling is enabled for job 2023-05-27 12 23 45-449341954761765565. The number of workers will be between 1 and 60.

Select log names

Clear X

Search log names

dataflow.googleapis.com/worker-startup

system dataflow.googleapis.com/system

vm-health dataflow.googleapis.com/vm-health

vm-monitor dataflow.googleapis.com/vm-monitor

worker dataflow.googleapis.com/worker

worker-startup dataflow.googleapis.com/worker-startup

CLOUD MONITORING API

ViolationAutoResolveEventV1 monitoring.googleapis.com/ViolationAutoR...

ViolationOpenEventV1 monitoring.googleapis.com/ViolationOpenE...

Cancel Apply

Troubleshooting using Job Metrics Tab

Troubleshoot slow/stuck dataflow jobs

Throughput dropping to zero

Check under "Job Metrics" tab for various metrics



Troubleshoot slow/stuck dataflow jobs

High CPU Utilization

JOB GRAPH

EXECUTION DETAILS

JOB METRICS

COST

RECOMMENDATIONS

AUTOSCALING

RESET ZOOM



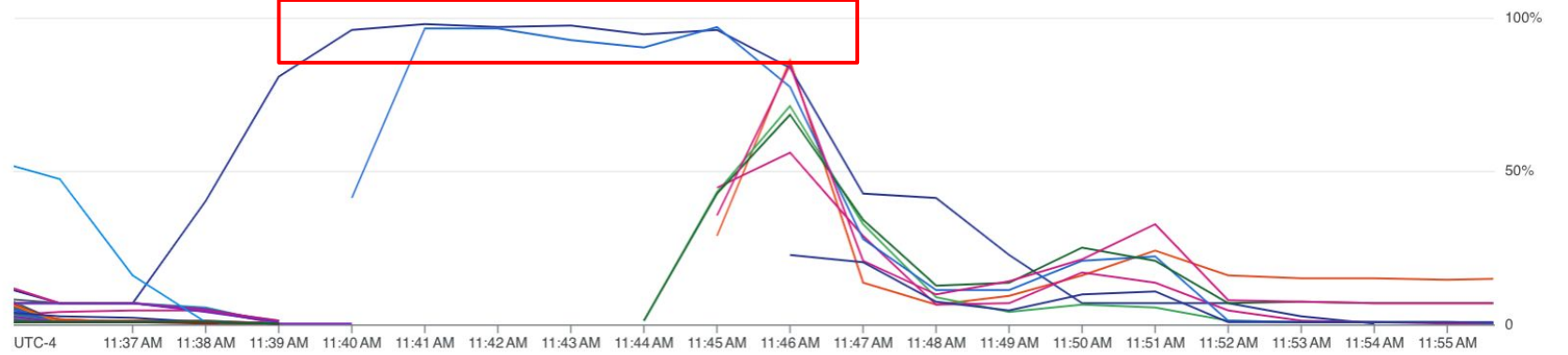
Auto Refresh

1 HOUR

CPU utilization (All Workers)



Create alerting policy



Name

Value

Troubleshoot slow/stuck dataflow jobs

High CPU Utilization

JOB GRAPH

EXECUTION DETAILS

JOB METRICS

COST

RECOMMENDATIONS

AUTOSCALING

RESET ZOOM



Auto Refresh

1 HOUR ▾

Metrics



CPU utilization

OVERVIEW METRICS

Throughput

Errors

RESOURCE METRICS

CPU utilization

Memory utilization

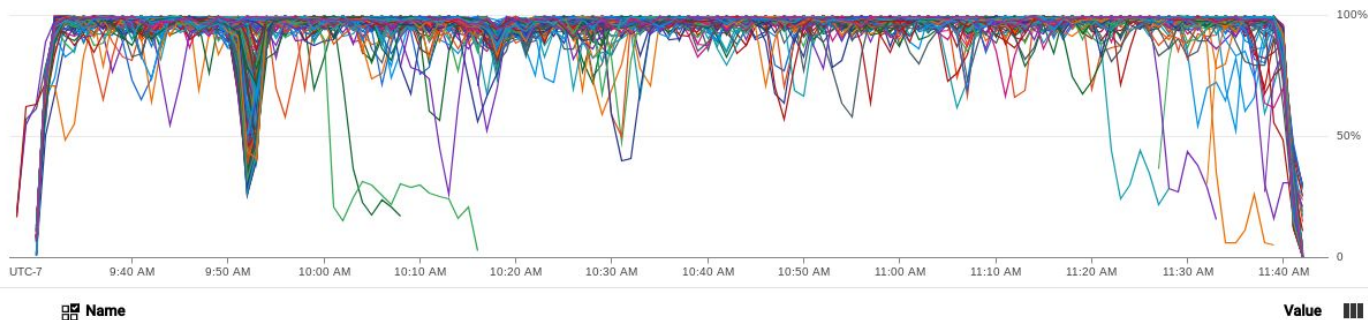
CPU utilization (All Workers) ⓘ



by instance name (mean)

