

КАК (НЕ) НАДО ПРОВОДИТЬ НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ

**ГРИГОРИЙ КОШЕЛЕВ
КОНТУР**



— ПИШУ НА JAVA
И ТЕСТИРУЮ (СВОЙ) КОД

https://tg.me/chnl_GregoryKoshelev



- ПИШУ НА JAVA
И ТЕСТИРУЮ (СВОЙ) КОД
- ВЫСТУПАЮ НА КОНФЕРЕНЦИЯХ
 - * DOTNEXT
 - * JOKER
 - * JPOINT
 - * DEVOOPS
 - * HEISENBUG (NEW!)

https://tg.me/chnl_GregoryKoshelev

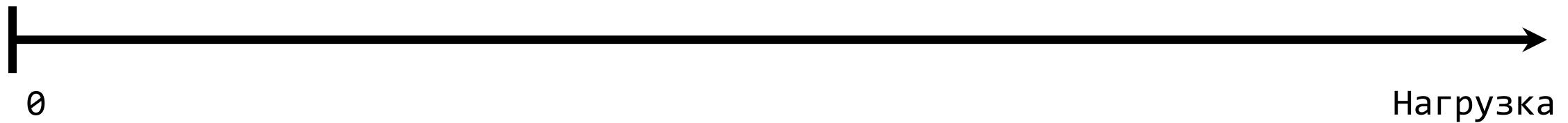


— С 2018 ПО 2022
СОЗДАВАЛ HIGHLOAD СИСТЕМУ
ДЛЯ ТЕЛЕМЕТРИИ
VOSTOK HERCULES

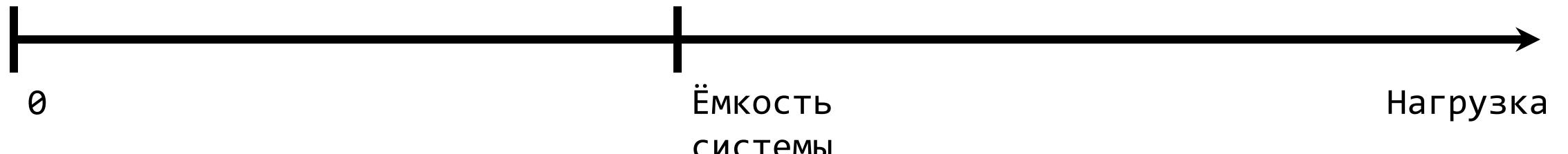
<https://github.com/vostok/hercules>

PERFORMANCE-ТЕСТИРОВАНИЕ

PERFORMANCE-ТЕСТИРОВАНИЕ



PERFORMANCE-ТЕСТИРОВАНИЕ



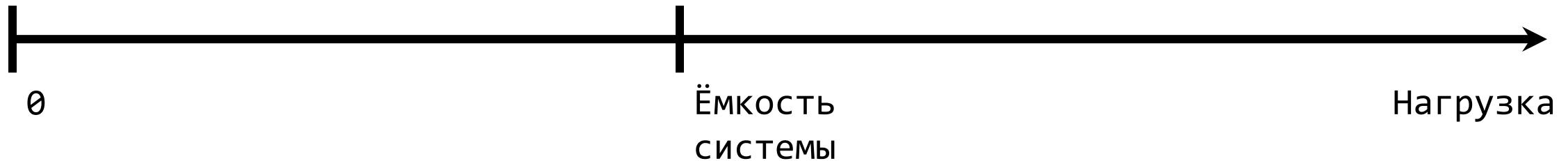
PERFORMANCE-ТЕСТИРОВАНИЕ



PERFORMANCE-ТЕСТИРОВАНИЕ

ТЕСТИРОВАНИЕ МАСШТАБИРУЕМОСТИ

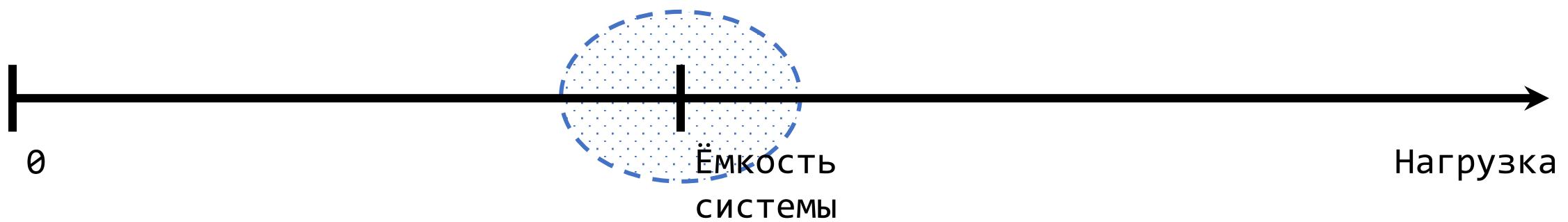
SCALABILITY TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ТЕСТИРОВАНИЕ МАСШТАБИРУЕМОСТИ

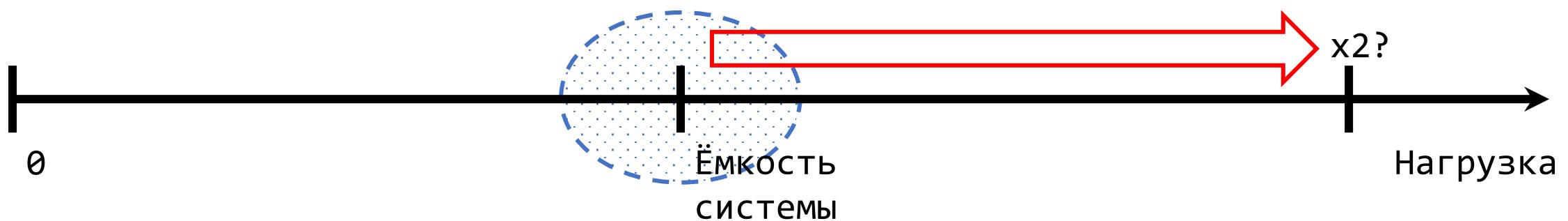
SCALABILITY TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ТЕСТИРОВАНИЕ МАСШТАБИРУЕМОСТИ

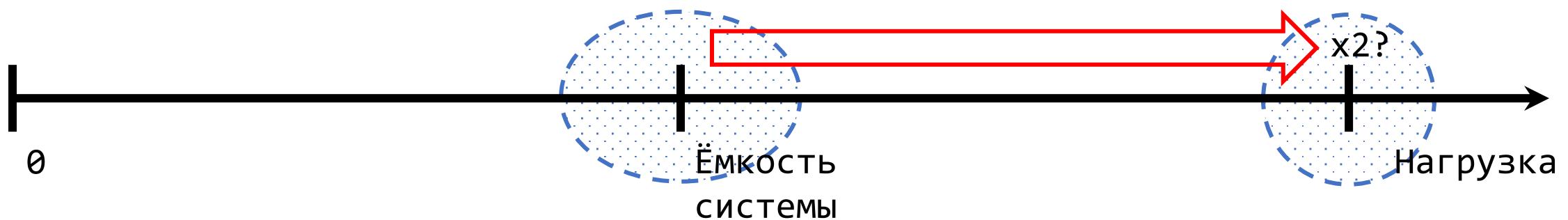
SCALABILITY TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ТЕСТИРОВАНИЕ МАСШТАБИРУЕМОСТИ

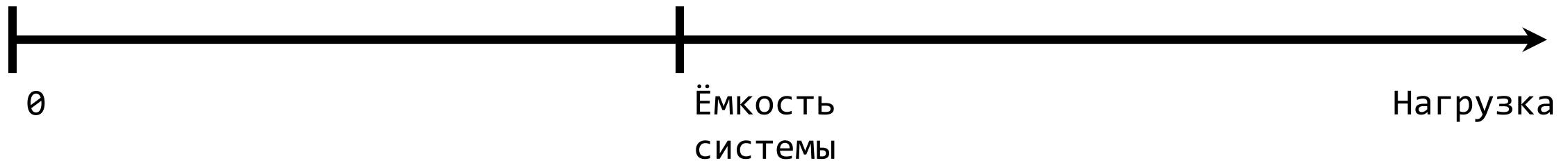
SCALABILITY TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ

LOAD TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ

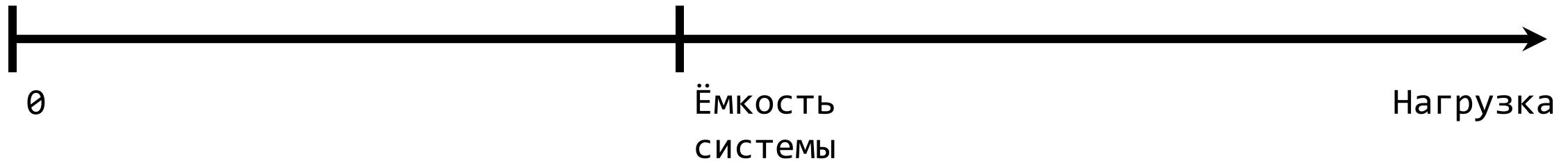
LOAD TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

СТРЕСС-ТЕСТИРОВАНИЕ

STRESS TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

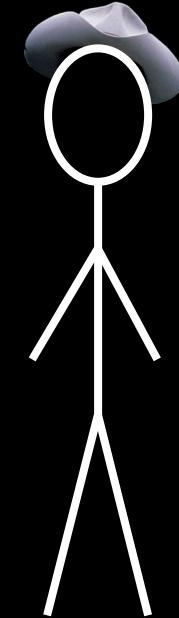
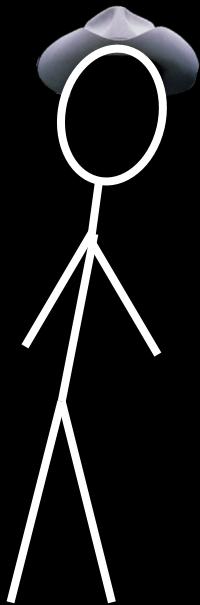
СТРЕСС-ТЕСТИРОВАНИЕ

STRESS TESTING



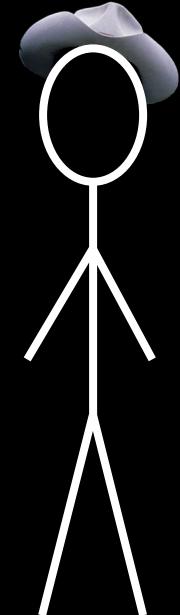
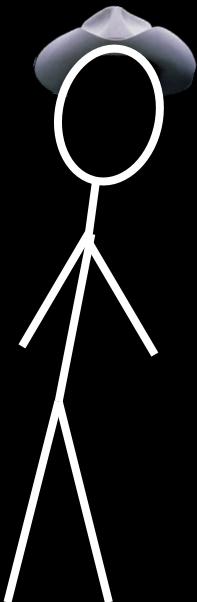
НА НЕСКОЛЬКО RPS БОЛЬШЕ

НА НЕСКОЛЬКО RPS БОЛЬШЕ



НА НЕСКОЛЬКО RPS БОЛЬШЕ

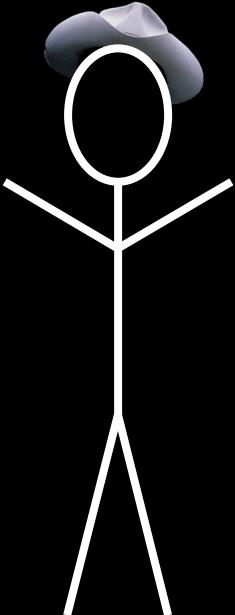
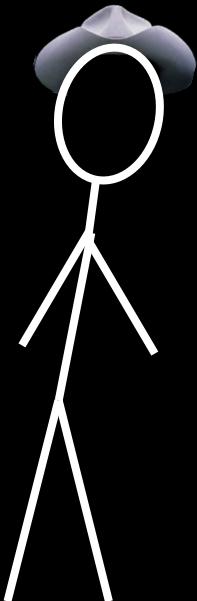
БУТЧ, КАК ДЕЛА С НАШИМ
НОВЫМ СЕРВИСОМ?



НА НЕСКОЛЬКО RPS БОЛЬШЕ

БУТЧ, КАК ДЕЛА С НАШИМ
НОВЫМ СЕРВИСОМ?

ПОЛНАЯ ШЛЯПА, КИД!

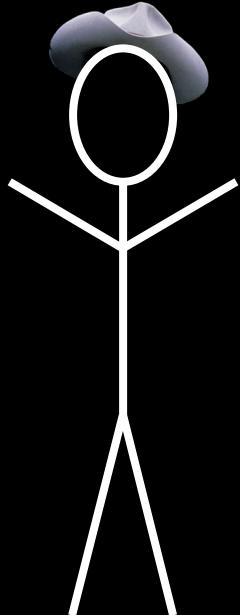
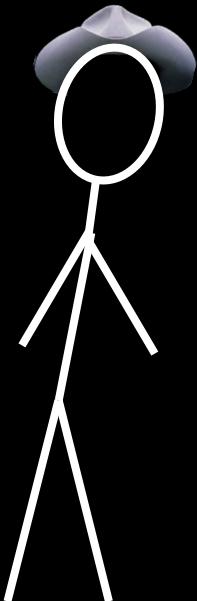


НА НЕСКОЛЬКО RPS БОЛЬШЕ

БУТЧ, КАК ДЕЛА С НАШИМ
НОВЫМ СЕРВИСОМ?

ПОЛНАЯ ШЛЯПА, КИД!

???



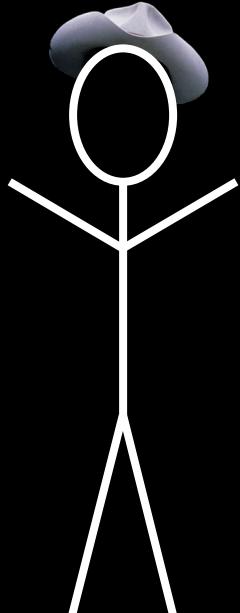
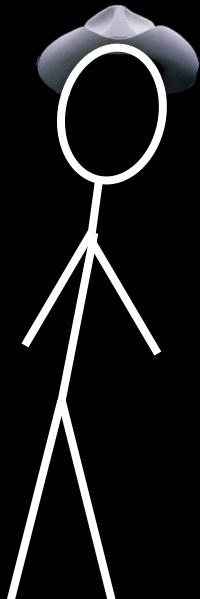
НА НЕСКОЛЬКО RPS БОЛЬШЕ

БУТЧ, КАК ДЕЛА С НАШИМ
НОВЫМ СЕРВИСОМ?

ПОЛНАЯ ШЛЯПА, КИД!

???

СЕРВИС НЕ ВЫВЕЗ
300КК/НАНОСЕК
ЗАПРОСОВ!



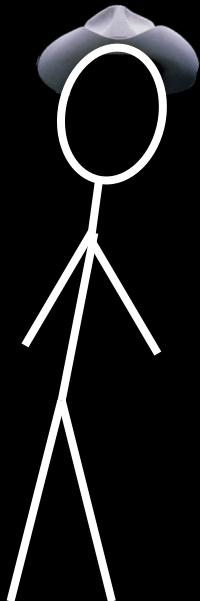
НА НЕСКОЛЬКО RPS БОЛЬШЕ

БУТЧ, КАК ДЕЛА С НАШИМ
НОВЫМ СЕРВИСОМ?

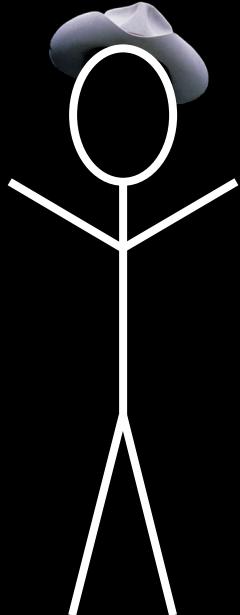
ПОЛНАЯ ШЛЯПА, КИД!

???

СЕРВИС НЕ ВЫВЕЗ
300КК/НАНОСЕК
ЗАПРОСОВ!



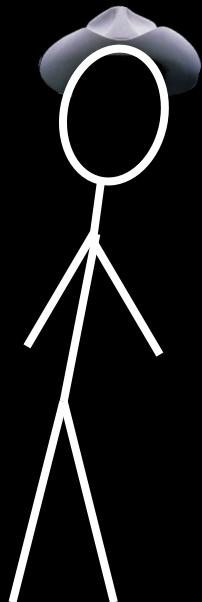
ОУ-ЩИТ!
А КАКАЯ БУДЕТ
НАГРУЗКА ОТ ЮЗЕРОВ?



НА НЕСКОЛЬКО RPS БОЛЬШЕ

БУТЧ, КАК ДЕЛА С НАШИМ
НОВЫМ СЕРВИСОМ?

???

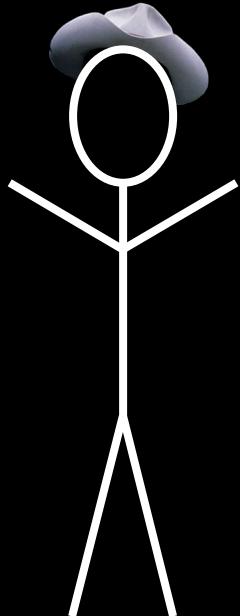


ОУ-ЩИТ!
А КАКАЯ БУДЕТ
НАГРУЗКА ОТ ЮЗЕРОВ?

ПОЛНАЯ ШЛЯПА, КИД!

СЕРВИС НЕ ВЫВЕЗ
300КК/НАНОСЕК
ЗАПРОСОВ!

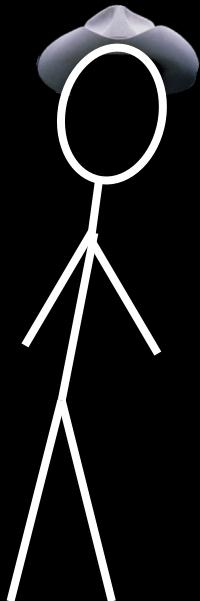
НУ...
ЗАПРОСОВ 100...



НА НЕСКОЛЬКО RPS БОЛЬШЕ

БУТЧ, КАК ДЕЛА С НАШИМ
НОВЫМ СЕРВИСОМ?

???



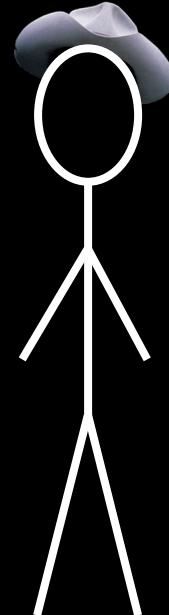
ОУ-ЩИТ!
А КАКАЯ БУДЕТ
НАГРУЗКА ОТ ЮЗЕРОВ?

ПОЛНАЯ ШЛЯПА, КИД!

СЕРВИС НЕ ВЫВЕЗ
300КК/НАНОСЕК
ЗАПРОСОВ!

НУ...
ЗАПРОСОВ 100...

В ЧАС



НА НЕСКОЛЬКО RPS БОЛЬШЕ

DIRECTED BY
RICHARD J. GATLING



НА НЕСКОЛЬКО RPS БОЛЬШЕ

DIRECTED BY
RICHARD J. GATLING

PERFORMANCE-ТЕСТИРОВАНИЕ

СТРЕСС-ТЕСТИРОВАНИЕ

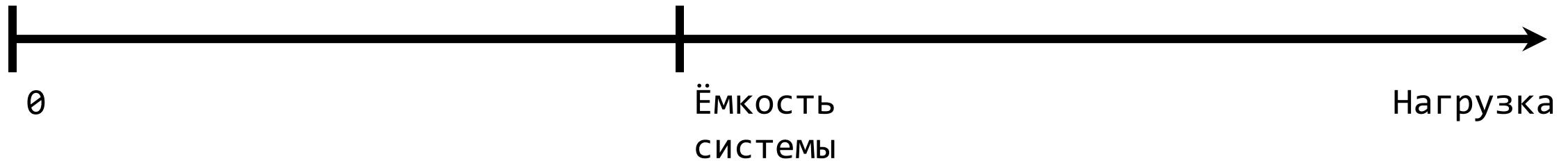
STRESS TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ПИКОВОЕ ТЕСТИРОВАНИЕ

SPIKE TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ПИКОВОЕ ТЕСТИРОВАНИЕ

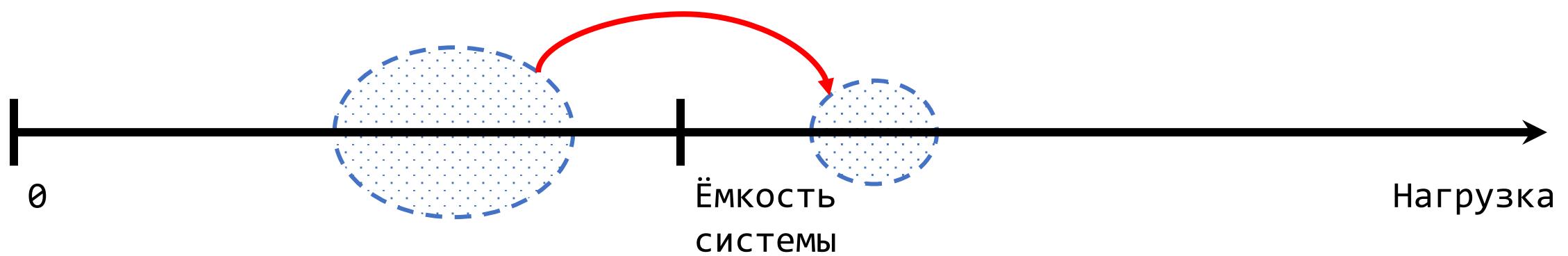
SPIKE TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ПИКОВОЕ ТЕСТИРОВАНИЕ

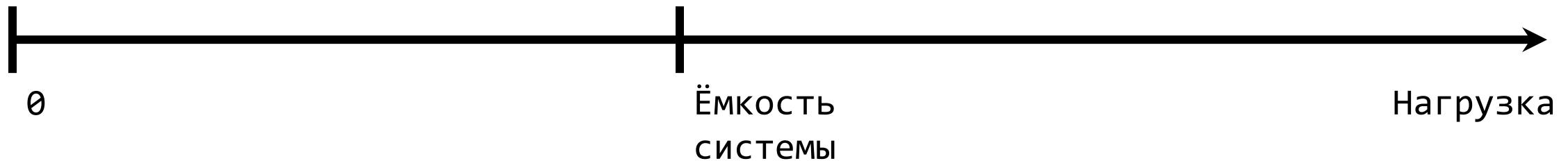
SPIKE TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ТЕСТИРОВАНИЕ НА ВЫНОСЛИВОСТЬ

ENDURANCE TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ТЕСТИРОВАНИЕ НА ВЫНОСЛИВОСТЬ

ENDURANCE TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ТЕСТИРОВАНИЕ НА ВЫНОСЛИВОСТЬ

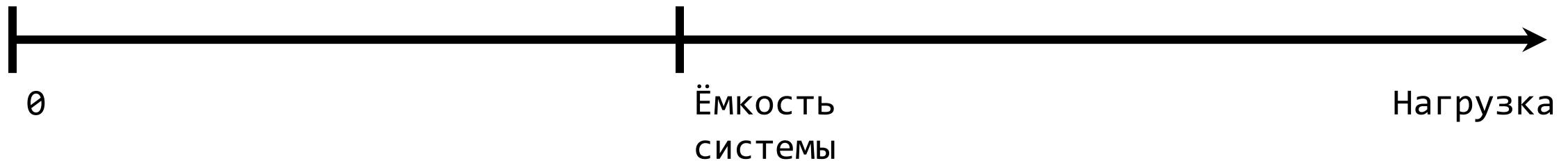
ENDURANCE TESTING
SOAK TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ОБЪЁМНОЕ ТЕСТИРОВАНИЕ

VOLUME TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ОБЪЁМНОЕ ТЕСТИРОВАНИЕ

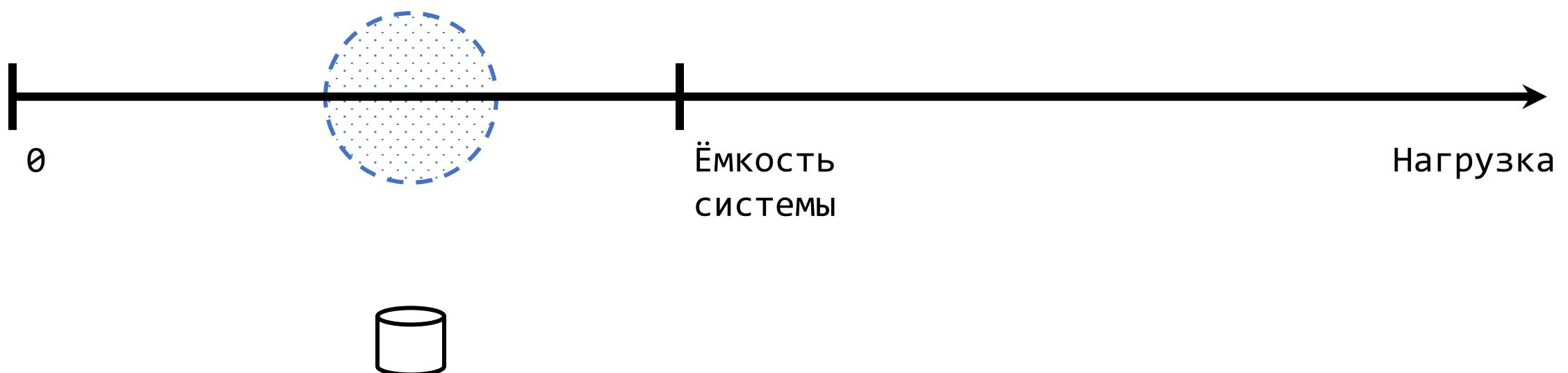
VOLUME TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ОБЪЁМНОЕ ТЕСТИРОВАНИЕ

VOLUME TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ОБЪЁМНОЕ ТЕСТИРОВАНИЕ

VOLUME TESTING



PERFORMANCE-ТЕСТИРОВАНИЕ

ОБЪЁМНОЕ ТЕСТИРОВАНИЕ

VOLUME TESTING



ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

— APACHE JMETER



ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

— APACHE JMETER

- * НАПИСАН НА JAVA
- * GUI
- * СКРИПТЫ НА GROOVY
 - + ДРУГИЕ ЯЗЫКИ

ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
- GATLING



ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
 - * НАПИСАН НА SCALA + JAVA
- GATLING
 - * СКРИПТЫ НА SCALA + KOTLIN + JAVA

ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter**
- GATLING**
- YANDEX TANK**



ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
- GATLING
- YANDEX TANK
 - * НАПИСАН НА PYTHON
 - * ИСПОЛЬЗУЕТ РАЗНЫЕ ГЕНЕРАТОРЫ НАГРУЗКИ:
 - PANDORA (GO)
 - PHANTOM (AMMO)
 - JMETER

ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
- GATLING
- YANDEX TANK
- K6



ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
 - GATLING
 - YANDEX TANK
 - K6
- * НАПИСАН НА GO
 - * СКРИПТЫ НА JS

ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
- GATLING
- YANDEX TANK
- K6
- LOCUST



ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
- GATLING
- YANDEX TANK
- K6
- LOCUST
 - * НАПИСАН НА PYTHON
 - * СКРИПТЫ НА PYTHON

ИНСТРУМЕНТЫ ДЛЯ LOAD TESTING

- APACHE JMeter
- GATLING
- YANDEX TANK
- K6
- LOCUST

...

