splunk> .conf2017

Metrics Analysis with the Splunk Platform

How to work with metrics for Monitoring, Alerting, and ad-hoc analysis at scale

Michael Porath | Product Management, Splunk
Allan Yan | Principal Software Engineer, Splunk

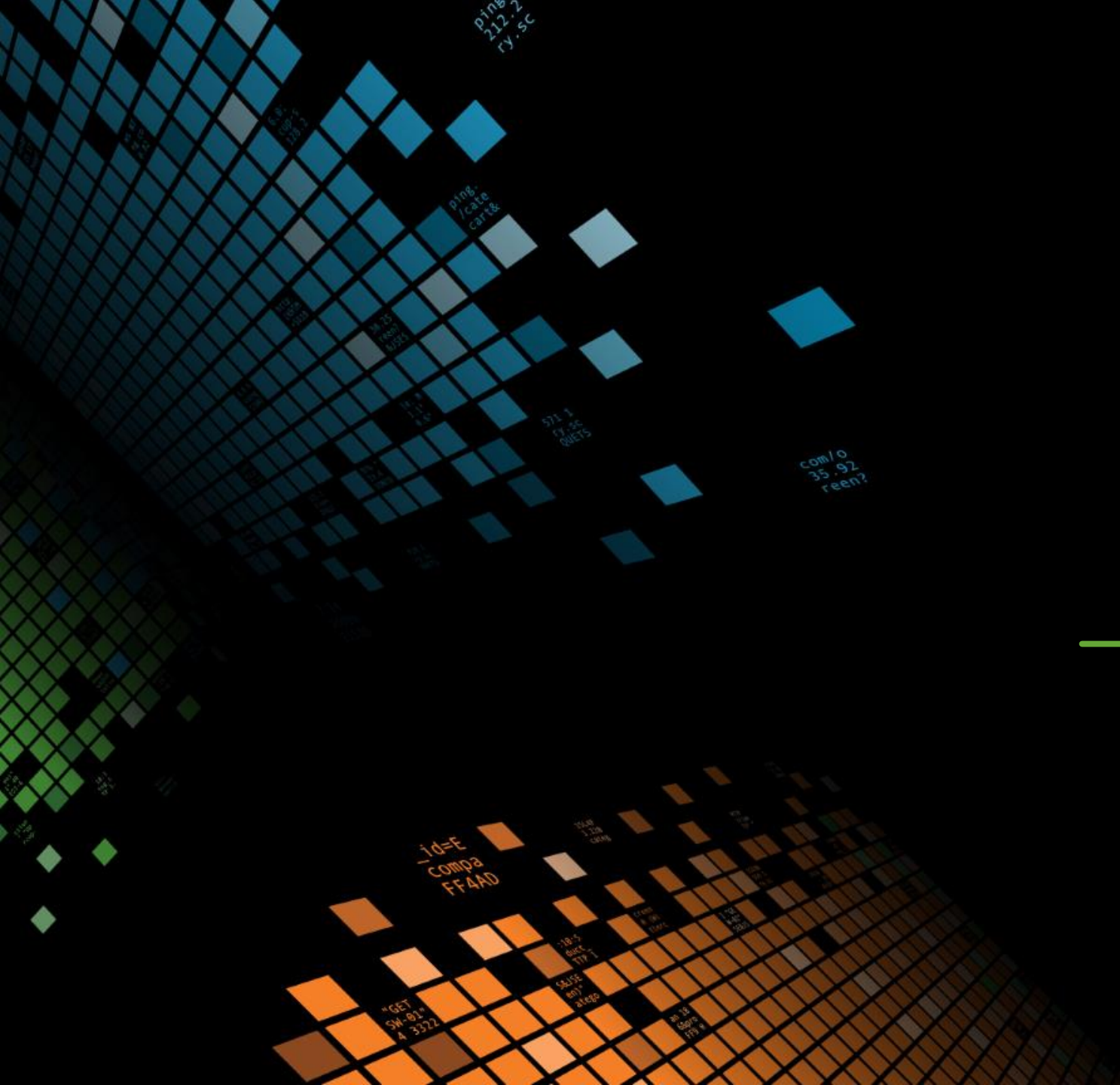
September 2017 | Washington, DC

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> .conf2017



Why Metrics?

Section subtitle goes here

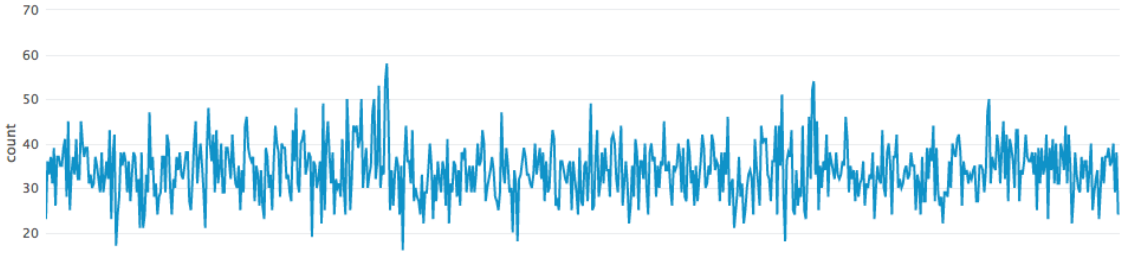
splunk> .conf2017

Why Metrics?

… when you already use logs?

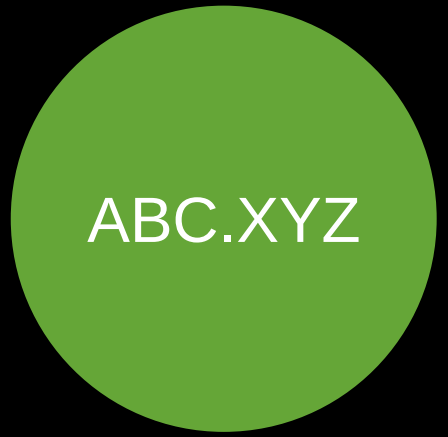
- **Logs**
  - Unstructured data
  - Needle in the haystack
  - Can tell you all about the “why”
  - Answers questions you might not even have yet
  - Very versatile

- **Metrics**
  - Structured Data
  - Best way to observe a process or device
  - Easy way to do monitoring
  - You know what you want to measure
  - e.g. performance, CPU, Number of users, memory used, network latency, disk usage



Terminology - What is a Measurement?

