Optimize parallelism for reading from Apache Kafka to Dataflow

BEAM SUMMIT
NYC 2025

Introduction

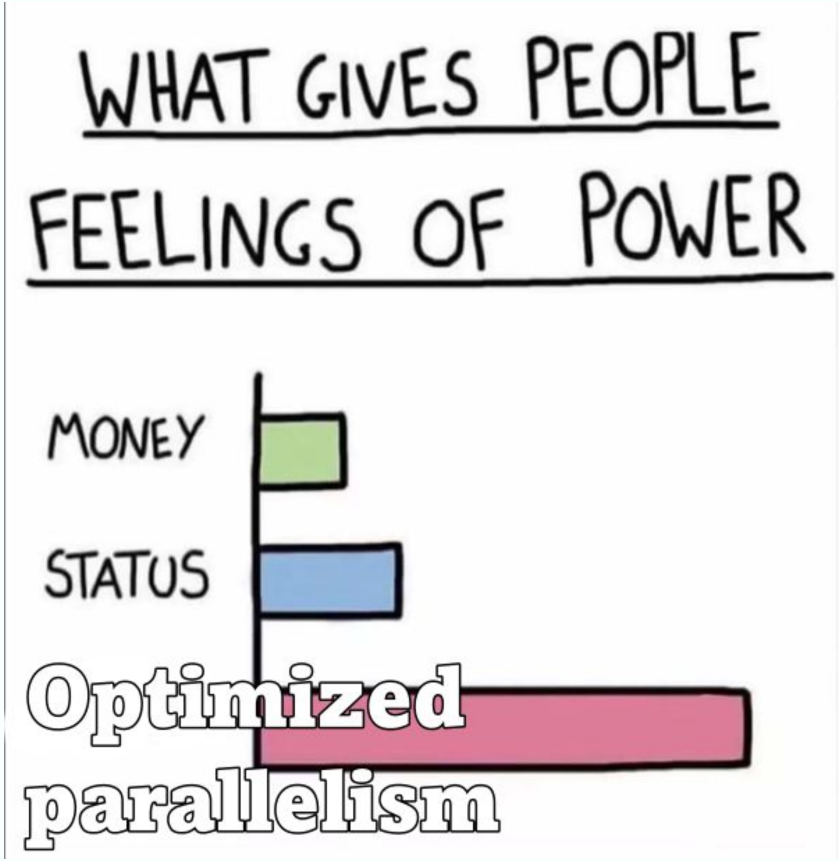
Consumption from Kafka suddenly stopped!

Messages from Kafka are being lost/dropped in the Dataflow pipeline

Most of the workers show very little CPU utilization

Supriya Koppa

Technical Support Specialist, Google Cloud

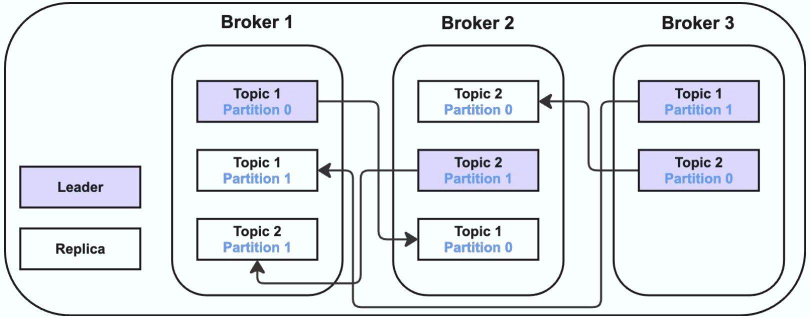


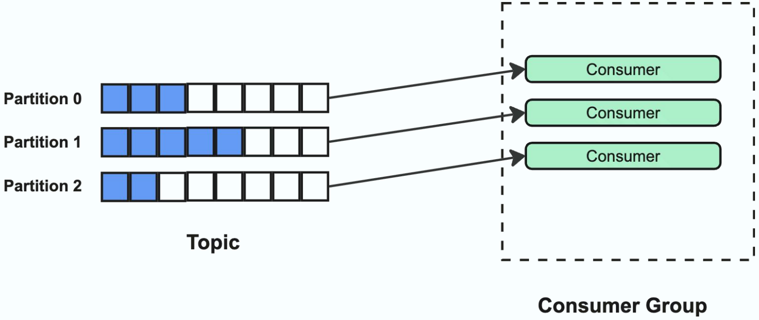
Agenda

- Introduction
- Kafka
  - Fundamentals
  - Parallelism in Kafka
- Dataflow
  - Fundamentals
  - Dataflow as a Kafka consumer
- The challenge of parallelism
  - Common complaints and attempted mitigations
  - And their symptoms
- Recommended ways to unlock parallelism
- Let’s review
- Questions… Thoughts…

