

Kafka: Тестирование и эксплуатация в условиях ограниченных ресурсов

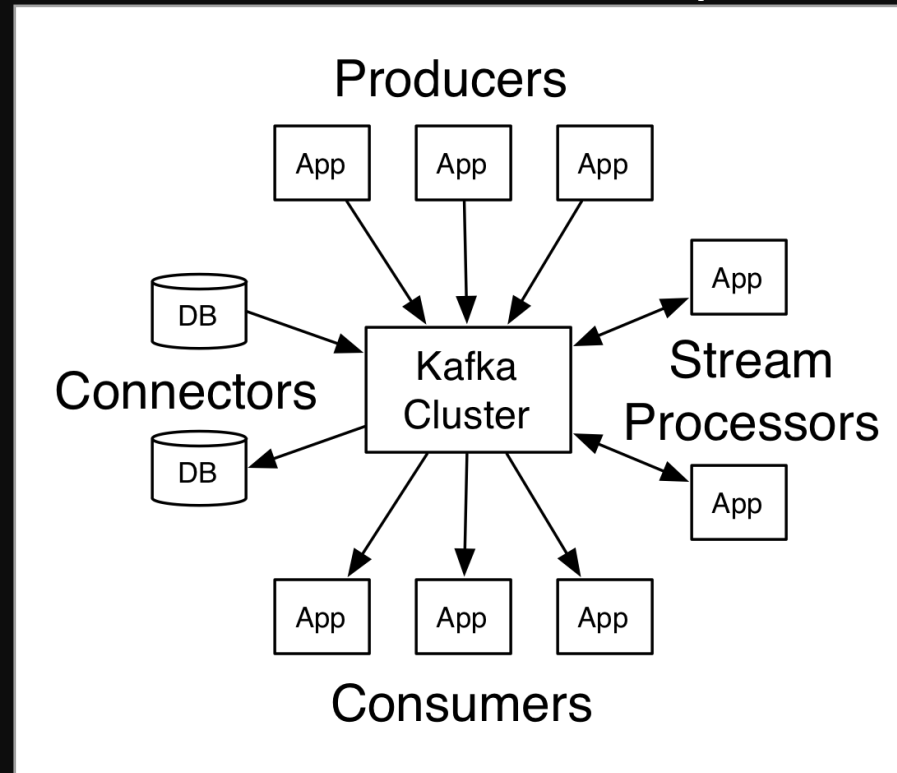
О себе

- В области разработки ПО > 10 лет, тестирование > 5 лет.
- Основное направление тестирования - инфраструктурные решения (интеграционные шины и системы хранения данных)
- В команде роль - TestOps
- Основной стек - Java/Linux, Scala, JMeter, Gatling



○ Καφκε

- Apache Kafka[®] is a **distributed streaming platform**
- Kafka[®] is used for building real-time data pipelines and streaming apps. It is **horizontally scalable, fault-tolerant, wicked fast**, and runs in production in thousands of companies.



О сборе телеметрии для сервисов Контура

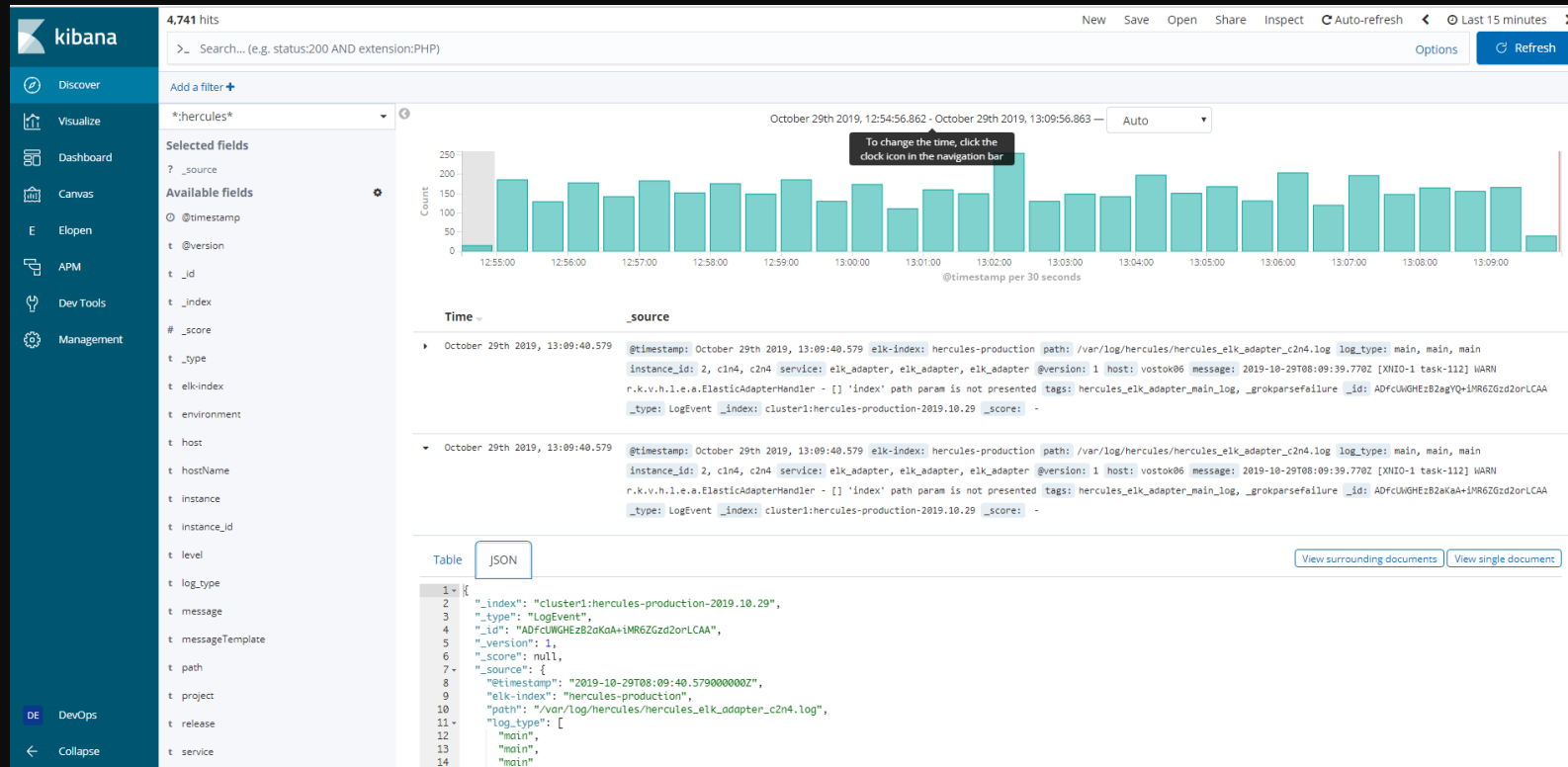
- Логи 100 000 EPS
- Метрики 2 000 000 EPS
- Распределенные трассировки
850 000 EPS
- Другие события



Пример для логов



Пример для логов. ELK



Пример для логов. Sentry

The image displays two dashboards side-by-side. The left dashboard is Kibana, showing a search for '*hercules*' with 4,741 hits. It features a bar chart of log counts over time and a table of log entries with a JSON view. The right dashboard is Sentry, showing a list of unresolved issues for the 'xcom' project. The issues include various exceptions like BadHttpRequestException, XcomApiClientException, ClusterClientException, InvalidOperationException, and XcomApiClientException, each with a description and a small graph.

Kibana Dashboard:

- Search: `*hercules*`
- 4,741 hits
- Bar chart: Count vs. Time (October 29th 2019, 12:54:56.862 - October 29th 2019, 13:09:56.863)
- Table: Log entries with fields like @timestamp, @version, _id, _index, _score, _type, _source.
- JSON view:

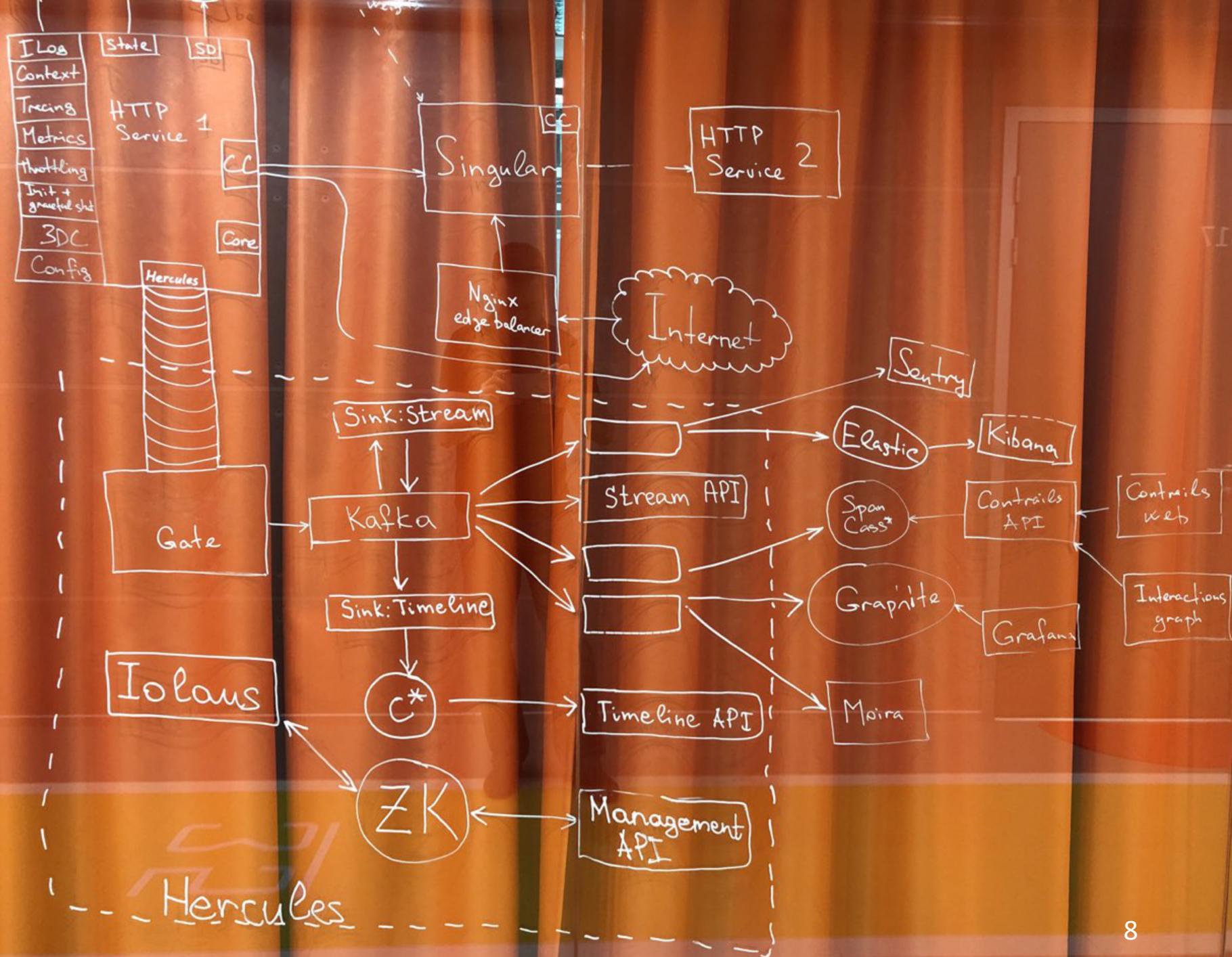
```
{ 1: { 2: "_index": "cluster1:hercules-production-2019.10.29", 3: "_type": "LogEvent", 4: "_id": "ADFcUWHEzB2oKaA+iMR6ZGzd2orLCAA", 5: "_version": 1, 6: "_score": null, 7: "_source": { 8: "@timestamp": "2019-10-29T08:09:40.579000000Z", 9: "elk-index": "hercules-production", 10: "path": "/var/log/hercules/hercules_elk_adapt", 11: "log_type": [ 12: "main", 13: "main", 14: "main" ] } }
```

Sentry Dashboard:

- Project: xcom
- Unresolved Issues (54)
- Sort by: Last Seen
- Issues list:

