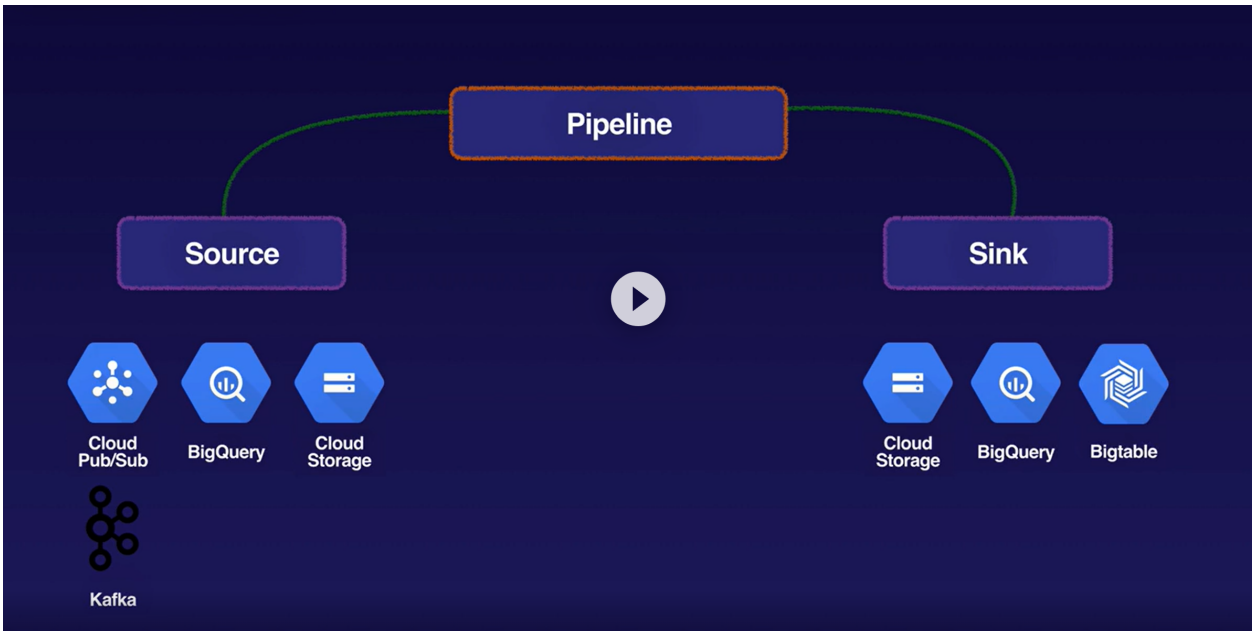
Pipelines with Cloud Dataflow

Target Complete Date	@December 9, 2022
Status	Completed
Reviewed	Reviewed
# Time to complete (Hours)	2
Type	Cloud Guru

Dataflow introduction

Big Data Ecosystem

Dataflow is a very powerful ETL tool for transforming data. It is a fully managed, serverless tool. Handles autoscaling of worker resources. Uses the open source Apache Beam SDK. Real-time and batch processing.



- Each pipeline has a source and a sink.

