OPENTELEMETRY METRICS 101

Reese Lee, New Relic
Reese Lee
Developer Relations Engineer
New Relic

- Previously Technical Support
- OpenTelemetry End User WG
  - Adoption and implementation
  - Feedback loop to improve the project
- Malaysia → Pacific Northwest
- Summited Mt. Hood (11,249′)
- Visited 14 countries
AGENDA

01 METRICS OVERVIEW
02 OPENTELEMETRY OVERVIEW
03 METRICS DIP
04 WHAT'S NEXT?
1. WHAT IS A METRIC?
2. WHY ARE METRICS USEFUL?
WHAT IS A METRIC?

A metric is a measurement about a service captured at runtime. Metrics represent aggregations of multiple measurements, and can be used to identify trends.

- Throughput
- Response time
- Error rate
- CPU utilization
- Number of active users
- Total processed orders
- Total processed orders of a specific item
WHY ARE METRICS USEFUL?

**DATA VOLUME REDUCTION**
Reducing the volume of data

**PERFORMANCE**
Monitoring your system

**ALERTS**
Alerting on breached SLOs

**VISUALIZATION**
Powering graphs, charts, and dashboards
OPENTELEMETRY OVERVIEW

1. WHAT IS OPENTELEMETRY?
2. WHY OPENTELEMETRY FOR METRICS?
WHAT IS OPENTELEMETRY?

OpenTelemetry is...

- An observability framework built on an open standard
- The merging of OpenCensus and OpenTracing in 2019
- 2nd most active CNCF project in terms of contributions (after Kubernetes)
- Aims to standardize instrumentation and telemetry generation, collection, and transmission
WHAT IS OPENTELEMETRY?

OpenTelemetry provides a set of APIs and SDKs, tools and components (such as the Collector), instrumentation libraries, semantic conventions, and a protocol (called OTLP).

- Java
- .NET
- Python
- Ruby
- ... and more

ONE STANDARDIZED SET OF TOOLS
Why OpenTelemetry for Metrics?

- Exemplars
- Enrich metrics attributes via Baggage and Context

Original goal of OpenTelemetry (OpenCensus + OpenTracing)

Minimum goal: Prometheus and Statsd

Freedom from vendor lock-in!
03
METRICS
DIP
METRICS DIP

1. SESSION SCOPE
2. METRICS IN OPENTELEMETRY
3. ARCHITECTURE
4. METRIC INSTRUMENTS, TYPES, AND USE CASES
   a. What is an instrument?
   b. What instruments does OpenTelemetry provide?
   c. Why is instrument selection important?
   d. How do I choose an instrument?
SESSION SCOPE

10% THIS SESSION
High-level overview of metrics concepts

40% DEEP DIVE
There is so much more we could get into!

50% EVERYTHING ELSE!?
And perhaps even beyond that... (e.g., implementation)
METRICS IN OPENTELEMETRY

**API**
Used to instrument code

**SDK**
Used to implement the API

- **Meter Provider**
- **Meters**
- **Instruments**
- **Scopes**
- **Measurements**: a value and a set of attributes
METRICS IN OPENTELEMETRY

**AGGREGATION**
The process of combining multiple measurements into a single point

**TEMPORALITY**
Related to whether the reported values of additive quantities include previous measurements

**MONOTONICITY**
Related to whether the value is always increasing, or always increasing and decreasing at the same time

**DIMENSION**
An attribute associated with a metric, can be used to filter and aggregate data

**CARDINALITY**
How many unique dimensions are associated with a metric
ARCHITECTURE

Meter provider

Meter

Instrument

Measurement

Measurement

Measurement

Metric reader

Metric exporter

Data analytics tool

(no-op implementation)
# Metric Instruments, Types, and Use Cases

**What is an instrument?**

Instruments report measurements and have the following fields:

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instrument name</td>
</tr>
<tr>
<td>Kind</td>
</tr>
<tr>
<td>Measure of unit (optional)</td>
</tr>
<tr>
<td>Description (optional)</td>
</tr>
</tbody>
</table>
### What Instruments Does Opentelemetry Provide?

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Aggregate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter</td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="Green" /></td>
<td>Sum</td>
</tr>
<tr>
<td>Up/down counter</td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="White" /></td>
<td>Sum</td>
</tr>
<tr>
<td>Async counter</td>
<td><img src="#" alt="White" /></td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="Green" /></td>
<td>Sum</td>
</tr>
<tr>
<td>Async up/down counter</td>
<td><img src="#" alt="White" /></td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="White" /></td>
<td>Sum</td>
</tr>
<tr>
<td>Histogram</td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="White" /></td>
<td><img src="#" alt="White" /></td>
<td>Histogram</td>
</tr>
<tr>
<td>Gauge</td>
<td><img src="#" alt="White" /></td>
<td><img src="#" alt="White" /></td>
<td><img src="#" alt="White" /></td>
<td>Last value</td>
</tr>
</tbody>
</table>
WHY IS INSTRUMENT SELECTION IMPORTANT?

Default aggregation reflects the intended use of the measurements.

Instrument type
- measurements are aggregated
- the type of metric that is exported
- impacts the way you can query and analyze it.
HOW DO I CHOOSE AN INSTRUMENT?