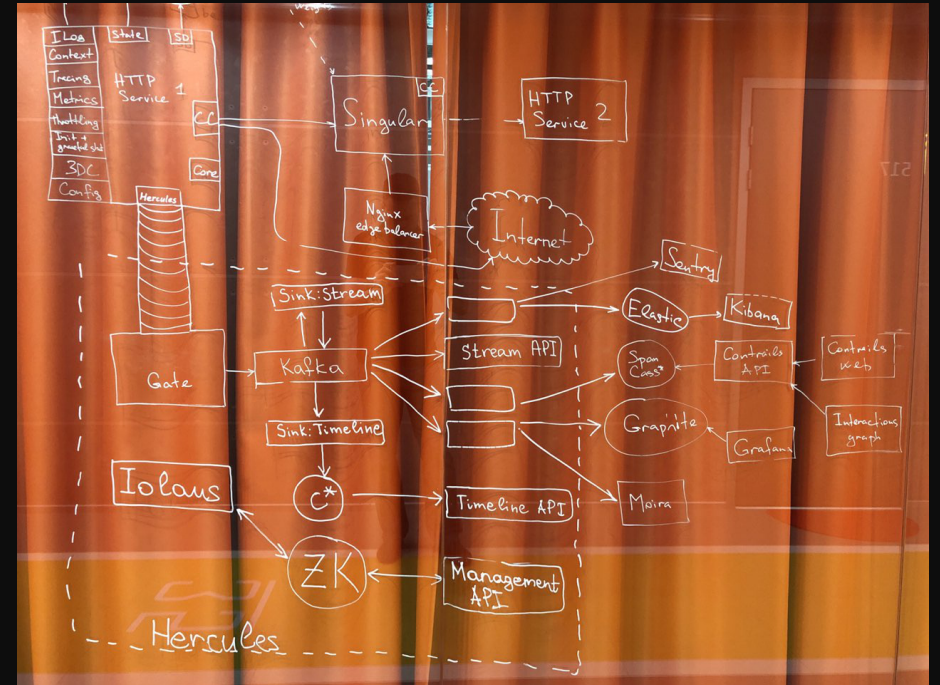
 - BadHttpRequestException** 6154dd78-c95f-459f-97d7-5d289460f85d
in C:\buildAgent\work\8b2d86cfd9265dc8\Front\FrontStartup.cs
Unexpected end of request content.
XCOM-T9 5 hours ago - 22 days old
 - XcomApiClientException** 4f315ed0-579d-4854-af96-7480fed56b77
in C:\buildAgent\work\8b2d86cfd9265dc8\Api.Client\OperationResult.cs
Api call failed with code: Unauthorized
XCOM-11Z 6 hours ago - 12 days old
 - ClusterClientException** 0ebaa670-4227-49f9-99d5-ae1556db82a3
in C:\buildAgent\work\8b2d86cfd9265dc8\Api.Internal.Client\InternalApiClient.cs
Response status code 'BadGateway' indicates unsuccessful outcome.
XCOM-SC 6 hours ago - a month old
 - InvalidOperationException** b5e25285-021b-4aed-89a4-862ecb4eb1d7
in C:\buildAgent\work\8b2d86cfd9265dc8[Integrations.Telphin\Client\TelphinResponse.cs
Request failed with code RequestTimeout
XCOM-121 21 hours ago - 9 days old
 - InvalidOperationException** c2fa97c5-b048-4116-9807-b6a0d9a71703
in C:\buildAgent\work\8b2d86cfd9265dc8[Integrations.Kontur\Analytics\AnalyticsRawDataProvider.cs
Nullable object must have a value.
XCOM-12C a day ago - 4 days old
 - XcomApiClientException** 06748e71-f995-4d5a-ab6e-f7f33c87c1b3
in C:\buildAgent\work\8b2d86cfd9265dc8\Api.Client\OperationResult.cs
Api call failed with code: RequestTimeout

Наша система транспорта телеметрии (Hercules)



Наша система транспортировки телеметрии (Hercules)

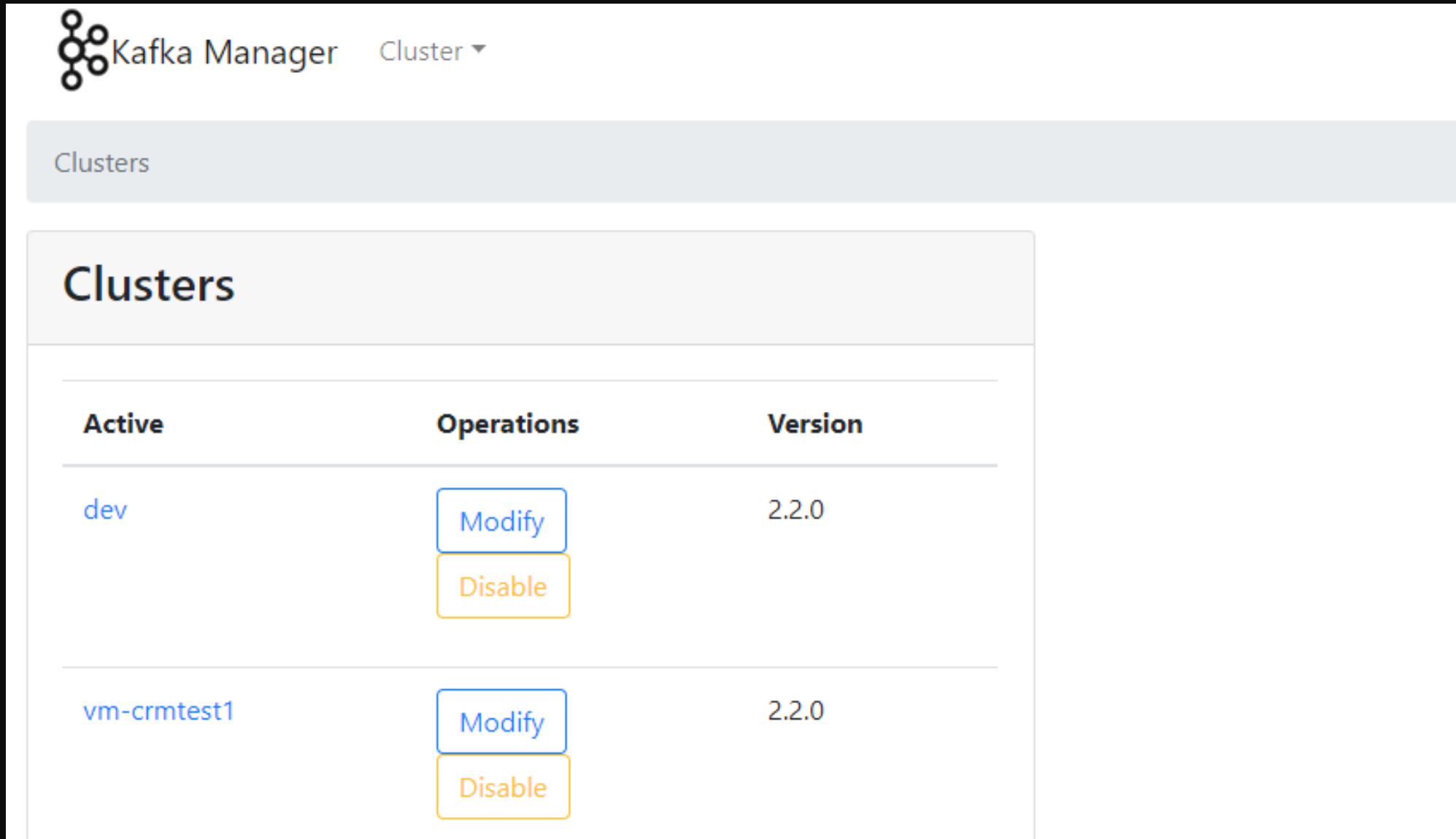
- Доставляем логи, метрики, трассировки и другие сообщения
- > 1.000.000 EPS
- Храним от трех дней
- Позволяем многократно читать данные



План доклада

- Поговорим о базовых архитектурных особенностях кафки
- Поговорим об основных задачах в тестировании систем с кафкой в основе
- Посмотрим конкретные методы, приемы для тестирования, основные ошибки
- Немного затронем эксплуатацию

Kafka-Manager



The screenshot shows the Kafka Manager web interface. At the top left is the Kafka Manager logo and the text "Kafka Manager Cluster". Below this is a "Clusters" header. The main content area is titled "Clusters" and contains a table with two columns: "Active" and "Operations". The table lists two clusters: "dev" and "vm-crmtest1", both with a version of "2.2.0". Each cluster has "Modify" and "Disable" buttons.

Active	Operations	Version
dev	Modify Disable	2.2.0
vm-crmtest1	Modify Disable	2.2.0

Topics

Show 10 entries

Search:

Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %	# Replicas	Under Replicated %	Producer Message/Sec	Summed Recent Offsets
traces_prod	48	9	100	33	0	3	0	483949.11	1,333,232,374,771
traces_cloud	48	9	100	22	0	3	0	114857.03	1,275,827,682,157
legacy_logs_elk_c1	48	9	100	22	0	3	0	37827.55	512,351,541,849
legacy_logs_elk_c2	48	9	100	22	0	3	0	24924.54	339,853,098,453
__consumer_offsets	50	9	100	33	0	3	0	23953.39	214,659,377,620
traces_snitch_prod	48	9	100	33	0	3	0	93003.23	109,618,380,602
traces_snitch_cloud	48	9	100	22	0	3	0	19024.26	53,142,990,772
logsc1_logules	12	9	100	33	0	3	0	5884.71	15,788,817,651
logs_houston_prod_alko	3	5	55	20	0	3	0	1569.52	10,703,013,026

Topic Summary

Replication	3
Number of Partitions	48
Sum of partition offsets	1,333,273,657,954
Total number of Brokers	9
Number of Brokers for Topic	9
Preferred Replicas %	100
Brokers Skewed %	33
Brokers Leader Skewed %	0
Brokers Spread %	100
Under-replicated %	0
Config	Value
retention.ms	86400000

Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	243k	492k	514k	535k
Bytes in /sec	98m	200m	209m	218m
Bytes out /sec	262m	603m	631m	658m
Bytes rejected /sec	0.00	0.00	0.00	0.00

Operations

Delete Topic

Reassign Partitions

Generate Partition Assignments

Add Partitions

Update Config

Manual Partition Assignments

Partitions by Broker

Broker	# of Partitions	# as Leader	Partitions	Skewed?	Leader Skewed?
1	15	5	(5,6,7,10,11,15,24,25,26,31,32,33,37,38,42)	false	false
2	18	6	(0,1,8,9,13,14,18,19,20,27,34,35,36,40,41,45,46,47)	true	false
3	15	5	(2,3,4,12,16,17,21,22,23,28,29,30,39,43,44)	false	false
4	15	5	(4,5,6,9,10,14,23,24,25,30,31,32,36,37,41)	false	false
5	17	5	(0,7,8,12,13,17,18,19,26,33,34,35,39,40,44,45,46)	true	false
6	16	6	(1,2,3,11,15,16,20,21,22,27,28,29,38,42,43,47)	false	false
7	15	5	(3,4,5,9,13,17,22,23,24,29,30,31,36,40,44)	false	false
8	16	5	(6,7,8,11,12,16,18,25,26,32,33,34,38,39,43,45)	false	false
9	17	6	(0,1,2,10,14,15,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47)	true	false

← Brokers

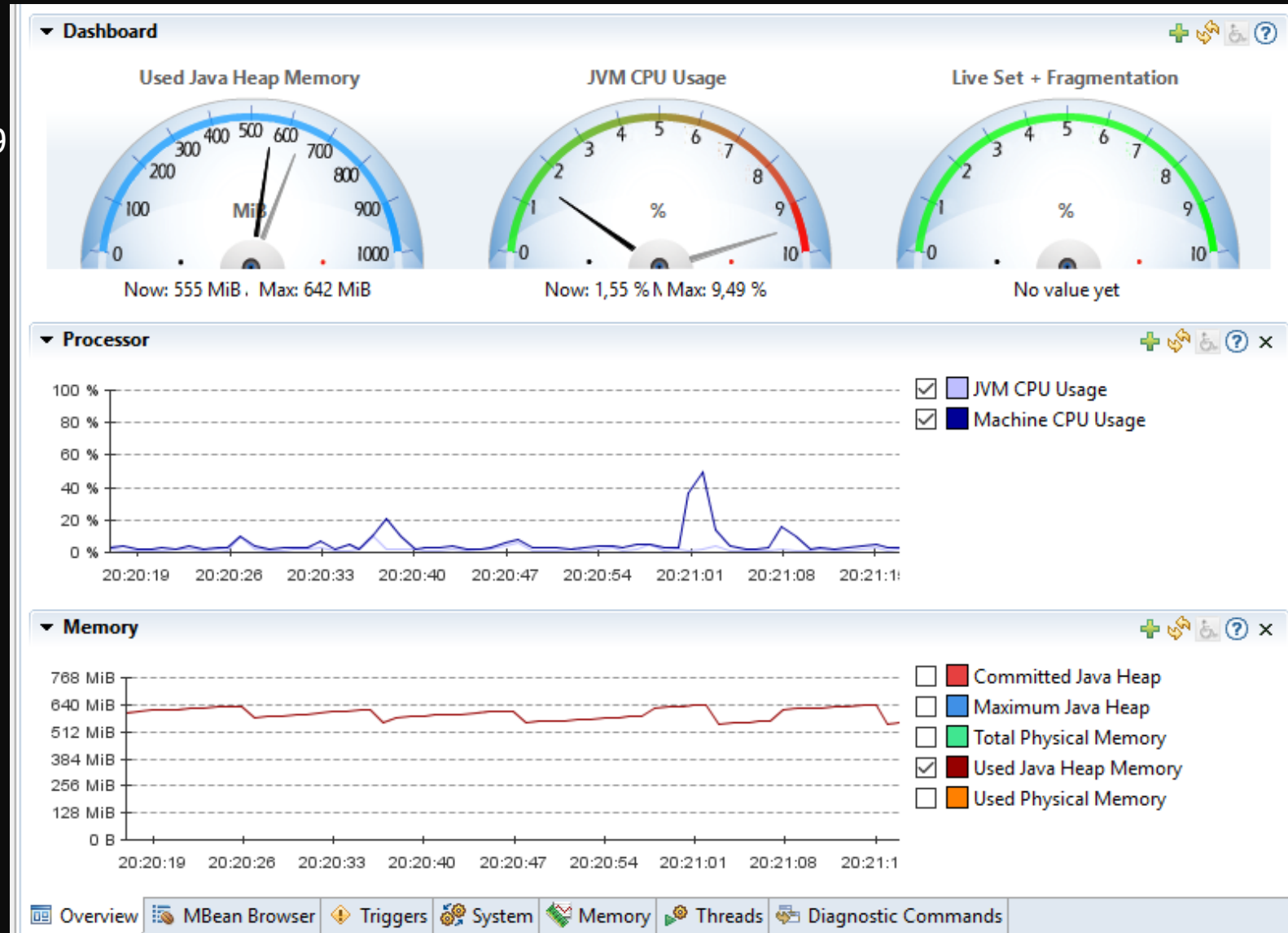
Id	Host	Port	JMX Port	Bytes In	Bytes Out
1	vm-hercules01	PLAINTEXT:9092	9999		
2	vm-hercules02	PLAINTEXT:9092	9999		
3	vm-hercules03	PLAINTEXT:9092	9999		

Combined Metrics

Please enable JMX polling [here](#).