APACHE JMETER VS GATLING

APACHE JMeter VS GATLING

— JVM

— JVM

APACHE JMeter VS GATLING

- JVM
- блокирующий IO
- JVM
- неблокирующий IO

APACHE JMeter VS GATLING

- JVM
- БЛОКИРУЮЩИЙ IO
ApacheHttpClient
- JVM
- НЕБЛОКИРУЮЩИЙ IO
Netty

APACHE JMETER VS GATLING

- JVM
- БЛОКИРУЮЩИЙ ИО
ApacheHttpClient
- VIRTUAL USER
 \Leftrightarrow THREAD
- JVM
- НЕБЛОКИРУЮЩИЙ ИО
Netty
- VIRTUAL USER
 \Leftrightarrow “MESSAGE”

APACHE JMETER VS GATLING

- JVM
- БЛОКИРУЮЩИЙ ИО
ApacheHttpClient
- VIRTUAL USER
 \Leftrightarrow THREAD
- JVM
- НЕБЛОКИРУЮЩИЙ ИО
Netty
- VIRTUAL USER
 \Leftrightarrow “MESSAGE”
Akka

**ВЫДУМАННЫЕ
НЕВЫДУМАННЫЕ
ИСТОРИИ**

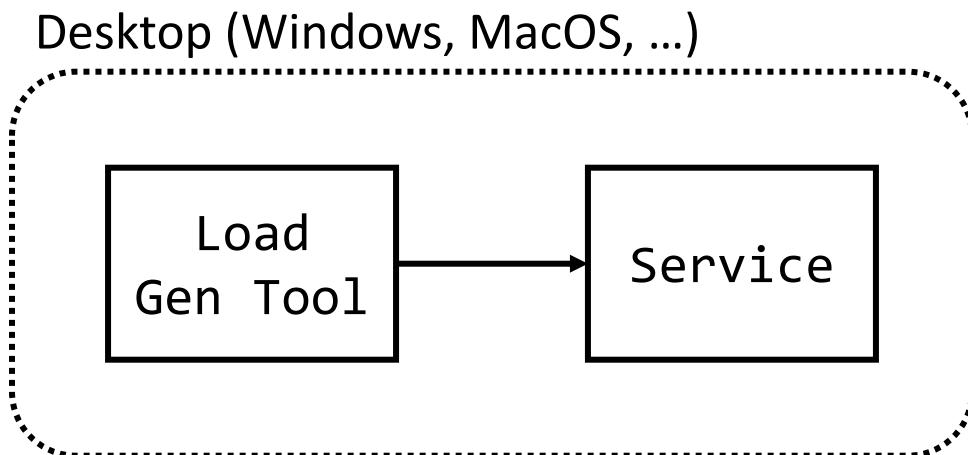
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

**ПОЧЕМУ НЕ НАДО ЗАПУСКАТЬ НАГРУЗОЧНЫЕ
ТЕСТЫ НА МАШИНЕ РАЗРАБОТЧИКА?**

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

ПОЧЕМУ НЕ НАДО ЗАПУСКАТЬ НАГРУЗОЧНЫЕ
ТЕСТЫ НА МАШИНЕ РАЗРАБОТЧИКА?



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

**ПОЧЕМУ НЕ НАДО ЗАПУСКАТЬ НАГРУЗОЧНЫЕ
ТЕСТЫ НА МАШИНЕ РАЗРАБОТЧИКА?**

1. SANDBOX (ANTIVIRUS, WINDOWS DEFENDER)

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

ПОЧЕМУ НЕ НАДО ЗАПУСКАТЬ НАГРУЗОЧНЫЕ
ТЕСТЫ НА МАШИНЕ РАЗРАБОТЧИКА?

1. SANDBOX (ANTIVIRUS, WINDOWS DEFENDER)
2. КУЧА СТОРОННЕГО СОФТА

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

ПОЧЕМУ НЕ НАДО ЗАПУСКАТЬ НАГРУЗОЧНЫЕ
ТЕСТЫ НА МАШИНЕ РАЗРАБОТЧИКА?

1. SANDBOX (ANTIVIRUS, WINDOWS DEFENDER)
2. КУЧА СТОРОННЕГО СОФТА
3. РАЗНЫЕ НАСТРОЙКИ

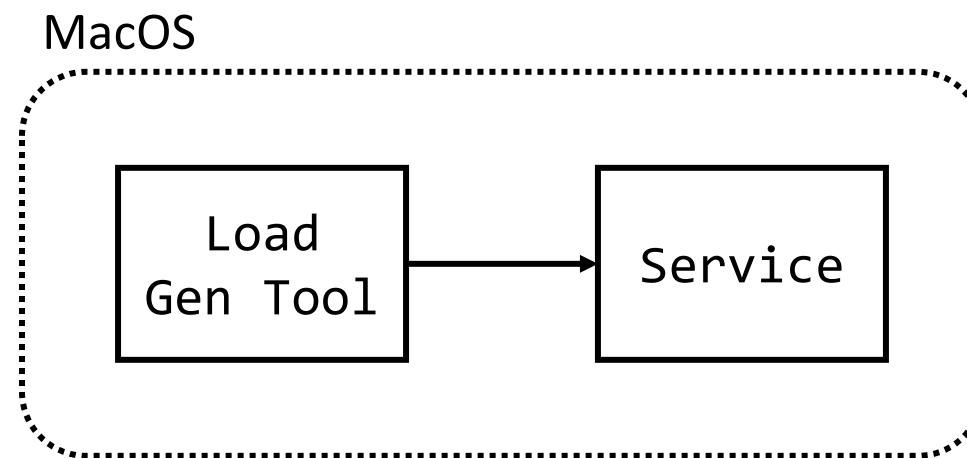
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

ПОЧЕМУ НЕ НАДО ЗАПУСКАТЬ НАГРУЗОЧНЫЕ
ТЕСТЫ НА МАШИНЕ РАЗРАБОТЧИКА?

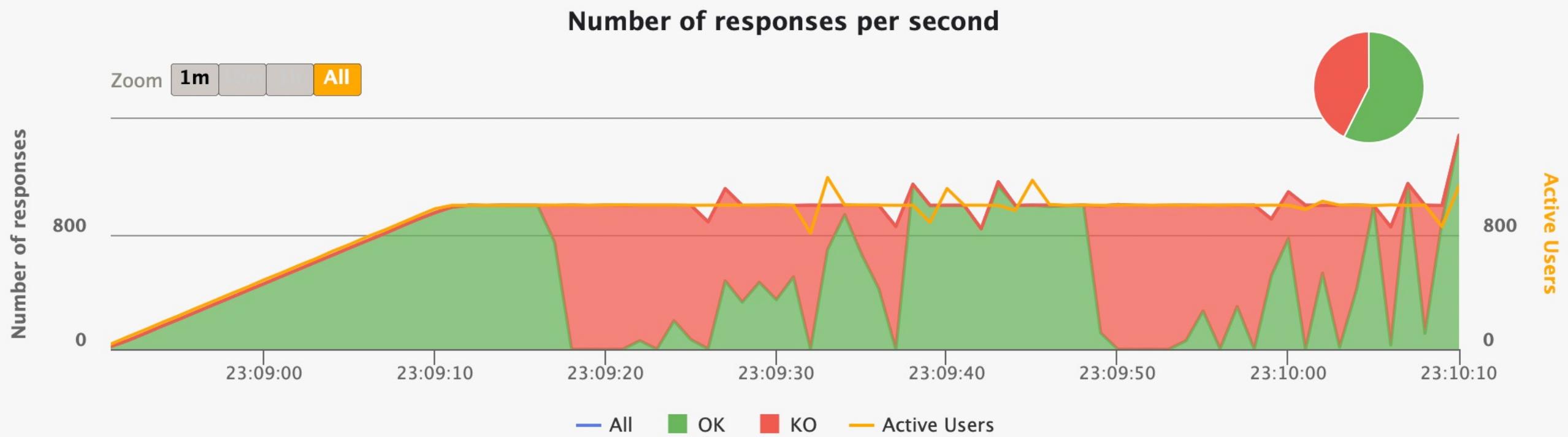
1. SANDBOX (ANTIVIRUS, WINDOWS DEFENDER)
2. КУЧА СТОРОННЕГО СОФТА
3. РАЗНЫЕ НАСТРОЙКИ

j.n.SocketException: Too many open files

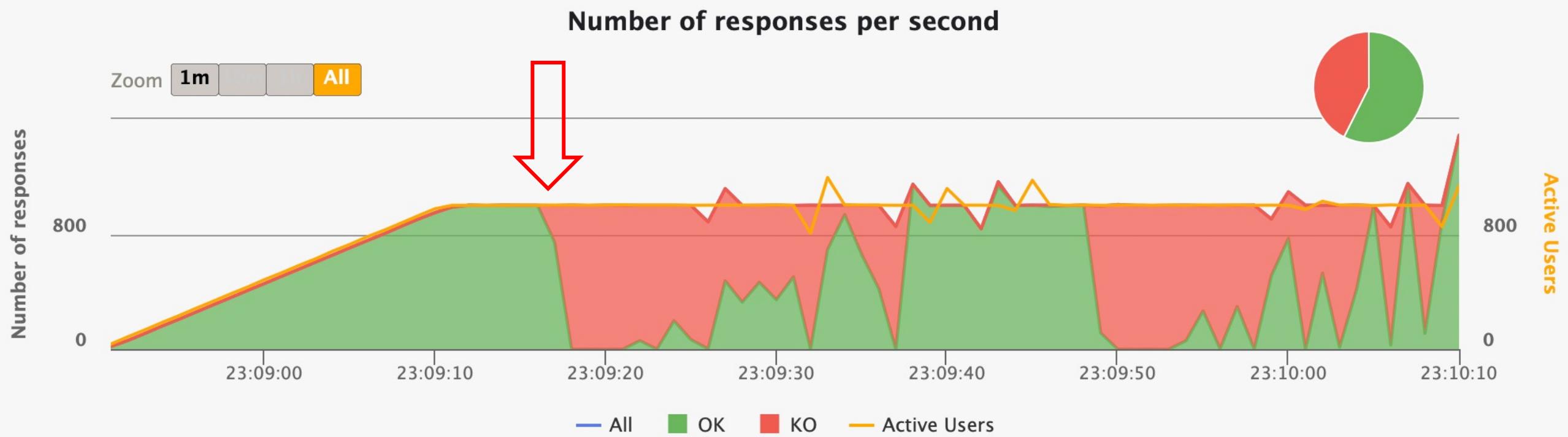
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



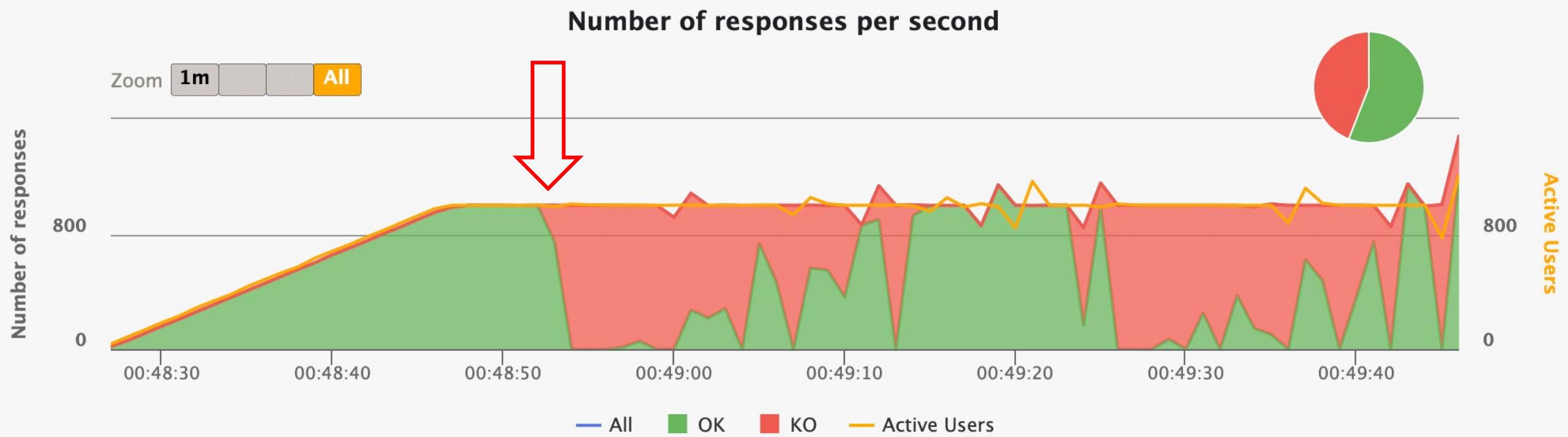
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



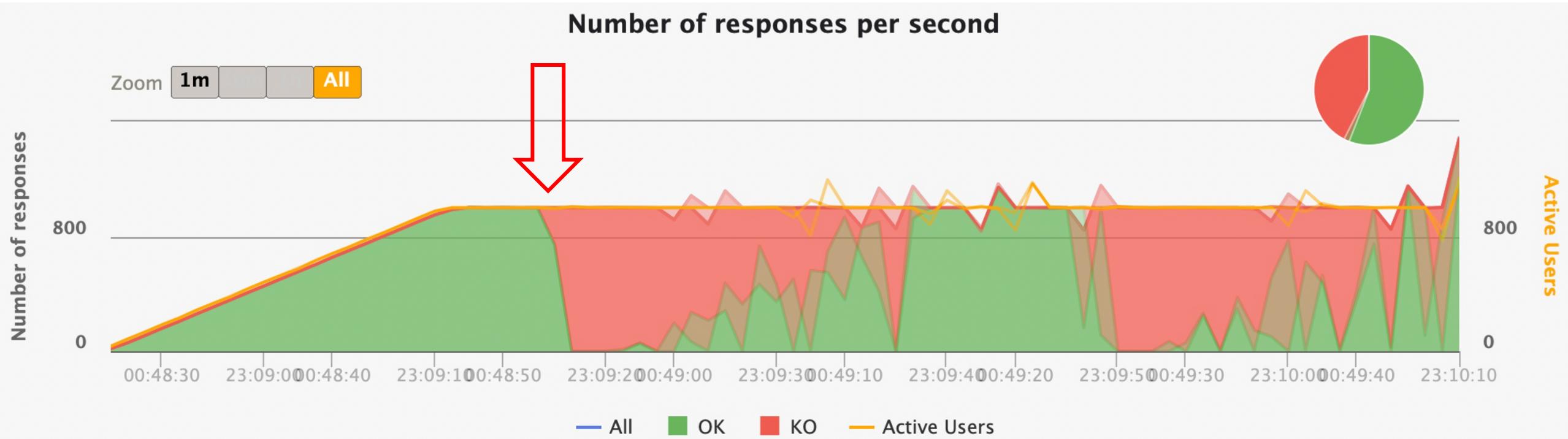
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



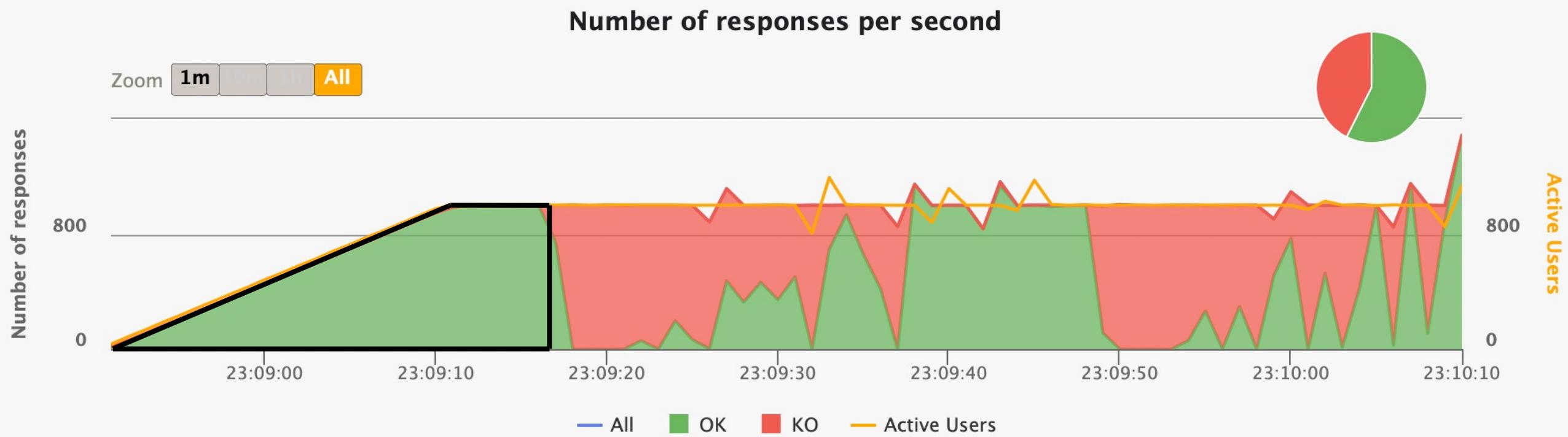
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



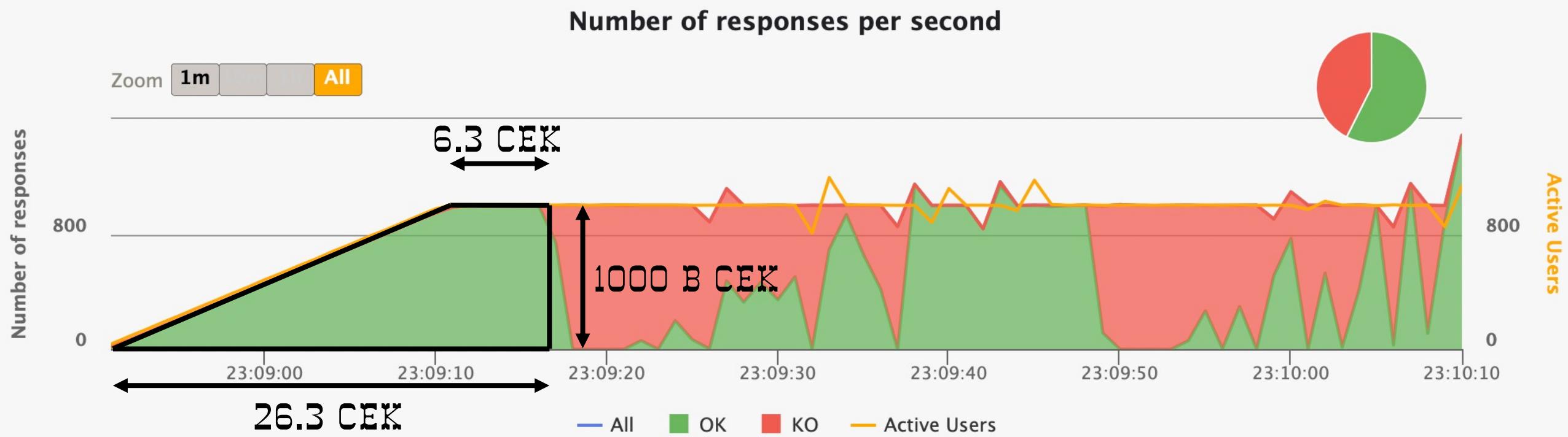
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



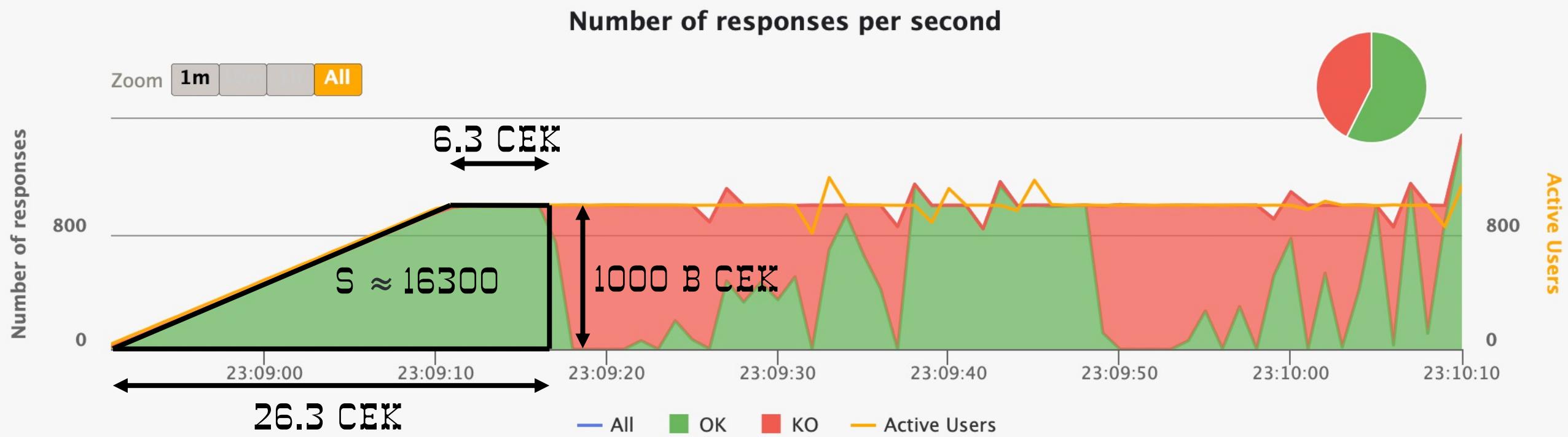
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

```
% sysctl net.inet.ip.portrange  
net.inet.ip.portrange.hilast: 65535  
net.inet.ip.portrange.hifirst: 49152  
net.inet.ip.portrange.last: 65535  
net.inet.ip.portrange.first: 49152
```

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

```
% sysctl net.inet.ip.portrange  
net.inet.ip.portrange.hilast: 65535  
net.inet.ip.portrange.hifirst: 49152  
net.inet.ip.portrange.last: 65535  
net.inet.ip.portrange.first: 49152
```

$$65535 - 49152 = 16383$$

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

```
% sysctl net.inet.ip.portrange  
net.inet.ip.portrange.hilast: 65535  
net.inet.ip.portrange.hifirst: 49152  
net.inet.ip.portrange.last: 65535  
net.inet.ip.portrange.first: 49152
```

TIME_WAIT

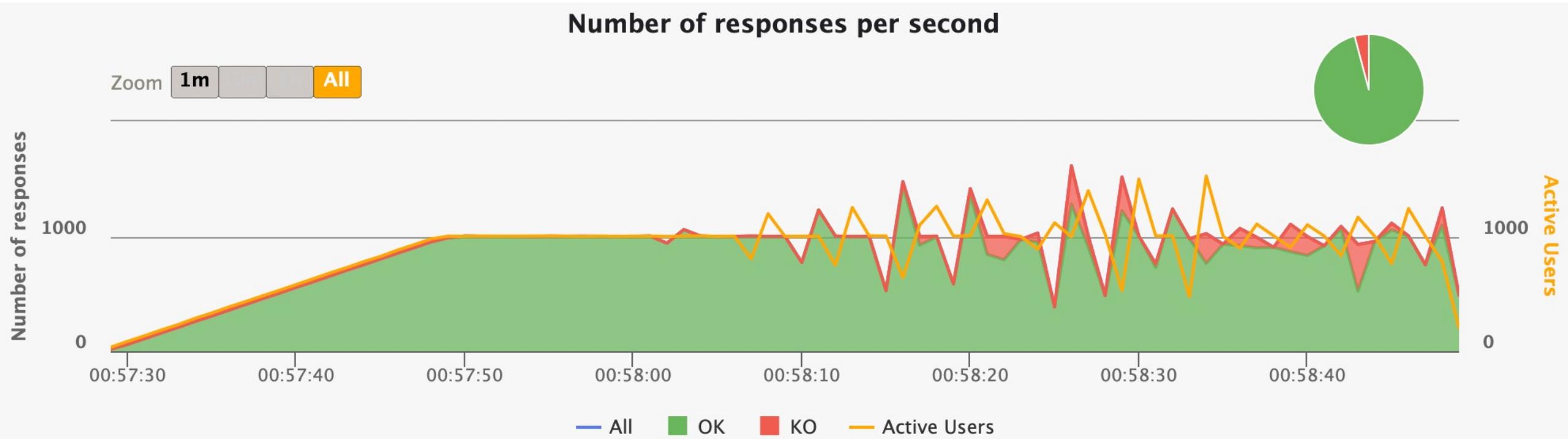
$$65535 - 49152 = 16383$$

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

```
% sysctl net.inet.ip.portrange  
net.inet.ip.portrange.hilast: 65535  
net.inet.ip.portrange.hifirst: 49152 -> 32768  
net.inet.ip.portrange.last: 65535  
net.inet.ip.portrange.first: 49152 -> 32768
```

$$65535 - 32768 = 32767$$

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

ПОЧЕМУ НЕ НАДО ЗАПУСКАТЬ НАГРУЗОЧНЫЕ
ТЕСТЫ НА МАШИНЕ РАЗРАБОТЧИКА?

