Applying Transformations to Streaming Data

Janani Ravi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

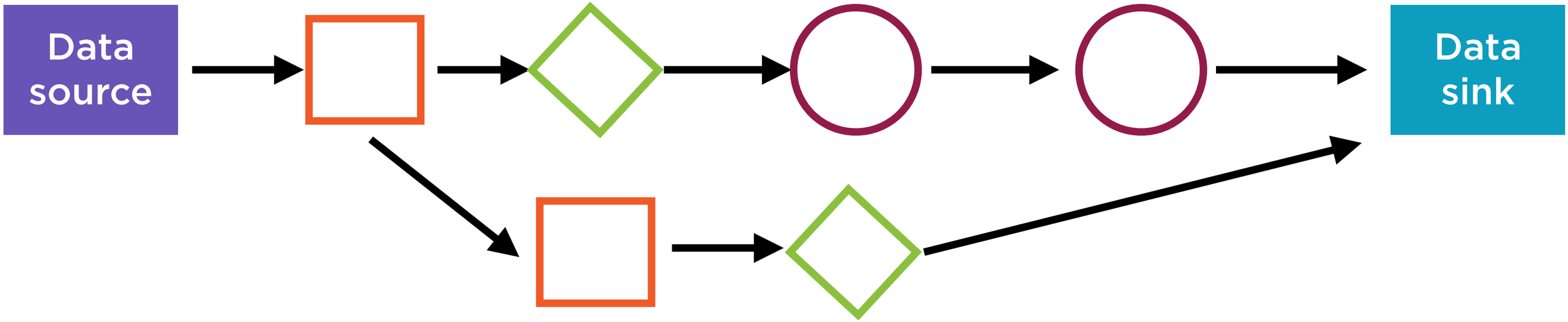
Core Beam transformations

ParDo, GroupByKey, CoGroupByKey, Combine, Flatten, Partition

Applying core transformation to input data

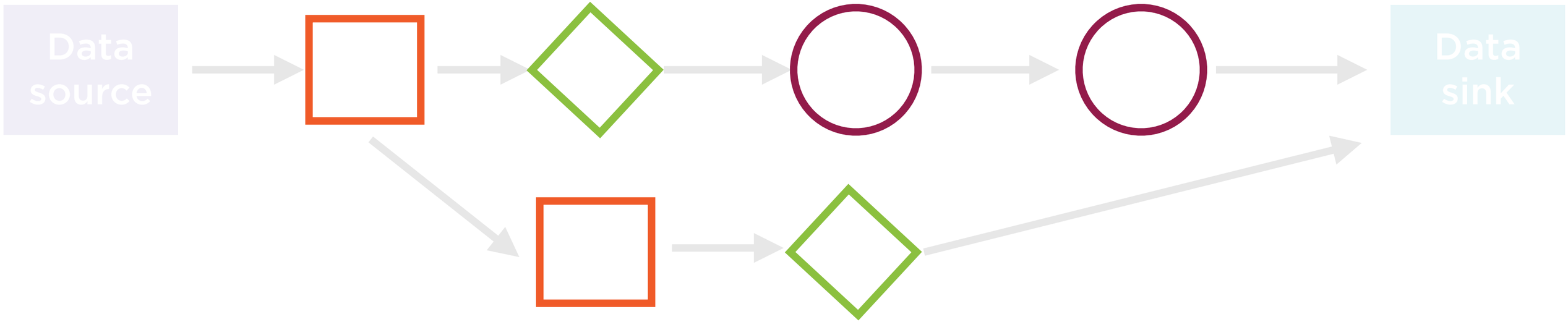
Requirements for writing user code for Beam transforms

Pipeline



Pipeline: Entire set of computations

PTransform



PTransforms: Nodes in DAG

PTransform

Interface in the Beam SDK; represents single step of the pipeline that takes in an input PCollection and transforms it to zero or more output PCollections.