Time



Metric Name

`system.cpu.idle`

Measure

numeric data point, different types such as count, gauge, timing, sample, etc

Dimensions

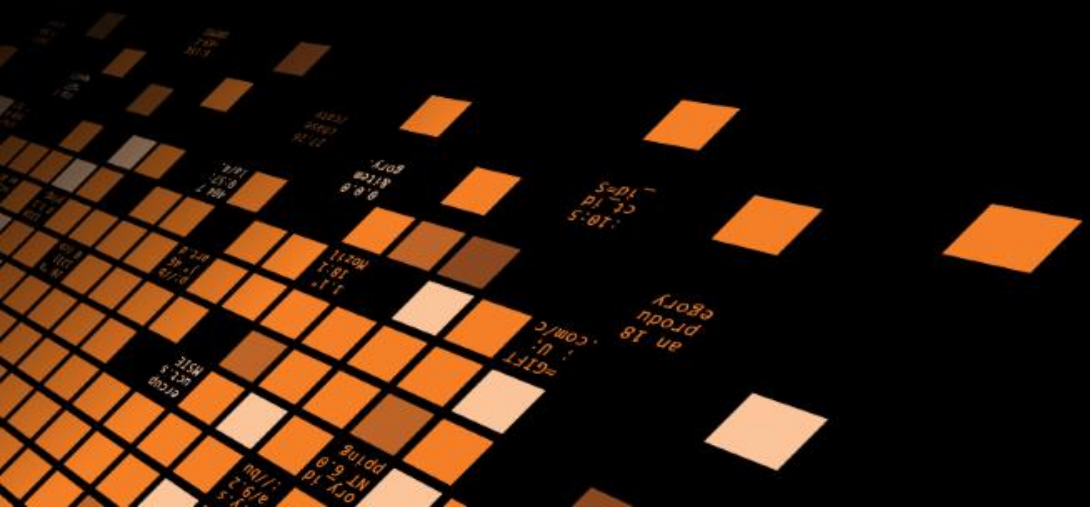
Host (10.1.1.100, web01.splunk.com)

Region (e.g., us-east-1, us-west-1, us-west-2, us-central1)

IntanceTypes (e.g., t2.medium, t2.large, m3.large)

splunk> .conf2017

“Splunk provides one platform to analyze and investigate across both Logs and Metrics



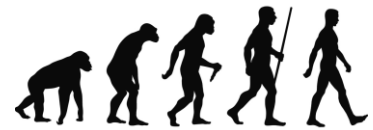
splunk> .conf2017

Evolution of Splunk Search Capabilities

Raw event search on log events

- Initially, Splunk was optimized for Raw event search.
- Sparse search
  - look for an event or events that occurs infrequently within a large dataset
  - Needle in the haystack or rare term
- retrieve events from an index or indexes
- typically used when you want to troubleshoot a problem
- Examples:
  - checking error codes: index=mylogs 404
  - correlating events: index=mylogs region=east-1 earliest=-1s
  - investigating security issues: sshd failed OR failure

Raw Event Search



splunk> .conf2017

Evolution of Splunk Search Capabilities (cont.)

Statistical analysis on log events

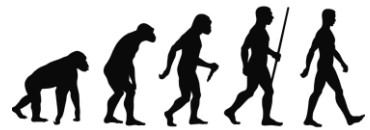
- Over time, **added** optimization for statistical queries.
- Dense search
  - scan through and report on many events
  - perform statistical calculation
- Always require fields and statistical commands.
  - Extract fields at index (faster) or search (flexible) time
  - Build DM and accelerate it with summaries
- Examples
  - Count number of errors occurred: error | stats count
  - Median memory usage for each app: … | stats median(mem_used) BY app
  - Avg thruput of each host: … | tstats avg(thruput) BY host
  - Count unique ip address logged on each host in the last 24 hours: …| tstats dc(ip) WHERE <dataset> BY host earliest=-1d

Splunk 3.0

Splunk 5.0

Optimization for Statistical Queries

Raw Event Search



splunk> .conf2017

Evolution of Splunk Search Capabilities (cont.)

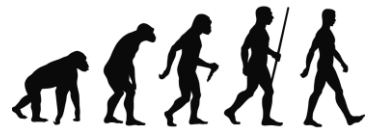
Metric analysis on metric data points

- With the advent of containers, large number of devices in IoT, larger infrastructures, collecting metrics and measuring the behavior becomes more important.
- Dense search performs statistical calculation
- Always has a schema
- Data types are known
- Vey high volume
- Search is time sensitive
- Examples
  - Calculate average memory usage per application in a week with one hour window
  - Calculate 95th percentile of CPU idle time of each host in the last 24 hours
  - Calculate the total down time of each server in the last 6 month
  - List the dimensions of each metric
  - List the OS running on each data center of each region

Optimization for Metrics Queries

Optimization for Statistical Queries

Raw Event Search

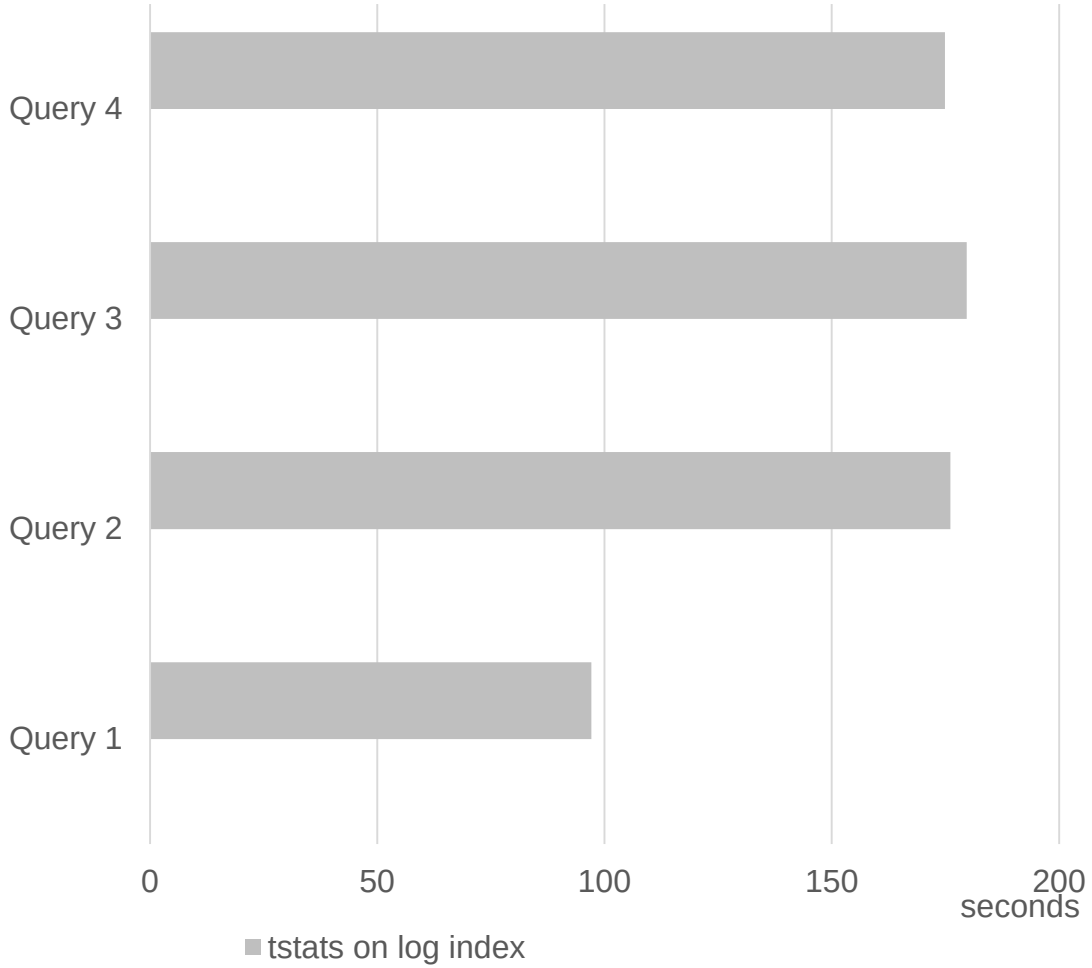


Design Goals for Metrics Analysis

- tstats based solution is great for log data which does not enforce a schema at ingestion time
- But it is not optimized for metric analysis:
  - High data volumes
  - Structured data
  - Index time field
  - Aggregation on numeric field
  - Rarely search across all metric series
  - Cheaper and faster real-time search
  - Fast retrieval of metrics catalog information

