Teraslice

Stream Processing with NodeJS

https://github.com/terascope/teraslice

Austin Godber
@godber@az.social
May 17th, 2023

Should I use Teraslice?

CASE
EXTENDAHOE
UTILITY PLUS
CASE
UTILITY PLUS

CAT
7495
CAT
1
1

RHEINBRAUN
288
Rheinbraun
SIEMENS

Outline

- ETL and Stream Processing
- Teraslice Description and Features
- Teraslice Job
- Example Problem
- Teraslice Processor

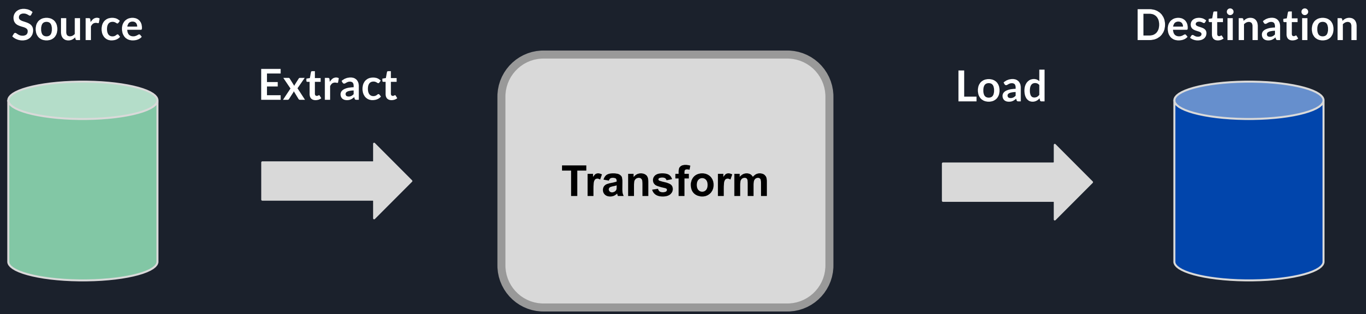
Outline

- ETL and Stream Processing
- Teraslice Description and Features
- Teraslice Job
- Example Problem
- Teraslice Processor

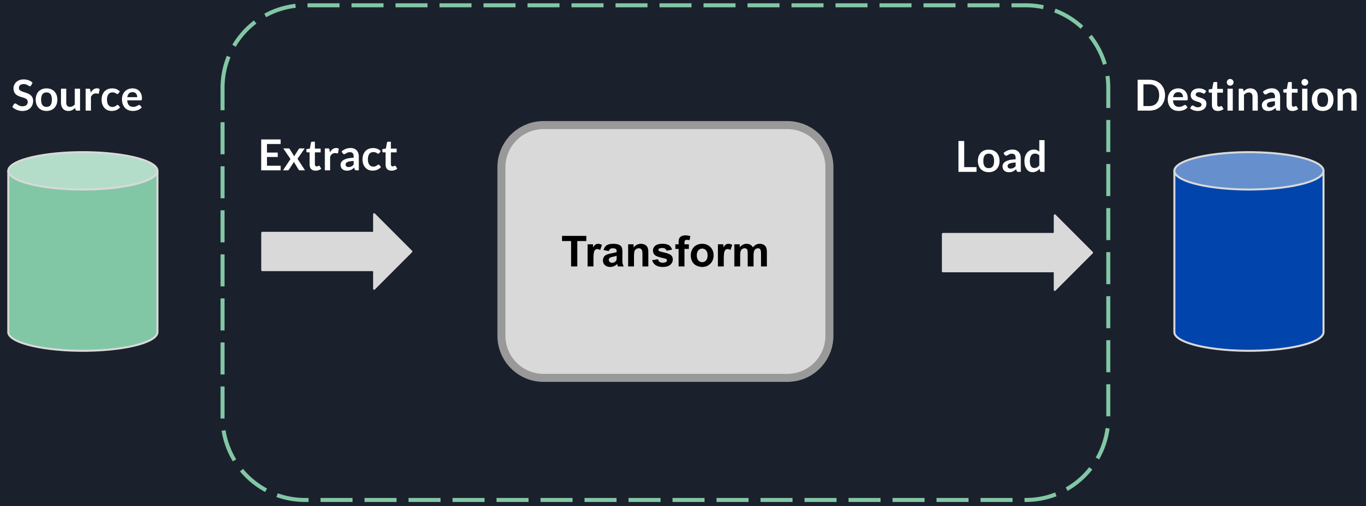
Extract Transform and Load

- Extract
  - Reading data from a data source like a database
  - Validation of the read data
- Transform
  - Modify the data according to need
    - Filtering
    - Mapping
    - Creating derived values
    - Deduplication
    - Enrichment from other sources
    - etc
- Load
  - Write modified data to destination system
  - Write audit log

