



**Не ждите сообщений,
получайте их
Асинхронное тестирование
производительности**

**Смирнов Вячеслав
Райффайзенбанк**

Ускоряю Raiffeisen Business Platform



**Сколько ресурсов
нужно, чтобы подать
нагрузку на JMS или AMQP
с Gatling (или JMeter)?**

**Минимум ресурсов
нужно, чтобы подать
нагрузку на JMS или AMQP
с Gatling (Send).
Единицы потоков**

Очень много ресурсов
нужно, чтобы подать
нагрузку на **JMS** или **AMQP**
с Gatling (**Request/Reply**).
Десятки тысяч потоков!

Используя

- **подход SharedHashMap**
 - **AsyncLog-плагин**
 - **получение без ожидания**
- сделаем тесты ЛЁГКИМИ**

1 Очереди

2 JMeter и Gatling

3 Проблемы и задачи

4 Подходы к решению

5 Нарработки

1 Очереди

2 JMeter и Gatling

3 Проблемы и задачи

4 Подходы к решению

5 Наработки

1 Очереди

Школа Highload.

Урок № 4. Масштабирование во
времени

(Хакер № 163, 2012 год, стр. 108-112)

1 Очереди

1.1 Асинхронная обработка

1.2 Проверка на дубликаты

1.3 Очередь как шина

1.4 Надежность узлов

Очередь как шаблон

Может быть в DB и в MQ

- MySQL, PostgreSQL, ...
- IBM.MQ
- RabbitMQ
- ActiveMQ

Очередь как шаблон

Может быть в DB и в MQ



1.1 Асинхронная обработка

Собрать данные сейчас, а обработать потом:

- пересчитать статистику
- обновить индексы в базе данных
- сформировать выписки по счёту

1.1 Асинхронная обработка

**Ответ может прийти
нескоро**

- **десятки секунд**
- **минуты**
- **часы**

1.2 Проверка на дубликаты

На 10 запросов 1 ответ



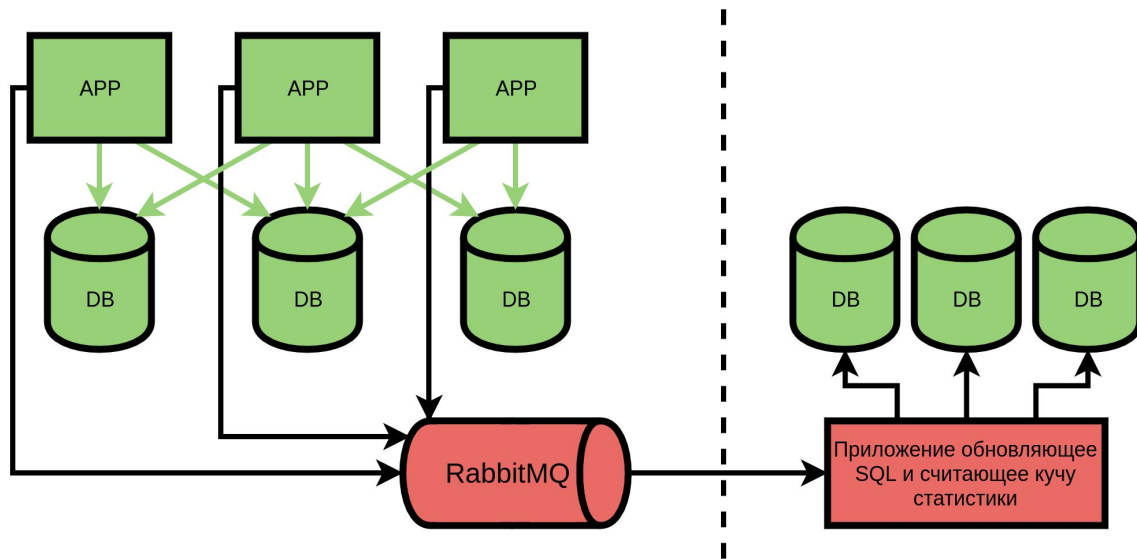
1.2 Проверка на дубликаты

На 10 запросов 1 ответ

- **удаление дублей**
- **накопление батчей**
- **периодические ответы**

1.3 Очередь как шина

Ответ не придёт (отправителю)



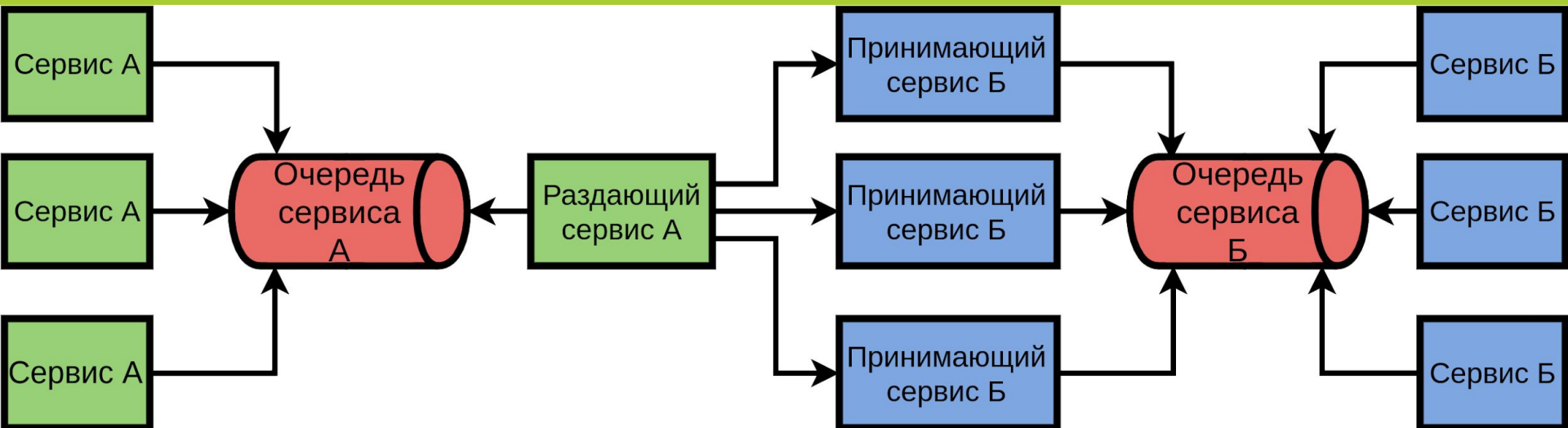
1.3 Очередь как шина

Ответ не придёт (отправителю)

- ответ в другой системе
- по другому протоколу
- его вообще нет

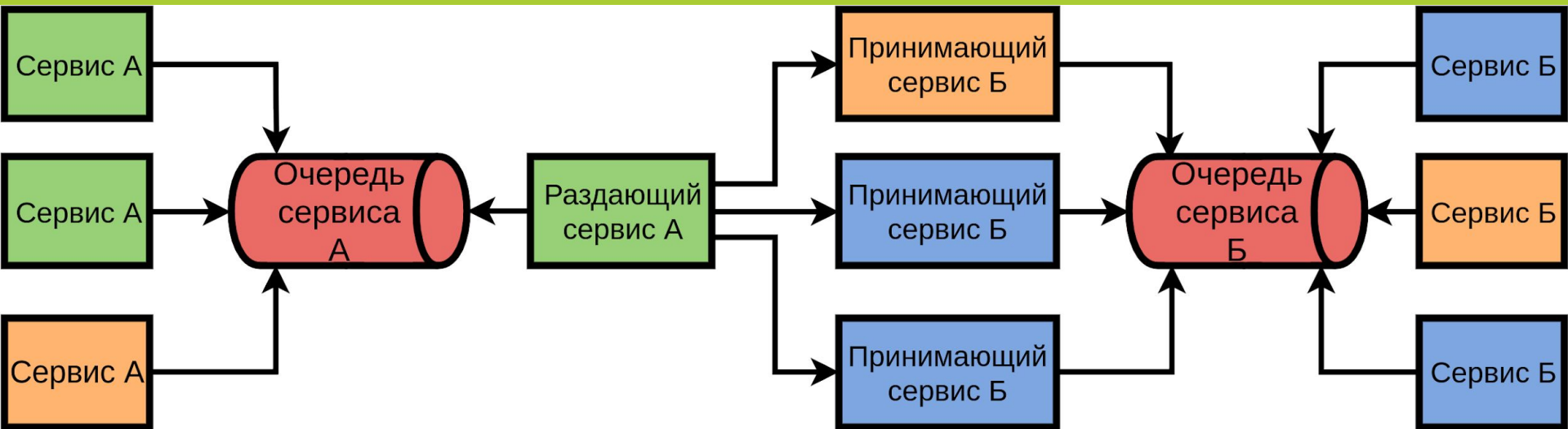
1.4 Надежность узлов

Шина защищает от потерь



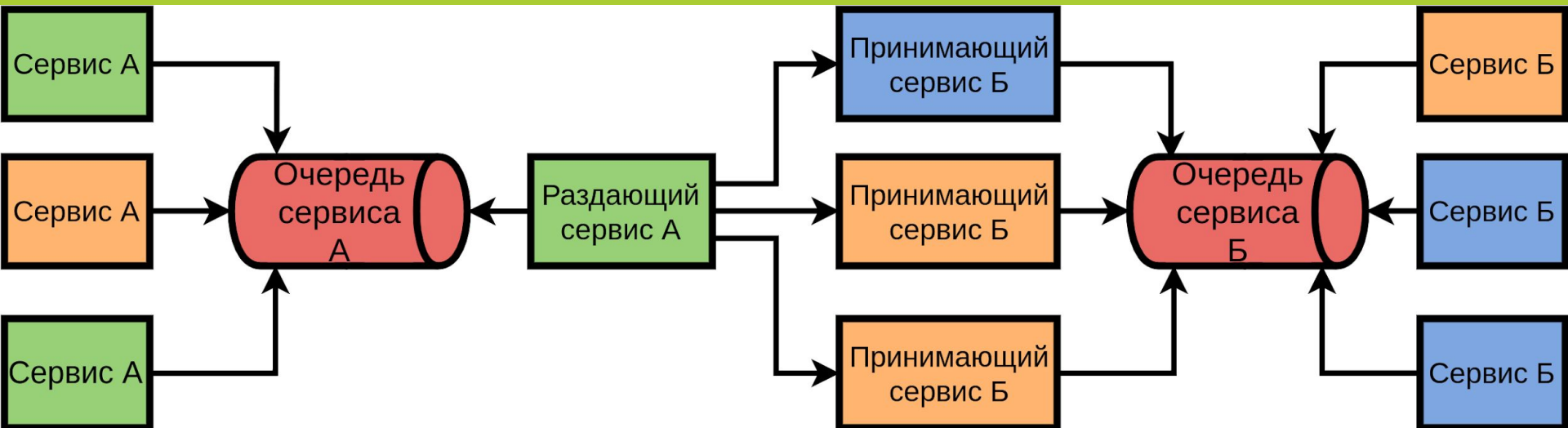
1.4 Надежность узлов

Шина защищает от потерь



1.4 Надежность узлов

Шина защищает от потерь



Очереди в тестировании

Нужно дожидаться ответа

- не быстрого
- не гарантированного
- даже по иному протоколу

1 Очереди

2 JMeter и Gatling

3 Проблемы и задачи

4 Подходы решения

5 Наработки

2 JMeter и Gatling

2.1 Работа с JMS

2.2 Работа с AMQP

2.3 Отправка запросов

2.4 Ожидание ответов

Работа с JMS в JMeter

Sampler	Операции
Point-to-point	send, get, browse, reqreply, clear
JMS Publisher	send
JMS Subscriber	get

JMeter Point-to-point (send)

JMS Point-to-Point

Name: JMS Point-to-Point: send request \${queueNameOut}

Comments:

