Metrics

Diba Salehiyeh , .Net Team

May 2023

گروه نرم افزاری پارت
Part Software Group

Contents

- Monitoring
- Metrics
- Metrics Type
- Metrics Data Collectors
- Metrics Data Visualization
- Summary

Monitoring

- Observability
- Monitoring
- APM

گروه نرم افزاری پارت
Part Software Group

What is observability?

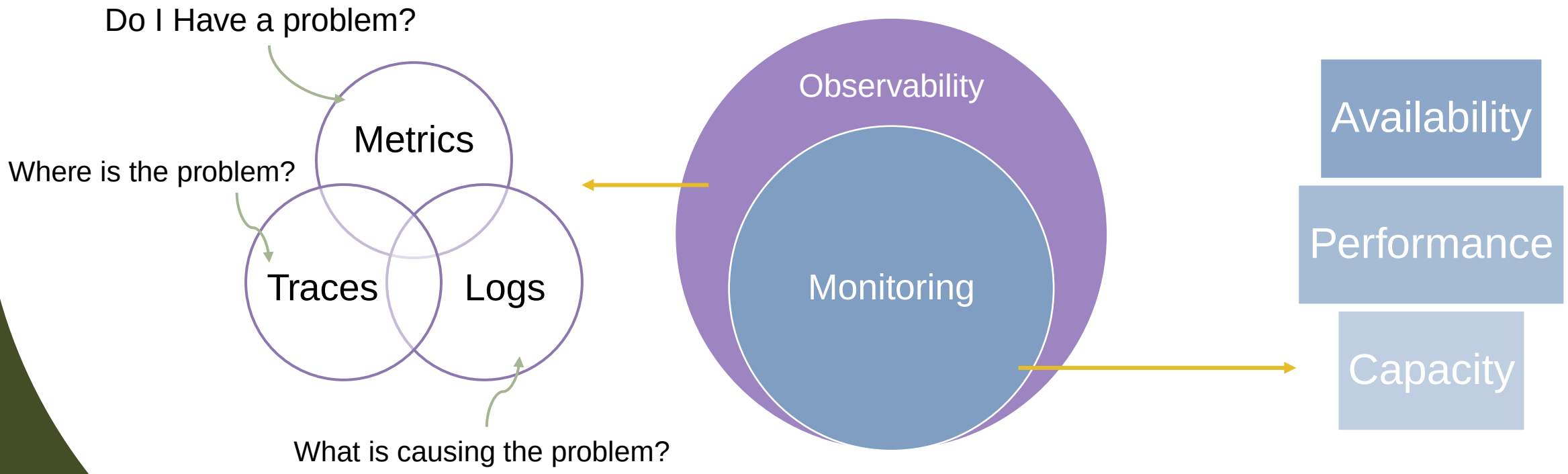
The goal of observability is to enable teams to understand how a system behaves over time.

Enables teams to identify potential issues before they become major problems.

What is monitoring?

- The first tenet of an observable system is monitoring itself.
- Monitoring involves collecting data from various sources within a system to track its performance and identify issues.
- Monitoring is continuous observing of the system for observing CPU usage, memory, routers, switches, availability and performance of systems in network

Observability vs Monitoring?