Kafka fundamentals

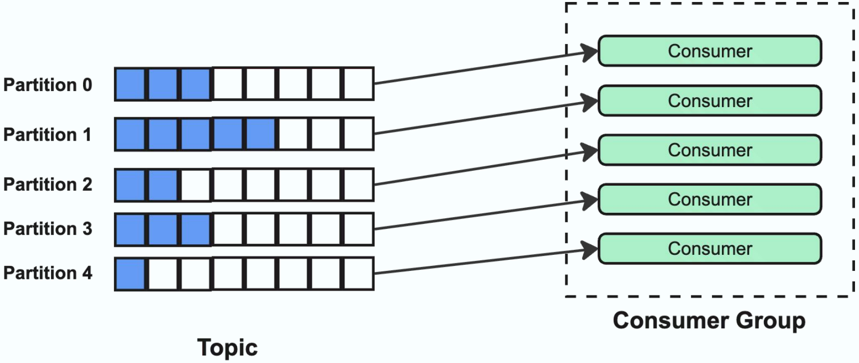
> Producers
> Kafka cluster (broker, topic)
> Kafka Topic (partition, replication)
> Consumers (consumer group, consumer offset)





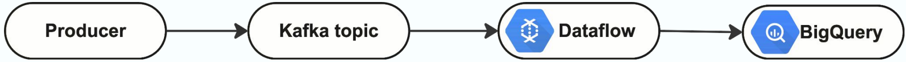
Parallelism in Kafka

> Parallelism is inherently tied to Partitions
> Generally, more the partitions, higher the processing parallelism

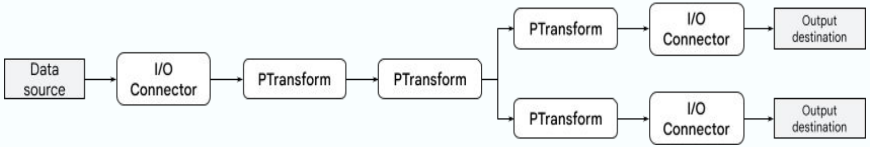


Dataflow fundamentals

Apache Beam SDK
Managed
Scalable
Stream and Batch
Exactly-once processing
GCP products
Monitoring