Extract Transform and Load

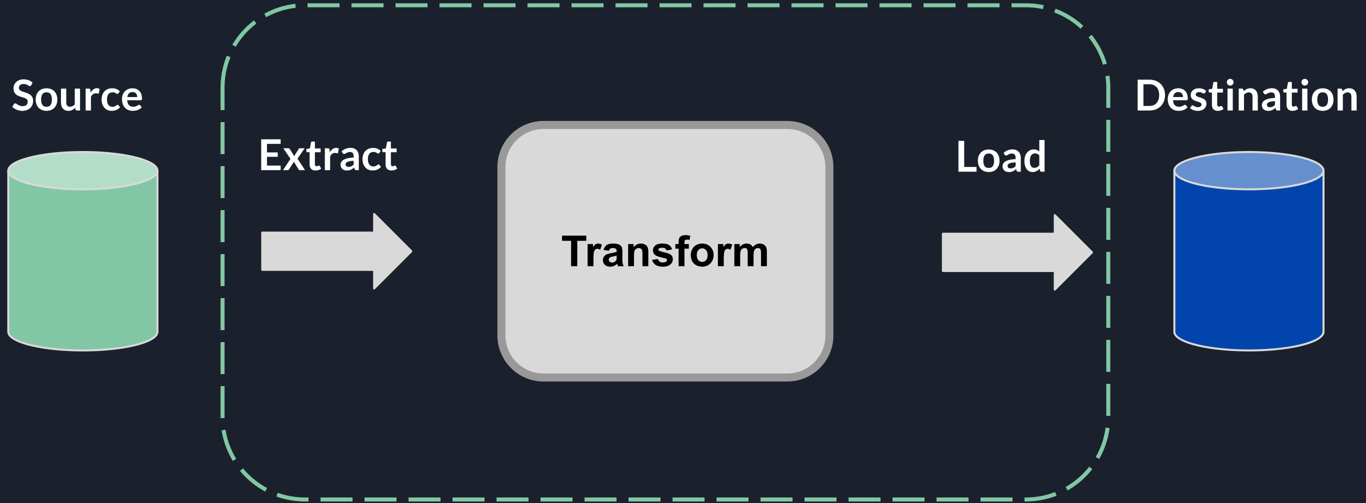


Extract Transform and Load



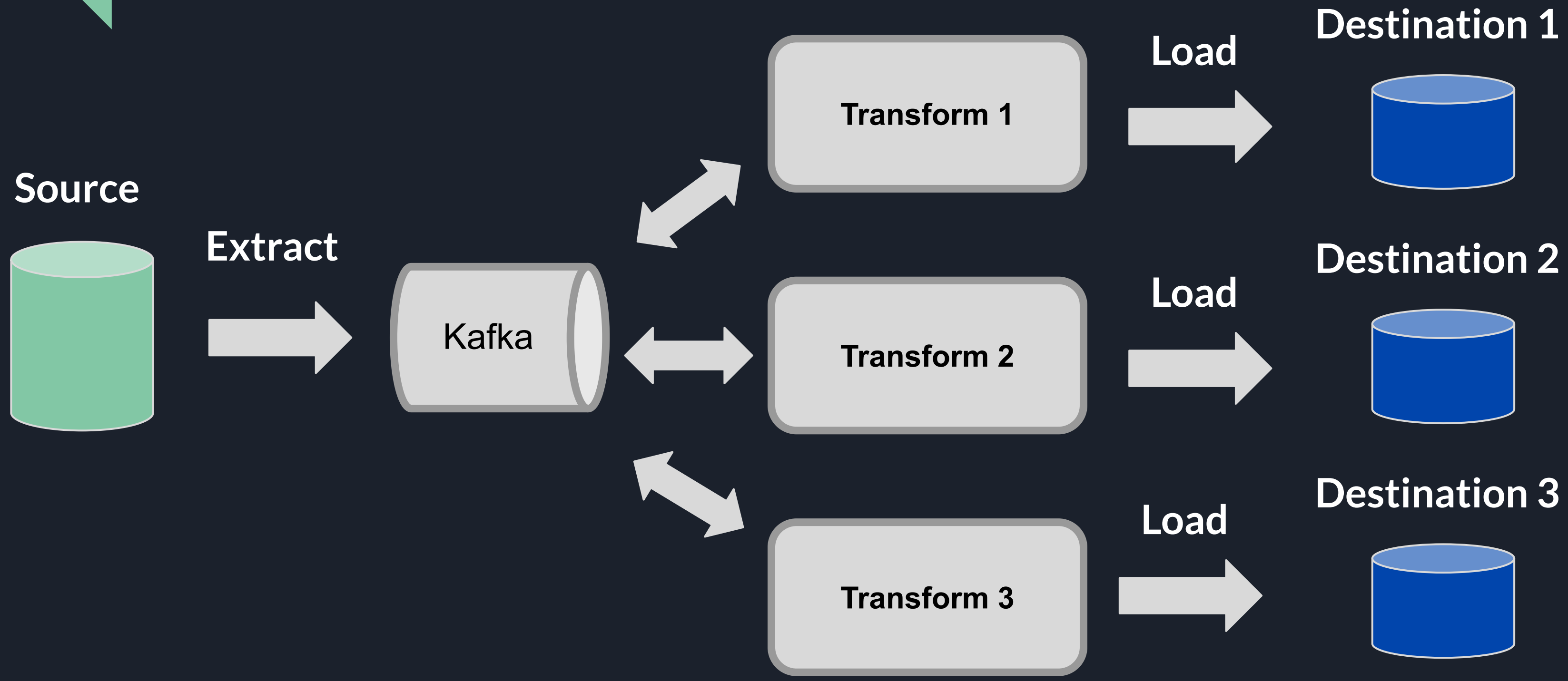
Usually batchy!

Extract Transform and Load



Maybe a single program

Streaming ETL



Outline

- ETL and Stream Processing
- Teraslice Description and Features
- Teraslice Job
- Example Problem
- Teraslice Processor

Teraslice Background

“Teraslice provides scalable data processing pipelines implemented using components written in JavaScript. It uses a distributed model to spread work across a cluster of computers and can easily process millions of records per second.”

https://terascope.github.io/teraslice/

Teraslice Background

Features

- Originally designed to “re-index” large (100B+) Elasticsearch Indices
- Enhanced to become a generic stream processing tool
- Written in Javascript/Typescript
- Custom “processors” can be implemented in Javascript and uploaded separately
- Management REST API
    - https://terascope.github.io/teraslice/docs/management-apis/overview
- Jobs are defined in JSON
    - https://terascope.github.io/teraslice/docs/jobs/configuration#examples
- Runs locally with native clustering mechanism or in Kubernetes

Teraslice Background

Components

- https://github.com/terascope/teraslice
- https://github.com/terascope/elasticsearch-assets