Search Improvement

Search Response Time Over 270M Data Points



1. **Average for a metric series**
   avg(_value) WHERE metric_name=mem.used

2. **Average for a metric series, split by low cardinality field**
   avg(_value) WHERE metric_name=mem.used BY region

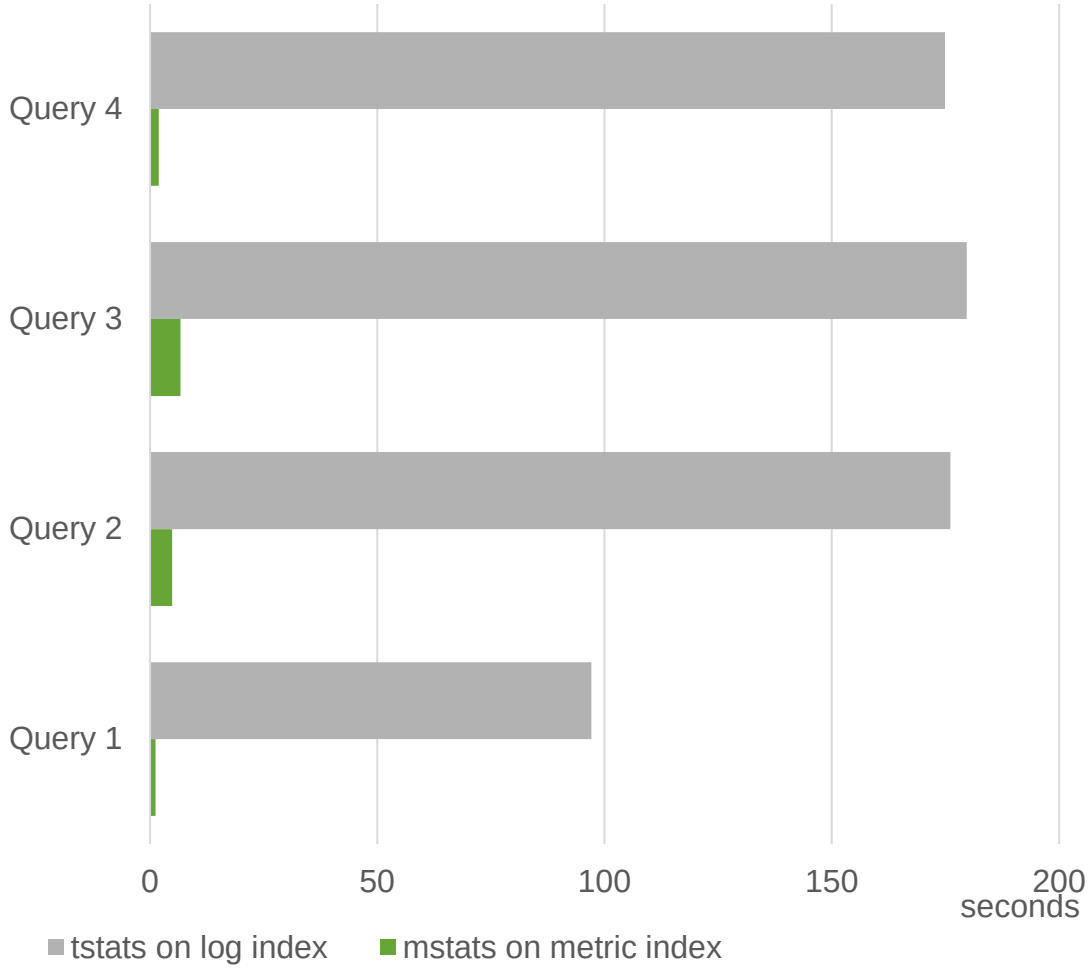
3. **Average for a metric series, split by a high cardinality field**
   avg(_value) WHERE metric_name=mem.used BY host

4. **Average for a metric time series (grouped by time)**
   avg(_value) WHERE metric_name=mem.used span=30s

splunk> .conf2017

Search Improvement

Search Response Time Over 270M Data Points



1. **Average for a metric series**
   avg(_value) WHERE metric_name=mem.used

2. **Average for a metric series, split by low cardinality field**
   avg(_value) WHERE metric_name=mem.used BY region

3. **Average for a metric series, split by a high cardinality field**
   avg(_value) WHERE metric_name=mem.used BY host

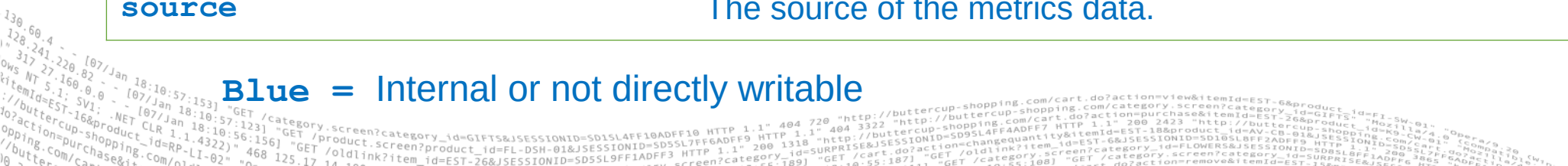
4. **Average for a metric time series (grouped by time)**
   avg(_value) WHERE metric_name=mem.used span=30s

splunk> .conf2017

Structure of a metrics index

Field	Required	Description	Example
metric_name	•	The metric name.	os.cpu.user
_time	•	The timestamp of the metric in UNIX time notation.	
_value	•	The numeric value of the metric.	42.12345
<dim0>...<dimN>		An arbitrary number of dimensions.	e.g. ip=10.2.1.166
metric_type	•	Currently only gauge “g” is supported	
_dims	•	Dimension names. Dimensions indicate how metrics are split. Internal, should not be changed	
host	•	The origin host.	
index	•	The metrics index name.	
sourcetype	•	The data structure of the metric.	
source		The source of the metrics data.	

Blue = Internal or not directly writable



splunk> .conf2017

Introducing mstats

- New SPL command optimized for statistical queries on metrics data
- Like tstats, it is a generating command that generates reports.
- unlike tstats, it can search from both on-disk data (historical search) and in-memory data (realtime search)
- mstats cannot search event index
- tstats and other generating commands (search, metadata etc.) cannot search metrics index

Syntax

| **mstats** <stats-function> …

WHERE index=<*metric_index*> **AND** metric_name=<*metricname*> …]

[span=<*timespan*>] [**BY** <*metricname*|*dimension*>]

Metrics Catalog

- New **SPL command**: mcatalog
- optimized to list catalog information (e.g. metric names, dimensions) of metric store

Syntax

| **mcatalog** values(<field>) …

[**WHERE** index=*<metric_index>*

AND metric_name=*<metricname>* …]]

[**BY** *<metricname|dimension>*]