JMX (Java Management eXtention)

- Dcom.sun.management.jmxremote=true
- Dcom.sun.management.jmxremote.port=9999
- Dcom.sun.management.jmxremote.authenticate=false
- Dcom.sun.management.jmxremote.ssl=false
- Djava.rmi.server.hostname=vm-hercules01
- Djava.net.preferIPv4Stack=true



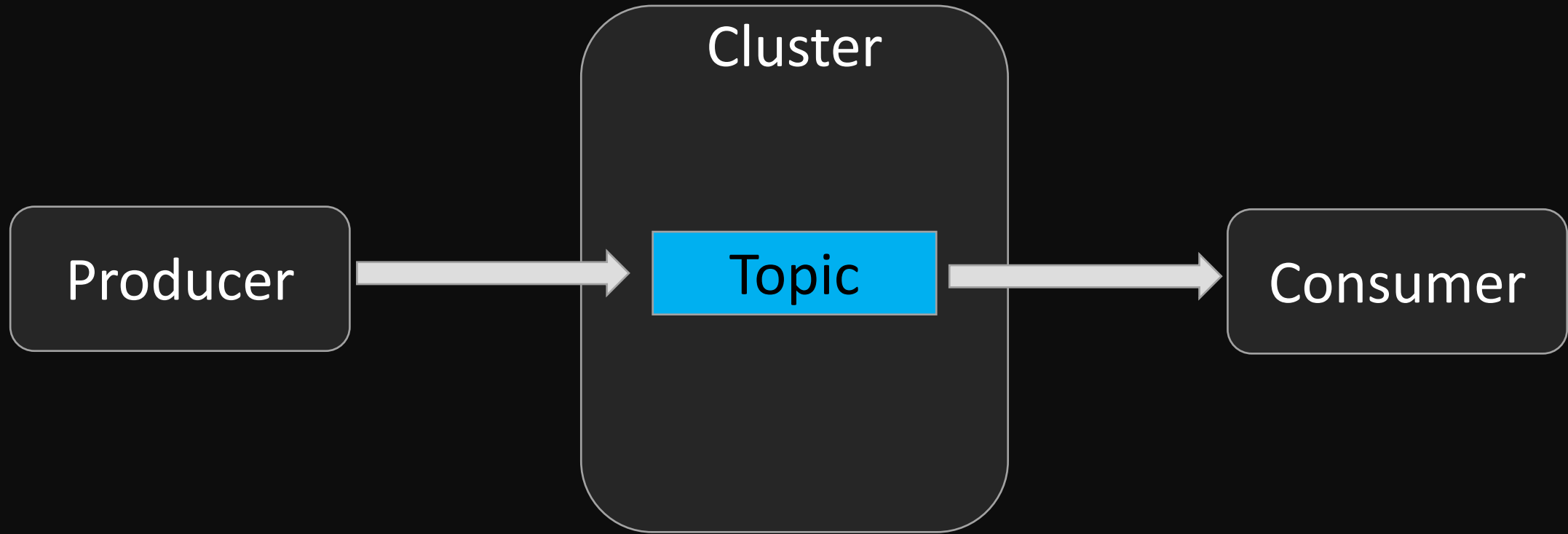
← Brokers

Id	Host	Port	JMX Port	Bytes In	Bytes Out
1	vm-hercules01	PLAINTEXT:9092	9999	0.1k	0.1k
2	vm-hercules02	PLAINTEXT:9092	9999	0.3k	0.3k
3	vm-hercules03	PLAINTEXT:9092	9999	2.1k	2.1k

Combined Metrics

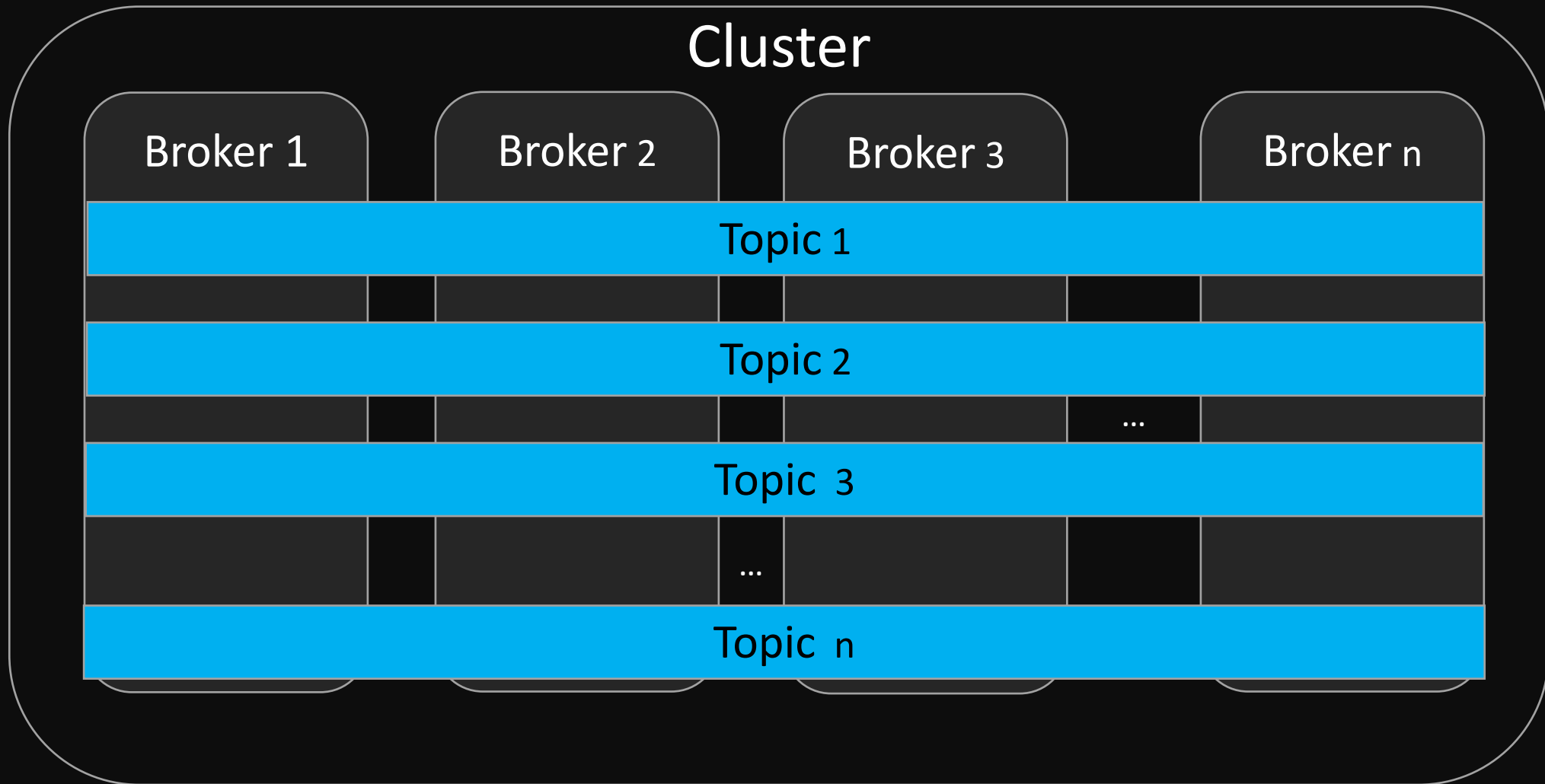
Rate	Mean	1 min	5 min	15 min
Messages in /sec	46.14	22.23	22.15	22.12
Bytes in /sec	22k	2.7k	2.7k	2.7k
Bytes out /sec	11k	2.7k	2.7k	2.7k
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

Producer, cluster, consumer, topic

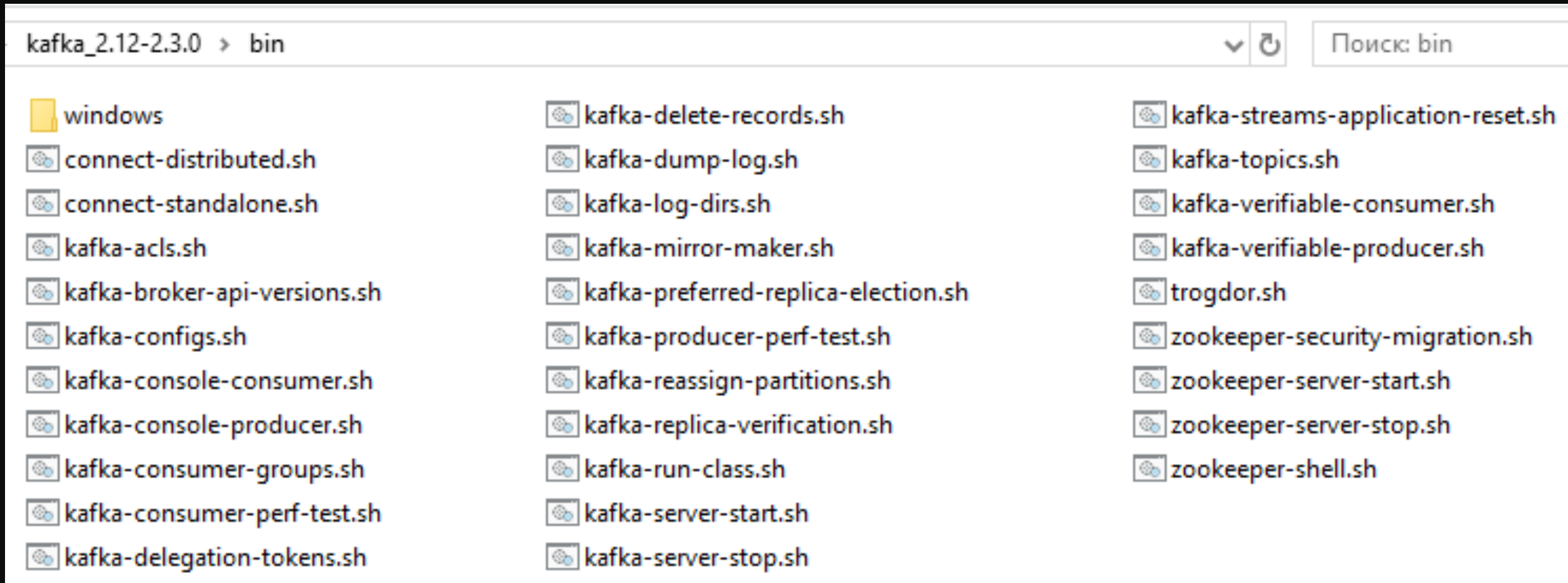


`bootstrap-server = host1:9092, host2:9092, ..., hostN:9092`

Кластер, брокер, топик



Встроенные утилиты



Отправка сообщений

```
kafka-topics.sh --create --bootstrap-server localhost:9092  
--replication-factor 1 --partitions 1 --topic test
```

```
kafka-console-producer.sh --broker-list localhost:9092  
--topic test
```

```
kafka-console-consumer.sh --bootstrap-server localhost:9092  
--topic test
```



```
root@vm-hercules01:/usr/lib/kafka/bin
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-console-producer --broker-list vm-hercules01:9092 --topic test
```