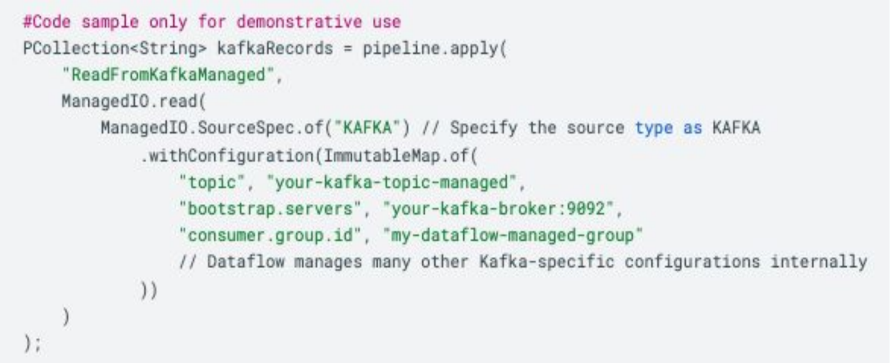


Storage
Analysis
Visualization



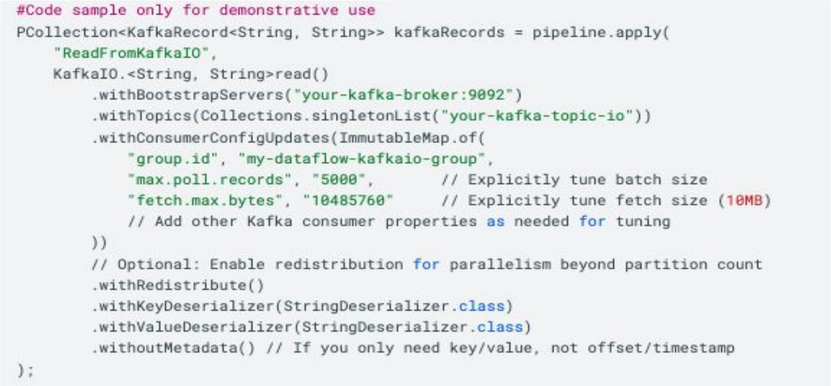
Dataflow as a Kafka consumer

> Managed I/O



```
#Code sample only for demonstrative use
PCollection<String> kafkaRecords = pipeline.apply(
    "ReadFromKafkaManaged",
    ManagedIO.read(
        ManagedIO.SourceSpec.of("KAFKA") // Specify the source type as KAFKA
            .withConfiguration(ImmutableMap.of(
                "topic", "your-kafka-topic-managed",
                "bootstrap.servers", "your-kafka-broker:9092",
                "consumer.group.id", "my-dataflow-managed-group"
                // Dataflow manages many other Kafka-specific configurations internally
            ))
    )
);
```

> KafkaIO



```
#Code sample only for demonstrative use
PCollection<KafkaRecord<String, String>> kafkaRecords = pipeline.apply(
    "ReadFromKafkaIO",
    KafkaIO.<String, String>read()
        .withBootstrapServers("your-kafka-broker:9092")
        .withTopics(Collections.singletonList("your-kafka-topic-io"))
        .withConsumerConfigUpdates(ImmutableMap.of(
            "group.id", "my-dataflow-kafkaio-group",
            "max.poll.records", "5000",        // Explicitly tune batch size
            "fetch.max.bytes", "10485760"      // Explicitly tune fetch size (10MB)
            // Add other Kafka consumer properties as needed for tuning
        ))
        // Optional: Enable redistribution for parallelism beyond partition count
        .withRedistribute()
        .withKeyDeserializer(StringDeserializer.class)
        .withValueDeserializer(StringDeserializer.class)
        .withoutMetadata() // If you only need key/value, not offset/timestamp
);
```

Challenge of parallelism

“Reading from Kafka is slow, or has stopped”

“CPU usage of Dataflow workers is skewed”

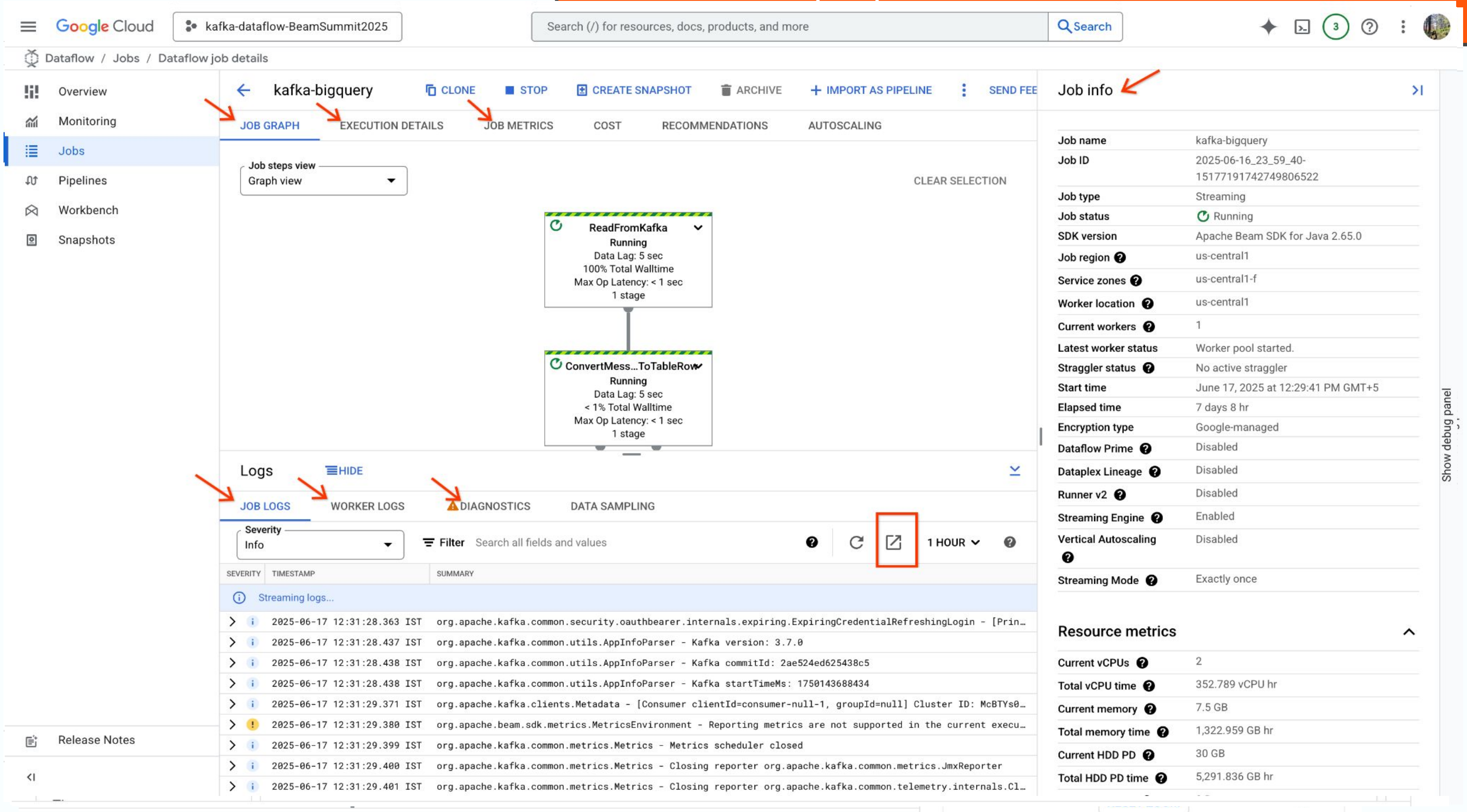
“The number of partitions was increased, but it did not resolve the problem”