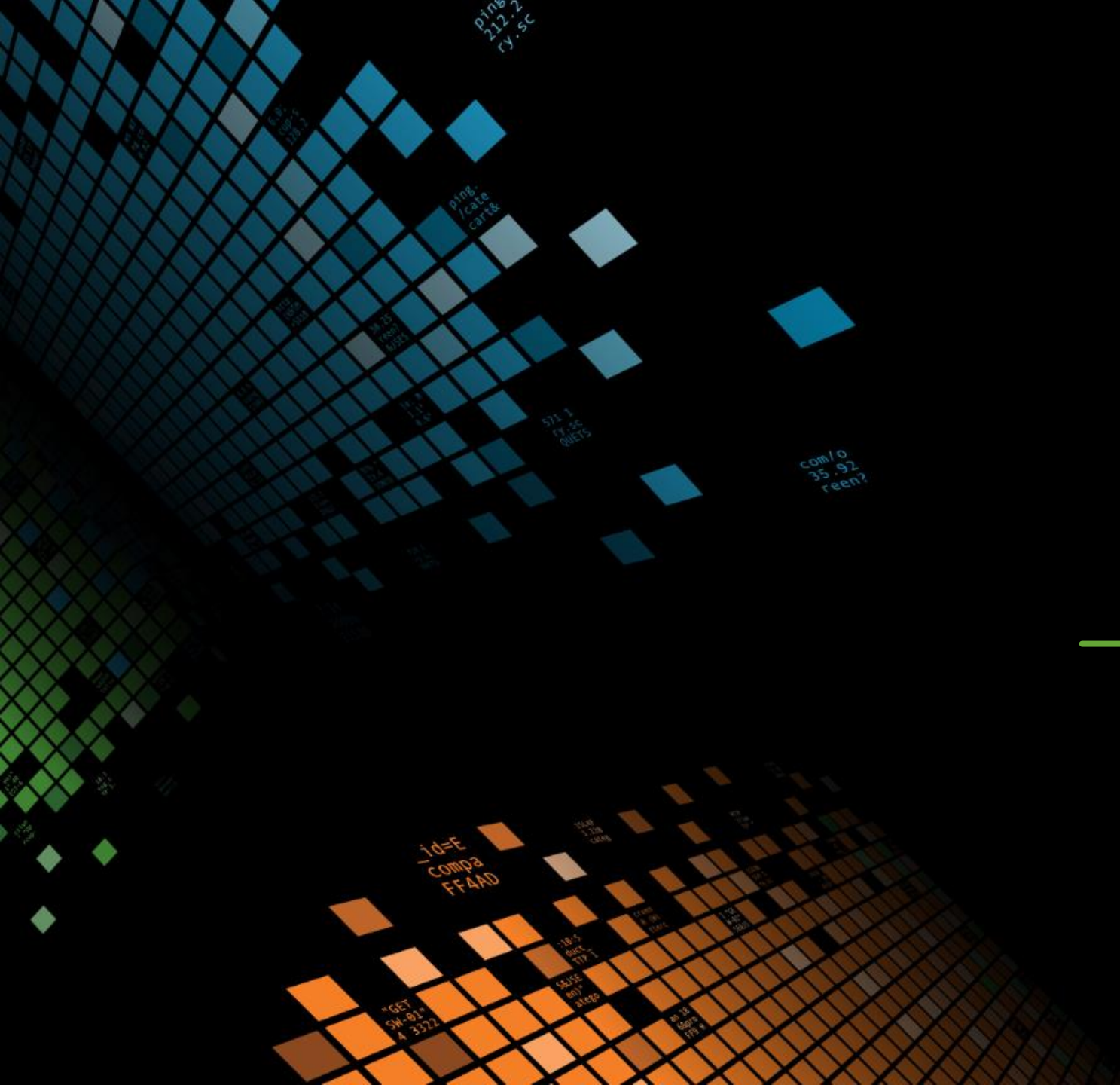
- New **REST endpoints**
- list metric names: /services/catalog/metricstore/metrics
- list dimension names: /services/catalog/metricstore/dimensions
- list dimension values: /services/catalog/metricstore/dimensions/{dimension-name}/values
- You can also use filters with these endpoints to limit results by index, dimension, and dimension values.

splunk> .conf2017

Compare & Contrast Stats Commands

	Historical Search	Realtime Search (Summary)	Realtime Search (Raw)	Metric Index	Event Index	Aggregate on Index-time Field	Aggregate on Search-time Field	Aggregate on _value Field
mstats	X	X		X				X
mcatalog	X			X		X		
tstats	X				X	X	X	
search+stats	X		X		X	X	X	

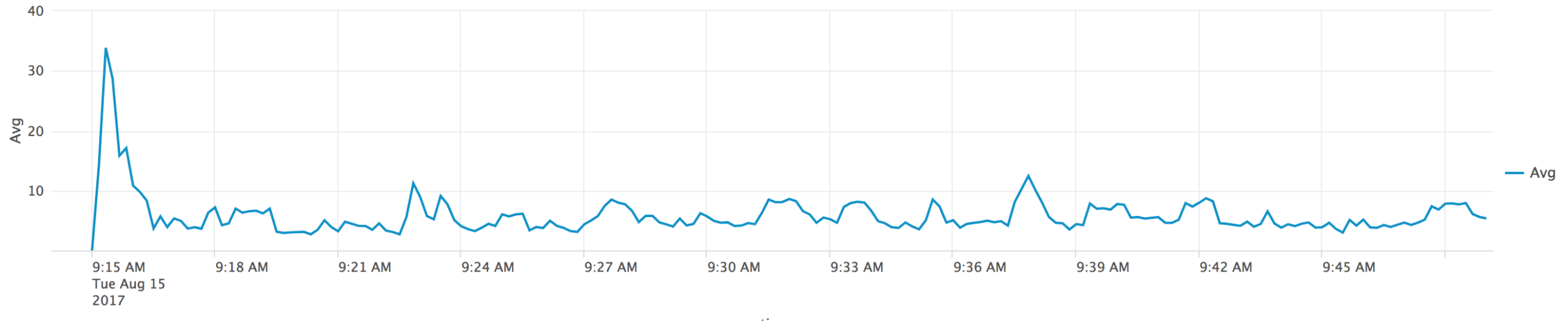
splunk> .conf2017



Use Cases

Example: Simple query

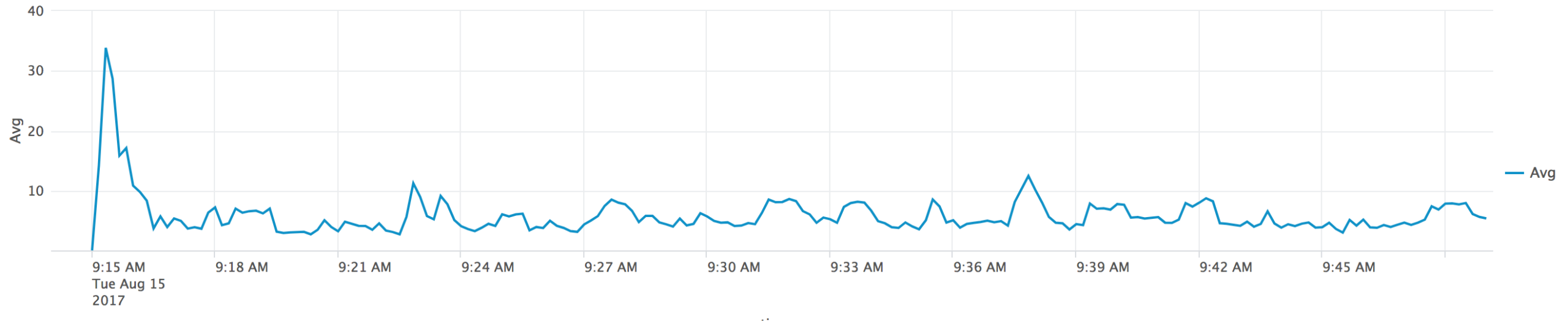
```
| mstats avg(_value) prestats=true WHERE metric_name="cpu.system.value" span=10s
| timechart avg(_value) as "Avg" span=10s
```



