Auto Scaling System for AWS

Shuo Wang @ GrowingIO

About GrowingIO

- GrowingIO 是基于用户行为的新一代数据分析产品，无需埋点即可采集全量、实时用户行为数据，数据分析更精细，帮助管理者、产品经理、市场运营、数据分析师、增长黑客等提升转化率、优化网站 / APP，实现用户快速增长和变现。

Purpose of Auto Scaling System

- Dynamically scale your Amazon EC2 capacity up or down automatically according to conditions you define.

- Cost saving according to your necessary

AWS EC2 Pricing

- On-Demand Instances, 100%
- Reserved Instances, ~60%
- Spot Instances, ~30%

On-Demand Instances

- On-Demand instances let you pay for compute capacity by the hour with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs.

Reserved Instances

- Standard RIs: These provide the most significant discount (up to 75% off On-Demand) and are best suited for steady-state usage.

- Convertible RIs: These provide a discount (up to 45% off On-Demand) and the capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value. Like Standard RIs, Convertible RIs are best suited for steady-state usage.

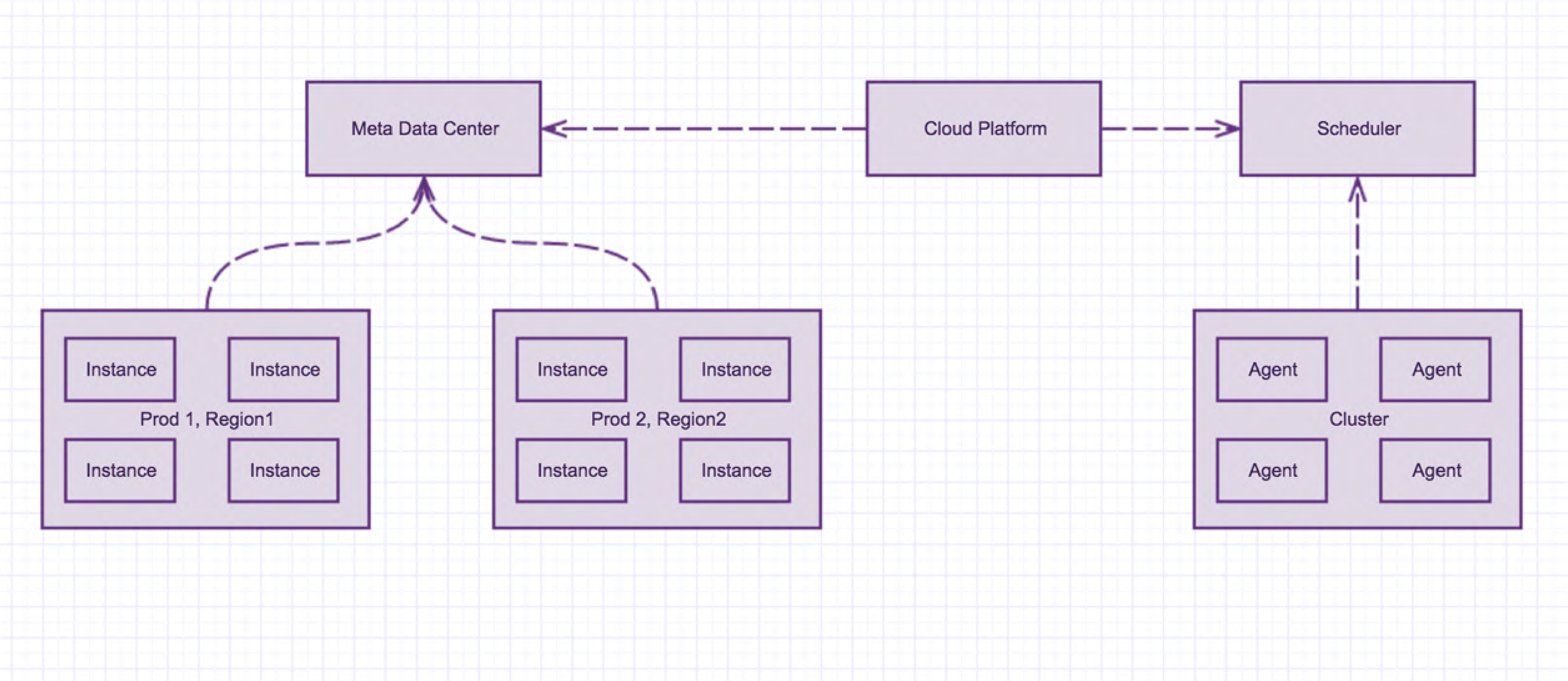
- Scheduled RIs: These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

