ntic master School

UNIVERSIDAD COMPLUTENSE DE MADRID

4

Streaming

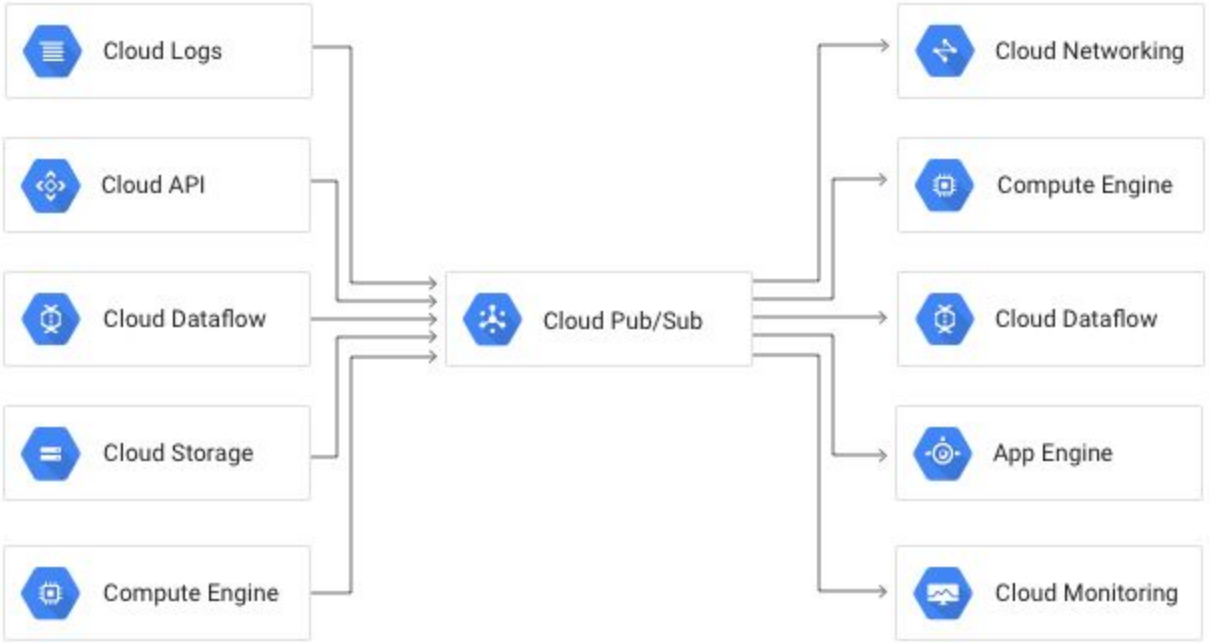
Sistemas que capturan datos en tiempo real, y permiten responder a preguntas como cuántos clientes han comprado en mi web en los últimos 10 minutos.

Pub/Sub: 100% serverless event delivery

Google Cloud Pub/Sub

- Reliable and real-time messaging
- Global by design and highly available
- Uses Google's private fiber network and worldwide points of interconnect
- Only pay for what you use

