Beam for Real-time Manufacturing Data Analysis

Jeswanth Yadagani - Oden Technologies
Jeswanth Yadagani
ML Engineer
Oden Technologies
Outline of this talk

- Oden and its customers
- Second bullet goes here
- Slide No. 10
- Fourth point is very important
- Fifth one is a secret! Don’t share
- blah blah blah........
- This is an application of the third point
- The animations and flow charts
- Beam it is!
- Classic THANK YOU slide
Outline of this talk

“ns=2;s=MQTT.Line3BlendingPLC.Line3 Blending PLC_Device1_01MixerAmp(amps):
{"value": 316.299988, "timestamp": 1627765372, "metric": "7f6f8571-417c-5d48-b6ab-8f74eb974fe7", "uuid": "b24fe84e-1423-4c4a-9b99-7d6fb6742781", "route": "/metric_alpha"}”

“Bad process conditions observed on Line 2”
Oden Technologies
Oden’s Customers

Medium to Large scale industries manufacturing wires & cables, pipes, chemical resins, paper and pulp
Our Product - Live Data and Alerting

- Metrics Streaming at a second resolution
- Real-time status indication of the process
- Real-time intelligent alerts
Our Product - Historic Analytics

- Centralize and compare metrics for root cause analysis.
- Custom dashboards indicating performance indexes of interest for comparison.
How do we Ingest Data into our Platform?
Data Ingestion Architecture

Sensor Readings → OPC-UA Server → Operator Logs

Google Cloud
Data Ingestion Architecture

Sensor Readings → OPC-UA Server → Cloud IoT Core

Operator Logs
Data Ingestion Architecture

Sensor Readings → OPC-UA Server → Cloud IoT Core → Streaming Calculations
→ Pub Sub
→ Dataflow
Data Ingestion Architecture

Sensor Readings → OPC-UA Server → Cloud IoT Core

Streaming Calculations:
- Pub Sub
- Dataflow

Storage:
- Cloud Big Table
- Heroic Timeseries
Data Ingestion Architecture

- Sensor Readings
- Operator Logs
- Data Maps
- Metadata
- Process Targets

OPC-UA Server -> Cloud IoT Core

Streaming Calculations
- Pub Sub
- Dataflow

Storage
- Cloud Big Table
- Heroic Timeseries

API
Data Ingestion Architecture

Sensor Readings -> OPC-UA Server -> Cloud IoT Core

Operator Logs

Data Maps

Metadata

Process Targets

Streaming Calculations

Pub Sub

Dataflow

Storage

Cloud Big Table

Heroic Timeseries

Cloud SQL

Postgres

API
Customer Questions/Requirements
Questions! Questions! Questions!

What is today’s production?
Questions! Questions! Questions!

What is today's production?

Can I run at higher rate?
Questions! Questions! Questions!

What is today's production?

Schedule Maintenance time now?

Can I run at higher rate?
Questions! Questions! Questions!

What is today’s production?

Schedule Maintenance time now?

How to reduce waste?

Can I run at higher rate?
Questions! Questions! Questions!

- Is the quality of product good?
- What is today’s production?
- How to reduce waste?
- Schedule Maintenance time now?
- Can I run at higher rate?
Questions! Questions! Questions!

- Is the quality of product good?
- What is today’s production?
- Schedule Maintenance time now?
- How to reduce waste?
- Can I run at higher rate?
- At Max possible efficiency?
Questions! Questions! Questions!

Know all questions of interest from customers.

Equip customers with tools to find answers on their own.
Streaming Metric Calculations with Apache Beam
THE LOOPS

BEAM! BEAM! BEAM!

- Streaming Calculations
- Pub Sub
- Dataflow
- Stateless Calculation
- Windowed Calculation
- Stateful Calculation
THE LOOPS

Cloud IoT Core

Raw Events
THE LOOPS

Not in this talk!
THE LOOPS

Cloud IoT Core → Raw Events → Acquisition → Metrics
Operations on the data that is streaming in the present moment
Stateless Calculation

Raw streams: diameter-x and diameter-y
Stateless Calculation

Raw streams: diameter-x and diameter-y

User Requirement: What is the diameter of cable that is being produced?
Stateless Calculation

Raw streams: diameter-x and diameter-y

User Requirement: What is the diameter of cable that is being produced?

diameter = Stateless(diameter-x, 
diameter-y, 
function = mean)
Stateless Calculation

