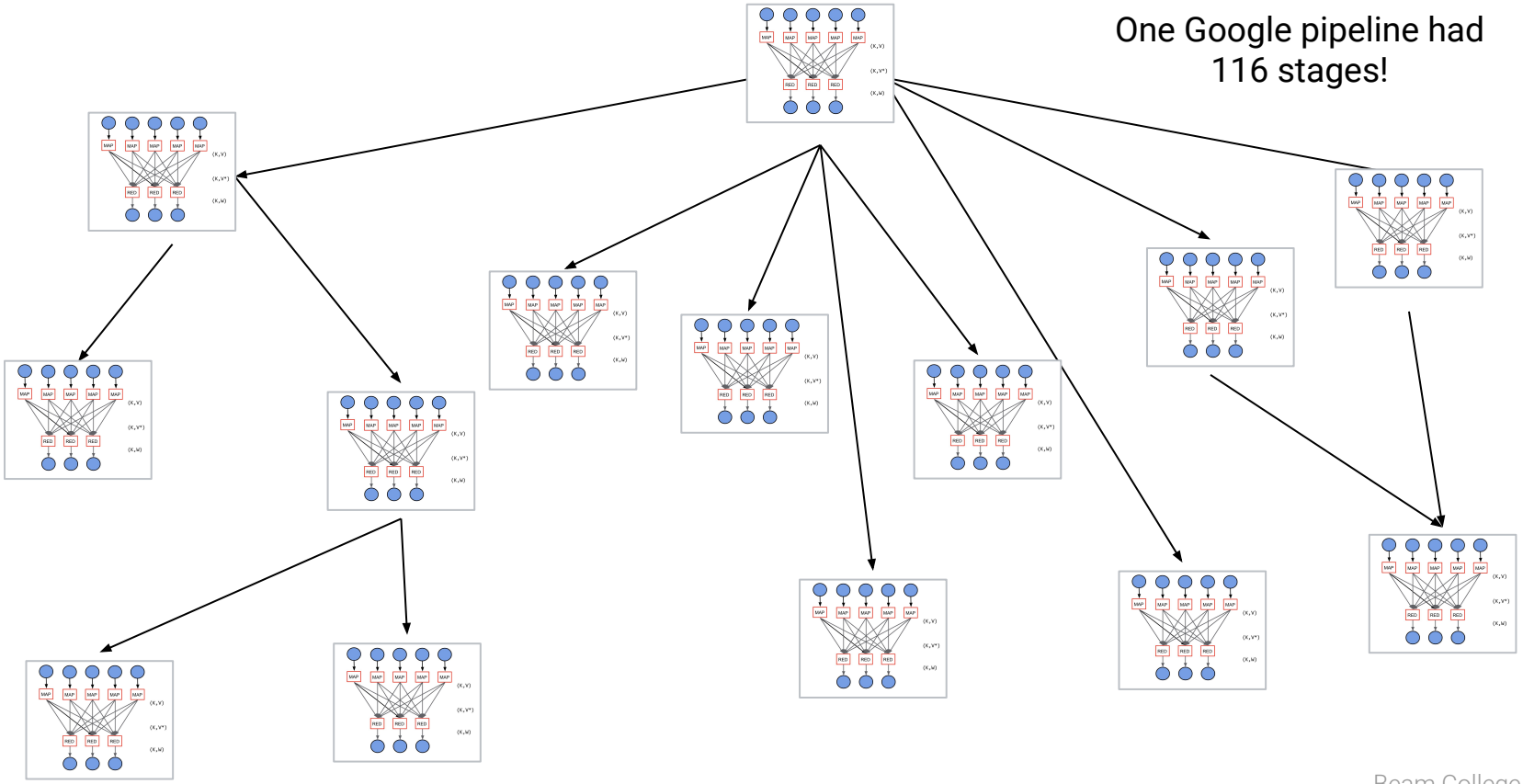
Describing a pipeline declaratively - DAG