Pub/Sub: Google Cloud Pub/Sub passes messages

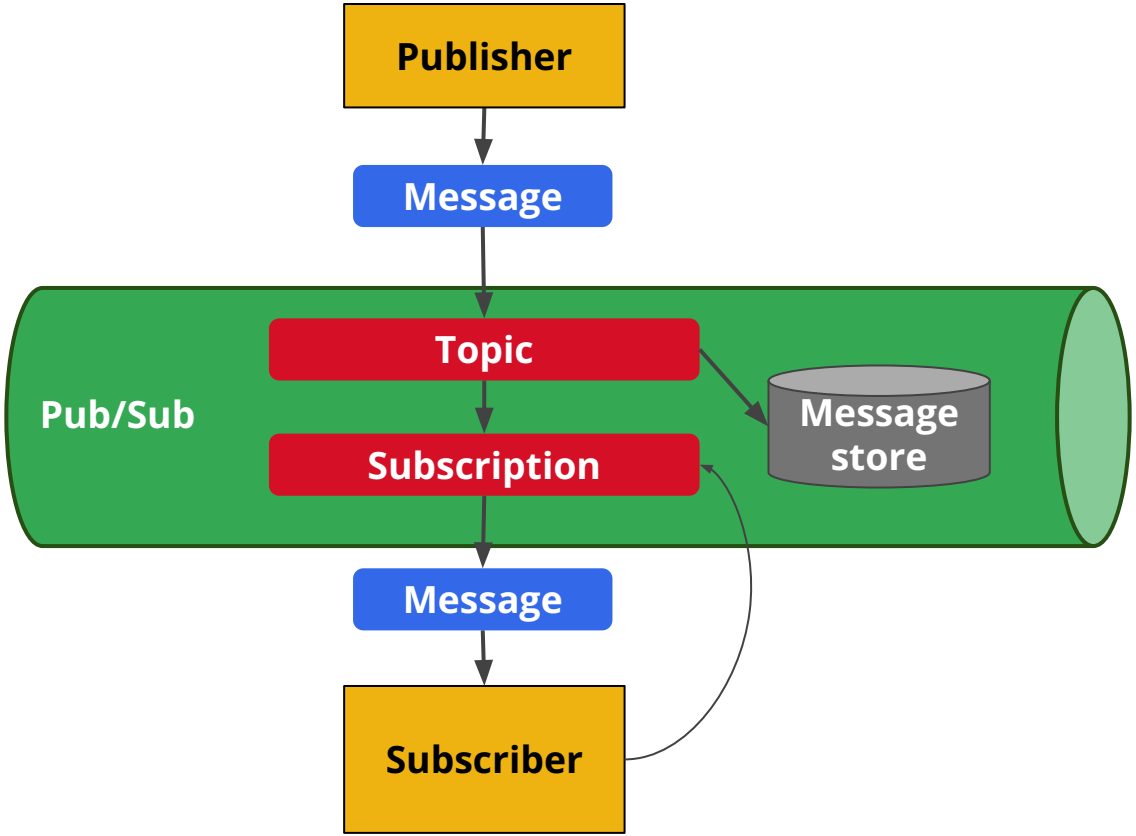


Pub/Sub: What is Google Cloud Pub/Sub

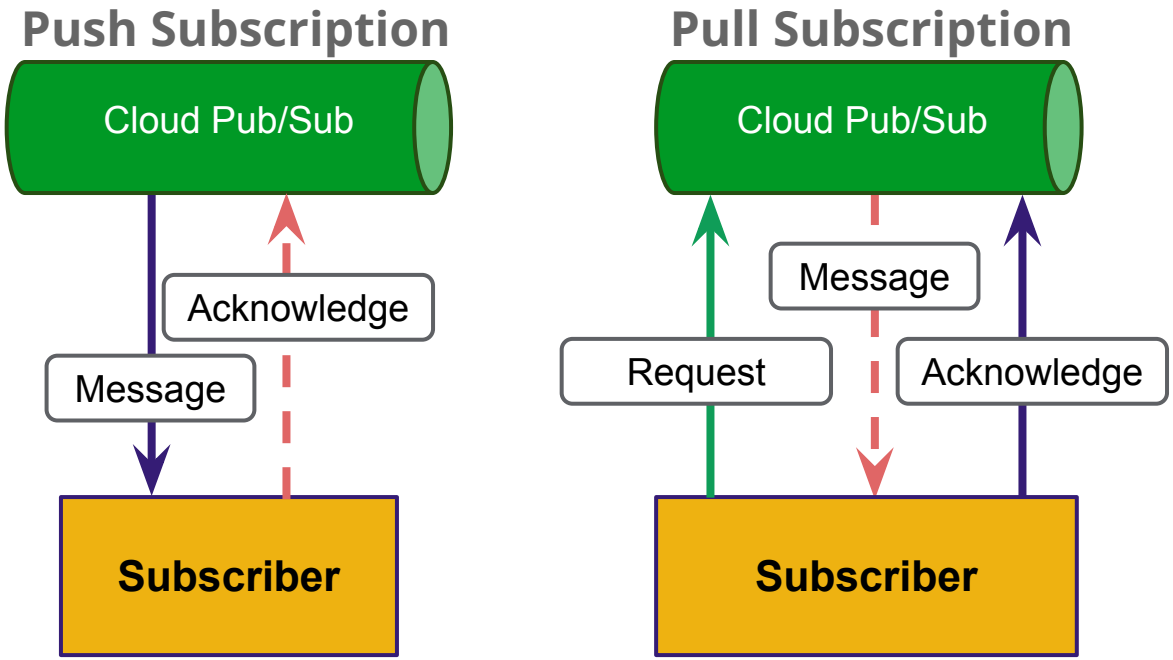
- Cloud Pub/Sub is a service to capture and rapidly pass massive amounts of data (or messages) between software applications with world-class security.
- It uses the **Publish - Subscribe** pattern:
  - **Publisher** applications can send **messages** to a **topic**.
  - **Subscriber** applications can subscribe to that topic to receive the message when the subscriber is ready (asynchronously).
- Pub/Sub acts as a buffer between sending and receiving software applications, making it easier for developers to connect applications.

- Cloud Pub/Sub provides:
  - **Scale**—It supports Google's services, such as Gmail and ads.
  - **Reliability**—Dedicated resources in every Google Cloud Platform region enhance availability without increasing latency.
  - **Performance**—Sub-second notifications even when tested at over 1 million messages per second.
  - **Cost-efficiency**—A "pay for what you use" service.
  - **Ease of use and implementation**—Because it's a fully managed service, there's no need to manage your own open source software implementation. Get started in minutes, not days.

Pub/Sub: Architecture



Pub/Sub: Architecture



Pub/Sub: Messaging is a shock-absorber

Availability

- Buffer messages and requests during outages
- Prevent message overloads that cause outages
- Redirect requests to recover from outages

Throughput

- Smooth out spikes in new request rate
- Balance load across multiple servers
- Balance arrival rate with service rate
- “Fan-in” from many devices

Latency

- Accept requests closer to the network edge
- Optimize message flow across regions

Pub/Sub: Messaging is a shock-absorber

Sources

- New data sources can plug into old data flows
- New data sources can use new schemas
- Common security policies for all sources

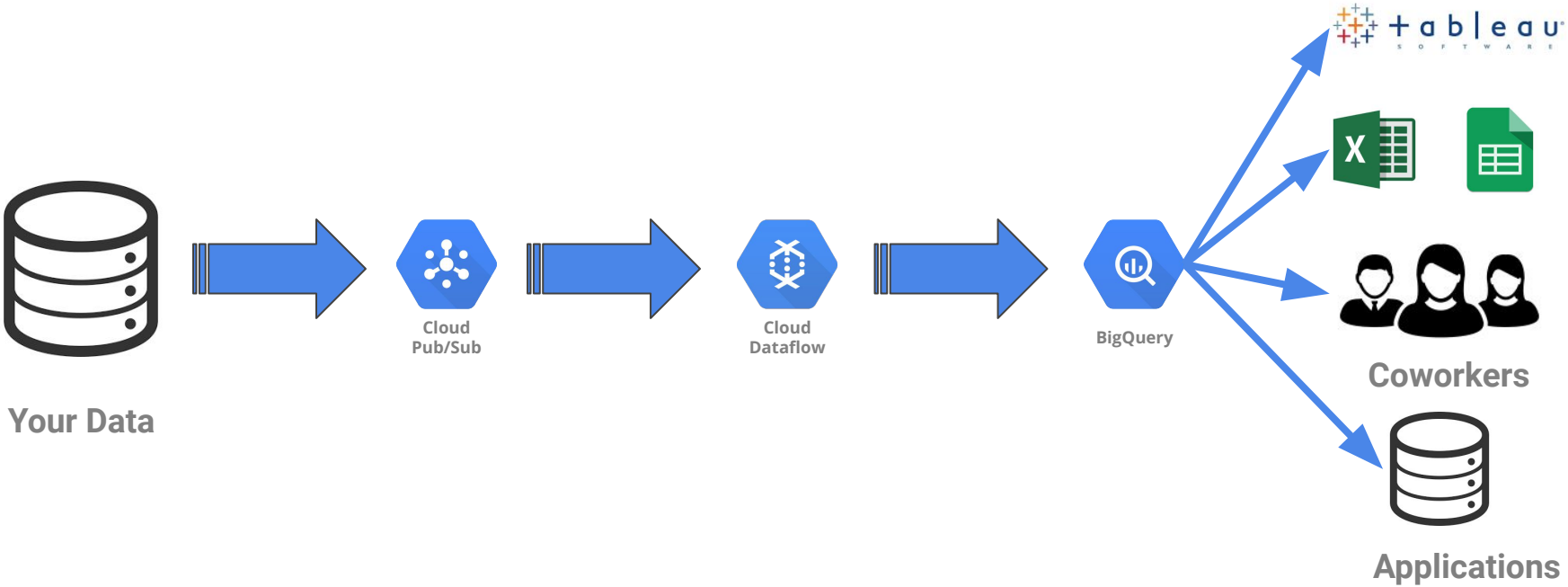
Sinks

- Data can be sent to new destinations
- “Push” and “pull” delivery are both available
- Spans organizational boundaries

Transforms

- Can merge streams into new topics
- Sends messages to Dataproc and Dataflow for transformation

Pub/Sub: Use case: big data stream



Pub/Sub: examples in use at Google

Chat & Mobile

Every time your Gmail displays a new message, it's because of a push notification to your browser or mobile device.