Raw streams: diameter-x and diameter-y

User Requirement: What is the diameter of cable that is being produced?

diameter = Stateless(diameter-x, diameter-y, function = mean)
THE LOOPS

Cloud IoT Core → Raw Events → Acquisition → Metrics

Stateless Calculation
Operations on the data that is gathered into a window of predefined size
Windowed Calculation

Raw streams: diameter-x and diameter-y
Specifications: diameter-USL and diameter-LSL
Windowed Calculation

Raw streams: diameter-x and diameter-y
Specifications: diameter-USL and diameter-LSL

User Requirement: Is diameter under specification limits?
Windowed Calculation

Raw streams: diameter-x and diameter-y
Specifications: diameter-USL and diameter-LSL

User Requirement: Is diameter under specification limits?

diameter_avg, diameter_sd = Window(diameter,
functions = [avg, sd],
size = 60s,
slide = 5s)
Raw streams: diameter-x and diameter-y
Specifications: diameter-USL and diameter-LSL

User Requirement: Is diameter under specification limits?

\[
diameter_{avg}, diameter_{sd} = \text{Window}(diameter, \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad
Raw streams: diameter-x and diameter-y
Specifications: diameter-USL and diameter-LSL

User Requirement: Is diameter under specification limits?

diameter_avg, diameter_sd = Window(diameter, 
functions = [avg, sd], 
size = 60s, 
slide = 5s)

limit_bool = diameter-USL > diameter_avg > diameter-LSL

stability_bool = diameter_avg - 3*diameter_sd > diameter > 
diameter_avg + 3*diameter_sd

violation_bool = ~(limit_bool AND stability_bool)

This stream tells us if diameter is off specs
THE LOOPS

Cloud IoT Core → Raw Events → Acquisition → Metrics

Windowed Calculation

Stateless Calculation
Operations that leverage advantages of maintaining memory and are custom reset upto certain events of interest
Raw streams: diameter-x, diameter-y and linespeed
Specifications: diameter-USL and diameter-LSL

User Requirement: How much of the cable is off specs in this batch?
Stateful Calculation

Raw streams: diameter-x, diameter-y and linespeed
Specifications: diameter-USL and diameter-LSL

User Requirement: How much of the cable is off specs in this batch?

violation_duration = Stateful(violation_bool,
  function = time_aggregate,
  reset_on = new batch)
Stateful Calculation

Raw streams: diameter-x, diameter-y and linespeed
Specifications: diameter-USL and diameter-LSL

User Requirement: How much of the cable is off specs in this batch?

\[
\text{violation\_duration} = \text{Stateful}(
\text{violation\_bool},
\text{function} = \text{time\_aggregate},
\text{reset\_on} = \text{new \ batch})
\]

\[
\text{off\_spec\_cable} = \text{Stateless}(
\text{linespeed},
\text{violation\_duration\_duration},
\text{function} = \text{multiply})
\]
Algebra of streaming analysis: LOOPS allow us to create arbitrary combinations of these three calculations to address any customer requirement.
Rhino JS interpreter embedded within beam ParDos
Alerting on Bad Process Conditions
Cable Manufacturing

- Copper is pulled from an in-spool into an extruder.
- Plastic is melted over the copper to make wire.
- Wire is cooled.
- Wire is pulled into an out-spool.
Cable Manufacturing

- Copper is pulled from an in-spool into an extruder.
- Plastic is melted over the copper to make wire.
- Wire is cooled.
- Wire is pulled into an out-spool.

- A laser measures the diameter of the wire to monitor its closeness to spec.
Cable Manufacturing

- Copper is pulled from an in-spool into an extruder.
- Plastic is melted over the copper to make wire.
- Wire is cooled.
- Wire is pulled into an out-spool.

- A laser measures the diameter of the wire to monitor its closeness to spec.

![Diagram showing the manufacturing process with independent variables including Line Speed, Vacuum Pressure, Melt Temp, and Outer Diameter. Dependent Variables include Cable Manufacturing stages.]
Process Alerts with streaming pipeline

Decision trees are trained on historic data to extract rules corresponding to bad and good production.
Process Alerts with streaming pipeline

Decision trees are trained on historic data to extract rules corresponding to bad and good production.

