DoFn Lifecycle & user code requirements

Israel Herraiz
Miren Esnaola

DoFn — Serialization

- A `DoFn` **must be serializable**, so the instance can be serialized in the main program and sent to the remote worker.
- A `DoFn` **can have instance variable state**. Non-transient instance variable state is serialized in the main program and then deserialized on the remote worker.
- Non-static inner classes (even anonymous ones) capture their enclosing class' instance in their serialized state, so **watch out** for `DoFns` **declared as anonymous inner classes**.

  Potential issues:

  - They can include much more than intended in the serialized state.
  - They can include things that aren't serializable.

  Solutions:

  - Define the DoFn as a named, static class.
  - Define the DoFn as an anonymous inner class inside of a static method.

DoFn — Serialization

The state is initialized...	This is suitable if the state...	This is not suitable if the state...
... using the `DoFn`’s constructor	... is known when the `DoFn` instance is created in the main program and is not too large.	... must only be used for a single bundle, as `DoFn`'s may be used to process multiple bundles.
... passing a singleton `PCollectionView` as a side input to the `DoFn`	... needs to be computed by the pipeline, or is very large and it is best read from file(s) rather than sent serialized.	
... using the method with the annotation `DoFn.StartBundle` in the `DoFn` instance	... is the same for all instances of this `DoFn` for all program executions (e.g., setting up empty caches or initializing constant data).	

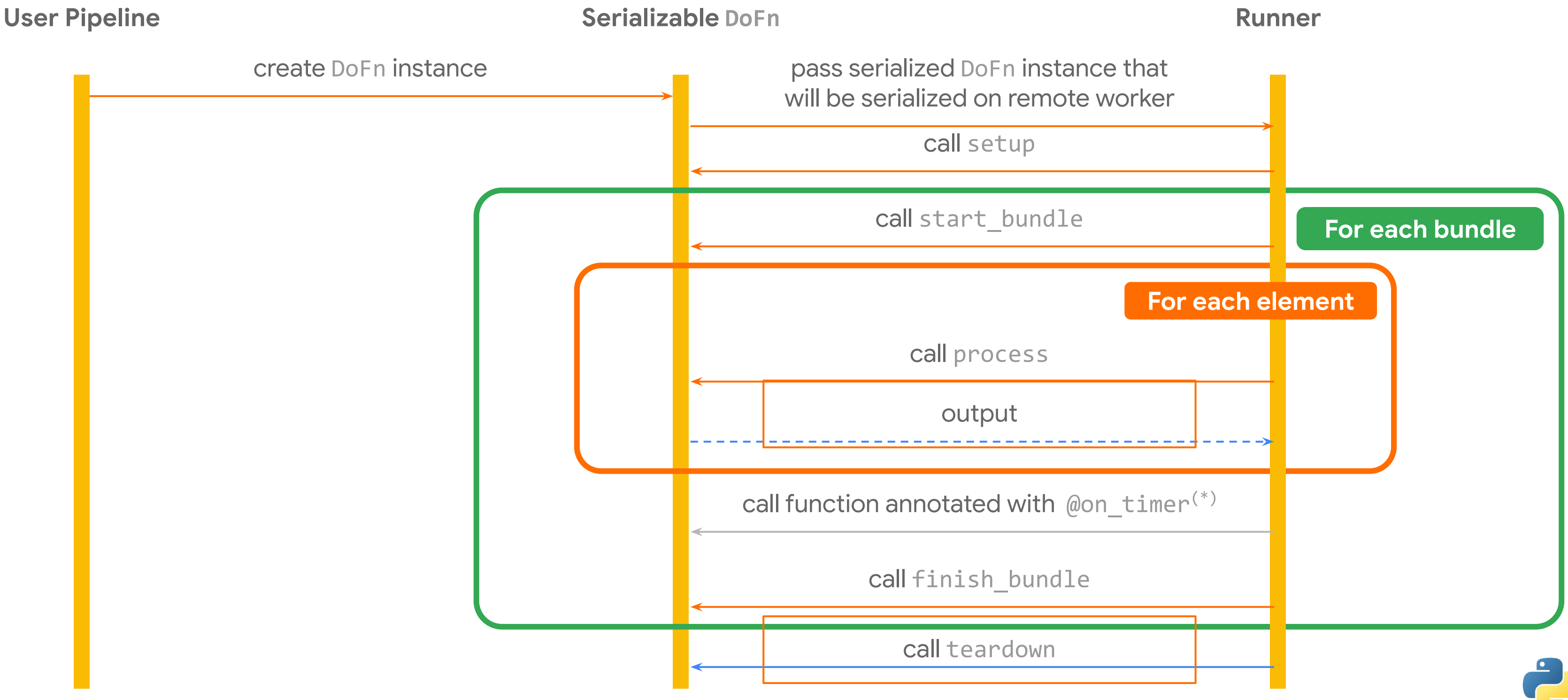
DoFn — Serialization

The state is initialized...	This is suitable if the state...	This is not suitable if the state...
... using the `DoFn`'s constructor.	... is known when the `DoFn` instance is created in the main program and is not too large.	... must only be used for a single bundle, as `DoFn`'s may be used to process multiple bundles.
... passing a side input to the `DoFn`.	... needs to be computed by the pipeline, or is very large and it is best read from file(s) rather than sent serialized.	
... using the `start_bundle`, method in the `DoFn` instance.	... is the same for all instances of this `DoFn` for all program executions (e.g., setting up empty caches or initializing constant data).	

DoFn — Thread-compatibility

- The `DoFn` should be **thread-compatible**, as each instance of a function is accessed by a single thread at a time on a worker instance.
- **Beam SDKs are not thread-safe**. If developers create their own threads in the user code, they must provide their own synchronization.

DoFn — Lifecycle



User Pipeline

Serializable DoFn

Runner

create DoFn instance

pass serialized DoFn instance that will be serialized on remote worker

call setup

For each bundle

call start_bundle

For each element

call process

output

call function annotated with @on_timer(*)

call finish_bundle

call teardown

(*) on_timer might be called more than once per bundle or can span over several bundles.

Thank you!

Questions?

BEAM
COLLEGE