گروه نرم افزاری پارت
Part Software Group

Metrics

- Visualized data with pictorial graphs
- Customizable alerts

Traces

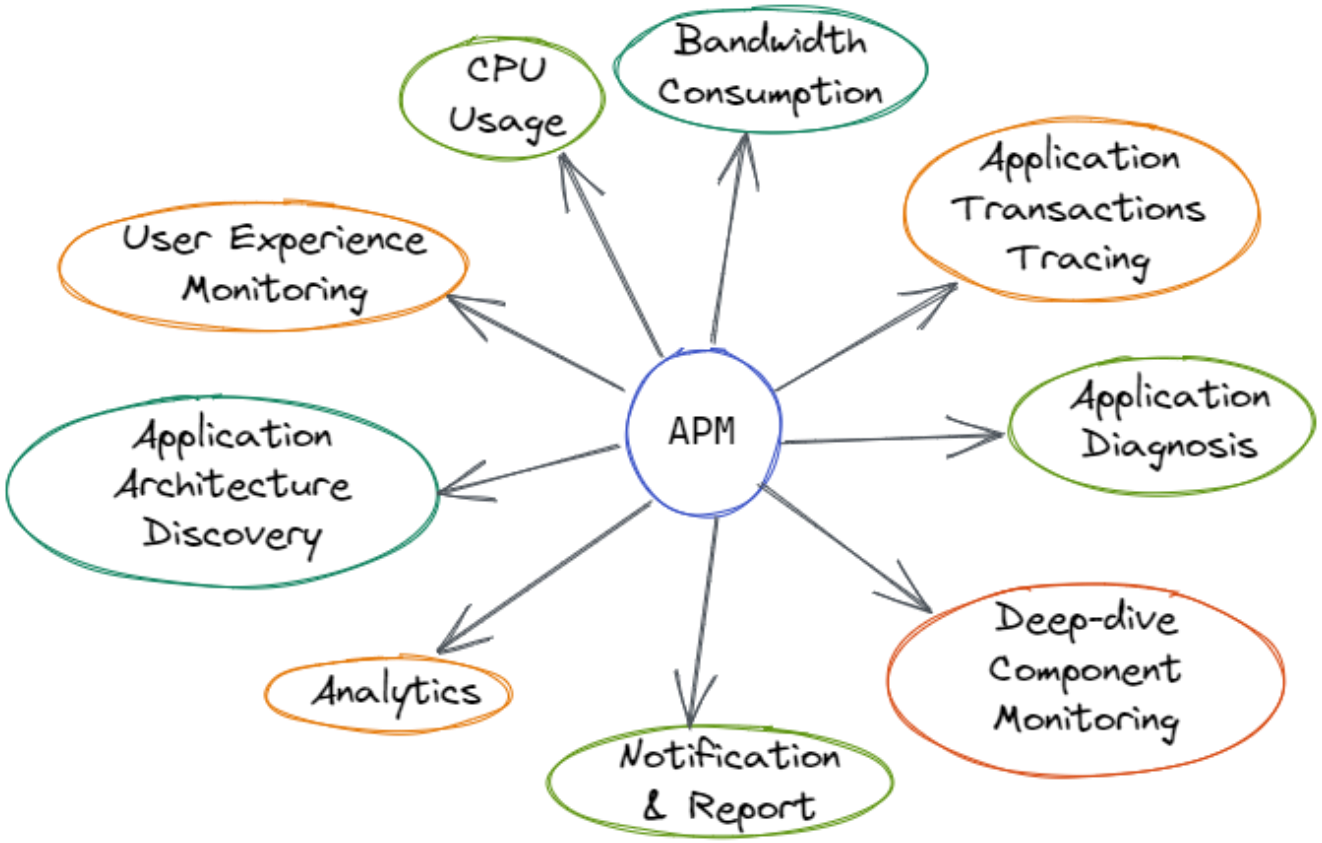
- Tracing program flows and data sequence of
- Latency tracking and Root cause analysis

Logs

- Complete details on the source, time, and place of occurrence of errors
- Huge, clumsy data pools
- Analyzing log files is not easy

Application Performance Monitoring

A collection of tools and processes for *tracking* the performance, reliability and user experience of an application