Spot Instances

- Spot instances run when your bid price exceeds the Spot price, and provide the reliability, security, performance, control, and elasticity of Amazon EC2, at low market-driven prices.

- Reduce your operating costs by up to 50-90% with Spot instances, compared to On-Demand instances.

Architecture of Auto Scaling System



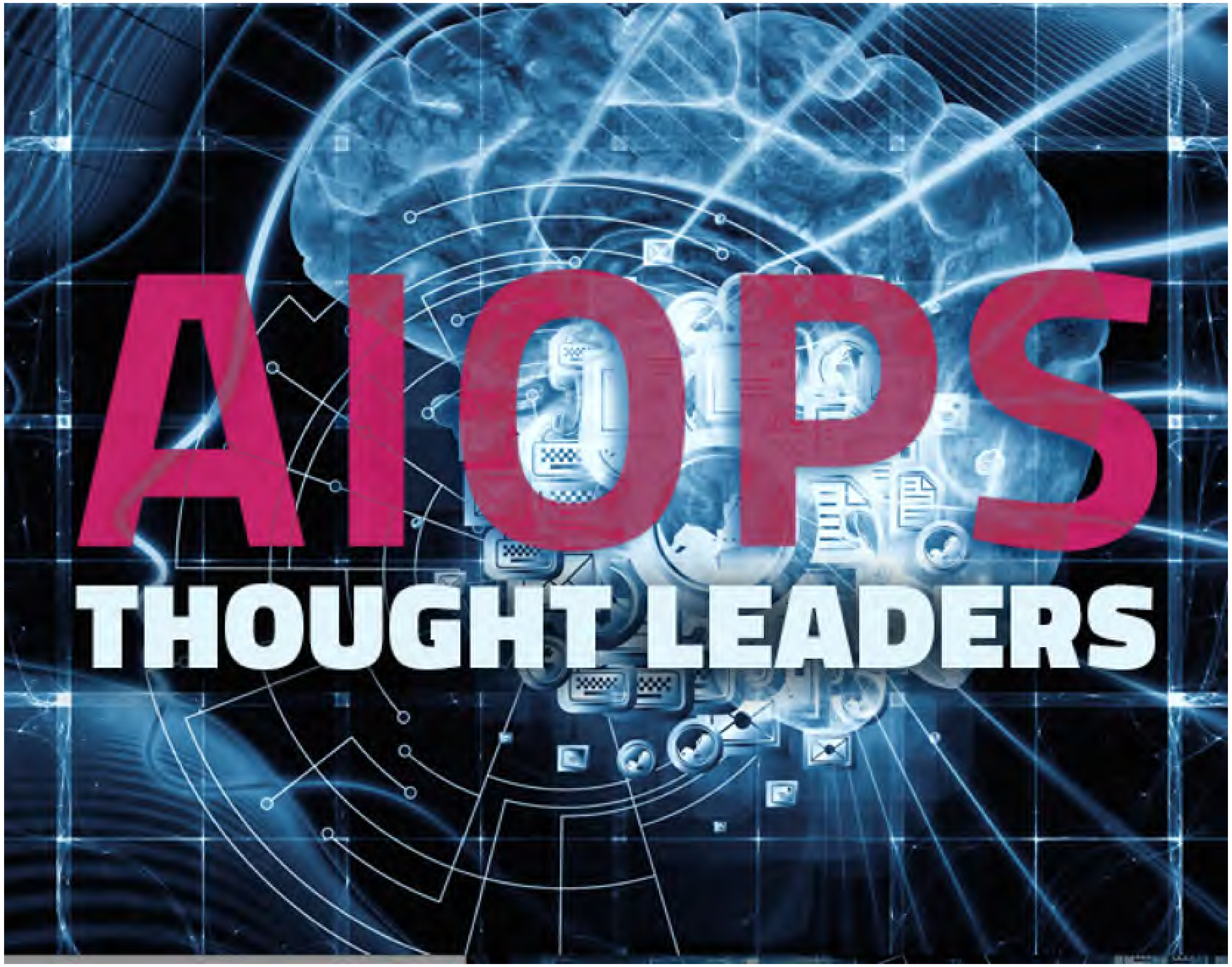
Pain Points & Challenges