Ads and budgets

One of the most important real-time information streams in the company is advertising revenue—we use Pub/Sub to broadcast budgets to our entire fleet of search engines.

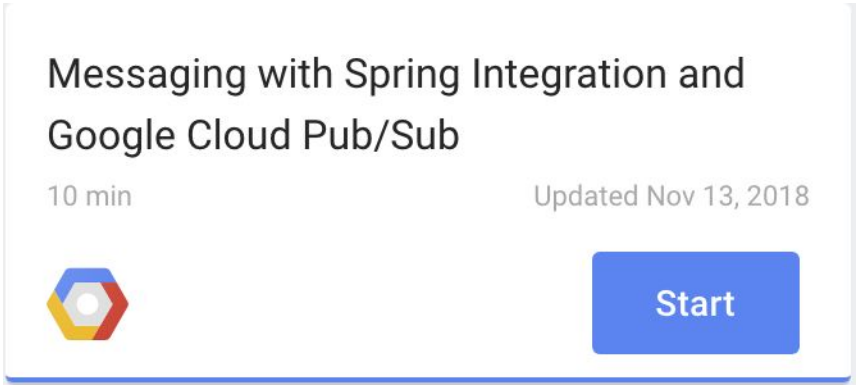
Push notifications

Google Cloud Messaging for Android delivers billions of messages a day, reliably and securely, for Google's own mobile apps and the entire developer community.

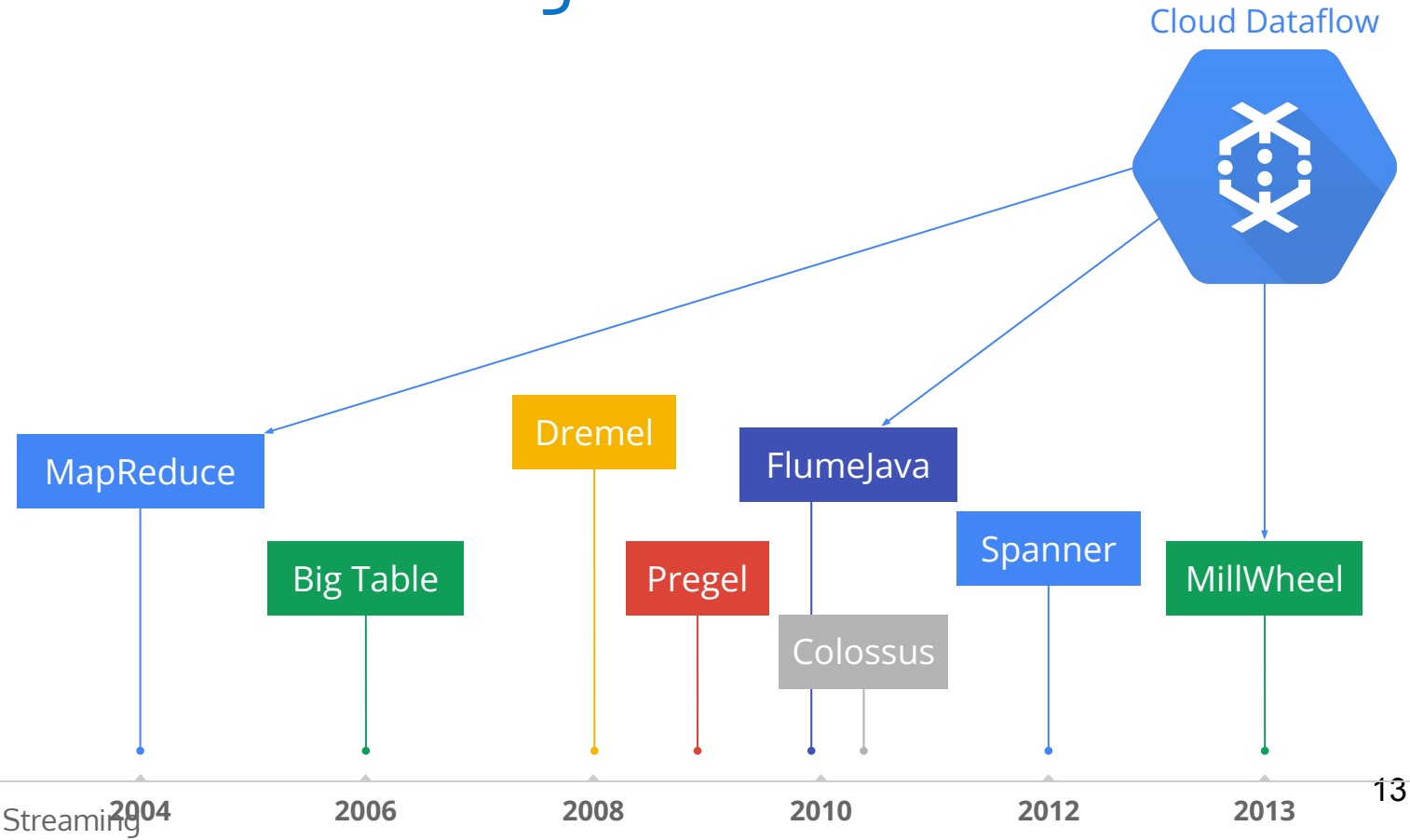
Instant search

Updating search results as you type is a feat of real-time indexing that depends on Pub/Sub to update caches with breaking news.

PubSub - Codelab



Dataflow - Processing as a Service



Dataflow

Google Cloud Dataflow

- Dataflow is a unified programming model and a managed service for developing and executing a wide range of data processing patterns including ETL, batch computation, and continuous computation.
- Cloud Dataflow frees you from operational tasks like resource management and performance optimization.

Dataflow

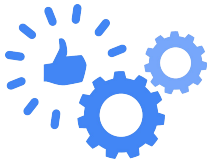
Servicio completamente gestionado y modelo de programación para el proceso de Big Data

- Gestión de Recursos integrado.
- A demanda.
- Ejecución de los trabajos inteligente.
- Auto escalado.
- Modelo de programación unificado.
- Open Source.
- Monitorage.
- Integración.
- Procesado confiable y consistente.