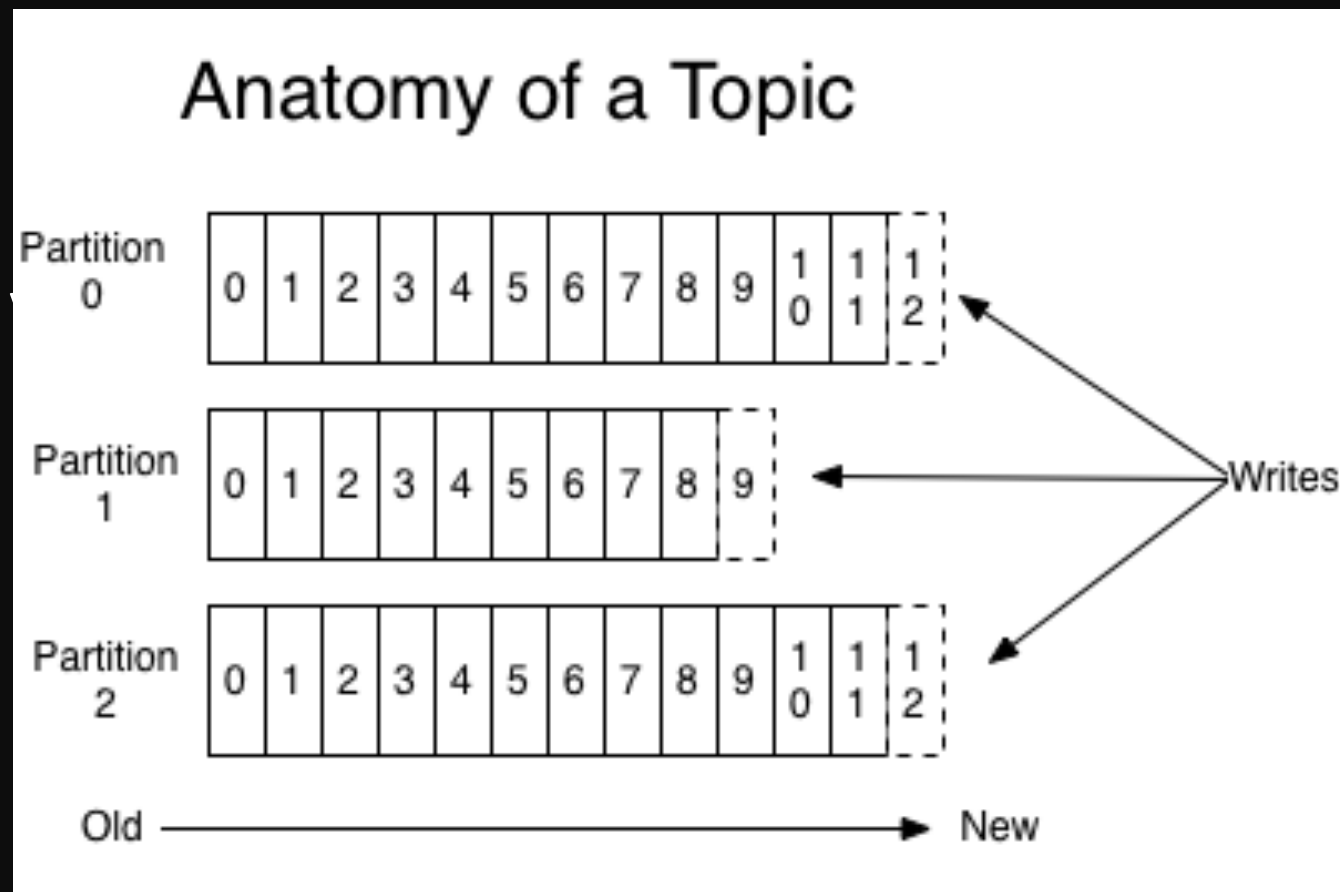
```
root@vm-hercules01:/usr/lib/kafka/bin
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-console-consumer --bootstrap-server vm-hercules01:9092 --topic test
```

Producer, cluster, consumer, topic

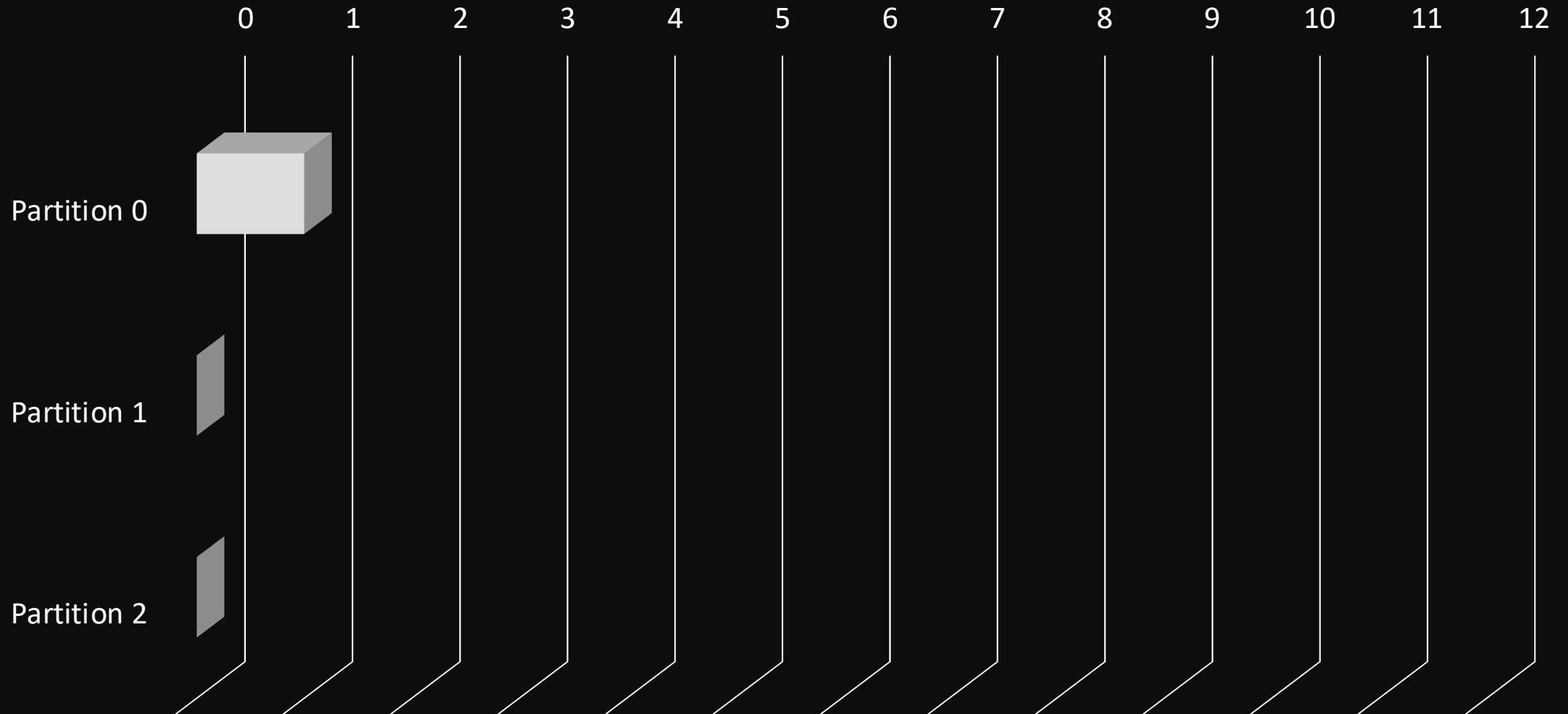
```
build.sbt x BasicSimulation.scala x KafkaProducerExample.scala x logback.xml x gatling.conf x recorder.conf x
1 import java.util.Properties
2 import org.apache.kafka.clients.producer.KafkaProducer
3 import org.apache.kafka.clients.producer.ProducerRecord
4
5 object KafkaProducerExample extends App {
6   val props = new Properties()
7   props.put("bootstrap.servers", "vm-hercules01:9092")
8   props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer")
9   props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer")
10  val producer = new KafkaProducer[String, String](props)
11
12  for (x <- 1 to 100) producer.send(new ProducerRecord[String, String](topic = "test", x.toString))
13 }
14
15
16
```

Партиции, офсеты

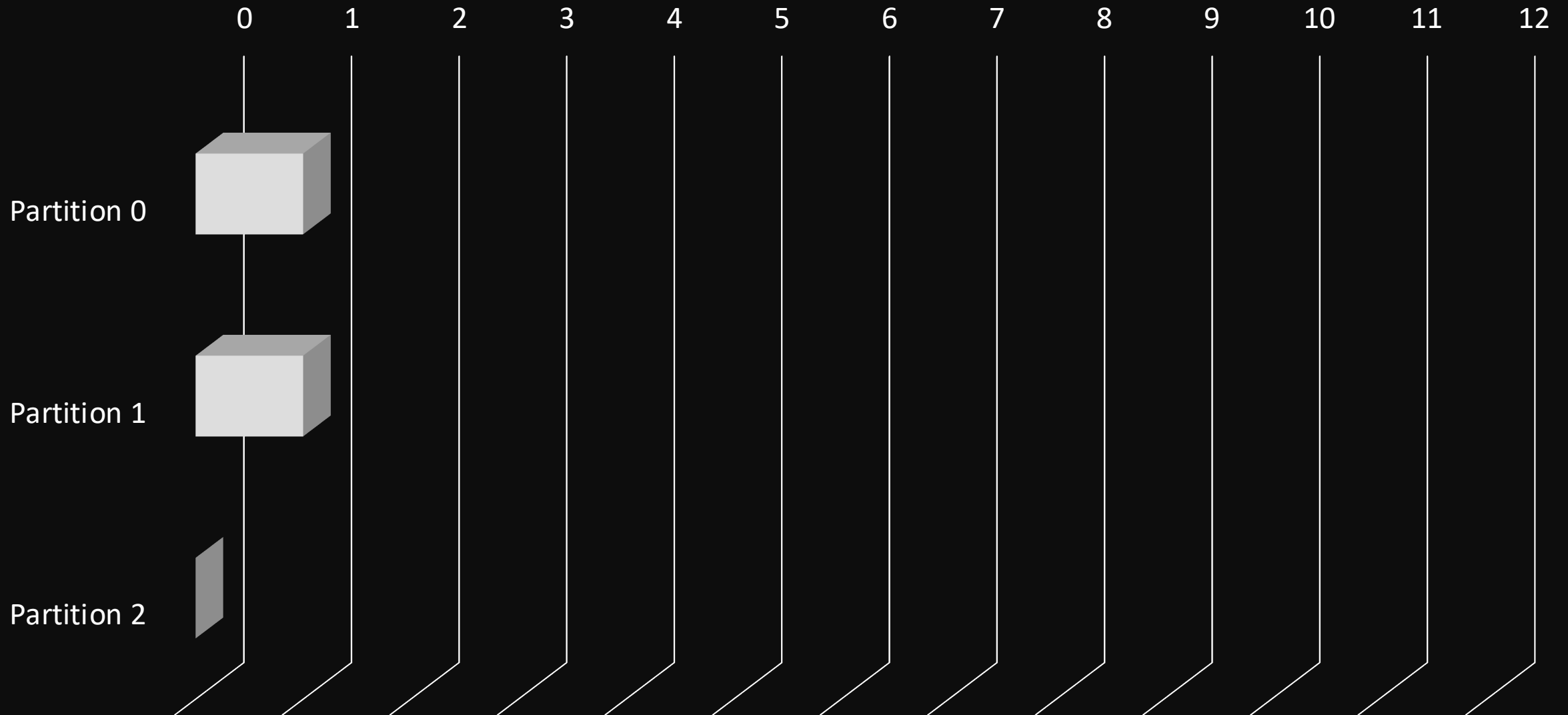
```
kafka-topics.sh --create \  
--bootstrap-server localhost:9092 \  
--replication-factor 1 \  
--partitions 3 --topic test
```



