Oops I *actually* Wrote a Portable Beam Runner in Go

Robert Burke
Google
lostluck.dev
Who am I?

- Robert Burke
- TL for the Beam Go SDK at Google
- Beam Committer
- Self styled Beam Go Busybody
- @lostluck {github, twitter}
- Work: Complete and Improve the Go SDK
- Play: Destiny 2, and Travel
- Canadian in Seattle
Agenda

- Briefly: State of the Go SDK 2023
- Goals of a the Runner
- What’s in a Name?
- Features of this new Runner
  - Currently, In Progress, When “Complete”
- Does it work? - Demo
- Architecture Overview
- Questions?
State of the Go SDK
State of the Go SDK 2023

- Coming in Beam Go v2.49.0
  - Timer API support
  - Local Portable Go SDK Runner

- But since last State of the Go SDK:
  - State API
  - `periodic.Impulse`, `periodic.Sequence`, and support for slowly changing side inputs.
  - `FileIO.Read` and File abstraction including `fileio.MatchContinuously`
  - `textio.ReadWithFilename`
  - Go spannerio reads/queries can now scale
  - MongoDB IO that scales
  - Cross Language
    - Automatic Python service startup
    - Python Transforms: Dataframes and Run Inference
  - Dataflow specific: Flex Templates, Cloud Profiler support
Beam Portability
Portable Worker

SDK

Runner

FnAPI
Goals
Goals of the Runner

- Local, for fast startup and ease of testing on a single machine.
- Portable, in that it uses the Beam FnAPI to communicate with Beam SDKs of any language.
- Go simple concurrency enables clear structures for testing batch and streaming jobs.
- Make it easier to develop new SDKs
  - Or new SDK features.
- Catch errors before Production through Variants
Goals of the Runner: Variants

- "default" or "test" for the common case: ensuring each DoFn in your pipeline can execute. Uses available beam features default in the SDK. No resilience to fail quickly.
- "fast" is performance focused, uses all performance beam can muster at local scale.
- Emulations like "flink" "dataflow" "spark" to which enable/disable beam features to approximate the behavior of their namesakes
  - Eg. Flink does not combiner lift.
  - Eg. Dataflow supports State Caching
- Customize a variant to your need via a pipeline option.
Naming
What’s in a Name?

sdk, default, comp, beam, local, fake, model, prism, unit, teach, portable, handlebar, universal, collab, beamgo, lens
What's in a Name?

Prism
Features of Prism - Current

- Impulse, Flatten, GBK
- DoFns
  - SideInputs (Map, Iterable), Zero or more outputs
  - Splittable DoFns, ProcessContinuations
- Combiners
  - Lifted and Unlifted
- Log collection
- Loopback mode execution (--environment_type=LOOPBACK)
- Available in the Go SDK since v2.46.0
- Metrics collection - Basics (user counter, pcoll counts & samples)
Features of Prism - In Progress

- State and Timers
- TestStream & Triggers
- Standalone Binary (first available in v2.49)
  - For executing multiple jobs
  - Basic UI for viewing progress, metrics, logs
- Supports Docker Container execution
  - Cross Language Support
- Variants
- Metrics collection - beyond basics
- WebUI
Features of Prism - When Complete

- Implements every part available in the Beam model, and makes it testable
  - ParDo Fusion
  - State Backed Iterables
  - Element Sampling
  - Drain and Cancel support
  - State Cache
  - Parameterized Windowed Values
  - PubSub IO
  - Worker Status
  - Resource Hints
  - Custom WindowFns
Demo & Tour

T1:> go install "github.com/apache/beam/sdks/v2/go/cmd/prism@latest"
T1:> prism
2023/06/13 15:54:43 INFO Serving Job Management endpoint=[::]:8073

T2:> go run *.go --runner=universal --endpoint=localhost:8073
   --environment_type=LOOPBACK --job_name="DEMO"
QUESTIONS?

@lostluck {github, twitter}
lostluck.dev
Oops I *actually* Wrote a Portable Beam Runner in Go
Appendix: See Speaker notes for links