JMS Resources

QueueConnection Factory \${queueConnectionFactory}

JNDI name Request queue \${queueNameOut}

JNDI name Receive queue

Number of samples to aggregate

JMS Selector

Message properties

Communication style request_only

Use alternate fields for message correlation

Use Request Message Id Use Response Message Id

Timeout (ms)

Expiration (ms) 100000

Priority (0-9) 4

Use non-persistent d

Content

```
1 <?xml version="1.0" encoding="UTF-8"?>
```

```
2 <output>${ims body}</output>
```

JMeter Point-to-point (read)

JMS Point-to-Point

Name: JMS Point-to-Point: read \${queueNameIn}

Comments:

JMS Resources

QueueConnection Factory \${queueConnectionFactory}

JNDI name Request queue \${queueNameIn}

JNDI name Receive queue

Number of samples to aggregate

JMS Selector

Message properties

Communication style read

Use alternate fields for message correlation

Use Request Message Id Use Response Message Id

Timeout (ms) \${readTimeOut}

Expiration (ms) 0

Priority (0-9) 4

Use non-persistent

Content

JMeter JMS Publisher (send)

JMS Publisher

Name: JMS Publisher: \${queueNameOut}

Comments:

Use jndi.properties file

Initial Context Factory \${initialContextFactory}

Provider URL \${providerURL}

Connection Factory \${queueConnectionFactory}

Destination \${queueNameOut} Setup At startup Each sample Use non-persistent destination

Use Authorization? User \${user}

Password

Expiration (ms) 100000

Priority (0-9) 4

Reconnect on error codes (regex)

Number of samples to aggregate 1

JMS Properties

Name:	Value	Class of value
JMSCorrelationId	\${JMSCorrelationId}	java.lang.String

Add

Delete

JMeter JMS Subscriber (get)

JMS Subscriber

Name: JMS Subscriber: \${queueNameIn}

Comments:

Use jndi.properties file

Initial Context Factory \${initialContextFactory}

Provider URL \${providerURL}

Connection Factory \${queueConnectionFactory}

Destination \${queueNameIn}

Setup • At startup • Each sample

Durable Subscription ID

Client ID

JMS Selector

Use Authorization?

User \${user}

Password

Number of samples to aggregate 1

Store Response

Работа с JMS в Gatling

Плагин	Операции
JMS Plugin	reqreply, send

Работа с JMS в Gatling

JMS

Define the JMS requests sent in your scenario

Start

```
jms
```

Commons

```
requestReply
```

```
send
```

```
queue
```

```
destination
```

```
replyQueue
```

Defines a target destination

```
queue(name: Expression[String])
```

Where name is the name of the queue

Работа с JMS в Gatling

JMS

Define th

Start

jms

Comm

requestReply

send

queue

destination

replyQueue

GO TO THE CHEAT-SHEET!

Defines a target destination

```
queue(name: Expression[String])
```

Where name is the name of the queue

Работа с JMS в Gatling

Просто отправка запроса

```
.exec(jms("send")  
  .send  
  .queue("queueInputName")  
  .textMessage("text")  
)
```

Работа с JMS в Gatling

Отправка и ожидание ответа

```
.exec(jms("requestReply")  
  .requestReply  
  .queue("queueInputName")  
  .replyQueue("queueOutputName")  
  .textMessage("text")  
)
```

2 JMeter и Gatling

2.1 Работа с JMS

2.2 Работа с AMQP

2.3 Отправка запросов

2.4 Ожидание ответов

**Чем подавать нагрузку на
RabbitMQ в 2019 году?**

Может Apache.JMeter 5.1+?

Может Gatling 3.0+?

JMeter AMQP (первый)

[jvallee/JMeter-Rabbit-AMQP](#) (105 ★)

Last commit	2016-05-17	три года назад
JMeter	2.11	2014 , работает в 5.1
Amqp	3.5.1	2015 , обновляется до 4.11
Issues	21	2015 , критично heartbeat
Pull Requests	9	2017 , удобно maven
Fork	109	много, но активных 11

JMeter AMQP (первый)



[jvallee/JMeter-Rabbit-AMQP](#) (105 ★)

Last commit	2016-05-17	три года назад
JMeter	2.11	2014 , работает в 5.1
Amqp	3.5.1	2015 , обновляется до 4.11
Issues	21	2015 , критично heartbeat (1 сек)
Pull Requests	9	2017 , удобно maven
Fork	109	много, но активных 11

AMQP heartbeat (1 cek)



AMQP Consumer

AMQP Consumer

AMQP Consumer

AMQP Consumer

AMQP Consumer

AMQP Consumer


AMQP Consumer



Форки JMeter AMQP

[polarnik/JMeter-Rabbit-AMQP-info](https://github.com/polarnik/JMeter-Rabbit-AMQP-info) (top 11 forks)

From	Last commit	Fork from / Commits	JMeter / Amqp	Change
jvallee	2016-05-17		2.11 / 3.5.1	
zeph1rus	2018-10-24	74 / 85	5.0 / 4.8.3	Added options to manage <u>heartbeat</u>
aliesbelik	2018-09-26	70 / 80	4.0 / 5.4.1	Maven, amqp v5.4.1, default content type: text/plain utf-8, Add support for AMQP queue (x-max-priority) & message priority
sergiogouveia	2017-10-26	70 / 79	2.13 / 3.6.0	Maven, x-dead-letter-exchange, Response as JSON, added gzipped consumed messages decode



JMeter AMQP (zeph1rus)

AMQP Publisher

Name: AMQP Publisher

Comments:

Exchange

Exchange jmeterExchange

Exchange Type direct

Durable?

Redeclare?

Auto Delete?

Queue

Queue jmeterQueue

Routing Key jmeterRoutingKey

Durable?

Redeclare?

Message TTL

Exclusive

Expires

Auto Delete?

Connection

Virtual Host /

Host localhost

Port 5672

SSL?

Username guest

Password guest

Timeout (s) 1000

Heartbeat 60

Number of samples to Aggregate 1

Persistent?

Use Transactions?

Routing Key

Message Type

Reply-To Queue

Correlation Id

ContentType

Message Id

JMeter AMQP Publisher

AMQP Publisher

Name:

Comments:

Exchange

Exchange

Exchange Type

Durable?

Redeclare?

Auto Delete?

Queue

Queue

Routing Key

Durable?

Redeclare?

Message TTL

Exclusive

Expires

Auto Delete?

Connection

Virtual Host

Host

Port

SSL?

Username

Password

Timeout (0 for In

Heartbeat

JMeter AMQP Consumer

AMQP Consumer

Name:

Comments:

Exchange

Exchange

Exchange Type

Durable?

Redeclare?

Auto Delete?

Queue

Queue

Routing Key

Durable?

Redeclare?

Message TTL

Exclusive

Expires

Auto Delete?

Connection

Virtual Host /

Host

Port

SSL?

Username

Password

Timeout (0 for

Heartbeat

Gatling AMQP (первый)

[maiha/gatling-amqp](#) (14 ★)

Last commit	2016-09-02	три года назад
Gatling	2.2.0-M3	2015 , не работает в 3.0+
Amqp	4.9.0	2015 , обновляется до 4.11
Issues	2	2018 , критично gatling-2.2.2
Pull Requests	2	2018
Fork	25	три активных, для gatling-2

Варианты подачи нагрузки

- использовать JMeter
- попробовать Gatling 2.1
- написать свой плагин
- подождать сообщество

Варианты подачи нагрузки

- использовать JMeter
- попробовать Gatling 2.1
- написать свой плагин
- подождать сообщество

Gatling для AMQP (сейчас)

Написали свой плагин

Возможно будет тут:

github.com/Raiffeisen-DGTL

Gatling для AMQP (сейчас)

[TinkoffCreditSystems/gatling-amqp-plugin](https://github.com/TinkoffCreditSystems/gatling-amqp-plugin) (14 ★)

Last commit	2019-10-06	НОВЫЙ
Gatling	3.2.0	2019 , работает в 3.0+
Amqp	5.7.1	2019 , НОВЫЙ
Issues	2	2019 , нет Consume/Get
Pull Requests	1	2019 , развивается
Fork	5	один активный

2 JMeter и Gatling

2.1 Работа с JMS

2.2 Работа с AMQP

2.3 Отправка запросов

2.4 Ожидание ответов

Req/Reply, Send, Get, Clear

		R/R	Send	Get	Clear
JMeter	JMS Publisher, Subscriber		есть	есть	
JMeter	JMS Point-to-point	есть	есть	есть	есть
JMeter	AMQP Publisher, Consumer		есть	есть	есть
Gatling	JMS Plugin	есть*	есть		
Gatling	AMQP Plugin	есть*	есть		

* Удобные селекторы в Gatling

2 JMeter и Gatling

2.1 Работа с JMS

2.2 Работа с AMQP

2.3 Отправка запросов

2.4 Ожидание ответов

1.3 Отправка запросов

Есть во всех протоколах

Send (Publish, Request):

- **AMQP, JMS**
- **...**

1.3 Отправка запросов

Есть во всех протоколах

Send (Publish, Request):

- **AMQP, JMS**
- **MQTT, JDBC, ...**

1.3 Отправка запросов

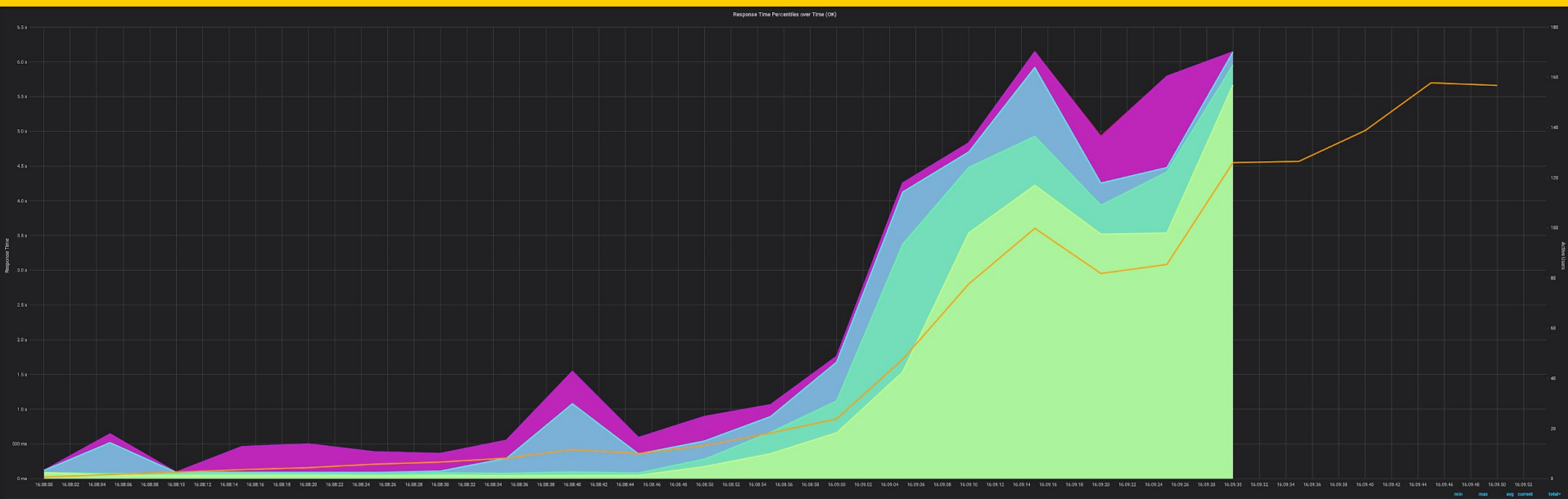
Часто её одной достаточно

**Не ждите сообщений,
отправляйте их**



Просто отправлять запросы

Но как измерять время?





Как измерять время?

