Cloud Dataflow

A Google Service and SDK

The Presenter

Alex Van Boxel, Software Architect, software engineer, devops, tester,...

at Vente-Exclusive.com

Twitter	@alexvb
Plus	+AlexVanBoxel
E-mail	alex@vanboxel.be
Web	http://alex.vanboxel.be

Dataflow

what, how, where?

Dataflow is...

A set of SDK that define the programming model that you use to **build** your **streaming** and **batch** processing pipeline (*)

Google Cloud Dataflow is a fully managed service that will **run and optimize** your pipeline

Dataflow where...

ETL

- Move
- Filter
- Enrich

Analytics

- Streaming Compu
- Batch Compu
- Machine Learning

Composition

- Connecting in Pub/Sub
  - Could be other dataflows
  - Could be something else

Cloud Dataflow Benefits

NoOps Model

- Resources allocated on demand
- Lifecycle managed by the service
- Optimization
- Rebalancing

Cloud Dataflow Benefits

Unified Programming Model

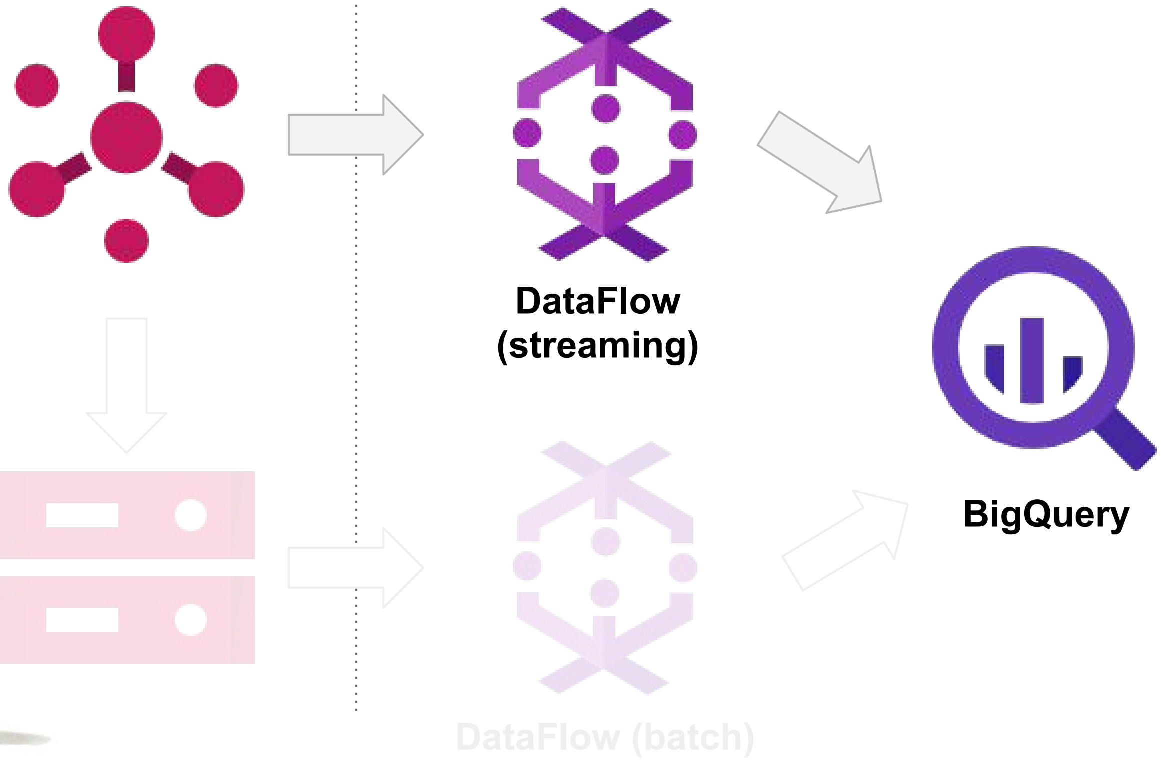
Unified: Streaming and Batch

Open Sourced

- Java 8 implementation
- Python in the works

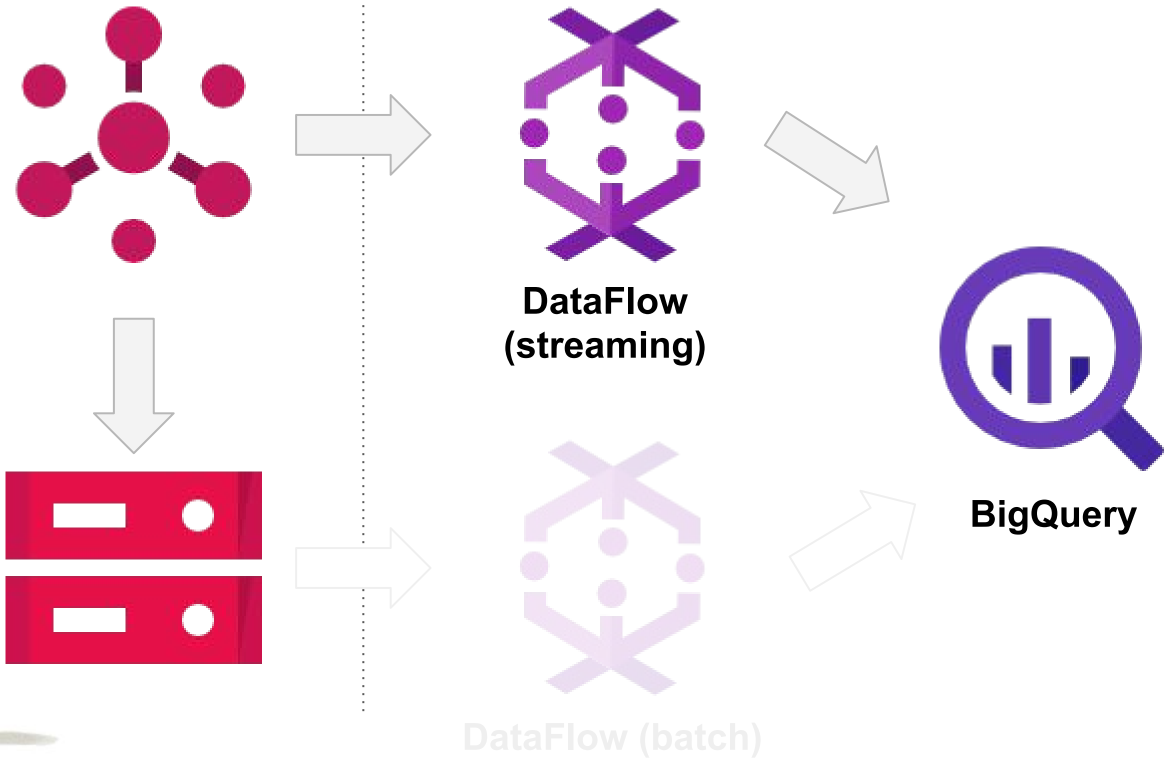
Cloud Dataflow Benefits

Unified Programming Model (streaming)



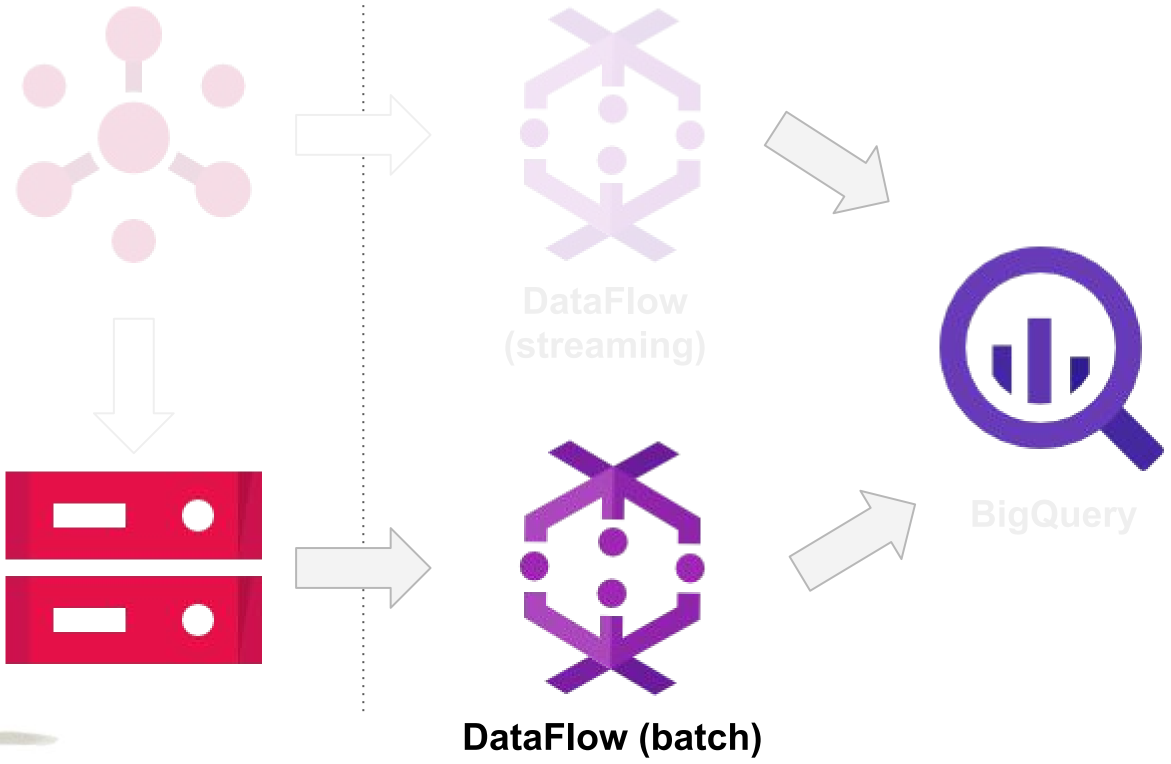
Cloud Dataflow Benefits

Unified Programming Model (stream and backup)