splunk> .conf2017

Example: Simple query

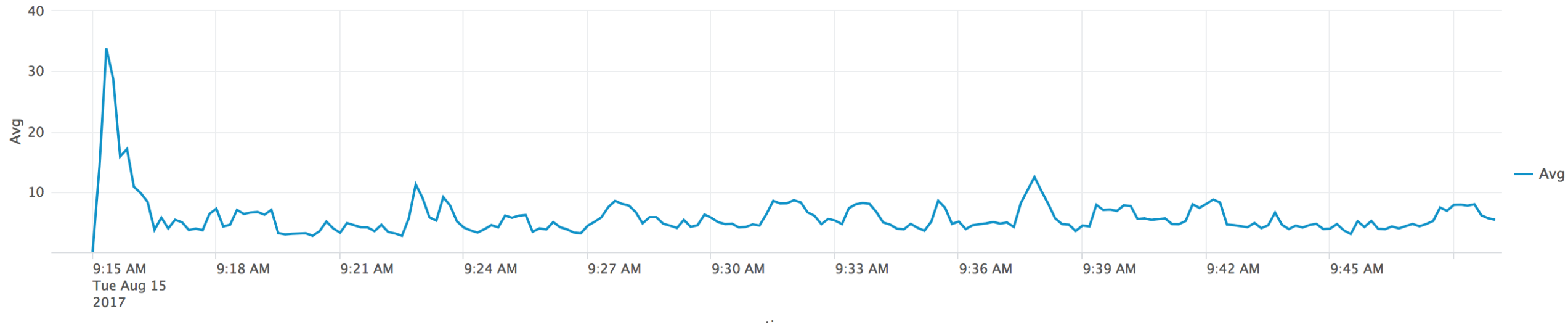
```
| mstats avg(_value) prestats=true WHERE metric_name="cpu.system.value" span=10s
| timechart avg(_value) as "Avg" span=10s
```



splunk> .conf2017

Example: Simple query

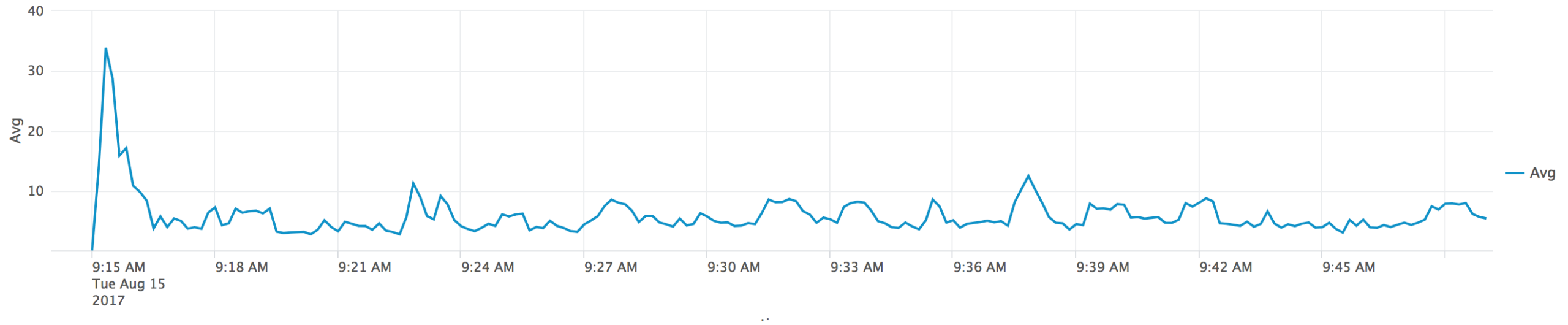
```
| mstats avg(_value) prestats=true WHERE metric_name="cpu.system.value" span=10s
| timechart avg(_value) as "Avg" span=10s
```



splunk> .conf2017

Example: Simple query

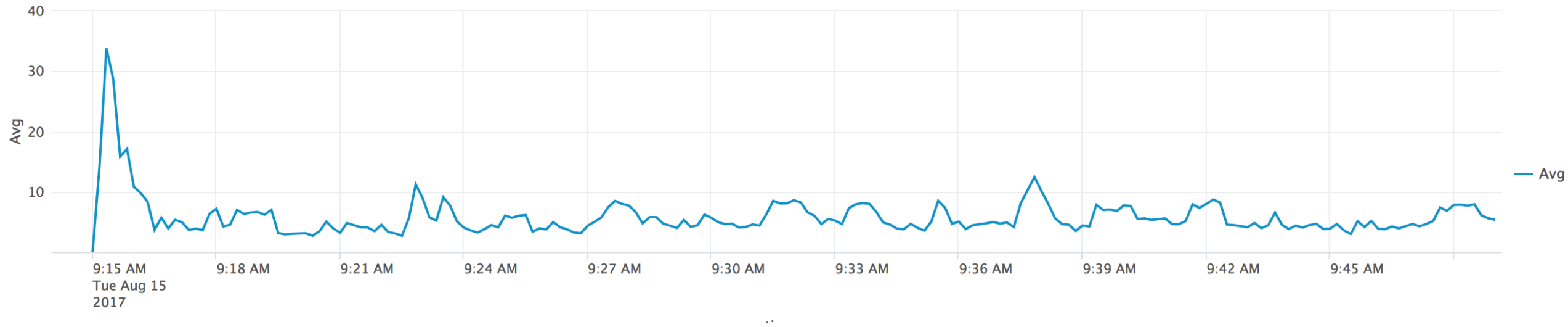
```
| mstats avg(_value) prestats=true WHERE metric_name="cpu.system.value" span=10s
| timechart avg(_value) as "Avg" span=10s
```



splunk> .conf2017

Example: Simple query

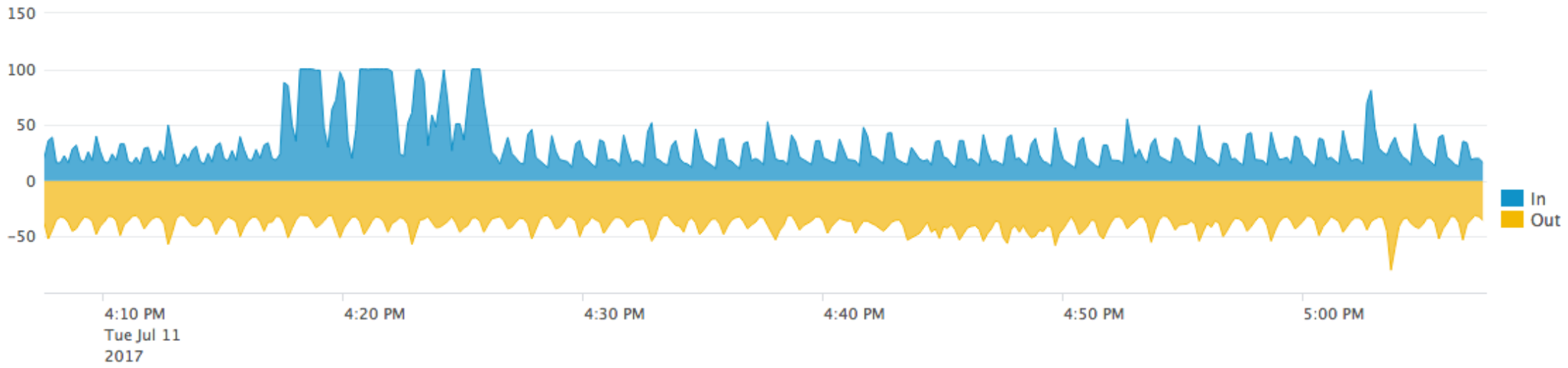
```
| mstats avg(_value) prestats=true WHERE metric_name="cpu.system.value" span=10s
| timechart avg(_value) as "Avg" span=10s
```