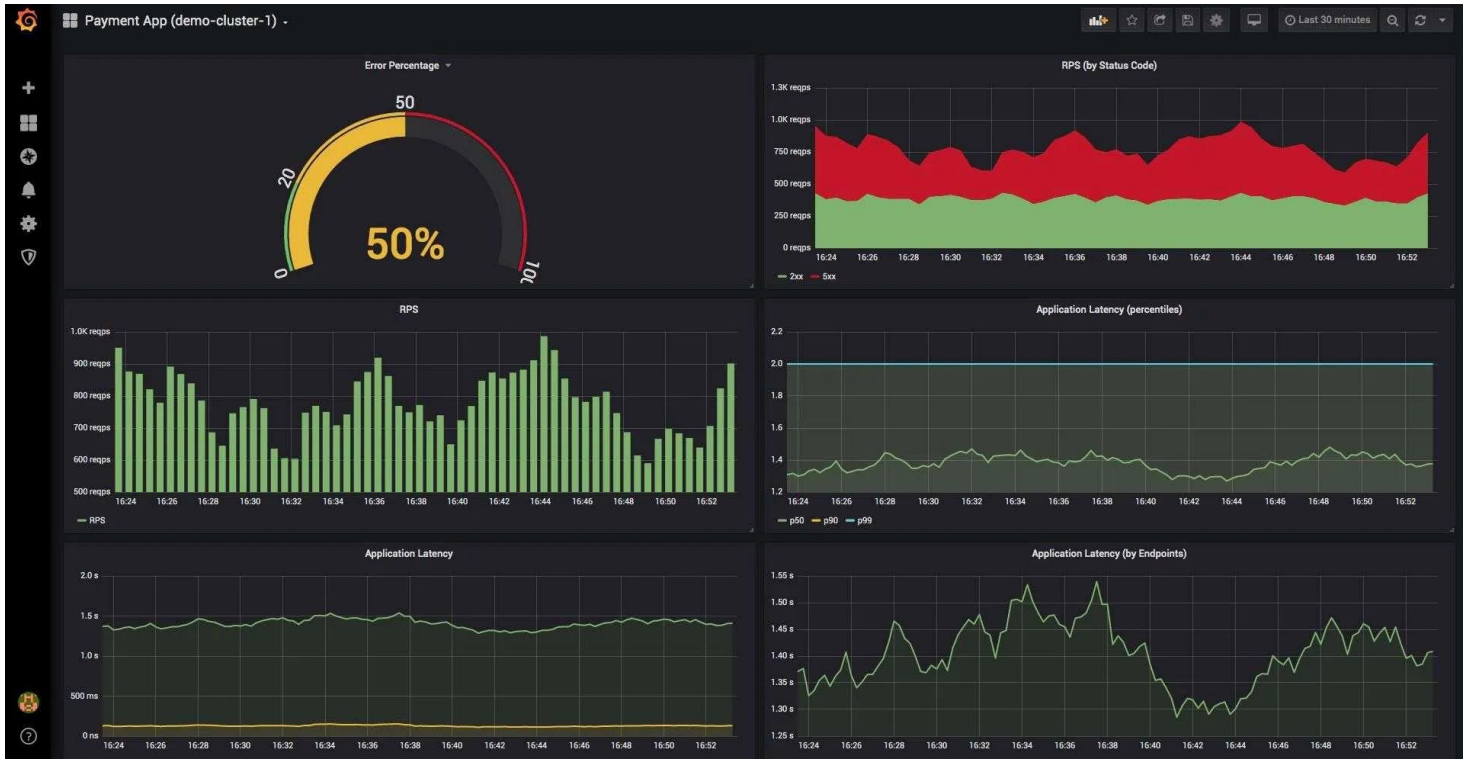
Metrics

What is metrics?

- Quantitative data collected to evaluate various aspects of software development processes and products
- **Goals**:
  - Evaluation of product reliability, maintainability and usability
  - Tracking the progress of software development projects and identifying potential problems or delays

What is metrics? (cont.)

- CPU metrics
- Disk metrics
- Memory metrics
- Average response time
- Error rates
- Request rate
- Service failures and restarts