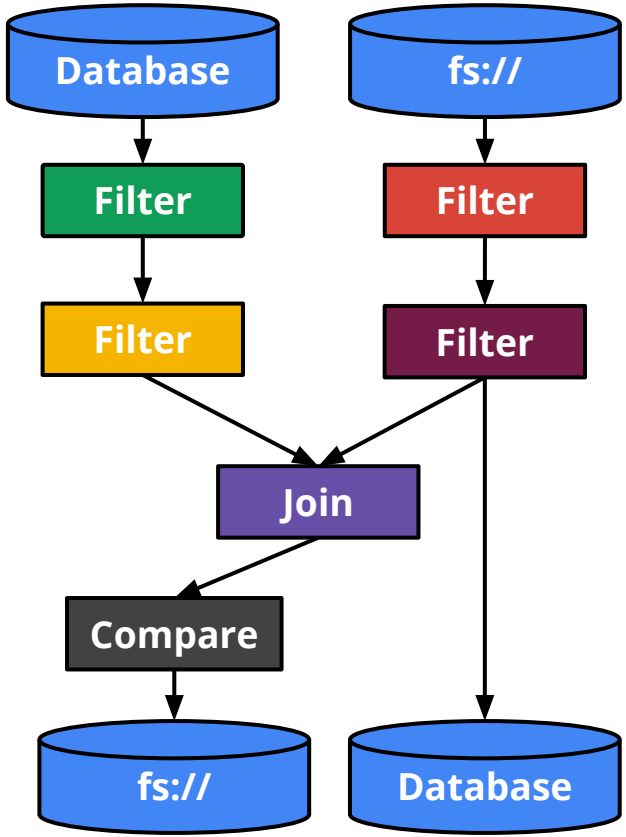
Tyson Hamilton
@tysonjh

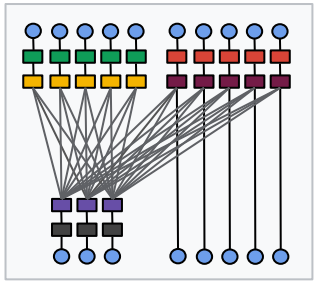
How do we fix this?



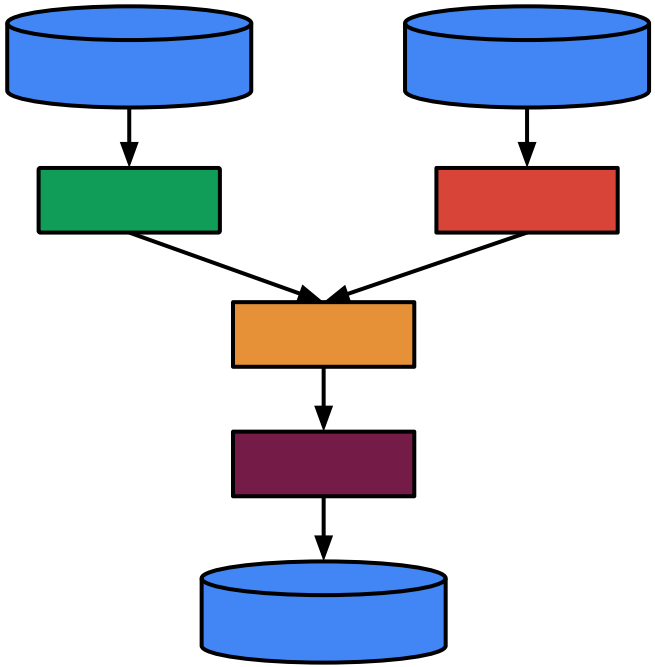
One Google pipeline had 116 stages!

Abstracting from the execution...



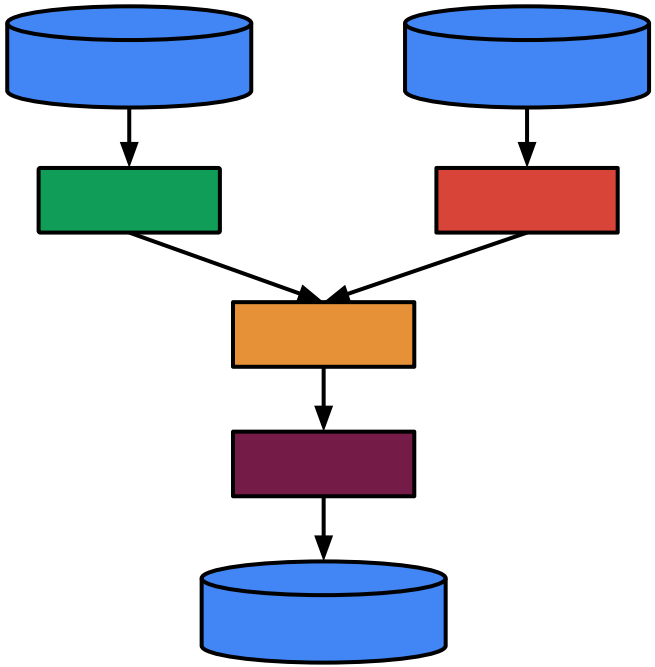


What is a pipeline?



