Troubleshooting Slow Running Beam Pipelines

By Mehak Gupta
Google Cloud, Canada
Hello!

I’m Mehak

Technical Solutions Specialist at Google Cloud
Goals

- Apache Beam pipeline troubleshooting techniques that would empower professionals to research and resolve Beam issues by themselves.

- Self service skills would reduce MTTR (Mean Time To Recover) 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 Slow Running Beam Pipelines

Troubleshooting Workflow

- Check job errors
- View step logs

- Determine the cause

- Batch pipeline
  - Use the execution details tab to check for slow or stuck stages

- Worker code exceptions

- Streaming pipeline
  - Use the Cloud Monitoring for Dataflow view to see metrics

- Look for parallelism bottlenecks

- Find the slow or stuck stage
## Troubleshoot slow/stuck dataflow jobs

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>End time</th>
<th>Elapsed time</th>
<th>Start time</th>
<th>Status</th>
<th>SDK version</th>
<th>ID</th>
<th>Region</th>
<th>Insights</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordcount6</td>
<td>Streaming</td>
<td>Apr 11, 2023, 3:08:03 PM</td>
<td>48 days 19 hr</td>
<td>Running</td>
<td>2.43.0</td>
<td>2023-04-11-17-08-07-3018704630079566</td>
<td>us-east1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wordcount5</td>
<td>Batch</td>
<td>May 28, 2023, 12:31:04 PM</td>
<td>21 hr 30 min</td>
<td>May 27, 2023, 3:00:29 PM</td>
<td>Succeeded</td>
<td>2.46.0</td>
<td>2023-05-27-12-00-28-1004078216489481412</td>
<td>us-east1</td>
<td></td>
</tr>
</tbody>
</table>
Troubleshoot slow/stuck dataflow jobs

Troubleshooting using Logs Explorer View
Troubleshoot slow/stuck dataflow jobs

Check logs here
Troubleshoot slow/stuck dataflow jobs
Troubleshoot slow/stuck dataflow jobs

- Relative time (Ex: 15m, 1h, 1d, 1w)
- Last 30 days
  - Last 15 seconds: 15s
  - Last 30 seconds: 30s
  - Last 1 minute: 1m
  - Last 6 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)
Troubleshoot slow/stuck dataflow jobs
Troubleshoot slow/stuck dataflow jobs

Select which logs you want to view from here:

- worker-startup
- worker
- docker & kubelet
- shuffler
Troubleshoot slow/stuck dataflow jobs
Troubleshoot slow/stuck dataflow jobs

Troubleshooting using Job Metrics Tab
Troubleshoot slow/stuck dataflow jobs

Throughput dropping to zero

Check under "Job Metrics" tab for various metrics
High CPU Utilization

Troubleshoot slow/stuck dataflow jobs

CPU utilization (All Workers)
Troubleshoot slow/stuck dataflow jobs

High CPU Utilization

** JOB METRICS **
Troubleshoot slow/stuck dataflow jobs

Data Freshness

Data freshness by stages

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>per_stage_data_watermark_age</td>
<td>1.728d</td>
</tr>
</tbody>
</table>
Troubleshoot slow/stuck dataflow jobs

System Latency

System latency by stages

May 14, 2023, 8:40:00 AM
per_stage_system_lag 1s

UTC-4 May 12 May 13 May 14
0 5s 10s

Metric

- per_stage_system_lag 1s
Troubleshoot slow/stuck dataflow jobs

Stragglers in batch job
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.
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
Troubleshoot slow/stuck dataflow jobs

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:

<table>
<thead>
<tr>
<th>Query</th>
<th>Saved</th>
<th>Suggested</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last 14 days</td>
<td>Search all fields</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.Sleepers.sleep(Sleepers.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.Sleep$1.sleep(Sleeper.java:42)
at app/com.google.api.client.util.BackOffUtils.next(BackOffUtils.java:48)
at app/org.apache.beam.sdk.io.gcp.bigquery.BigQueryHelpers$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

<table>
<thead>
<tr>
<th>SEVERITY</th>
<th>TIMESTAMP</th>
<th>SEQUENCE</th>
<th>SUMMARY</th>
<th>DETAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>2023-05-23 19:04:88.886 EDT</td>
<td>1</td>
<td>Detected missing event columns in BigQuery schema. Schema must be updated manually, if required. Dropping/Missing attributes from Event payload.</td>
<td>Details: BigQuery schema. Schema must be updated manually, if required. Dropping/Missing attributes from Event payload.</td>
</tr>
<tr>
<td>i</td>
<td>2023-05-23 19:04:88.886 EDT</td>
<td>2</td>
<td>Detected missing event columns in BigQuery schema. Schema must be updated manually, if required. Dropping/Missing attributes from Event payload.</td>
<td>Details: BigQuery schema. Schema must be updated manually, if required. Dropping/Missing attributes from Event payload.</td>
</tr>
<tr>
<td>i</td>
<td>2023-05-23 19:04:88.886 EDT</td>
<td>3</td>
<td>No BigQuery job with job id beam bq_job_LOAD_00001_00000 not found, so -</td>
<td>-</td>
</tr>
<tr>
<td>i</td>
<td>2023-05-23 19:04:88.886 EDT</td>
<td>4</td>
<td>Job id beam bq_job_LOAD_00001_00000-72 not found, so -</td>
<td>-</td>
</tr>
<tr>
<td>i</td>
<td>2023-05-23 19:04:88.886 EDT</td>
<td>5</td>
<td>Load job beam bq_job_LOAD_00001_00000-71 failed, will -</td>
<td>-</td>
</tr>
<tr>
<td>i</td>
<td>2023-05-23 19:04:88.886 EDT</td>
<td>6</td>
<td>Job beam bq_job_LOAD_00001_00000-72 pending, retrying.</td>
<td>-</td>
</tr>
</tbody>
</table>
Troubleshoot slow/stuck dataflow jobs

Apache Beam Issues/Feature Request
Troubleshoot slow/stuck dataflow jobs

Scenario 2: GC Thrashing/OOM
## Troubleshoot slow/stuck dataflow jobs

### GC Thrashing/OOM: Diagnostics Tab

<table>
<thead>
<tr>
<th>Occurrences</th>
<th>Count</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>StatusRuntimeException: UNAVAILABLE: keepalive watchdog timeout</td>
</tr>
</tbody>
</table>
Troubleshoot slow/stuck dataflow jobs

**GC Thrashing/OOM**

**Max worker memory utilization (estimated bytes/sec)**

- **Max worker memory capacity**: 29.39GB
- **Max worker memory usage**: 0.63GB
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
```

---

**HashMap operations**

**Global Windowing**

**Coder operations**

**ArrayList operations**
QUESTIONS?

mhkgupta@google.com

linkedin.com/in/mhkgupta