Dataflow: Managing Massive Data In/Out

ETL

- Movement
- Filtering
- Enrichment
- Shaping



Analysis

- Reduction
- Batch computation
- Continuous computation

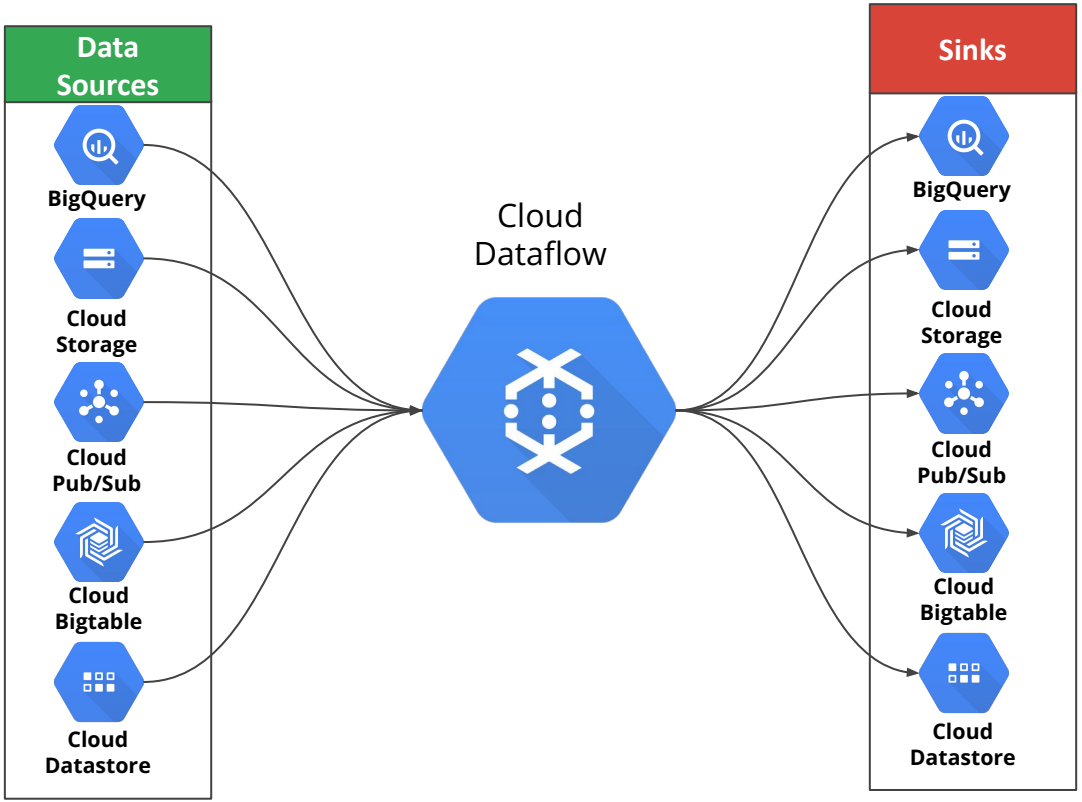
Orchestration

- Composition
- External orchestration
- Simulation

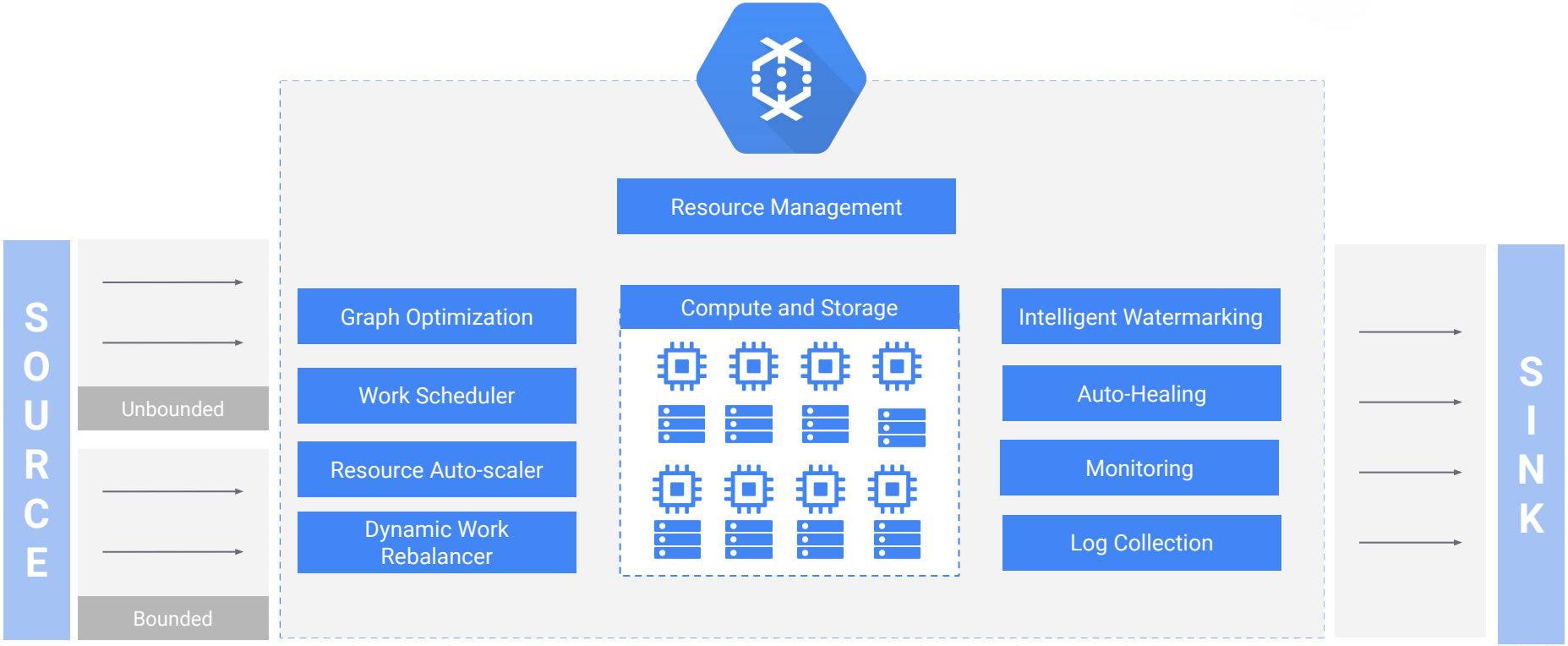
What is Cloud Dataflow?

