

BEAM
SUMMIT

How to Write a Beam IO

- Team Lead for Beam IOs at Google
- theotherjohn@
- Work: Focus on Kafka IO
- Hobbies: Hiking, Running and 40k

What is an IO?

An IO is a transform where the primary purpose of the transform is reading data from or writing data to some external resource.

```
public class BadIO extends PTransform<PBegin, PCollection<String>>{

    private final String filename;

    public BadIO(String filename){
        this.filename = filename;
    }

    @Override
    public PCollection<String> expand(PBegin input) {
        return input.apply(Impulse.create())
            .apply(ParDo.of(new BadDoFn(filename)));
    }

    public static class BadDoFn extends DoFn<byte[],String> {
        private final String filename;

        public BadDoFn(String filename){
            this.filename = filename;
        }

        @ProcessElement
        public void processElement(OutputReceiver<String> receiver)
            throws FileNotFoundException {
            Scanner scanner = new Scanner(new File(filename));
            while (scanner.hasNext()){
                receiver.output(scanner.next());
            }
        }
    }
}
```

How to Write a *Good* Beam IO



Agenda



- ~~What is an IO?~~
- Challenges in Big Data IO
- Splittable DoFns and Reading Data
- Watermarks
- Writing Data
- What is a Good IO?
- Questions
- Further Reading

- Parallelizing Reads from a Single Datasource
- Handling of inevitable exceptions
- Unbounded Reads
 - Checkpointing
 - Sparse Data

IO Design

Bad News: There is no easy way to extend or copy code to make an IO.

Every source or sink requires its own design.



- Base DoFns don't work
- Transform 1 input element to unbounded output elements
- Solves 3 main problems:
 - Parallel Reads
 - Unbounded Reads
 - Dynamic Reads

- Impulse
- Poll Kafka Partitions (Regular DoFn)
 - Solves Parallelism
- Partitions -> Kafka Records (SDF)
 - RestrictionTracker tracks record offsets
 - ProcessContinuation.resume() -> More work to be done
 - ProcessContinuation.stop() -> This tracker is complete
 - Solves Unbounded Read & Dynamic Read

- *When* is your data?
- Watermark is calculated from upstream watermarks.
- Sourcing your timestamps:
 - Wall Time
 - Data timestamp
- Sparse Data

- How can I parallelize data?
 - Topics, partitions, etc.
- How can I track data?
 - Rows, offsets, etc.
- What configurations should I expose to the users?
- What should the default configurations be?

- No Special Beam constructs
- Writes have a higher risk of data loss bugs being triggered
 - Remote Calls
 - Data leaving runner ecosystem

- What exceptions can I receive, and which ones can I retry?
 - Schema mismatch vs. Quota exceeded
- What information can I pass downstream in the pipeline?
- How can I batch writing data?
- What configurations should I expose to the users?
- What should the default configurations be?

- Following IO convention
- Supporting Cross Language IOs
 - Right Now! In Upper Bay
 - Tomorrow at 14:00 and 14:30 in Upper Bay
- Integrating with Performance Testing
 - Thursday at 17:15 in Palisades

John Casey

QUESTIONS?

theotherjohn@google.com
Github: johnjcasey

Further Reading

- [How to Write a Beam IO](#)
- [Splittable DoFns](#)
- [Multi-Language Pipelines](#)
- [IO Standards Doc \(Wip\)](#)

Text with horizontal image





Lorem ipsum dolor sit amet, ignota utamur hendrerit ne his, modo signiferumque nam id. Eos ne choro virtute posidonium. Eu everti philosophia per. Legere theophrastus eu vim, sit decore omittam definiebas ne. Id diam dolorum definiebas vix, dignissim pertinacia adolescens eu vim.

Text and vertical image

The image shown is just a sample, replace with your own.

Subtitle 1

Some text

Subtitle 2

Other text