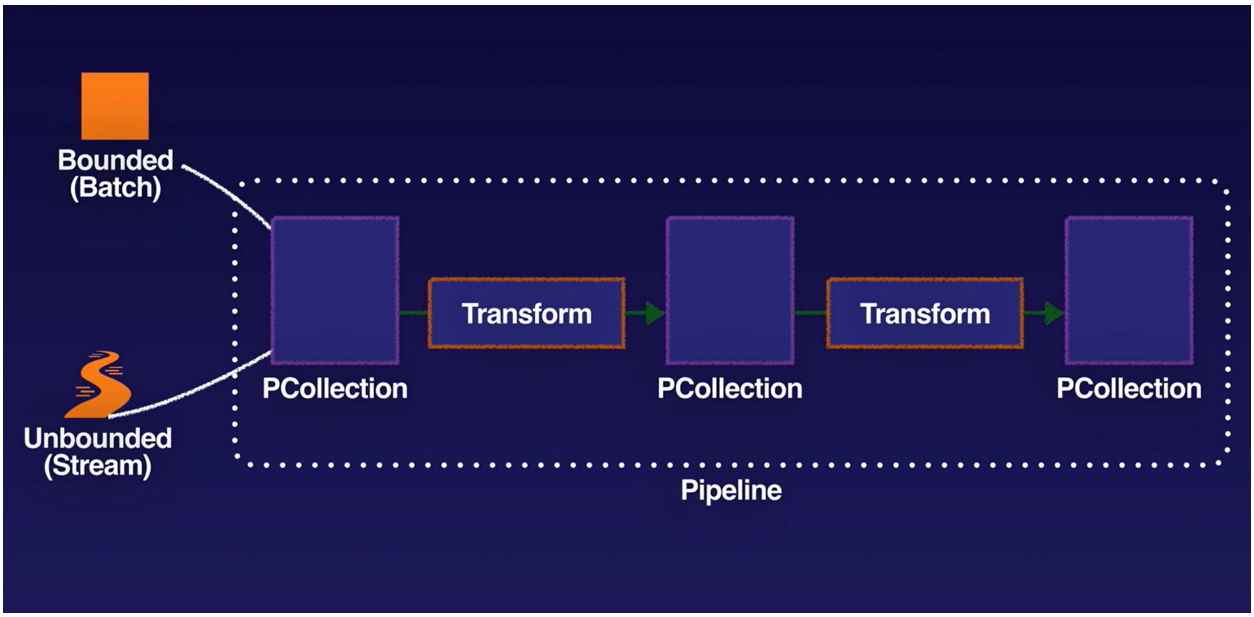
Driver program and runner

Driver program is written in Python or Java. It defines your pipeline. Full set of transformations your data undergoes. From ingestion to final output.

Driver program submitted to **runner** for processing. The runner is software which manages executing of your pipeline. It translates the pipeline for your back-end.

- Dataflow represents Driver program and runner.

PCollections and transforms



- PCollection represents data as it is transformed in the pipeline. It represents a potentially distributed multi-element dataset. It can represent both batch and streaming data. If it is batch data it said to be bounded. Else it is unbounded.
- PCollection are initially created from an external data source.
- Transforms use PCollection and outputs Pcollections.
- Multiple transforms define your pipeline.

Pipeline lifecycle

Design

- Understanding necessary transformations

Create

- Implement transformations

Testing

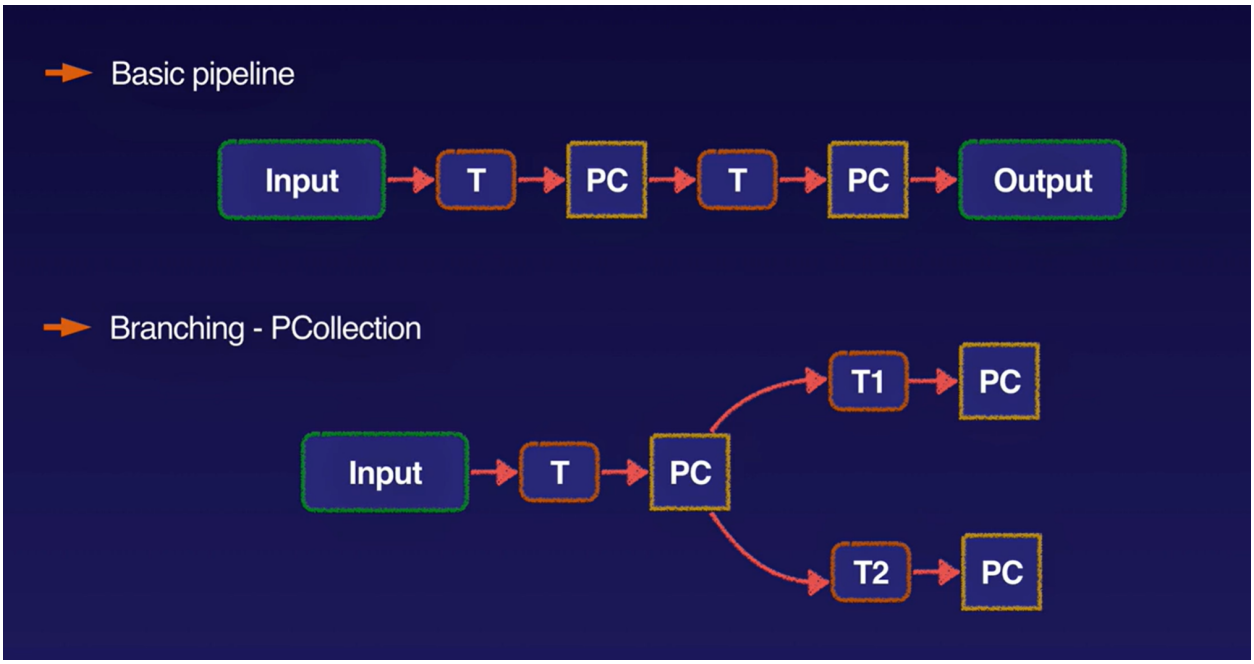
- Hard to debug failed pipeline on remote system
- Test on local machine to pick up complex errors.