splunk> .conf2017

Example: Ingress vs egress data

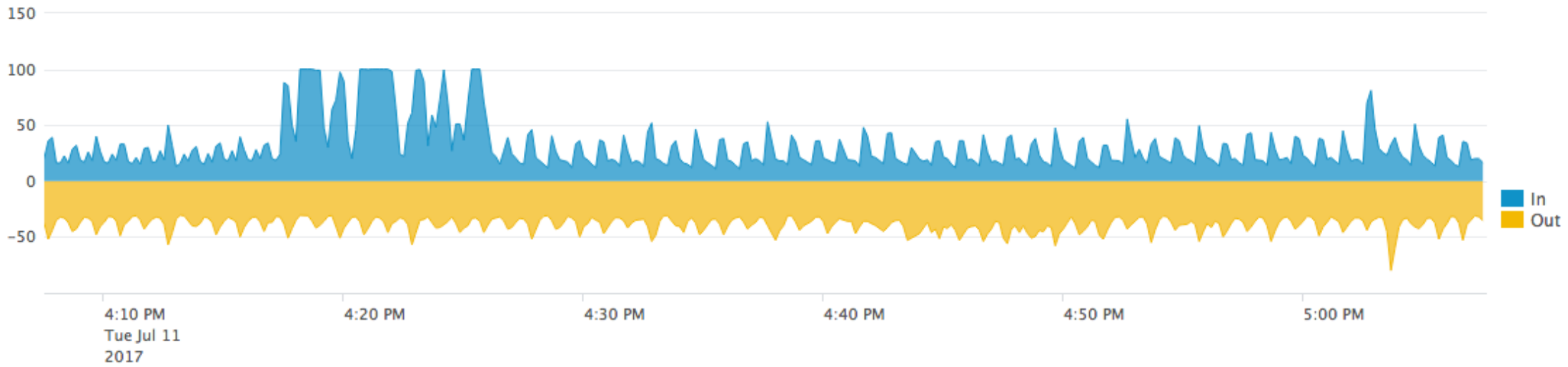
```
| mstats avg(_value) as "In" WHERE metric_name="network.rx" span=10s
| appendcols
    [| mstats avg(_value) as "Out" WHERE metric_name="network.tx" span=10s
    | eval Out = -Out]
| timechart first("In") AS "In" first("Out") AS "Out"
```



Example: Ingress vs egress data

With wildcards

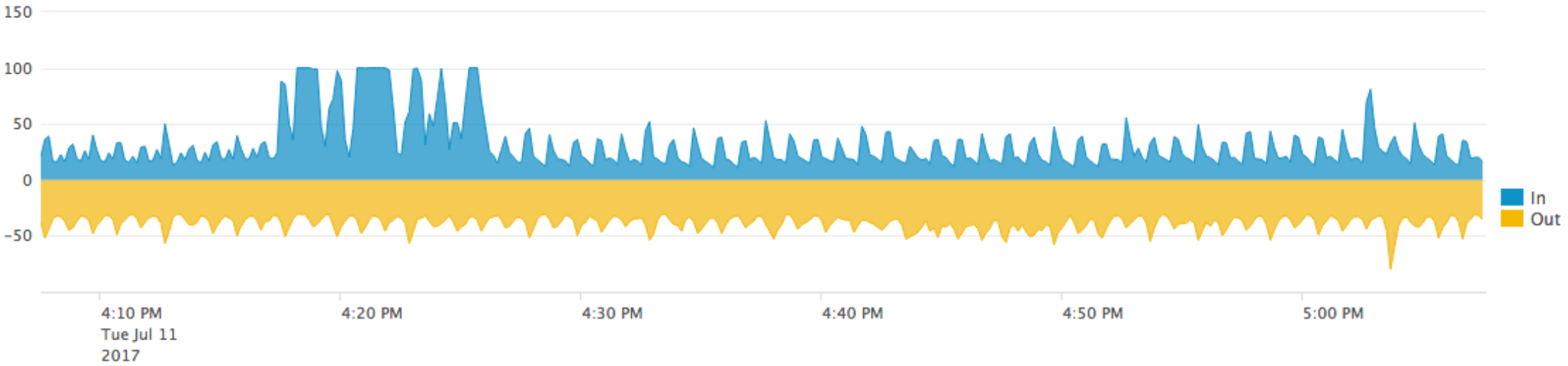
```
| mstats avg(_value) prestats=t WHERE metric_name="network.*" span=10s BY metric_name
| timechart avg(_value) span=10s BY metric_name
| rename network.rx AS In, network.tx AS Out
| eval Out = -Out
```



Example: Ingress vs egress data

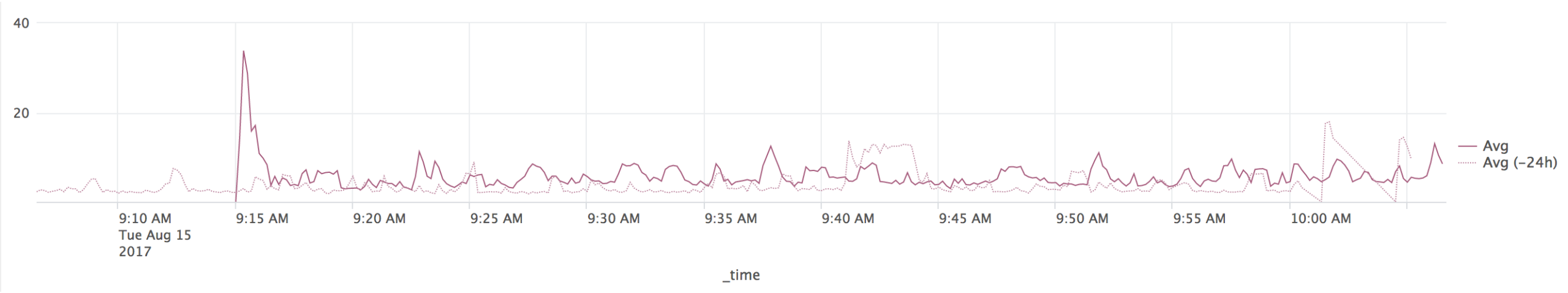
With wildcards

```
| mstats avg(_value) prestats=t WHERE metric_name="network.*" span=10s BY metric_name
| timechart avg(_value) span=10s BY metric_name
| rename network.rx AS In, network.tx AS Out
| eval Out = -Out
```



Example: Compare today vs yesterday

```
| mstats avg(_value) AS "Avg" WHERE metric_name="cpu.system.value" span=10s
| append
   [| mstats avg(_value) AS "Avg (-24h)" WHERE metric_name="cpu.system.value" earliest=-24h-1h latest=-24h span=10s
    | eval _time = round(relative_time(_time, "+24h"))
]
| timechart first("Avg") AS "Avg", first("Avg (-24h)") AS "Avg (-24h)" span=10s
```