1. SANDBOX (ANTIVIRUS, WINDOWS DEFENDER)
2. КУЧА СТОРОННЕГО СОФТА
3. РАЗНЫЕ НАСТРОЙКИ
4. «СЛАБОЕ» ЖЕЛЕЗО

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

**НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ –
ЭТО ВЫСОКАЯ УТИЛИЗАЦИЯ РЕСУРСОВ**

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ –

ЭТО ВЫСОКАЯ ЧУТИЛИЗАЦИЯ РЕСУРСОВ

И ДЛЯ СЕРВИСА,

И ДЛЯ ГЕНЕРАТОРА НАГРУЗКИ

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

JMETER:

- 1 VU = 1 THREAD
- 1 THREAD = 2 MB RAM

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

JMETER:

- 1 VU = 1 THREAD
- 1 THREAD = 2 MB RAM

4 000 VU = 8 000 MB RAM

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

НА СЕРВЕРЕ 32 GB RAM:

- OS ~ 1 GB
- SERVICE ~ 8 GB
- JMeter ~ 10 GB (2 + 8)

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

НА СЕРВЕРЕ 32 GB RAM:

- OS ~ 1 GB
- SERVICE ~ 8 GB
- JMeter ~ 10 GB (2 + 8)

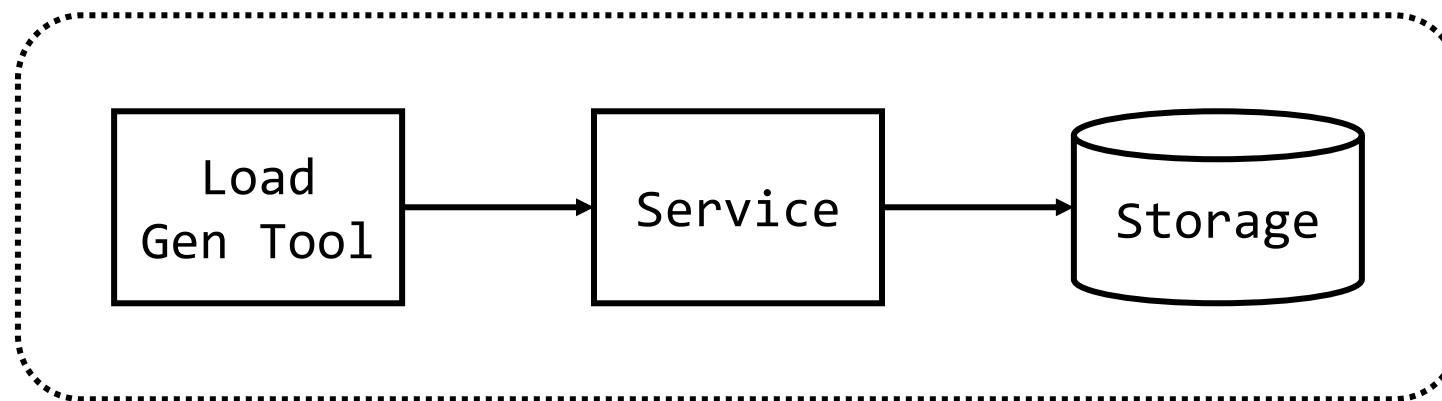
ЗАНЯТО 19 ИЗ 32 GB

ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

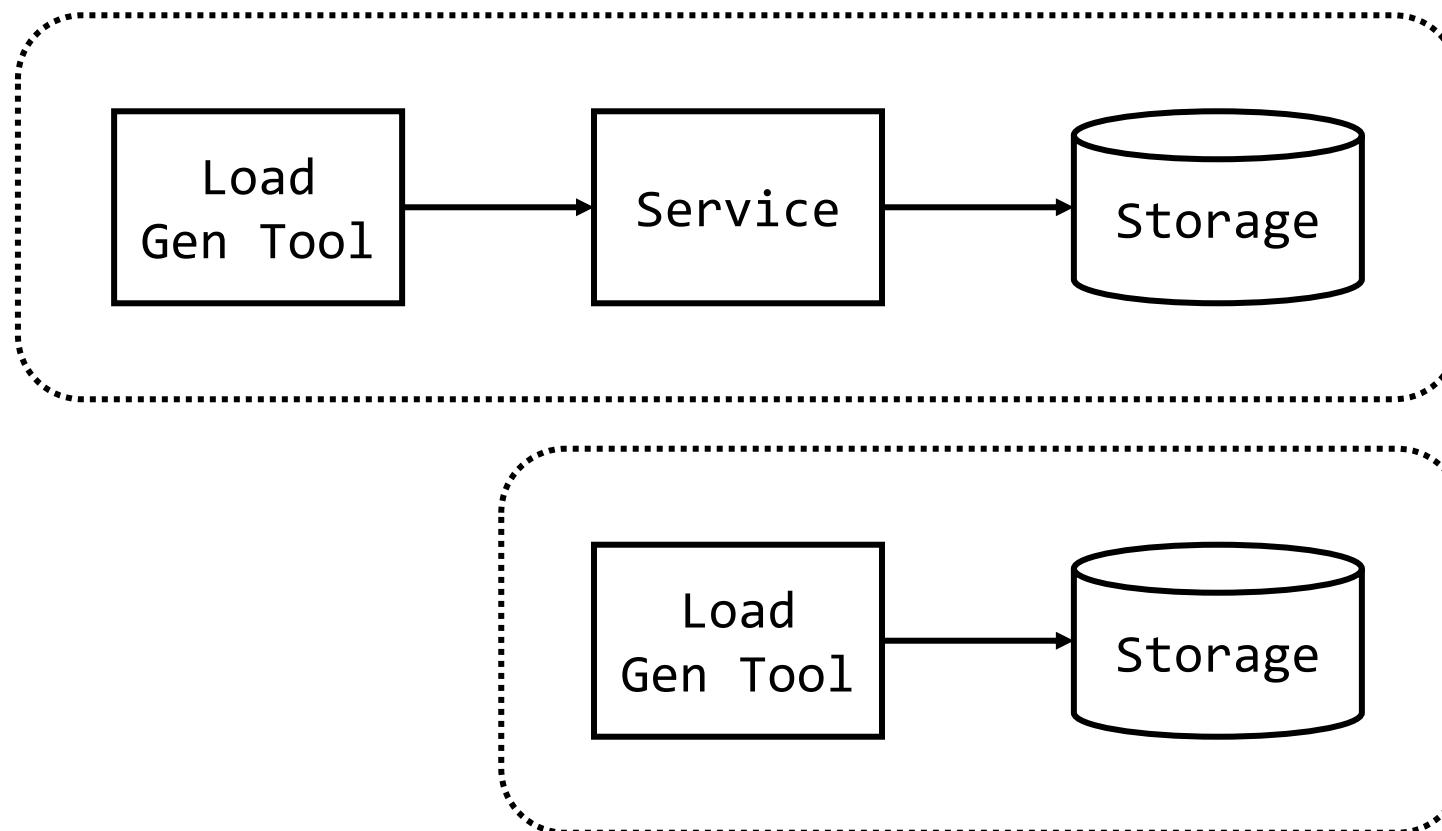
UBUNTU, RHEL 7:

```
$ cat /proc/sys/vm/swappiness  
60
```

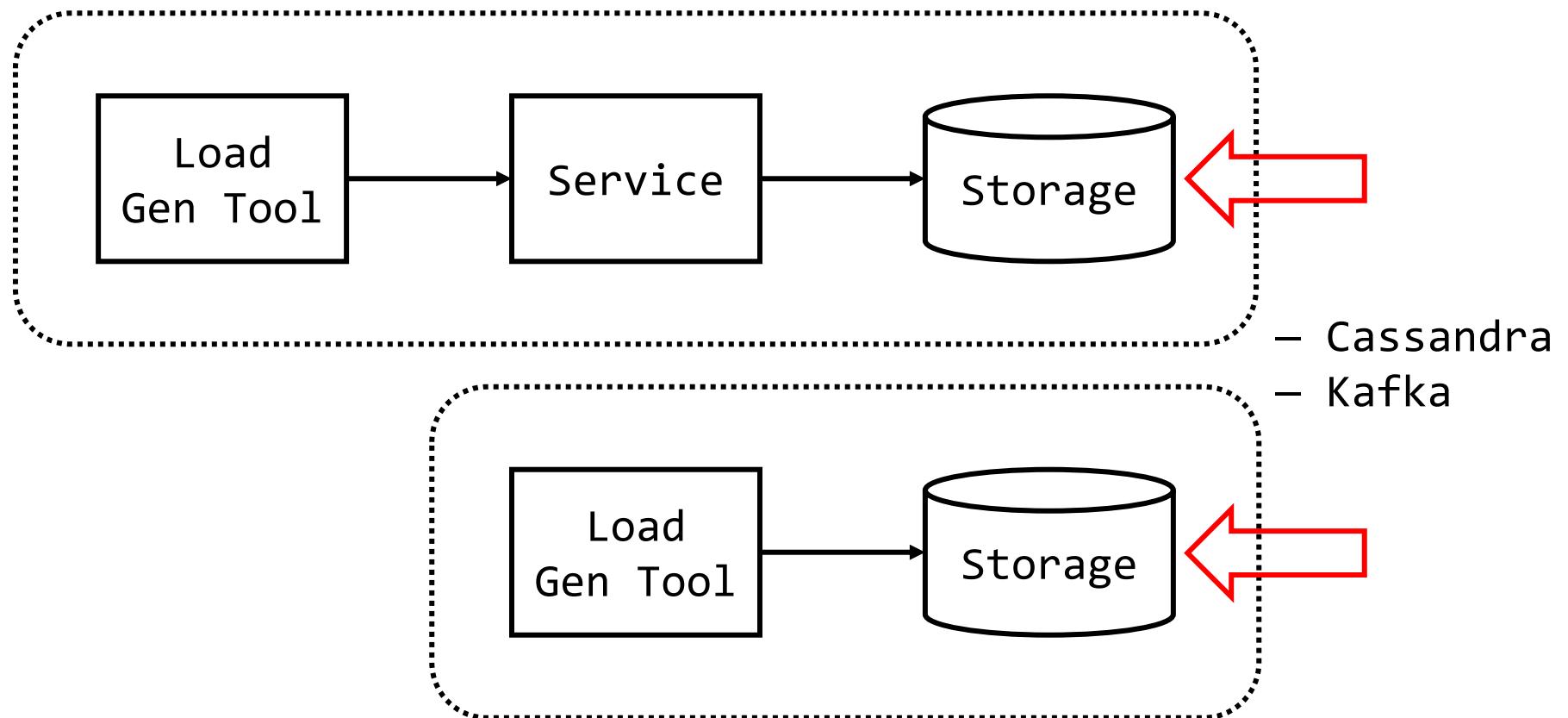
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



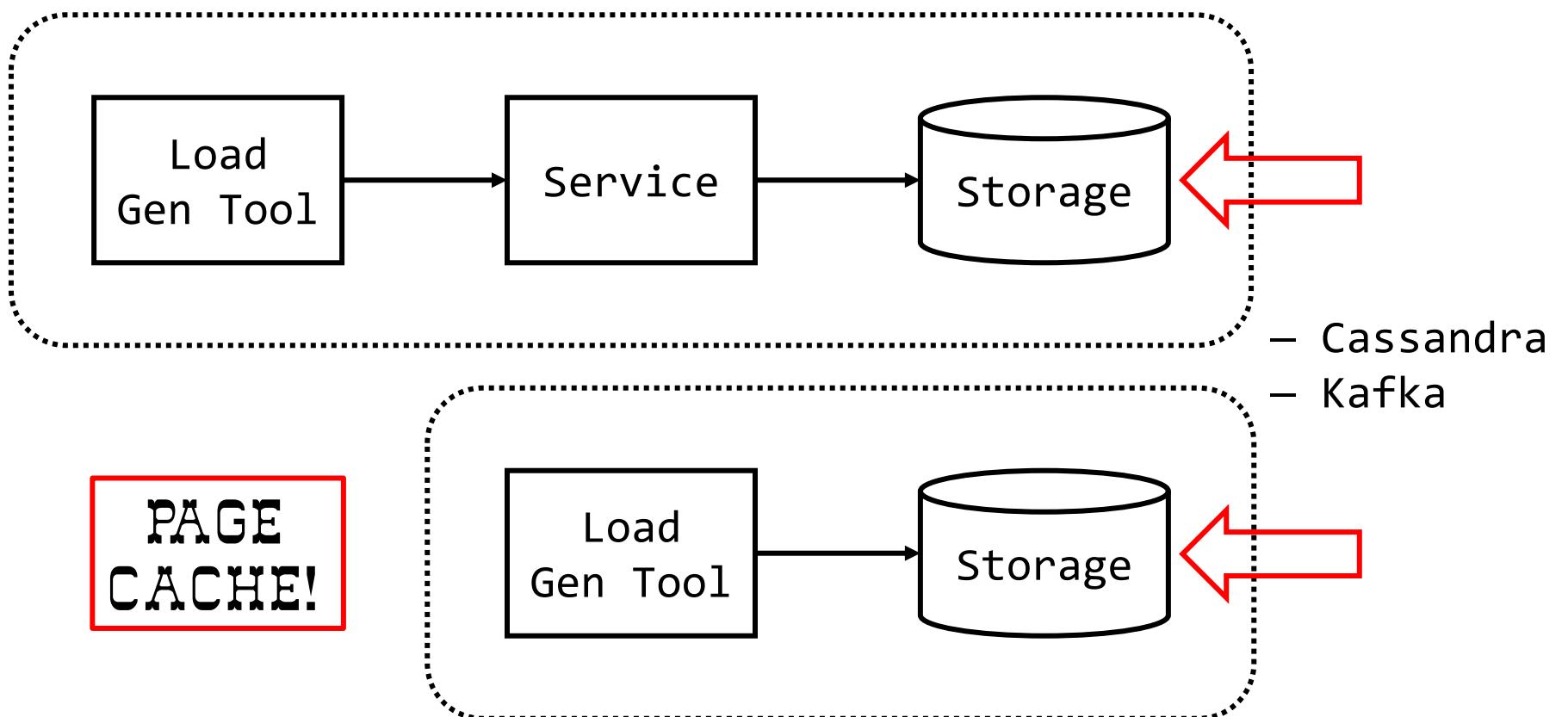
ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ



ОДНАЖДЫ НА СТАРОМ СЕРВЕРЕ

18.04 / 13:45 – 14:30



ДОКЛАД

Зал 2

Когда всё пошло по Кафке 3: где заканчивается Apache Kafka и начинает работу Consumer

Третий доклад из серии, в котором поговорим об устройстве и работе Consumer. Подробно разберем KIP, которые в значительной степени повлияли на работу Consumer. Покрутим настройки Kafka и Consumer.



Григорий Кошелев

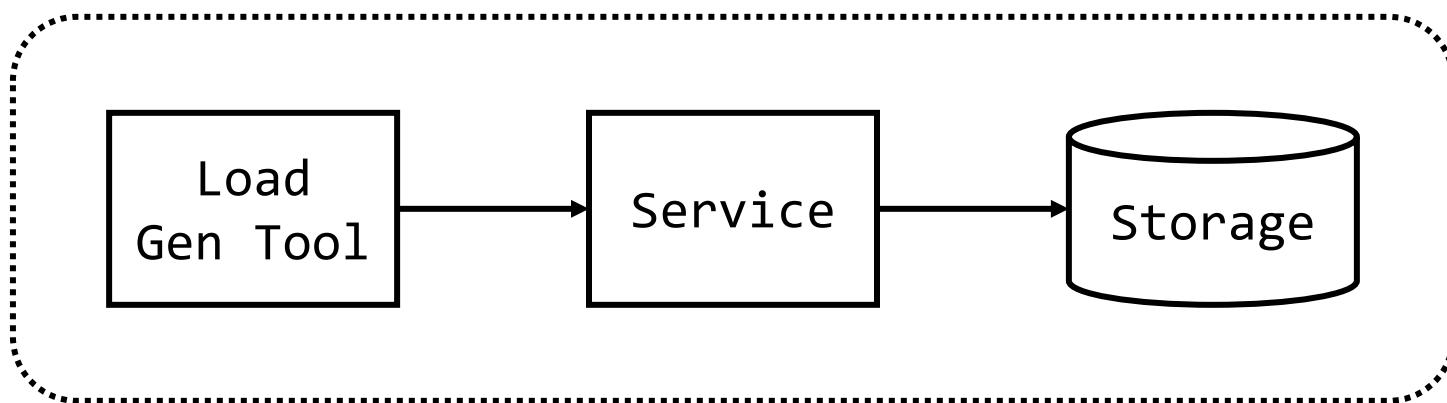
Контур

RU

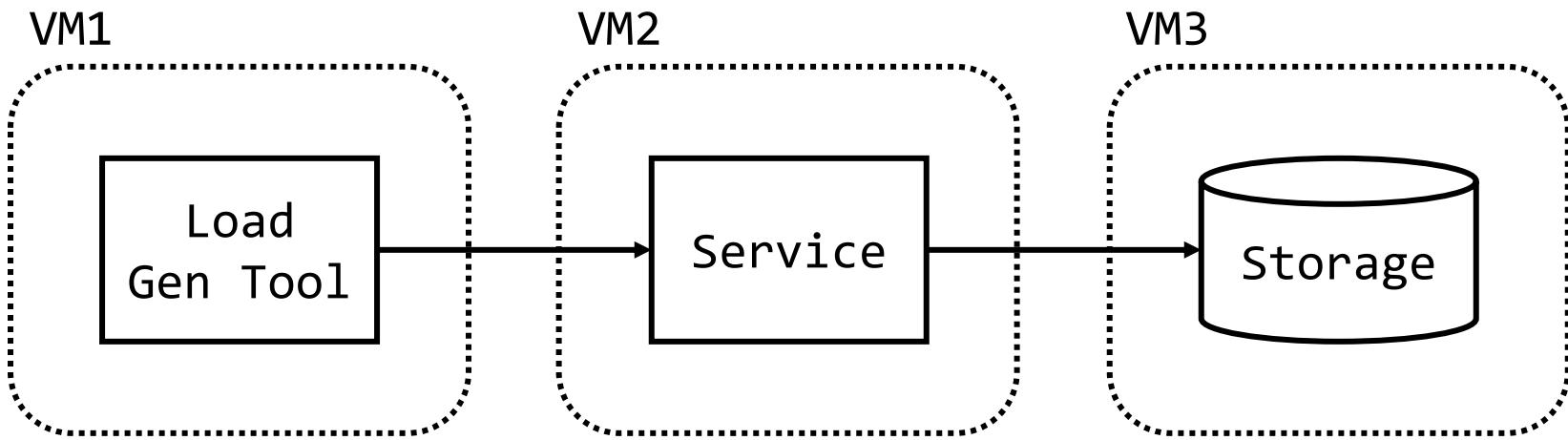


Working with data

Ровно в полночь



РОВНО В ПОЛНОЧЬ



РОВНО В ПОЛНОЧЬ

№ Результата + версия	Результат в Попугаях
Вечерний (18:30), v.1.0	38.0

РОВНО В ПОЛНОЧЬ

№ Результата + версия	Результат в Попугаях
Вечерний (18:30), v.1.0	38.0
Прогноз, v.2.0	42.0

РОВНО В ПОЛНОЧЬ

№ Результата + версия	Результат в Попугаях
Вечерний (18:30), v.1.0	38.0
Прогноз, v.2.0	42.0

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	Результат в Попугаях
Вечерний (18:30), v.1.0	38.0
Прогноз, v.2.0	42.0
Ночной (00:10), v.2.0	36.6

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№	Имя	Результат в Попугаях
V	v.1.0	38.0
	v.2.0	42.0
	v.2.0	36.6



БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	Результат в Попугаях
Вечерний (18:30), v.1.0	38.0
Прогноз, v.2.0	42.0
Ночной (00:10), v.2.0	36.6
Дневной (12:00), v.2.0	40.4

БОЛЬШЕ – ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	П		
Вечерний (18:30), v.1.0	38.0		
Прогноз, v.2.0	42.0		
Ночной (00:10), v.2.0	36.6		
Дневной (12:00), v.2.0	40.4		

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	П		
Вечерний (18:30), v.1.0	38.0	100.0%	
Прогноз, v.2.0	42.0		
Ночной (00:10), v.2.0	36.6		
Дневной (12:00), v.2.0	40.4		

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	П		
Вечерний (18:30), v.1.0	38.0	100.0%	
Прогноз, v.2.0	42.0	+10.5%	
Ночной (00:10), v.2.0	36.6		
Дневной (12:00), v.2.0	40.4		

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	П		
Вечерний (18:30), v.1.0	38.0	100.0%	
Прогноз, v.2.0	42.0	+10.5%	
Ночной (00:10), v.2.0	36.6	-3.7%	
Дневной (12:00), v.2.0	40.4		

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	П		
Вечерний (18:30), v.1.0	38.0	100.0%	
Прогноз, v.2.0	42.0	+10.5%	
Ночной (00:10), v.2.0	36.6	-3.7%	
Дневной (12:00), v.2.0	40.4	+6.3%	

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	П		
Вечерний (18:30), v.1.0	38.0	100.0%	
Прогноз, v.2.0	42.0	+10.5%	100.0%
Ночной (00:10), v.2.0	36.6	-3.7%	
Дневной (12:00), v.2.0	40.4	+6.3%	

БОЛЬШЕ — ЛУЧШЕ!

РОВНО В ПОЛНОЧЬ

№ Результата + версия	П		
Вечерний (18:30), v.1.0	38.0	100.0%	
Прогноз, v.2.0	42.0	+10.5%	100.0%
Ночной (00:10), v.2.0	36.6	-3.7%	-12.9%
Дневной (12:00), v.2.0	40.4	+6.3%	-3.8%

БОЛЬШЕ — ЛУЧШЕ!