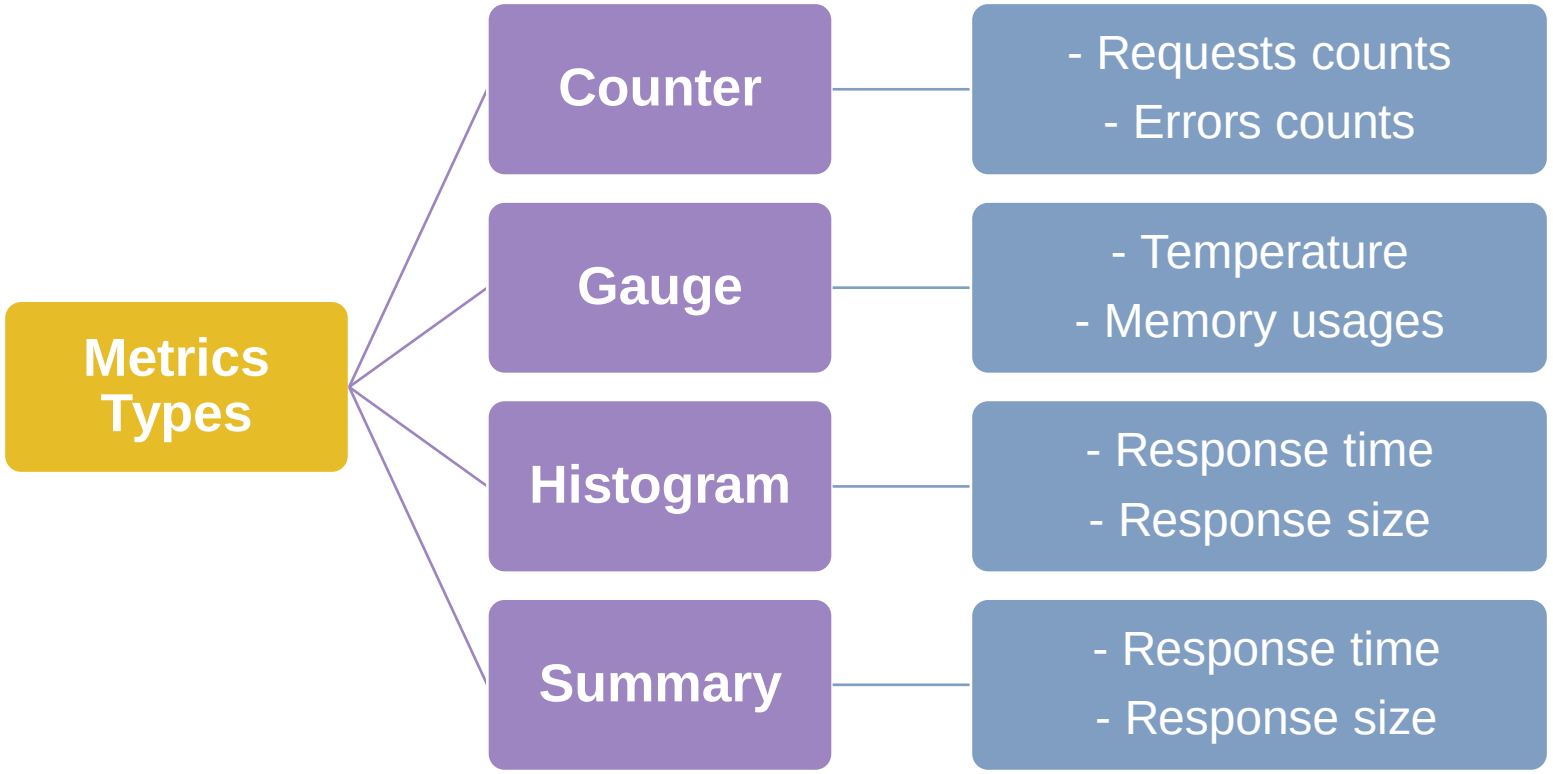


Metrics Types

- Types
- Output formats
- Advantages & disadvantages
- How choose good metrics

گروه نرم افزاری پارت
Part Software Group

Metrics types



Metrics types (cont.)

Counter

- Increasing counter
- Usages : - Requests counts

  - Completed tasks count

  - Errors count

  - ~~Running processes count~~

0

Metrics types (cont.)

Counter

- Increasing counter
- Usages : - Requests counts

  - Completed tasks count

  - Errors count

  - ~~Running processes count~~

1

Metrics types (cont.)

Counter

- Increasing counter
- Usages : - Requests counts

  - Completed tasks count

  - Errors count

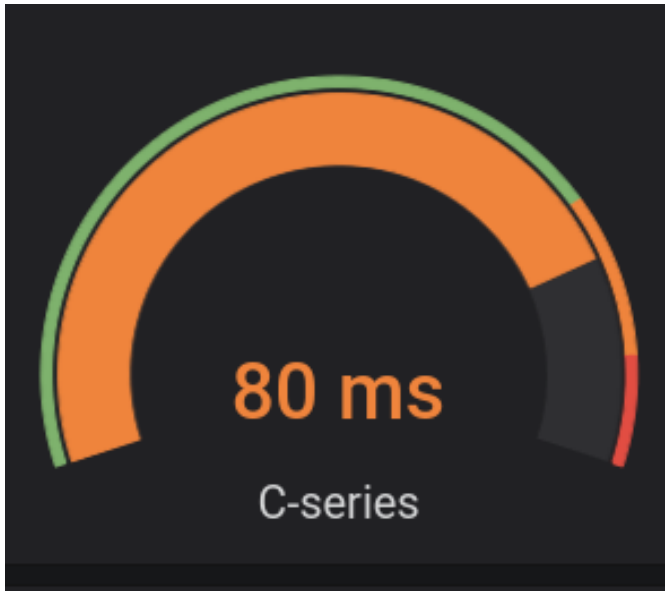
  - ~~Running processes count~~

2

Metrics types (cont.)

Gauge

- Increasing or decreasing counter
- Usages : - Temperature
  - Memory usage
  - Running processes count



Metrics types (cont.)