Considerations

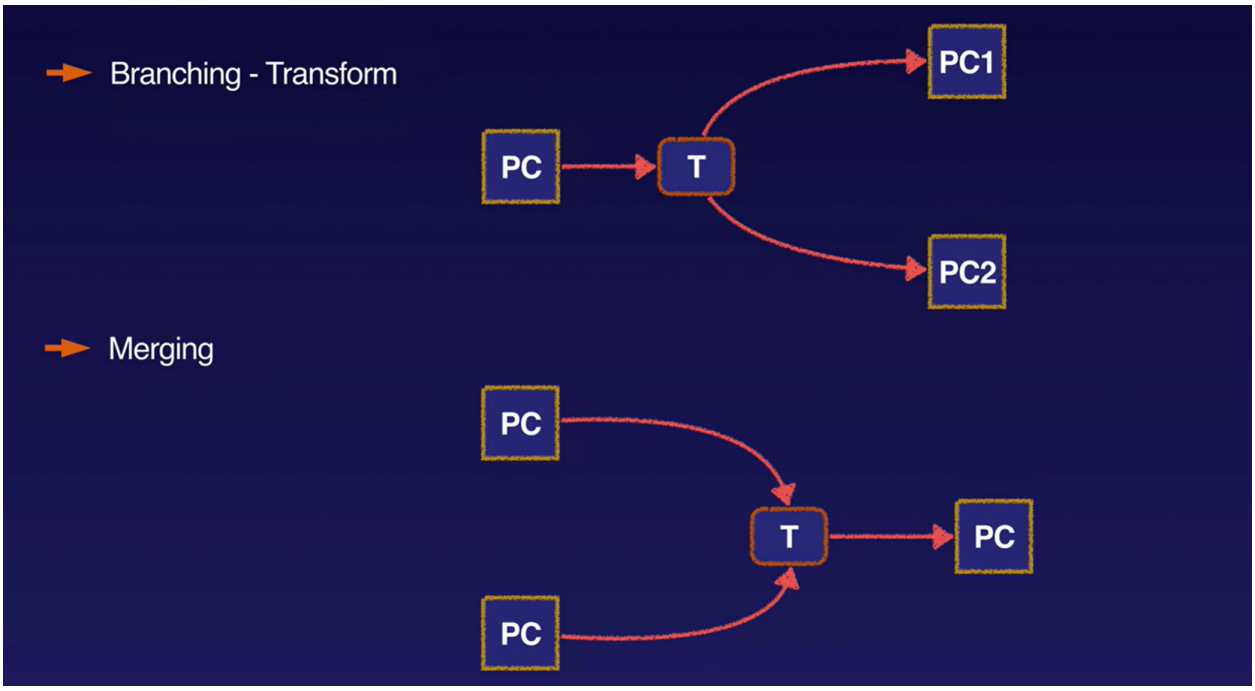
Location of data

- Where is the data sourced
- Input data structure and format
- Transformation objectives
- Output data structure and location

Pipeline structure



- You can have a basic pipeline which is linear
- You can also have a branching pipeline
- Branching can also occur on transforms



- Pipelines can also be merged, and should be merged.