По Java MBean метрикам

- общая длительность
- **telegraf, influxdb, grafana, Camel**
 - `inputs.jolokia2_agent.metric`
 - `org.apache.camel:type=threadpools`

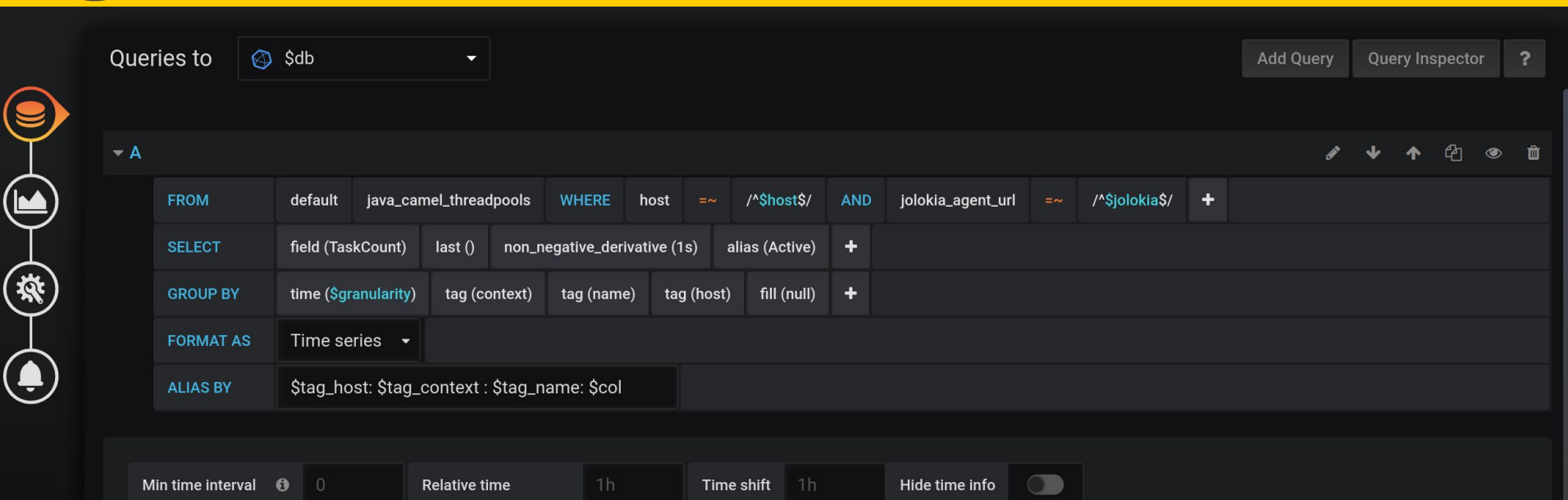
По Java MBean метрикам

1

telegraf.conf (jolokia2)

```
[[inputs.jolokia2_agent.metric]]
  name = "java_camel_threadpools"
  mbean = "org.apache.camel:type=threadpools,
          context=*,name=*"
  tag_keys = ["context", "name"]
```

По Java MBean метрикам grafana, запрос к influxdb



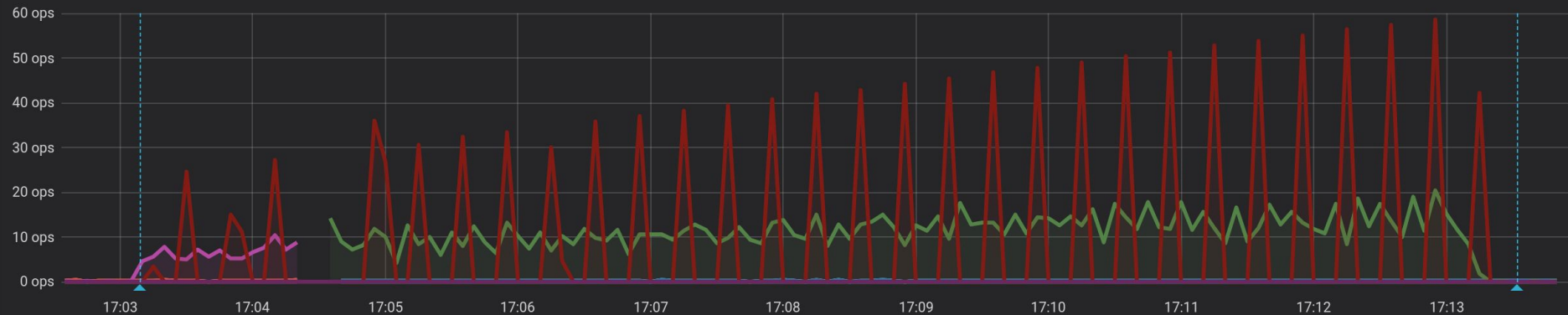
The screenshot shows the Grafana InfluxDB query editor interface. At the top, there is a dropdown menu for "Queries to" set to "\$db". To the right are buttons for "Add Query", "Query Inspector", and a help icon. On the left side, there is a vertical navigation bar with icons for Home, Dashboard, Query Editor, Settings, and Alerts. The main query editor area is titled "A" and contains the following query configuration:

Clause	Value
FROM	default java_camel_threadpools WHERE host =~ /^\$host\$/ AND jolokia_agent_url =~ /^\$jolokia\$/ +
SELECT	field (TaskCount) last () non_negative_derivative (1s) alias (Active) +
GROUP BY	time (\$granularity) tag (context) tag (name) tag (host) fill (null) +
FORMAT AS	Time series
ALIAS BY	\$tag_host: \$tag_context : \$tag_name: \$col

At the bottom of the interface, there are controls for "Min time interval" (set to 0), "Relative time" (set to 1h), "Time shift" (set to 1h), and "Hide time info" (toggle switch).

По Java MBean метрикам grafana, работа системы

Camel. Task Processing



AppServer: camel-2 : "aggregate1(aggregate)": Active

min	max	avg	current	total
0 ops	59 ops	9 ops	0 ops	1.2622K ops

AppServer: camel-2 : "JmsConsumer(0x73044cdf)": Active

0 ops	20 ops	11 ops	0 ops	1.2348K ops
-------	--------	--------	-------	-------------

AppServer: camel-2 : "JmsConsumer(0x65bcf7c2)": Active

0 ops	10 ops	5 ops	9 ops	100 ops
-------	--------	-------	-------	---------

AppServer: camel-2 : "AggregateProcessor(0x6a10b263)": Active

0 ops	0 ops	0 ops	0 ops	22 ops
-------	-------	-------	-------	--------

Как измерять время?

По логам сервиса

- общая длительность
- **telegraf, influxdb, grafana, logback**
 - `inputs.logparser.grok`
 - `processor.regex`

Настроить формат лога

logback.xml

```
1 <configuration>
2   <conversionRule conversionWord="coloredLevel" converterClass="play.api.libs.logback.ColoredLevel"/>
3
4   <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
5     <File>/opt/logs/service/application.log</File>
6     <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
7       <!-- daily rollover -->
8       <fileNamePattern>/opt/logs/service/application_%d{yyyyMMdd}_%i.log.gz</fileNamePattern>
9       <maxFileSize>100MB</maxFileSize>
10      <maxHistory>30</maxHistory>
11    </rollingPolicy>
12    <encoder>
13      <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS} --- %level --- %thread --- %logger ---
14      %message%n%rootException{full}</pattern>
15    </encoder>
16  </appender>
```

Форматную строку лога

logback.xml (pattern)

```
%date{yyyy-MM-dd HH:mm:ss.SSS}  
--- %level --- %thread --- %logger ---  
%message%n%rootException{full}
```

Мониторим лог с Telegraf 2

telegraf.conf (logparser)

```
1 [[inputs.logparser]]
2   files = ["/opt/logs/service/application.log"]
3   from_beginning = false
4   [inputs.logparser.grok]
5     patterns = ['%{TIMESTAMP_ISO8601:timestamp:ts-"2006-01-02 15:04:05.000"} --- %{DATA:}']
6     measurement = "logparser_application_log"
7     timezone = "Europe/Moscow"
8 [[processors.regex]]
9   namepass = ["logparser_application_log"]
10  [[processors.regex.fields]]
11    key = "msg"
12    pattern = ".*"
13    replacement = "1"
14
15  [[processors.regex.tags]]
```


Форматная строка парсера

telegraf.conf (logparser)

```
%{TIMESTAMP_ISO8601:timestamp:ts-"2006-01-02 15:04:05.000"} --- %{DATA:level:tag} ---  
%{DATA:thread:tag} --- %{DATA:logger:tag} ---  
%{NOTSPACE:msg:string}
```

По логам сервиса

grafana, запрос к influxdb

Queries to default

Add Query

Query Inspector

?

A

FROM default logparser_application_log WHERE host =~ /^\$host\$/ AND path =~ /^\$path\$/ AND level =~ /^\$level\$/ AND thread_group =~ /^\$thread_group\$/ +