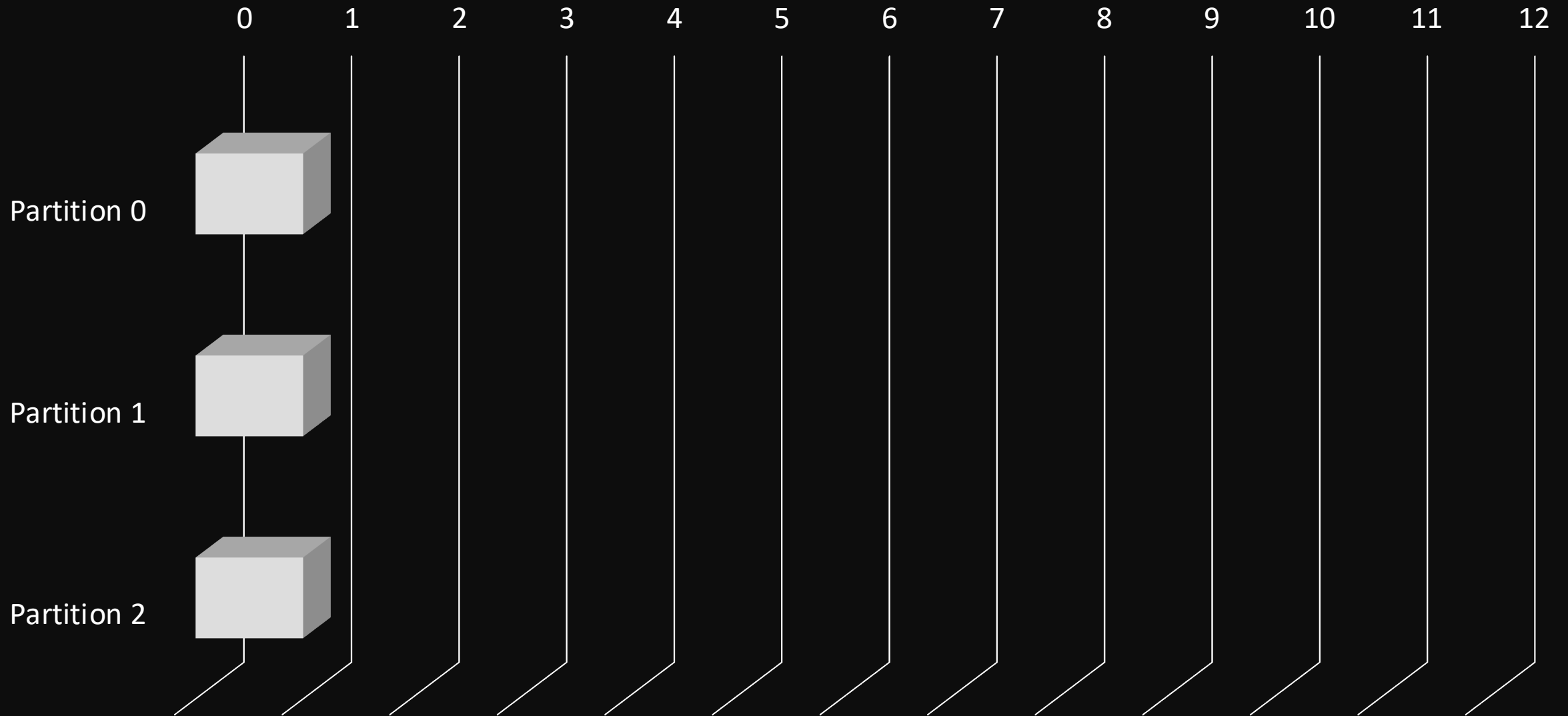
Анатомия топика



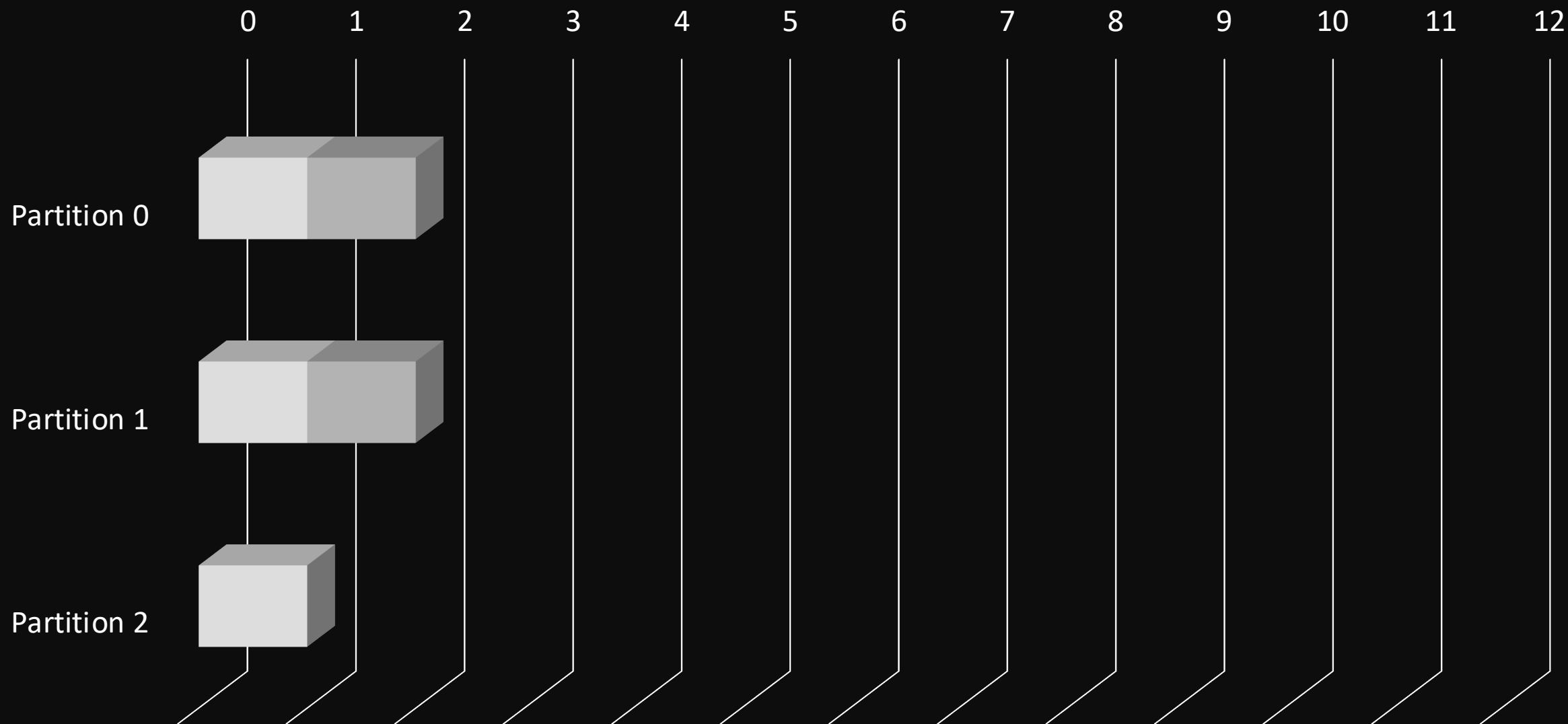
Анатомия топика



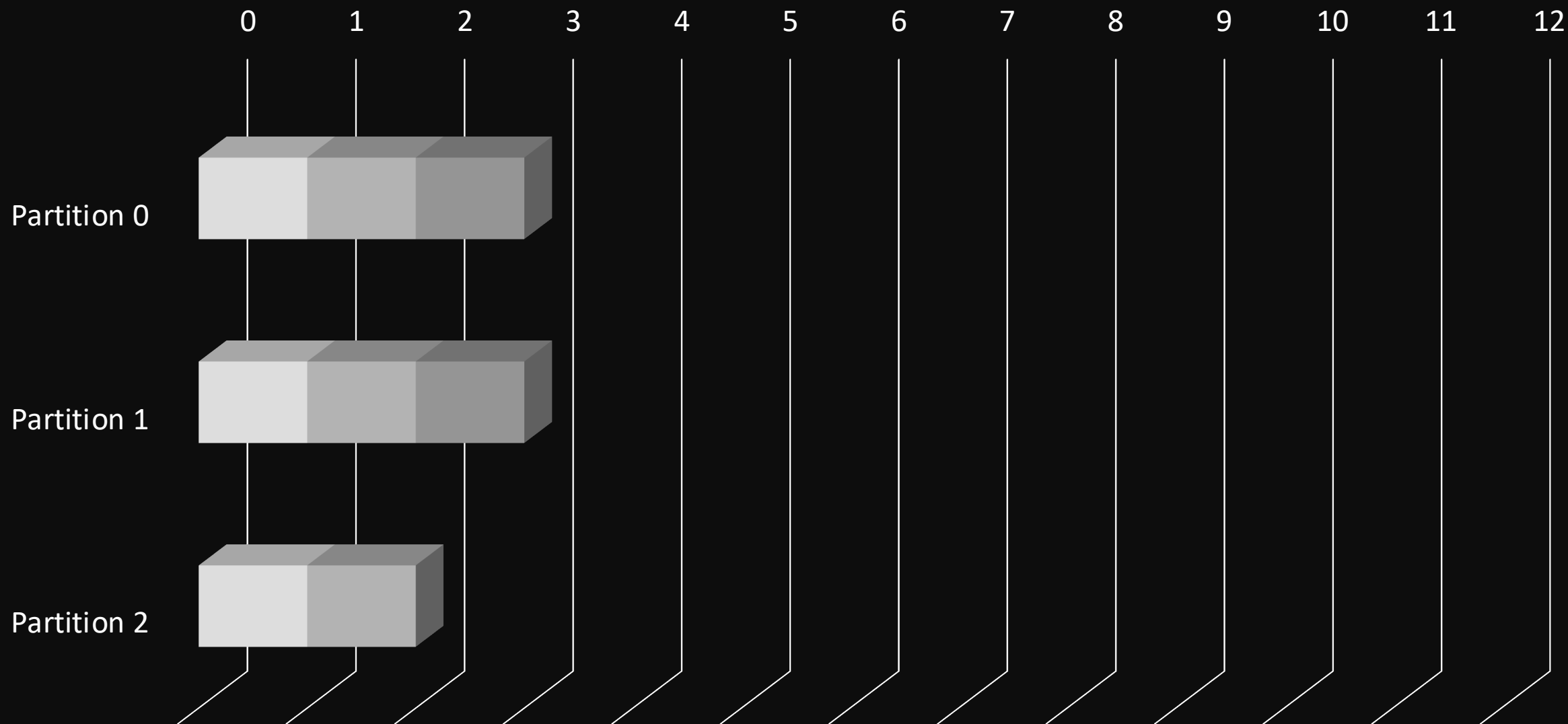
Анатомия топика



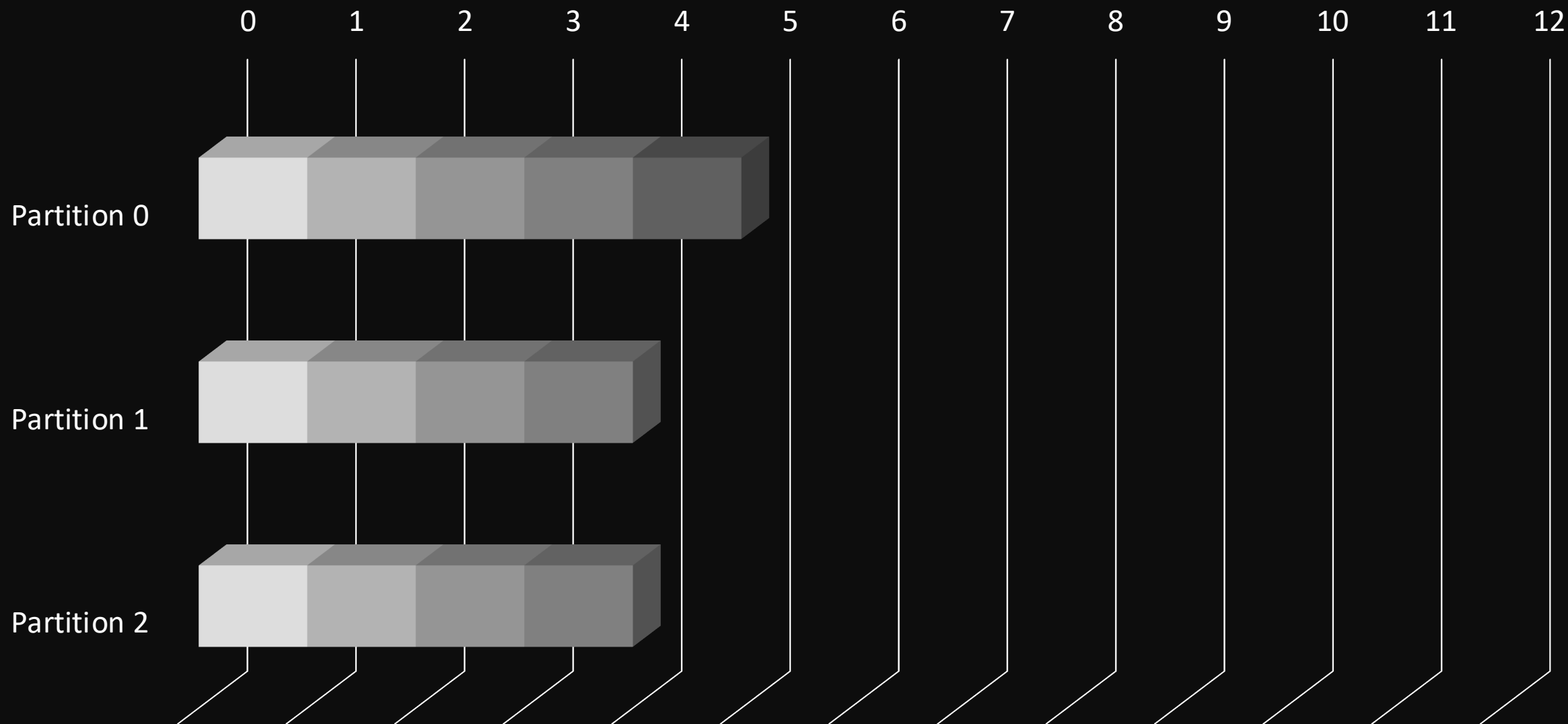
Анатомия топика



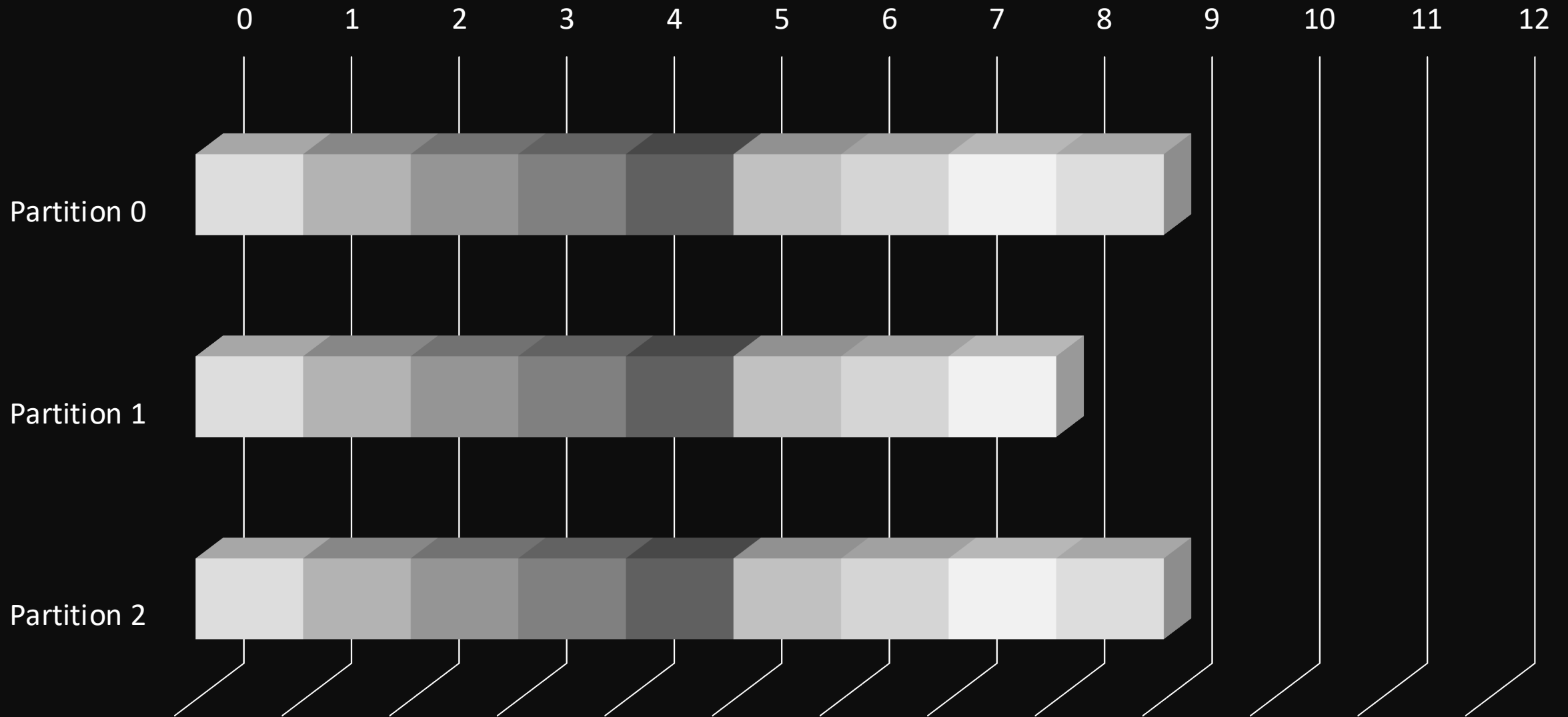
Анатомия топика



Анатомия топика



Анатомия топика



Анатомия топики. Директории партиций

test_stream_api-46	09.09.2019 19:21:04
test_stream_api-47	09.09.2019 19:21:04
test-0	13.11.2019 17:03:21
test-1	13.11.2019 17:03:21
test1p-0	13.11.2019 2:01:47
test-2	13.11.2019 23:29:13
test-3	13.11.2019 23:29:11
test3p-0	13.11.2019 14:03:21
test3p-1	13.11.2019 23:29:13
test3p-2	13.11.2019 14:03:21
test-4	13.11.2019 17:03:21
test5p-0	13.11.2019 23:29:13

Анатомия топика. Директории партиций

Имя	Размер	Ид
00000000000000000000000004.index	10 240 KB	13
00000000000000000000000004.log	0 KB	13
00000000000000000000000004.timeindex	10 240 KB	13
leader-epoch-checkpoint	1 KB	13

```
root@vm-hercules01:/usr/lib/kafka/bin
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-console-producer --broker
-list vm-hercules01:9092 --topic test3p
>
```

```
mahetov@vm-hercules01:~
[mahetov@vm-hercules01 ~]$ tail -f /var/lib/kafka/test3p-0/0000000000000000
00000.log
```

```
mahetov@vm-hercules01:~
[mahetov@vm-hercules01 ~]$ tail -f /var/lib/kafka/test3p-1/0000000000000000