Mitigations commonly attempted

“Restarting the job”

“Adding more workers, or better workers”

What the symptoms look like...



ReadFromKafka
Running
Data Lag: 5 sec
100% Total Walltime
Max Op Latency: < 1 sec
1 stage

ConvertMess...ToTableRow
Running
Data Lag: 5 sec
< 1% Total Walltime
Max Op Latency: < 1 sec
1 stage

```
SEVERITY  TIMESTAMP                    SUMMARY
Streaming logs...
>  i  2025-06-17 12:31:28.363 IST  org.apache.kafka.common.security.oauthbearer.internals.expiring.ExpiringCredentialRefreshingLogin - [Prin…
>  i  2025-06-17 12:31:28.437 IST  org.apache.kafka.common.utils.AppInfoParser - Kafka version: 3.7.0
>  i  2025-06-17 12:31:28.438 IST  org.apache.kafka.common.utils.AppInfoParser - Kafka commitId: 2ae524ed625438c5
>  i  2025-06-17 12:31:28.438 IST  org.apache.kafka.common.utils.AppInfoParser - Kafka startTimeMs: 1750143688434
>  i  2025-06-17 12:31:29.371 IST  org.apache.kafka.clients.Metadata - [Consumer clientId=consumer-null-1, groupId=null] Cluster ID: McBTYs0…
>  !  2025-06-17 12:31:29.380 IST  org.apache.beam.sdk.metrics.MetricsEnvironment - Reporting metrics are not supported in the current execu…
>  i  2025-06-17 12:31:29.399 IST  org.apache.kafka.common.metrics.Metrics - Metrics scheduler closed
>  i  2025-06-17 12:31:29.400 IST  org.apache.kafka.common.metrics.Metrics - Closing reporter org.apache.kafka.common.metrics.JmxReporter
>  i  2025-06-17 12:31:29.401 IST  org.apache.kafka.common.metrics.Metrics - Closing reporter org.apache.kafka.common.telemetry.internals.Cl…
```

Job info

Job name	kafka-bigquery
Job ID	2025-06-16_23_59_40-15177191742749806522
Job type	Streaming
Job status	Running
SDK version	Apache Beam SDK for Java 2.65.0
Job region	us-central1
Service zones	us-central1-f
Worker location	us-central1
Current workers	1
Latest worker status	Worker pool started.
Straggler status	No active straggler
Start time	June 17, 2025 at 12:29:41 PM GMT+5
Elapsed time	7 days 8 hr
Encryption type	Google-managed
Dataflow Prime	Disabled
Dataplex Lineage	Disabled
Runner v2	Disabled
Streaming Engine	Enabled
Vertical Autoscaling	Disabled
Streaming Mode	Exactly once