SELECT field (msg) count () +

GROUP BY time (\$g) tag (thread_group) tag (logger) tag (level) fill (0) +

FORMAT AS Time series

ALIAS BY \$tag_level :: \$tag_thread_group :: \$tag_logger

Min time interval 0

Relative time

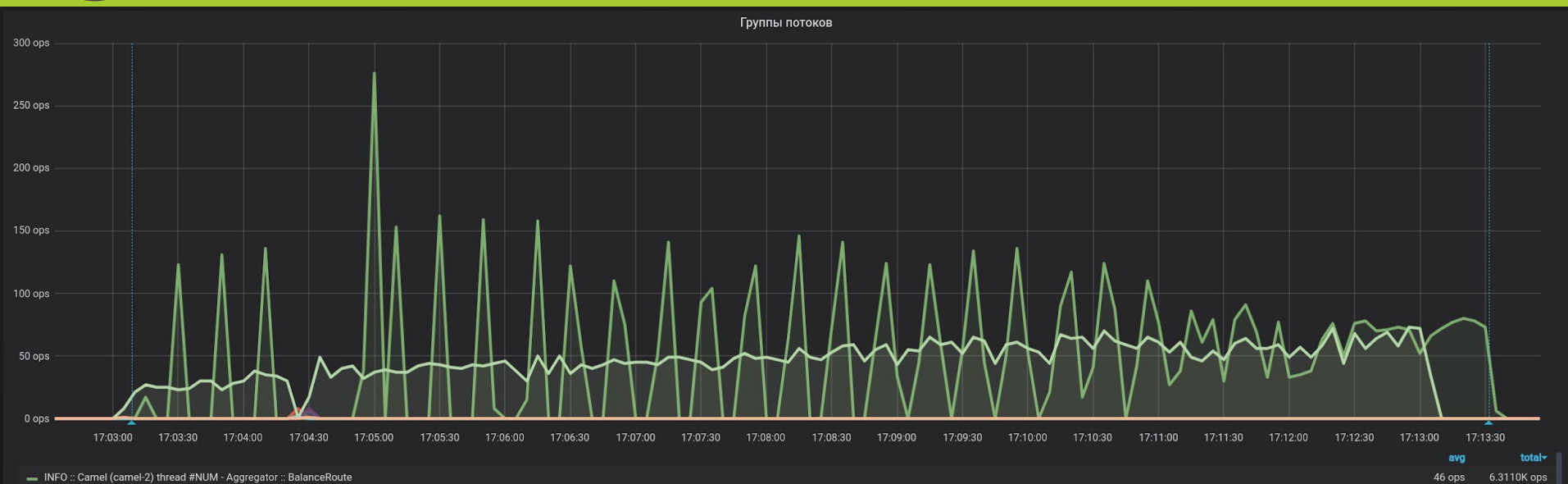
1h

Time shift 1h

Hide time info



По логам сервиса grafana, работа системы



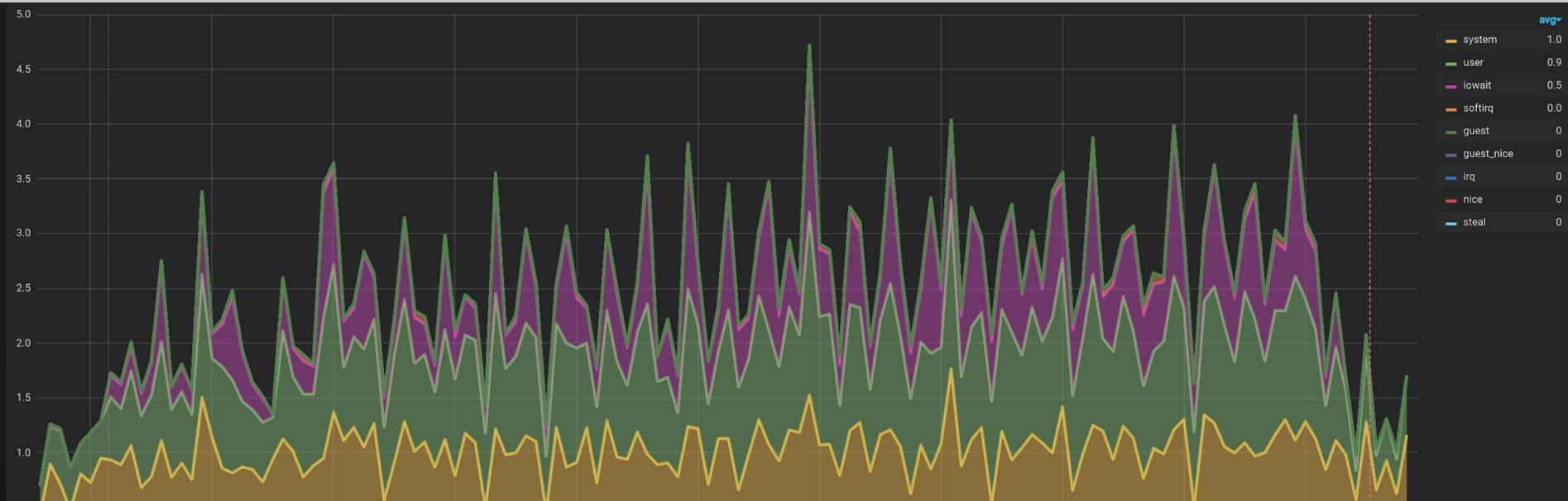
Как измерять время?

По утилизации ресурсов

- общая длительность
- **telegraf, influxdb, grafana**
 - `inputs.cpu, .disk, .mem`
 - `inputs.procstat, .jolokia2_agent`

Как измерять время?

По утилизации ресурсов



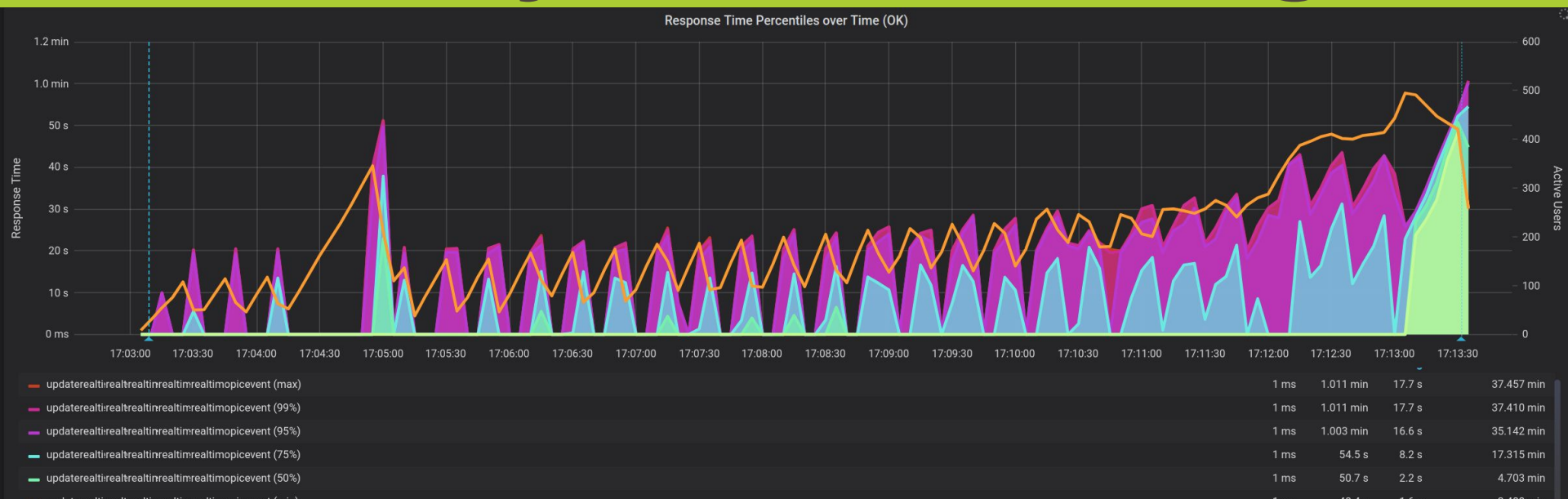
Как измерять время?

По отчету JMeter/Gatling

- до переполнения очереди, ERROR
- influxdb, grafana
 - backend listener в JMeter
 - `gatling.data.writers = [graphite]`

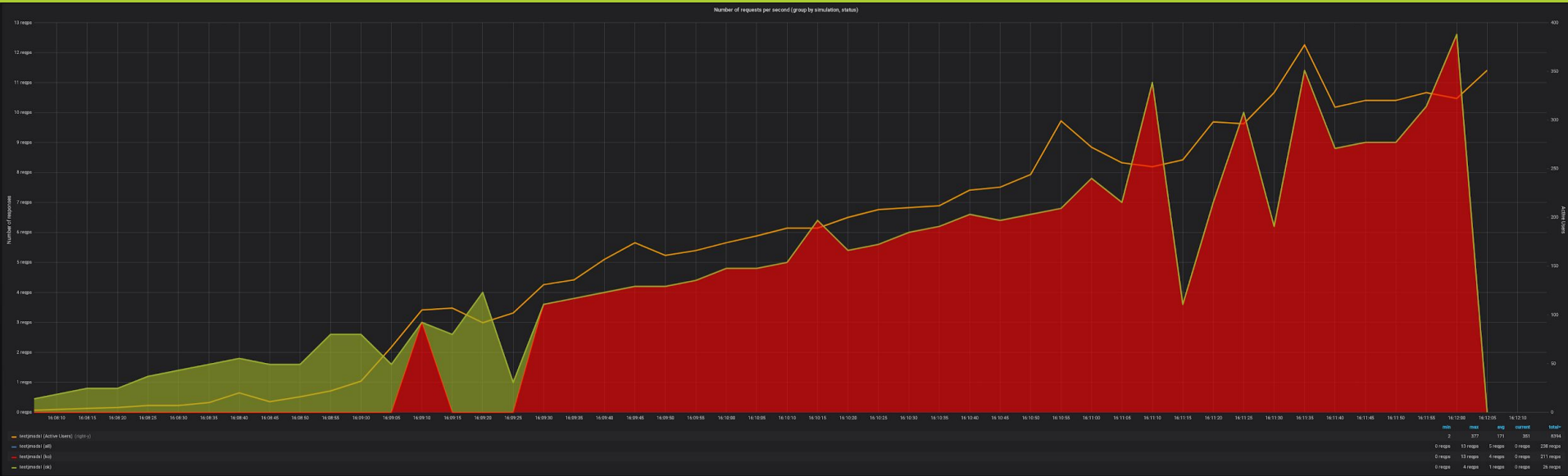
Как измерять время?

По отчету JMeter/Gatling



Как замерять время?

По отчету JMeter/Gatling



1.3 Отправка запросов

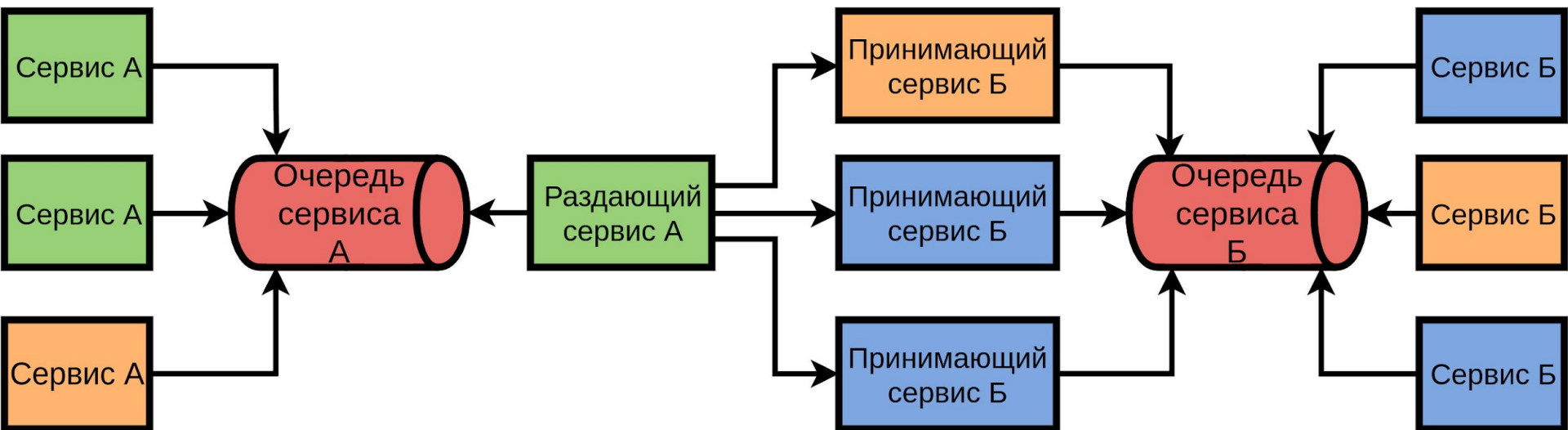
Часто send достаточно

**Мониторинг хороший
жизнь упрощает**



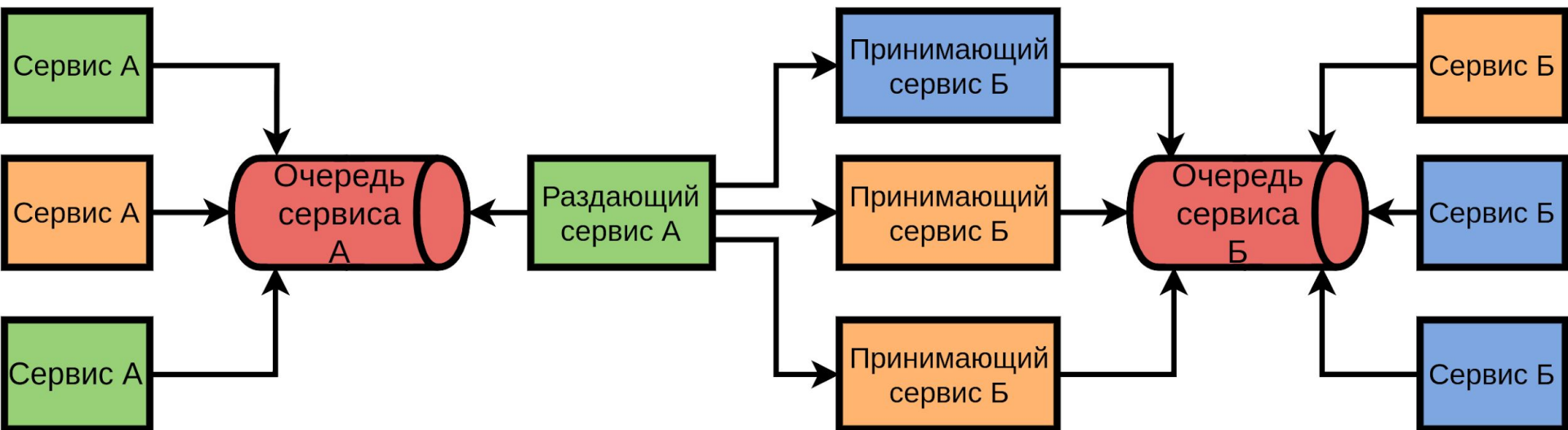
Просто отправлять запросы

А все ли ответы пришли?