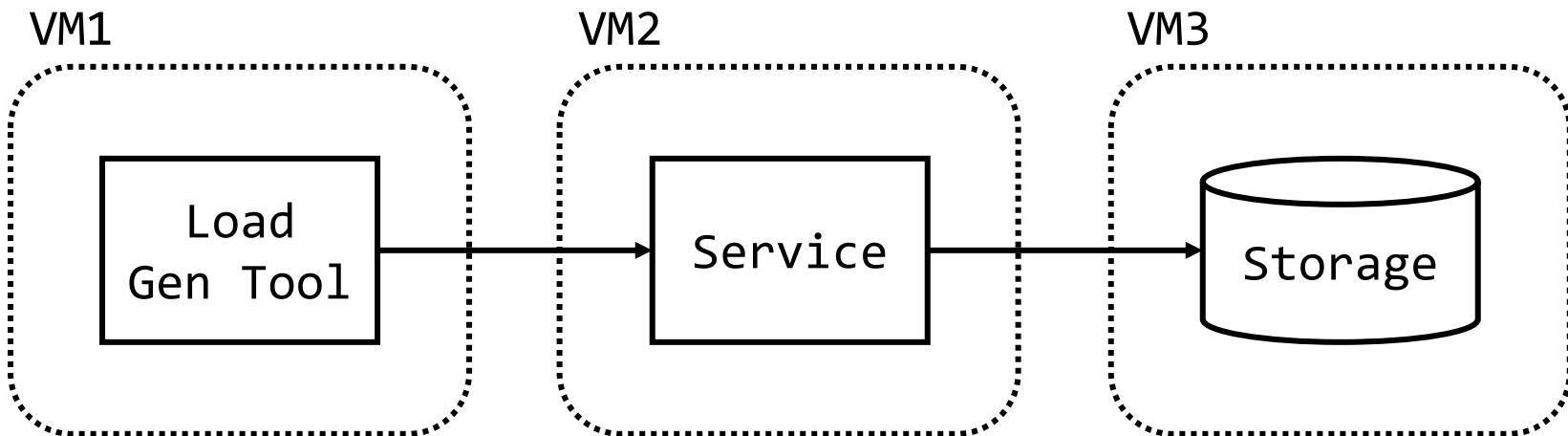


РОВНО В ПОЛНОЧЬ

КАК ТАК ВЫШЛО?

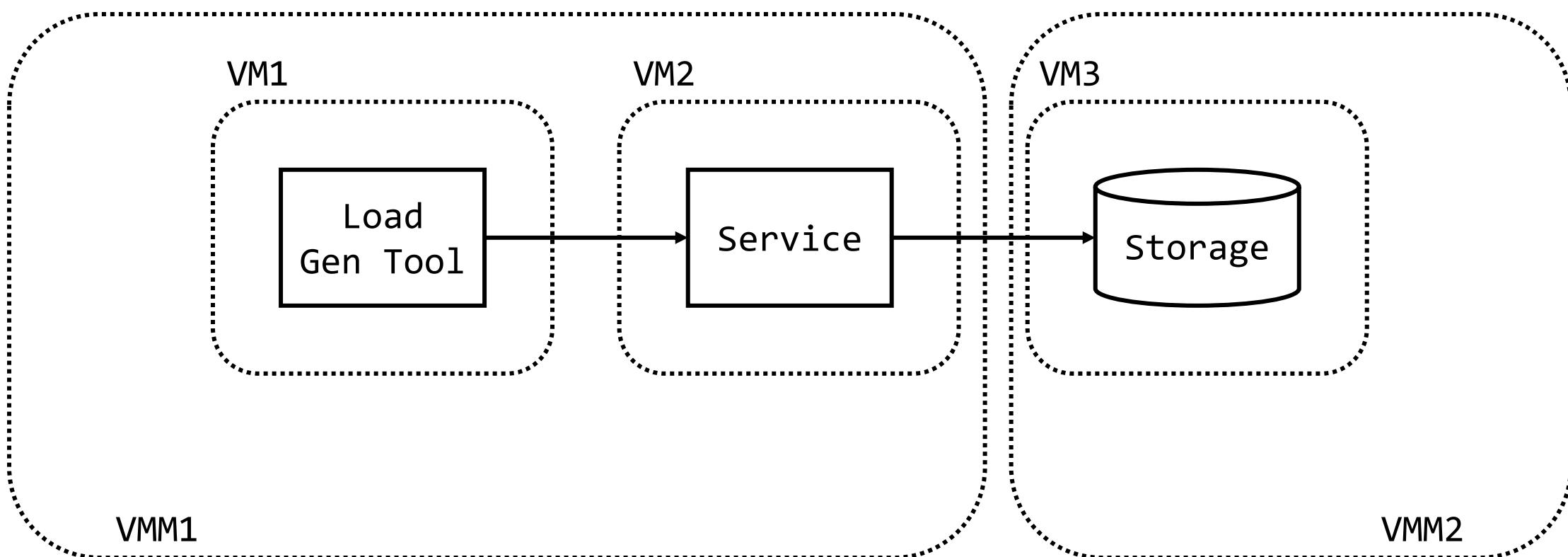
РОВНО В ПОЛНОЧЬ

КАК ТАК ВЫШЛО?



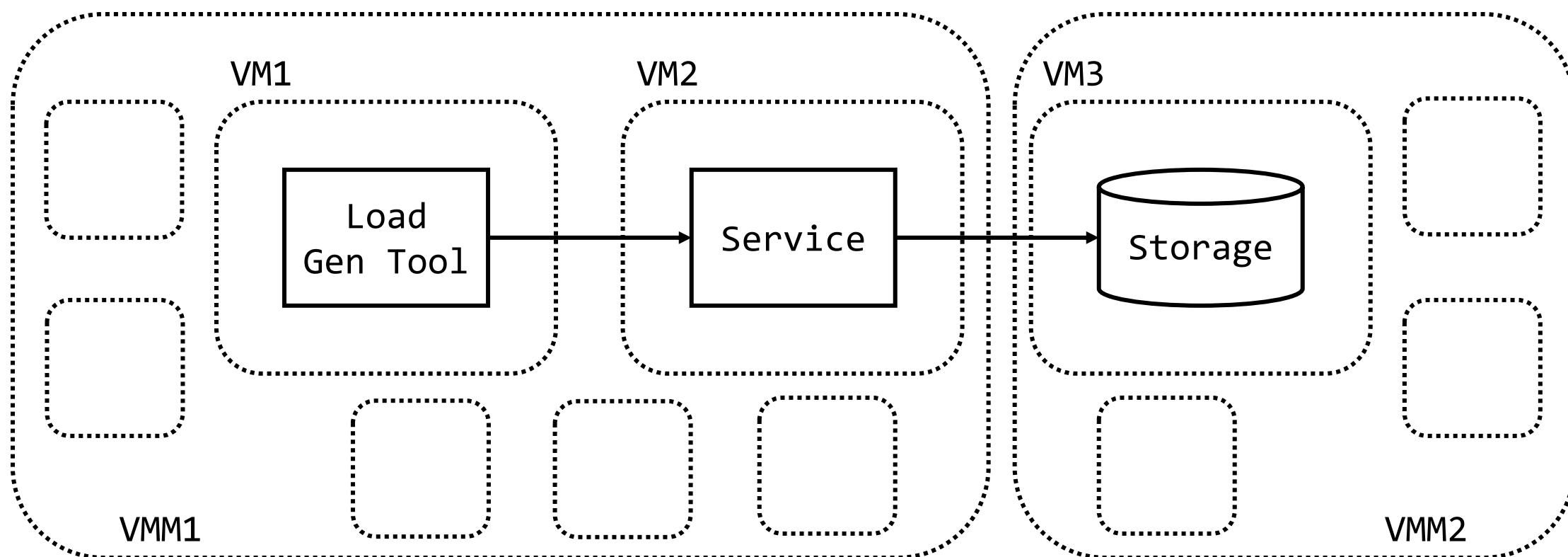
РОВНО В ПОЛНОЧЬ

КАК ТАК ВЫШЛО?



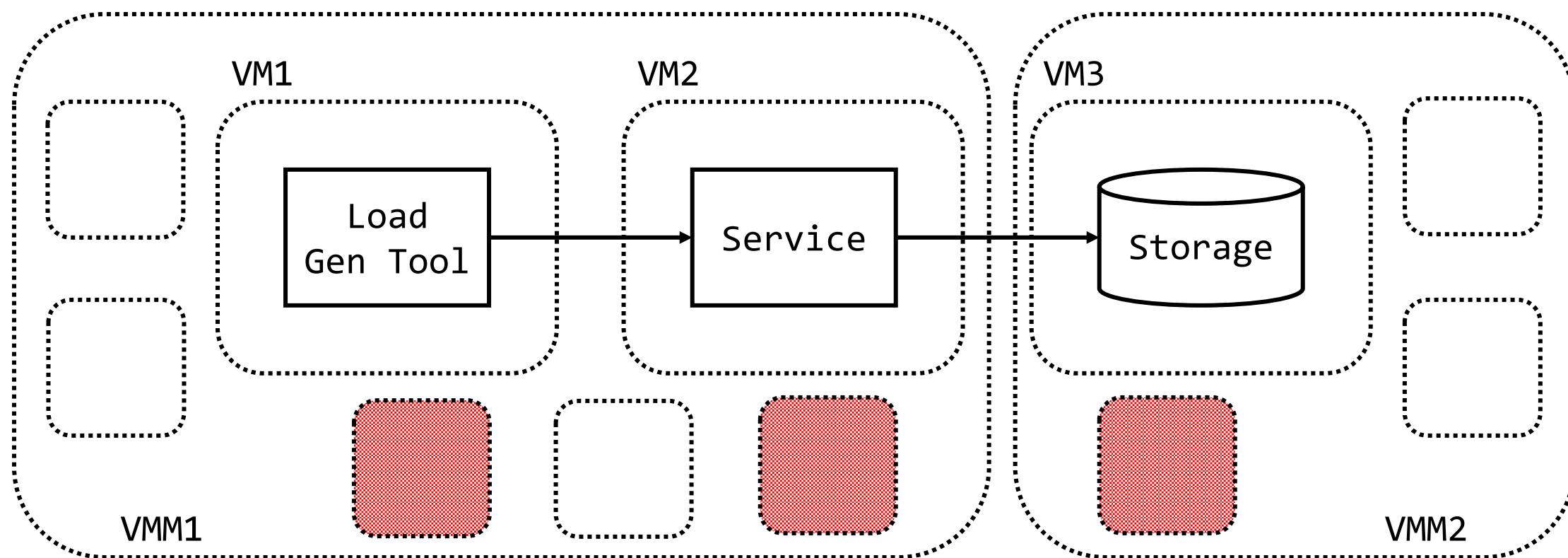
РОВНО В ПОЛНОЧЬ

КАК ТАК ВЫШЛО?

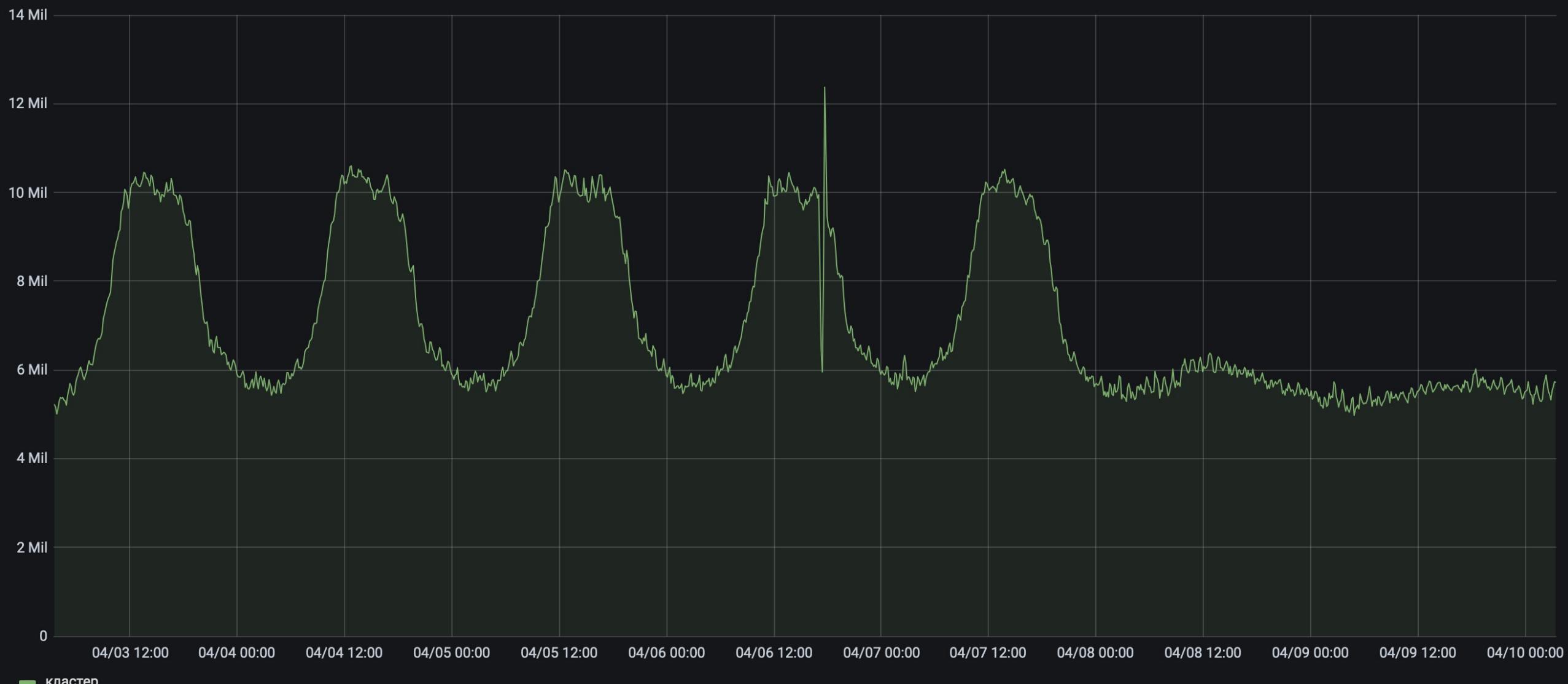


РОВНО В ПОЛНОЧЬ

КАК ТАК ВЫШЛО?

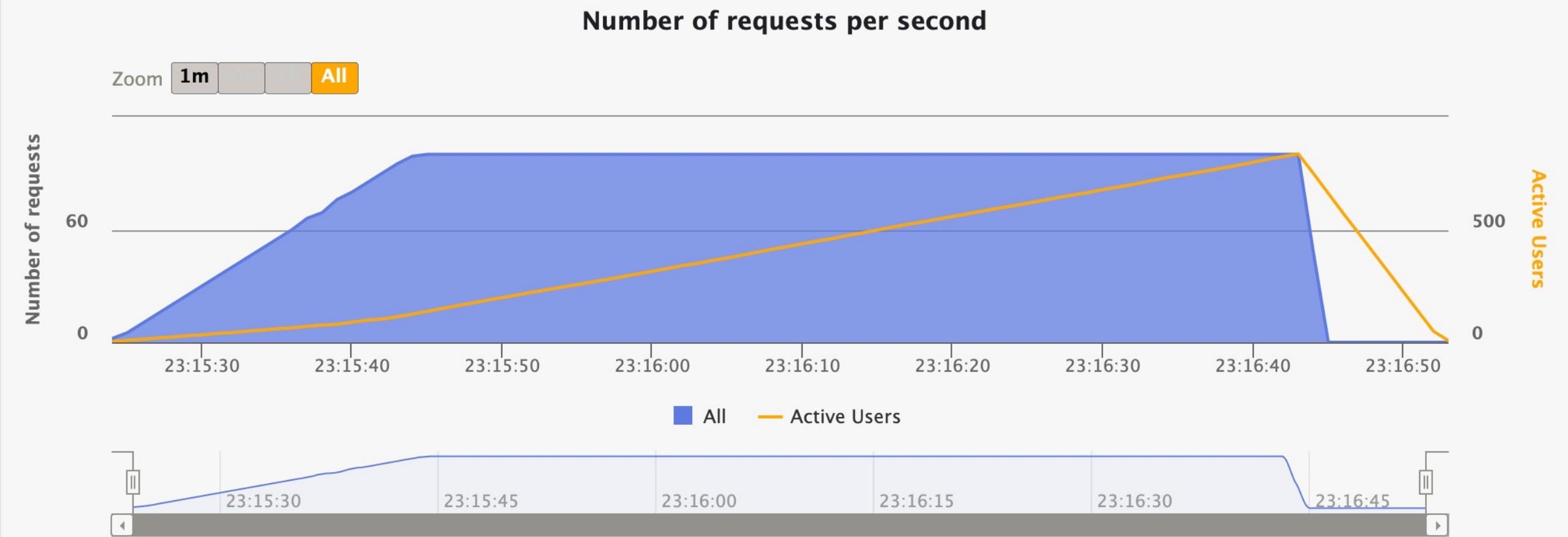


Messages In Per Sec

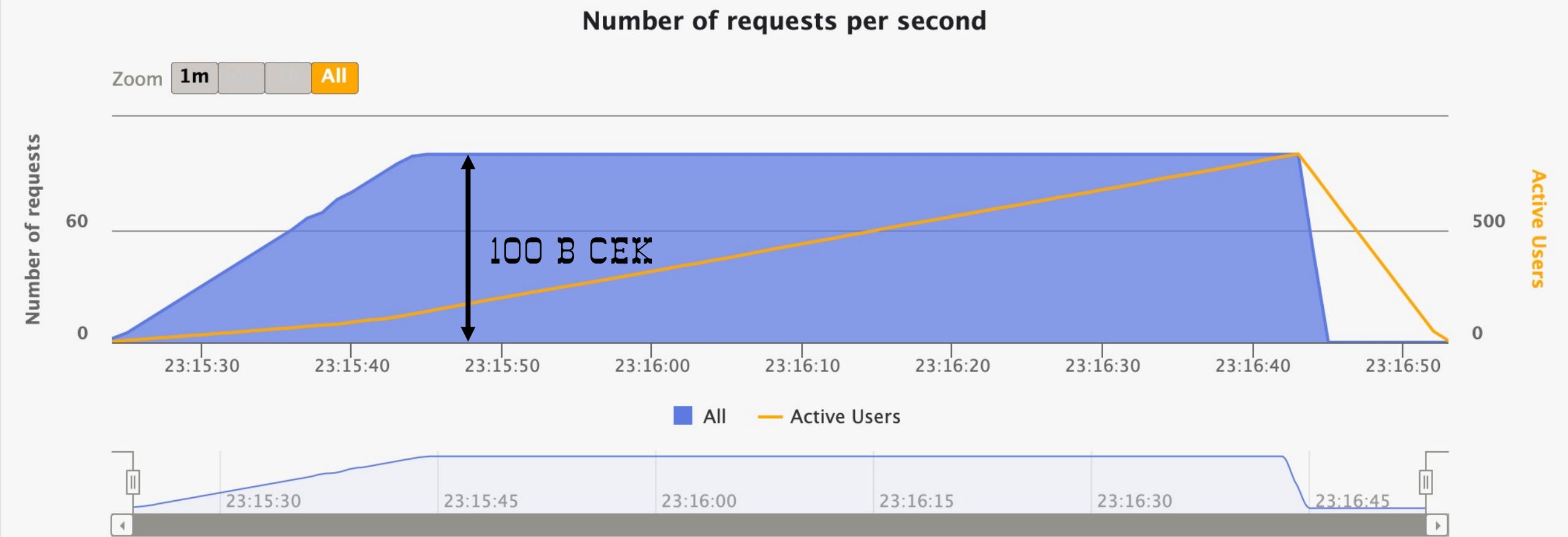


ВЕЛИКА ЛИ ЁМКОСТЬ?

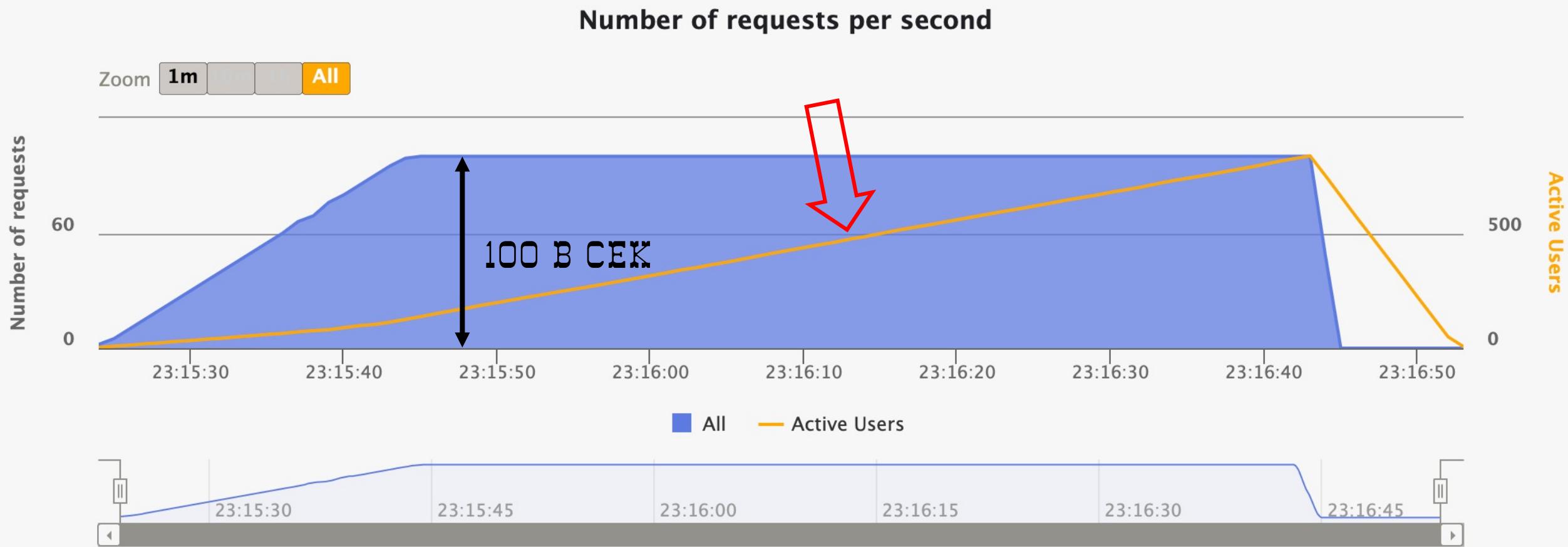
ВЕЛИКА ЛИ ЁМКОСТЬ?



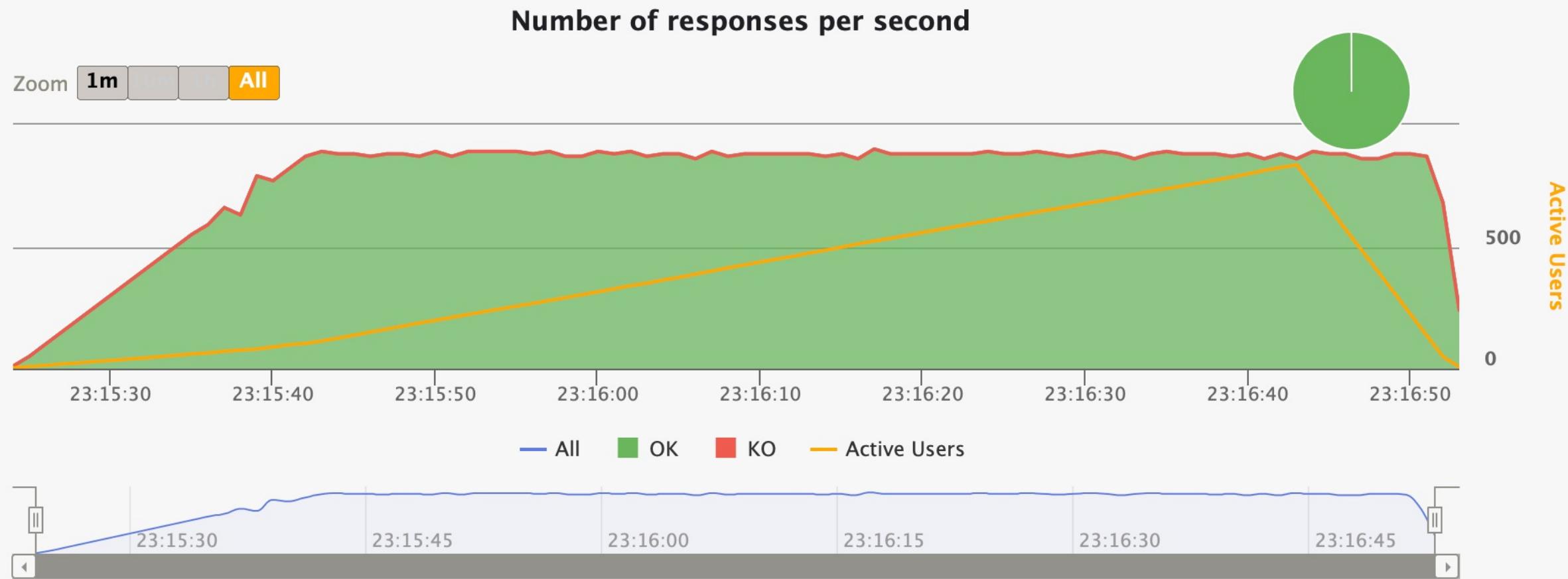
ВЕЛИКА ЛИ ЁМКОСТЬ?