- https://github.com/terascope/kafka-assets
- https://github.com/terascope/file-assets
- https://github.com/terascope/standard-assets

Teraslice Alternatives

- Vector (vector.dev) (Rust)
- Spark Streaming (Java)
- Flink (Java)
- Kafka Streams (Java)
- Opensearch Data Prepper Pipelines (Config)
- Elasticsearch Ingest Pipelines (Config)

Outline

- ETL and Stream Processing
- Teraslice Description and Features
- Teraslice Job
- Example Problem
- Teraslice Processor

```
{
   "name": "Data Generator To Kafka",
   "lifecycle": "persistent",
   "workers": 1,
   "assets": [
       "standard",
       "kafka"
   ],
   "operations": [
       {
           "_op": "data_generator",
           "size": 500000
       },
       {
           "_op": "kafka_sender",
           "connection": "default",
           "topic": "test1",
           "size": 25000
       }
   ]
}
```

```
{
    "name": "Data Generator To Kafka",
    "lifecycle": "persistent",
    "workers": 1,
    "assets": [
        "standard:v1.1.15",
        "kafka"
    ],
    "operations": [
        {
            "_op": "data_generator",
            "size": 500000
        },
        {
            "_op": "kafka_sender",
            "connection": "default",
            "topic": "test1",
            "size": 25000
        }
    ]
}
```