- No Instances available
- Protected instances can not terminate
- Network issue, Brazil

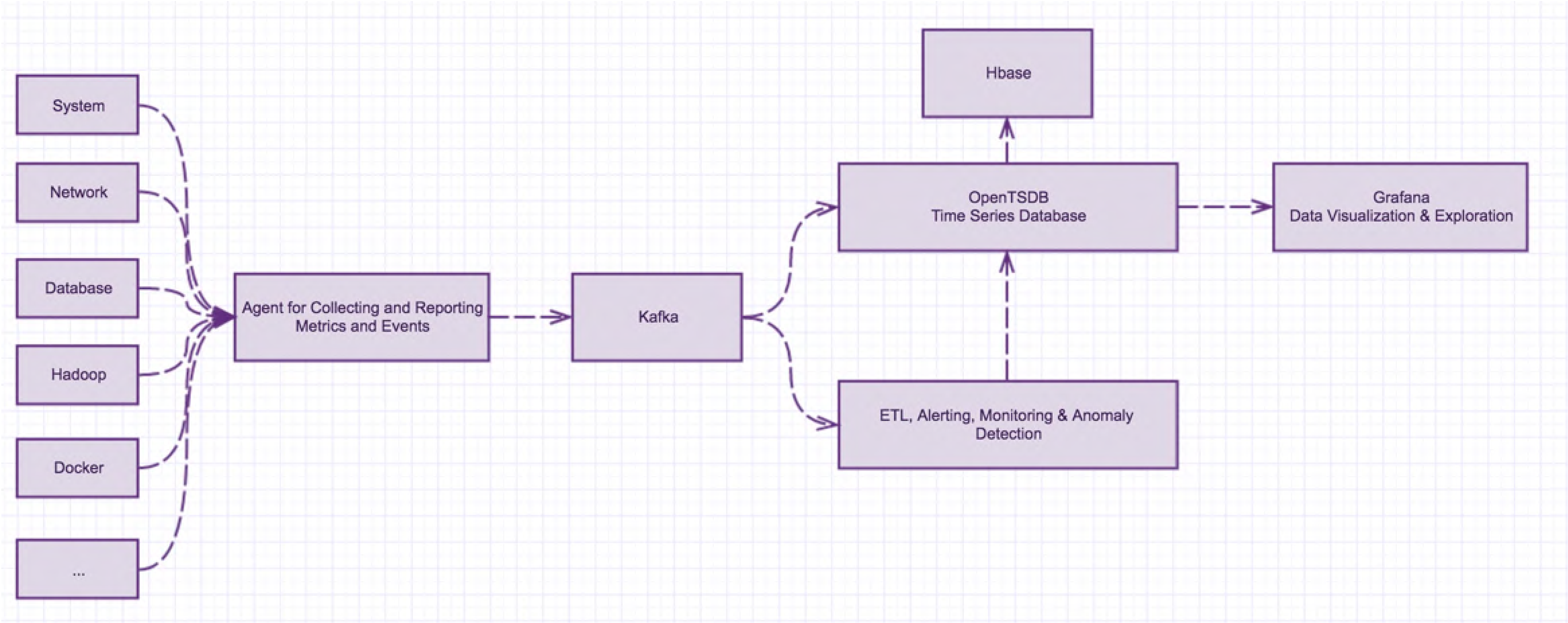
Basic Principles

- Launch Spot Instances as many as possible if necessary
- Launch another instance type if failed
- Launch another zone’s instances if failed
- Shutdown On Demand as many as possible if necessary
- Ensure clusters can work properly at least On-Demand instances

