Cost Effective Solutions for Beam pipelines in Dataflow

Sharan Teja M

BEAM SUMMIT

September 4-5, 2024

Sunnyvale, CA. USA

Agenda

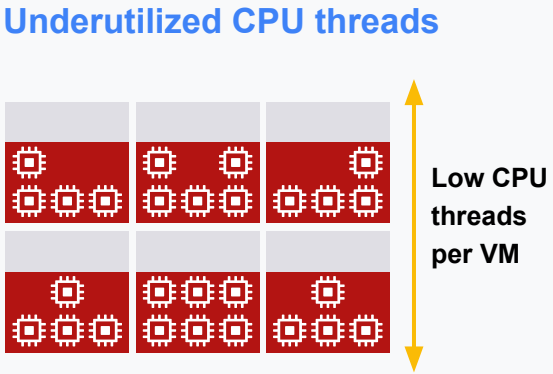
- **Introduction to optimise Dataflow pipeline costs**
- **Batch pipelines cost saving guidelines :**
  - Dynamic thread scaling
  - Right fitting
- **Streaming pipelines cost saving guidelines :**
  - Tune Horizontal Autoscaling for Streaming pipelines

Batch Pipeline Optimizations

BEAM SUMMIT

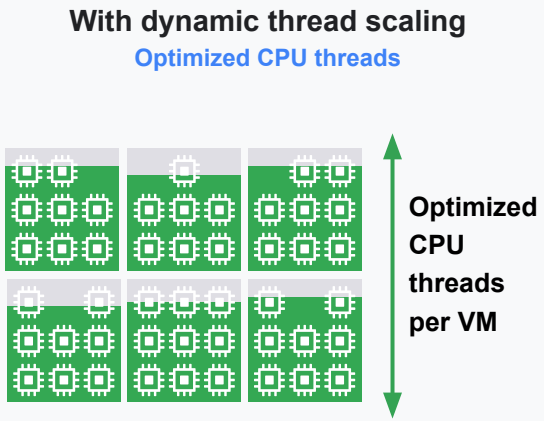
Challenge 1 - Underutilized workers

- By default, workers run one thread per vCPU.
- The analysis showed worker resources are not fully utilized.
- Underutilized CPU threads lead to inefficient spend.