Cloud Dataflow Benefits

Unified Programming Model (batch)

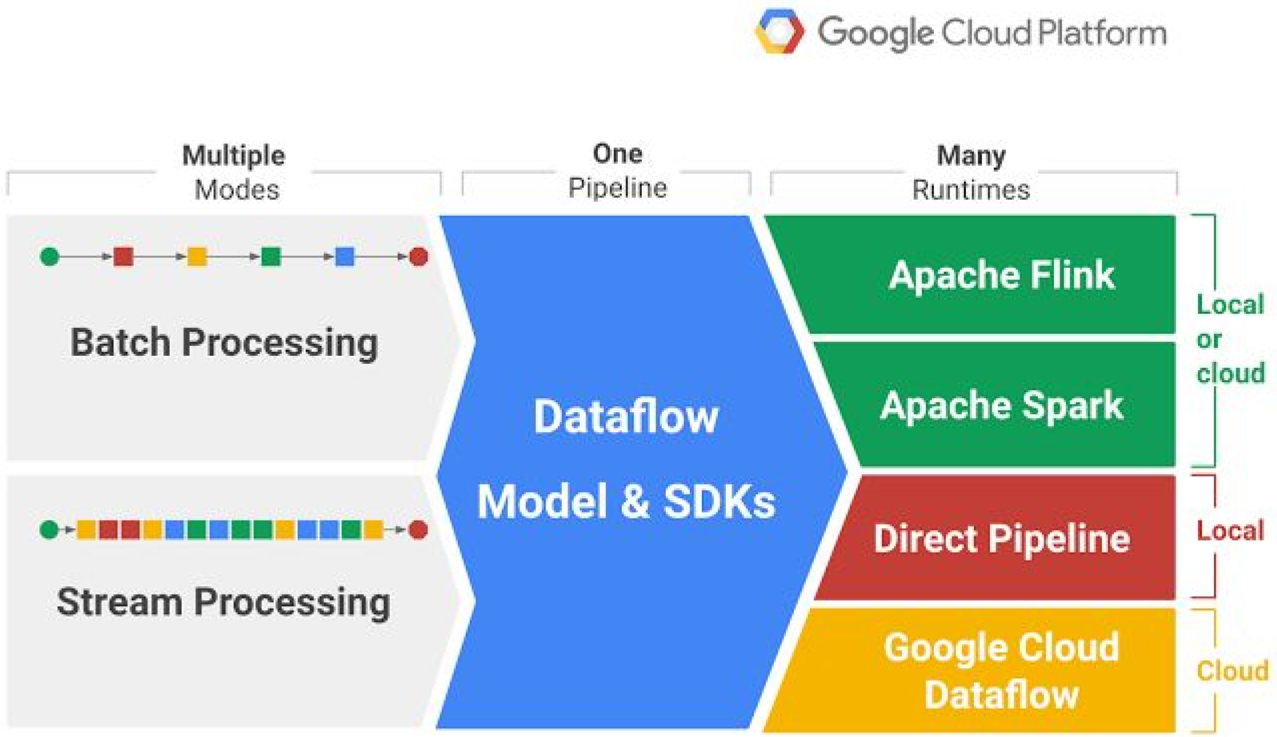


Beam

Apache Incubation

Cloud Dataflow Benefits

DataFlow SDK becomes Apache Beam



DataFlow SDK becomes Apache Beam

- **Pipeline first, runtime second**
  - focus your data pipelines, not the characteristics runner
- **Portability**
  - portable across a number of runtime engines
  - runtime based on any number of considerations
    - performance
    - cost
    - scalability

DataFlow SDK becomes Apache Beam

- **Unified model**
  - Batch and streaming are integrated into a unified model
  - Powerful semantics, such as windowing, ordering and triggering
- **Development tooling**
  - tools you need to create portable data pipelines quickly and easily using open-source languages, libraries and tools.

DataFlow SDK becomes Apache Beam

- **Scala** Implementation
- Alternative Runners
  - **Spark** Runner from Cloudera Labs
  - **Flink** Runner from data Artisans

DataFlow SDK becomes Apache Beam

- Cloudera
- data Artisans
- Talend
- Cask
- PayPal

SDK Primitives

Autopsy of a dataflow pipeline

Pipeline

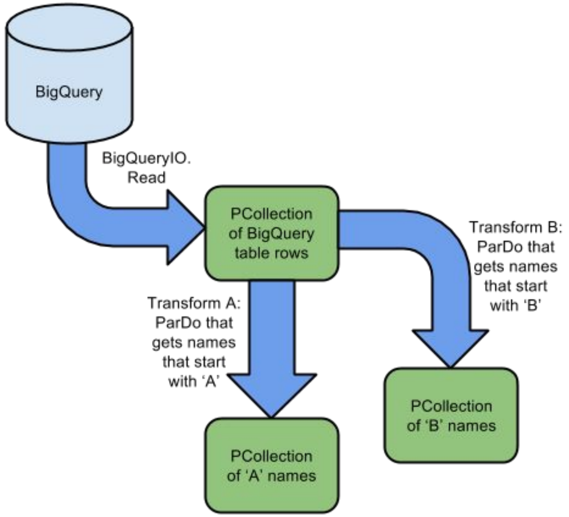
- **Inputs**: Pub/Sub, BigQuery, GCS, XML, JSON, …
- **Transforms**: Filter, Join, Aggregate, …
- **Windows**: Sliding, Sessions, Fixed, …
- **Triggers**: Correctness….
- **Outputs**: Pub/Sub, BigQuery, GCS, XML, JSON, ...

PCollection<T>

- Immutable collection
- Could be bound or unbound
- Created by
  - a backing data store
  - a transformation from other PCollection
  - generated
- PCollection<KV<K,V>>

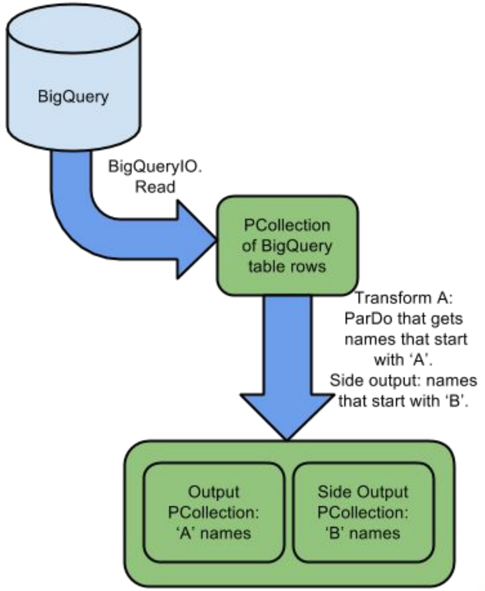
ParDo<I,O>

- Parallel Processing of each element in the PCollection
- Developer provide DoFn<I,O>
- Shards processed independent of each other


BigQuery
BigQueryIO.
Read
PCollection of BigQuery table rows
Transform A: ParDo that gets names that start with 'A'
Transform B: ParDo that gets names that start with 'B'
PCollection of 'A' names
PCollection of 'B' names

ParDo<I,O>

Interesting concept of side outputs: allows for data sharing


BigQuery
BigQueryIO. Read
PCollection of BigQuery table rows
Transform A: ParDo that gets names that start with 'A'. Side output: names that start with 'B'.
Output PCollection: 'A' names
Side Output PCollection: 'B' names