Counter

```
# HELP Valid_Signature_Count total valid response
# TYPE Valid_Signature_Count counter
Valid_Signature_Count 2
```

Gauge

```
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1683092373.22
```

Metrics types (cont.)

Histogram

Show the distribution of a data set by dividing it into buckets and counting the number of values that fit into each bucket

- Usages : - Response time
  - Response Size

Metrics types (cont.)

Histogram (cont.)

Bucket 0	Bucket 1	Bucket 2	Bucket 3	Bucket i	Bucket N	Bucket +N
[0,0.001]	[0,0.002]	[0,0.004]	[0,0.008]	[0, $0.001*2^{i}$]	[0, $0.001*2^{N}$]	[0, +inf]
0	0	0	0	0	0	0

Metrics types (cont.)

Histogram (cont.)

- LoginRequestDurationTime (1) = 0.001

Bucket 0	Bucket 1	Bucket 2	Bucket 3	Bucket i	Bucket N	Bucket +N
[0,0.001]	[0,0.002]	[0,0.004]	[0,0.008]	[0, $0.001*2^{i}$]	[0, $0.001*2^{N}$]	[0, +inf]
1	1	1	1	1	1	1

Metrics types (cont.)

Histogram (cont.)

- LoginRequestDurationTime (2) = 0.003

Bucket 0	Bucket 1	Bucket 2	Bucket 3	Bucket i	Bucket N	Bucket +N
[0,0.001]	[0,0.002]	[0,0.004]	[0,0.008]	[0, $0.001*2^{i}$]	[0, $0.001*2^{N}$]	[0, +inf]
1	1	2	2	2	2	2

Metrics types (cont.)

Histogram (cont.)

```
# HELP Request_Duration_HSG response time
# TYPE Request_Duration_HSG histogram
Request_Duration_HSG_sum{code="200", endpoint="api/Validator/pdfValidate"} 7.033534700000001
Request_Duration_HSG_count{code="200", endpoint="api/Validator/pdfValidate"} 12
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.001"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.002"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.004"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.008"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.016"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.032"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.064"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.128"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.256"} 0
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="0.512"} 5
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="1.024"} 11
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="2.048"} 12
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="4.096"} 12
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="8.192"} 12
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="16.384"} 12
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="32.768"} 12
Request_Duration_HSG_bucket{code="200", endpoint="api/Validator/pdfValidate", le="+Inf"} 12
```

Metrics types (cont.)

Summary

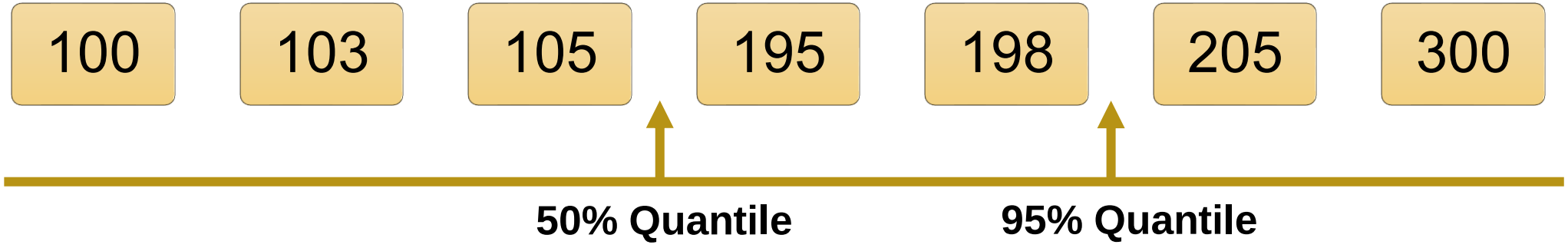
Summarizing the data distribution by dividing it into equal parts

- Usages : - Response time
  - Response Size

Metrics types (cont.)

Summary (cont.)

Data:



Quantile: distribution between 0-100%

Metrics types (cont.)

Summary (cont.)

```
# HELP http_request_duration_seconds Duration of HTTP requests
# TYPE http_request_duration_seconds summary
http_request_duration_seconds{quantile="0.5"} 1.34
http_request_duration_seconds{quantile="0.99"} 2.34
```

Metrics types (cont.)

Summary (cont.)