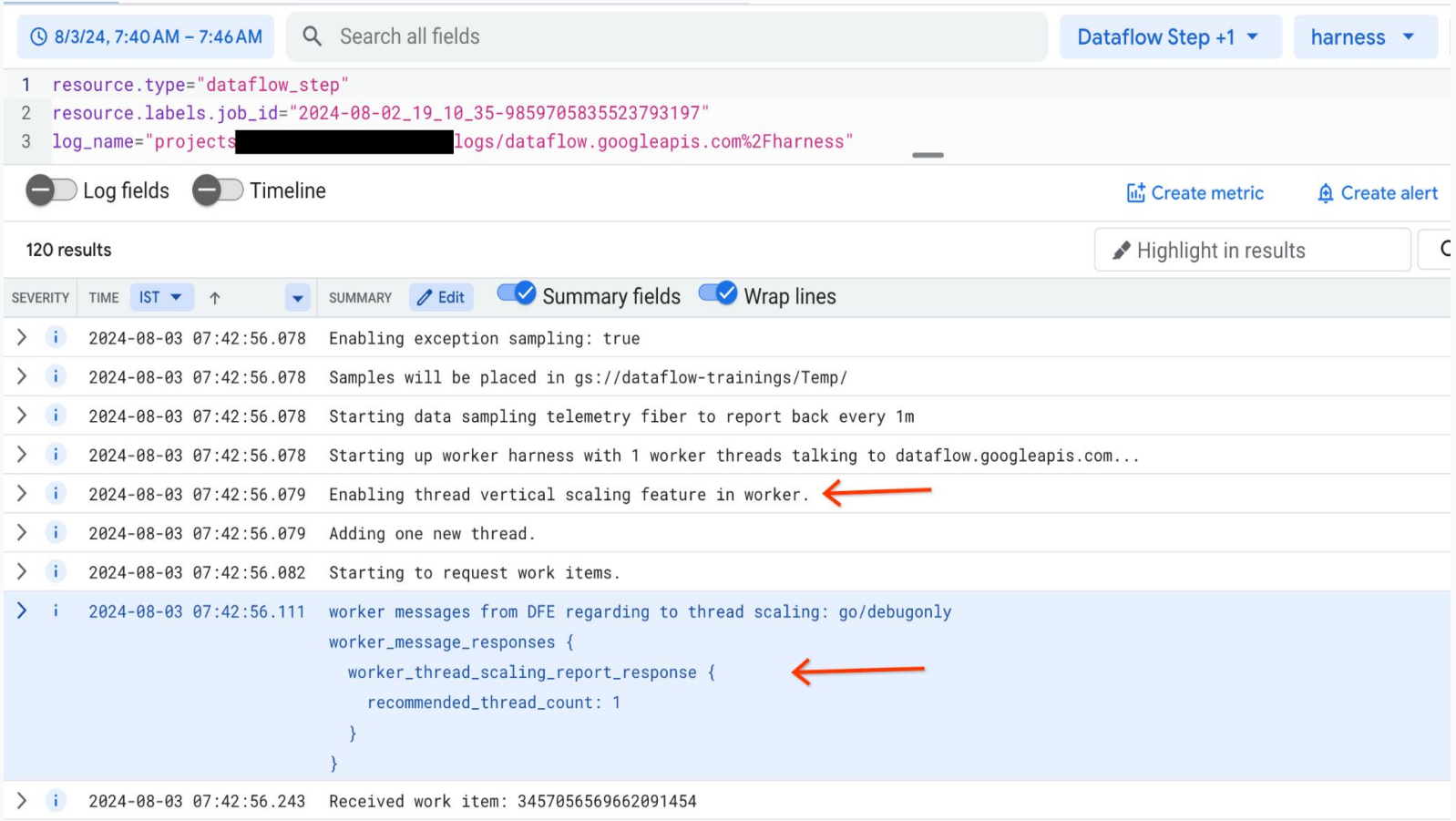


Dynamic thread scaling

- Dynamic thread scaling is an opt-in feature to automatically optimize the CPU threads per worker.
- Scales up to a maximum of two threads per vCPU when :
  - Memory utilization of the worker is less than 50%.
  - CPU utilization on the worker is less than 65%.
- Scales down to one thread per CPU when memory utilisation of worker is more than 70%.
- Can be enabled by passing "*--dataflowServiceOptions=enable_dynamic_thread_scaling*"
- More details in the documentation.



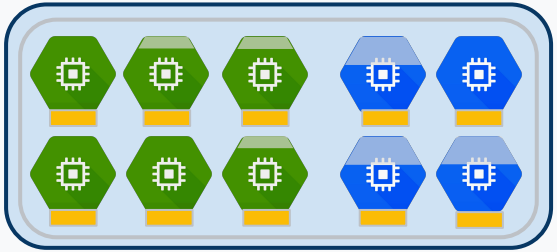
Logging


8/3/24, 7:40 AM – 7:46 AM
Search all fields
Dataflow Step +1
harness
1 resource.type="dataflow_step"
2 resource.labels.job_id="2024-08-02_19_10_35-9859705835523793197"
3 log_name="projects[redacted]logs/dataflow.googleapis.com%2Fharness"
Log fields
Timeline
Create metric
Create alert
120 results
Highlight in results
SEVERITY
TIME
IST
SUMMARY
Edit
Summary fields
Wrap lines
2024-08-03 07:42:56.078 Enabling exception sampling: true
2024-08-03 07:42:56.078 Samples will be placed in gs://dataflow-trainings/Temp/
2024-08-03 07:42:56.078 Starting data sampling telemetry fiber to report back every 1m
2024-08-03 07:42:56.078 Starting up worker harness with 1 worker threads talking to dataflow.googleapis.com...
2024-08-03 07:42:56.079 Enabling thread vertical scaling feature in worker.
2024-08-03 07:42:56.079 Adding one new thread.
2024-08-03 07:42:56.082 Starting to request work items.
2024-08-03 07:42:56.111 worker messages from DFE regarding to thread scaling: go/debugonly
worker_message_responses {
worker_thread_scaling_report_response {
recommended_thread_count: 1
}
}
2024-08-03 07:42:56.243 Received work item: 3457056569662091454

Challenge 2 - Inefficient GPU usage in ML pipelines

- While running ML pipelines, GPU's are attached to all workers in the Dataflow worker pool by default.
- Only specific stages require GPU. Hence, GPU's are not utilised optimally.