Dataflow SDK Primitives

PTransform

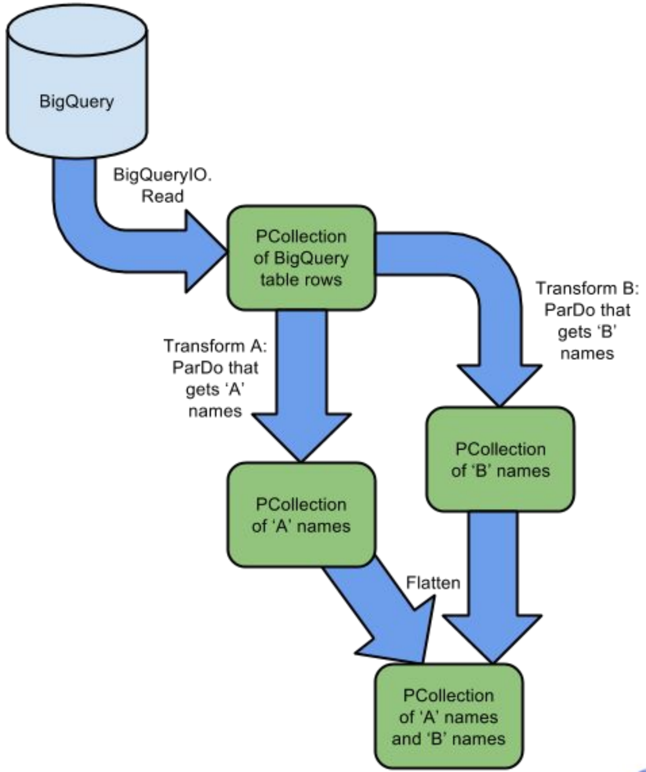
- Combine small operations into bigger transforms
- Standard transform (COUNT, SUM,...)
- Reuse and Monitoring

```
public class BlockTransform extends PTransform<PCollection<Legacy>, PCollection<Event>> {

  @Override
  public PCollection<Event> apply(PCollection<Legacy> input) {
    return input.apply(ParDo.of(new TimestampEvent1Fn()))
            .apply(ParDo.of(new FromLegacy2EventFn())).setCoder(AvroCoder.of(Event.class))
            .apply(Window.<Event>into(Sessions.withGapDuration(Duration.standardMinutes(20))))
            .apply(ParDo.of(new ExtactChannelUserFn()))
  }
}
```

Dataflow SDK Primitives

Merging

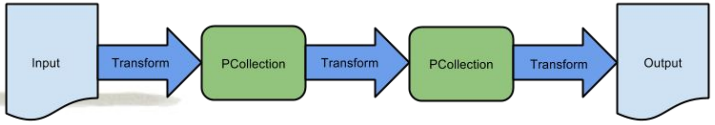
- **GroupByKey**
  - Takes a PCollection of KV<K,V> and groups them
  - Must be keyed
- Flatten
  - Join with same type


BigQuery
BigQueryIO. Read
PCollection of BigQuery table rows
Transform A: ParDo that gets 'A' names
Transform B: ParDo that gets 'B' names
PCollection of 'A' names
PCollection of 'B' names
Flatten
PCollection of 'A' names and 'B' names

Input/Output

```
PipelineOptions options = PipelineOptionsFactory.create();
Pipeline p = Pipeline.create(options);
PCollection<TableRow> weatherData = p
       .apply(BigQueryIO.Read.named("ReadWeatherStations")
       .from("clouddataflow-readonly:samples.weather_stations"));
```

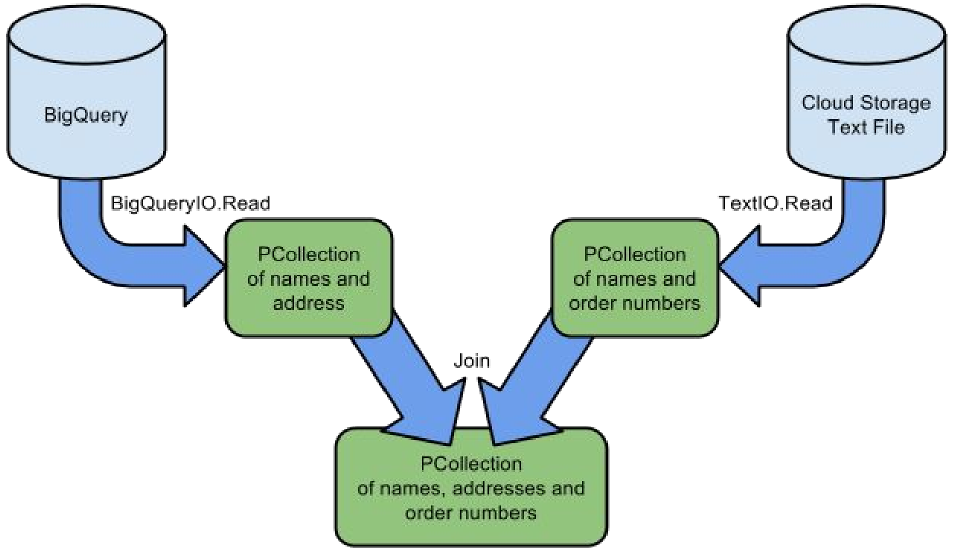
```
PipelineOptions options = PipelineOptionsFactory.create();
Pipeline p = Pipeline.create(options);

// streamData is Unbounded; apply windowing afterward.
PCollection<String> streamData =
       p.apply(PubsubIO.Read.named("ReadFromPubsub")
               .topic("/topics/my-topic"));
```



Dataflow SDK Primitives

Input/Output

- Multiple Inputs + Outputs


BigQuery
Cloud Storage Text File
BigQueryIO.Read
TextIO.Read
PCollection of names and address
PCollection of names and order numbers
Join
PCollection of names, addresses and order numbers

Primitives on steroids

Autopsy of a dataflow pipeline

Windows

- Split unbounded collections (ex. Pub/Sub)
- Works on bounded collection as well

```
.apply(Window.<Event>into(Sessions.withGapDuration(Duration.standardMinutes(20))))
```

```
Window.<CAL>into(SlidingWindows.of(Duration.standardMinutes(5)));
```