Просто отправлять запросы

А все ли ответы пришли?



А все ли ответы пришли?

Посчитать количество

- **ответов в очереди**
- **статистика по логу**

А все ли ответы пришли?

Статистика по логу

```
(zcat -f application*.log.gz && cat application.log) |  
grep -E "2019-" |  
awk -F " --- " '{ print $2 " " $4 }' |  
sort |  
uniq -c |  
sort -nr
```

А все ли ответы пришли?

Статистика по логу

```
22002 INFO org.apache.camel.spring.spi.TransactionErrorHandler
13749 DEBUG bitronix.tm.BitronixTransaction
10285 TRACE org.apache.camel.management.MBeanInfoAssembler
7641 DEBUG bitronix.tm.journal.TransactionLogAppender
5687 DEBUG bitronix.tm.BitronixTransactionManager
4636 DEBUG bitronix.tm.timer.TaskScheduler
1824 TRACE org.apache.camel.util.ServiceHelper
1426 TRACE org.apache.camel.support.ServiceSupport
940 TRACE org.apache.camel.processor.aggregate.Aggregate
764 DEBUG bitronix.tm.journal.TransactionLogAppender
```

А все ли ответы пришли?

Посчитать количество

**Стабильные тестовые
данные стабильные
ответы дадут**



А все ли ответы пришли?

Посчитать количество

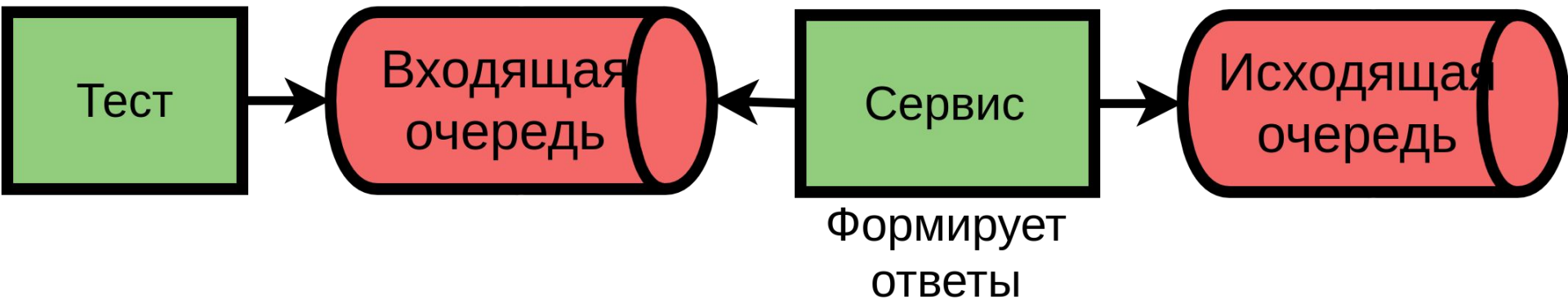
Логов мониторинг нагрузке нужен



Изображение с сайта <https://www.pinterest.ru/pin/323344448243498152/>

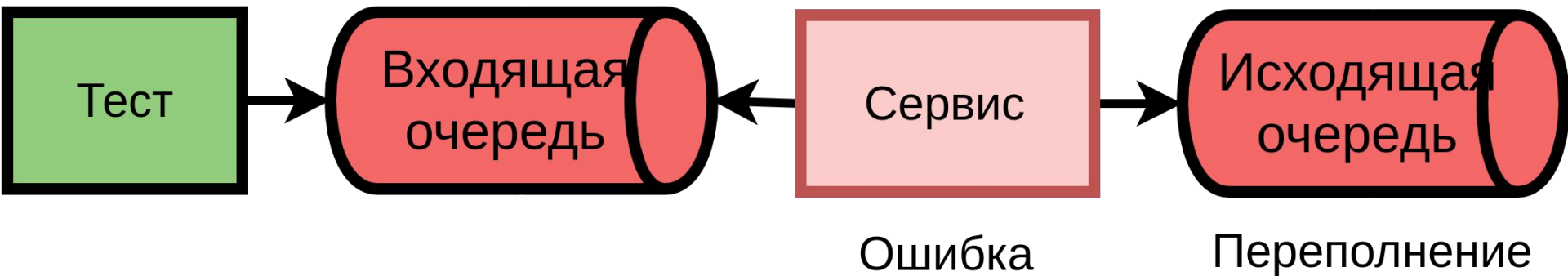
Просто отправлять запросы

Что делать если очередь ответов переполняется?



Просто отправлять запросы

Что делать если очередь ответов переполняется?



Очередь переполняется

Настроить Retention Policy

RabbitMQ:

- `x-overflow: drop-head`
- `max-length: 10000`

Очередь переполняется

Вычитывать в тесте

В отдельном потоке:

- **read/get/consume**
- **clear/purge**

2 JMeter и Gatling

2.1 Работа с JMS

2.2 Работа с AMQP

2.3 Отправка запросов

2.4 **Ожидание** ОТВЕТОВ

JMeter и Gatling (примеры)

Ожидание ответов

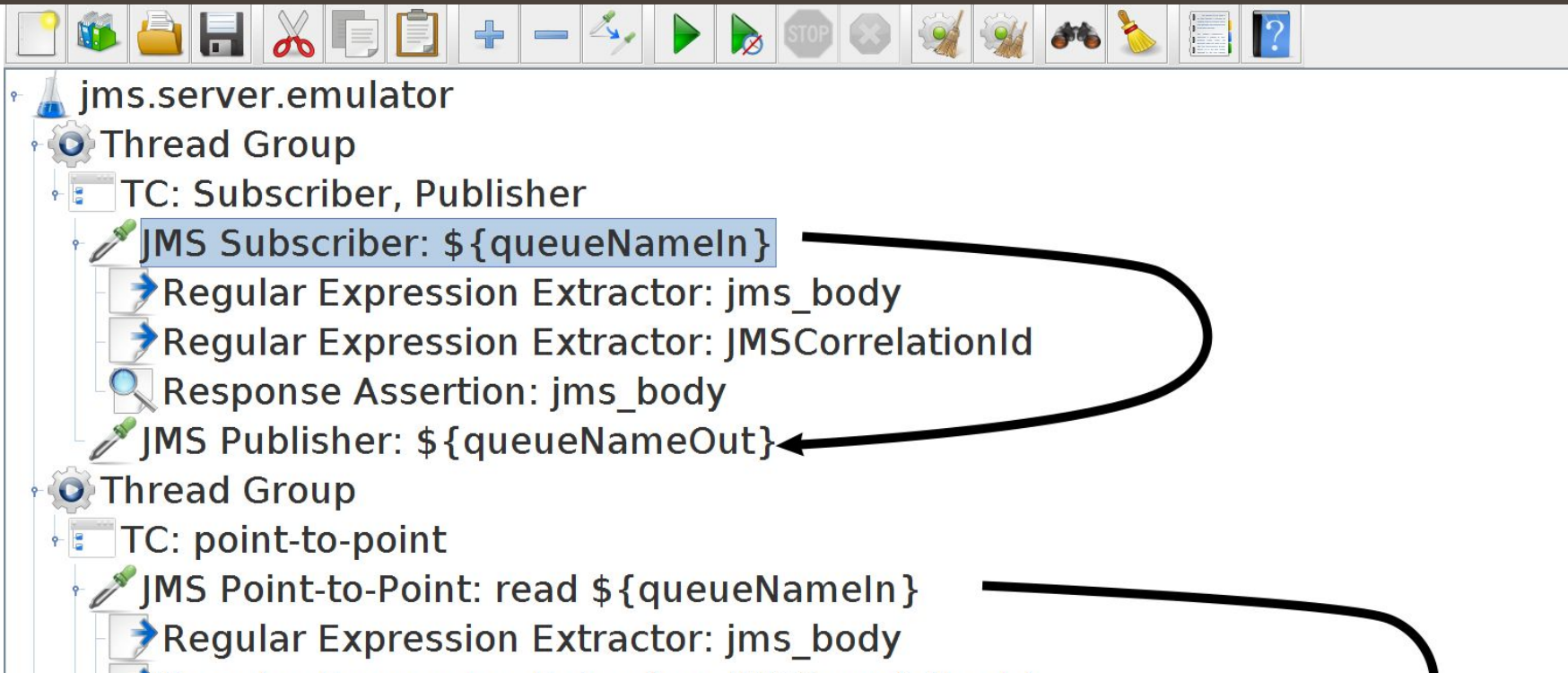
- JMS Request/Reply
- AMQP Request/Reply

Gatling-тест JMS Req/Reply

Сервер на Apache.JMeter

- **читает очередь Input**
- **пишет в очередь Output**
- **в 4 потока без пауз**

Сервер на Apache.JMeter



The screenshot displays the Apache JMeter GUI with a test plan configuration. The test plan is named "jms.server.emulator" and contains two Thread Groups. The first Thread Group is named "Thread Group" and contains a Test Controller (TC) named "TC: Subscriber, Publisher". This TC contains four elements: "JMS Subscriber: \${queueNameIn}" (highlighted in blue), "Regular Expression Extractor: jms_body", "Regular Expression Extractor: JMSCorrelationId", and "Response Assertion: jms_body". The second Thread Group is also named "Thread Group" and contains a TC named "TC: point-to-point". This TC contains two elements: "JMS Point-to-Point: read \${queueNameIn}" and "Regular Expression Extractor: jms_body". A large black arrow points from the "JMS Subscriber" element to the "JMS Publisher" element, indicating a dependency or data flow. Another black arrow points from the "JMS Point-to-Point" element to the right side of the screen.

- jms.server.emulator
 - Thread Group
 - TC: Subscriber, Publisher
 - JMS Subscriber: `${queueNameIn}`
 - Regular Expression Extractor: `jms_body`
 - Regular Expression Extractor: `JMSCorrelationId`
 - Response Assertion: `jms_body`
 - JMS Publisher: `${queueNameOut}`
 - Thread Group
 - TC: point-to-point
 - JMS Point-to-Point: read `${queueNameIn}`
 - Regular Expression Extractor: `jms_body`



jms.server.emulator

Сервер на Apache.JMeter

- Regular Expression Extractor: jms_body
- Regular Expression Extractor: JMSCorrelationId
- Response Assertion: jms_body
- JMS Publisher: \${queueNameOut}
- Thread Group
 - TC: point-to-point
 - JMS Point-to-Point: read \${queueNameIn}
 - Regular Expression Extractor: jms_body
 - Regular Expression Extractor: JMSCorrelationId
 - Response Assertion: jms_body
 - JMS Point-to-Point: send request \${queueNameOut}
- Backend Listener
- Aggregate Report

Тест на Gatling 3.3

```
def putMessageAndWait(queueInputName: String, queueOutputName: String, messageText: String): ChainBuilder =
  exec{
    session =>
      val now = System.currentTimeMillis()
      session.set("createTime", dfDateTime.format(now))
        .set("messageText", messageText)
  }
  .exec(jms( requestName = "requestReply"
    .requestReply
    .queue(queueInputName)
    .replyQueue(queueOutputName)
    .noJmsReplyTo
    .textMessage(text = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"
      + "<input>${messageText} --- ${createTime}</input>")
  )
```