1 min interval (mean)

Create alerting policy



Logs

HIDE



1



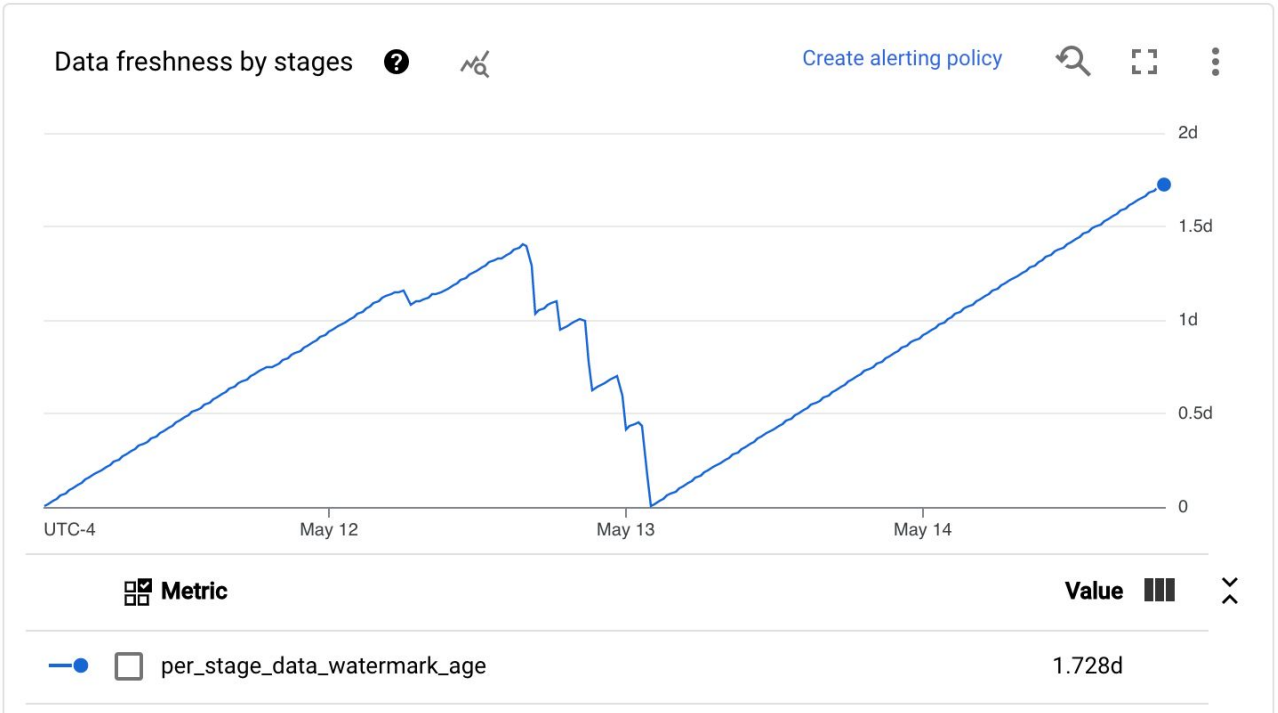
1

Troubleshoot slow/stuck dataflow jobs

Data Freshness

Data freshness

SAVE AS DASHBOARD

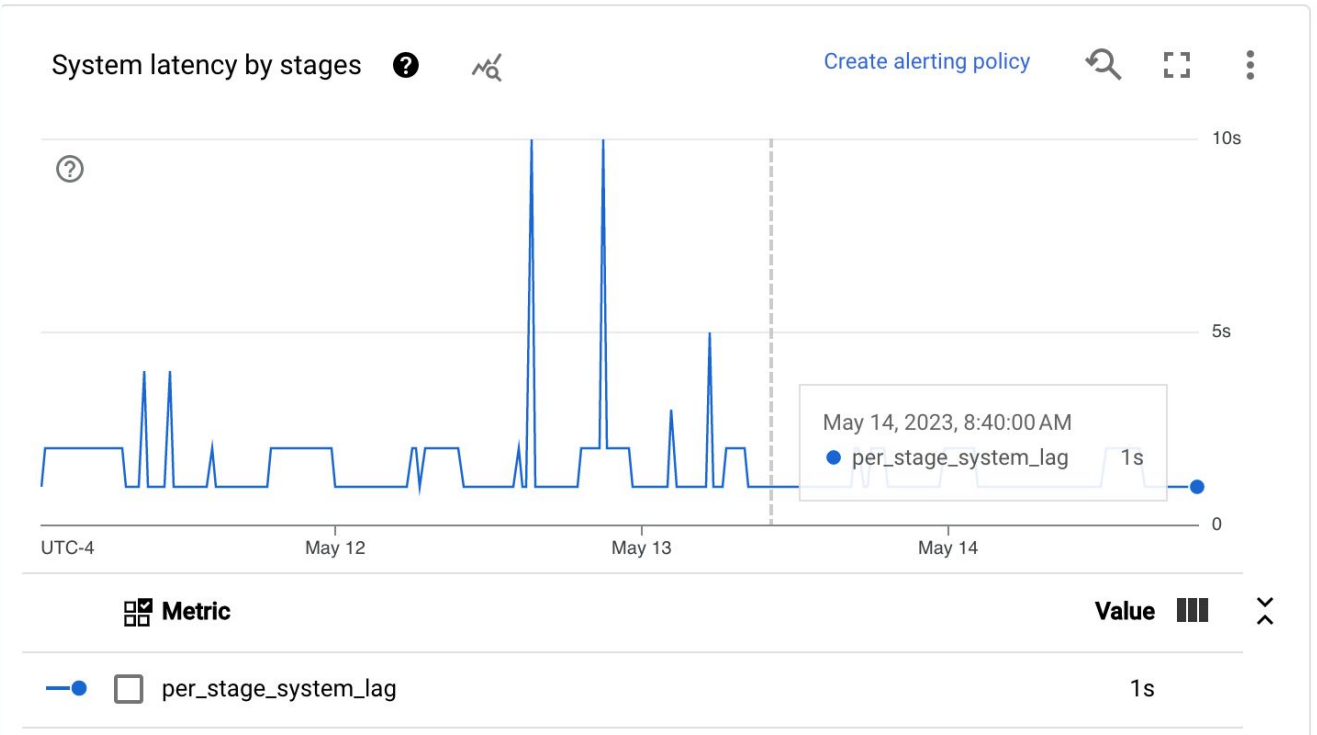


Troubleshoot slow/stuck dataflow jobs

System Latency

System latency

SAVE AS DASHBOARD



Stragglers in batch job

Troubleshoot slow/stuck dataflow jobs

When a batch job takes a long time to process data, it would be best to check on the [Straggler Workers](#)

How to check it?

Under Execution details, select Stage progress in graph view list

The screenshot displays the 'EXECUTION DETAILS' tab of a Databricks job. The 'Graph view' dropdown is set to 'Stage progress', and the 'Filter stages' dropdown shows '29 options selected'. A 'Critical path' toggle is visible. The main area shows a list of stages with progress bars. The stage 'F642 - 28 min 47 sec - 100%' is highlighted in cyan and has a warning icon (triangle) and the text '1 Straggler detected' next to it, which is circled in red. Other stages include 'F632 Progress', 'F642 Progress', 'F629 - 4 sec - 100%', 'F629 Progress', and 'F579 - 0 sec - 100%'.

Troubleshoot slow/stuck dataflow jobs

There can be various causes of stragglers:

- **Hot Keys:** Hot keys can create stragglers because they limit ability of Dataflow to process elements in parallel.
 - a. Re-key your data. Apply a ParDo transform to output new key-value pairs.

- **Re-shuffle your data** to avoid a single worker having extra load

Scenario 1: Long active user operation

Troubleshoot slow/stuck dataflow jobs

Processing Stuck/ Operation ongoing



Error

Operation ongoing in step {step name} for at least {duration}

OR

Processing stuck in step {step name} for at least {duration}

Troubleshoot slow/stuck dataflow jobs

Processing Stuck/ Operation ongoing

From Logs Explorer

Query:

Query Saved (0) Suggested (2) Library

🕒 Last 14 days 🔍 Search all fields

```
1 resource.type="dataflow_step"
2 resource.labels.job_id=$JOB_ID
3 logName:"/logs/dataflow.googleapis.com%2Fworker"
```