**Analysis**
How do you want to analyze the data?

**Sync or async**
Do you need the measurement synchronously, or can it be reported on a set interval?

**Monotonicity**
Are the values monotonic?
## Counter

<table>
<thead>
<tr>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Default aggregation</th>
<th>Example usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>Sum</td>
<td>Number of bytes sent, total orders processed, total cart adds, total cart add failures, total checkouts, total checkout failures</td>
</tr>
</tbody>
</table>

**Use when...**
- you want to count things and compute the rate at which things happen
- the sum of the things is more meaningful than the individual values
### UP/DOWN COUNTER

<table>
<thead>
<tr>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Default aggregation</th>
<th>Example usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅</td>
<td>✅</td>
<td>❌</td>
<td>Sum</td>
<td>Number of open connections, number of active users, queue size, memory in use</td>
</tr>
</tbody>
</table>

Use when...
- you want to process positive and negative increments
### Async Counter

<table>
<thead>
<tr>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Default aggregation</th>
<th>Example usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>❌</td>
<td>✅</td>
<td>✅</td>
<td>Sum</td>
<td>CPU time, cache hits and misses, total network bytes transferred</td>
</tr>
</tbody>
</table>

#### Use when...
- you need a sum of your measurements, but they may be too expensive to report synchronously, or it is more appropriate to record on set intervals.
## Async Up/Down Counter

<table>
<thead>
<tr>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Default aggregation</th>
<th>Example usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>❌</td>
<td>✅</td>
<td>❌</td>
<td>Sum</td>
<td>Memory utilization, process heap size, number of active shards, changes in the number of active users</td>
</tr>
</tbody>
</table>

Use when...
- you need a non-monotonic additive counter to report on set intervals

![Graph showing number of users over time](image)
### Histogram

<table>
<thead>
<tr>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Default aggregation</th>
<th>Example usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅</td>
<td>❌</td>
<td>❌</td>
<td>Explicit bucket histogram</td>
<td>HTTP server response times, client duration, request rate</td>
</tr>
</tbody>
</table>

Use when...
- you want to analyze the distribution of measurements to identify trends
- you want to calculate the min, max, and average response time
# GAUGE

<table>
<thead>
<tr>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Default aggregation</th>
<th>Example usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>Last value</td>
<td>CPU utilization, temperature of hardware at this point in time, average memory consumption</td>
</tr>
</tbody>
</table>

Use when...

- you want to report data that’s not useful to aggregate across dimensions and you have access to measurements asynchronously
- you want finer-grain control of when a non-additive measurement is made, particularly when its purpose is a distribution
What is an instrument?

Instruments report measurements and have the following fields:

- **Instrument name**: telescopes_sold
- **Kind**: counter
- **Measure of unit (optional)**: telescope
- **Description (optional)**: “Total telescopes sold”
METRICS IN OPENTELEMETRY

VIEW

- Allows you to customize the metrics output by the SDK:
  - Process or ignore instruments
  - Override aggregation strategy
  - Attributes
WHAT'S NEXT?
WHAT'S NEXT?

1. RECAP
2. WHAT TO EXPLORE NEXT?
3. CREDITS, REFERENCES & CONTACT INFO
RECAP

1. What a metric is, and why they’re useful for observability
2. What OpenTelemetry is, and the utility and customization options it provides in metric generation and collection
3. Metric concepts as they apply in OpenTelemetry
4. OpenTelemetry metric instruments, and how to choose one
<table>
<thead>
<tr>
<th></th>
<th>Synchronous</th>
<th>Additive</th>
<th>Monotonic</th>
<th>Default aggregation</th>
<th>Example usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>Sum</td>
<td>Number of bytes sent, total orders processed</td>
</tr>
<tr>
<td>Up down counter</td>
<td>✅</td>
<td>✅</td>
<td>❌</td>
<td>Sum</td>
<td>Number of open connections, number of active users</td>
</tr>
<tr>
<td>Histogram</td>
<td>✅</td>
<td>❌</td>
<td>❌</td>
<td>Histogram</td>
<td>Response times, search results latency</td>
</tr>
<tr>
<td>Async counter</td>
<td>❌</td>
<td>✅</td>
<td>✅</td>
<td>Sum</td>
<td>Cache hits and misses, CPU time</td>
</tr>
<tr>
<td>Async up down</td>
<td>❌</td>
<td>✅</td>
<td>❌</td>
<td>Sum</td>
<td>Memory utilization, number of active users</td>
</tr>
<tr>
<td>counter</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>Last value</td>
<td>CPU utilization, hardware temperature</td>
</tr>
</tbody>
</table>
WHAT TO EXPLORE NEXT?

- Instrumentation and implementation - try it out yourself!
- Views API
- Data point types
- Adding metric attributes (or dimensions)
- Push- vs pull-based exporting
- Application runtime metrics
- OpenTelemetry collector metrics processors
- Infrastructure metrics
- … and so much more!
CREDITS & REFERENCES

CREDITS

- Jack Berg, New Relic
- Vijay Samuel, eBay

REFERENCES

- Exponential Histograms: Better Data, Zero Configuration – Jack Berg
- Cloud-Native Observability with OpenTelemetry – Alex Boten
- OpenTelemetry docs
- OpenTelemetry Metrics Primer for Java Developers – Asaf Mesika
THANK YOU!

@reesesbytes
Reese Lee on CNCF Slack

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon and infographics & images by Freepik

Booth #212!