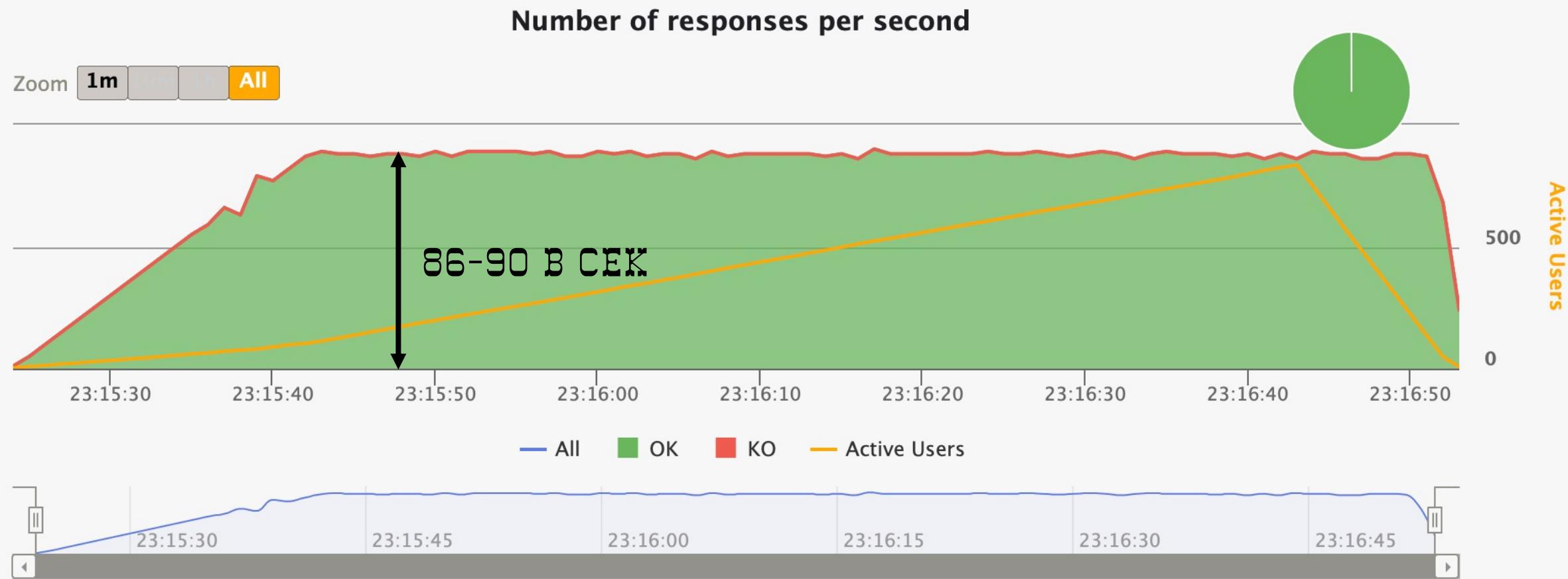
ВЕЛИКА ЛИ ЁМКОСТЬ?



ВЕЛИКА ЛИ ЁМКОСТЬ?



ВЕЛИКА ЛИ ЁМКОСТЬ?



ВЕЛИКА ЛИ ЕМКОСТЬ?

Response Time Percentiles over Time (OK)

Zoom **1m** **1h** **All**

Response Time (ms)

5k
0

23:15:30 23:15:40 23:15:50 23:16:00 23:16:10 23:16:20 23:16:30 23:16:40 23:16:50

min 25% 50% 75% 80% 85% 90% 95% 99% max Active Users

Active Users

500
0



ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 RP\$, НО ЁМКОСТЬ 86-90 RP\$?



ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 RP\$, НО ЁМКОСТЬ 86-90 RP\$?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 RP\$, НО ЁМКОСТЬ 86-90 RP\$?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ **Х**.

ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 RP\$, НО ЁМКОСТЬ 86-90 RP\$?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ **X**.
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.

ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 RP\$, НО ЁМКОСТЬ 86-90 RP\$?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ **X**.
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.
3. GO TO 1.

ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 RP\$, НО ЁМКОСТЬ 86-90 RP\$?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ **X**.
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.
3. GO TO 1.

ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 **XP\$**, НО ЁМКОСТЬ 86-90 RP\$?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ **X**.
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.
3. GO TO 1.



ВЕЛИКА ЛИ ЁМКОСТЬ?

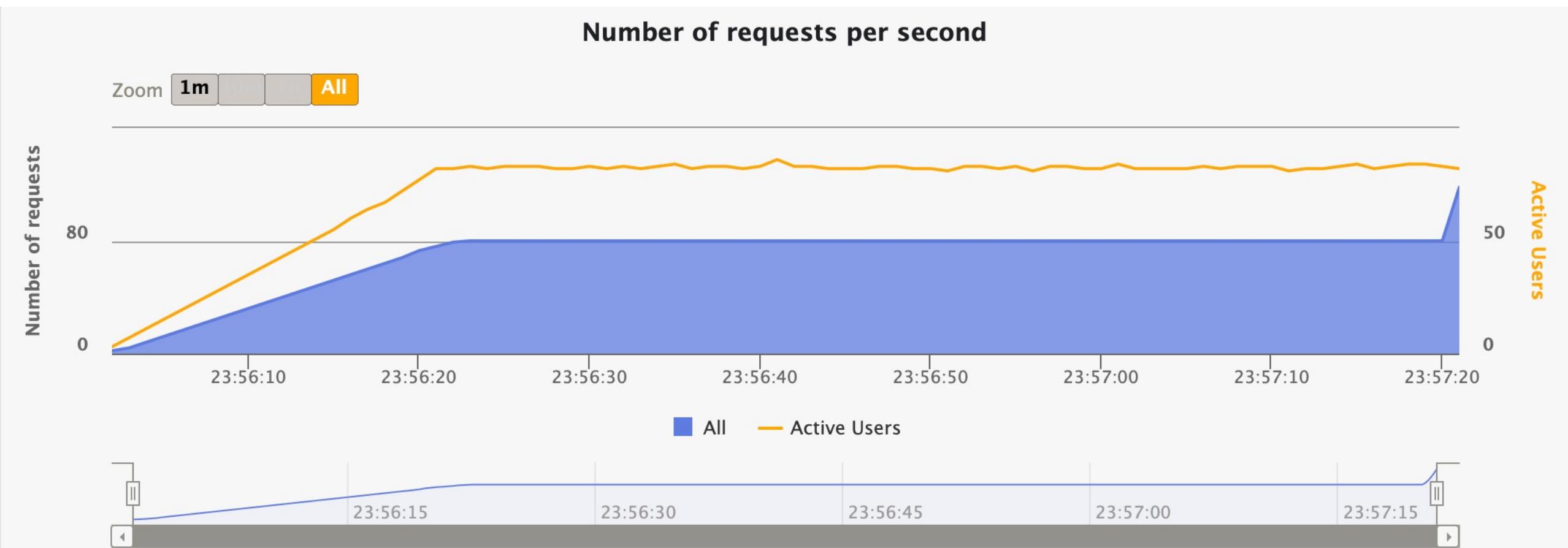
ХОТИМ 100 XP\$, НО ЁМКОСТЬ 86-90 XP\$?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

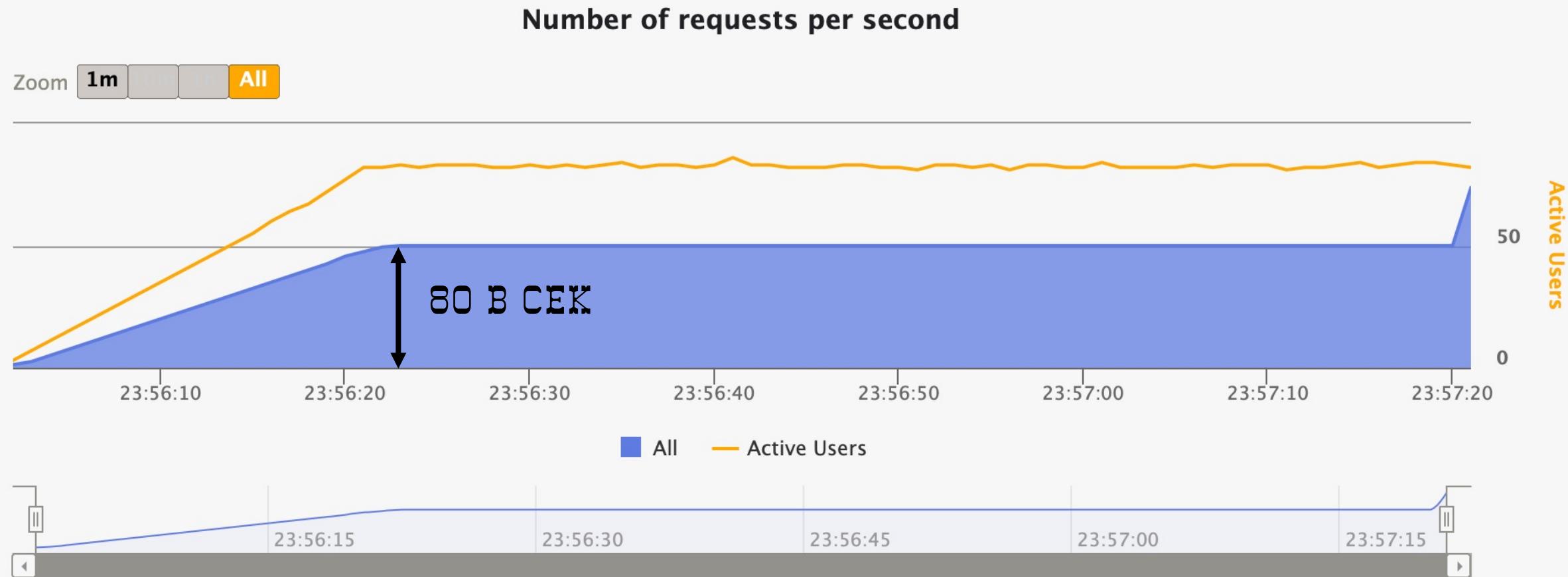
1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ X.
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.
3. GO TO 1.



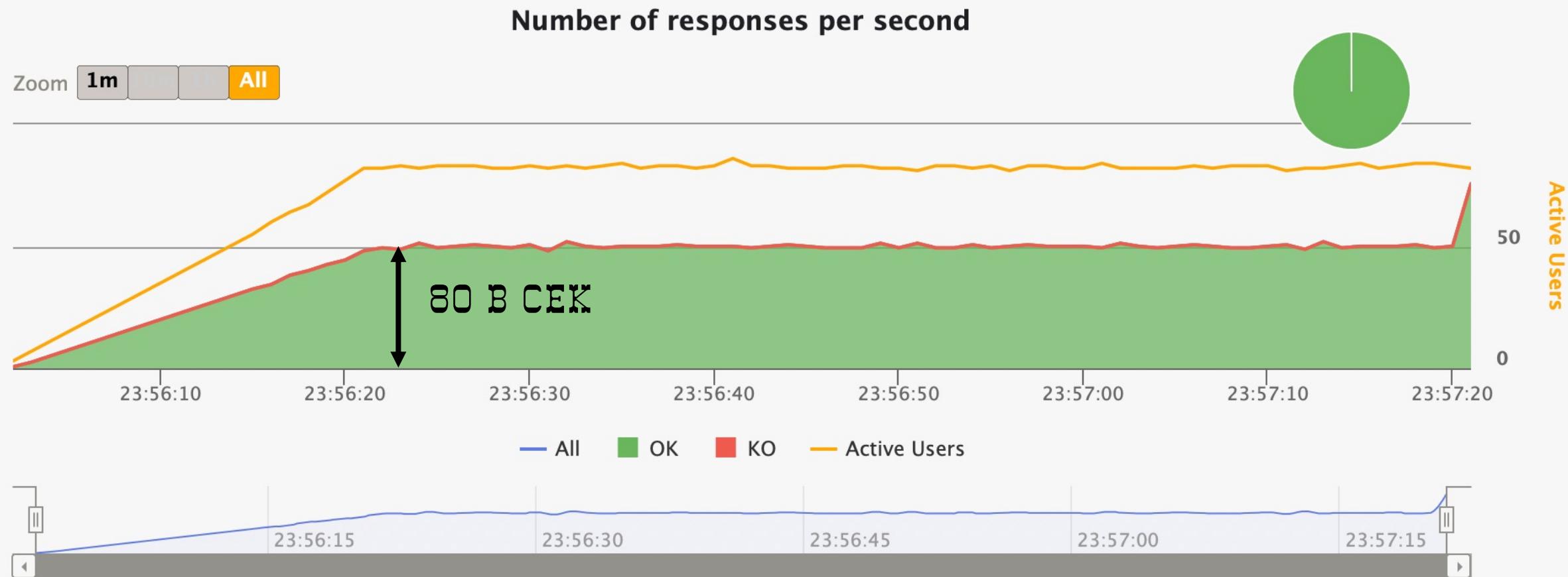
ВЕЛИКА ЛИ ЕМКОСТЬ?



ВЕЛИКА ЛИ ЕМКОСТЬ?

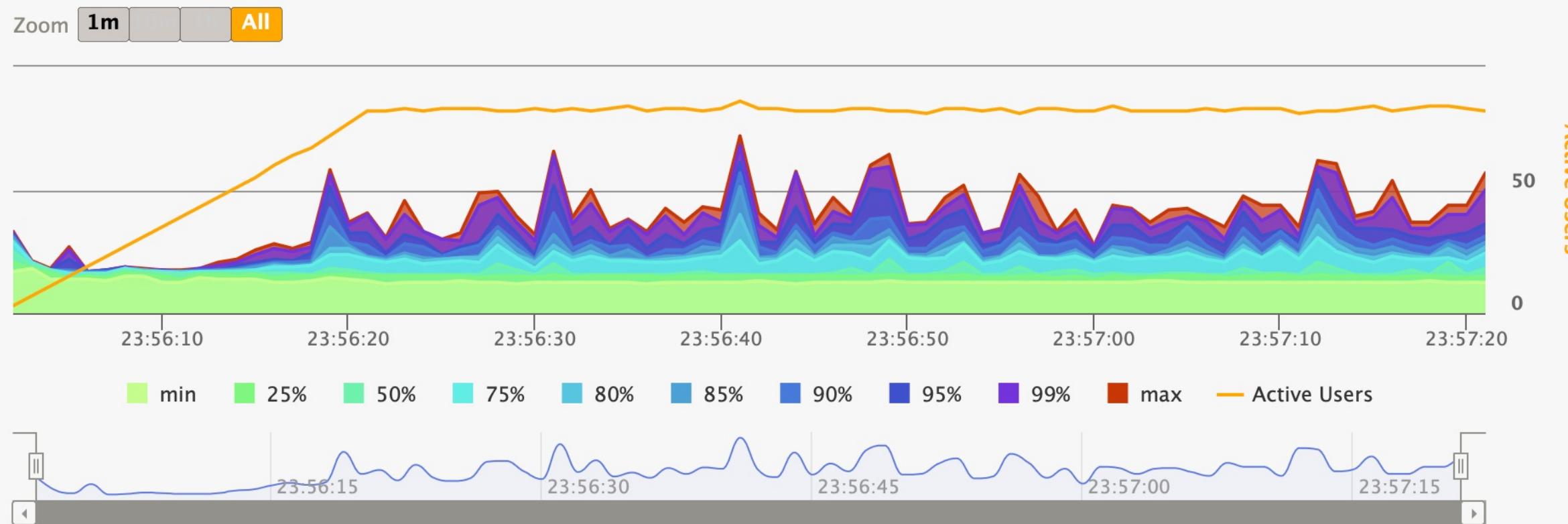


ВЕЛИКА ЛИ ЁМКОСТЬ?



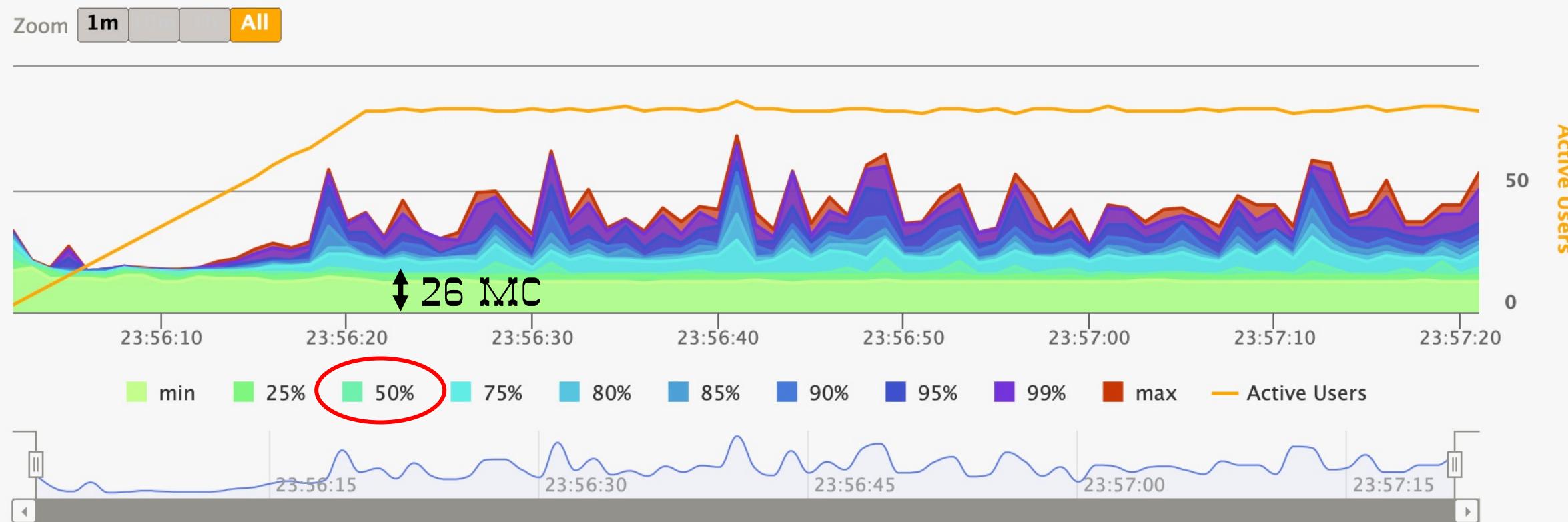
ВЕЛИКА ЛИ ЁМКОСТЬ?

Response Time Percentiles over Time (OK)



ВЕЛИКА ЛИ ЁМКОСТЬ?

Response Time Percentiles over Time (OK)



ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 XPS, НО ЁМКОСТЬ 86-90 XPS?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ X.
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.
3. GO TO 1.

ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ 100 XPS, НО ЁМКОСТЬ 86-90 XPS?

ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ **4х.**
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.
3. GO TO 1.

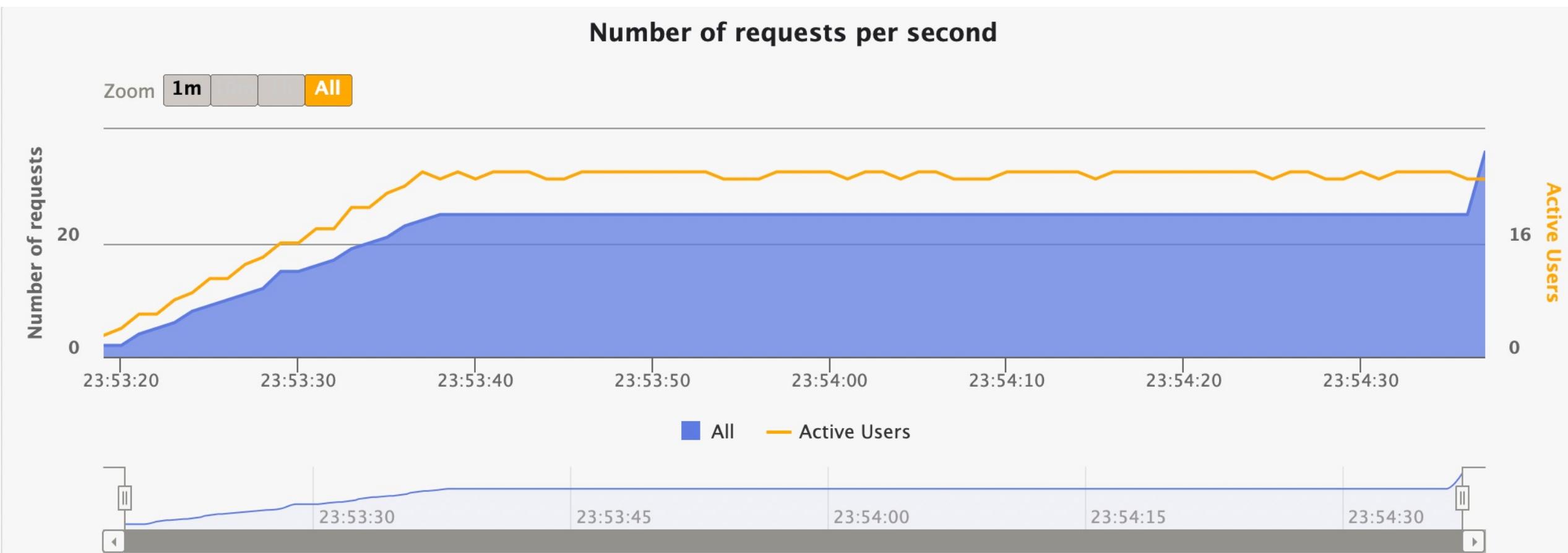
ВЕЛИКА ЛИ ЁМКОСТЬ?

ХОТИМ **100 XPS**, НО ЁМКОСТЬ 86-90 **XPS**?

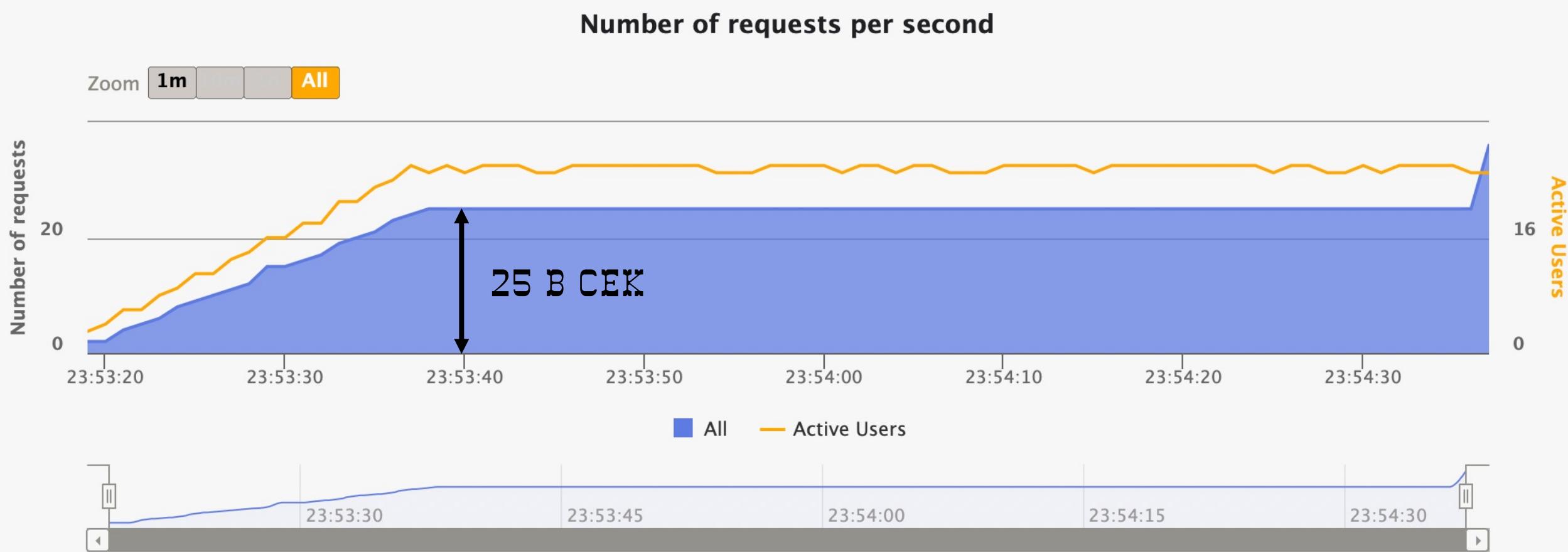
ПОЛЬЗОВАТЕЛЬСКИЙ СЦЕНАРИЙ

1. ПОЛЬЗОВАТЕЛЬ КОПИТ ПАЧКУ РАЗМЕРОМ **4х.**
2. ОТПРАВЛЯЕТ ДАННЫЕ В СЕРВИС.
3. GO TO 1.