Resource metrics

Current vCPUs	2
Total vCPU time	352.789 vCPU hr
Current memory	7.5 GB
Total memory time	1,322.959 GB hr
Current HDD PD	30 GB
Total HDD PD time	5,291.836 GB hr

Recommended ways to unlock parallelism

Kafka

> Partitions

Golden rule

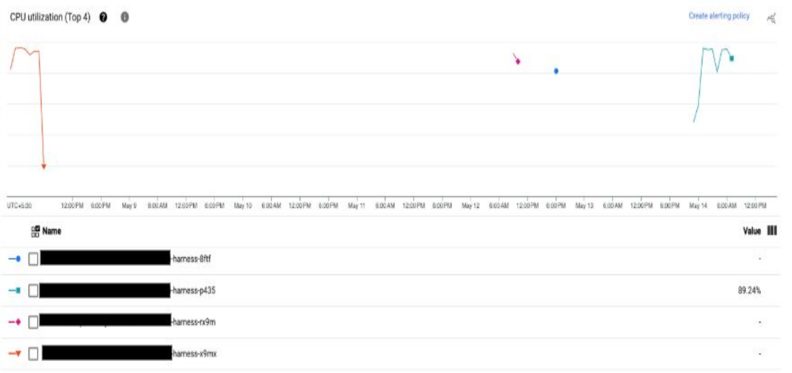
```
partitions = 2 * number_of_vCPUs
```

```
Example: n1-standard-4, 10 max workers.
partitions = 2 * 4 * 10 = 80
```

```
dataflow_fanout = 4 * max_num_workers
```

```
partitions = 4 * 10 = 40
```

> Hot partition



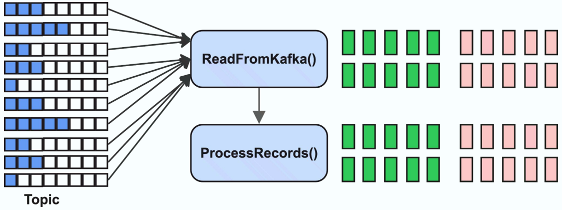
	Name	Value
● □	[redacted]-harness-8ftf	-
■ □	[redacted]-harness-p435	89.24%
◆ □	[redacted]-harness-rx9m	-
▼ □	[redacted]-harness-x9mx	-