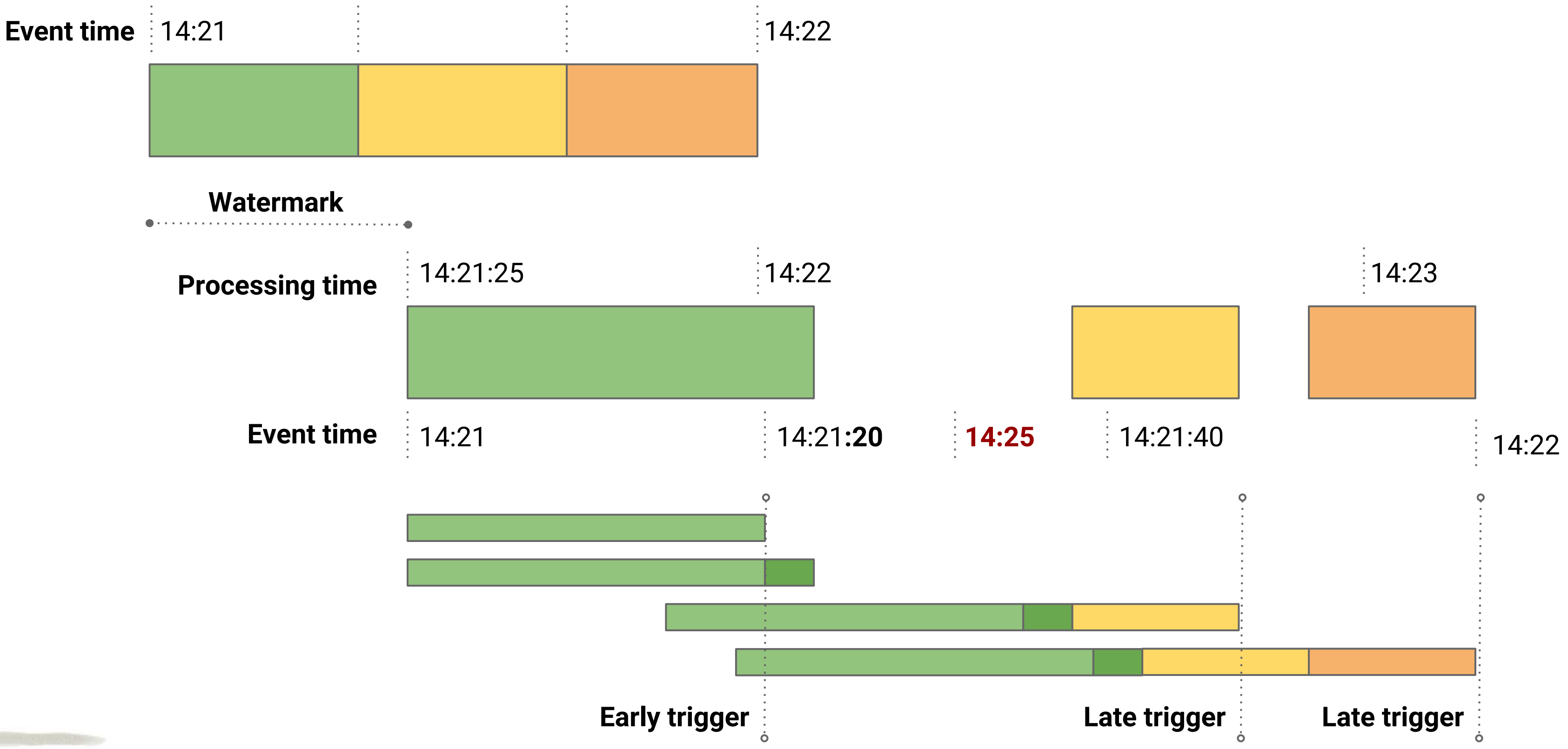
Triggers

- Time-based
- Data-based
- Composite

```
AfterEach.inOrder(
        AfterWatermark.pastEndOfWindow(),
        Repeatedly.forever(AfterProcessingTime
                 .pastFirstElementInPane()
                 .plusDelayOf(Duration.standardMinutes(10)))
                 .orFinally(AfterWatermark
                          .pastEndOfWindow()
                          .plusDelayOf(Duration.standardDays(2))));
```

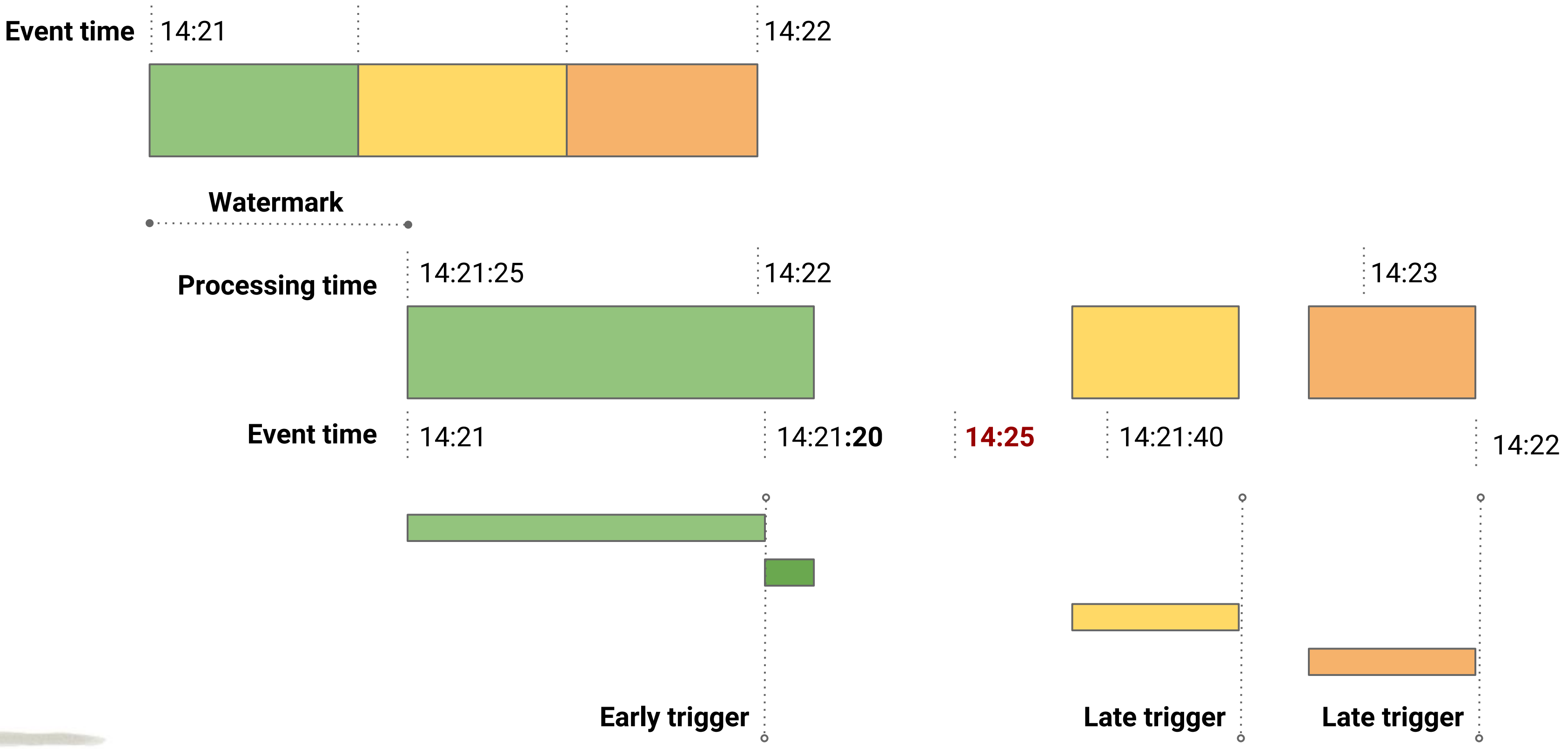
Dataflow SDK Primitives

Input/Output



Dataflow SDK Primitives

Input/Output



Google Cloud Dataflow

The service

What you need

- Google Cloud SDK
  - utilities to authenticate and manage project
- Java IDE
  - Eclipse
  - **IntelliJ**
- Standard Build System
  - Maven
  - **Gradle**

What you need

```
subprojects {
   apply plugin: 'java'

   repositories {
       maven {
           url  "http://jcenter.bintray.com"
       }
       mavenLocal()
   }

   dependencies {
       compile 'com.google.cloud.dataflow:google-cloud-dataflow-java-sdk-all:1.3.0'
       testCompile 'org.hamcrest:hamcrest-all:1.3'
       testCompile 'org.assertj:assertj-core:3.0.0'
       testCompile 'junit:junit:4.12'
   }
}
```

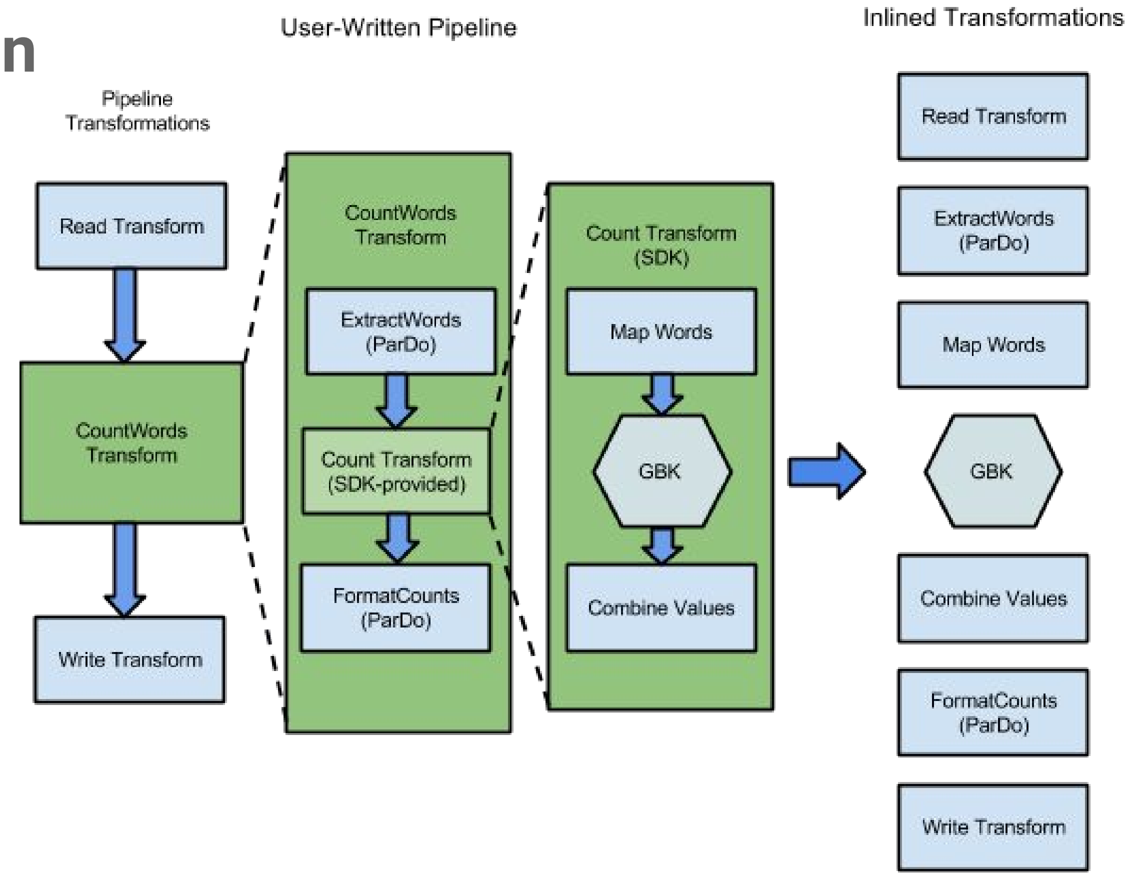
How it work

```
Pipeline p = Pipeline.create(options);

    p.apply(TextIO.Read.from("gs://dataflow-samples/shakespeare/*"))

     .apply(FlatMapElements.via((String word) -> Arrays.asList(word.split("[^a-zA-Z']+")))
         .withOutputType(new TypeDescriptor<String>() {}))
     .apply(Filter.byPredicate((String word) -> !word.isEmpty()))
     .apply(Count.<String>perElement())
     .apply(MapElements
         .via((KV<String, Long> wordCount) -> wordCount.getKey() + ": " + wordCount.getValue())
         .withOutputType(new TypeDescriptor<String>() {}))

     .apply(TextIO.Write.to("gs://YOUR_OUTPUT_BUCKET/AND_OUTPUT_PREFIX"));
```