Dataflow



Stage 1
Stage 2
GPU

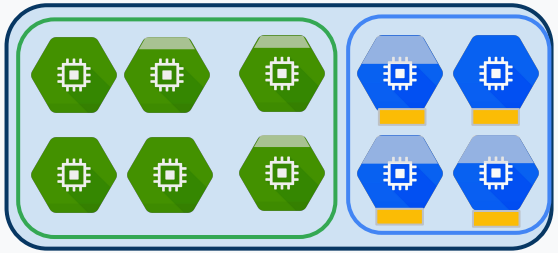
Right fitting with GPU's

- Provision customized GPUs for each stage of job.
- GPU's attached to worker pools that need it.
- Enable using ResourceHints :

```
pcoll | MyPTransform().with_resource_hints(
    min_ram="4GB",
    accelerator="type:nvidia-tesla-l4;count:1;install-nvidia-driver")
```

- More details in the documentation.

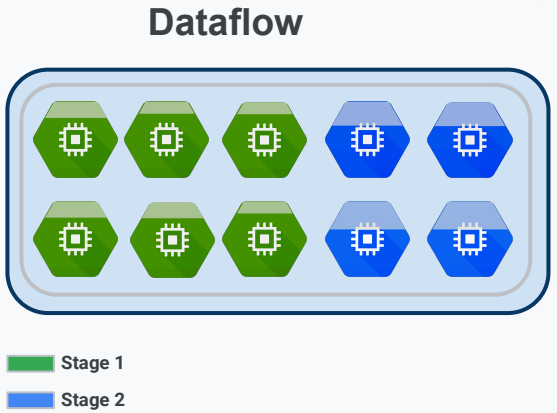
Dataflow Right fitting



Stage 1
Stage 2
GPU

Challenge 3 - Underutilized workers

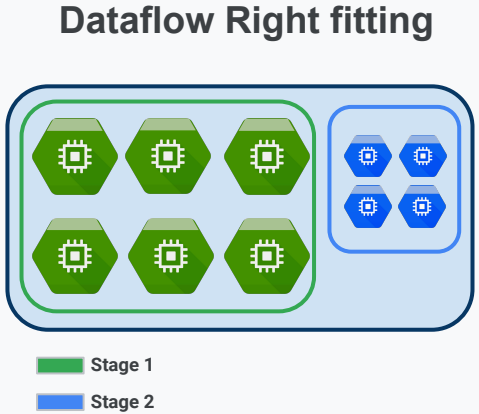
- Homogeneous workers for all the pipeline stages.
- Some workers may be under or over utilized for a stage.
- Overprovisioned resources lead to inefficient spend.



Right fitting

- Customize memory to specific pipeline steps.
- Provides additional pipeline flexibility and capability, and potential cost savings.
- Enable using ResourceHints :

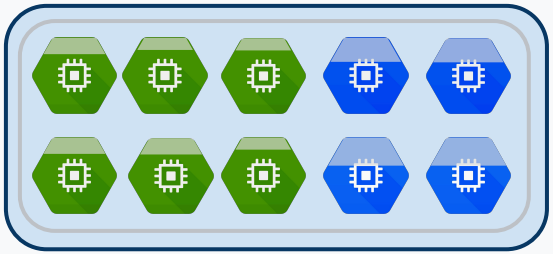
```
pcoll | beam.ParDo(BigMemFn()).with_resource_hints(
    min_ram="30GB")
```



Horizontal scaling

- Worker pools are independently auto scaled.