Recommended ways to unlock parallelism

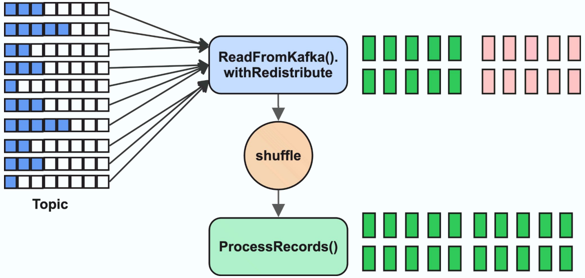
Dataflow

> Add redistribute

Replace `KafkaIO.Read`

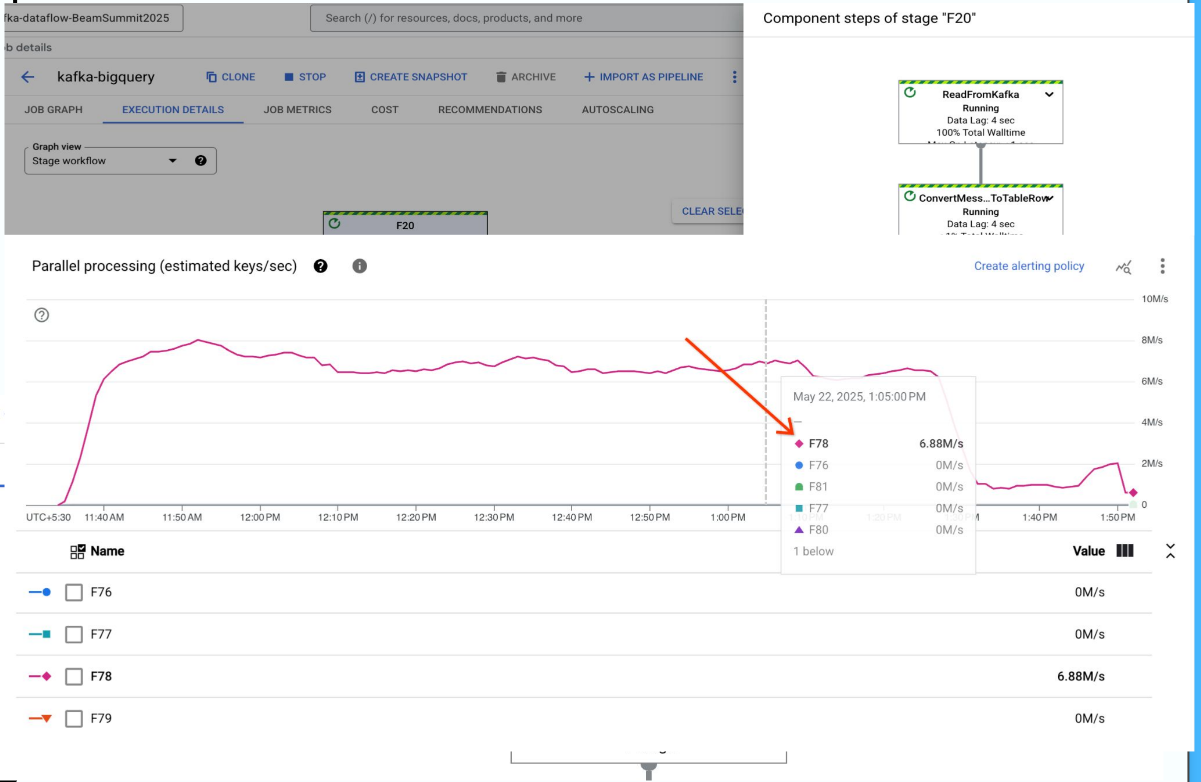


with `KafkaIO.Read.withRedistribute`



> `KafkaIO.Read.withRedistributeNumKeys`

> Autoscaling (monitoring metrics)


fka-dataflow-BeamSummit2025

Search (/) for resources, docs, products, and more

b details

← kafka-bigquery

CLONE · STOP · CREATE SNAPSHOT · ARCHIVE · + IMPORT AS PIPELINE

JOB GRAPH · EXECUTION DETAILS · JOB METRICS · COST · RECOMMENDATIONS · AUTOSCALING

Graph view: Stage workflow

CLEAR SELE

F20

Component steps of stage "F20"

ReadFromKafka
Running
Data Lag: 4 sec
100% Total Walltime

ConvertMess...ToTableRow
Running
Data Lag: 4 sec

Parallel processing (estimated keys/sec)

Create alerting policy

May 22, 2025, 1:05:00 PM

◆ F78	6.88M/s
● F76	0M/s
● F81	0M/s
■ F77	0M/s
▲ F80	0M/s
1 below	

Name	Value
□ F76	0M/s
□ F77	0M/s
□ F78	6.88M/s
□ F79	0M/s

Recommended ways to unlock parallelism

KafkaIO

```
> unboundedReaderMaxElements
> unboundedReaderMaxReadTimeMs
In DataflowPipelineDebugOptions
```

```
> max.poll.records
> fetch.max.bytes
> max.partition.fetch.bytes
In KafkaIO.Read<K,V> withConsumerConfigUpdates
```

```
> consumerPollingTimeout
In KafkaIO.Read<K,V>
withConsumerPollingTimeout
```

Let’s review

Kafka

> Partitions

Golden rule

```
partitions = 2 * number_of_vCPUs
```

```
Example: n1-standard-4, 10 max workers.
partitions = 2 * 4 * 10 = 80
```

```
dataflow_fanout = 4 * max_num_workers
```

```
partitions = 4 * 10 = 40
```

> Hot partition

Dataflow

> Add redistribute

Replace `KafkaIO.Read`
with `KafkaIO.Read.withRedistribute`

`KafkaIO.Read.withRedistributeNumKeys`

> Autoscaling (monitoring metrics)

KafkaIO

```
> unboundedReaderMaxElements
> unboundedReaderMaxReadTimeMs
In DataflowPipelineDebugOptions
```

```
> max.poll.records
> fetch.max.bytes
> max.partition.fetch.bytes
In KafkaIO.Read<K,V>
withConsumerConfigUpdates
```

```
> consumerPollingTimeout
In KafkaIO.Read<K,V>
withConsumerPollingTimeout
```

Thank you for listening!

QUESTIONS?
Thoughts?

BEAM SUMMIT
NYC 2025