Service

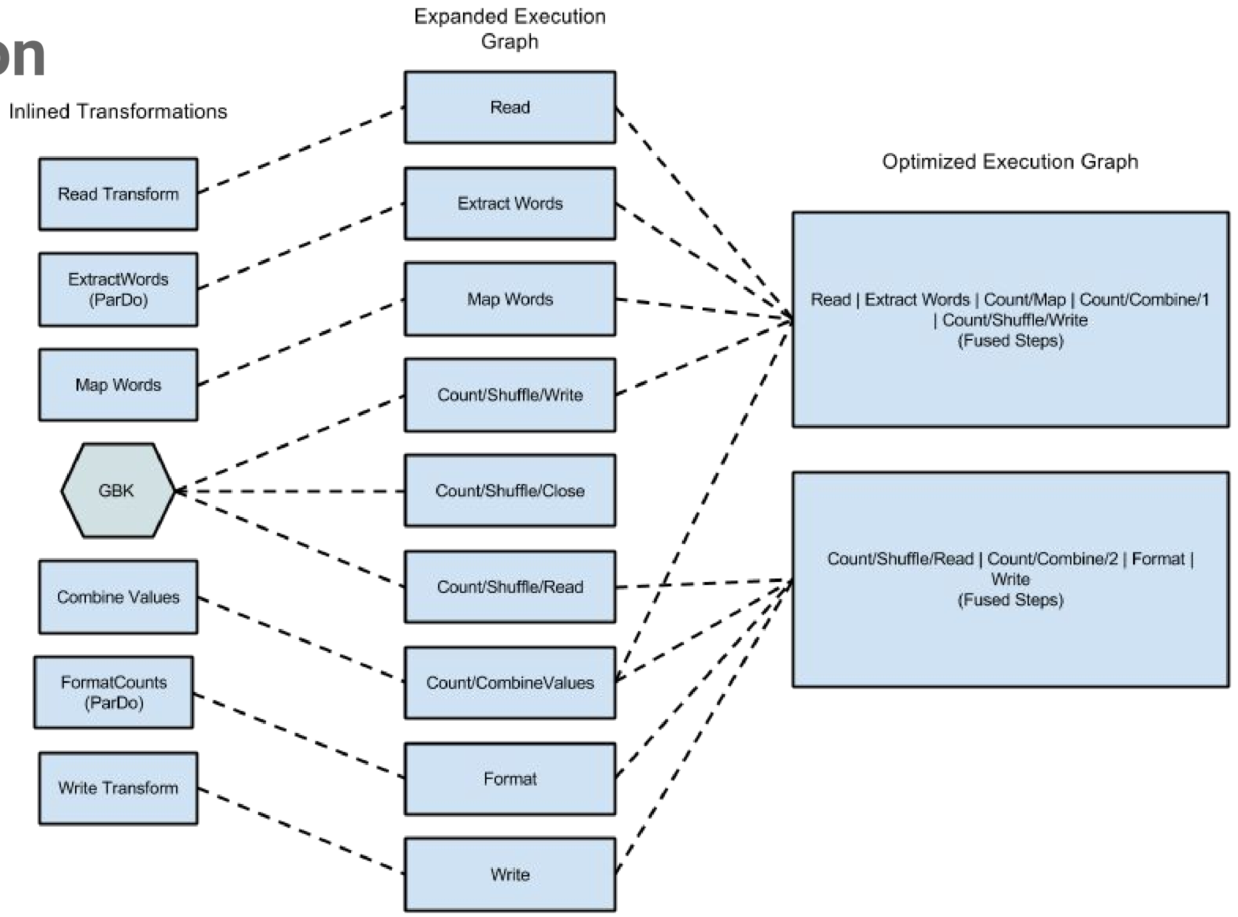
- Stage
- Optimize (Flume)
- Execute

Optimization



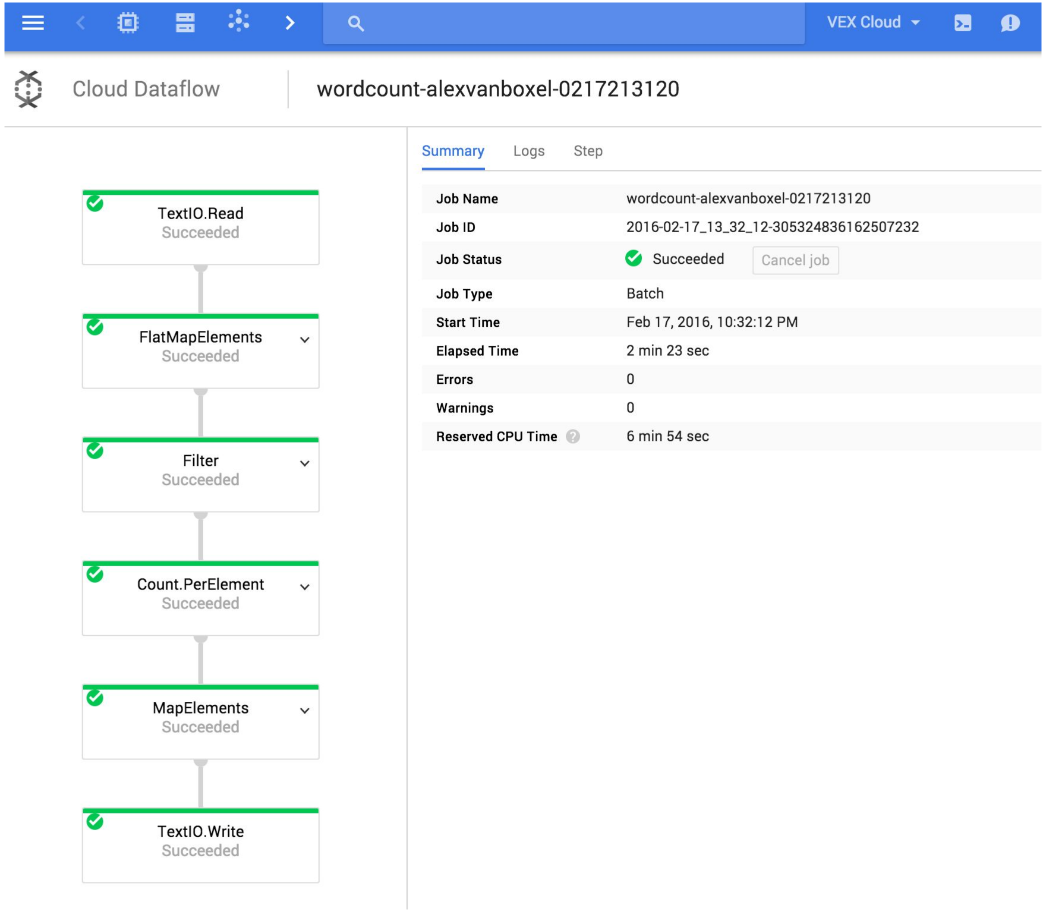
Google Cloud Dataflow Service

Optimization



Google Cloud Dataflow Service

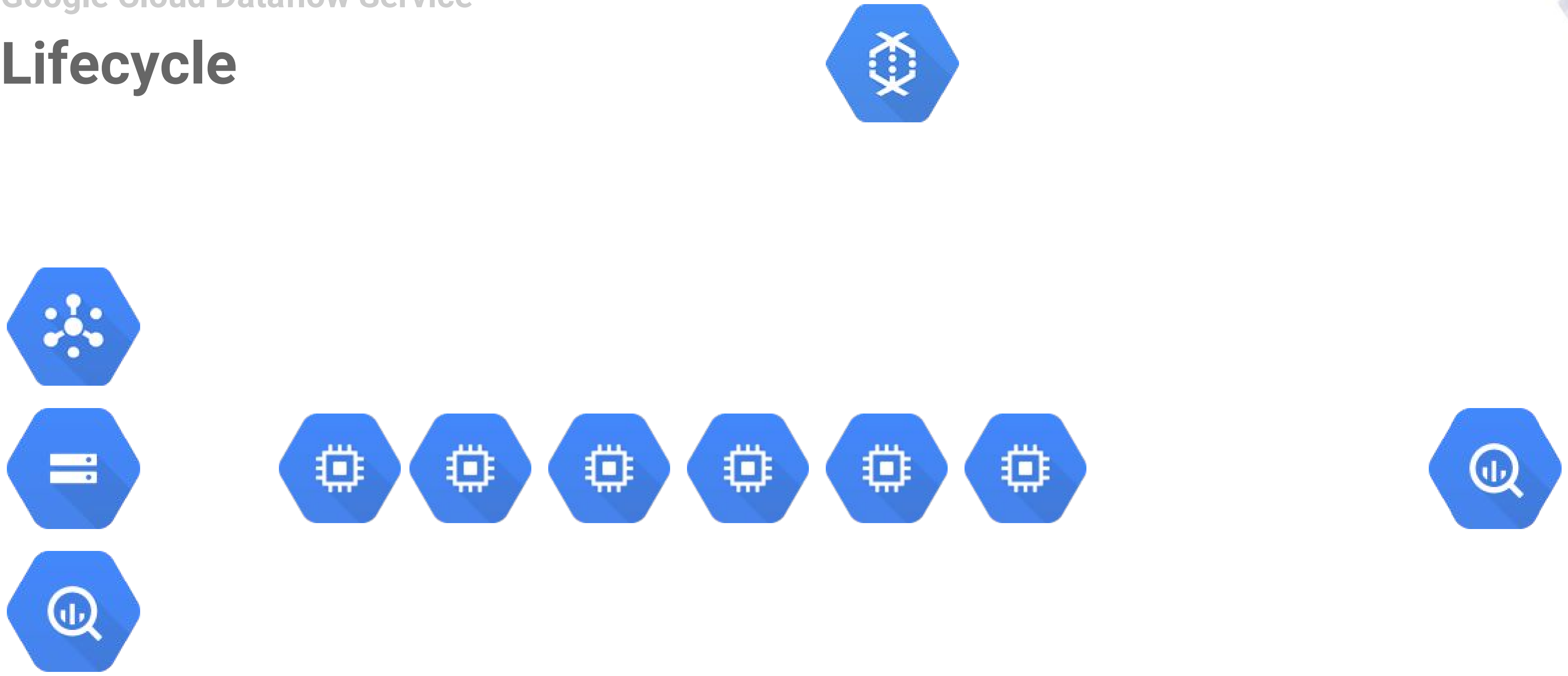
Console



wordcount-alexvanboxel-0217213120