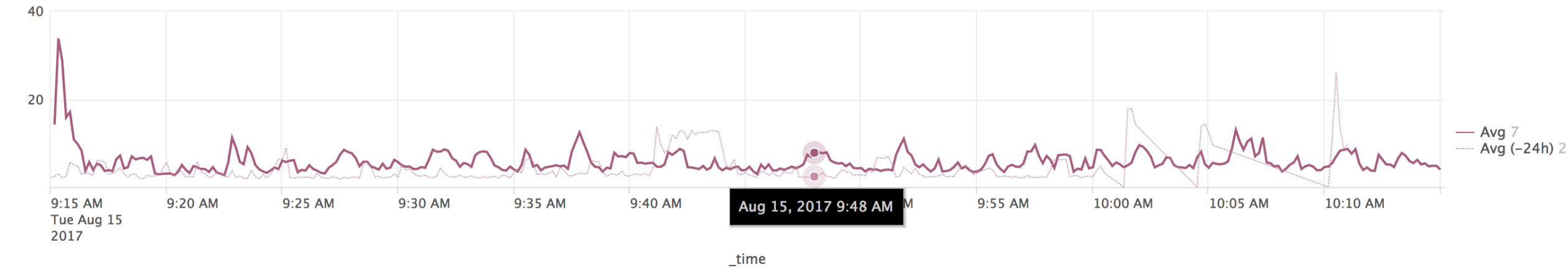


splunk> .conf2017

Chart Configurations

Charting improvements in 7.0

```
    <option name="charting.drilldown">none</option>
    <option name="charting.chart">line</option>
    <option name="charting.chart.nullValueMode">connect</option>
    <option name="charting.fieldColors">{"Avg":"#a65c7d","Avg (-24h)":"#C18EA5"}</option>
    <option name="charting.gridLinesX.showMajorLines">true</option>
    <option name="charting.axisY.abbreviation">auto</option>
    <option name="charting.axisY.includeZero">true</option>
    <option name="charting.legend.mode">seriesCompare</option>
    <option name="charting.fieldDashStyles">{"Avg (-24h)":"shortDot"}</option>
    <option name="charting.lineWidth">1</option>
</chart>
```



splunk> .conf2017

Annotate with Events

splunk> .conf2017

Examples

- “display **meaningful events** over performance data”

- “When we deploy new software releases on the servers, **we set a downtime in Nagios to avoid alarming during this time**. Dashboards (for example, count of events) are not aware of these downtimes. So the chart shows zero events for the deployment time **without any explanation**“

Event Annotations

Event Annotations

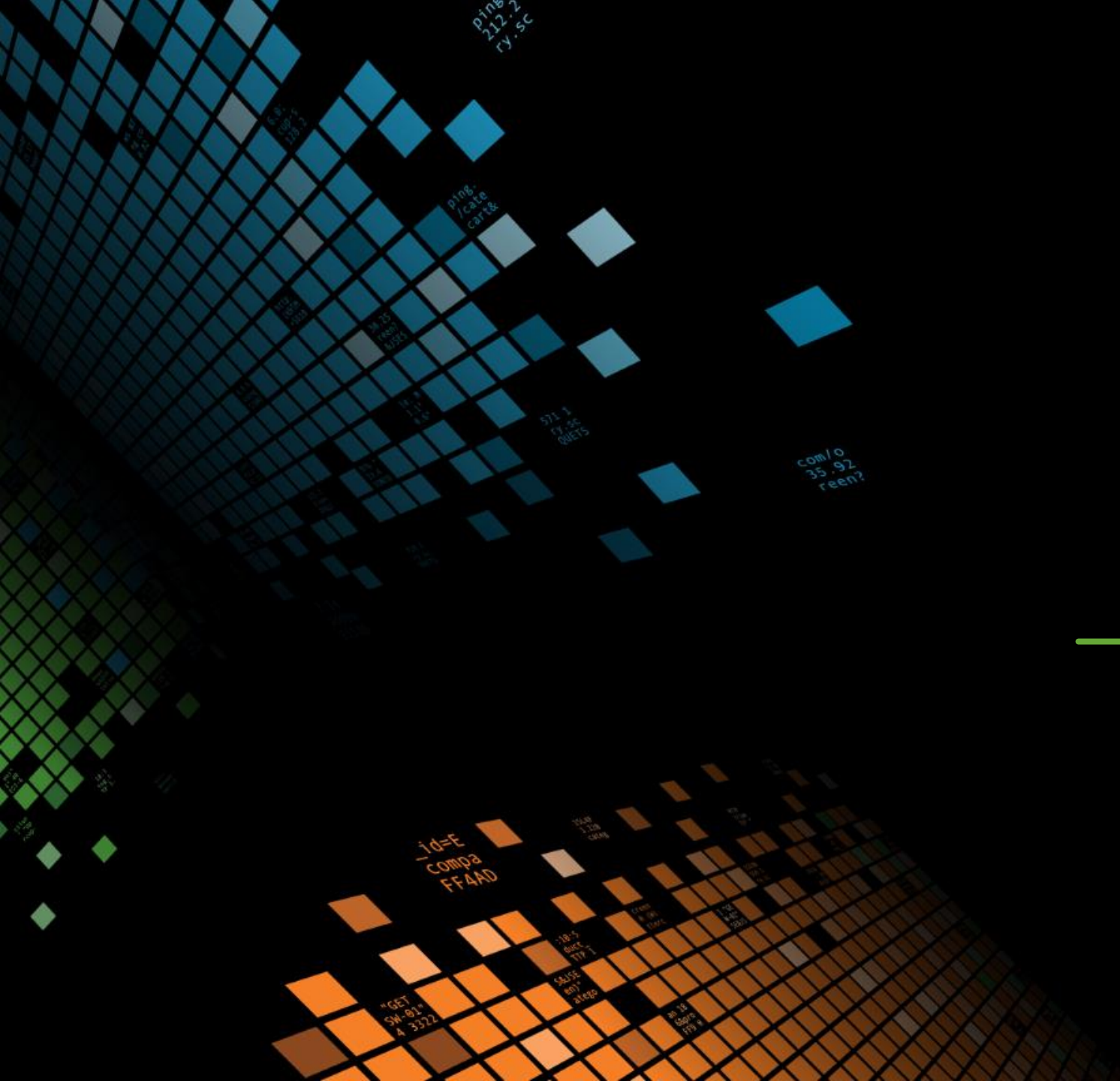
Configuration

- Based on two separate searches:
  - Time series (`mstats`)
  - Events (Non-transforming search)
- Currently needs to be configured in SimpleXML

```
<!-- Base search that drives the visualization -->
<search>
 <query>| mstats avg(_value) WHERE
      metric_name=sensor.heartrate</query>
 <earliest>-24h@h</earliest>
 <latest>now</latest>
</search>

<!-- Secondary search that drives the annotations -->
<search type="annotation">
 <query>
  index=car_logs crash
  | eval annotation_label = "Aaaaaah"
  | eval annotation_category = "I'm Spinning"
</query>
 <earliest>-24h@h</earliest>
 <latest>now</latest>
</search>

<!-- Customize the event annotation colors based on category name -->
<option name="charting.annotation.categoryColors">
 {"crash":"0xff3300","start":"0xff9900"}
</option>
```