- Google Cloud Dataflow is a is a tool for developing and executing a wide range of data processing patterns—for example extract, transform, and load (ETL)—on very large data sets.
- Use Cloud Dataflow for nearly any kind of data processing task, encompassing both batch and streaming data processing.
- Dataflow can handle an unbounded or “infinite” data set from a continuously updating source such as Google Cloud Pub/Sub. That is, Dataflow can process practically any amount of data arriving at any time.
- Dataflow is particularly useful for embarrassingly parallel data processing tasks, in which the problem can be decomposed into many smaller bundles of data that can be processed independently, making it very fast (in the same way MapReduce works).

Data Sources and Sinks for Dataflow

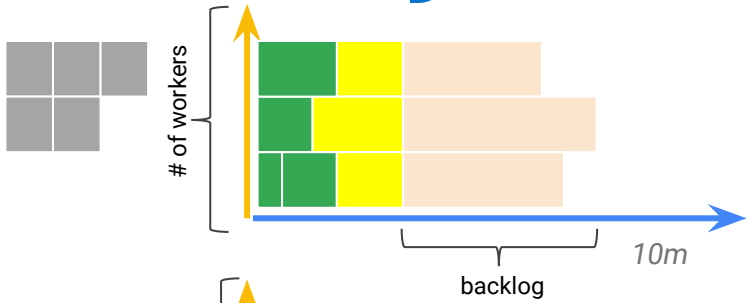


Cloud Dataflow: Under the Hood

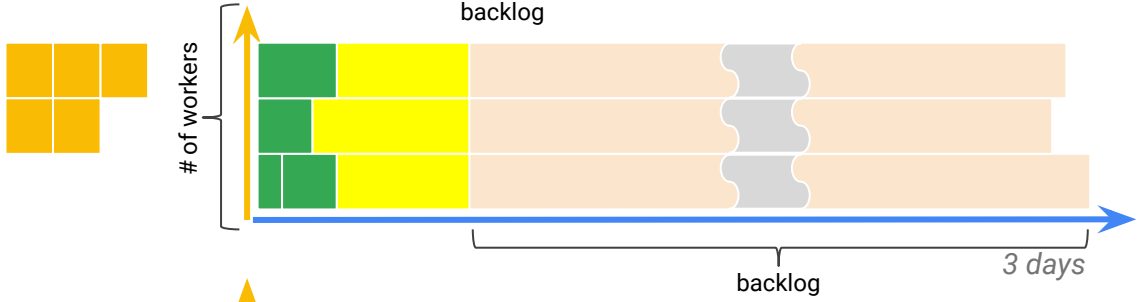


Cloud Dataflow: Autoscaling

Start off with **3** workers,
things are looking okay

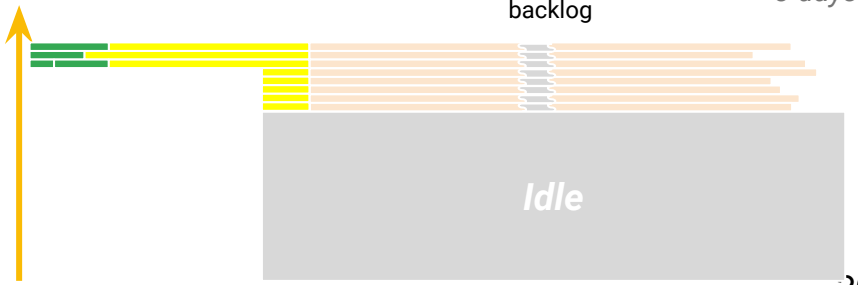


Re-estimation shows there's
orders of magnitude more work:
need **100** workers!



You have 100 workers
but you don't have 100 pieces of work!

...and that's really the most important part



Cloud Dataflow: why do people use it?

Usually **the goal** of big data efforts is to reduce the time required to answer questions for making faster, better decisions.

Examples of questions prospects want answered:

- “How many online sales did I make in the last hour due to advertising conversion?”
- “What was the average viewing time over the past seven days compared to last year?”
- “Which version of my web page do people like better?”
- “Which transactions look fraudulent?”

Cloud Dataflow **speeds up the rate at which questions can be answered** by:

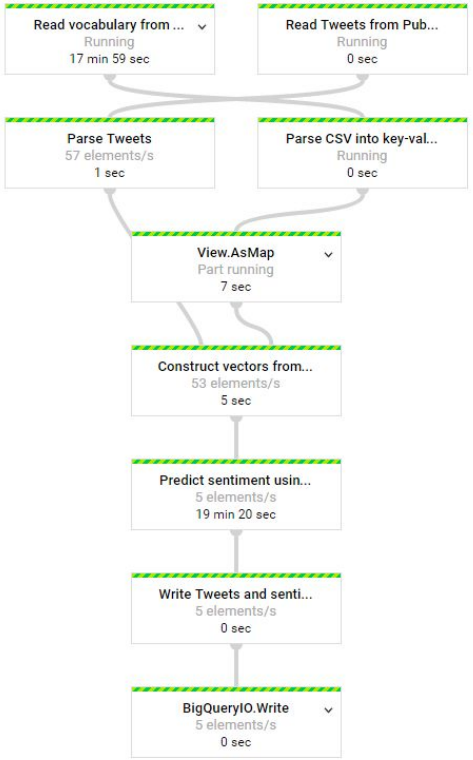
- Integrating data from multiple sources and preparing it for analysis.
- Analyzing event data streams using the Dataflow service.

How Cloud Dataflow works ?

