DoFn Lifecycle & user code requirements

Israel Herraiz
Miren Esnaola

Friends of ParDo

ParDo	1	0, 1 or many	✓
Filter	1	0 or 1	✗
MapElements	1	1	✗
FlatMapElements	1	0, 1 or Many	✗
WithKeys	value	(f(value), value)	✗
Keys	(key, value)	key	✗
Values	(key, value)	value	✗

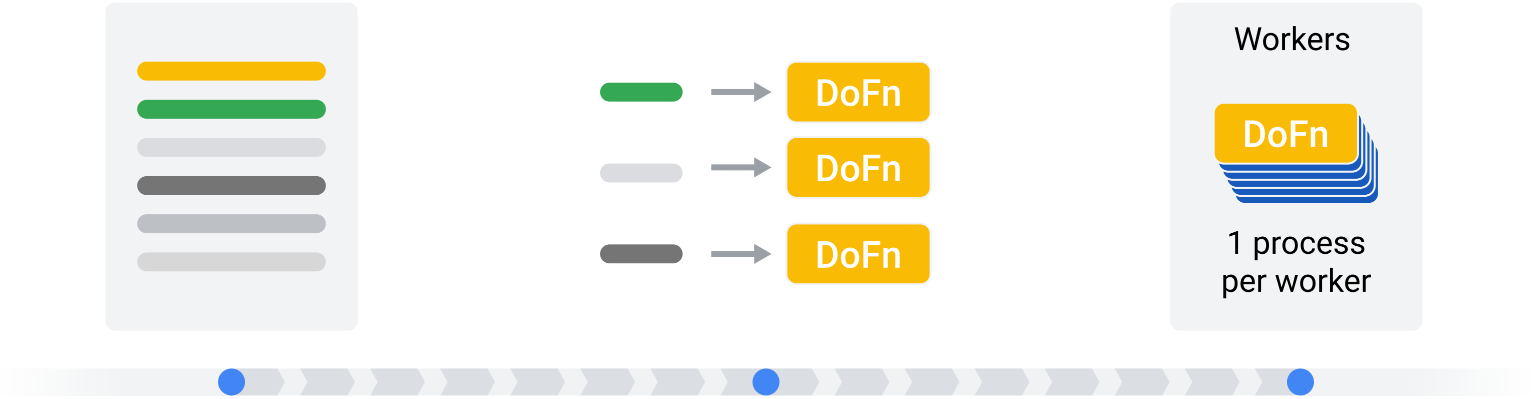
Friends of ParDo

ParDo	1	0, 1 or many	✓
Filter	1	0 or 1	✕
MapElements	1	1	✕
FlatMapElements	1	0, 1 or Many	✕
WithKeys	value	(f(value), value)	✕
Keys	(key, value)	key	✕
Values	(key, value)	value	✕

Friends of ParDo

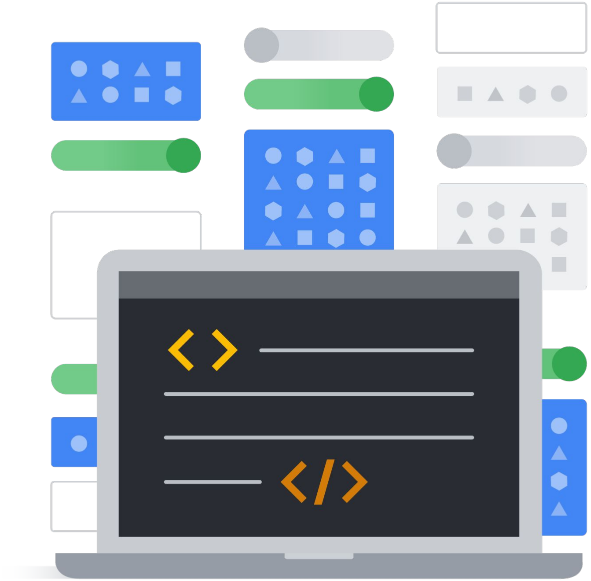
ParDo	1	0, 1 or many	✓
Filter	1	0 or 1	✗
MapElements	1	1	✗
FlatMapElements	1	0, 1 or Many	✗
WithKeys	value	(f(value), value)	✗
Keys	(key, value)	key	✗
Values	(key, value)	value	✗

Data bundles



Methods of DoFn

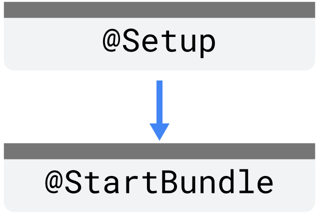
```
class MyDoFn(beam.DoFn):
    def setup(self):
        pass
    def start_bundle(self):
        pass
    def process(self, element):
        pass
    def finish_bundle(self):
        pass
    def teardown(self):
        pass
```