Dataflow



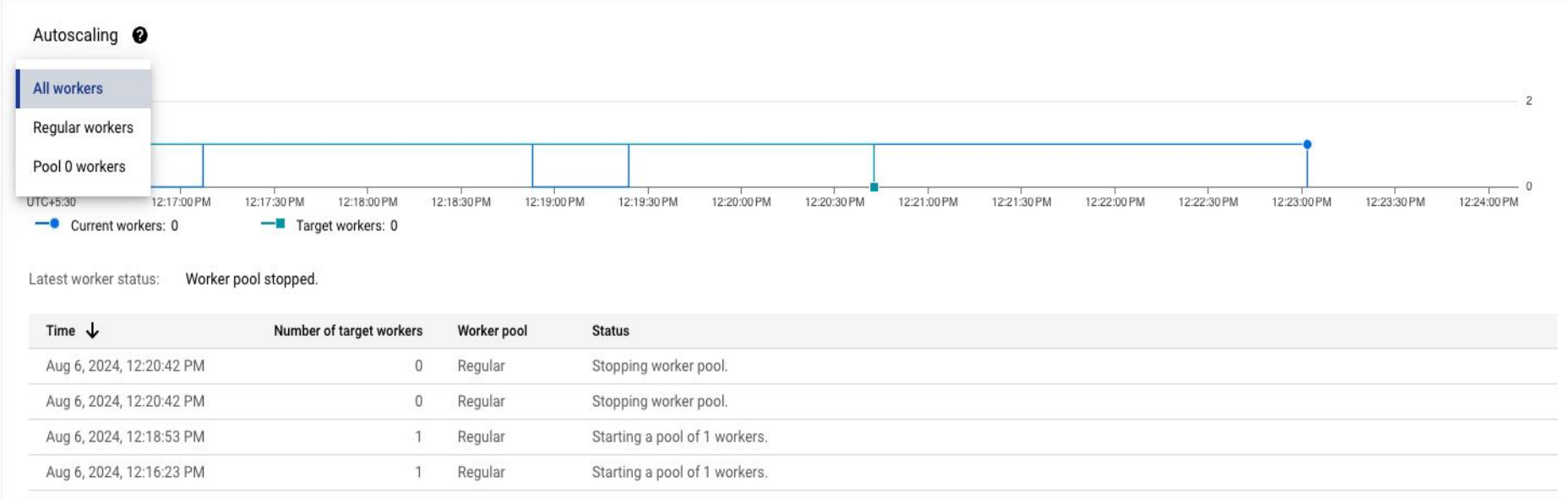
Stage 1
Stage 2

Dataflow Right fitting

Right fitting
+
Horizontal autoscaling

Monitoring

- Monitor horizontal autoscaling events of each worker pool.



Time ↓	Number of target workers	Worker pool	Status
Aug 6, 2024, 12:20:42 PM	0	Regular	Stopping worker pool.
Aug 6, 2024, 12:20:42 PM	0	Regular	Stopping worker pool.
Aug 6, 2024, 12:18:53 PM	1	Regular	Starting a pool of 1 workers.
Aug 6, 2024, 12:16:23 PM	1	Regular	Starting a pool of 1 workers.

Streaming Pipeline Optimizations

BEAM SUMMIT

Challenge 4 - Streaming pipelines tradeoff

- Tradeoff between cost and latency when streaming pipelines process high amounts of data.
- Aggressive upscaling.
- Less aggressive upscaling.

Tune Horizontal Autoscaling for streaming pipelines

- Set autoscaling range using "*--numWorkers*" and "*--maxNumWorkers*".
- Using worker_utilization_hint, target CPU utilization can be tuned in the range [0.1,0.9].
- Setting lower value :
  - Allows scaling up workers aggressively and achieves lower peak latencies.
  - May lead to higher costs.
- Setting higher value :
  - Prevents excessive upscaling at the expense of higher latency.
  - Saves resources and keep costs lower
- If the business use-cases are tolerable to higher latency, set "*--dataflow_service_options=worker_utilization_hint=X*" to higher value to save cost.
- More details in the documentation.

Thank you!

Questions?

sharantej@google.com

BEAM SUMMIT