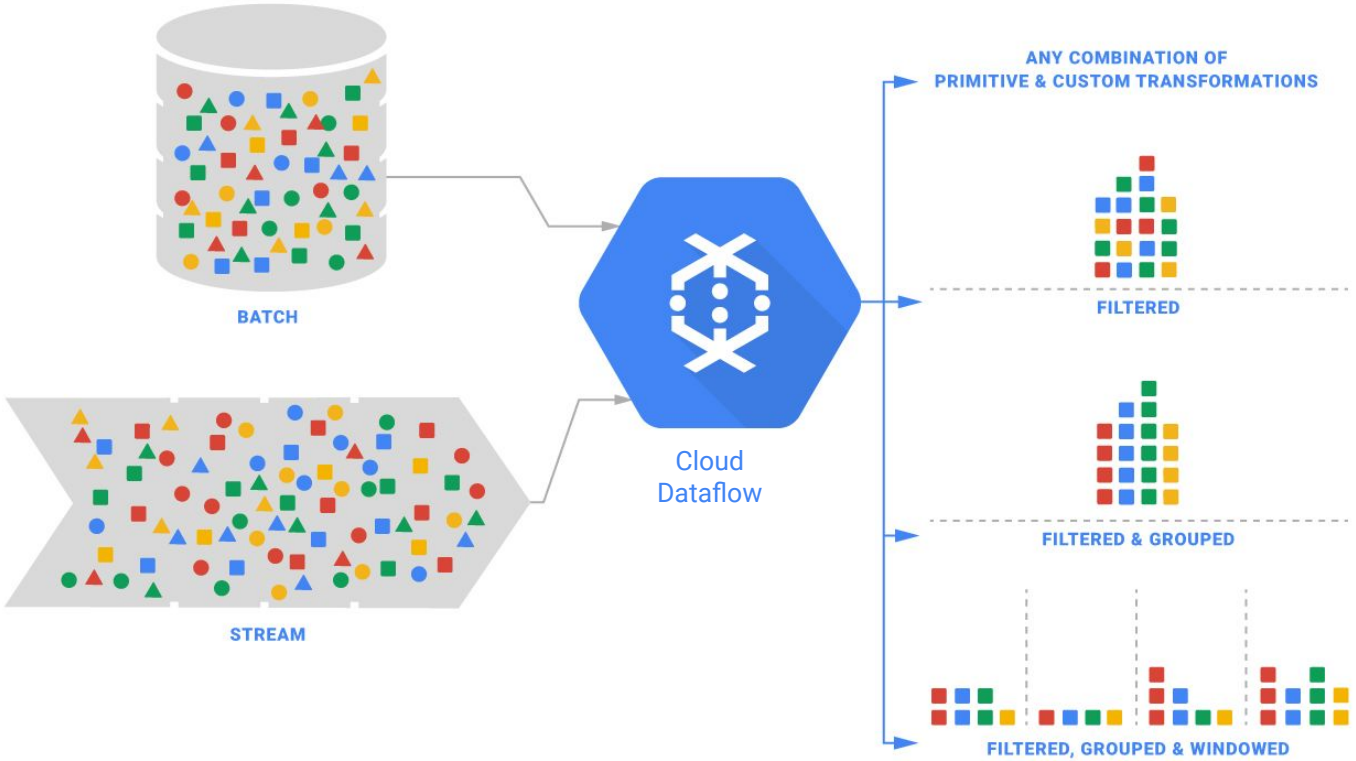
- Dataflow provides an easy way to create your data processing jobs. Each job is represented by a data processing **pipeline** that you create by writing a program.

- Each pipeline reads some input data, performs some transforms on that data to gain useful or actionable intelligence about it, and produces some resulting output data.

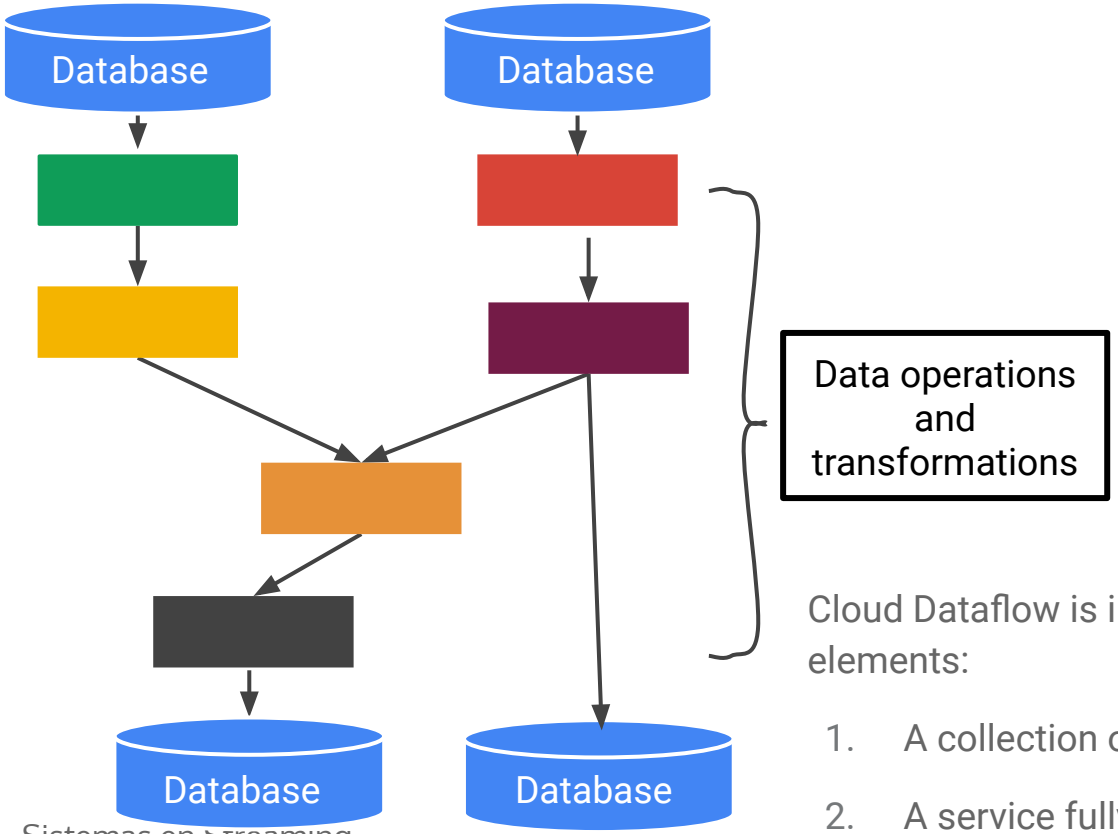
- A pipeline's transforms might include filtering, grouping, comparing, or joining data.



How Cloud Dataflow works ?



Dataflow: What are data pipelines?