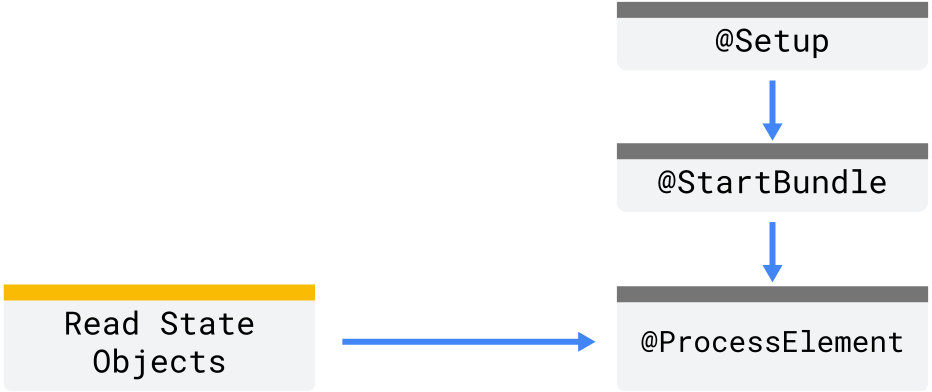
The lifecycle of a DoFn

`@Setup`

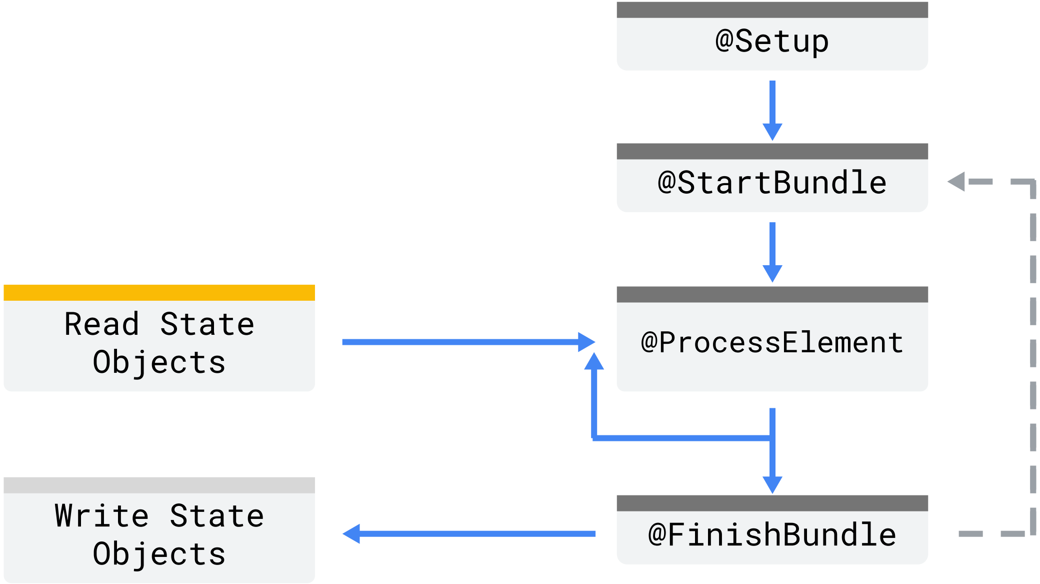
The lifecycle of a DoFn



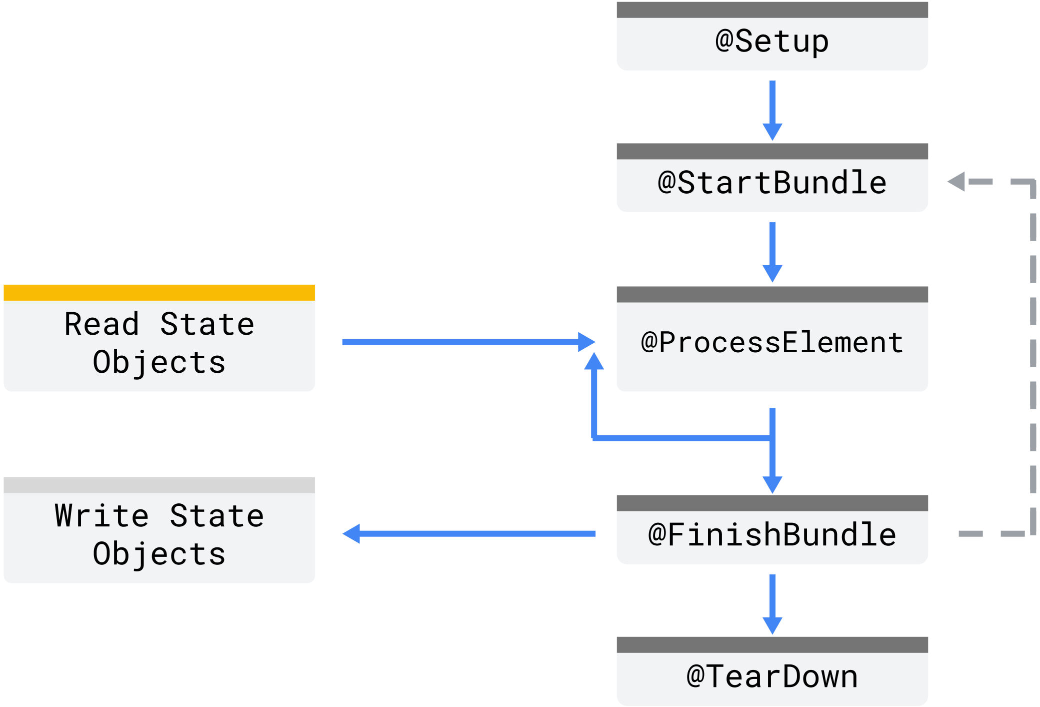
The lifecycle of a DoFn



The lifecycle of a DoFn



The lifecycle of a DoFn



Lifecycle of a DoFn

	This a good place to...	This is not a good place to...
`DoFn.Setup`	● connect to database instances ● open network connections ● start a helper process	● perform external side-effects that later need cleanup (e.g. creating temporary files on distributed filesystems, starting VMs, initiating data export jobs)
`DoFn.StartBundle`	● start keeping track of a batch of elements	
`DoFn.FinishBundle`	● do batch calls on a bundle of elements (e.g. running a database query)	
`DoFn.Teardown`	● close database connections ● close network connections ● shut down a helper process	● flush a batch of buffered records to a database ● delete temporary files on a distributed filesystem

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.

Thank you!

Questions?