Pin Version

```
{
    "name": "Data Generator To Kafka",
    "lifecycle": "persistent",
    "workers": 1,
    "assets": [
        "standard",
        "kafka"
    ],
    "operations": [
        {
            "_op": "data_generator",
            "size": 500000
        },
        {
            "_op": "kafka_sender",
            "connection": "default",
            "topic": "test1",
            "size": 25000
        }
    ]
}
```

$\left.\begin{matrix}\\ \\ \\ \\ \\ \end{matrix}\right\}$ **Processors**

```
{
    "name": "Data Generator To Kafka",
    "lifecycle": "persistent",
    "workers": 1,
    "assets": [
        "standard",
        "kafka"
    ],
    "operations": [
        {
            "_op": "data_generator",
            "size": 500000
        },
        {
            "_op": "kafka_sender",
            "connection": "default",
            "topic": "test1",
            "size": 25000
        }
    ]
}
```

Source

Destination

```
{
    "name": "Data Generator To Kafka",
    "lifecycle": "persistent",
    "workers": 1,
    "assets": [
        "standard",
        "kafka"
    ],
    "operations": [
        {
            "_op": "data_generator",
            "size": 500000
        },
        {
            "_op": "kafka_sender",
            "connection": "default",
            "topic": "test1",
            "size": 25000
        }
    ]
}
```

Insert Mutations & Transforms

Outline

- ETL and Stream Processing
- Teraslice Description and Features
- Teraslice Job
- Example Problem
- Teraslice Processor

Weather Station Streaming Problem

Imagine you operate thousands of weather stations all around the world and you had customers who want real time alerts from those weather stations under specified conditions.

Each measurement from one of these weather stations looks like this:

```
{
    "station_id": "USW00003192",
    "date": "2016-04-23T00:00:00",
    "AWND": 2.4,
    "PRCP": 0,
    "TMAX": 29.4,
    "TMIN": 18.9,
}
```

Weather Station Streaming Problem

Imagine you operate thousands of weather stations all around the world and you had customers who want real time alerts from those weather stations under specified conditions.

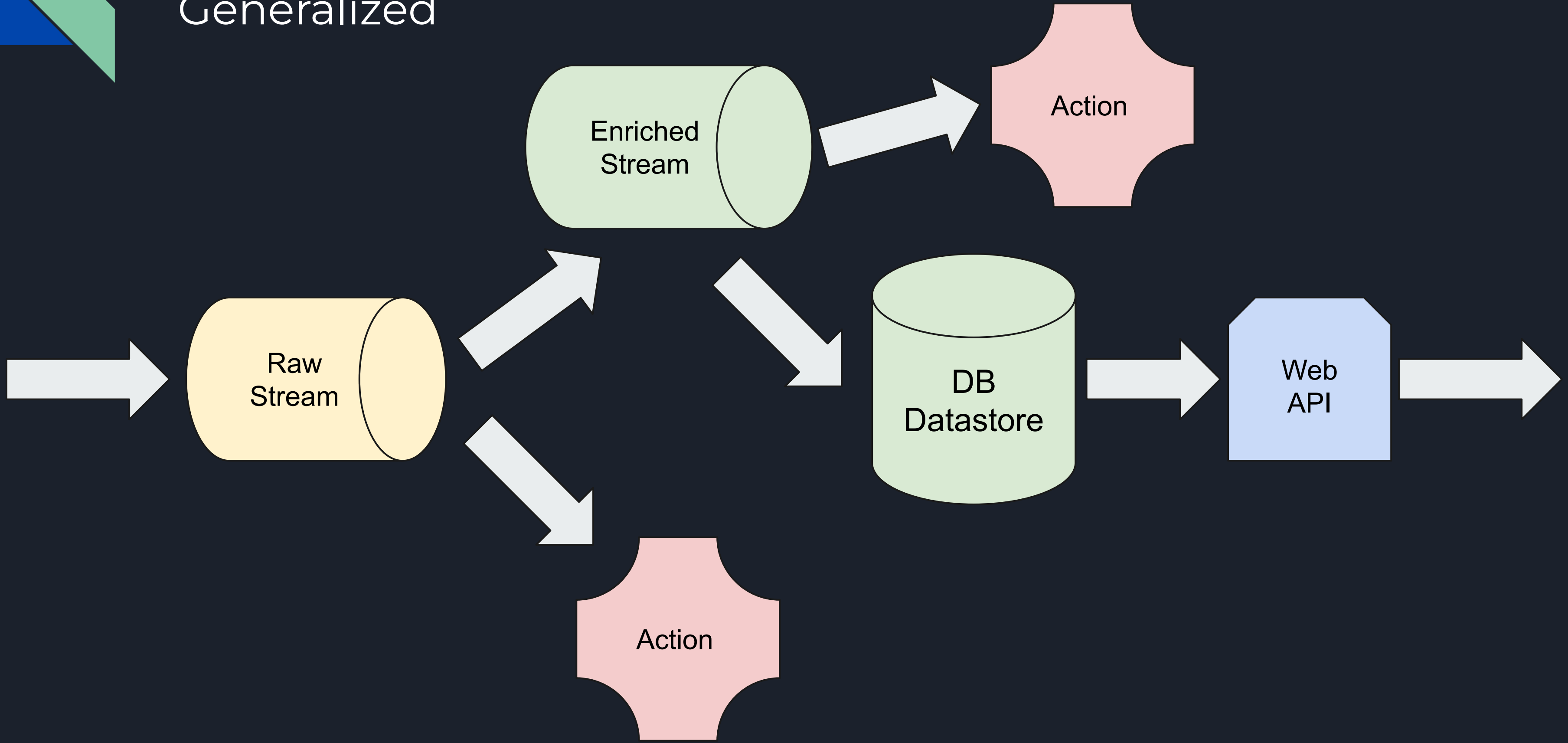
Each measurement from one of these weather stations looks like this:

Or more likely …

```
"USW00003192","2016-04-23T00:00:00”,2.4,0,29.4,18.9
```

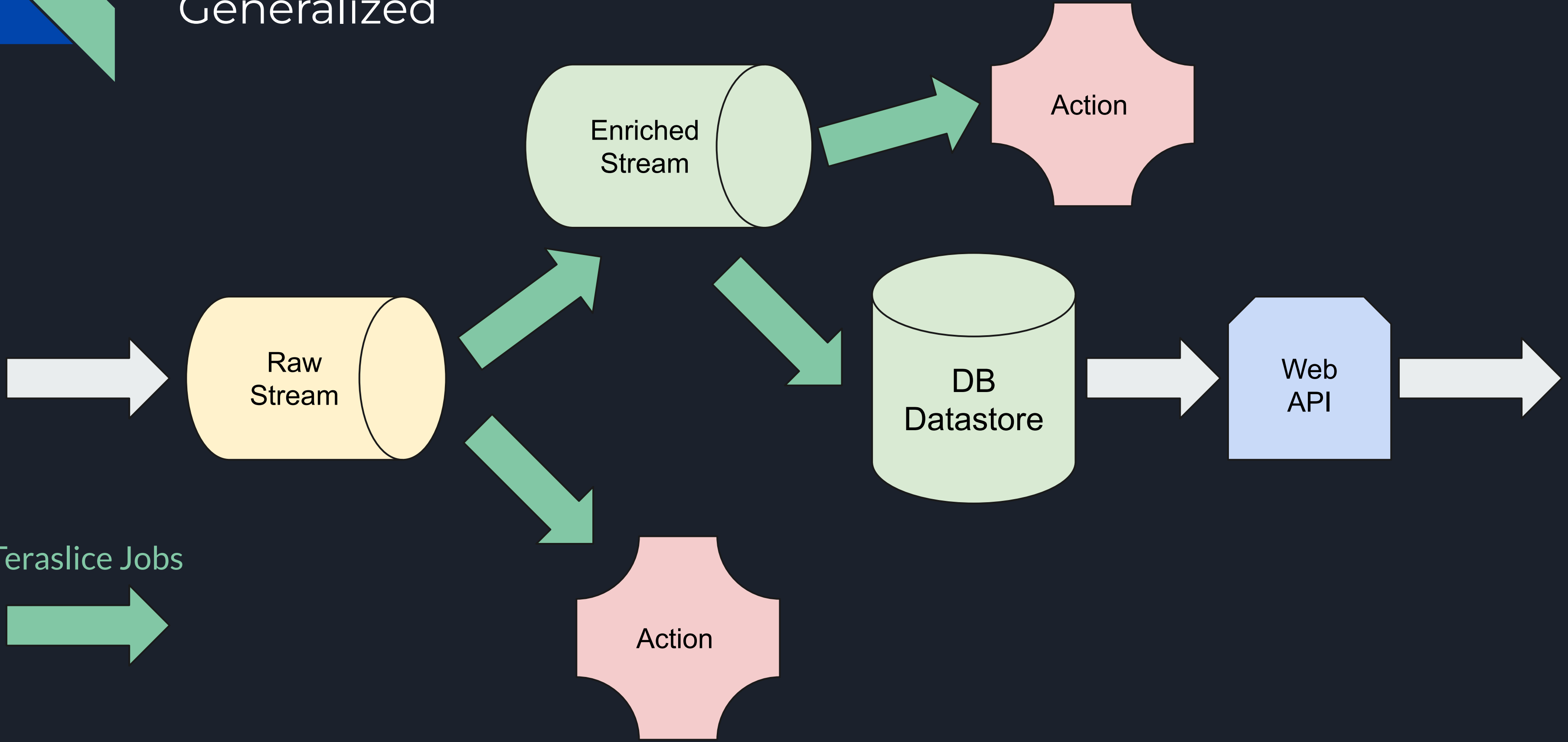
Weather Station Streaming Problem

Generalized



Weather Station Streaming Problem

Generalized



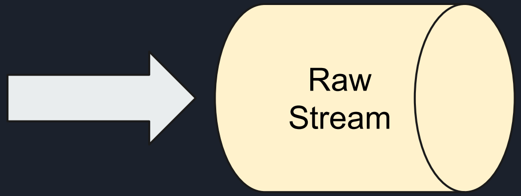
Teraslice Jobs

PAUSE ⚠️

Batch based ETL workflows are great for many use cases, but as the incoming rate increases the batch processing time gets larger. When that processing time is TOO long for your use case, that’s when you consider switching to streaming … if you can.