Demo

Alerting with Metrics

splunk> .conf2017

Alert on a threshold

Using existing alerts

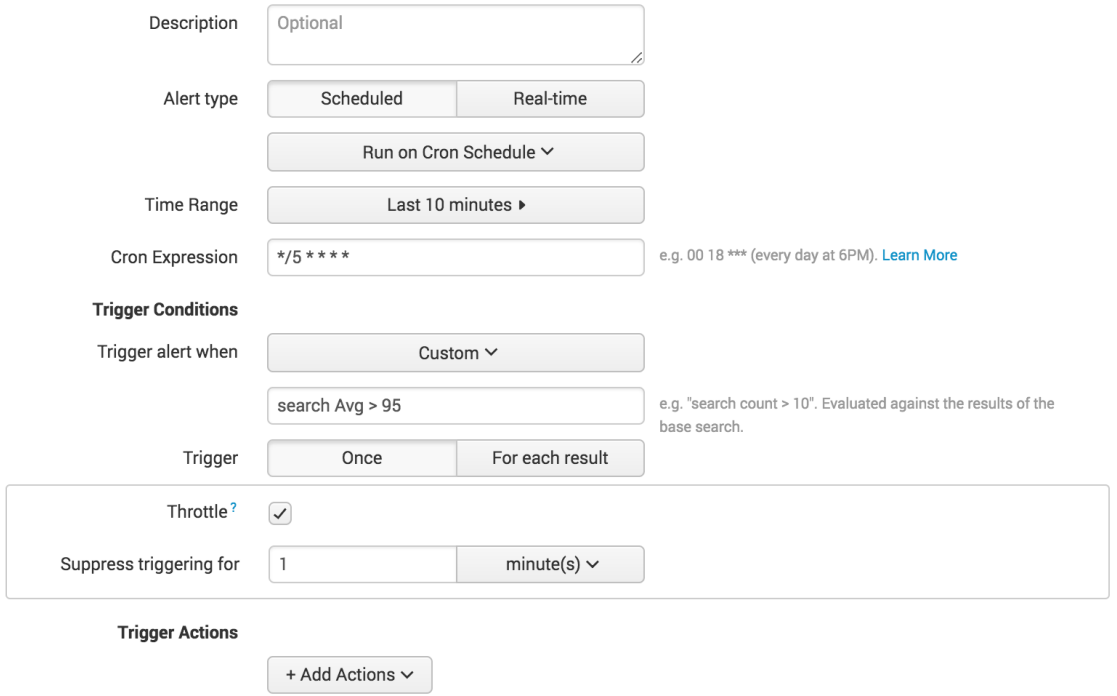
- Base Search

```
| mstats avg(_value) AS "Avg" WHERE metric_name="cpu.system.value" span=10s
```

- Trigger Condition

```
| search Avg > 95
```

- Alert Settings
  - E.g. Run every 5 minutes
  - Check if threshold (95) was crossed in the last 10 minutes



Key Takeaways

Splunk provides one platform to analyze and investigate across both Logs and Metrics

1. Splunk natively supports metrics at scale
2. Metrics use many of the same capabilities you know and love working with events (SPL, alerts, visualizations, dashboards)
3. Metrics + Logs = ❤

Tomorrow, Wednesday
4.35PM, Room 144ABC

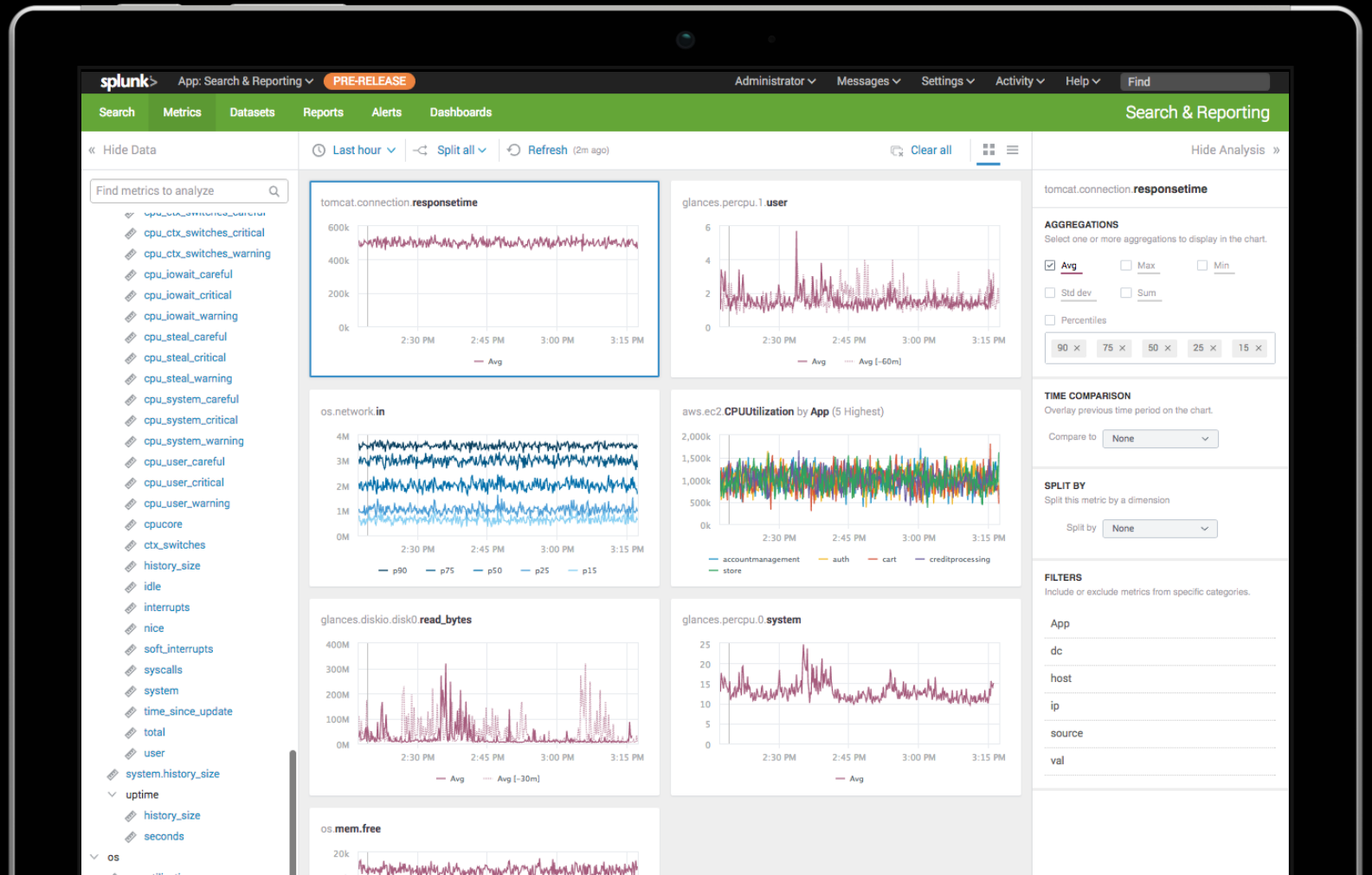
Getting Metrics In

Splunking Metrics –
The Right Way

splunk> .conf2017

Sneak Preview

Prototype of Metrics Analysis UI



- Query logs and metrics in the same environment
- New user interface to quickly visualize, aggregate, and analyze any indexed metric
- Support for multiple dimensions allows easy grouping and filtering
- **See us at Splunk Labs!**

splunk> .conf2017

Early Access Program

- **Requirements**
  - You have metrics use cases
  - Willingness to use Metric Analysis UI and give feedback
  - Regular assistance from Splunk Product Management to setup metrics deployment

- metric-analysis-eap@splunk.com

Don't forget to rate this session in the .conf2017 mobile app