

Архитектура вычислительных систем

Мониторинг инфраструктуры и приложений

Романюта Алексей Андреевич

alexey-r.98@yandex.ru

Кафедра вычислительных систем
Сибирский государственный университет телекоммуникаций и информатики



Мониторинг

- Что мониторить?
- Как мониторить?
- Чем мониторить?
- А как реагировать?

Мониторинг

За какими показателями следить?

- Загрузка системы – cpu, mem, disk io, network activity
- Статистика приложения – http request count, http code stats, database connections, database query time, etc...

Мониторинг

Что такое timeseries/временной ряд?

- Временной ряд (time series) — это признак, значения которого измеряются через *постоянные* временные интервалы.
- К таким данным относятся, например, цены на акции, объёмы продаж чего-либо, изменения температуры с течением времени и т.д.
Нас интересуют метрики серверов/вычислительных систем/приложений — загрузка процессора, утилизация памяти, количество обработанных запросов, etc...

Мониторинг

Что такое метрика?

- Метрика в мониторинге — это показатель использования ресурсов/отклонения/некоторого состояния, который можно собирать в аппаратной или программной среде

```
# HELP diskio_read_bytes Telegraf collected metric
# TYPE diskio_read_bytes counter
diskio_read_bytes{dc="dc",host="myhost",name="dm-0"} 1.170207744e+10
diskio_read_bytes{dc="dc",host="myhost",name="loop0"} 827392
diskio_read_bytes{dc="dc",host="myhost",name="loop1"} 2.6624e+06
diskio_read_bytes{dc="dc",host="myhost",name="loop2"} 2.275328e+06
diskio_read_bytes{dc="dc",host="myhost",name="loop3"} 2.332672e+06
diskio_read_bytes{dc="dc",host="myhost",name="loop4"} 833536
diskio_read_bytes{dc="dc",host="myhost",name="loop5"} 2.128896e+06
diskio_read_bytes{dc="dc",host="myhost",name="loop6"} 14336
diskio_read_bytes{dc="dc",host="myhost",name="sda"} 1.19168e+10
diskio_read_bytes{dc="dc",host="myhost",name="sda1"} 1.6514048e+07
diskio_read_bytes{dc="dc",host="myhost",name="sda2"} 4.252672e+07
diskio_read_bytes{dc="dc",host="myhost",name="sda3"} 1.1828386816e+10
```

Типы метрик

Gauge – Просто значение в момент времени

Counter – постоянно растущий счетчик

Histogram – подсчитывает метрики в настраиваемых сегментах

Summaries – квантили и тп. Больше для сводок и обзорных этапов

Мониторинг

Договоренности и зачем это нужно

- SLO (Service Level Objective) — декларируемый уровень надёжности сервиса. Это желаемая цель, тот уровень качества, который хочет видеть наш клиент.
- К примеру, хотим обеспечить:
 - время ответа не больше 100 мс на 95 %;
 - количество ответов с кодом 500 не больше 0,2% от общего числа ответов сервера;
 - время обработки критического тикета не больше 20 минут.
- Эти три метрики и будут SLO.

Мониторинг

Договоренности и зачем это нужно

- SLI (Service Level Indicator) — фактическое состояние метрик. Это то, как работает сервис в реальности.
- Мы измеряем метрики и узнаём, что наши SLI, например такие:
 - время ответа — 730 мс;
 - 0,01% ответов с кодом 500;
 - 22 минуты на обработку критического тикета.

Мониторинг

Договоренности и зачем это нужно

- SLA (Service Level Agreement) — договорённости или контракты о том, что делать, если SLI отличаются от SLO в худшую сторону. То есть, если сервис работает не так, как обещали.
- В контракте SLA могут быть прописаны штрафные санкции. Это может быть возврат денег, дополнительные услуги, особый приоритет в будущем.

Мониторинг

Как собирать метрики?

- SNMP – протокол управления сетевыми устройствами (snmpwalk, snmpget).
- Загрузка системы – procfs, собираем bash скриптами например
- Статистика приложения
 - Встроенный http server, отдающий метрики при обращении(Prometheus like)
 - Прямая запись в базу данных метрик (influx, sql, etc...)

