Using Dead Letter Queues with Beam

John Casey
Google

BEAM SUMMIT

September 4-5, 2024
Sunnyvale, CA. USA

Beam And Error Handling

Beam doesn’t handle Errors

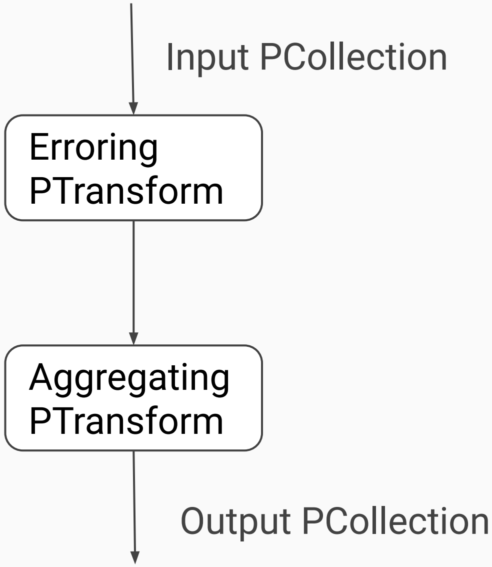
- Errors propagate to the Runner
- The Runner does whatever it likes with errors

Dataflow *does* handle Errors

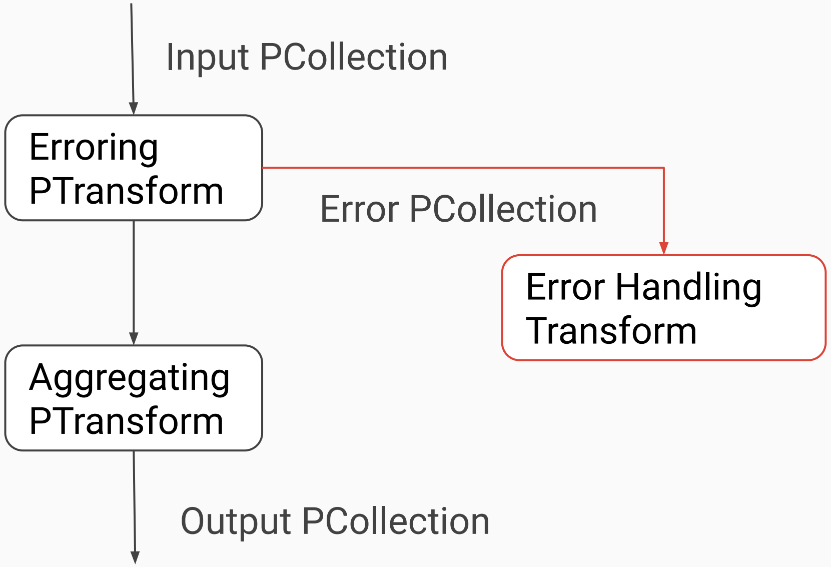
- In Batch, an error is retried 4 times
- In Streaming, an error is retried unlimited times

Unlimited Retrying can Poison Pipelines

Poisoned Pipeline



UnPoisoned Pipeline



Not all Errors are Bad Records

Handling Bad Records in With ErrorHandler

```
PCollection<?> records = ...
try (ErrorHandler errorHandler = pipeline.registerErrorHandler(KafkaIO.write(...))){
    records.apply(BigQueryIO.write(...).withBadRecordHandler(errorHandler));
}
```

Writing to Multiple Sinks

```
PCollection<?> records = ...
try (ErrorHandler errorHandler = pipeline.registerErrorHandler(KafkaIO.write(...))){
    records.apply(BigQueryIO.write(...).withBadRecordHandler(errorHandler));
    records.apply(BigtableIO.write(...).withBadRecordHandler(errorHandler));
}
```

Error in a Source

```
PCollection<KafkaRecord> records;
try (ErrorHandler errorHandler = pipeline.registerErrorHandler(FileIO.write(...))){
    records = pipeline.apply(KafkaIO.read(...).withBadRecordHandler(errorHandler)
}
records.apply(...);
```

What is an Error Handler?

```
PTransform<PCollection<BadRecord>,?>
```

What is a BadRecord?

```
COMPOSITE record {
    STRING          humanReadableRecord
    Nullable BYTES  encodedRecord
    Nullable STRING coder
}
COMPOSITE error {
    Nullable STRING exception
    Nullable STRING exceptionStacktrace
    STRING          description
}
```

IOs with Error Handling Support

- `KafkaIO`
- `PubsubIO`
- `BigtableIO`
- `BigQueryIO`
- `File, Text, and AvroIO`

Adding Error Handling to your Transforms

Handling an Exceptioning DoFn

```
public SomeDoFn extends DoFn<Type,String> {

@ProcessElement
public void processElement(ProcessContext c){
        String formattedRow;
        try {
          formattedRow = formatter.format(c.element());
        } catch (FormatException e) {
          log.error("Format Exception in SomeDoFn");
          badRecordHandler.route(c, c.element(), e,
            "SomeDoFn failed to format the input record.");
         return;
        }
        c.output(formattedRow);
}}
```

Handling an Exceptioning DoFn

```
public SomeTransform extends PTransform<PCollection<A>,PCollection<B>>{

public PCollection<B> expand(PCollection<A> collection){
    PCollectionTuple pCollectionTuple =
        input.apply(
            ParDo.of(new SomeDoFn(badRecordRouter))
           .withOutputTags(RECORDS,TupleTagList.of(BAD_RECORD_TAG)));

    errorHandler.addErrorCollection(
        pCollectionTuple
            .get(BAD_RECORD_TAG)
            .setCoder(BadRecord.getCoder(input.getPipeline())));

    return pCollectionTuple.get(RECORDS);
}}
```

Thank you!

Questions?

John Casey

theotherjohn@google.com

johncasey@apache.org

https://s.apache.org/beam-dlq-design