- Pipelines can also have several sources

DAG

Each Dataflow pipeline represents a DAG. A DAG is a graph with a finite number of vertices and edges. There are no directed cycles. The output of a transform cannot be the input of the same transform.

Pipeline creation

1. Create pipeline object
2. Create a PCollection using read or create transform
3. Apply multiple transforms as required
4. Write out final PCollection (to pipeline sink)
5. Execute pipeline using pipeline runner

Dataflow pipeline concepts

Pardo

A transform for generic parallel processing. A `ParDo` transform considers each element in the input `PCollection`, performs some processing function (your user code) on that element, and emits zero or more elements to an output `PCollection`

User defined function

Specifies the operation to apply to each element within the PCollection. Can be written in a different language from your runner program.

Aggregation

User defined functions can be used in aggregation.

Characteristics of PCollection

PCollection elements needs to be all the same type. Pcollections do not support random access to individual elements within collection. They are immutable and unchanging. Beam transforms uses input PCollections to create new output PCollections. A timestamp is associated with every element of a PCollection.

Apache Beam

Core beam transforms

- ParDo
- GroupByKey
- CoGroupByKey
- Combine
- Flatten - Merges multiple input collections into single logical output Pcollection
- Partition

Advanced Dataflow concepts

Event time

> 💡 The time a data element occurs is represented by the event time timestamp. This is contrasted with the time that your data is processed within the pipeline. Processing times are greater than event time.

Window types

- Fixed time windows: Constant non overlapping window
  - Each element within stream is assigned in a single window
- Sliding window: Can overlap
  - Elements can belong to more than one window
  - Useful for moving averages
- Per session window
  - Apply on a per key basis
- Single global window

Watermarks

- There is an event lag between event time and processed time.
- Watermark is the system notion for all the data in a certain window can be expected to arrive.
  - Late data is classified as data that did not make it in time for the expected window.