The screenshot shows the MIB Browser application interface. At the top, there is a header bar with the title 'MIB Browser'. Below it, there are input fields for 'Address' (10.10.10.1) and 'OID' (1.3.6.1.2.1.2.2.1.2), along with an 'Advanced...' button and an 'Operations' dropdown set to 'Get' with a 'GO' button. The main area is divided into two panes. The left pane, titled 'SNMP MIBs', shows a hierarchical tree view of MIBs. The right pane, titled 'Result Table', displays a table with three columns: 'Name/OID', 'Value', and 'Type'. The table contains three rows of data. Below the table, there are fields for 'Name', 'OID', 'Syntax', 'Access', and 'Description'.

Name/OID	Value	Type
1.3.6.1.2.1.2.2.1.2.1 (.iso.org.dod.interne...)	Vlan1	OctetString
1.3.6.1.2.1.2.2.1.2.2 (.iso.org.dod.interne...)	FastEthernet0/0	OctetString
1.3.6.1.2.1.2.2.1.2.3 (.iso.org.dod.interne...)	FastEthernet0/1	OctetString

Name :	.ifDescr
OID :	1.3.6.1.2.1.2.2.1.2
Syntax :	
Access :	
Description :	

Мониторинг

Чем собирать метрики?

- Существует несколько стеков приложений
 - Prometheus stack – Exporters(cadvisor, node exporter, etc...) , Prometheus, alertmanager
 - Influx – influxdb, telegraf, kapacitor
 - Можно комбинировать – telegraf + Prometheus или же использовать для хранения метрик Prometheus-like системы, такие как VictoriaMetrics.
- Визуализация:
 - Grafana – самый популярный инструмент визуализации метрик
- Отдельно существует стек, предназначенный для сбора логов
 - Elasticsearch, logstash/filebeat/fluentd, kibana
 - Opensearch, logstash/filebeat, os-dashboards
 - LOKI (Prometheus-like config)

Мониторинг

Чем собирать метрики?

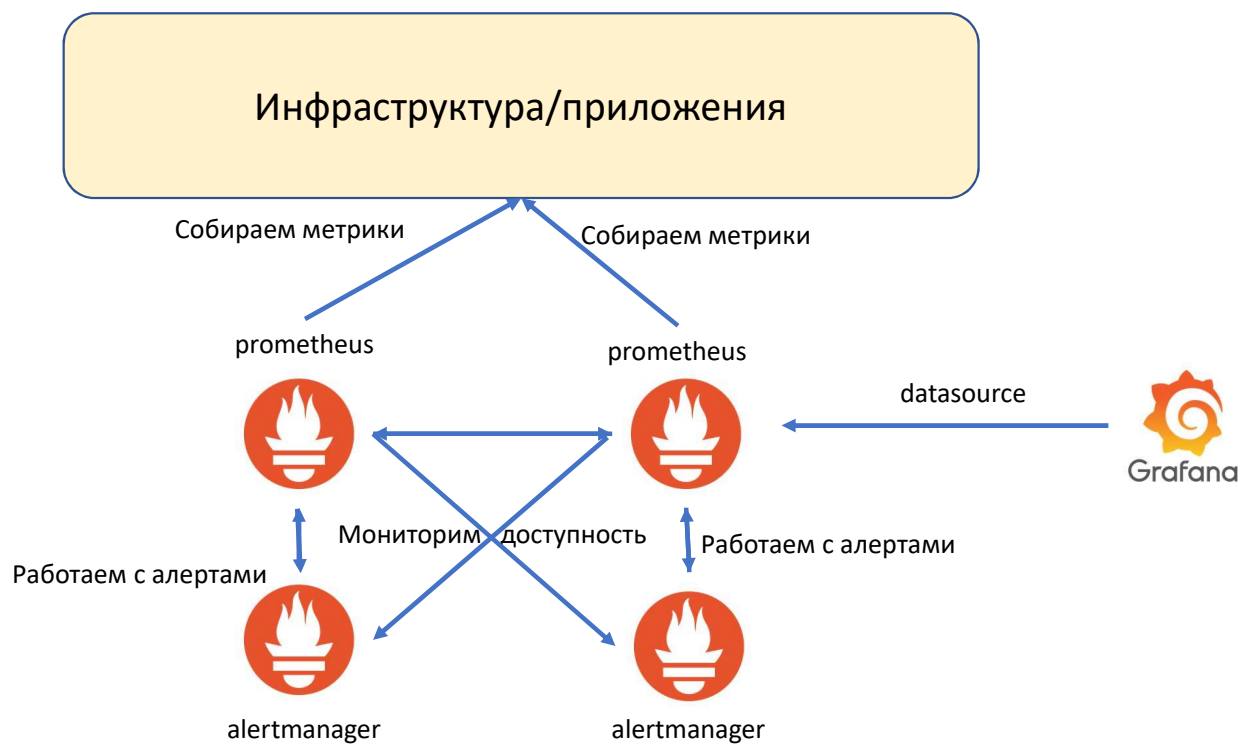
- Prometheus и его формат метрик стал практически стандартом в мониторинге
- Есть docker образы
- Есть kube-prometheus-stack helm chart
- Victoria metrics позволяет добавить отказоустойчивости в мониторинг.
Совместимо с метриками prometheus, поэтому переход как правило почти бесшовный
- Возможно хранить метрики в объектных хранилищах (s3)