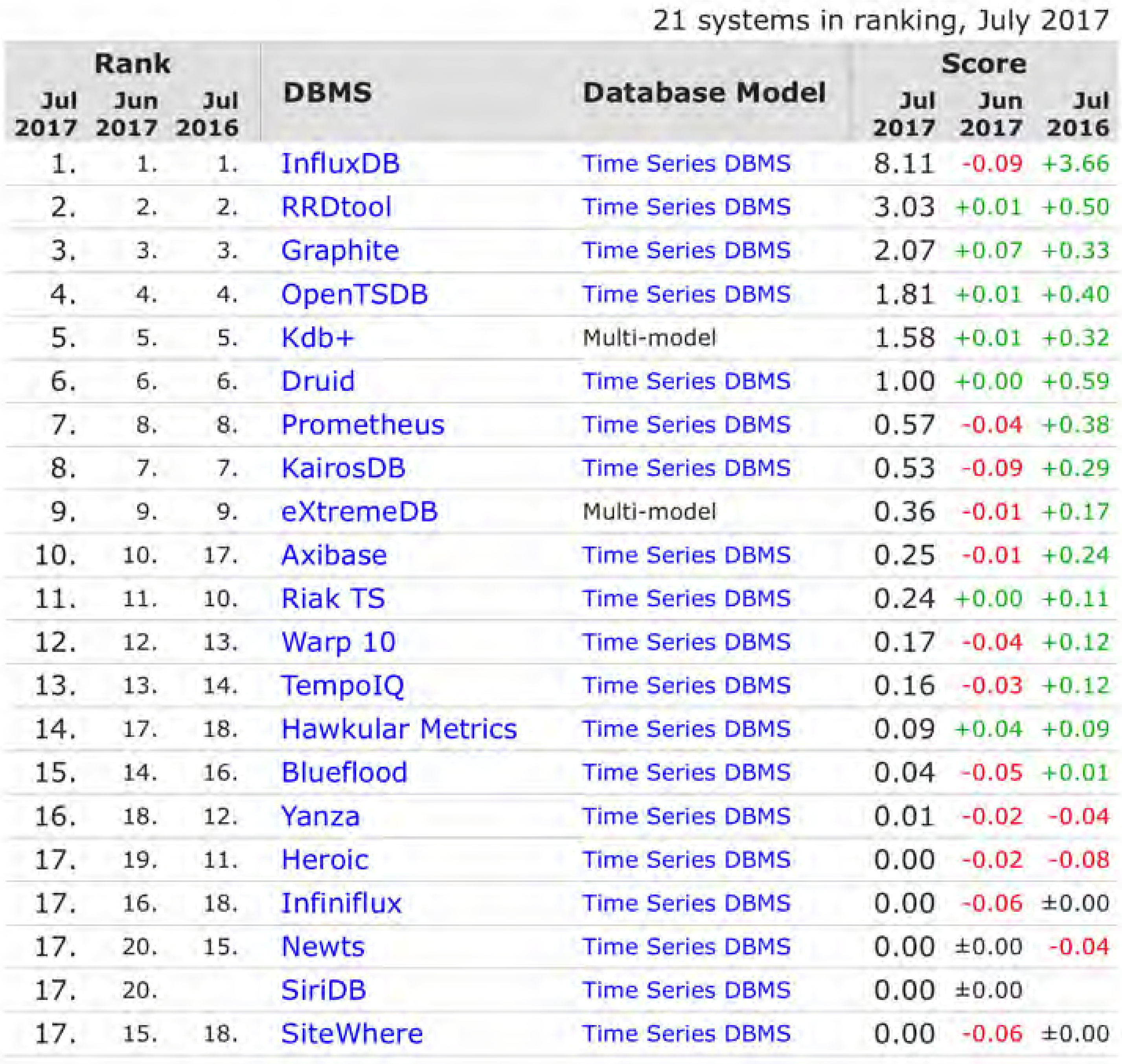
What's Next in DevOps



Architecture of Collecting Metrics


System
Network
Database
Hadoop
Docker
...
Agent for Collecting and Reporting Metrics and Events
Kafka
Hbase
OpenTSDB
Time Series Database
ETL, Alerting, Monitoring & Anomaly Detection
Grafana
Data Visualization & Exploration