Results:

```
ⓘ Operation ongoing in step Write to BQ/BatchLoads/SinglePartitionWriteTables/ParMultiDo(WriteTables) for at least 02h20m00s without outputting or completing
in state finish
at java.base@11.0.9/java.lang.Thread.sleep(Native Method)
at app//com.google.api.client.util.Sleeper$1.sleep(Sleeper.java:42)
at app//com.google.api.client.util.BackOffUtils.next(BackOffUtils.java:48)
at app//org.apache.beam.sdk.io.gcp.bigquery.BigQueryHelpers$PendingJobManager.nextBackOff(BigQueryHelpers.java:162)
at app//org.apache.beam.sdk.io.gcp.bigquery.BigQueryHelpers$PendingJobManager.waitForDone(BigQueryHelpers.java:148)
at app//org.apache.beam.sdk.io.gcp.bigquery.WriteTables$WriteTablesDoFn.finishBundle(WriteTables.java:380)
at app//org.apache.beam.sdk.io.gcp.bigquery.WriteTables$WriteTablesDoFn$DoFnInvoker.invokeFinishBundle(Unknown Source)
```



Troubleshoot slow/stuck dataflow jobs

Processing Stuck/ Operation ongoing



From Logs Explorer

 Operation ongoing in step Write to BQ/BatchLoads/SinglePartitionWriteTables/ParMultiDo(WriteTables) for at least 02h20m00s without outputting or completing in state finish

```
at java.base@11.0.9/java.lang.Thread.sleep(Native Method)
at app//com.google.api.client.util.Sleeper$1.sleep(Sleeper.java:42)
at app//com.google.api.client.util.BackOffUtils.next(BackOffUtils.java:48)
at app//org.apache.beam.sdk.io.gcp.bigquery.BigQueryHelpers$PendingJobManager.nextBackOff(BigQueryHelpers.java:162)
at app//org.apache.beam.sdk.io.gcp.bigquery.BigQueryHelpers$PendingJobManager.waitForDone(BigQueryHelpers.java:148)
at app//org.apache.beam.sdk.io.gcp.bigquery.WriteTables$WriteTablesDoFn.finishBundle(WriteTables.java:380)
at app//org.apache.beam.sdk.io.gcp.bigquery.WriteTables$WriteTablesDoFn$DoFnInvoker.invokeFinishBundle(Unknown Source)
```

<https://github.com/apache/beam/blob/master/sdks/java/io/google-cloud-platform/src/main/java/org/apache/beam/sdk/io/gcp/bigquery/BigQueryHelpers.java>

Troubleshoot slow/stuck dataflow jobs

Processing Stuck/ Operation ongoing



SEVERITY	TIMESTAMP	SUMMARY	EDIT
🔍 49% of results are similar and can be hidden. Hide similar entries Preview			
> i	2023-05-23 19:04:08.886 EDT	Detected missing event columns in [REDACTED] BigQuery schema. Schema must be updated manually, if required. Dropping/Missing attributes from Event payload. Details	
> i	2023-05-23 19:04:08.886 EDT	Detected missing event columns in [REDACTED] BigQuery schema. Schema must be updated manually, if required. Dropping/Missing attributes from Event payload. Details	
> i	2023-05-23 19:04:08.886 EDT	No BigQuery job with job id beam_bq_job_LOAD_[REDACTED]_00001_00000	
> i	2023-05-23 19:04:08.886 EDT	job id beam_bq_job_LOAD_[REDACTED]_00001_00000-72 not found, so ...	
> !	2023-05-23 19:04:08.886 EDT	Load job beam_bq_job_LOAD_[REDACTED]_00001_00000-71 failed, will...	
> i	2023-05-23 19:04:08.886 EDT	Job beam_bq_job_LOAD_[REDACTED]_00001_00000-72 pending. retrying.	