Triggers

- Trigger types
  - Event time trigger

Security and Access Dataflow

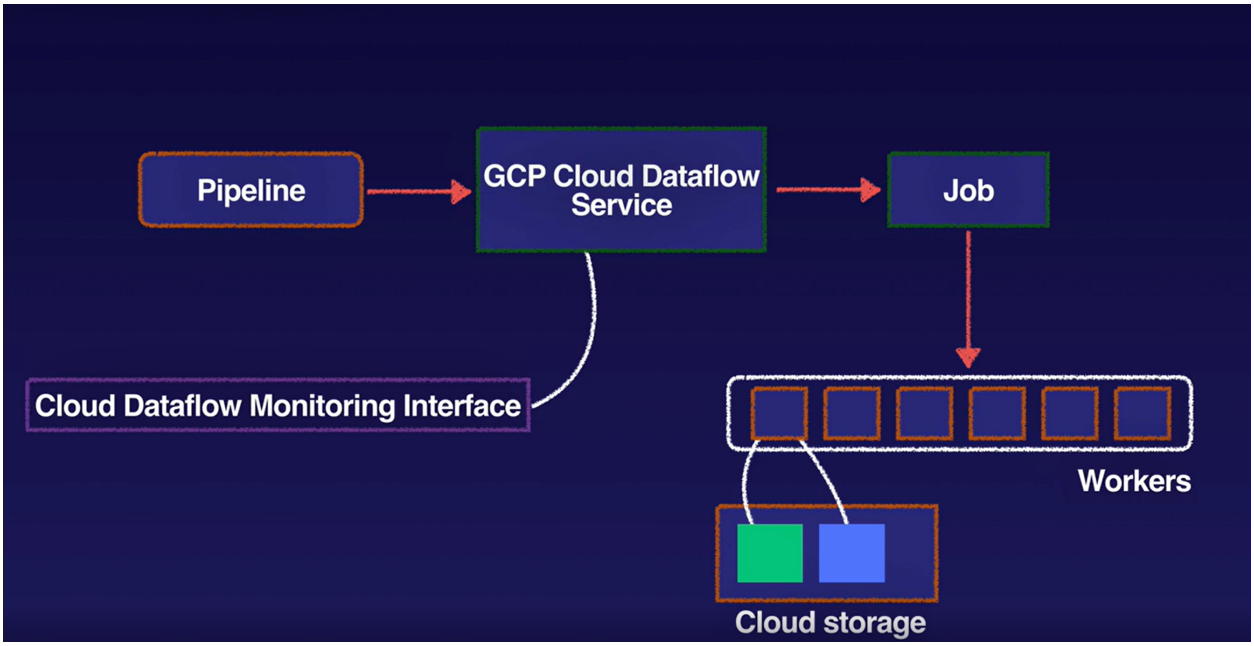
Run Cloud dataflow pipelines

- Locally - Testing
  - Requires Apache Beam SDK
- Cloud dataflow managed service - Testing

GCP service accounts

- Cloud dataflow service → dataflow service account
- Worker instances → Controller service account

Cloud Dataflow managed service



- Cloud dataflow service accounts
  - Automatically created
  - Manipulates job resources
  - Cloud dataflow service agent role
  - Read/Write access to project resources
- Controller service account - Used by workers
  - Uses project computer engine service account as controller service account

    - Compute engine instances execute pipeline operations
    - Metadata operations
    - User-managed controller service account

Access and security

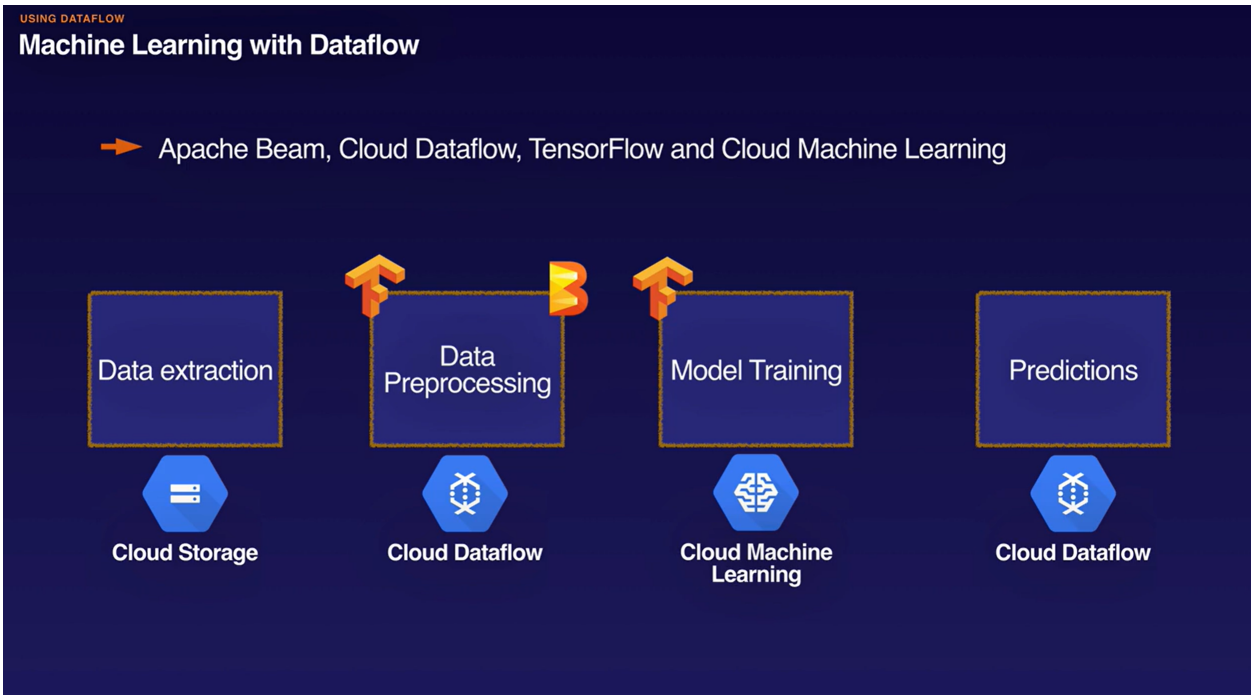
- Security mechanisms
    - Submission of the pipeline
    - Evaluation of the pipeline
    - Accessing telemetry or metrics
- Cloud Dataflow IAM roles