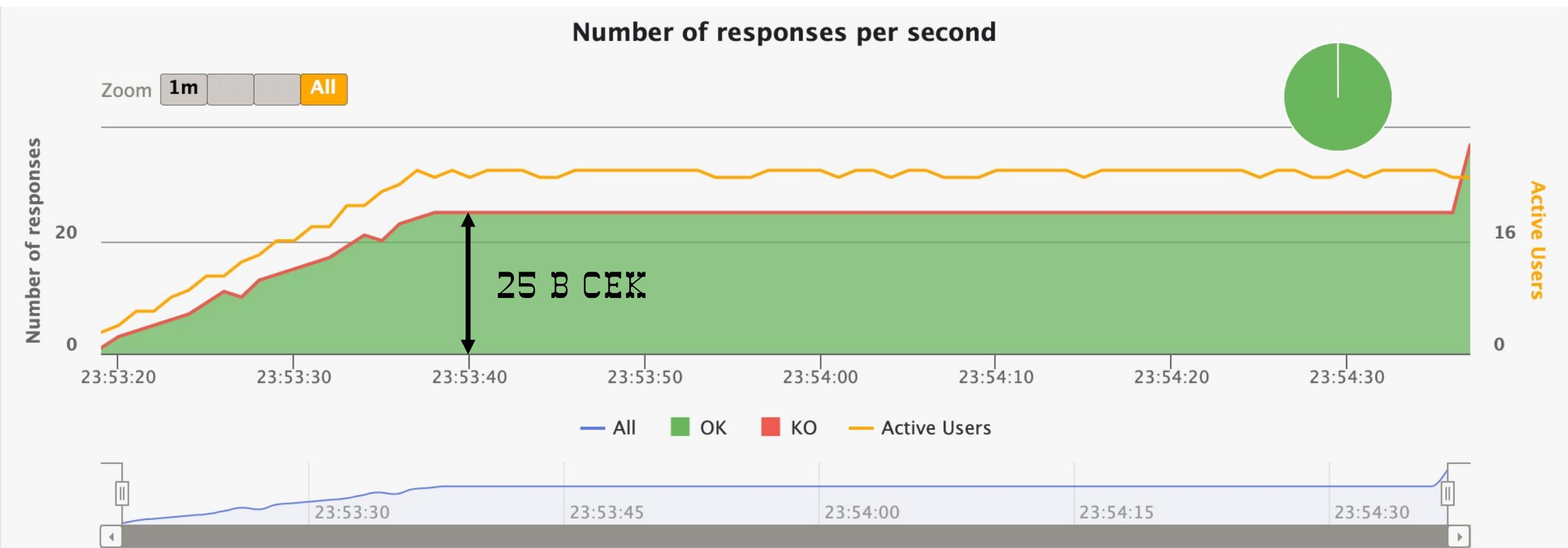
ВЕЛИКА ЛИ ЁМКОСТЬ?



ВЕЛИКА ЛИ ЁМКОСТЬ?

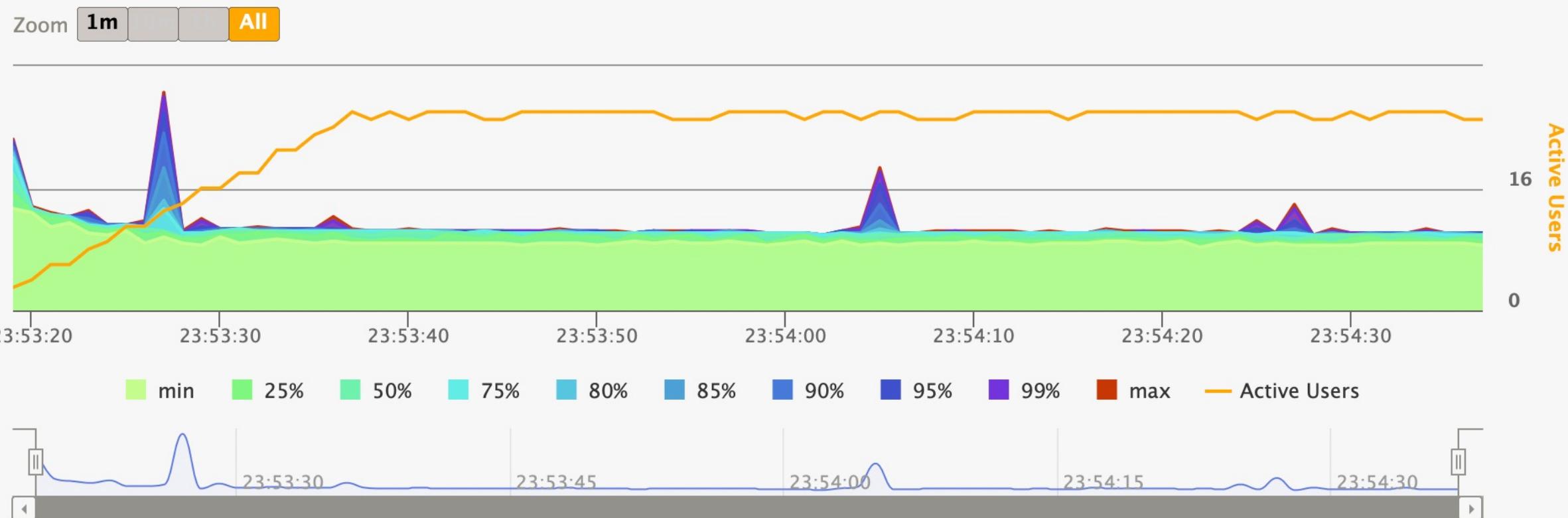


ВЕЛИКА ЛИ ЁМКОСТЬ?



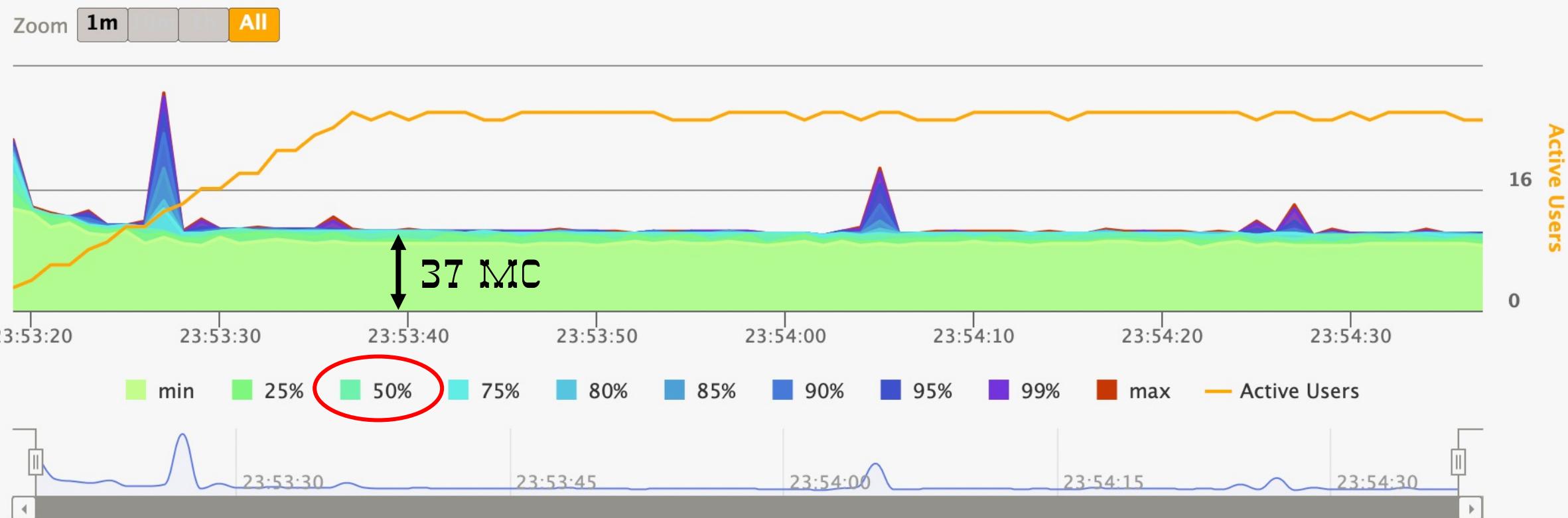
ВЕЛИКА ЛИ ЁМКОСТЬ?

Response Time Percentiles over Time (OK)



ВЕЛИКА ЛИ ЁМКОСТЬ?

Response Time Percentiles over Time (OK)



ВЕЛИКА ЛИ ЁМКОСТЬ?

Размер пачки	RPS	XPS	Перегружен	Min	50%	75%	95%
X	100	100X	Да	20	3 618	6 076	8 545
X	80	80X	Нет	19	26	35	55
4X	25	100X	Нет	31	37	38	39

ВЕЛИКА ЛИ ЁМКОСТЬ?

Размер пачки	RPS	XPS	Перегружен	Min	50%	75%	95%
X	100	100X	Да	20	3 618	6 076	8 545
X	80	80X	Нет	19	26	35	55
4X	25	100X	Нет	31	37	38	39
X	88	88X	Не дышать	19	28	39	70

ВЕЛИКА ЛИ ЁМКОСТЬ?

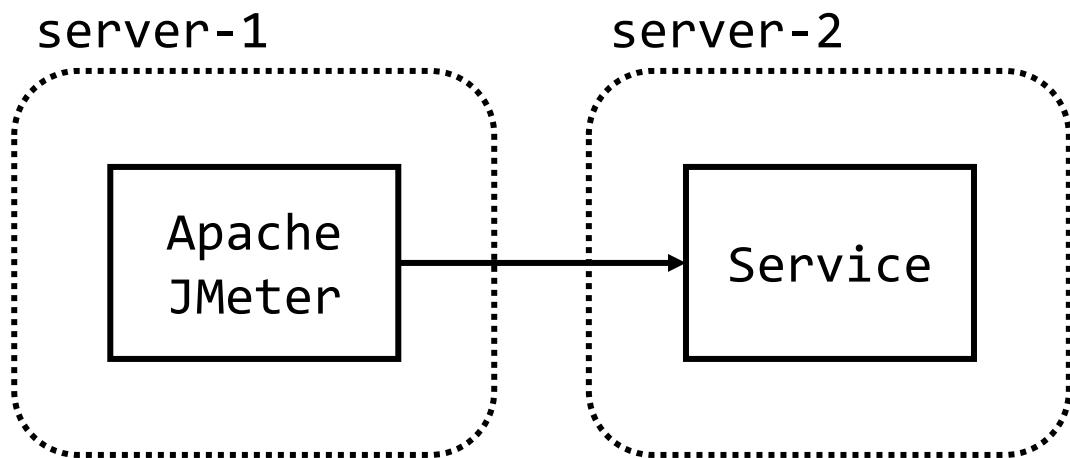
Размер пачки	RPS	XPS	Перегружен	Min	50%	75%	95%
X	100	100X	Да	20	3 618	6 076	8 545
X	80	80X	Нет	19	26	35	55
4X	25	100X	Нет	31	37	38	39
X	88	88X	Не дышать	19	28	39	70

ВЕЛИКА ЛИ ЁМКОСТЬ?

Размер пачки	RPS	XPS	Перегружен	Min	50%	75%	95%
X	100	100X	Да	20	3 618	6 076	8 545
X	80	80X	Нет	19	26	35	55
4X	25	100X	Нет	31	37	38	39
X	88	88X	Не дышать	19	28	39	70

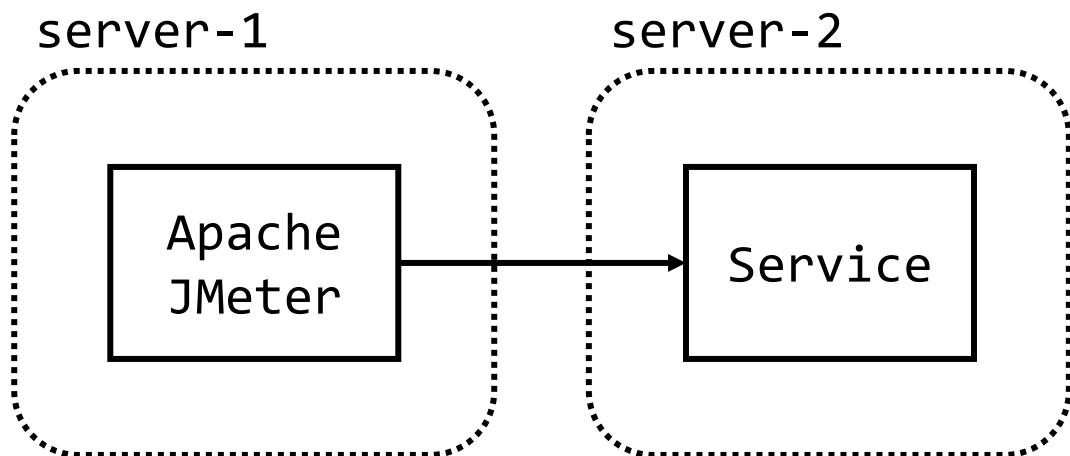
СЕРВИС НЕПОБЕЖДАЕМНЫЙ

СЕРВИС НЕПОБЕЖДАЕМЫЙ



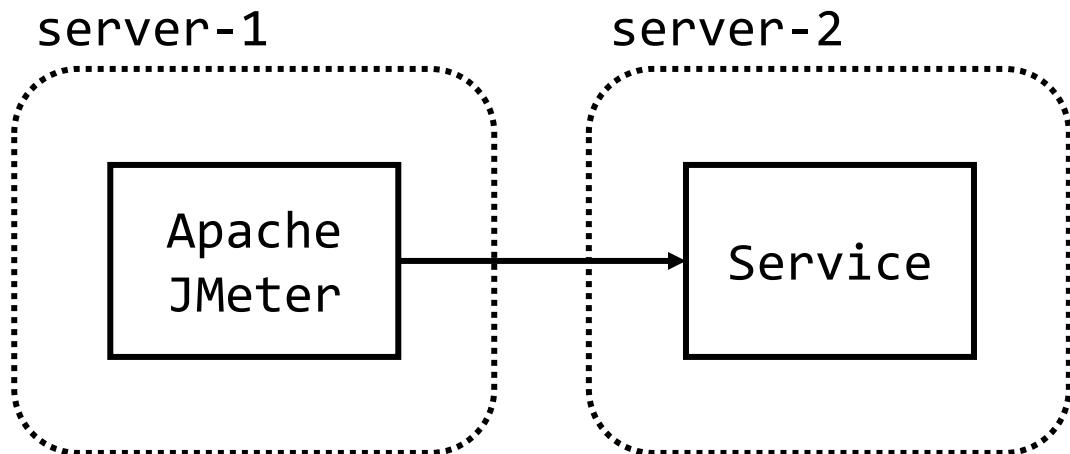
СЕРВИС НЕПОБЕЖДАЕМЫЙ

VU = 1024, кол-во запросов = 100



СЕРВИС НЕПОБЕЖДАЕМЫЙ

VU = 1024, кол-во запросов = 100

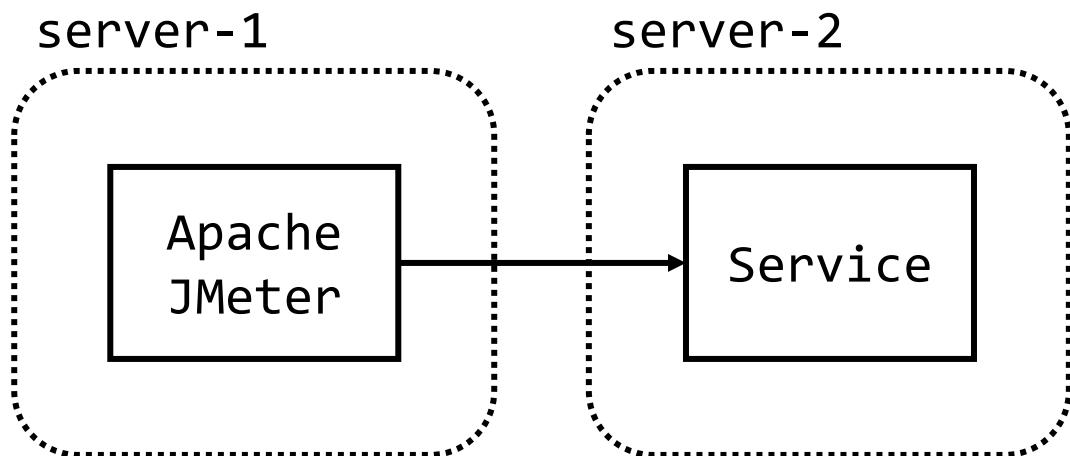


AVERAGE = 200 MS, THROUGHTPUT = 2000 RPS



СЕРВИС НЕПОБЕЖДАЕМЫЙ

VU = 1024, кол-во запросов = 100

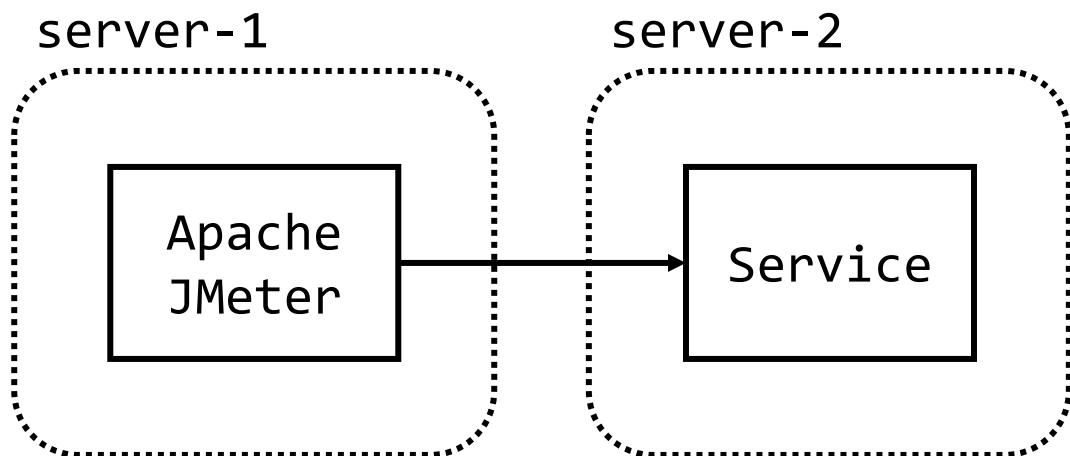


МЕТРИКИ:
«СЕРВИС ДАЖЕ НЕ ВСПОТЕЛ»

AVERAGE = 200 MS, THROUGHTPUT = 2000 RPS

СЕРВИС НЕПОБЕЖДАЕМЫЙ

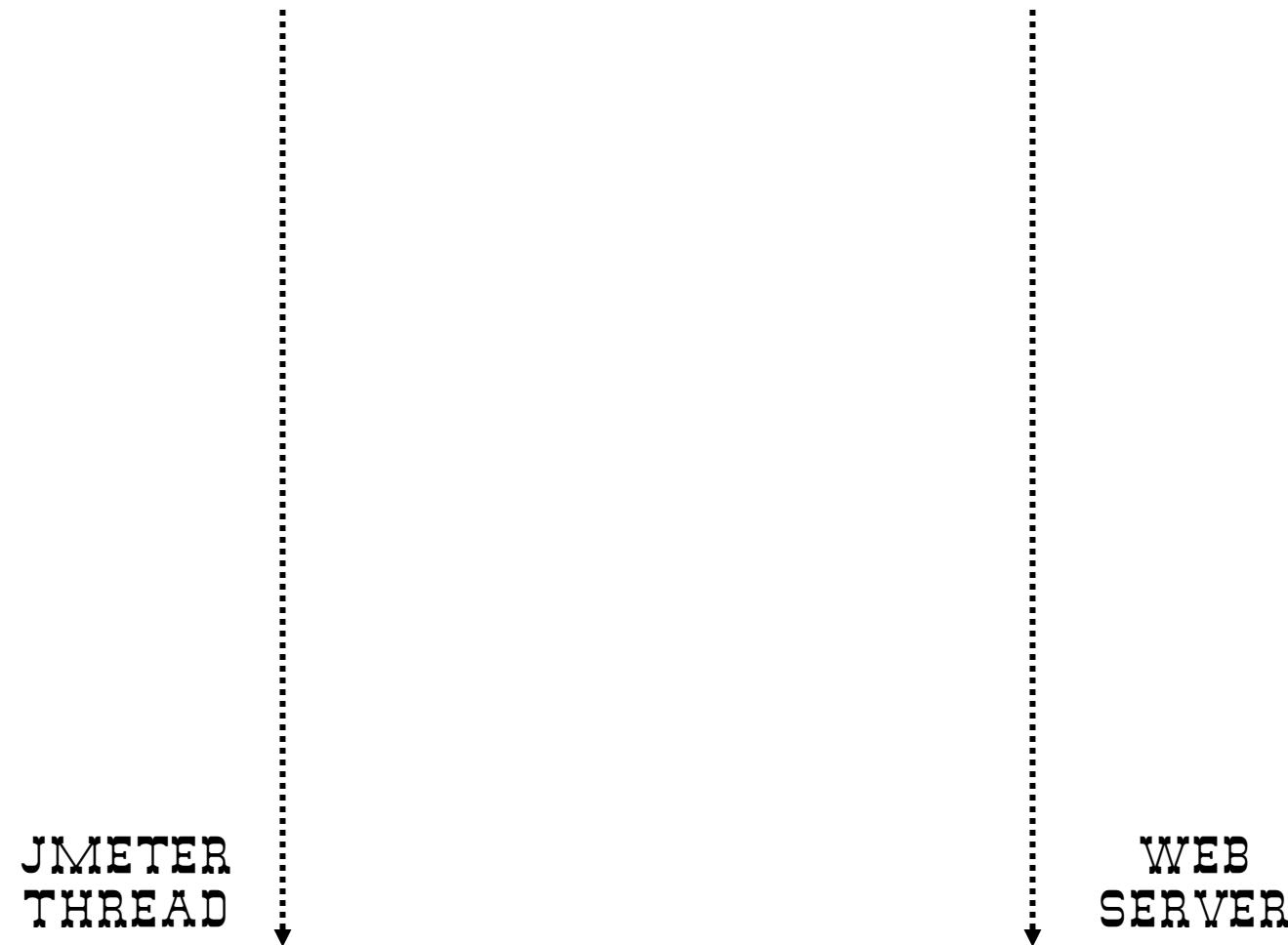
VU = 1024, кол-во запросов = 100



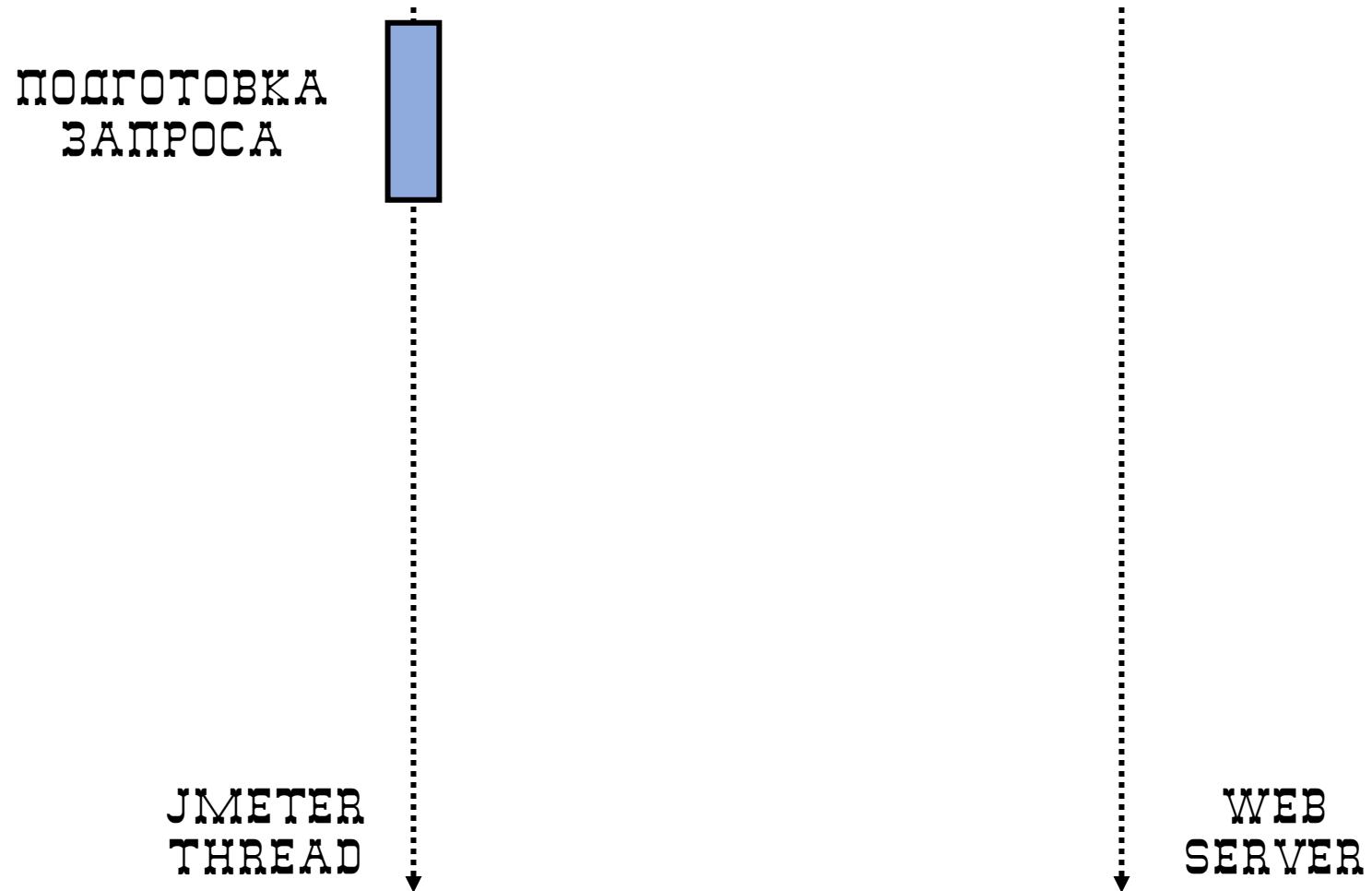
МЕТРИКИ:
«СЕРВИС ДАЖЕ НЕ ВСПОТЕЛ»