Victoria Metrics <https://docs.victoriametrics.com/>

Kube Prometheus stack for k8s - <https://github.com/prometheus-community/helm-charts/tree/main/charts/kube-prometheus-stack/>

Мониторинг

Отказоустойчивость мониторинга?

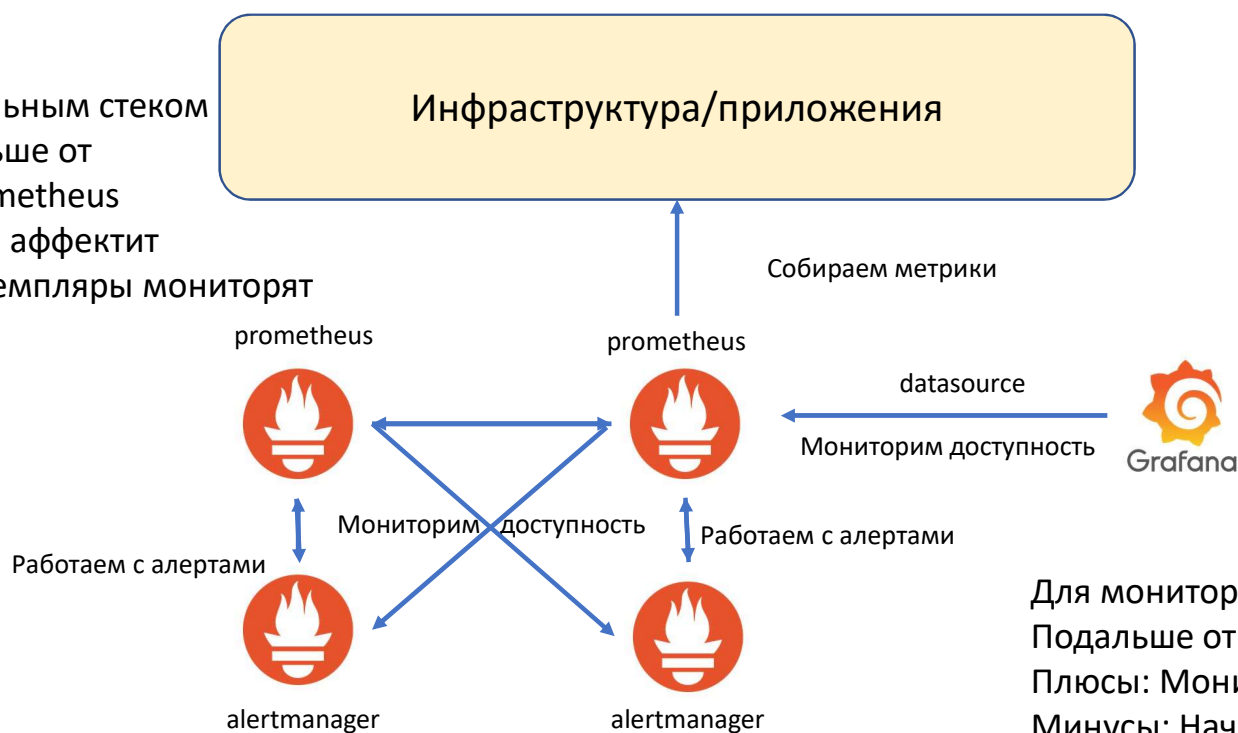


Мониторинг

Отказоустойчивость мониторинга?