Using Dataflow

Regional endpoints

- Manages metadata about Cloud Dataflow jobs
- Controls cloud dataflow workers
- Automatically selects best zone
    - There could be company restrictions that force dataflow workers to be executed in a given region

Machine learning



- This should give you an idea as to how cloud Dataflow will be used as part of a Machine Learning solution

Extra details

- Customer managed encryption keys
- Flexible resource Scheduling (FlexRS)
  - Advanced scheduling
  - Cloud dataflow shuffle service
  - Preemptible VMs
- Migrating MapReduce jobs to Cloud Dataflow
  - Any app engine map reduce workflows should be pushed to cloud dataflow
- Cloud dataflow with Pub/Sub seek

Cloud Dataflow SQL

- Develop and run cloud dataflow jobs from the BigQuery web UI
- Cloud Dataflow integrates with Apache Beam SQL

- Cloud Dataflow SQL is the GCP version of Apache Beam SQL

Exam tips

> 💡 Dataflow is an ideal managed solution for customers using Apache Beam. Dataproc and spark are best for batch, beam and dataflow for streaming.

- You need to understand some beam SDK functions
  - ParDo
  - DoFn is a template you to create user-defined functions that are referenced by pardo
- Sources & Sink
  - Sources are where data get written from
  - Sinks are where data are written to

  ☑️ ~~What is an I/O transform~~
- Windowing
  - Allows streaming data to be grouped into finite collections according to time or session-based windows.

☐ Look into windowing within beam

- dataflow vs Cloud composer
  - Dataflow is normally preferred for data ingestion
  - Cloud composer may sometimes be used for ad-hoc orchestration

DataFlow demo

> 💡 This demo includes creating a streaming pipeline that transforms data coming from a fake social media website.

⚙️ The first thing we do is create a bucket, and a pub/sub topic. Create a BigQuery table where all the data will end up. A bucket is needed because Dataflow needs a temporary and staging bucket to process data.

- We have to use environment variables within functions as Dataflow is distributed. We cannot always assume that the environment variables will be globally accessible.

Quiz

☑️ ~~Look into ParDo and DoFn~~

💡 A PCollection represents a potentially distributed, multi-element dataset that acts as the pipeline's data.

💡 ParDO is a beam transform for generic parallel processing. The ParDo processing paradigm is similar to the Map phase of a Map/Suffle reduce style algorithm.

☑️ ~~Look into map reduce programming paradigm and how beam approaches it~~

💡 A sliding window represents time intervals in the data stream; however; sliding time windows can overlap. This kind of windowing is useful when taking running averages of data.

💡 Trigger determine when to emit aggregated results as data arrives.

☑️ ~~Look into trigger~~

☑️ ~~How to apply local execution~~

Review

I/O transform

💡 In beam an I/O transform is a type of transform that provides I/O functionality for reading from and writing to external data sets.

ParDo and DoFN

💡 ParDo is a fundamental data transform that applies a transformation to each element of a data collection producing zero or more outputs for each element. The user defined function is known as DoFn. A DoFn can take one or more input elements, perform a transformation and output zero or more elements.

Beam vs MapReduce

💡 The Beam programming model allows for more complex data transformations and functions. Such as filter, join, Combine and Pardo.

Trigger

💡 A trigger is a mechanism that defines when a window of data is emitted as an output in a data processing pipeline. A window is a logical segment that is defined based on time or size criteria.

Local execution

> 💡 Local execution in beam allows for a development environment which runs on a developer's local machine.

IAM

- `roles/dataflow.admin`