00000.log
```

```
mahetov@vm-hercules01:~
[mahetov@vm-hercules01 ~]$ tail -f /var/lib/kafka/test3p-2/0000000000000000
00000.log
```

Анатомия топика

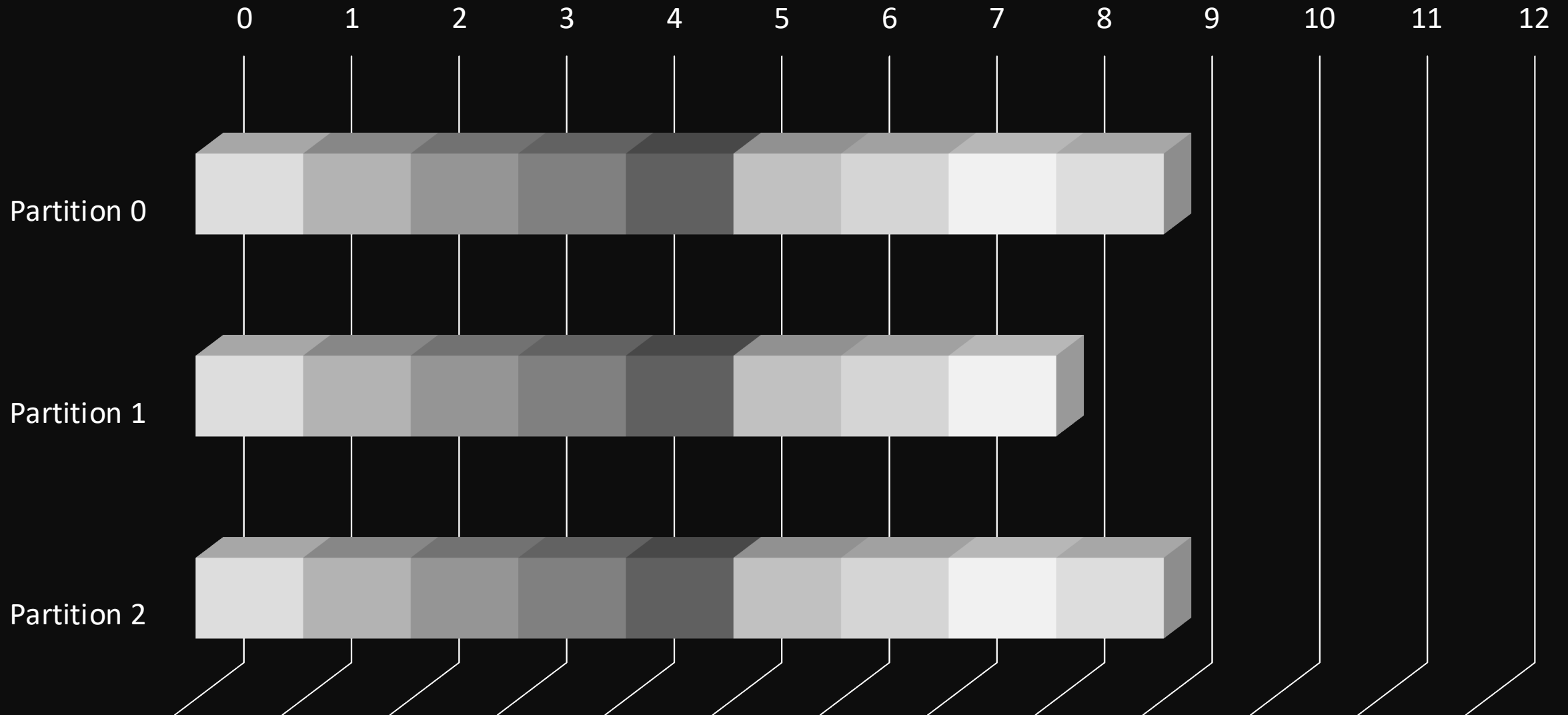
```
root@vm-hercules01:/usr/lib/kafka/bin
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-console-producer --broker ^
>My test message 01
>My test message 02
>My test message 04
>My test message 07
>My test message 10
>My test message 12
>

mahetov@vm-hercules01:~
[mahetov@vm-hercules01 ~]$ clear
[mahetov@vm-hercules01 ~]$ tail -f /var/lib/kafka/test3p-0/0000000000000000
00000.log
J_R_Cnc=-nc=-=====0$My test message 03J6xtnc=S=nc=S=====
0$My test message 06J=nc=i=nc=i=====0$My test message 09J=
$nccEnccE=====0$My test message 12

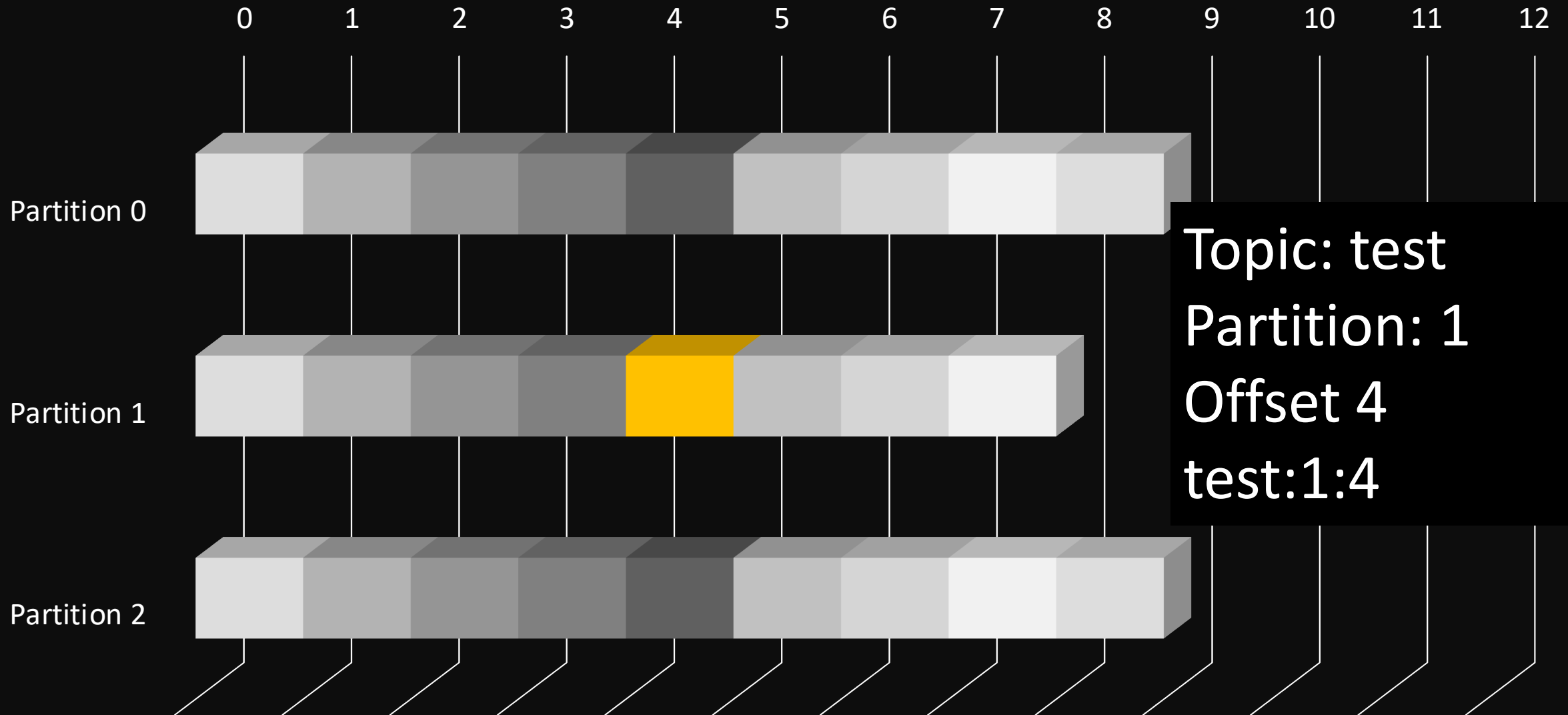
mahetov@vm-hercules01:~
[mahetov@vm-hercules01 ~]$ tail -f /var/lib/kafka/test3p-1/0000000000000000
00000.log
Jx=====0$My test message 01J|nc_E=nc_E=====
0$My test message 04J|nc_E=nc_E=====
0$My test message 07J|nc_vnc=v=====0$
My test message 10

mahetov@vm-hercules01:~
[mahetov@vm-hercules01 ~]$ tail -f /var/lib/kafka/test3p-2/0000000000000000
00000.log
J=nc=7~nc=7~=====0$My test message 02J1|;nc=Lnc=L=====
0$My test message 05J=UKnc=b=nc=b=====0$My test message 08J=
ncÃncÃ=====0$My test message 11
```