Weather Station Streaming Problem

Incoming data is a big array of JSON records



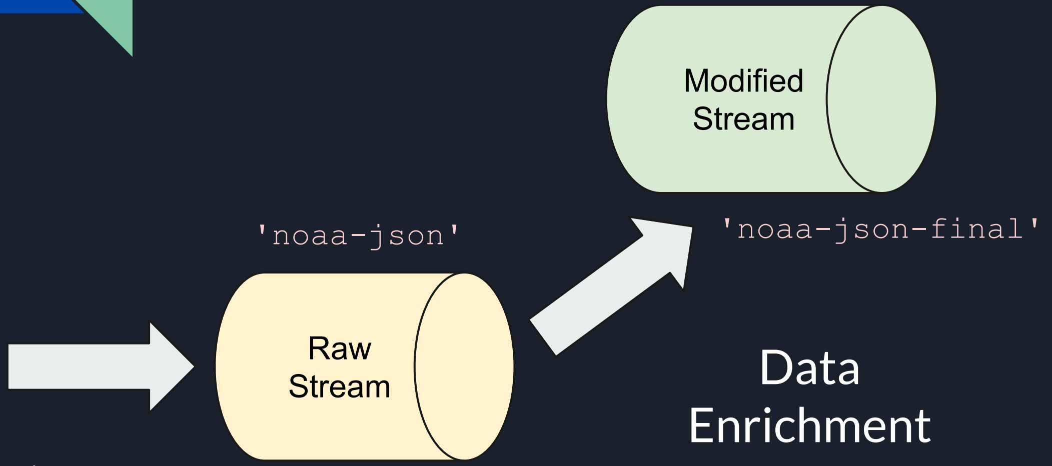
```
{
    "station_id": "USW00003192",
    "date": "2016-04-23T00:00:00",
    "AWND": 2.4,
    "PRCP": 0,
    "TMAX": 29.4,
    "TMIN": 18.9,
}
```

Weather Station Streaming Problem

Incoming data is global and needs interpretation

```
{
    "station_id": "USW00003192",
    "date": "2016-04-23T00:00:00",
    "AWND": 2.4,
    "PRCP": 0,
    "TMAX": 29.4,
    "TMIN": 18.9,
}
```