Combination:
\[
\text{mean(Melt Temp)} \geq 800\text{F} \\
\text{AND min(Vacuum Pressure)} < 4\text{Pa} \\
\text{AND Time since run start} > 15 \text{ min}
\]

Of the 84 segments 100.0% lead to bad production quality
Process Alerts with streaming pipeline

Decision trees are trained on historic data to extract rules corresponding to bad and good production.

\[
S1 = \text{Window(melt temperature, function = mean, size = 300s, slide = 60s)}
\]

Combination:
- \(\text{mean(Melt Temp)} \geq 800\text{F}\)
- \(\text{AND min(Vacuum Pressure)} < 4\text{Pa}\)
- \(\text{AND Time since run start} > 15\text{ min}\)

Of the 84 segments 100.0% lead to bad production quality.
Process Alerts with streaming pipeline

Decision trees are trained on historic data to extract rules corresponding to bad and good production.

S1 = Window(melt_temperature, function = mean, size = 300s, slide = 60s)

S2 = Window(vacuum_pressure, function = min, size = 300s, slide = 60s)

Combination:
- mean(Melt Temp) $\geq$ 800F
- AND min(Vacuum Pressure) < 4Pa
- AND Time since run start > 15 min

Of the 84 segments 100.0% lead to bad production quality
Process Alerts with streaming pipeline

Decision trees are trained on historic data to extract rules corresponding to bad and good production.

S1 = Window(melt_temperature, function = mean, size = 300s, slide = 60s)

S2 = Window(vacuum_pressure, function = min, size = 300s, slide = 60s)

S3 = Stateful(line_speed, function = time_aggregate, reset_on = new run)

Combination:
mean(Melt Temp) >= 800F
AND min(Vacuum Pressure) < 4Pa
AND Time since run start > 15 min

Of the 84 segments 100.0% lead to bad production quality.
Process Alerts with streaming pipeline

Decision trees are trained on historic data to extract rules corresponding to bad and good production.

Alerts are configured to notify operators on factory floor when process goes into bad production conditions.

\[
\text{S1} = \text{Window(melt\_temperature, function = mean, size = 300s, slide = 60s)}
\]

\[
\text{S2} = \text{Window(vacuum\_pressure, function = min, size = 300s, slide = 60s)}
\]

\[
\text{S3} = \text{Stateful(line\_speed, function = time\_aggregate, reset\_on = new run)}
\]

Combination:
- \(\text{mean(Melt Temp)} \geq 800\text{F}\)
- \(\text{min(Vacuum Pressure)} < 4\text{Pa}\)
- \(\text{Time since run start} > 15\text{ min}\)

Of the 84 segments, 100.0% lead to bad production quality.

\[
\text{Target Stream} = \text{Stateless(S1, S2, S3)}
\]
Alert!!!

Oden Alerts <alerts@oden.io>

to me

6:09 PM (1 hour ago)

Outer Diameter violation on Line 2

As of 6:09pm EDT, Outer Diameter on Line 2 is in bad process conditions for atleast 5 minutes

View line

Snooze this alert for: 30m 2h 8h 24h

Powered by Oden Technologies

Is this alert useful? Let us know!
That concludes the journey
Raw data from factory →
Alerting on meaningful process conditions
What we Love!
What we Love!
What we Love!

- Compatible with dataflow on GCP
What we Love!

- Compatible with dataflow on GCP
- Watermarks, windows, triggers, State API
What we Love!

- Compatible with dataflow on GCP
- Watermarks, windows, triggers, State API

Workarounds!
What we Love!

- Compatible with dataflow on GCP
- Watermarks, windows, triggers, State API

Workarounds!

- Custom watermark function using state API for per key functionality to deal with late data.
What we Love!

- Compatible with dataflow on GCP
- Watermarks, windows, triggers, State API

Workarounds!

- Custom watermark function using state API for per key functionality to deal with late data.
- Clogging of pipeline due to late data as a result of the LOOPS is addressed in another talk by Devon.
Custom watermark function using state API for per key functionality to deal with late data.

Clogging of pipeline due to late data as a result of the LOOPS is addressed in another talk by Devon.

What we Love!

- Compatible with dataflow on GCP
- Watermarks, windows, trigger, State API

Workarounds!

- Custom watermark function using state API for per key functionality to deal with late data.
- Clogging of pipeline due to late data as a result of the LOOPS is addressed in another talk by Devon.

Late Data Recoveries with Batch-Mode
Aug 6th 2pm EST

Devon Peticolas
Principal Engineer
Alert: End of Stream found!

We are hiring!
https://oden.io/careers/