Enrich me, if you can - A Pattern for Enriching a Stream using State and Timers

By Tobi Kaymak & Israel Herraiz
The Problem
Two Streams Need to be Joined
The “Core” one with the core info

```json
{
  "id": 123,
  "color": "gold",
  "can_dance": true
}
```
The Second one with “Lookups”

```json
{
    "id": 123,
    "serial_number": 456
}
```
Two Streams Need to be Joined

{ "id": 123,
"color": "gold",
"can_dance": true
}

{ "id": 123,
"current_serial": 456
}
Enriching Streaming Data
Enriching Streaming Data

(Slowly) updating side inputs
Enriching Streaming Data

(Batched) RPC calls
Is there another way?
State & Timers
Implementation details
Message Queues

- kafka
- Google Pub/Sub
1. Preload the Lookup Topic

(Shell) Script

BigQuery

Pub/Sub Topic Lookups
2. Start the Beam Pipeline

(Shell) Script

Core

Lookups

BigQuery
The Beam Pipeline
The Beam Pipeline
The StatefulDoFn

The input needs to be a PCollection of KV

Callback when the timer has expired

@ProcessElement
@OnTimer

Core State
Buffer
Count
Lookup State
Timer (30s)
GC-Timer (30d)
The StatefulDoFn (2)

class StatefulJoinFn(beam.DoFn):
    BUFFER_TIMER = TimerSpec('expiry', TimeDomain.WATERMARK)
    GC_TIMER = TimerSpec('gc_timer', TimeDomain.WATERMARK)

    CORE_BUFFER_BAG = BagStateSpec('core', coders.registry.get_coder(CoreType))
    CORE_COUNT_STATE = CombiningValueStateSpec('count_core', combine_fn=sum)
    LOOKUP_BUFFER_BAG = BagStateSpec('lookup', coders.registry.get_coder(LookupType))
    LOOKUP_COUNT_STATE = CombiningValueStateSpec('count_lookup', combine_fn=sum)

    def __init__(self):
        self.time_seconds = 30

    def process(self,
                input_element: Union[Tuple[str, CoreType], Tuple[str, LookupType]],
                element_timestamp=beam.DoFn.TimestampParam,
                core_count_state=beam.DoFn.StateParam(CORE_COUNT_STATE),
                core_state=beam.DoFn.StateParam(CORE_BUFFER_BAG),
                lookup_count_state=beam.DoFn.StateParam(LOOKUP_COUNT_STATE),
                lookup_state=beam.DoFn.StateParam(LOOKUP_BUFFER_BAG),
                timer=beam.DoFn.TimerParam(BUFFER_TIMER),
                gc_timer=beam.DoFn.TimerParam(GC_TIMER),
                ): [...]
Don’t miss out!

**Talk:** “Design considerations to operate a stateful streaming pipeline as a service” on Wednesday from 12:30-12:55 in Palisades with Bhupinder and Israel!

**Workshop:** “Complex Event Processing With State & Timers” on Thursday from 10:45-12:15 in Palisades with Miren and Israel!
Thank you ❤
References

- Prathap Reddy – Cache reuse across DoFn’s in Beam: https://medium.com/google-cloud/cache-reuse-across-dofns-in-beam-a34a926db848
- Chirag Shankar – Stateful Processing In Apache Beam/Cloud Dataflow: https://medium.com/google-cloud/stateful-processing-in-apache-beam-cloud-dataflow-109d1880f76a
- Iñigo San Jose – Dataflow Cookbook: https://cloud.google.com/blog/products/data-analytics/introducing-dataflow-cookbook
- Kenneth Knowles – Timely (and Stateful) Processing with Apache Beam: https://beam.apache.org/blog/timely-processing/

(CC-BY) Files by Plastic Donut from the Noun Project https://thenounproject.com/search/?q=batch&i=722276
This presentation has been designed using images from Flaticon.com
(CC 3.0 BY) Icons made by phatplus from https://www.flaticon.com/authors/phatplus
(CC 3.0 BY) Icons made by iconixar from https://www.flaticon.com/authors/iconixar
(CC 3.0 BY) Icons made by Those Icons from https://www.flaticon.com/authors/those-icons
(CC 3.0 BY) Icons made by Smashicons from https://www.flaticon.com/authors/smashicons
(CC 3.0 BY) Icons made by photo3idea_studio from https://www.flaticon.com/free-icon/fire_3163799
(CC 3.0 BY) Icons made by Icons made by Eucalyp from https://www.flaticon.com/free-icon/confidential_2857573
(CC 3.0 BY) Icons made by smalllikeart from https://www.flaticon.com/authors/smalllikeart
Do you have a Question for us?

github.com/tkaymak/beam_summit_2023_talk

Israel Herraiz
ihr@google.com

Tobi Kaymak
kaymak@google.com