Weather Station Streaming Problem



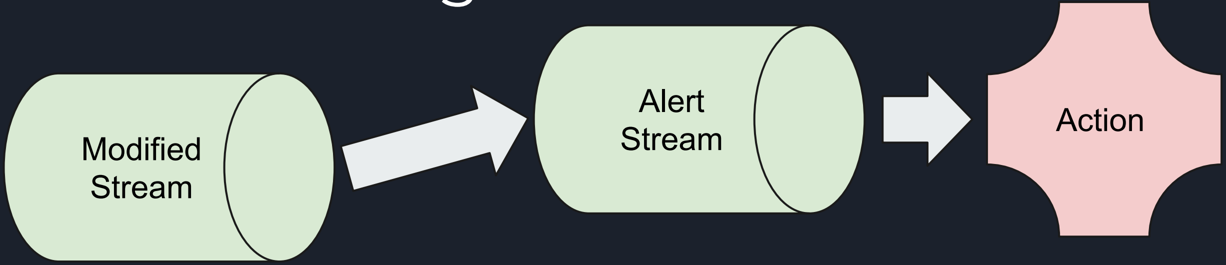
```
{
    "station_id": "USW00003192",
    "date": "2016-04-23T00:00:00",
    "AWND": 2.4,
    "PRCP": 0,
    "TMAX": 29.4,
    "TMIN": 18.9,
}
```

```
{
    "station": {
        "id": "USW00003192",
        "country_code": "US",
        "country": "United",
        "location": {
            "lat": 33.6228,
            "lon": -111.9106
        },
        "elevation": 449,
        "state_code": "AZ",
        "state": "ARIZONA",
        "name": "SCOTTSDALE MUNI AP"
    },
    "date": "2016-04-23T00:00:00",
    "AWND": 2.4,
    "PRCP": 0,
    "TMAX": 29.4,
    "TMIN": 18.9,
}
```

Say you had a farm near `USW00003192` and wanted an email, SMS or Notification if `TMIN` was approaching freezing.

Weather Station Streaming Problem
Generalized

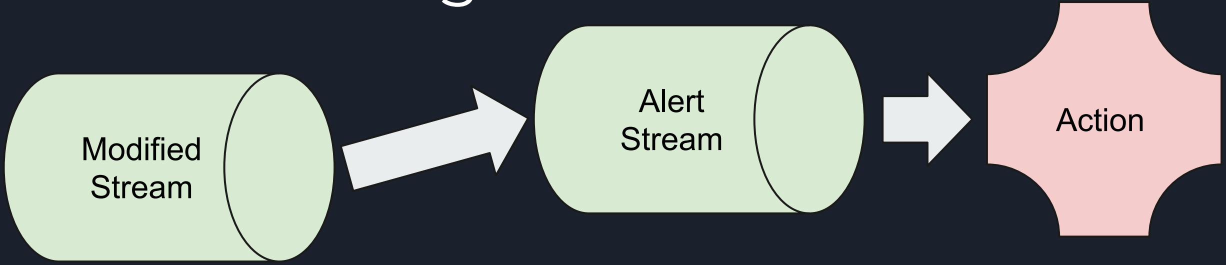
```
{
"station": {
    "id": "USW00003192",
    "country_code": "US",
    "country": "United",
    "location": {
        "lat": 33.6228,
        "lon": -111.9106
    },
    "elevation": 449,
    "state_code": "AZ",
    "state": "ARIZONA",
    "name": "SCOTTSDALE MUNI AP"
},
"date": "2016-04-23T00:00:00",
"AWND": 2.4,
"PRCP": 0,
"TMAX": 29.4,
"TMIN": 18.9,
}
```



Filter for condition and specific station and alert user

Weather Station Streaming Problem
Generalized

```
{
"station": {
    "id": "USW00003192",
    "country_code": "US",
    "country": "United",
    "location": {
        "lat": 33.6228,
        "lon": -111.9106
    },
    "elevation": 449,
    "state_code": "AZ",
    "state": "ARIZONA",
    "name": "SCOTTSDALE MUNI AP"
},
"date": "2016-04-23T00:00:00",
"AWND": 2.4,
"PRCP": 0,
"TMAX": 29.4,
"TMIN": 18.9,
}
```



Filter for condition and specific station and alert user

```
if (
    id === 'USW00003192'
    &&
    TMIN <= 1.0
)
```

Outline

- ETL and Stream Processing
- Teraslice Description and Features
- Teraslice Job
- Example Problem
- Teraslice Processor

Teraslice Asset/Processor Example

- Assets and Processors
  - A processor is the unit of code that manipulates the data
  - An asset is a collection of related processors
- Example Source
  - https://github.com/godber/presentations/tree/main/phxjs-godber-teraslice-2023/weather-alert-asset

Thank you!

Teraslice

Stream Processing with NodeJS

https://github.com/terascope/teraslice

https://github.com/godber/presentations

Austin Godber
@godber@az.social
May 17th, 2023