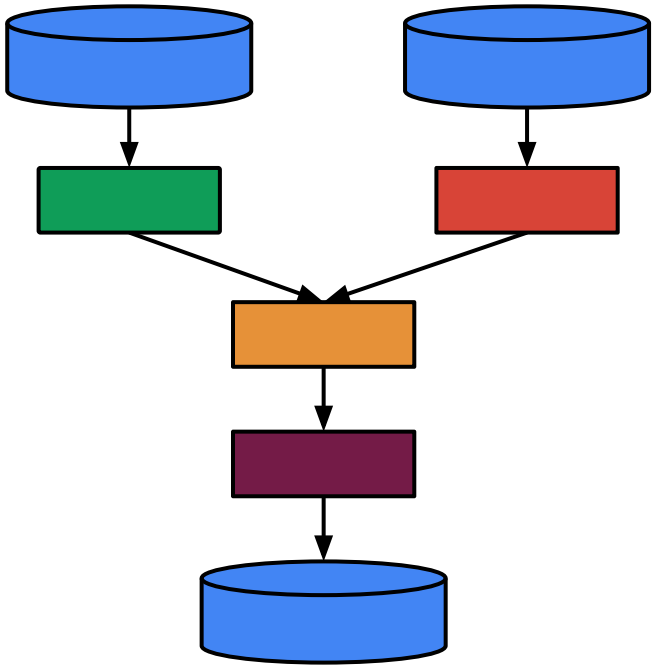
- A Directed Acyclic Graph of data transformations applied to one or more collections of data
- Possibly unbounded collections of data flow on the edges
- May include multiple sources and multiple *sinks*
- Optimized and executed as a unit

What is a pipeline?



Beam represents datasets using an abstraction called PCollection

What is a pipeline?



Data transformations are represented by an abstraction called PTransform

Retail Company: Popular Products

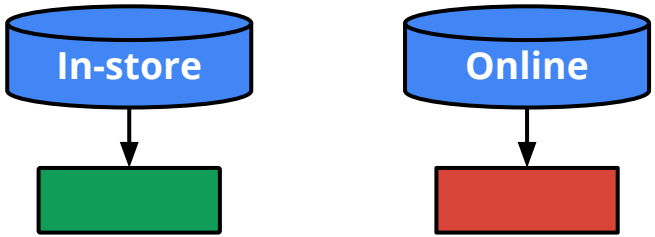
Question: What are Acme's most popular products?

- There's a lot of data to analyze!
- And it's split between two different input streams!

How do we even structure this?

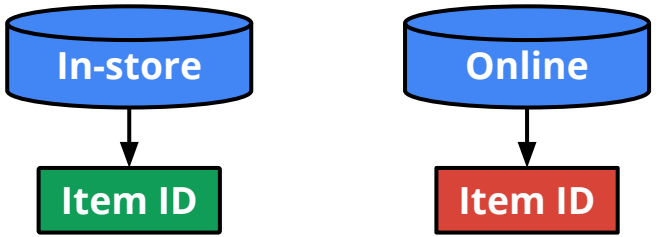
- The answer is a pipeline.