Core Transforms

Core Transforms in Apache Beam

ParDo

GroupByKey

CoGroupByKey

Combine

Flatten

Partition

Core Transforms in Apache Beam

ParDo	GroupByKey	CoGroupByKey
Combine	Flatten	Partition

ParDo

Similar to Map in Map-Reduce

Transforms each element of input PCollection

Emits zero, one, or more elements for each input element

Need to specify a DoFn object

- Beam SDK object encapsulating operation

ParDo

Filter elements based on condition

Format or perform type conversion

Extract parts of each element

Perform computations on each element

ParDo

In addition to main input, can also specify side inputs

Side inputs are additional inputs accessed each time element is processed

Useful to inject additional data at runtime based on each element value

Side inputs are complex to use with windowing - restrictions apply

Core Transforms in Apache Beam

ParDo

GroupByKey

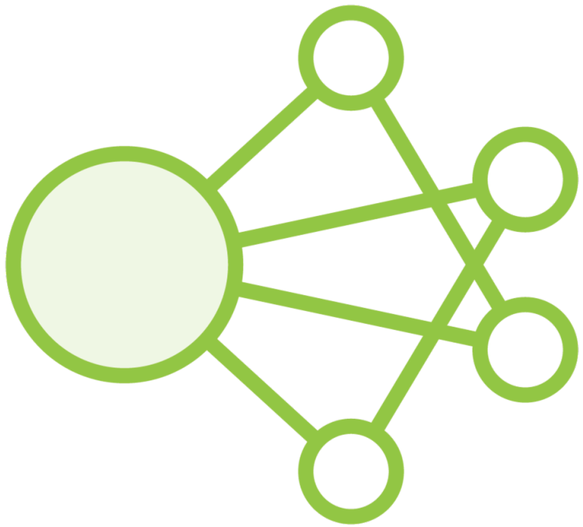
CoGroupByKey

Combine

Flatten

Partition

GroupByKey



Similar to Shuffle step in Map-Reduce

Process key-value pairs

Input is a multi map

- Multiple pairs have same key, but different values

Can use GroupByKey to collect all values associated with each unique key

Core Transforms in Apache Beam

ParDo

GroupByKey

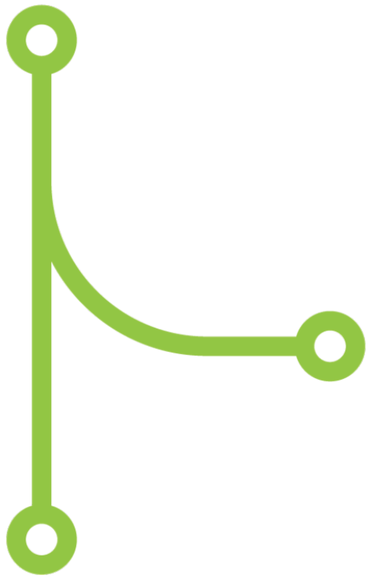
CoGroupByKey

Combine

Flatten

Partition

CoGroupByKey



Performs relational join of two or more key-value pairs

Input is tuple of keyed PCollection objects

Inputs must have same key type

Restrictions apply on CoGroupByKey and unbounded PCollections

- Same as those on GroupByKey

Core Transforms in Apache Beam

ParDo

GroupByKey

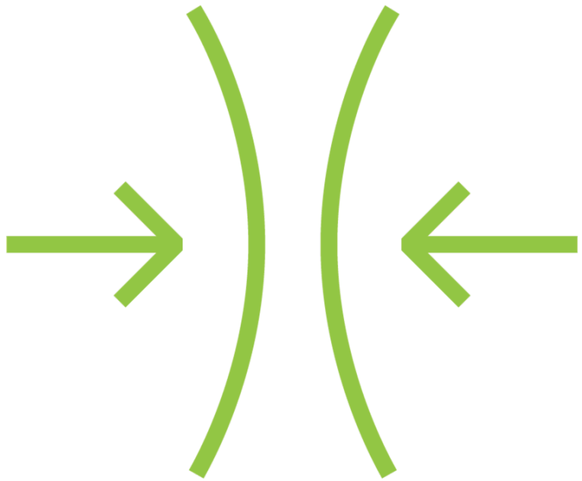
CoGroupByKey

Combine

Flatten

Partition

Combine



Combines collections of elements or values

Some variants work on entire PCollections

Other variants combine values for each key in keyed input PCollections

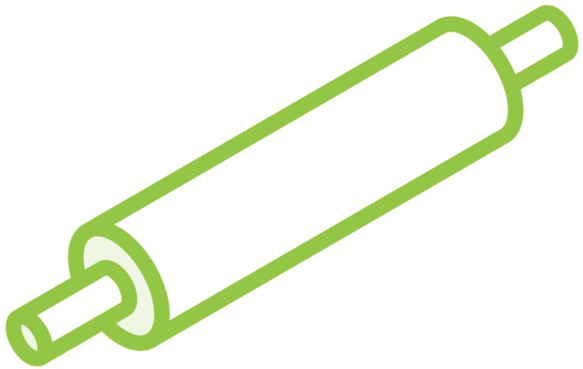