Ranking of TSDB



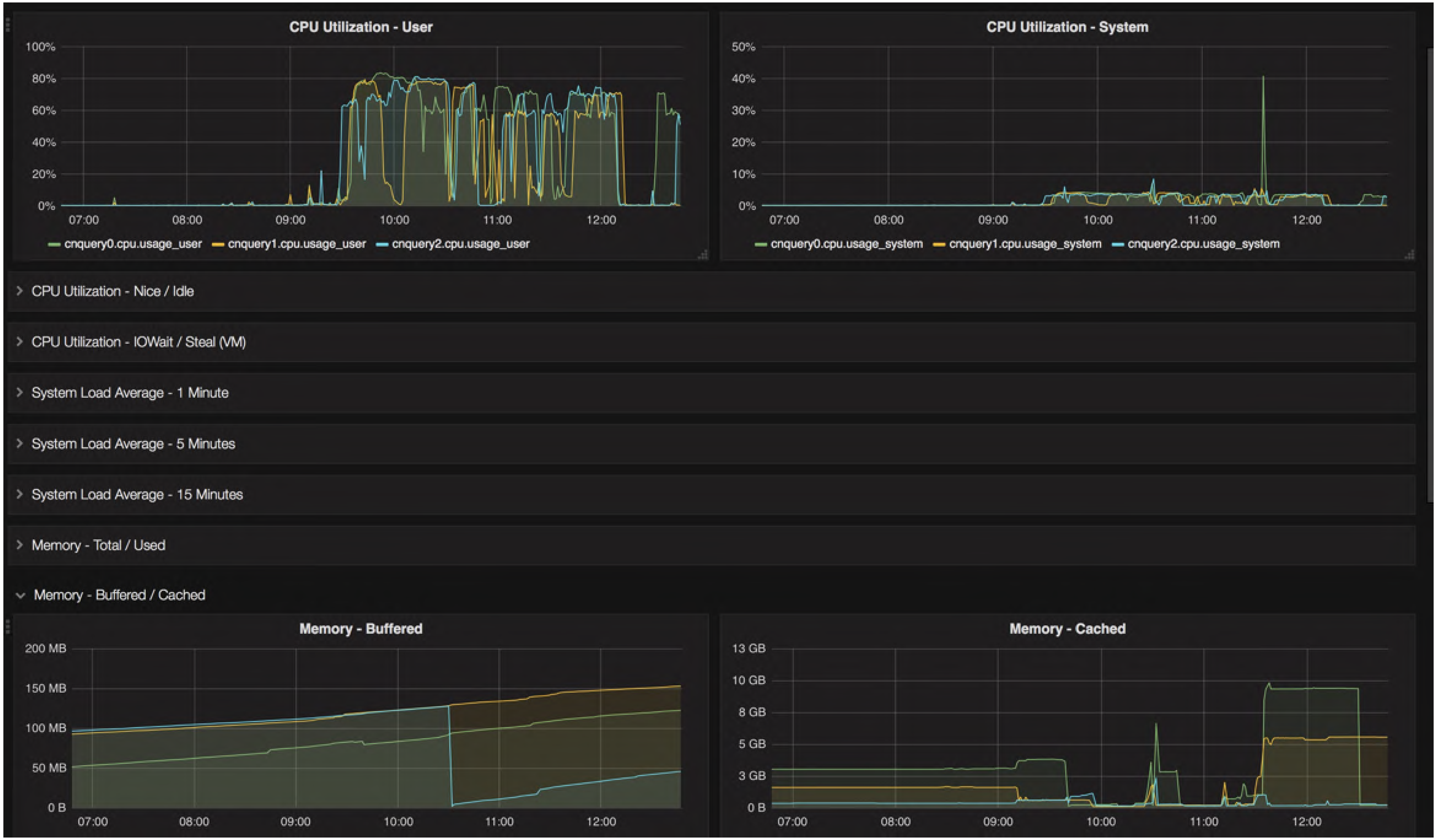
21 systems in ranking, July 2017

Rank			DBMS	Database Model	Score		
Jul 2017	Jun 2017	Jul 2016			Jul 2017	Jun 2017	Jul 2016
1.	1.	1.	InfluxDB	Time Series DBMS	8.11	-0.09	+3.66
2.	2.	2.	RRDtool	Time Series DBMS	3.03	+0.01	+0.50
3.	3.	3.	Graphite	Time Series DBMS	2.07	+0.07	+0.33
4.	4.	4.	OpenTSDB	Time Series DBMS	1.81	+0.01	+0.40
5.	5.	5.	Kdb+	Multi-model	1.58	+0.01	+0.32
6.	6.	6.	Druid	Time Series DBMS	1.00	+0.00	+0.59
7.	8.	8.	Prometheus	Time Series DBMS	0.57	-0.04	+0.38
8.	7.	7.	KairosDB	Time Series DBMS	0.53	-0.09	+0.29
9.	9.	9.	eXtremeDB	Multi-model	0.36	-0.01	+0.17
10.	10.	17.	Axibase	Time Series DBMS	0.25	-0.01	+0.24
11.	11.	10.	Riak TS	Time Series DBMS	0.24	+0.00	+0.11
12.	12.	13.	Warp 10	Time Series DBMS	0.17	-0.04	+0.12
13.	13.	14.	TempoIQ	Time Series DBMS	0.16	-0.03	+0.12
14.	17.	18.	Hawkular Metrics	Time Series DBMS	0.09	+0.04	+0.09
15.	14.	16.	Blueflood	Time Series DBMS	0.04	-0.05	+0.01
16.	18.	12.	Yanza	Time Series DBMS	0.01	-0.02	-0.04
17.	19.	11.	Heroic	Time Series DBMS	0.00	-0.02	-0.08
17.	16.	18.	Infiniflux	Time Series DBMS	0.00	-0.06	±0.00
17.	20.	15.	Newts	Time Series DBMS	0.00	±0.00	-0.04
17.	20.		SiriDB	Time Series DBMS	0.00	±0.00	
17.	15.	18.	SiteWhere	Time Series DBMS	0.00	-0.06	±0.00