Troubleshoot slow/stuck dataflow jobs

Apache Beam Issues/Feature Request



Product Solutions Open Source Pricing Search / Sign in Sign up

apache / beam Public Notifications Fork 4k Star 6.9k

Code Issues 4.1k Pull requests 208 Actions Projects Security Insights

is:issue is:open Labels 168 Milestones 2 **New issue**

4,052 Open 1,629 Closed Author Label Projects Milestones Assignee Sort

- Performance Regression or Improvement: Pytorch image classification on 50k images of size 224 x 224 with resnet 152 with Tesla T4 GPU:mean_load_model_latency_milli_secs **awaiting triage** **perf-alert**
#27077 opened 1 hour ago by github-actions [bot]
- Performance Regression or Improvement: Pytorch image classification on 50k images of size 224 x 224 with resnet 152 with Tesla T4 GPU:mean_inference_batch_latency_micro_secs **awaiting triage** **perf-alert**
#27076 opened 1 hour ago by github-actions [bot]
- [Feature Request]: BigqueryIO.java WriteTableRows RangePartitioning support **awaiting triage** **java** **new feature** **P2**
#27069 opened 6 hours ago by blakehice4 2 of 15 tasks
- [Bug]: Python KafkaIO read transform is inefficient when using the commit_offsets_in_finalize option **awaiting triage** **bug** **P2**
#27061 opened 20 hours ago by chamikaramj 15 tasks
- [Failing Test]: BigQueryIOWriteTest.testWriteFileSchemaUpdateOptionAllowFieldAddition **awaiting triage** **bigquery** **bug** **failing test** **flake** **java** **P2** **tests**
#27040 opened 2 days ago by Abacn 1 of 15 tasks
- [Bug][Go]: Metrics incremented in Setup methods are not recalled **bug** **go** **good first issue** **P3**
#27038 opened 2 days ago by lostluck 1 of 15 tasks
- [Bug]: beam.transforms.util.LogElements(with_timestamp=True, with_window=True) does not work with GlobalWindows **awaiting triage** **bug** **good first issue** **P3** **python**
#27036 opened 2 days ago by liferoad 1 of 15 tasks

Scenario 2: GC Thrashing/OOM

Troubleshoot slow/stuck dataflow jobs

GC Thrashing/OOM: Diagnostics Tab



Logs HIDE 30

JOB LOGS WORKER LOGS **DIAGNOSTICS**

Occurrences	Count	Error	First
	8	Shutting down JVM after 8 consecutive periods of measured GC thrashing. Memory is used/total/max = 7904/20103/37513 MB, GC last/max = 90.03/95.7... The worker was shut down after a long period of high memory pressure.	Dec 2022
	1	<u>StatusRuntimeException: UNAVAILABLE: keepalive watchdog timeout</u>	Jan 2023

Troubleshoot slow/stuck dataflow jobs



GC Thrashing/OOM

JOB GRAPH EXECUTION DETAILS **JOB METRICS** 💡 RECOMMENDATIONS (1)

▼ MORE HISTORY

Metrics

IK

Memory utilization

SAVE AS DASHBOARD

OVERVIEW METRICS

Throughput

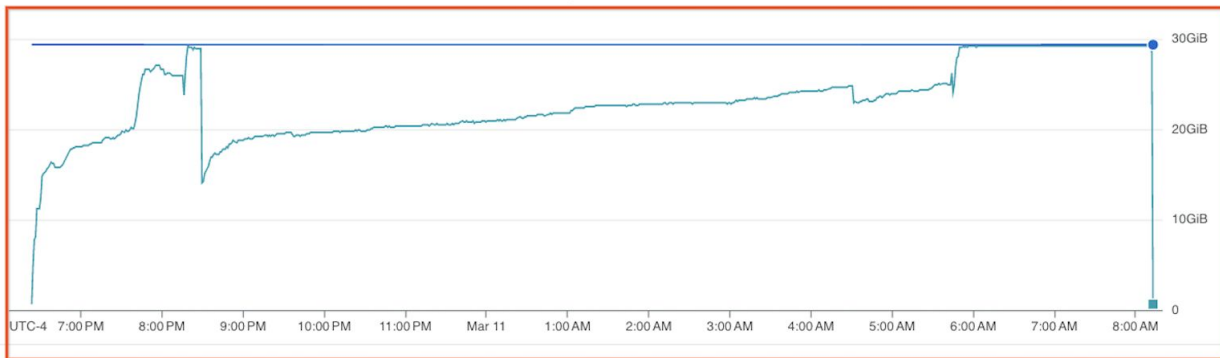
Errors

RESOURCE METRICS

CPU utilization

Memory utilization

Max worker memory utilization (estimated bytes/sec) ?



Metric

Name

Value

●

Max worker memory capacity

29.39GiB

■

Max worker memory usage

0.63GiB



General Recommendations

- Use machine types with higher memory
 - **Link: goo.gle/45USWe3**
- Decrease the parallelism of processing by reducing the number of worker harness threads
 - **Link: goo.gle/45RM6WT**
- Do vertical autoscaling (Enable Dataflow Prime)
 - **Link: goo.gle/3r3KZjv**

Performance Optimization using Dataflow profiling



- Cloud Profiler is available for Dataflow pipelines written in Apache Beam SDK for Java and Python, version 2.33.0 or later.
- It can be enabled at pipeline start time
- E.g. For Java SDK, to enable CPU profiling, start the pipeline with the following option:
`--dataflowServiceOptions=enable_google_cloud_profiler`



QUESTIONS?

mhkgupta@google.com

[linkedin.com/in/mhkgupta](https://www.linkedin.com/in/mhkgupta)

BEAM
SUMMIT

Resolving out of memory issues in Beam Pipelines

Zeeshan Khan

BEAM
SUMMIT

Benchmarking Beam pipelines on Dataflow

Pranav Bhandari

BEAM
SUMMIT