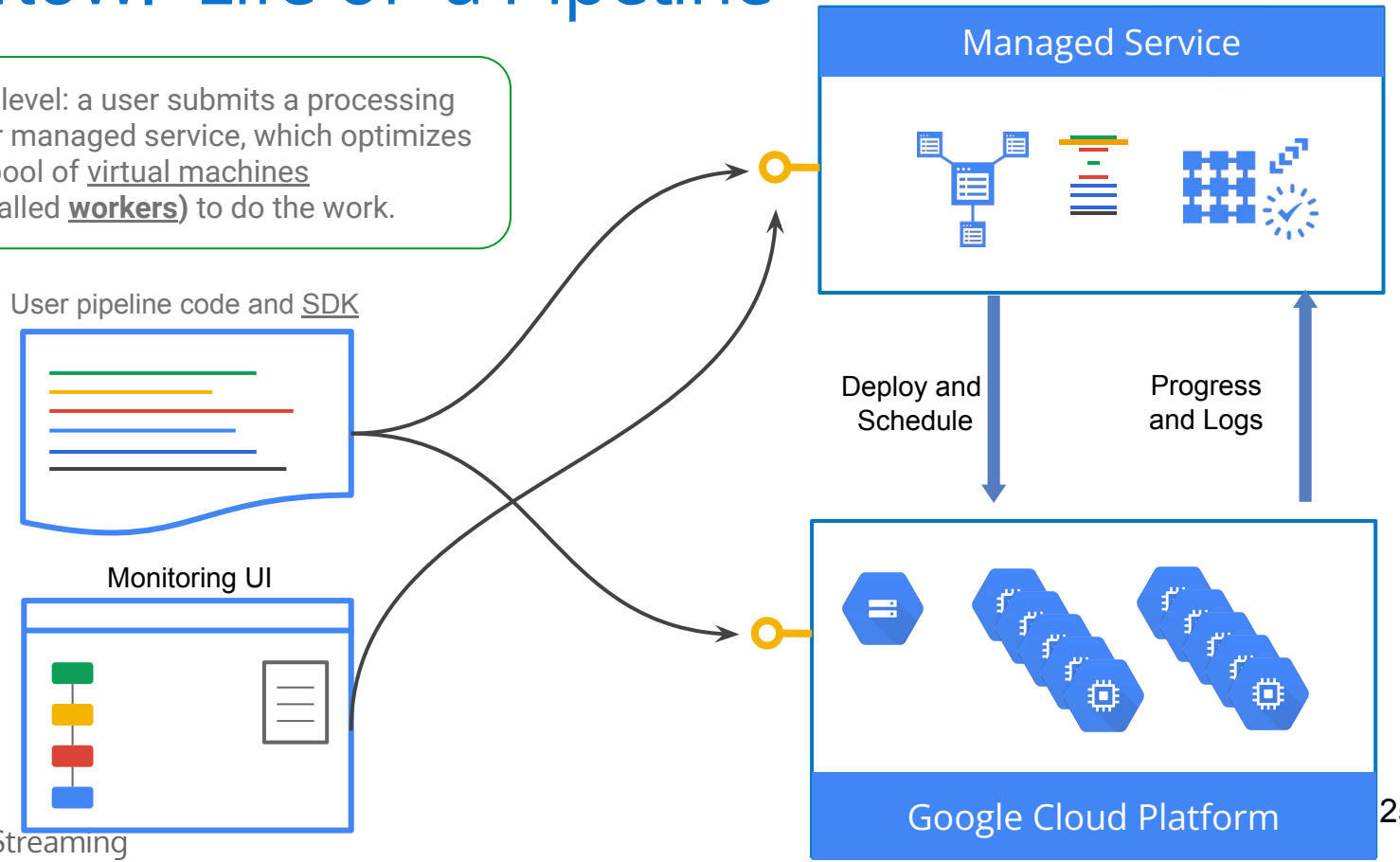
- A pipeline is a defined set of data processing transformations
- Optimized and executed as a unit
- Can include multiple inputs and multiple outputs
- Can perform many mathematical, logical, or transformation operations
- PCollections conceptually flows through the pipeline

Cloud Dataflow is implemented as two distinct elements:

1. A collection of software development kits.
2. A service fully managed by Google for running data pipelines.

Dataflow: Life of a Pipeline

At a very high level: a user submits a processing pipeline to our managed service, which optimizes it and runs a pool of virtual machines (sometimes called **workers**) to do the work.



Dataflow Python code

```python
def run(argv=None):
    """Main entry point; defines and runs the pipeline."""

    parser = argparse.ArgumentParser()
    parser.add_argument('--input',
                        dest='input',
                        default='gs://table/table.csv',
                        help='Input file to process.')

    parser.add_argument('--output',
                        dest='output',
                        default='project:dataset.table01',
                        help='Output BigQuery table for results specified as: PROJECT:DATASET.TABLE
    known_args, pipeline_args = parser.parse_known_args(argv)

    # Run the pipeline (all operations are deferred until run() is called).

    p = beam.Pipeline(options=PipelineOptions(pipeline_args))

    (p
    # Read the text file[pattern] into a PCollection.
    | 'Read from a File' >> beam.io.ReadFromText(known_args.input,skip_header_lines=1)
    | 'String To BigQuery Row' >> beam.ParDo(FormatDoFn())
    # Write the output using a "Write" transform
    | 'Write to BigQuery' >> beam.io.WriteToBigQuery(
        known_args.output,
        schema=TABLE_SCHEMA,
        create_disposition=beam.io.BigQueryDisposition.CREATE_IF_NEEDED,
        write_disposition=beam.io.BigQueryDisposition.WRITE_APPEND)

    # Run the pipeline (all operations are deferred until run() is called).
    )
    p.run()
```


Dataflow technologies that customers value

Automated input "sharding"

Removing the requirement to pre-sort input data

Dynamic work rebalancing

Removing the investment in manually balancing the load between resources

Auto-scaling of work resources

Removing the investment in manually scaling resources to match needs

Unified programming model

Developers can express the computation needs regardless of batch or streaming data input

Logical monitoring

Provides developers a logical view of pipeline behavior vs. a control view

On-demand resourcing

All resources are provided on demand, providing nearly limitless resource scale

Why do customers value Dataflow?

Less overhead-one system

Dataflow reduces operational tuning and operation management overhead found in traditional batch or streaming processing systems. Dataflow fully manages resources in Google Cloud Platform on a per-job basis, including spinning up and shutting down workers and accessing Google Cloud Storage buckets for both I/O and temporary file staging.

Price vs. performance

Dataflow provides intelligent optimization (without developer intervention) of resources to provide optimum price and performance.

Dataflow solves this pain points

Lack of existing ETL solution for BigQuery

Attempting to adopt or extend a BigQuery implementation and they need a reliable, scalable way to move, cleanse, and load data into BigQuery.

Dataflow provides optimized ETL for BigQuery.

Hadoop MapReduce cluster at capacity

Have existing investment in on-premise or on-cloud Hadoop, but they're at capacity and can't or don't want to make further investments in hardware or development.

Dataflow provides on demand and nearly limitless elasticity to offload or build new workflows outside of Hadoop.