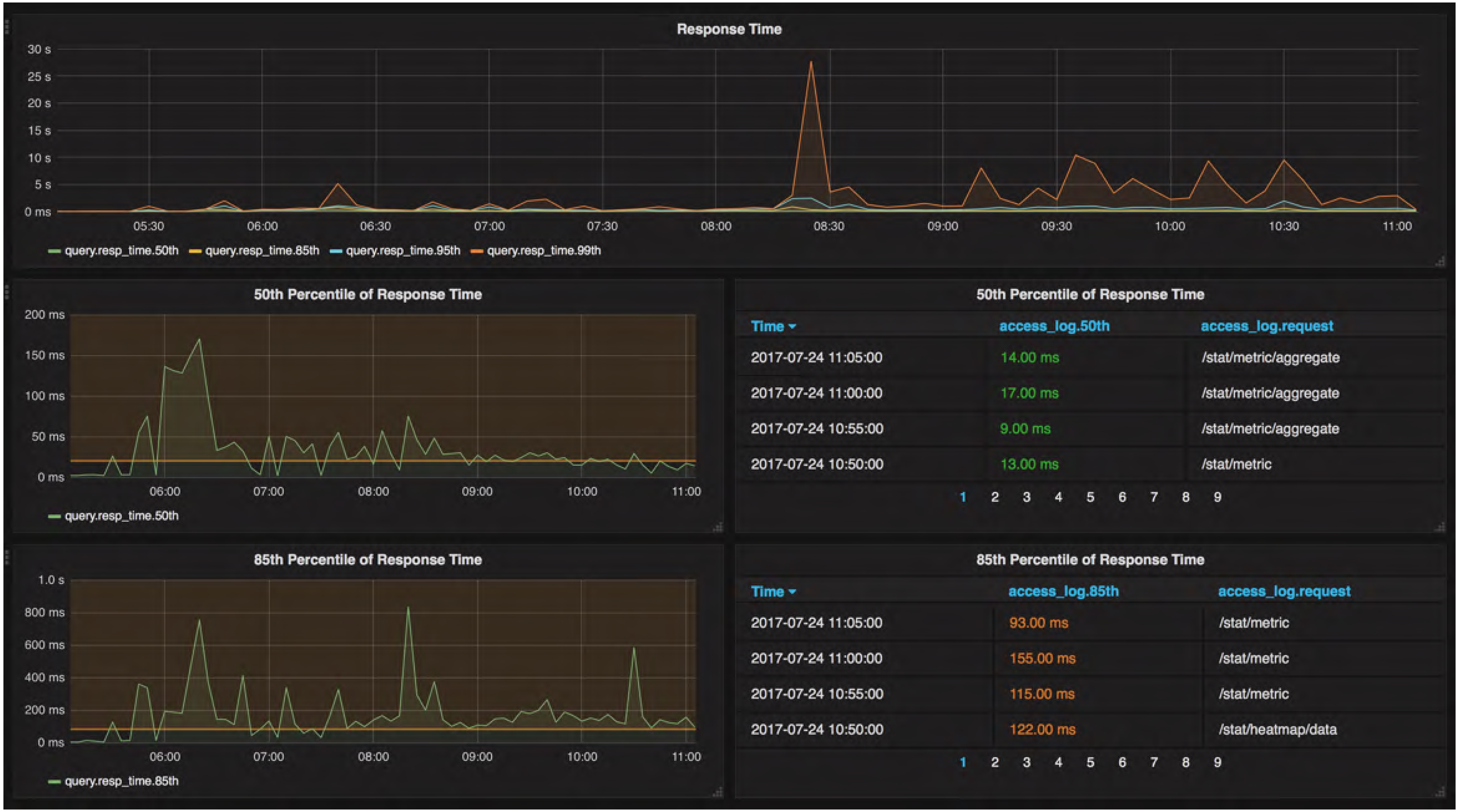
TSDB

- Timestamp
- Metric
- Value
- Tags…

Basic Metrics



Performance KPI



50th Percentile of Response Time

Time ▾	access_log.50th	access_log.request
2017-07-24 11:05:00	14.00 ms	/stat/metric/aggregate
2017-07-24 11:00:00	17.00 ms	/stat/metric/aggregate
2017-07-24 10:55:00	9.00 ms	/stat/metric/aggregate
2017-07-24 10:50:00	13.00 ms	/stat/metric

85th Percentile of Response Time

Time ▾	access_log.85th	access_log.request
2017-07-24 11:05:00	93.00 ms	/stat/metric
2017-07-24 11:00:00	155.00 ms	/stat/metric
2017-07-24 10:55:00	115.00 ms	/stat/metric
2017-07-24 10:50:00	122.00 ms	/stat/heatmap/data

Anomaly Detection

- Learn - To start the learning process simply send your metrics to Anomaly.
- Detect - Using all the patterns learnt from your metrics it’s possible to detect unusual behaviors.
- Improve - If you’re maintaining your system it will be detected as an anomaly. But is it really an anomaly?

We are hiring!

- Please do not hesitate to contact me!
- shuowang@growingio.com