Can use to perform simple operations such as sum

Can use to create complex combine functions

Core Transforms in Apache Beam

ParDo	GroupByKey	CoGroupByKey
Combine	**Flatten**	Partition

Flatten



Merges multiple PCollection objects into a single logical PCollection

Operates on input PCollections of same type

Core Transforms in Apache Beam

ParDo

GroupByKey

CoGroupByKey

Combine

Flatten

Partition

Partition

Splits a single PCollection into a fixed number of smaller collections

Divides based on partition function specified in code

Number of partitions must be known at graph construction time

- Can not change this based on data, or in middle of pipeline
- Can specify as command-line argument

Demo

Executing transforms using ParDo and DoFn

Demo

Grouping data using GroupByKey

Demo

Joining data using CoGroupByKey

Demo

Using Combine transforms to aggregate values

Demo

Using Flatten transforms to merge PCollections

Demo

Using Partition transforms to partition PCollections

Demo

Using a Composite transform to perform multiple simple transforms

User Code Requirements for Transforms

Transform Code Requirements

Beam transforms are executed in a distributed manner

Multiple copies of the function run on different machines on the cluster

Function copies do not communicate or share data

Functions may be retried on failure

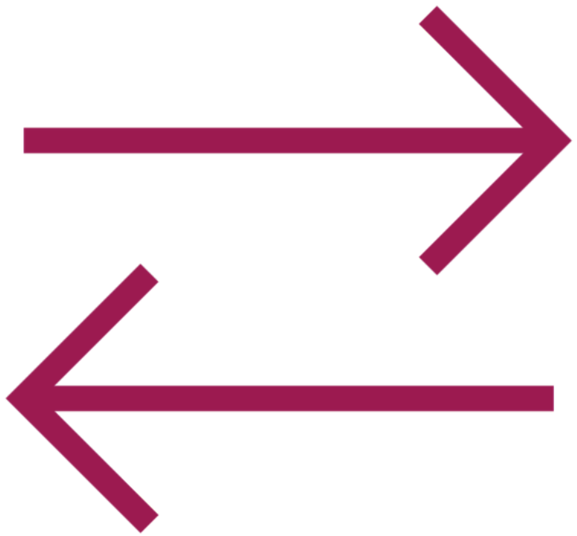
Transform Code Requirements

Function objects must be serializable

Function objects must be thread compatible

Beam SDKs are not thread-safe

Transform Code Requirements



Best practice is to make the function idempotent

Non-idempotent functions are supported but harder to ensure correctness

Summary

Core Beam transformations

ParDo, GroupByKey, CoGroupByKey, Combine, Flatten, Partition

Applying core transformation to input data

Requirements for writing user code for Beam transforms

Up Next:

Working with Windowing and Join Operations