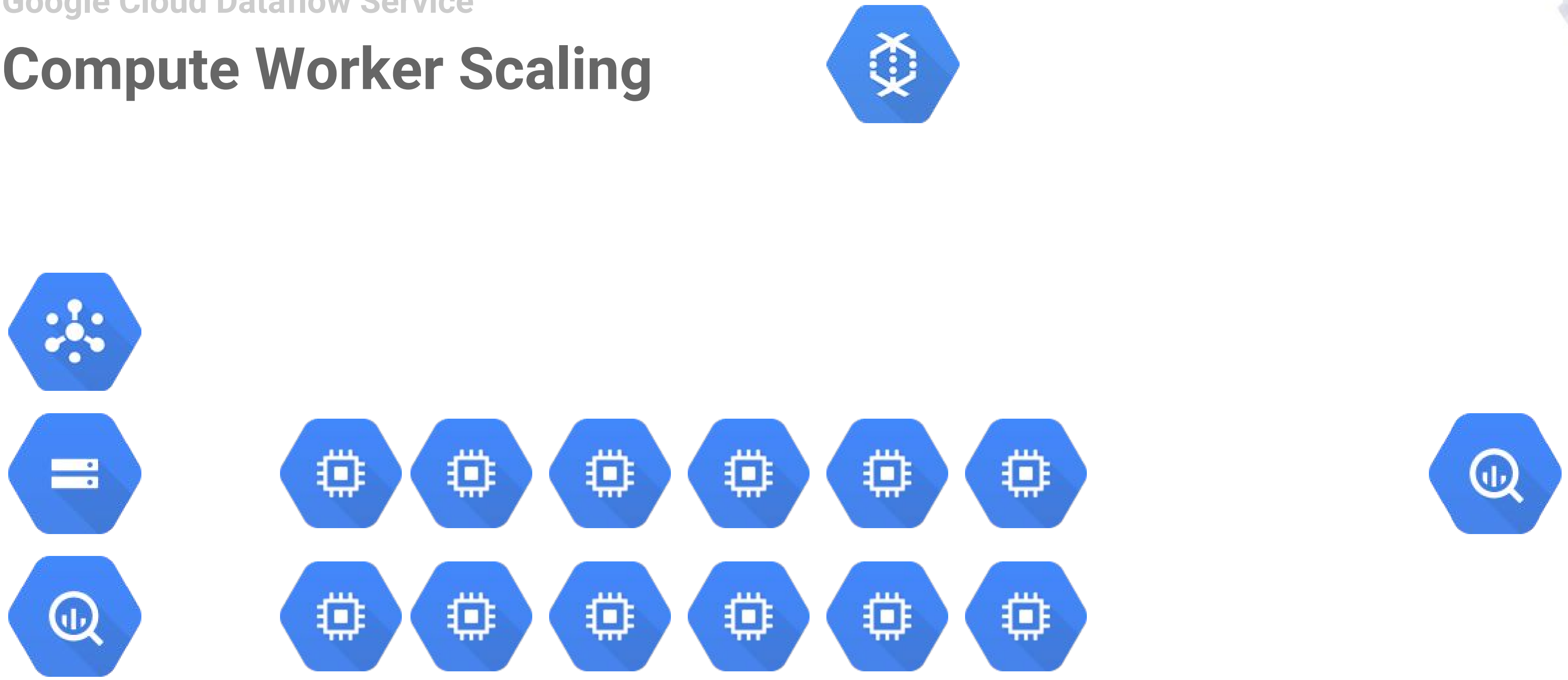
Job Name	wordcount-alexvanboxel-0217213120
Job ID	2016-02-17_13_32_12-305324836162507232
Job Status	✓ Succeeded
Job Type	Batch
Start Time	Feb 17, 2016, 10:32:12 PM
Elapsed Time	2 min 23 sec
Errors	0
Warnings	0
Reserved CPU Time	6 min 54 sec