Acme Beaver: Popular Products



Read in-store & online sales

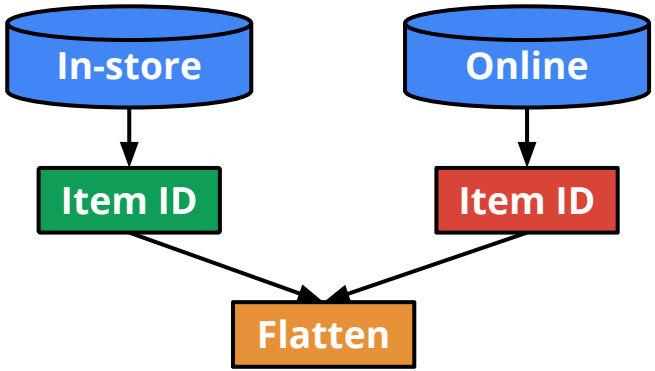
Acme Beaver: Popular Products



Read in-store & online sales

Extract item ids

Acme Beaver: Popular Products

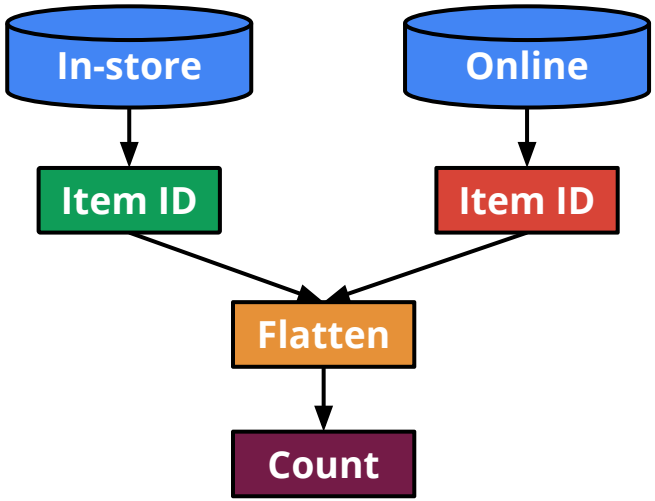


Read in-store & online sales

Extract item ids

Flatten to create a unified view

Acme Beaver: Popular Products



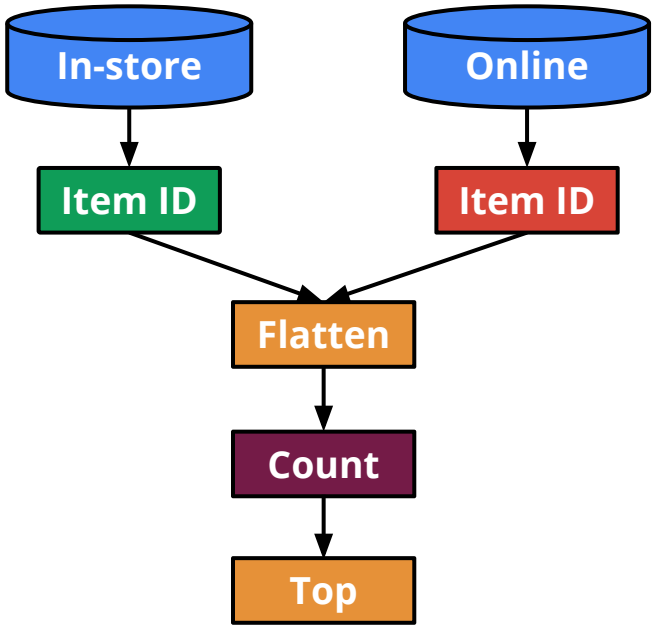
Read in-store & online sales

Extract item ids

Flatten to create a unified view

For each item id, count the # of sales

Acme Beaver: Popular Products



Read in-store & online sales

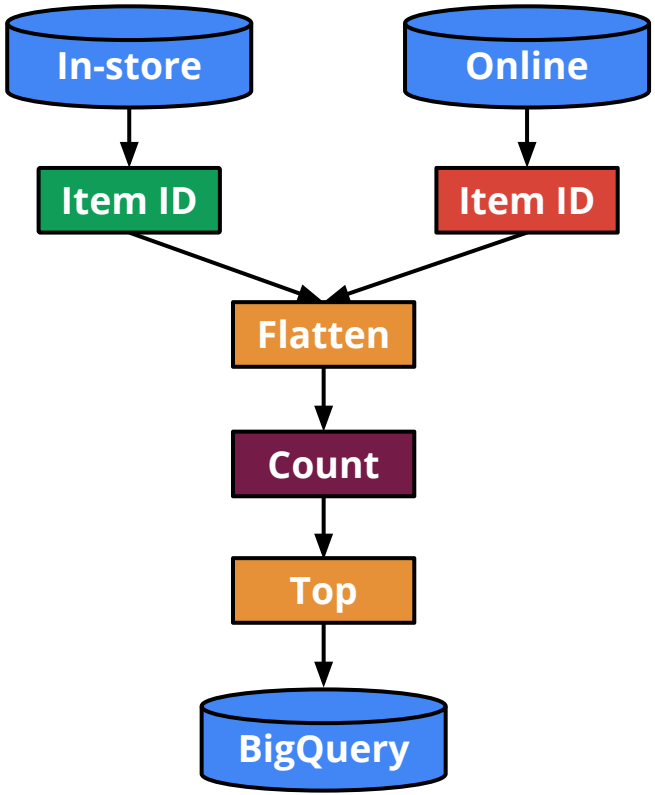
Extract item ids

Flatten to create a unified view

For each item id, count the # of sales

Pick items with the highest counts.