AVERAGE = 200 MS, THROUGHTPUT = 2000 RPS

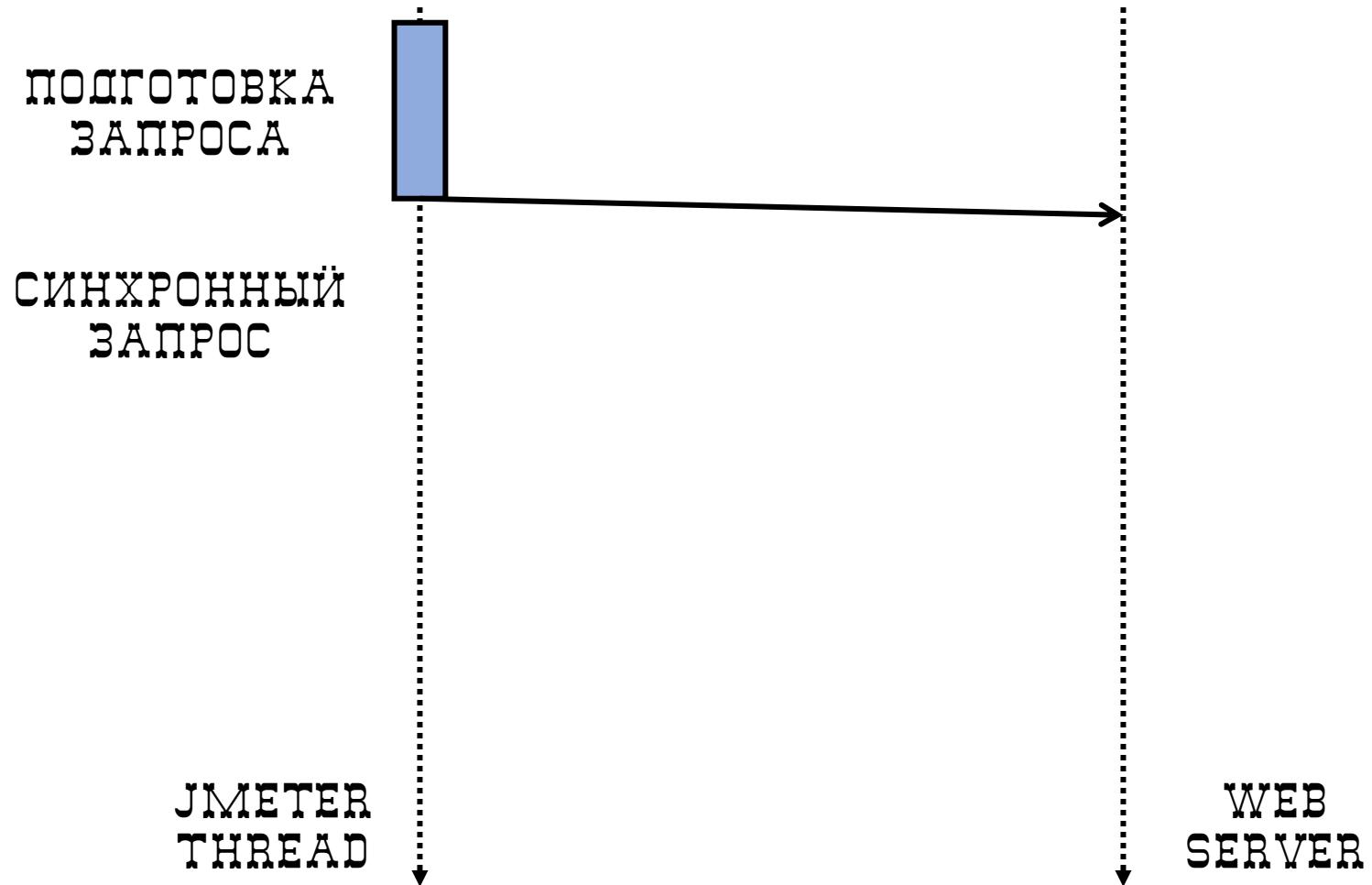
СЕРВИС НЕПОБЕЖДАЕМЫЙ



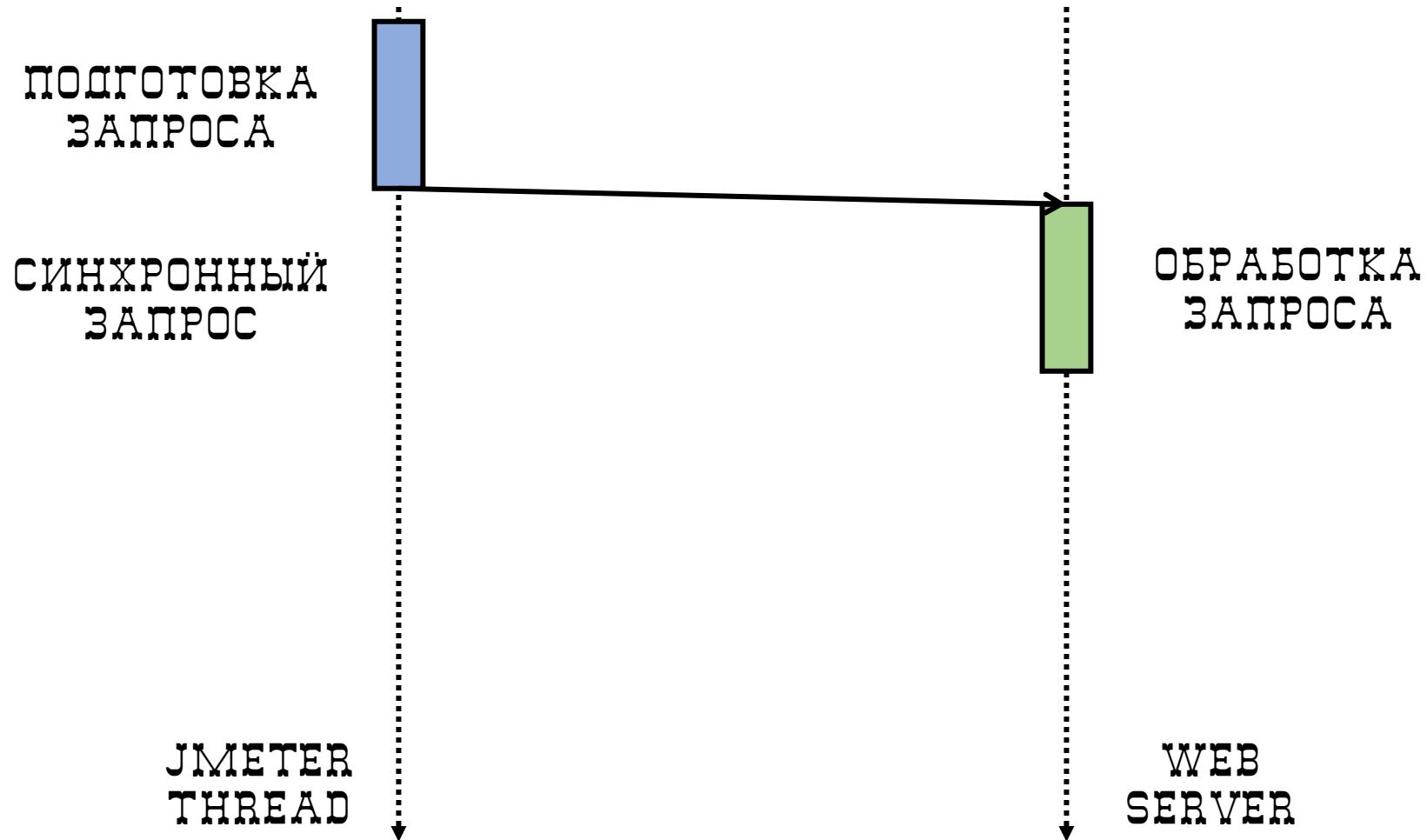
СЕРВИС НЕПОБЕЖДЕННЫЙ



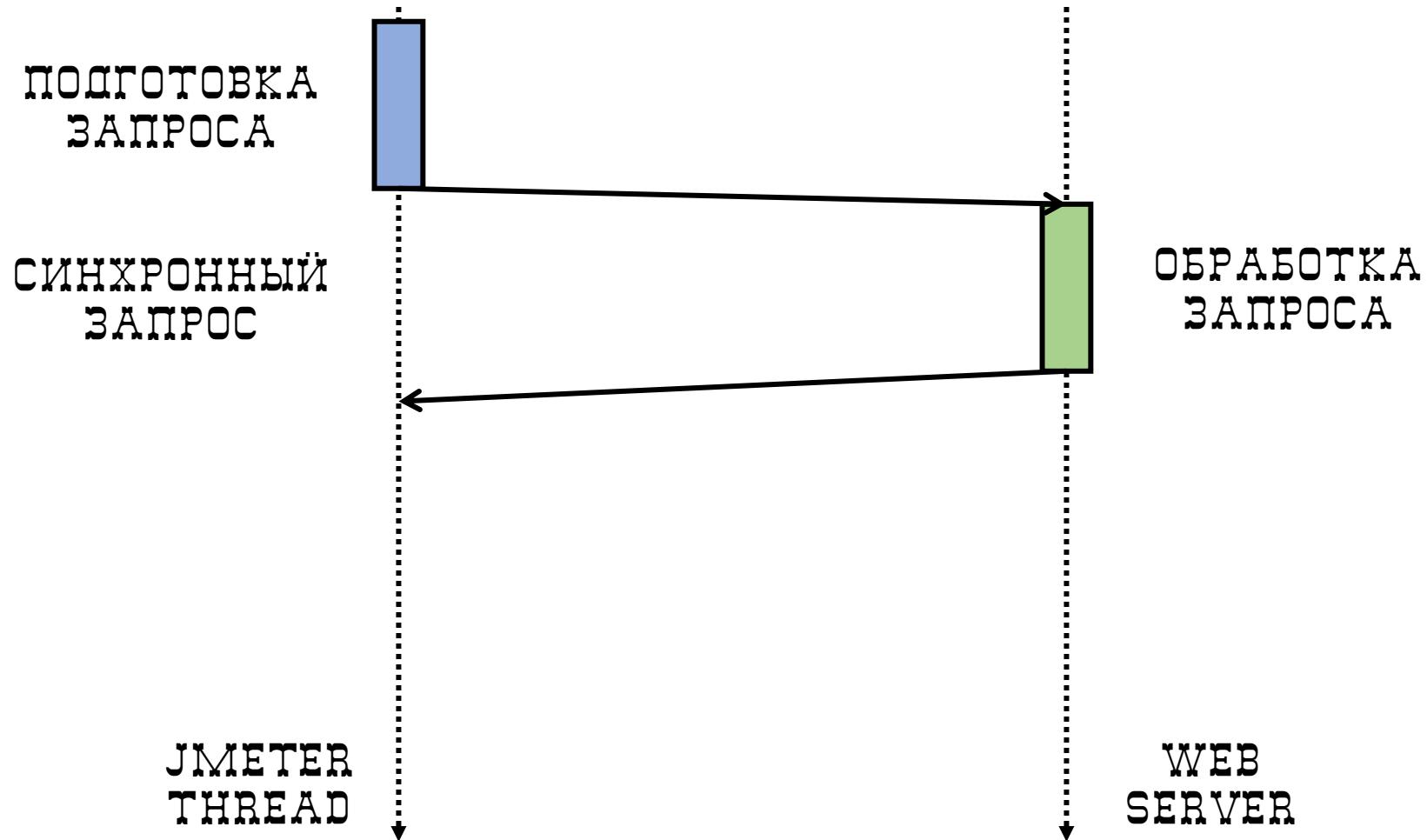
СЕРВИС НЕПОБЕЖДЕННЫЙ



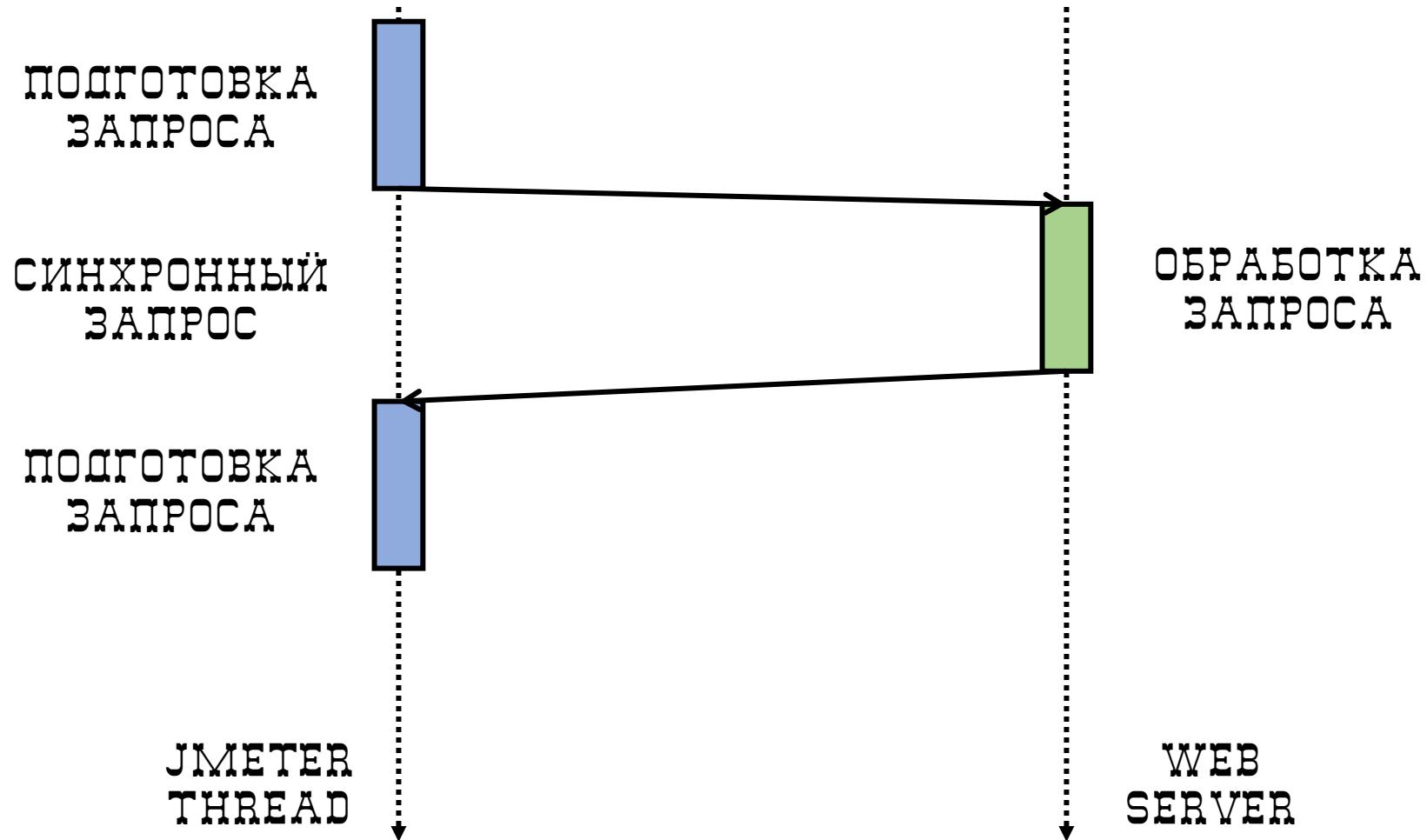
СЕРВИС НЕПОБЕЖДЕННЫЙ



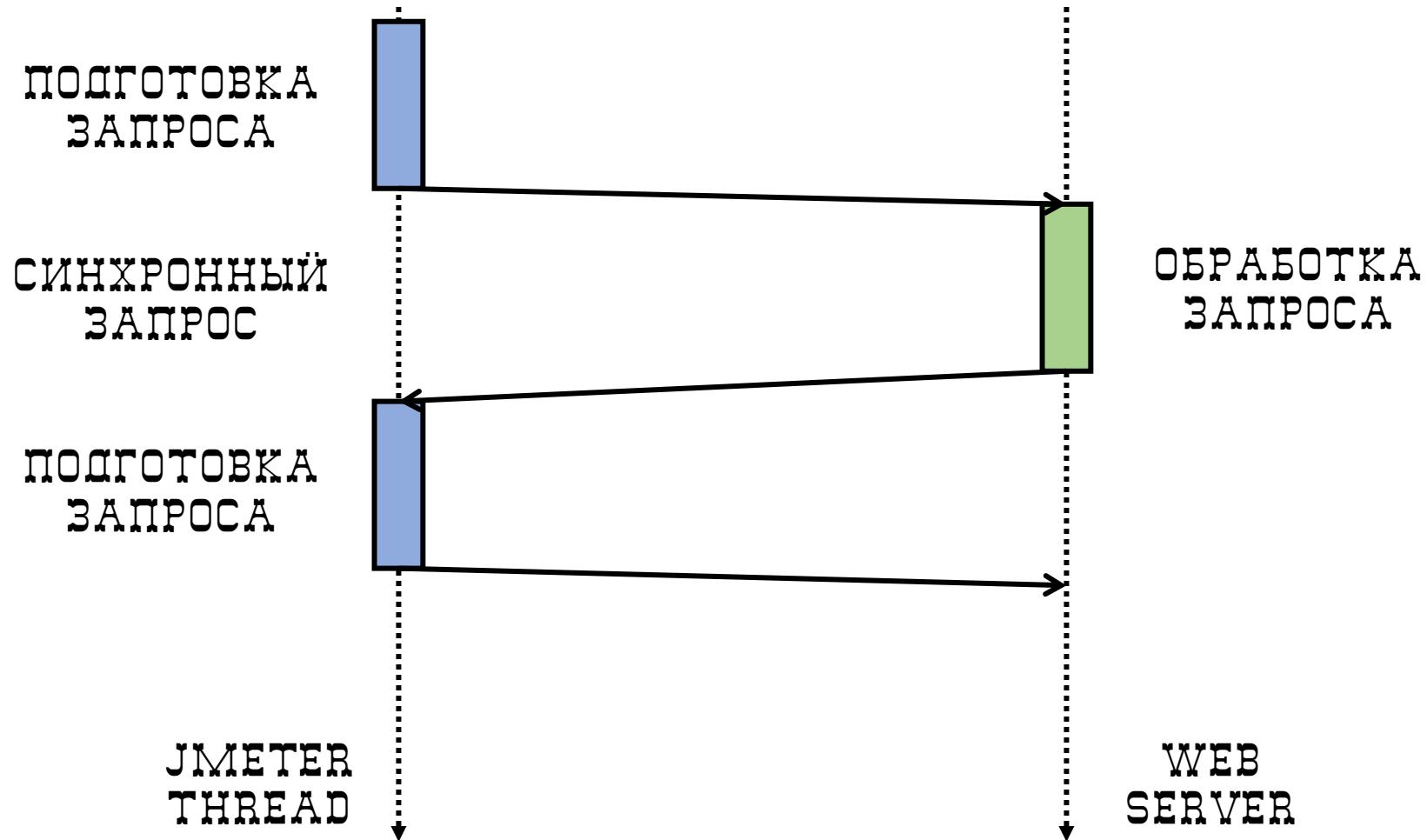
СЕРВИС НЕПОБЕЖДЕННЫЙ



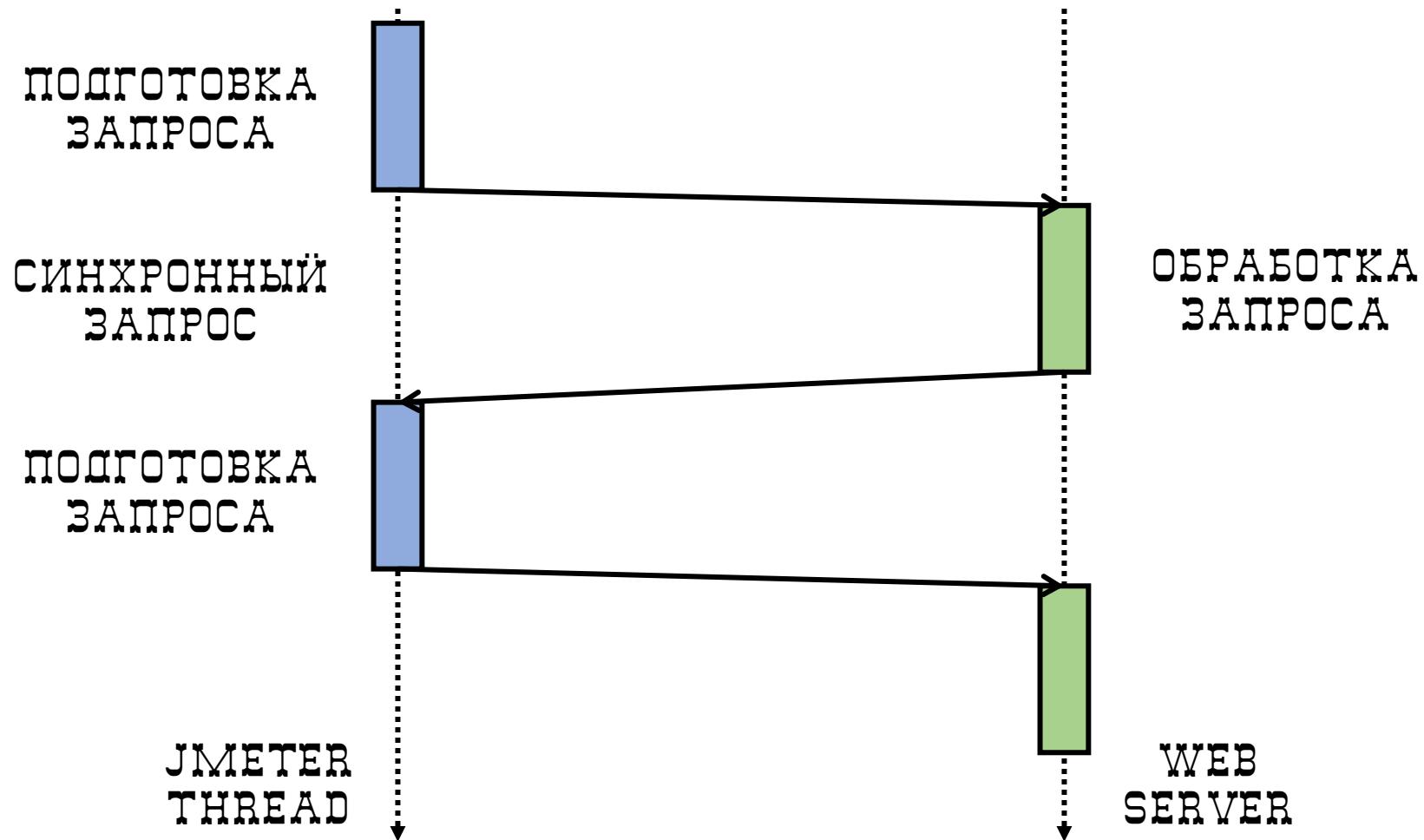
СЕРВИС НЕПОБЕЖДЕННЫЙ



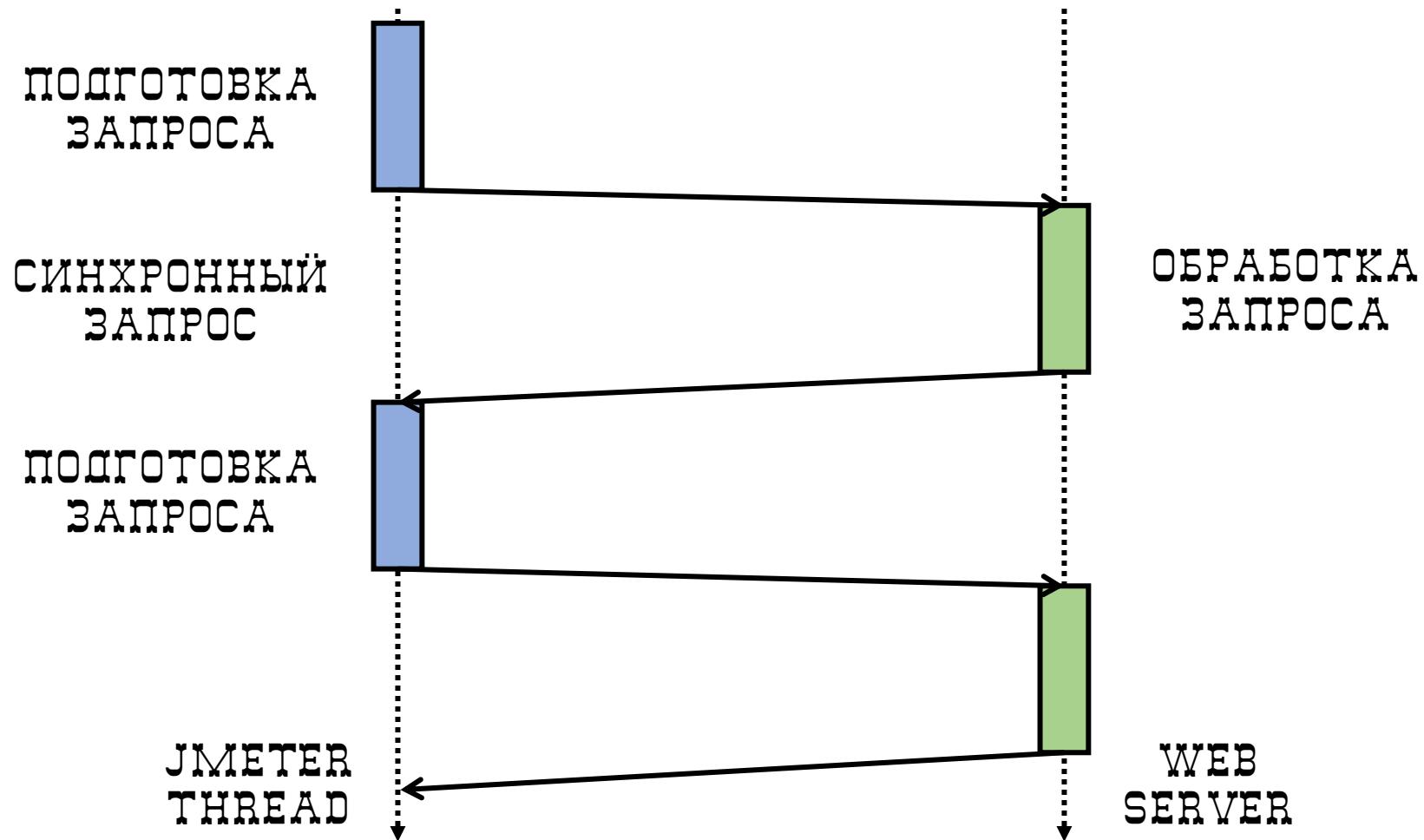
СЕРВИС НЕПОБЕЖДЕННЫЙ



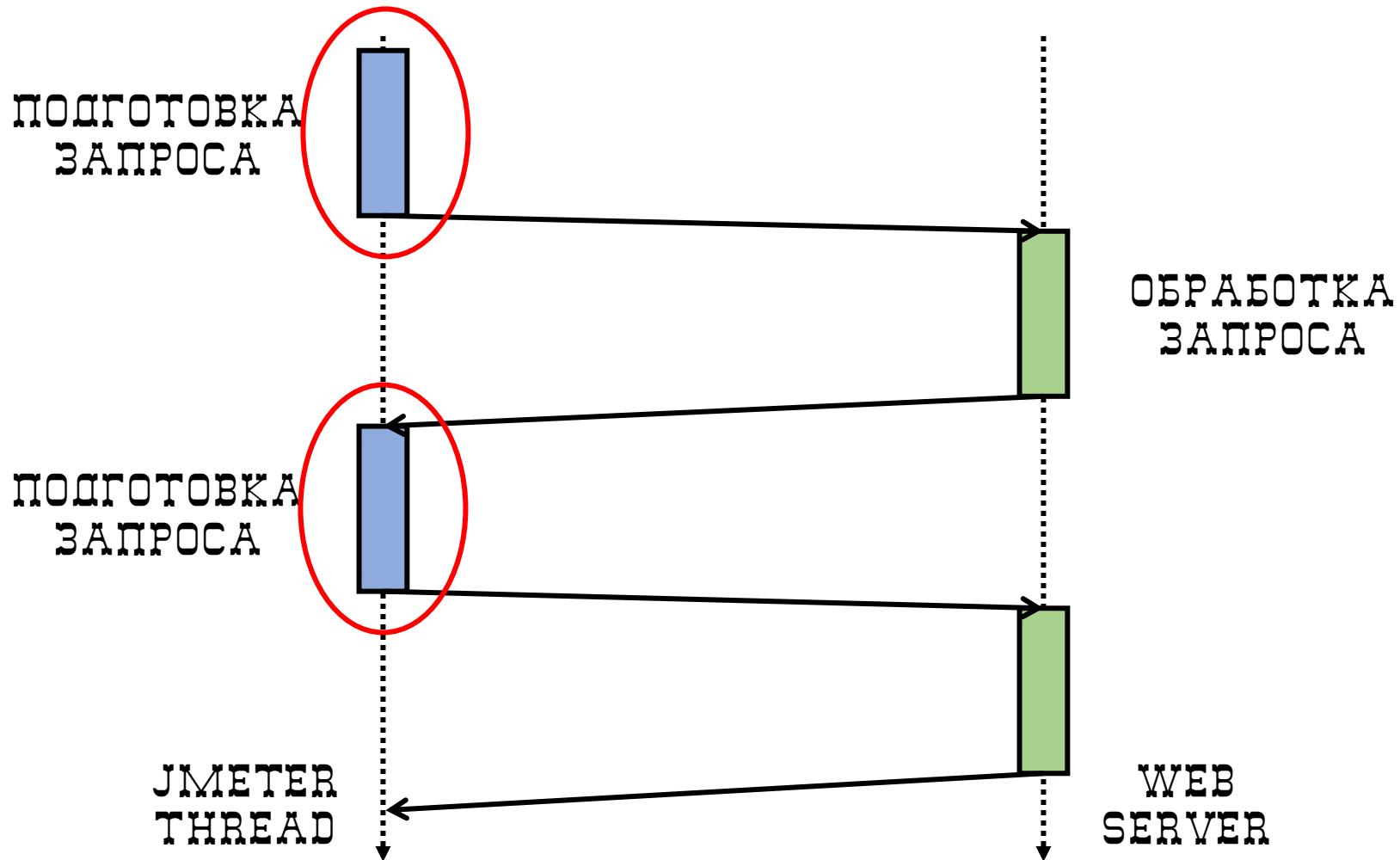
СЕРВИС НЕПОБЕЖДЕННЫЙ



СЕРВИС НЕПОБЕЖДЕННЫЙ



СЕРВИС НЕПОБЕЖДЕННЫЙ



СЕРВИС НЕПОБЕЖДАЕМЫЙ

ВРЕМЯ ПОДГОТОВКИ БОЛЬШЕ

ВРЕМЕНИ ОБРАБОТКИ

СЕРВИС НЕПОБЕЖДАЕМЫЙ

ВРЕМЯ ПОДГОТОВКИ БОЛЬШЕ

ВРЕМЕНИ ОБРАБОТКИ

ПОЧЕМУ ТАК?

СЕРВИС НЕПОБЕЖДАЕМЫЙ

ВРЕМЯ ПОДГОТОВКИ БОЛЬШЕ

ВРЕМЕНИ ОБРАБОТКИ

ПОЧЕМУ ТАК?

— COMPRESSION VS DECOMPRESSION: X3...6

СЕРВИС НЕПОБЕЖДАЕМЫЙ

ВРЕМЯ ПОДГОТОВКИ БОЛЬШЕ

ВРЕМЕНИ ОБРАБОТКИ

ПОЧЕМУ ТАК?

- COMPRESSION VS DECOMPRESSION: Х3...6
- ГЕНЕРАЦИЯ ДАННЫХ

СЕРВИС НЕПОБЕЖДАЕМЫЙ

ВРЕМЯ ПОДГОТОВКИ БОЛЬШЕ

ВРЕМЕНИ ОБРАБОТКИ

ПОЧЕМУ ТАК?

- COMPRESSION VS DECOMPRESSION: Х3...6
- ГЕНЕРАЦИЯ ДАННЫХ
 - * ВЫЧИСЛЕНИЕ ХЭШЕЙ

СЕРВИС НЕПОБЕЖДЕННЫЙ

ВРЕМЯ ПОДГОТОВКИ БОЛЬШЕ

ВРЕМЕНИ ОБРАБОТКИ

ПОЧЕМУ ТАК?

- COMPRESSION VS DECOMPRESSION: Х3...6
- ГЕНЕРАЦИЯ ДАННЫХ
 - * ВЫЧИСЛЕНИЕ ХЭШЕЙ
 - * UUID (ENTROPY)

СЕРВИС НЕПОБЕЖДАЕМЫЙ

ВРЕМЯ ПОДГОТОВКИ БОЛЬШЕ

ВРЕМЕНИ ОБРАБОТКИ

ПОЧЕМУ ТАК?

- COMPRESSION VS DECOMPRESSION: Х3...6