Для мониторинга отдельным стеком он должен быть подальше от сервера основного Prometheus

Плюсы: Мониторим, не аффектит основную систему, экземпляры мониторят друг друга



Для мониторинга графаной она должна быть подальше от сервера Prometheus
Плюсы: Мониторим
Минусы: Начинает тупить построение графиков

Мониторинг

А как реагировать? - **Алертинг**

- Задача алертинга – прислать сообщение, если что-то пошло не так
- Настраивается набором правил – например
$$\text{sum}(\text{http_request}\{\text{code}="5xx"\}[5\text{m}]) > 10$$
Сумма ошибок с кодом > 500 за последние 5 минут больше 10
- Варианты реализации
 - Дежурный инженер – постоянно смотрит в графики
 - Специализированное ПО – делает тоже самое, но без участия инженера
- Проблемы
 - Как правило, правило алертинга появляется после того, как всё развалилось
- ПО: Grafana/alertmanager, kapacitor, moira, etc...

Сработавший алерт должен делать как-то так



Мониторинг: Prometheus

Targets

All scrape pools ▾

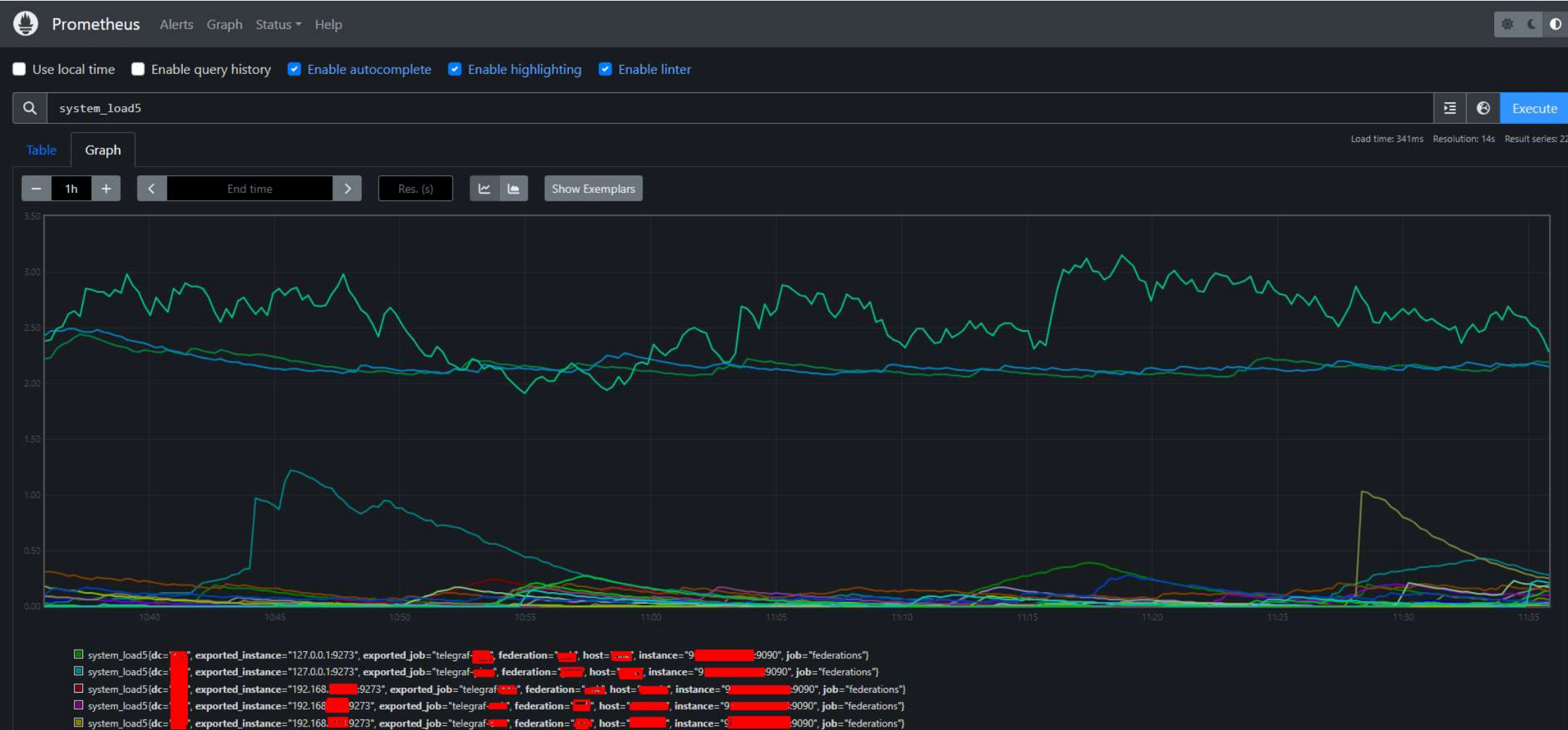
All Unhealthy Collapse All

🔍 Filter by endpoint or labels

federations (2/2 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://[REDACTED]:9090/federate match[]={job=~".+"}	UP	instance="[REDACTED]:9090" job="federations" ▾	28.000ms ago	54.112ms	
http://[REDACTED]:9090/federate match[]={job=~".+"}	UP	instance="[REDACTED]:9090" job="federations" ▾	5.433s ago	116.594ms	