	Summary	Histogram
Client Performance	Calculating quantile	Increasing counter
Server Performance	Less calculation	Much of calculation

Metrics output

Metrics output format

Graphite

```
webserver.requests_per_second 42 1620292196
```

InfluxDB

```
weather,location=us-midwest temperature=82 1620292196
```

OpenMetrics

```
# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total counter
http_requests_total{method="GET",path="/api/v1/status"} 23
http_requests_total{method="POST",path="/api/v1/data"} 12
```

Elasticsearch

```
{   "timestamp": "2023-05-06T12:30:00Z",
    "server": "web-server-1",
    "cpu_usage": 78.9 }
```

گروه نرم افزاری پارت
Part Software Group

Metrics advantages and disadvantages

Advantages	Disadvantages
Structured data	Metrics restarts after restarting service (could be good and also bad)
Help targeting of project	Difficult interpretation of some metrics
Recognizing system faults	It’s may hard to find suitable metric for system monitoring
Tracing user experience journey	

Metrics based on other type of data

Extract metrics from:

- Log-based metrics
- Metrics from spans
- Event-based metrics

Choosing metrics

Good metric features:

1. How application works right now
2. What aspect of project should be considered?

How choose good metric?

- Granular; don’t combine multi values
- Use “Rate” except “Total”
- Well-understood
- Choose appropriate time span

Important metrics

Hot-based metrics	Application metrics	Network & Connectivity	Server pool
• CPU • Memory • Disk space • Processes	• Error & success rates • Service failures & restarts • Performance & latency of responses • Resource usages	• Connectivity • Error rate & packet loss • Latency • Bandwidth utilization	• Pooled resource usage • Scaling adjustment indicators • Degraded instances • Health of collections of servers

API Important metrics

Application metrics

- Request per minute
- Latency
- Failure rate

Infrastructure metrics

- API uptime
- Time to first Hello World
- Memory & CPU usage

Metrics Data Collectors and Visualization

Metrics Data Collectors and Visualization

Metrics Data Collectors

Data Collectors

- Telegraf
- Node-Exporter
- Collectd

Time Series Data Bases

- Prometheus
- InfluxDB

گروه نرم افزاری پارت
Part Software Group

Metrics Data Visualization

- Grafana
- Prometheus
- Kibana
- DataDog
- Elastic

Visualization of data in Grafana

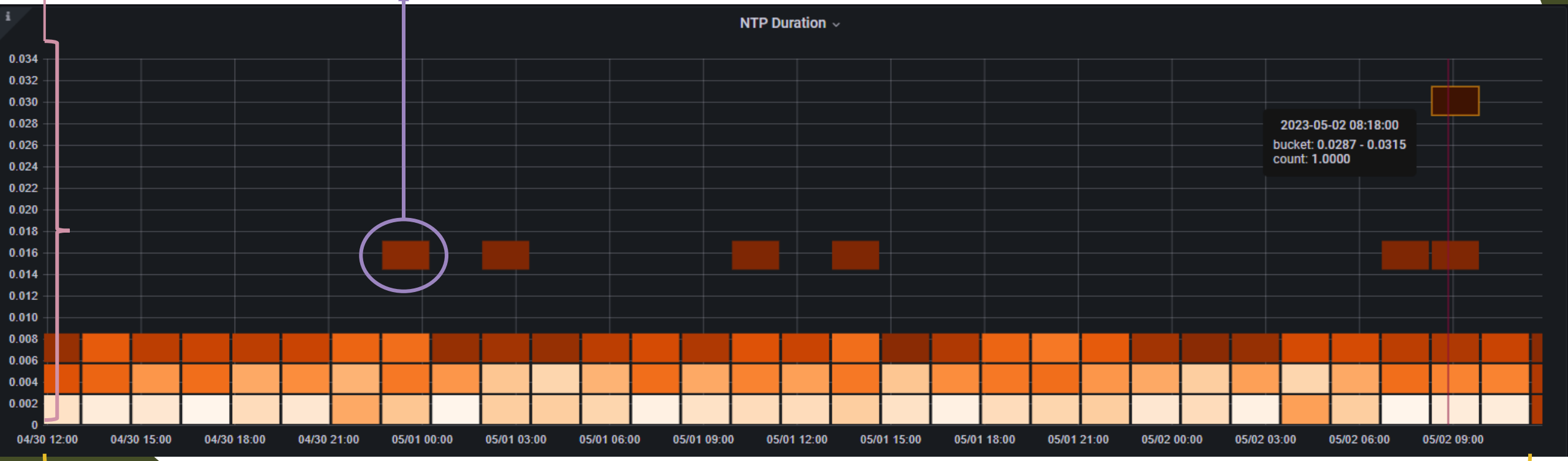
Avrage Response time

Total requests counts

Visualization of data in Grafana (cont.)