Struggling to launch streaming Spark cluster

Trying to get a real-time streaming processing application spun up.

Struggling with overhead of self deploying and managing a Spark cluster.

Dataflow provides a unified batch and streaming model that is fully managed.

Developers focus on development, not on operations.

Migrate existing MapReduce

Would like to migrate existing MapReduce or generic batch process to a real-time stream processing model.

Want to take a batch job; for example, process a log file every 24 hours—and migrate it to process a continuous stream of data and report on that data every 5 minutes.

Dataflow solves this pain points

Too much cost with Hadoop cluster

Typically will have 1 (if not 2) dedicated headcount just for cluster management. Dataflow effectively removes this cost, since the service is fully managed.

"Burstiness" of data rates or processing needs

Existing systems; for example, Hadoop cannot dynamically shift resources to meet incoming data rate changes.

This pain point is exacerbated by the fact that their incoming data rates are growing beyond their control, or the business is asking questions faster than they can respond.

Needs large-scale computation to develop metrics for management dashboards

Traditionally, these metrics would be built after data was loaded into their SQL database. This uses more resources, gets slower as data grows, and is troubled by data contention.

Dataflow provides computation language primitives, which can be run as a combination with traditional ETL processing.

Managing two systems: 1 - batch, 1- streaming

More advanced customers may have already implemented both a batch and streaming solution; however, they are then left with managing two different systems.

Dataflow provides a **unified model** for batch and stream processing.

Dataflow use cases

Batch data movement

Moving at rest data from one system to another, such as from Google Cloud Storage to BigQuery

Data reduction and enrichment

Reduce, compress, re-shape existing data into smaller, computed values, such as log files and geo tags

Continuous computation

Analyze real-time streaming inputs, such as click streams

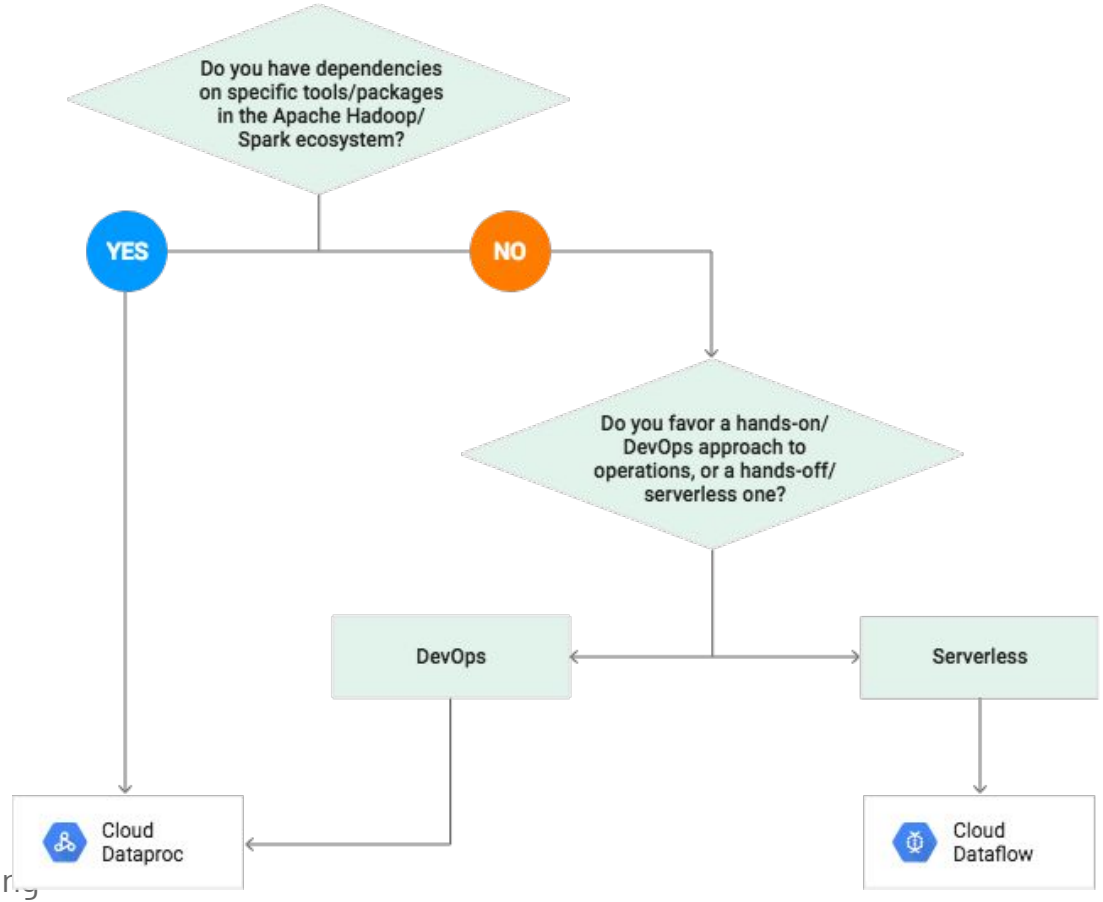
Continuous data movement

Real-time ETL over streaming inputs

Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Dataproc vs Dataflow

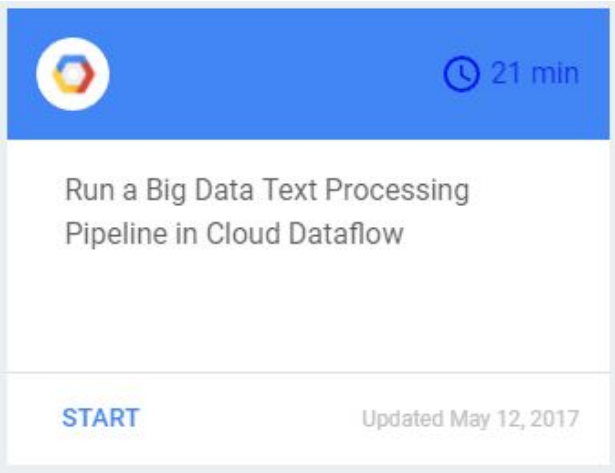


Dataproc vs Dataflow

Recommended Workloads

WORKLOADS	CLOUD DATAPROC	CLOUD DATAFLOW
Stream processing (ETL)		✓
Batch processing (ETL)	✓	✓
Iterative processing and notebooks	✓	
Machine learning with Spark ML	✓	
Preprocessing for machine learning		✓ (with Cloud ML Engine)

Dataflow - Codelab



MUCHAS GRACIAS

UNIVERSIDAD COMPLUTENSE DE MADRID