Анатомия топика. Офсет



Анатомия топика. Офсет



Партиции, офсеты

← test3p

Topic Summary

Replication	3
Number of Partitions	3
Sum of partition offsets	12

Partition Information

Partition	Latest Offset
0	4
1	4
2	4

Партиции, офсеты. 1000 записей

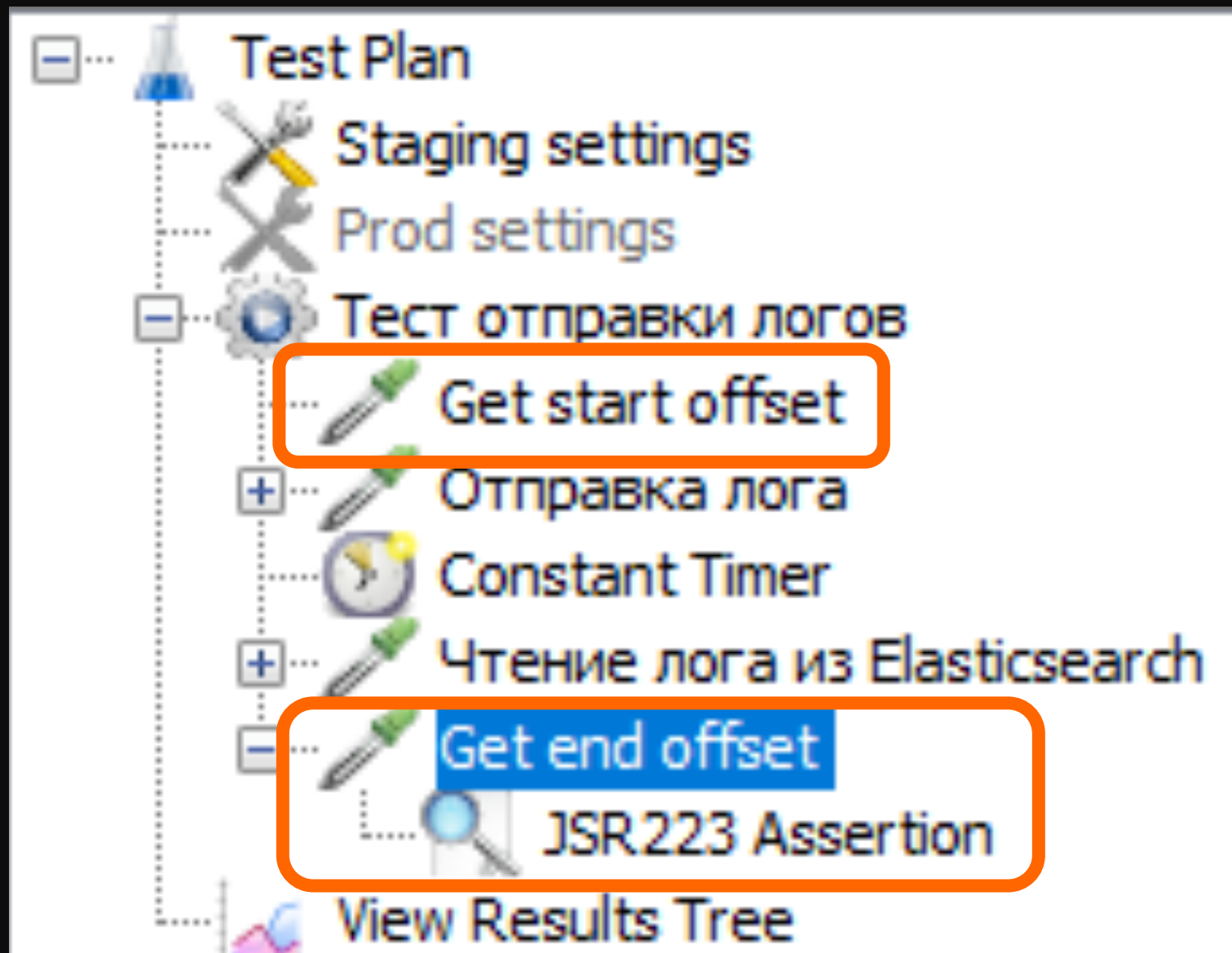
Partition	Latest Offset
0	1,000

Partition	Latest Offset
0	333
1	334
2	333

Функциональное тестирование

```
1 import java.util.Arrays;
2 import java.util.Properties;
3 import org.apache.kafka.clients.consumer.KafkaConsumer;
4 import org.apache.kafka.common.TopicPartition;
5 import org.apache.kafka.common.serialization.StringDeserializer;
6
7 Properties props = new Properties();
8 props.put("bootstrap.servers", "${kafkaHostPort}");
9 props.put("group.id", "consumer-tutorial1");
10 props.put("key.deserializer", StringDeserializer.class.getName());
11 props.put("value.deserializer", StringDeserializer.class.getName());
12 KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
13
14 String topic = "${kafkaTopic}";
15
16 List<TopicPartition> tp = new ArrayList<>();
17 consumer.partitionsFor(topic).each{x -> tp.add(new TopicPartition(topic, x.partition()))};
18 Map<TopicPartition, Long> topicPartitionLongMap = consumer.endOffsets(tp);
19 long offset = 0;
20 topicPartitionLongMap.each{k, v ->
21     offset += v;
22     log.info(v.toString());
23 };
24 consumer.close();
25 vars.put("startOffset", offset.toString())
26 System.out.println(offset.toString());
27 return "Current offset: " + offset.toString()
```