: JmsDslBuilderBase

: RequestReplyDslBuilderQueue

: RequestReplyDslBuilderMessage

: RequestReplyDslBuilderMessage

: RequestReplyDslBuilderMessage

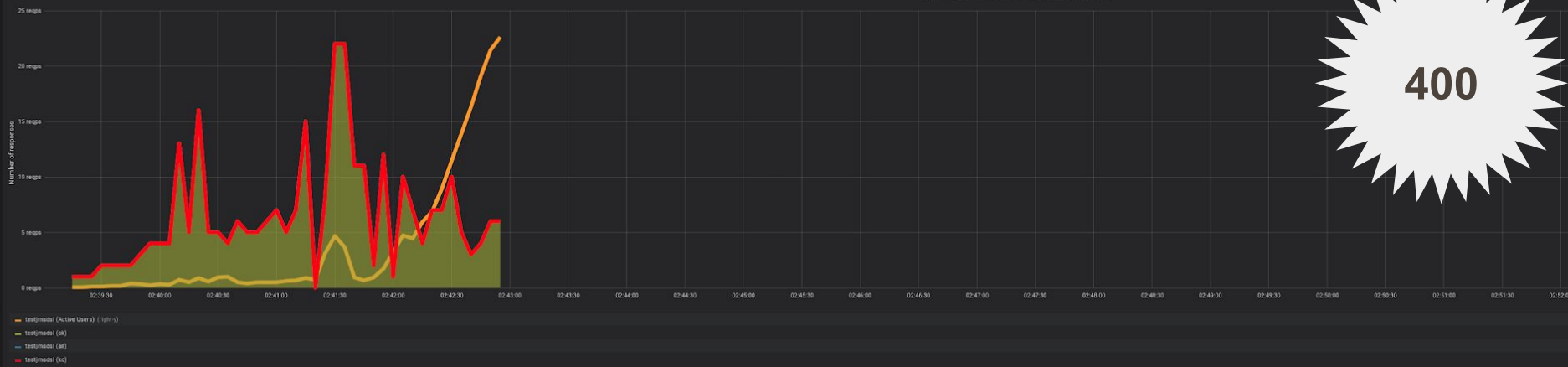
: RequestReplyDslBuilder

Тест на Gatling 3.3 (3 МИН)

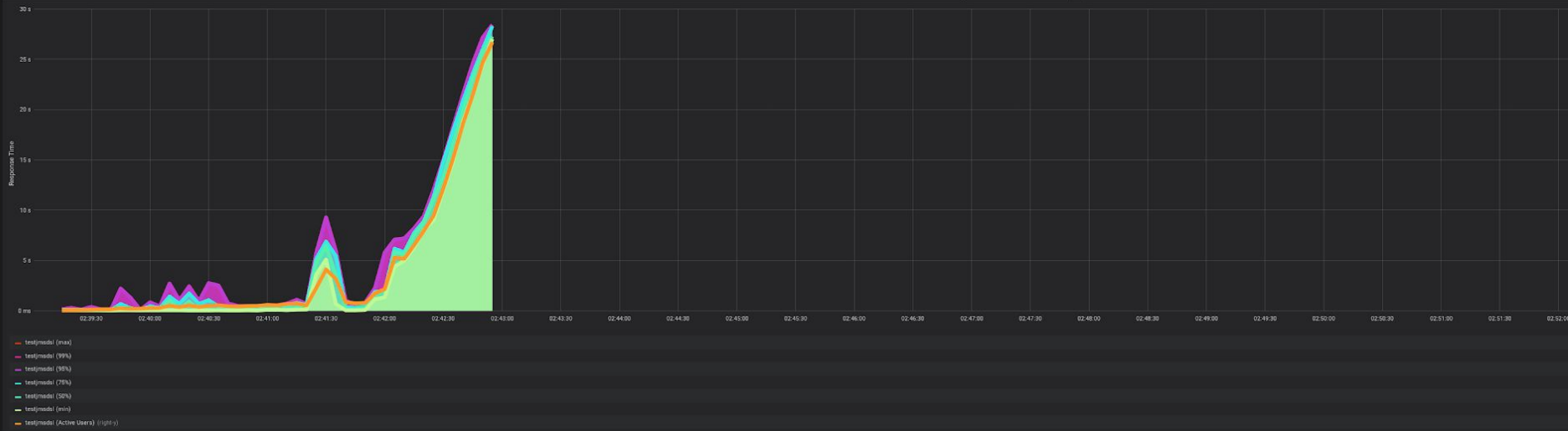
```
val jmsConfig = jms
    .connectionFactory(connectionFactory)
    .usePersistentDeliveryMode
    .replyTimeout(timeout = 3 * 60 * 1000)
    .matchByCorrelationId
    .listenerThreadCount(threadCount = 1)
```

▼ RPS

Number of requests per second (group by simulation, status)



Response Time Percentiles over Time (OK) ▼

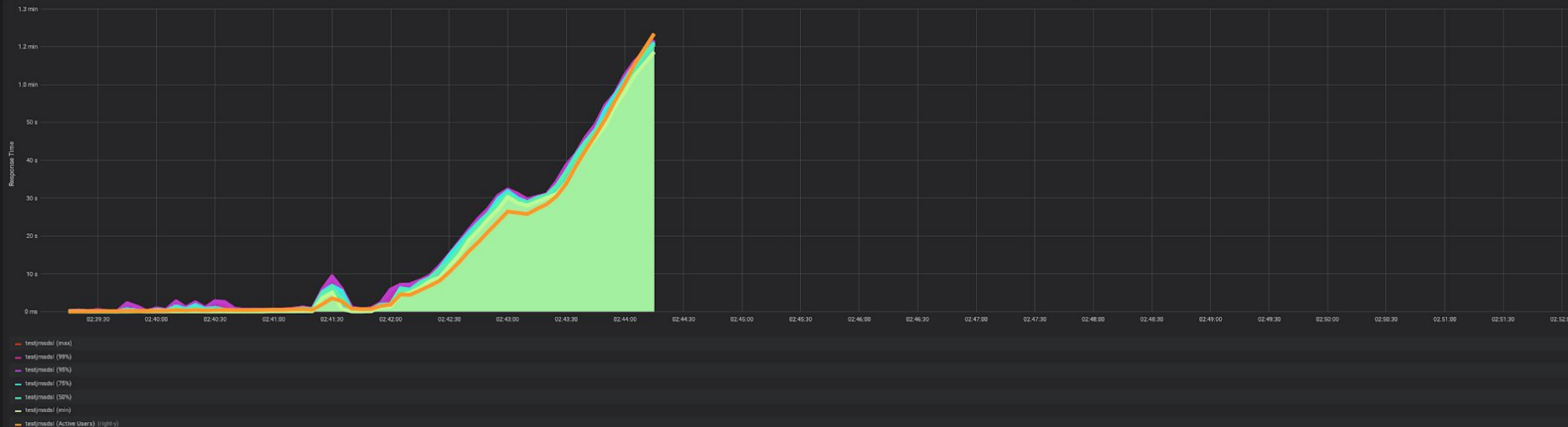


▼ RPS

Number of requests per second (group by simulation, status)

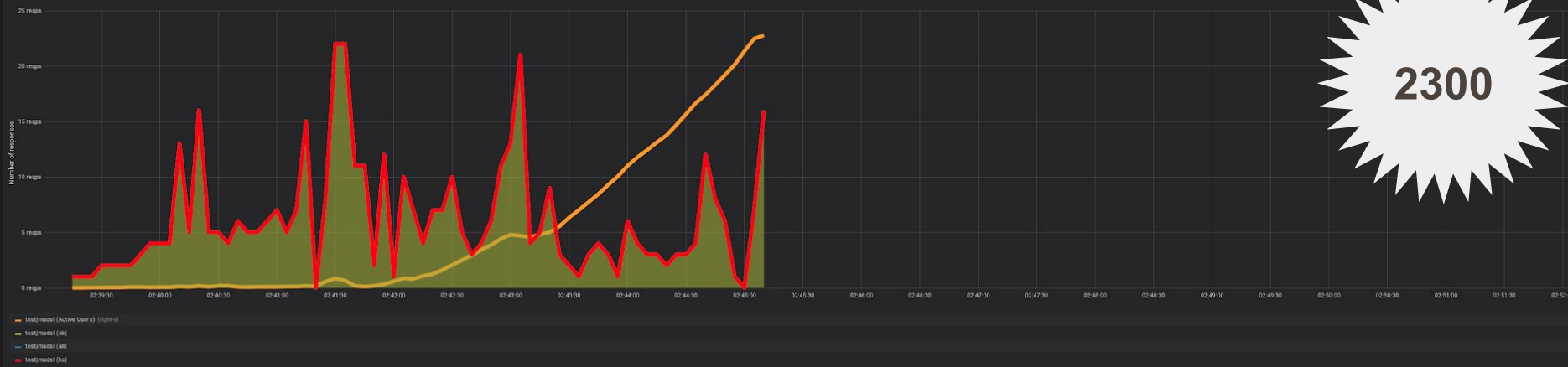


Response Time Percentiles over Time (OK) -

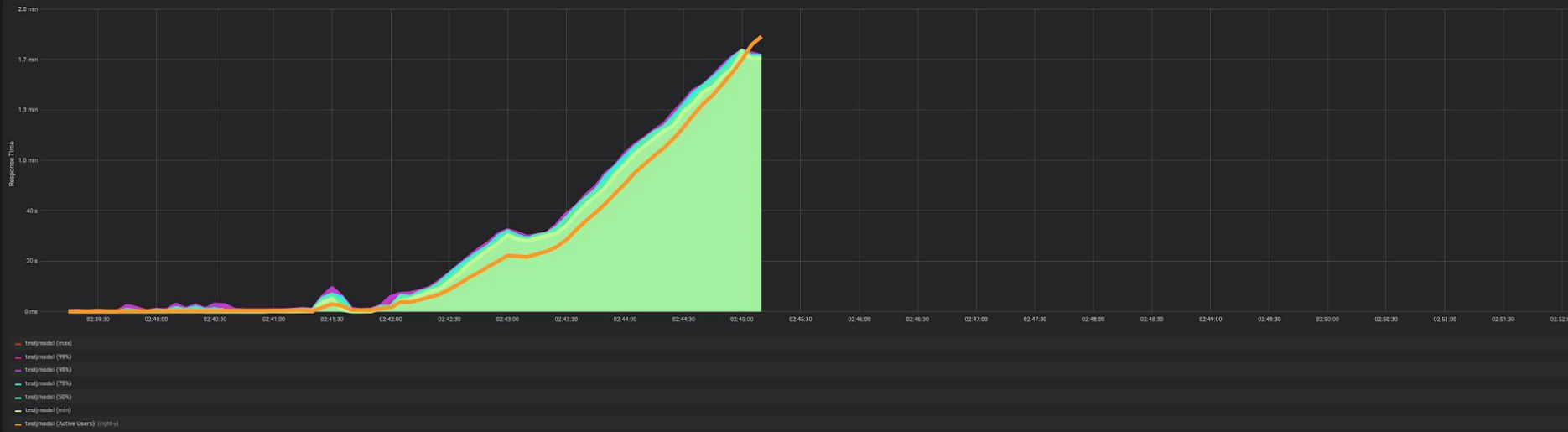


▼ RPS

Number of requests per second (group by simulation, status)