Мониторинг: Prometheus



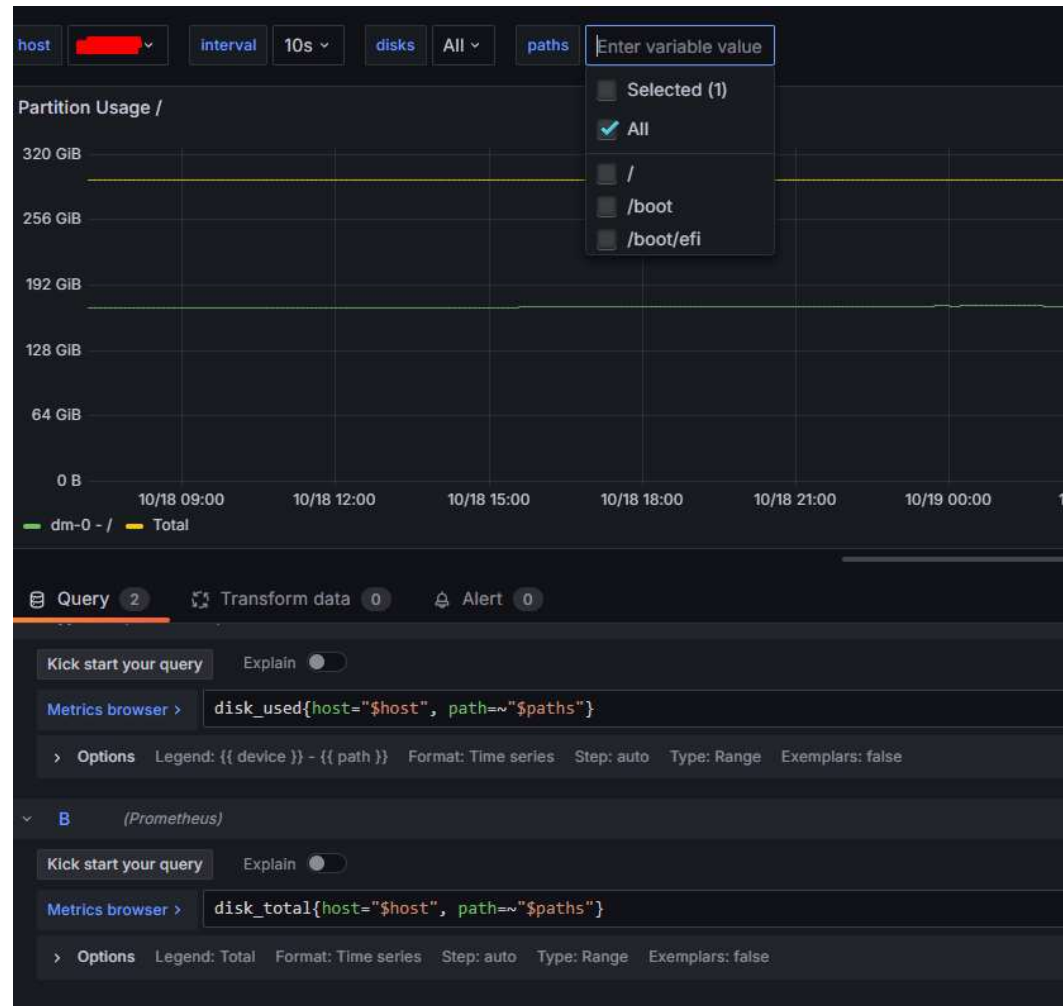
Мониторинг: Grafana



Мониторинг: Grafana



Мониторинг: Grafana



Live section

Романюта Алексей Андреевич

alexey-r.98@yandex.ru

Кафедра вычислительных систем
Сибирский государственный университет телекоммуникаций и информатики