Lifecycle

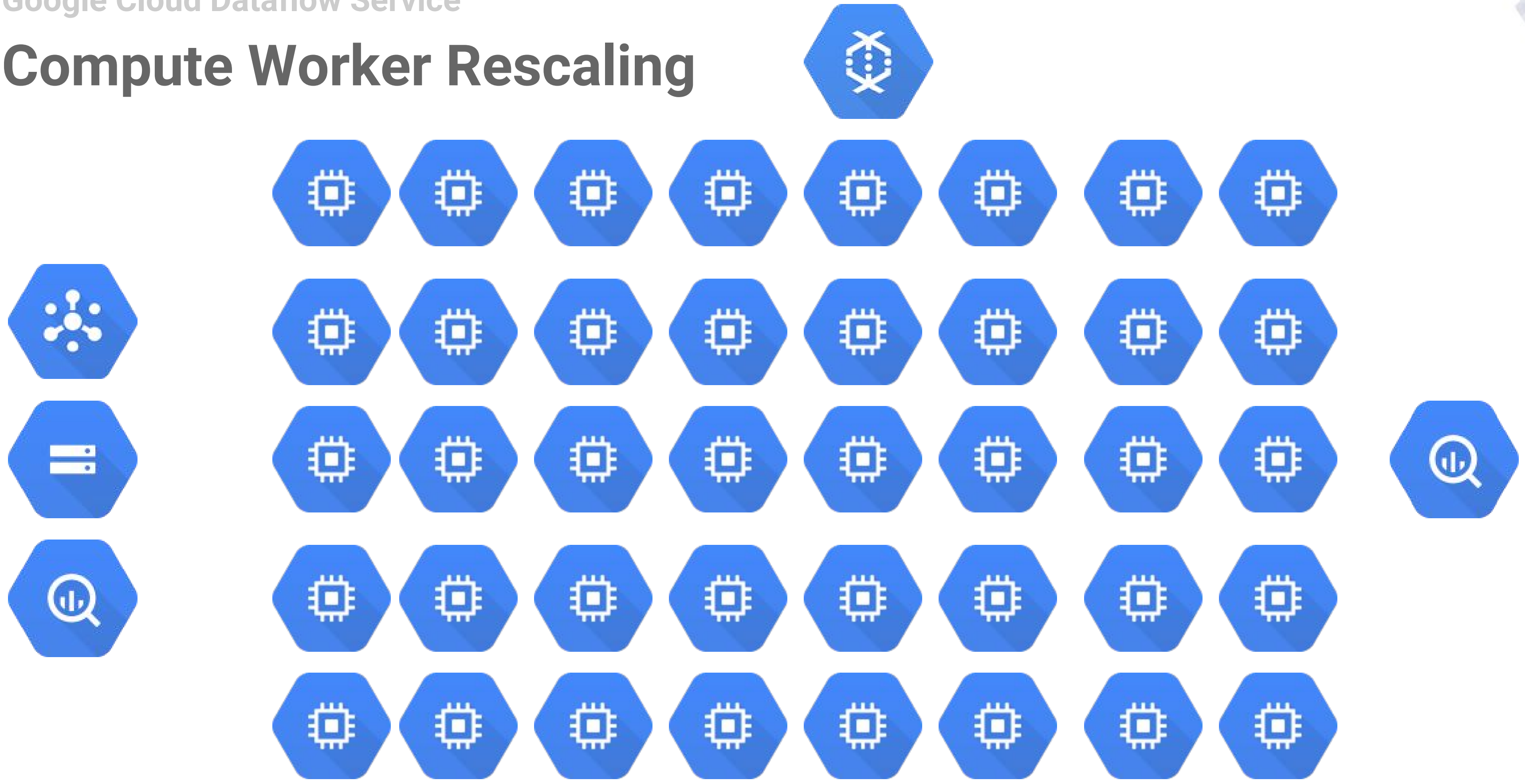


Google Cloud Dataflow Service

Compute Worker Scaling

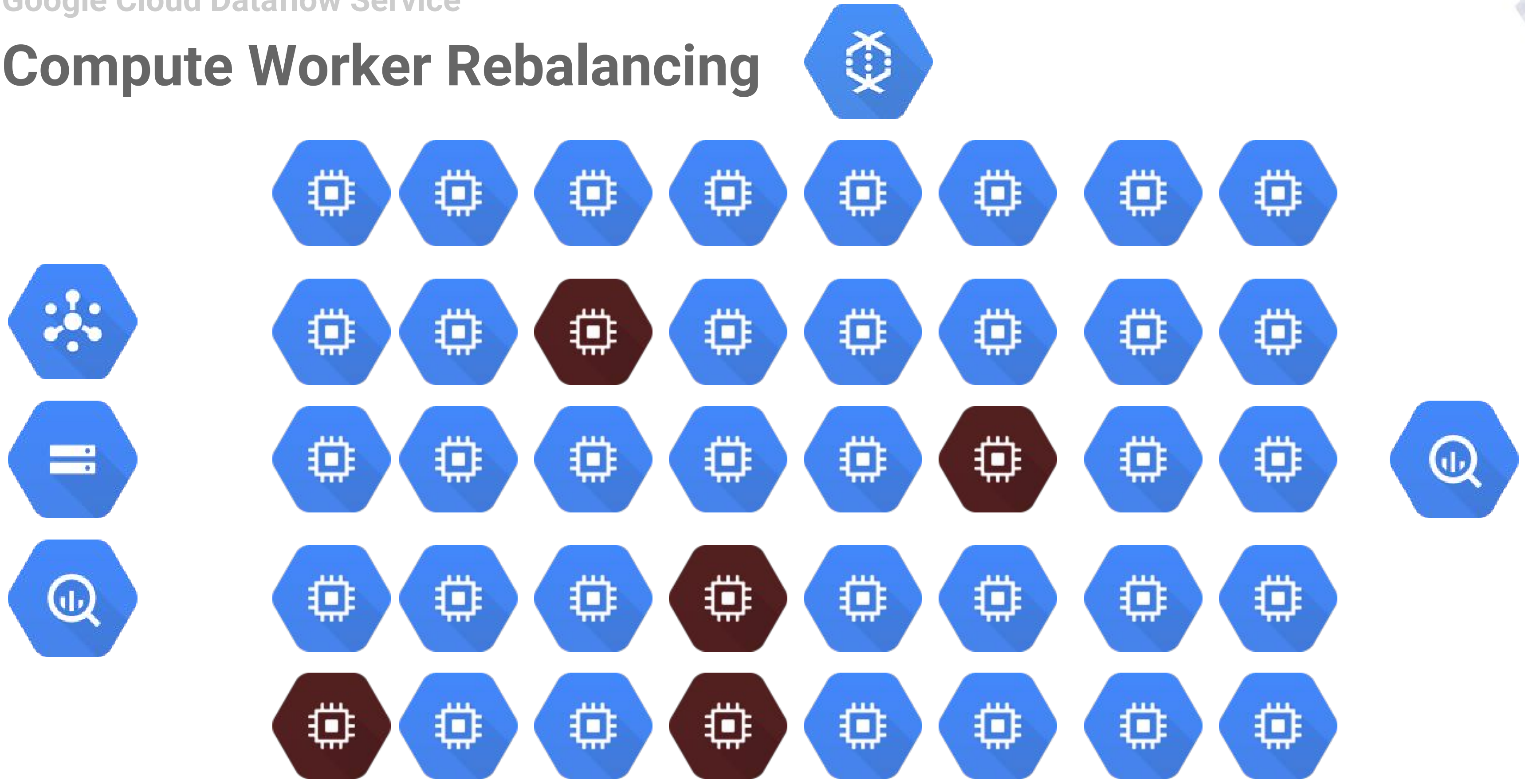


Compute Worker Rescaling



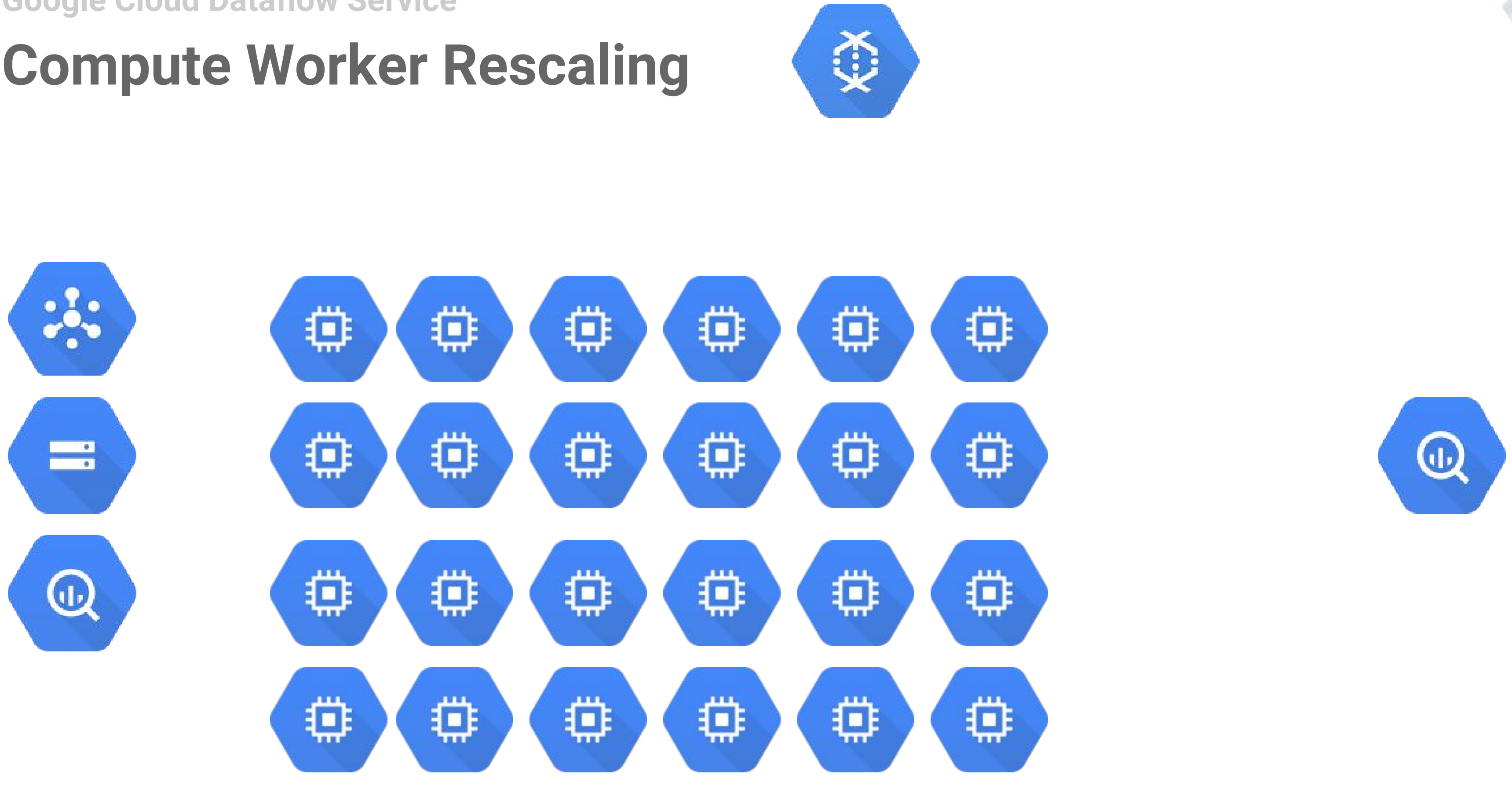
Google Cloud Dataflow Service

Compute Worker Rebalancing



Google Cloud Dataflow Service

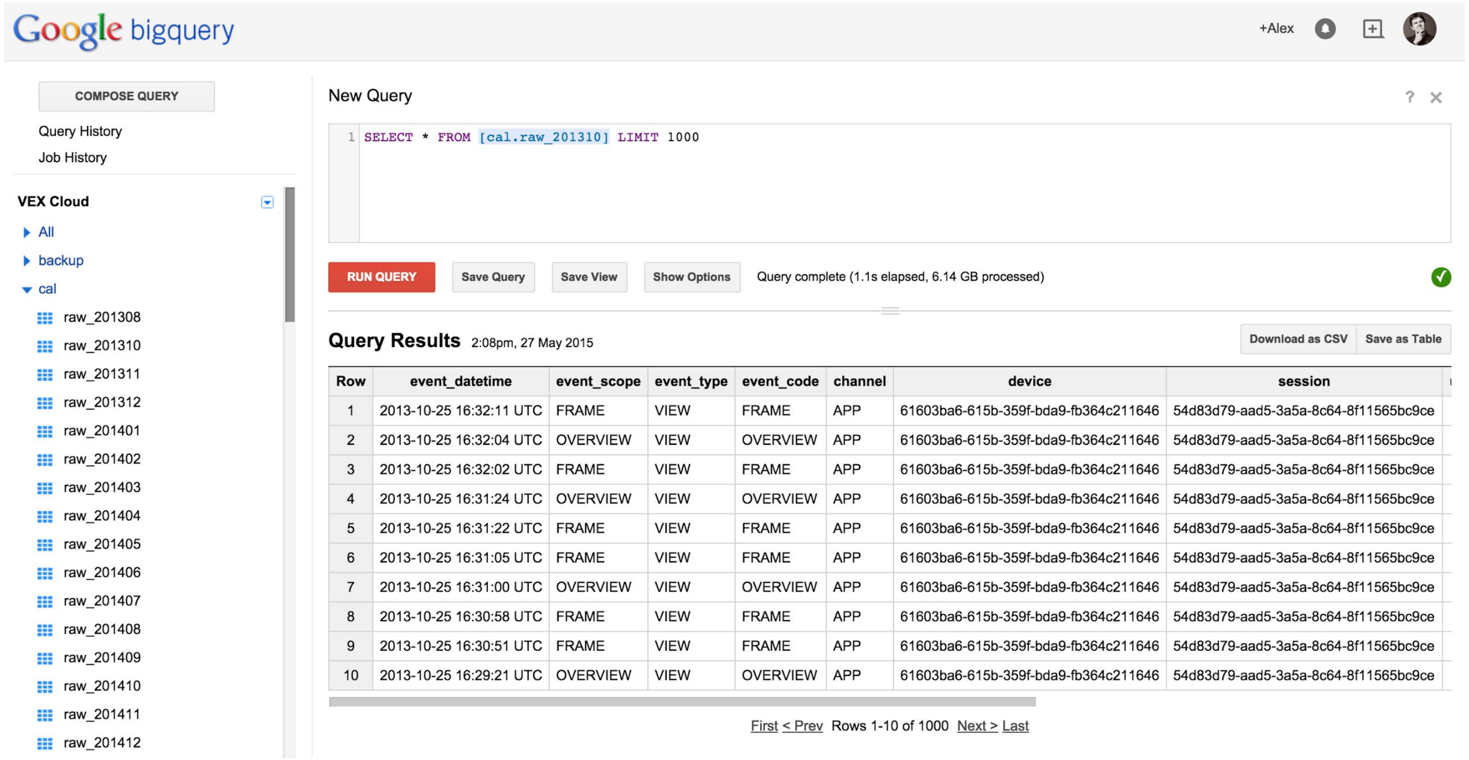
Compute Worker Rescaling



Lifecycle

Google Cloud Dataflow Service

BigQuery



Google bigquery

+Alex

COMPOSE QUERY

Query History

Job History

VEX Cloud

- All
- backup
- cal
  - raw_201308
  - raw_201310
  - raw_201311
  - raw_201312
  - raw_201401
  - raw_201402
  - raw_201403
  - raw_201404
  - raw_201405
  - raw_201406
  - raw_201407
  - raw_201408
  - raw_201409
  - raw_201410
  - raw_201411
  - raw_201412

New Query

```
SELECT * FROM [cal.raw_201310] LIMIT 1000
```

RUN QUERY | Save Query | Save View | Show Options

Query complete (1.1s elapsed, 6.14 GB processed)

Query Results 2:08pm, 27 May 2015

Download as CSV | Save as Table

Row	event_datetime	event_scope	event_type	event_code	channel	device	session
1	2013-10-25 16:32:11 UTC	FRAME	VIEW	FRAME	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
2	2013-10-25 16:32:04 UTC	OVERVIEW	VIEW	OVERVIEW	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
3	2013-10-25 16:32:02 UTC	FRAME	VIEW	FRAME	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
4	2013-10-25 16:31:24 UTC	OVERVIEW	VIEW	OVERVIEW	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
5	2013-10-25 16:31:22 UTC	FRAME	VIEW	FRAME	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
6	2013-10-25 16:31:05 UTC	FRAME	VIEW	FRAME	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
7	2013-10-25 16:31:00 UTC	OVERVIEW	VIEW	OVERVIEW	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