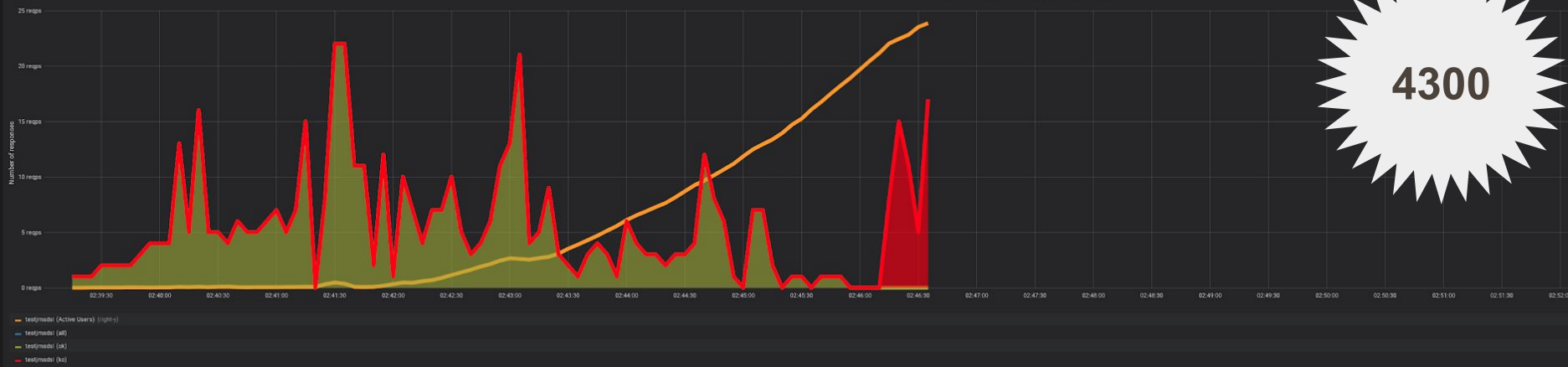


Response Time Percentiles over Time (OK) -



▼ RPS

Number of requests per second (group by simulation, status)

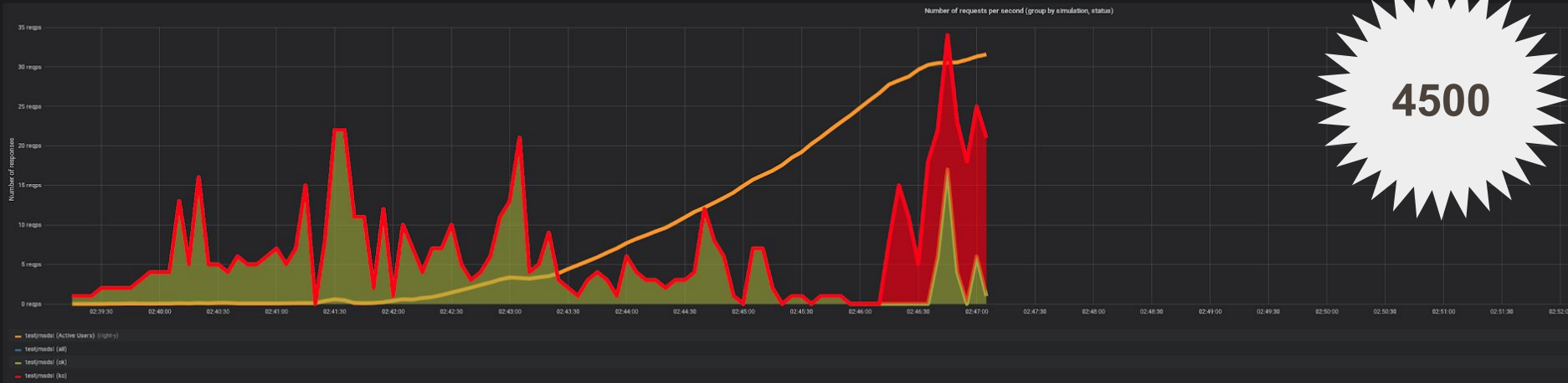


4300

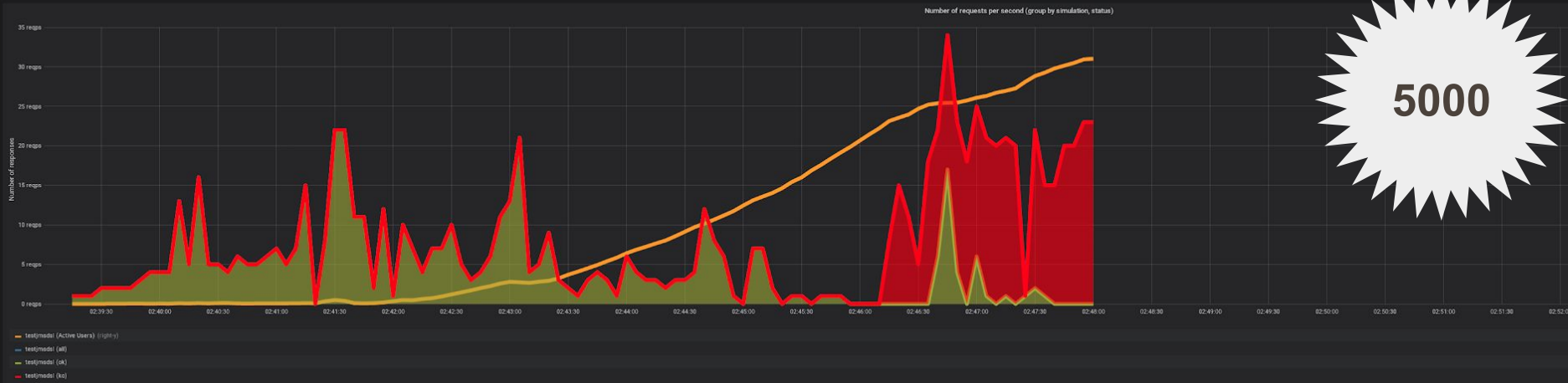
Response Time Percentiles over Time (OK)



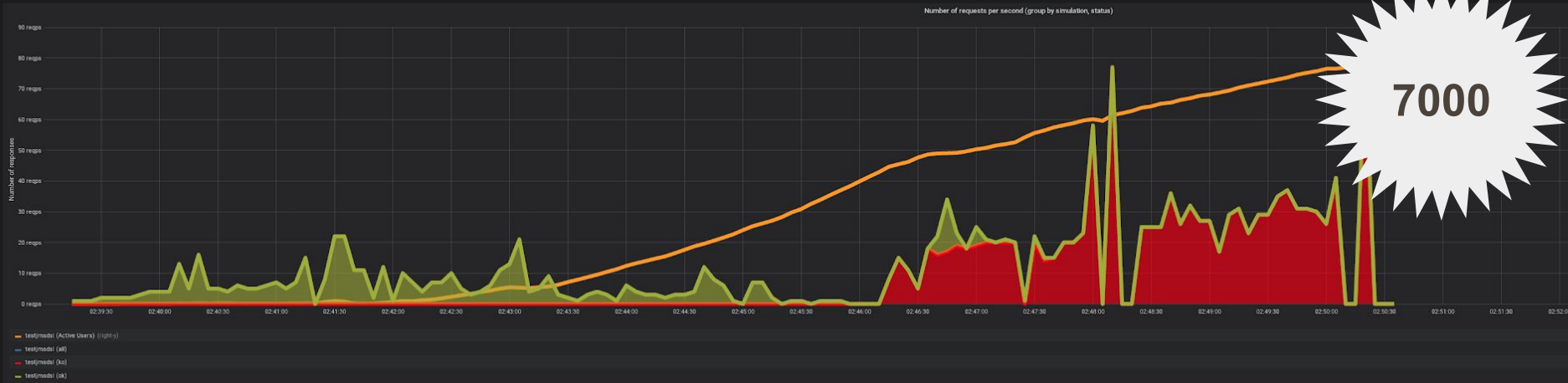
▼ RPS



▼ RPS



▼ RPS



RPS

Number of requests per second (group by simulation, status)



- test(model) (Active users) (right)
- test(model) (all)
- test(model) (ok)
- test(model) (ok)





- test(model) (error)
- test(model) (99%)
- test(model) (95%)
- test(model) (75%)
- test(model) (50%)
- test(model) (error)
- test(model) (Active users) (right)

Out of memory error

Причины Out of memory

- **ЛИМИТЫ**
 - **ulimit -n file descriptors**
 - **ulimit -u processes**
- **лимит памяти: -Xmx**

Другие проблемы

- ЛИМИТ ПОДКЛЮЧЕНИЙ К каналу IBM.MQ
 - **ShareConv = 1** 
 - **ShareConv = 0** 

**Не ждите сообщения –
получайте их (в тестах)**

**Send + (Get/Consume)
производительнее,
чем Request/Reply**



RabbitMQ in Depth

Don't get
messages;
consume them

livebook.manning.com/book/rabbitmq-in-depth/chapter-5



RabbitMQ IN DEPTH

Gavin M. Roy

Не получайте сообщения – потребляйте их

- **Consume** вместо **Get**
- без **CorrelationID**
- без **Selector**

В одном потоке

- **отправить запрос**

В другом потоке:

- **получить ответ**

В одном потоке

- **отправить запрос**

В другом потоке:

- **получить ответ**

**Не ждите сообщения –
получайте их (в тестах)**

Как сделать **Get в **Gatling**?**

**Готовое решение есть в
Apache.JMeter, не в Gatling**

**Не ждите сообщения –
получайте их (в тестах)**

Но как **замерить время?**

**От момента отправки до
момента получения**

**Не ждите сообщения –
получайте их (в тестах)**

Но если не пришел ответ?

**Не дождались
момента получения**

1 Очереди

2 JMeter и Gatling

3 Проблемы и задачи

4 Подходы решения

5 Наработки

3 Проблемы и задачи

- **Get/Read/... в Gatling**
- **как замерить время**
- **что если не было ответа
???**

Get/Read/... в Gatling

		R/R	Send	Get	Clear
JMeter	JMS Publisher, Subscriber		есть	есть	
JMeter	JMS Point-to-point	есть	есть	есть	есть
JMeter	AMQP Publisher, Consumer		есть	есть	есть
Gatling	JMS Plugin	есть	есть	нет	
Gatling	AMQP Plugin	есть	есть	нет	

Get/Read/... в Gatling

- как расширить Gatling
- без опыта разработки плагинов к Gatling

Как измерить **время**?

Сценарий отправки запроса

Send



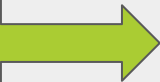
Get

Сценарий получения ответа

Как измерить **время**?

Сценарий отправки запроса

Send



Get

Сценарий получения ответа

Как измерить **время**?

- между событиями в разных сценариях
- нет общих сессий
- нет DSL

Что если не было **ответа**

Сценарий отправки запроса

Send



Error



Сценарий получения ответа

Что если не было **ответа**

Сценарий отправки запроса

Send



Error



Сценарий получения ответа

Что если не было **ответа**

- как залогировать ошибку
- нет TearDown событий
- нет DSL

1 Очереди

2 JMeter и Gatling

3 Проблемы и задачи

4 Подходы решения

5 Наработки

Get/Read/... в Gatling

- **воспользоваться ехес**
- **написать плагин**

Get/Read/... в Gatling

ВОСПОЛЬЗОВАТЬСЯ `exec`

```
val factory = new ConnectionFactory()
val conn = factory.newConnection()
val channel = conn.createChannel()
val response = channel.basicGet("queue", false);
channel.close()
conn.close()
```

Get/Read/... в Gatling

ВОСПОЛЬЗОВАТЬСЯ `exec`

```
exec {  
    session =>  
        val factory = new ConnectionFactory()  
        ...  
        val response = channel.basicGet("queue", false)  
        ...  
        conn.close()  
    session  
}
```

Get/Read/... в Gatling

ВОСПОЛЬЗОВАТЬСЯ exec

```
exec {  
    session =>  
        val factory = new ConnectionFactory()  
        ...  
        val response = channel.basicGet("queue", false)  
        ...  
        conn.close()  
        session.set("response", response)  
}
```

Get/Read/... в Gatling

воспользоваться exec

gist.github.com/polarnik/10dce91e372ed74ba7038250125f26a6

Get/Read/... в Gatling

```
exec{ session => ... session }
```

**Без плагина написания
расширить Gatling
МОЖНО**



Get/Read/... в Gatling

написать плагин

blog.codecentric.de/en/2017/07/gatling-load-testing-part-2-extending-gatling

Gatling Load Testing Part 2 – Extending Gatling

Как измерить время

- **запоминайте время**
- **программно создавайте результаты теста**

Как измерить время

- **ConcurrentHashMap**
- **StatsEngine** и **asyncLog**,
SampleResult (JMeter)