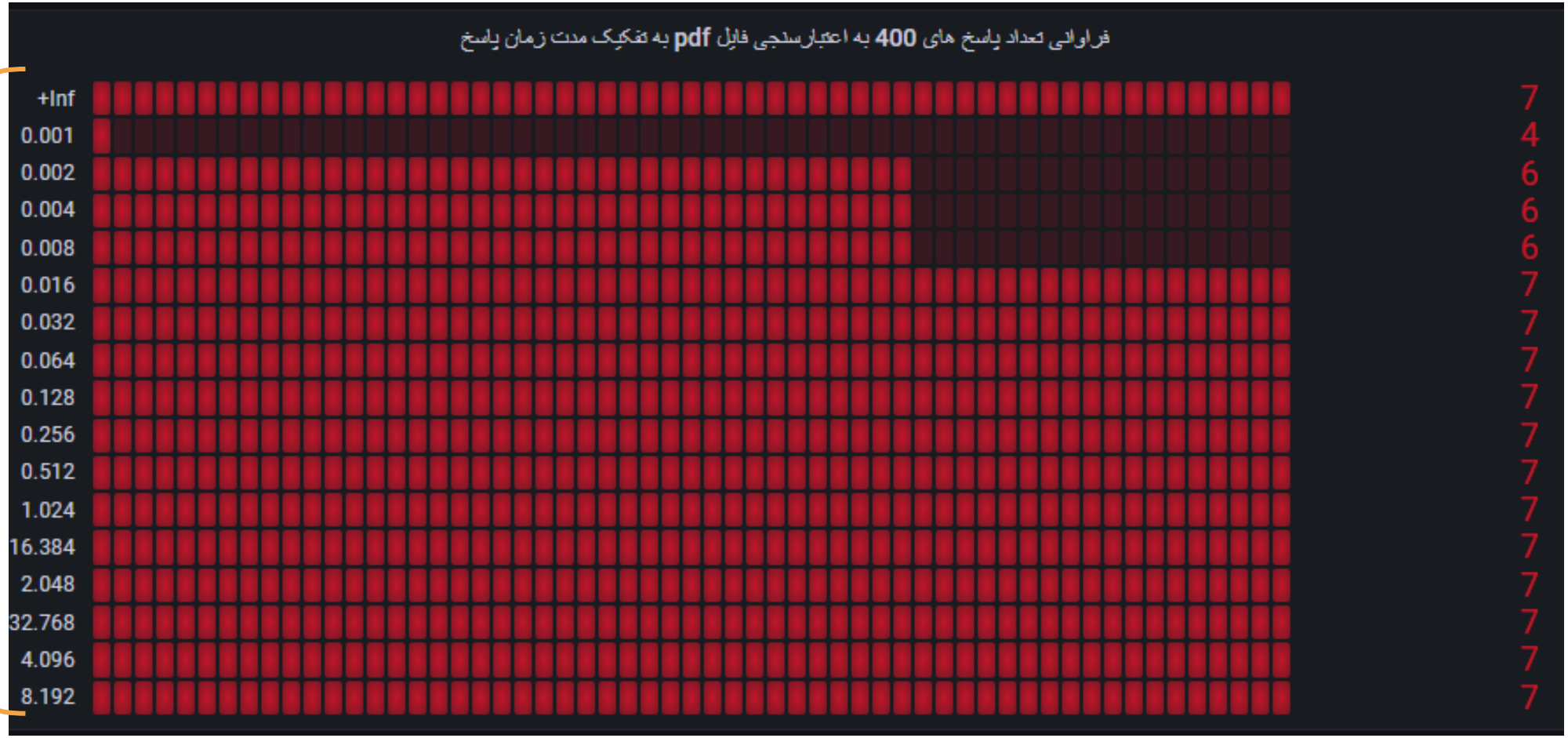
Metric (Response time)

Data counts



Date and time

Visualization of data in Grafana (cont.)



Metric
(Response time)

“If you can’t measure it, you can’t improved it.”

Peter F.Drucker

Summary

- Metrics
- Metrics type
- Metrics output formats
- How to choose good metrics?
- Data collector and storage
- Data visualization (Grafana)

Thank you for your attention