Acme Beaver: Popular Products



Read in-store & online sales

Extract item ids

Flatten to create a unified view

For each item id, count the # of sales

Pick items with the highest counts.

Write the result to BigQuery

DAG - Advantages

- Concentrate on the business logic
- Less code! Declarative vs imperative
- Deferred Execution
  - Performance optimizations (Fusion)
  - Contextual monitoring, troubleshooting, logging

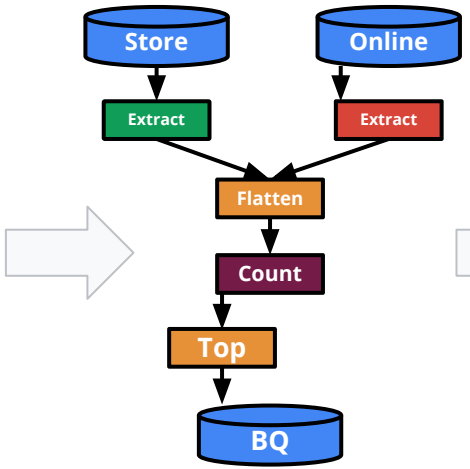
Deferred Execution

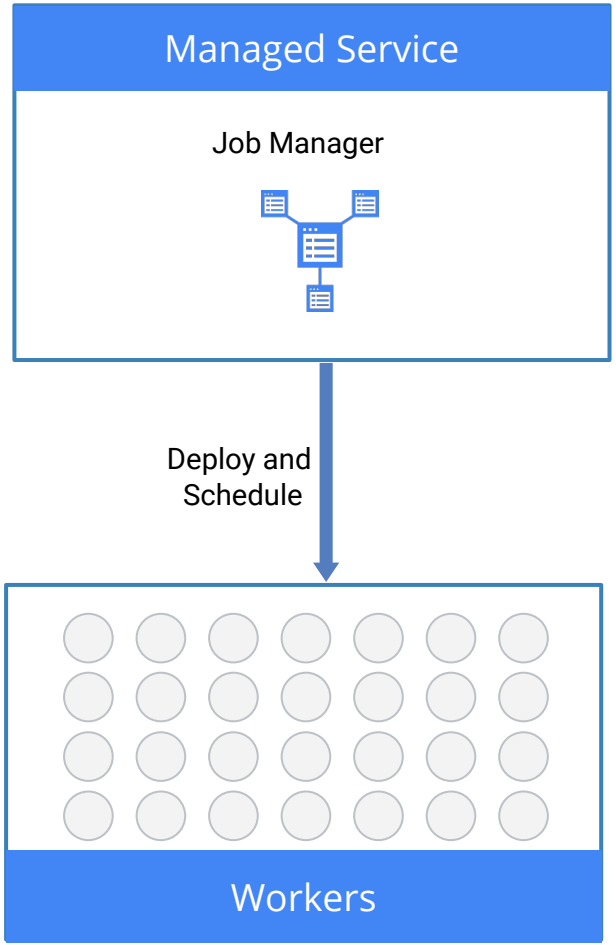
```
PCollection<Purchase> storePurchases =
    p.apply(TextIO.read(“purchases-store”)
        .apply(new ExtractId());

PCollection<Purchase> onlinePurchases =
    p.apply(TextIO.read(“purchases-online”)
        .apply(new ExtractId());

PCollectionList
.of(storePurchase).and(onlinePurchases)
    .apply(Flatten.pCollections())
    .apply(Count.perKey())
    .apply(Top.of(10, compareValues()))
    .apply(BigQueryIO.write(“topSales”));

p.run();
```





Summary

1. Demonstrated how to abstract the user from the control logic of a pipeline.
2. Core pipeline primitives.
3. Simple example.
4. Deferred execution.

Thank you!

Questions?