В одном потоке

- Отправить запрос с ID
- Сохранить время по ключу ID в (Shared) ConcurrentHashMap

В другом потоке:

- Получить ответ и ID из ответа
- Получить время по ID из HashMap
- Создать SampleResult, Stats

В одном потоке

- Отправить запрос с ID
- Сохранить время по ключу ID в (Shared) ConcurrentHashMap

В другом потоке:

- Получить ответ и ID из ответа
- Получить время по ID из HashMap
- Создать SampleResult, Stats

Сохранить время в JMeter

```
def dateStart = new Date();  
def sharedMap = SharedHashMap.GetInstance();  
  
sharedMap.put(ID, dateStart);
```

Получить время в JMeter

```
def dateEnd = new Date();  
def sharedMap = SharedHashMap.GetInstance();  
  
def dateStart = sharedMap.get(ID);  
def start = dateStart.getTime();  
def elapsed = dateEnd.getTime() - start;  
def newRes = new SampleResult(start, elapsed);  
prev.addSubResult(newRes);
```

Сохранить время в JMeter*

```
def testData = [:];
testData["dateStart"] = new Date();
def sharedMap = SharedHashMap.GetInstance();
def testDataQ = sharedMap.get(ID);
if (testDataQ == null) {
    testDataQ = new LinkedBlockingQueue();
    sharedMap.put(key, testDataQ);
}
testDataQ.add(testData);
```

Получить время в JMeter*

```
def sharedMap = SharedHashMap.GetInstance();
def key = ID;
def testData = SharedMap.get(key).poll();
if (testData != null) {
    def dateEnd = new Date();
    def dateStart = testData["dateStart"];
    def start = dateStart.getTime();
    def elapsed = dateEnd.getTime() - start;
    def newResult = new SampleResult(start, elapsed)
    prev.addSubResult(newResult);
} else { ... }
```

StatsEngine в Gatling

```
asynclog("label")
  .startTimestamp(
    "2019-11-25 00:19:00",
    "yyyy-MM-dd HH:mm:ss")
  .endTimestamp(
    "2019-11-25 00:29:00",
    "yyyy-MM-dd HH:mm:ss")
  .status(io.gatling.commons.stats.OK)
  .responseCode("200")
```


1 Очереди

2 JMeter и Gatling

3 Проблемы и задачи

4 Подходы решения

5 Нарботки

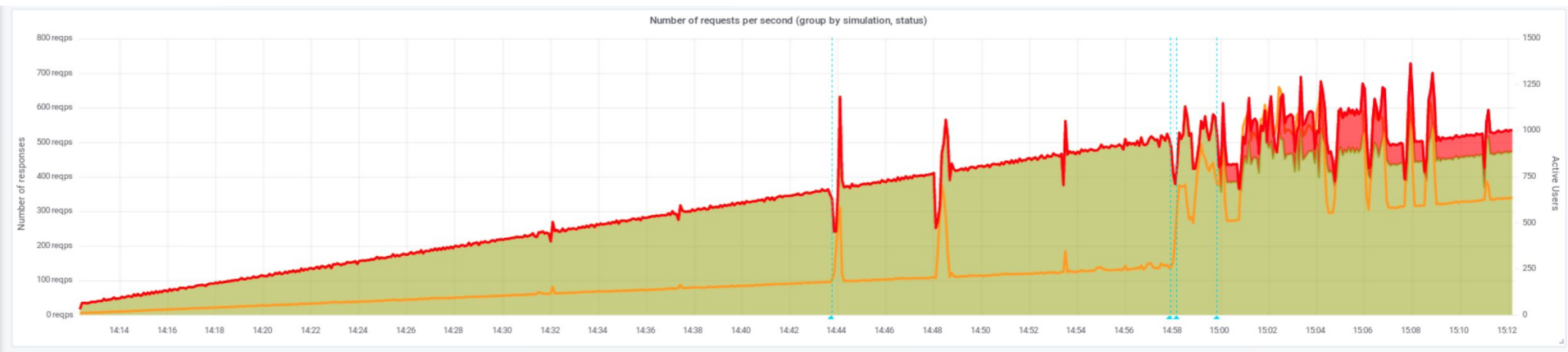
Наработки

- **Grafana-доска для Gatling**
- **SharedHashMap**
- **Gatling AsyncLog plugin**
- **тесты с JMeter и Gatling**

Grafana-доска для Gatling

Копия HTML-отчета

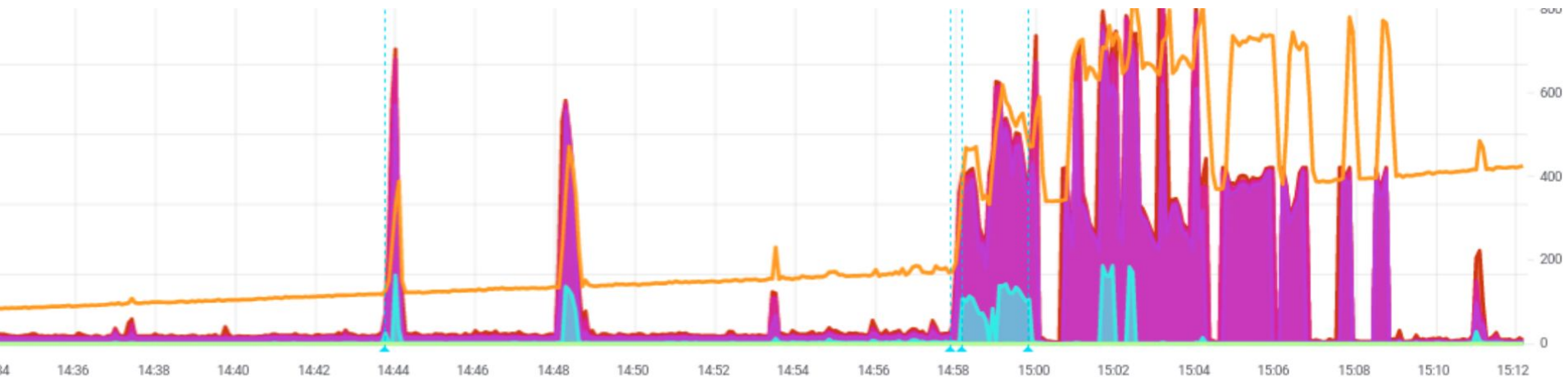
grafana.com/grafana/dashboards/9935



Grafana-доска для Gatling

Копия HTML-отчета

grafana.com/grafana/dashboards/9935



Grafana-доска для Gatling

Поддержка group("name"){

grafana.com/grafana/dashboards/9935

```
"gatling.*.*.*.stdDev measurement.simulation.request.status.field",  
"gatling.*.users.*.* measurement.simulation.measurement.request.field",  
"gatling.*.*.*.*.*.ok.* measurement.simulation.group1.group2.group3.request.status",  
"gatling.*.*.*.*.*.ko.* measurement.simulation.group1.group2.group3.request.status",  
"gatling.*.*.*.*.*.all.* measurement.simulation.group1.group2.group3.request.status",  
"gatling.*.*.*.*.*.ok.* measurement.simulation.group1.group2.request.status",  
"gatling.*.*.*.*.*.ko.* measurement.simulation.group1.group2.request.status",  
"gatling.*.*.*.*.*.all.* measurement.simulation.group1.group2.request.status"
```

SharedHashMap

habr

Публикации

Новости

Пользователи

Хабы

Компании

Стать автором



greatvovan 29 апреля 2015 в 11:37

JMeter: забудьте про BeanShell Sampler

Тестирование IT-систем

С помощью стандартных элементов тест-плана в Jmeter можно сделать многое, но далеко не всё. Для расширения функциональности и реализации более сложной логики принято использовать BeanShell Sampler — как-то во всём мире так исторически сложилось. И во всём

SharedHashMap

Проект из 15 строк кода

github.com/polarnik/

[io.github.qaload.SharedHashMap](https://github.com/polarnik/io.github.qaload.SharedHashMap)

```
1  import java.util.concurrent.ConcurrentHashMap;
2
3  public class SharedHashMap
4  {
5      private static final ConcurrentHashMap instance = new ConcurrentHashMap();
6
7      public static ConcurrentHashMap getInstance()
```

SharedHashMap

JMeter-проект с mvn install

[github.com/polarnik/
perflab_code/tree/jmeter-plugin-lib](https://github.com/polarnik/perflab_code/tree/jmeter-plugin-lib)

```
50     <!-- Build Settings -->
51     <build>
52         <finalName>SharedHashMap</finalName>
53         <plugins>
54
55             <plugin>
56                 <groupId>com.lazerycode.jmeter</groupId>
```


Gatling AsyncLog plugin

Простой Gatling-плагин

github.com/polarnik/gatling-asynclog-plugin/

The image shows a screenshot of an IDE with two windows. The left window displays the project structure of the 'gatling-asynclog-plugin' project, showing a directory tree with folders like '.idea', 'core', 'src', 'main', and 'scala'. Under 'scala', there is a folder 'qaload.gatling.asynclogplugin' containing sub-folders 'action' and 'request'. The 'action' folder contains 'LogAction.scala' and 'LogActionBuilder.scala'. The 'request' folder contains 'AsynclogAttributes.scala', 'AsynclogDsl.scala', and 'Predef.scala'. There is also an 'example' folder. The right window shows the documentation page for the 'Gatling AsyncLog Plugin'. The page has a title 'Gatling AsyncLog Plugin' and a section 'Назначение' (Purpose) with the text: 'Плагин для логирования операций в Gatling 3.2.1. Поддерживает логирование, в том числе, и операций, которые начались в одном сценарии'. Below that is a section 'Методы для замера длительности работы' (Methods for measuring work duration). At the bottom, there is a section 'Доступные действия' (Available actions) with a list of actions: 'start — старт', 'id — обновить идентификатор транзакции', 'log — залогировать операцию', 'stop — удалить транзакцию', and 'stopAll — удалить все транзакции'.

Project ▾

- gatling-asynclog-plugin [gatling.asynclogplugin] ~/Project/github
- .idea
- core
- src
 - main
 - scala
 - qaload.gatling.asynclogplugin
 - action
 - LogAction.scala
 - LogActionBuilder.scala
 - request
 - AsynclogAttributes.scala
 - AsynclogDsl.scala
 - Predef.scala
 - example

Gatling AsyncLog Plugin

Назначение

Плагин для логирования операций в Gatling 3.2.1. Поддерживает логирование, в том числе, и операций, которые начались в одном сценарии

Методы для замера длительности работы

Доступные действия

- start — старт
- id — обновить идентификатор транзакции
- log — залогировать операцию
- stop — удалить транзакцию
- stopAll — удалить все транзакции

Тесты с JMeter и Gatling

Docker, RabbitMQ, IBM.MQ

github.com/polarnik/

[Performance.testing.via.AMQP.with.Gatling](https://github.com/polarnik/Performance.testing.via.AMQP.with.Gatling)

```
mkdir -p /tmp/var/lib/influxdb
```

```
docker run --name=influxdb \  
  --network=loadlab \  
  -p 8086:8086 -p 2003:2003 -p 2004:2004 \  
  -v $PWD/infrastructure/influxdb/influxdb.conf:/etc/influxdb/  
  influxdb.conf:ro \  
  \
```

Тесты с JMeter и Gatling

Примеры тестов

github.com/polarnik/

[Performance.testing.via.AMQP.with.Gatling](https://github.com/polarnik/Performance.testing.via.AMQP.with.Gatling)

slido

Join at
slido.com
#qa_load

Audience Q&A Session