8	2013-10-25 16:30:58 UTC	FRAME	VIEW	FRAME	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
9	2013-10-25 16:30:51 UTC	FRAME	VIEW	FRAME	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce
10	2013-10-25 16:29:21 UTC	OVERVIEW	VIEW	OVERVIEW	APP	61603ba6-615b-359f-bda9-fb364c211646	54d83d79-aad5-3a5a-8c64-8f11565bc9ce

First < Prev Rows 1-10 of 1000 Next > Last

Testability

Test Driven Development Build-In

Testability - Built in

```
    KV<String, List<CalEvent>> input = KV.of("", new ArrayList<>());
    input.getValue().add(CalEvent.event("SESSION", "REFERRAL")
            .build());
    input.getValue().add(CalEvent.event("SESSION", "START")
            .data("{\"referral\":{\"Sourc...}").build());

    DoFnTester<KV<?, List<CalEvent>>, CalEvent> fnTester = DoFnTester.of(new
 SessionRepairFn());
    fnTester.processElement(input);
    List<CalEvent> calEvents = fnTester.peekOutputElements();
    assertThat(calEvents).hasSize(2);

    CalEvent referral = CalUtil.findEvent(calEvents, "REFERRAL");
    assertNotNull("REFERRAL should still exist", referral);

    CalEvent start = CalUtil.findEvent(calEvents, "START");
    assertNotNull("START should still exist", start);
    assertNull("START should have referral removed.",
```

Appendix

Follow the yellow brick road

Paper - Flume Java

FlumeJava: Easy, Efficient Data-Parallel Pipelines

http://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/FlumeJava.pdf

DataFlow/Bean vs Spark

Dataflow/Beam & Spark: A Programming Model Comparison

https://cloud.google.com/dataflow/blog/dataflow-beam-and-spark-comparison

Github - Integrate Dataflow with Luigi

Google Cloud integration for Luigi

https://github.com/alexvanboxel/luigiext-gcloud

The last slide

Questions and Answers

Twitter	@alexvb
Plus	+AlexVanBoxel
E-mail	alex@vanboxel.be
Web	http://alex.vanboxel.be