- ГЕНЕРАЦИЯ ДАННЫХ
 - * ВЫЧИСЛЕНИЕ ХЭШЕЙ
 - * UUID (ЕНТРОПЫ)
 - * БЛОКИРОВКИ

выводы

НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ



И ВОРВАТЬСЯ НА СЕРВЕР С ГАТЛИНГОМ

ЦЕЛИ

- SCALABILITY TESTING
- LOAD TESTING
- STRESS TESTING
- SPIKE TESTING
- ENDURANCE TESTING
- VOLUME TESTING

ОГРАНИЧЕННЫЕ РЕСУРСЫ

5 БАЗОВЫХ РЕСУРСА

ОГРАНИЧЕННЫЕ РЕСУРСЫ

5 БАЗОВЫХ РЕСУРСА
— СРУ

- * ЧАСТОТА
- * КОЛИЧЕСТВО ЯДЕР

ОГРАНИЧЕННЫЕ РЕСУРСЫ

5 БАЗОВЫХ РЕСУРСА

- СРУ
- RAM

* ОБЪЁМ ПАМЯТИ

ОГРАНИЧЕННЫЕ РЕСУРСЫ

5 БАЗОВЫХ РЕСУРСА

- CPU
- RAM
- DISK IO

- * СКОРОСТЬ ЗАПИСИ И ЧТЕНИЯ
- * IOPS
- * ОБЪЁМ ДИСКА

ОГРАНИЧЕННЫЕ РЕСУРСЫ

5 БАЗОВЫХ РЕСУРСА

- CPU
- RAM
- DISK IO
- NETWORK IO

- * ПРОПУСКНАЯ СПОСОБНОСТЬ
- * ЗАДЕРЖКА СЕТИ

ОГРАНИЧЕННЫЕ РЕСУРСЫ

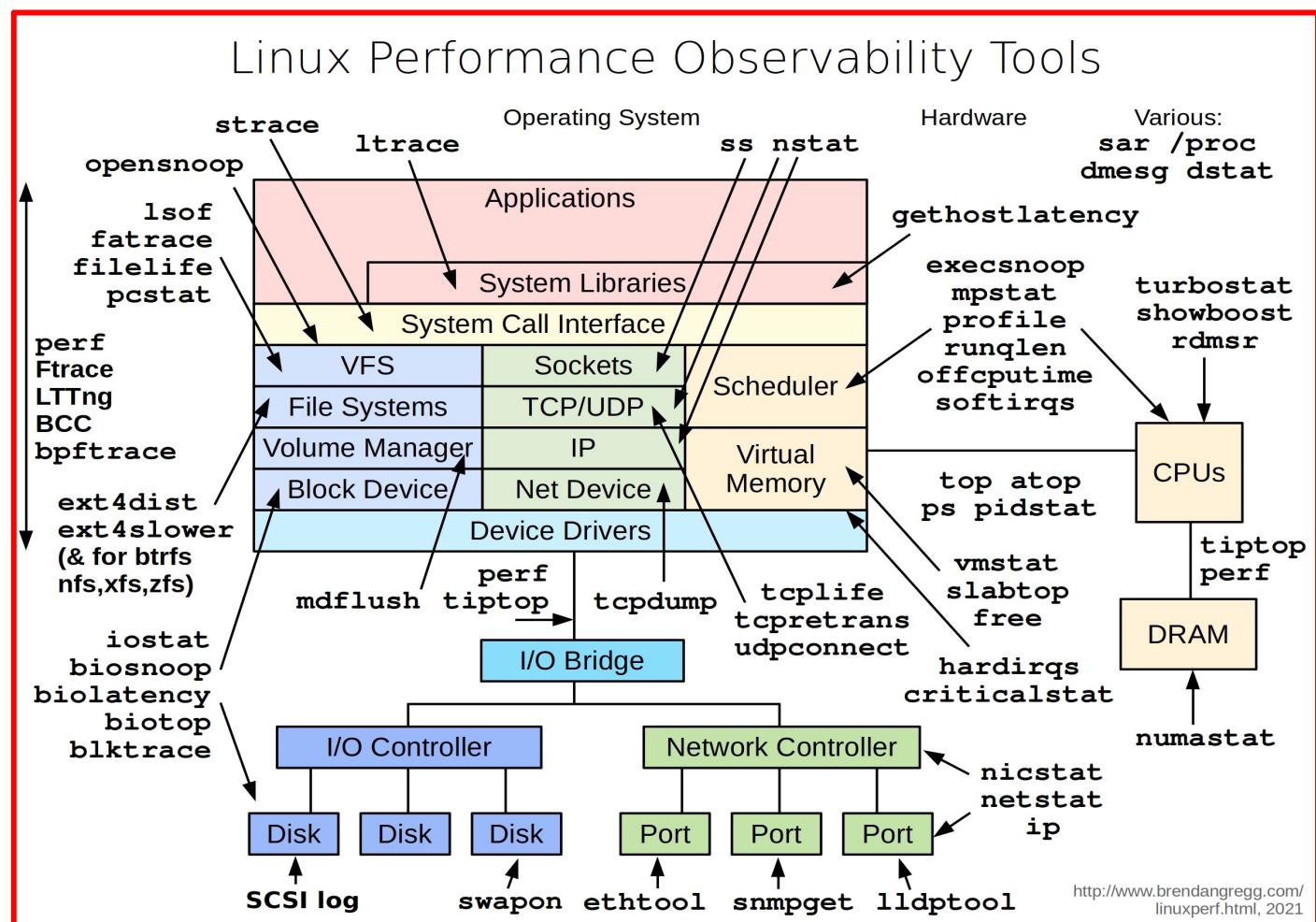
5 БАЗОВЫХ РЕСУРСА

- CPU
- RAM
- DISK IO
- NETWORK IO
- RUNTIME

- * БЛОКИРОВКИ
(GIL В PYTHON)
- * THREAD POOL
- * GARBAGE COLLECTION
- * CONNECTION POOL
- * УТЕЧКИ ПАМЯТИ
- ...

ОГРАНИЧЕННЫЕ РЕСУРСЫ

- CPU
- RAM
- DISK IO
- NETWORK IO
- RUNTIME



ОГРАНИЧЕННЫЕ РЕСУРСЫ

МЕТРИКИ

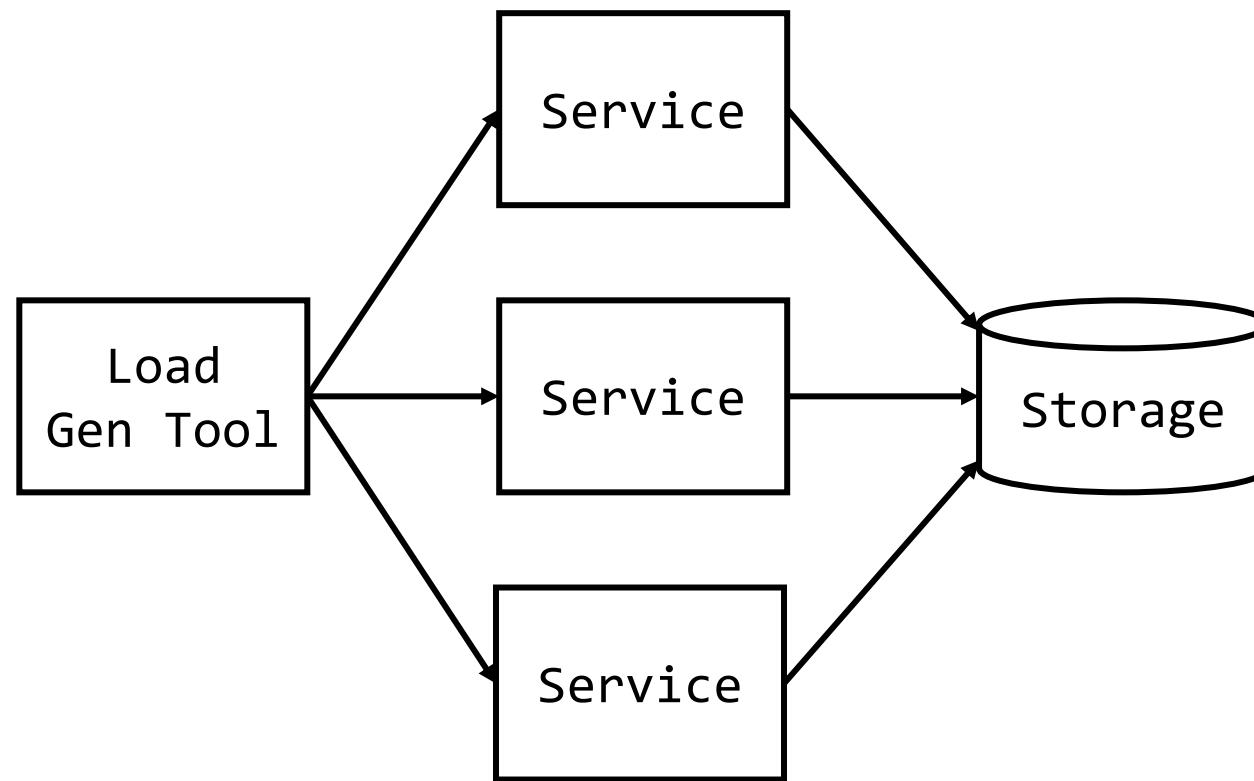
- СИСТЕМНЫЕ
- ПРИЛОЖЕНИЯ
- ГЕНЕРАТОРА НАГРУЗКИ

ЭКСПЕРИМЕНТЫ

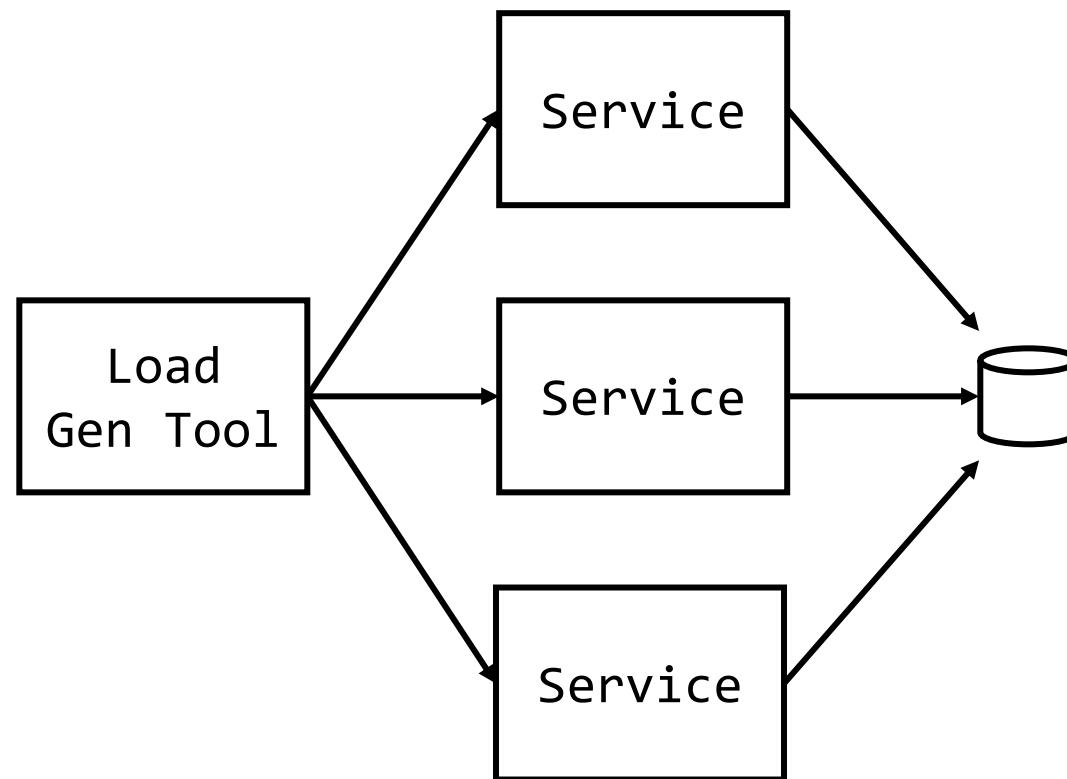
ЭКСПЕРИМЕНТЫ



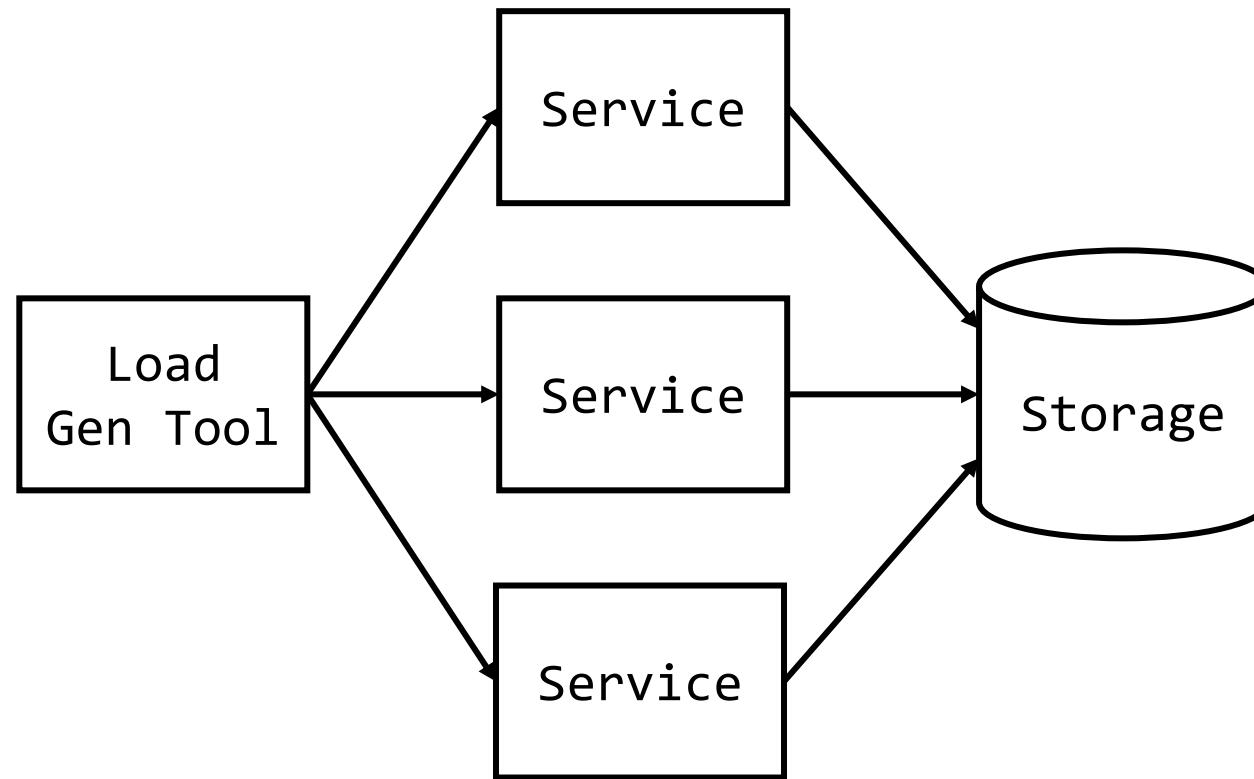
ЭКСПЕРИМЕНТЫ



ЭКСПЕРИМЕНТЫ



ЭКСПЕРИМЕНТЫ



**РАЗРАБОТЧИКИ ДЕЛАЮТ СЕРВИСЫ
БЫСТРЫМИ И МЕДЛЕННЫМИ**

**А GATLING ПОЗВОЛЯЕТ
УВИДЕТЬ ЭТУ РАЗНИЦУ**

https://tg.me/chnl_GregoryKoshelev