Проверка сдвига офсета



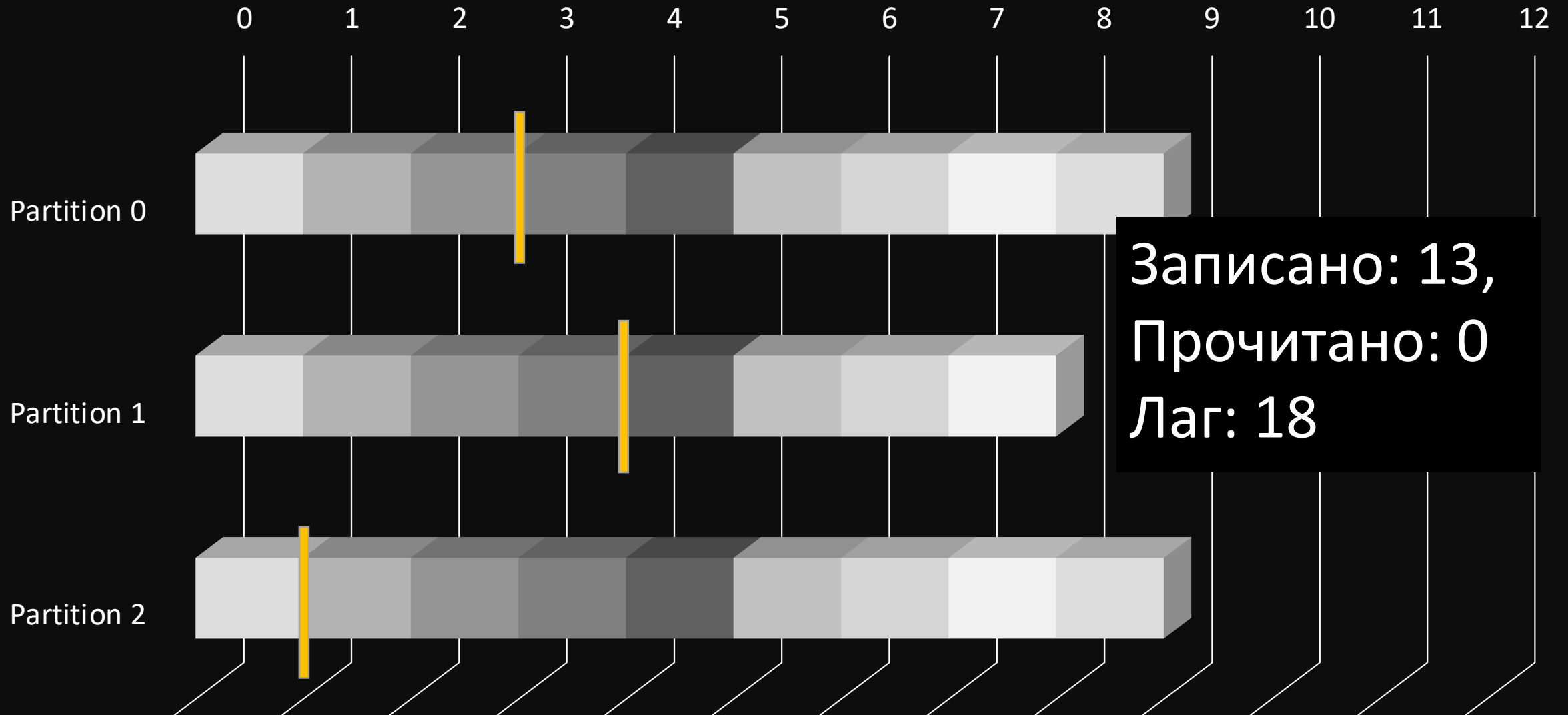
Анатомия топика. Чтение



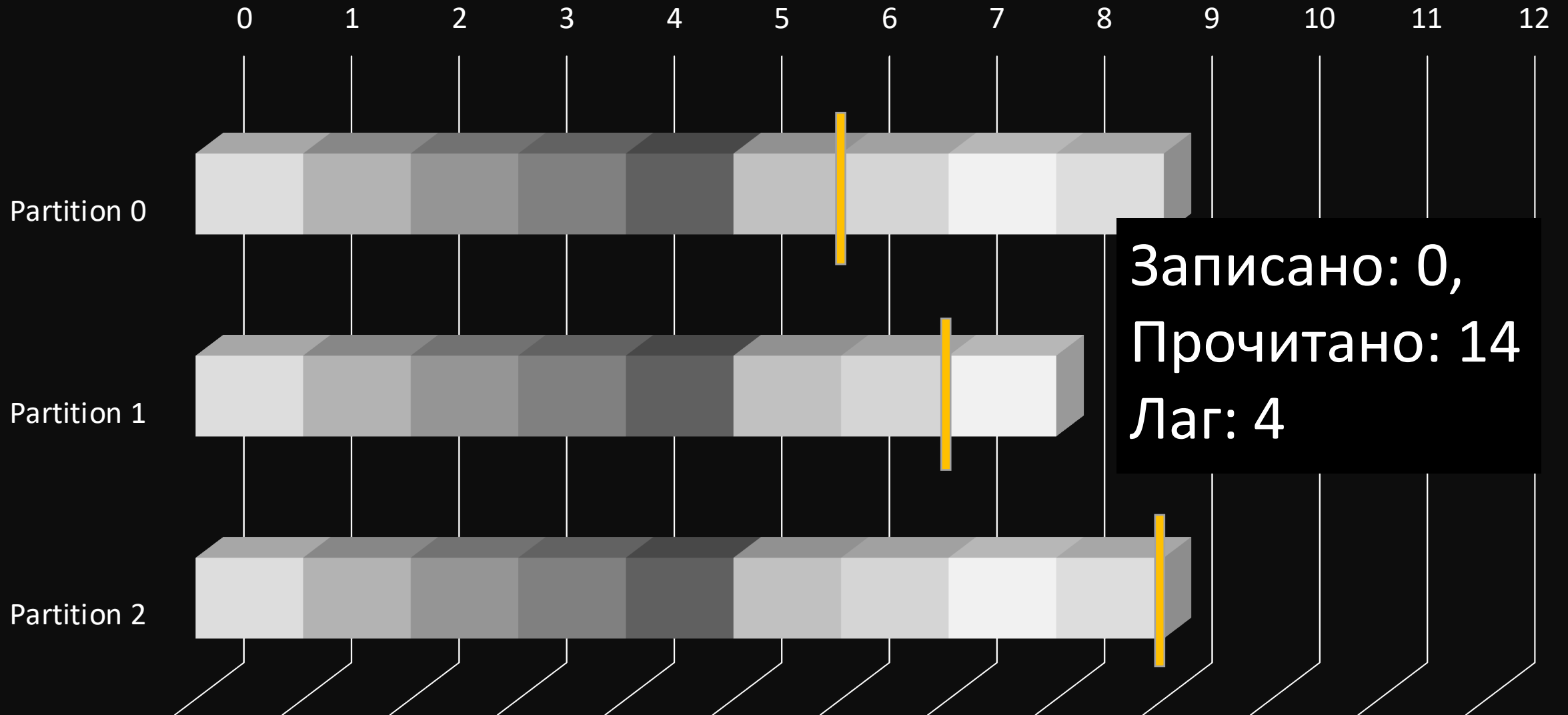
Анатомия топика. Чтение



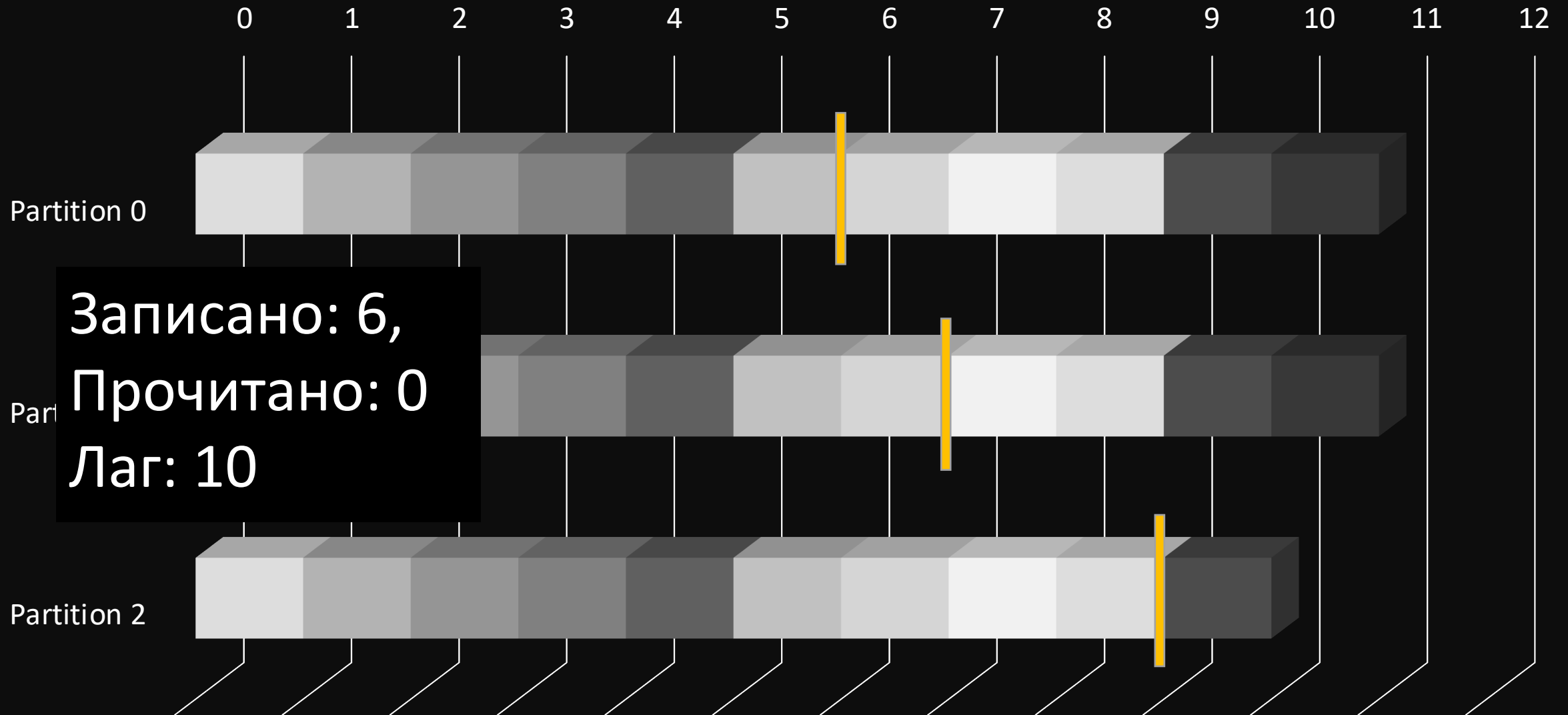
Анатомия топика. Чтение



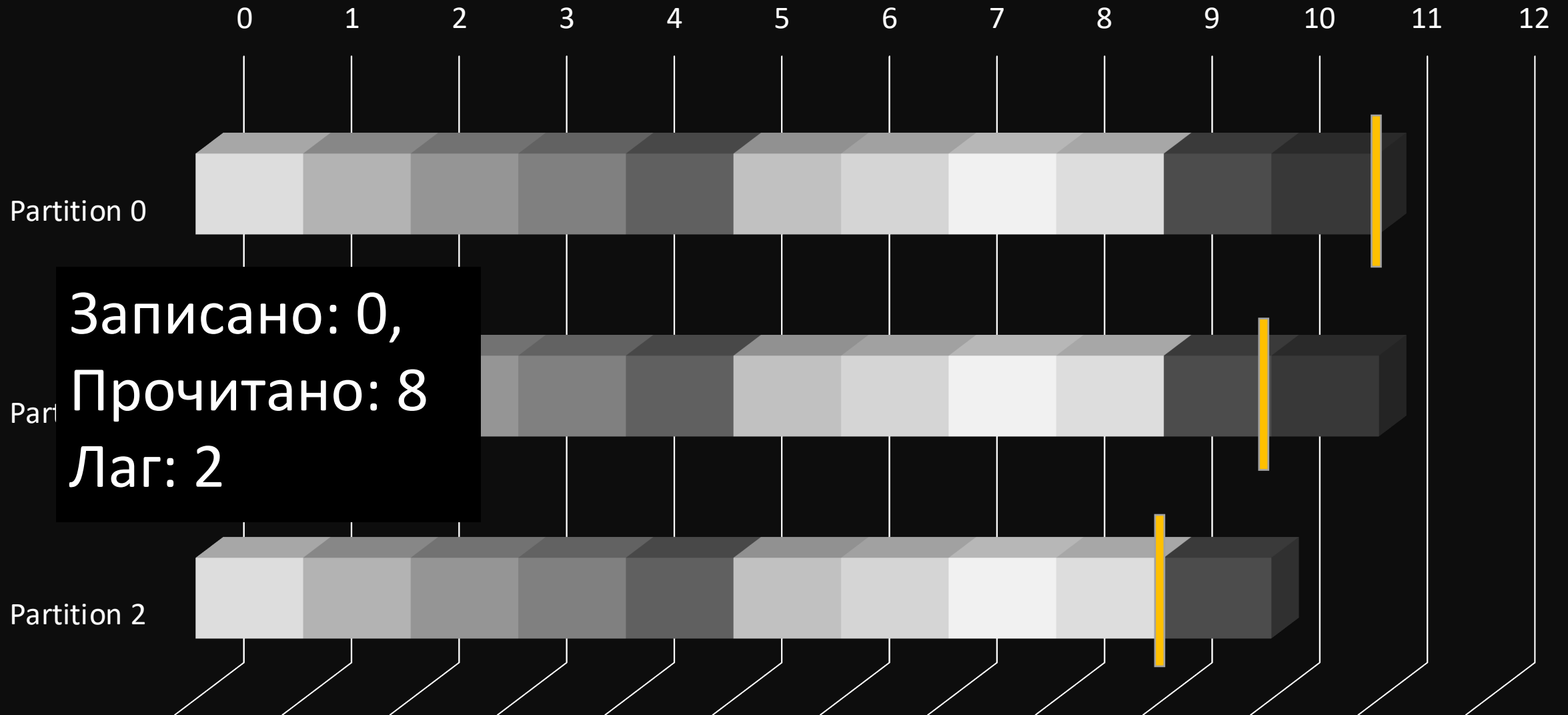
Анатомия топика. Чтение



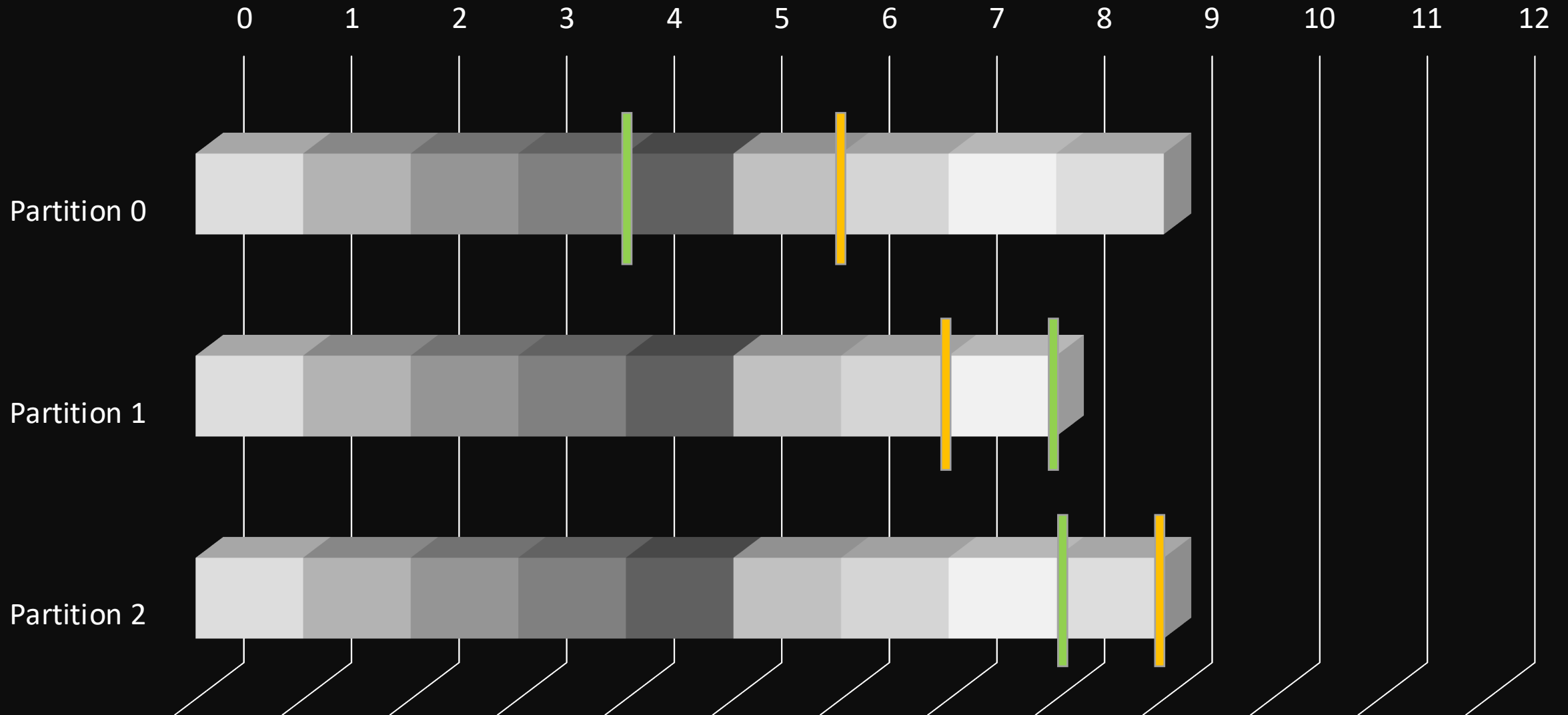
Анатомия топика. Чтение



Анатомия топика. Чтение



Анатомия топика. Два читателя



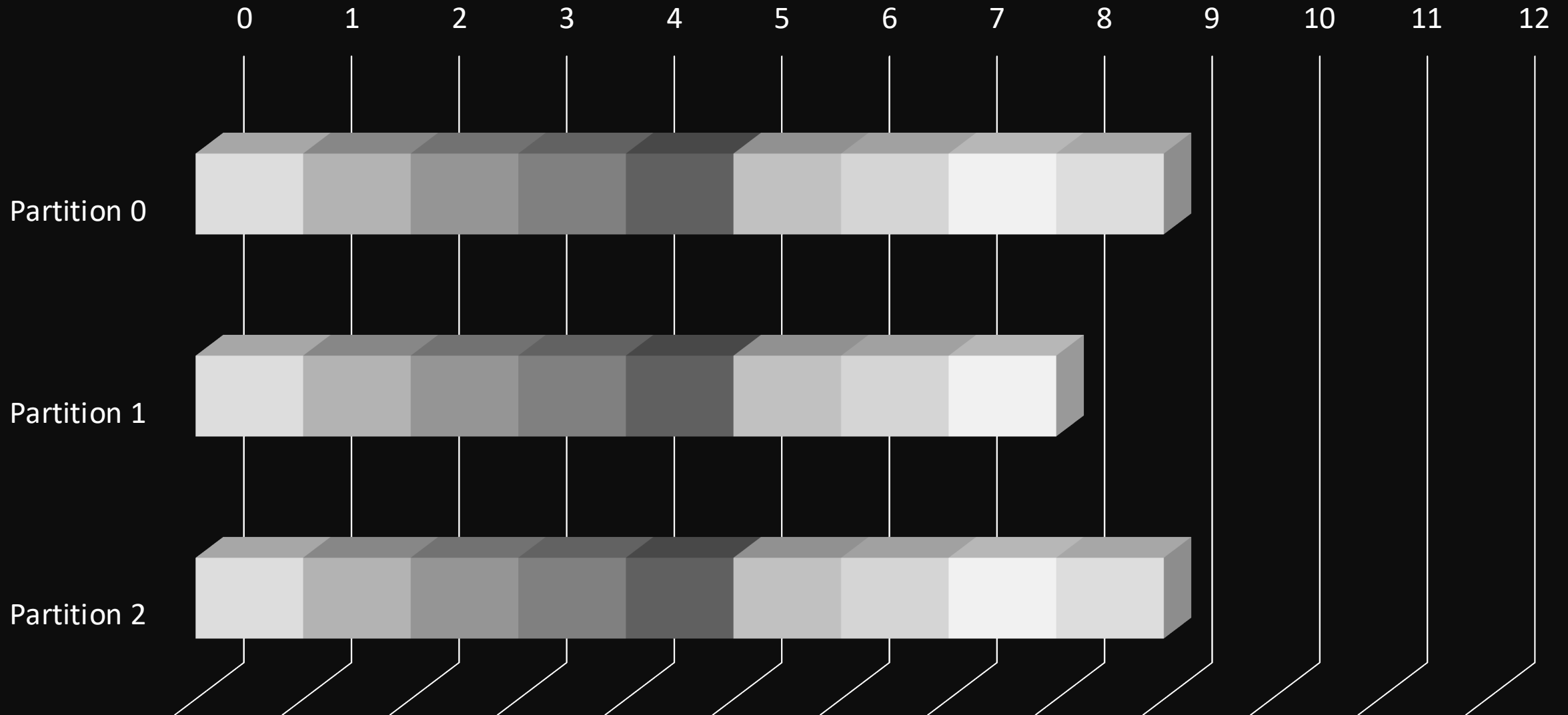
Анатомия топики. Retention

Удаление устаревших записей

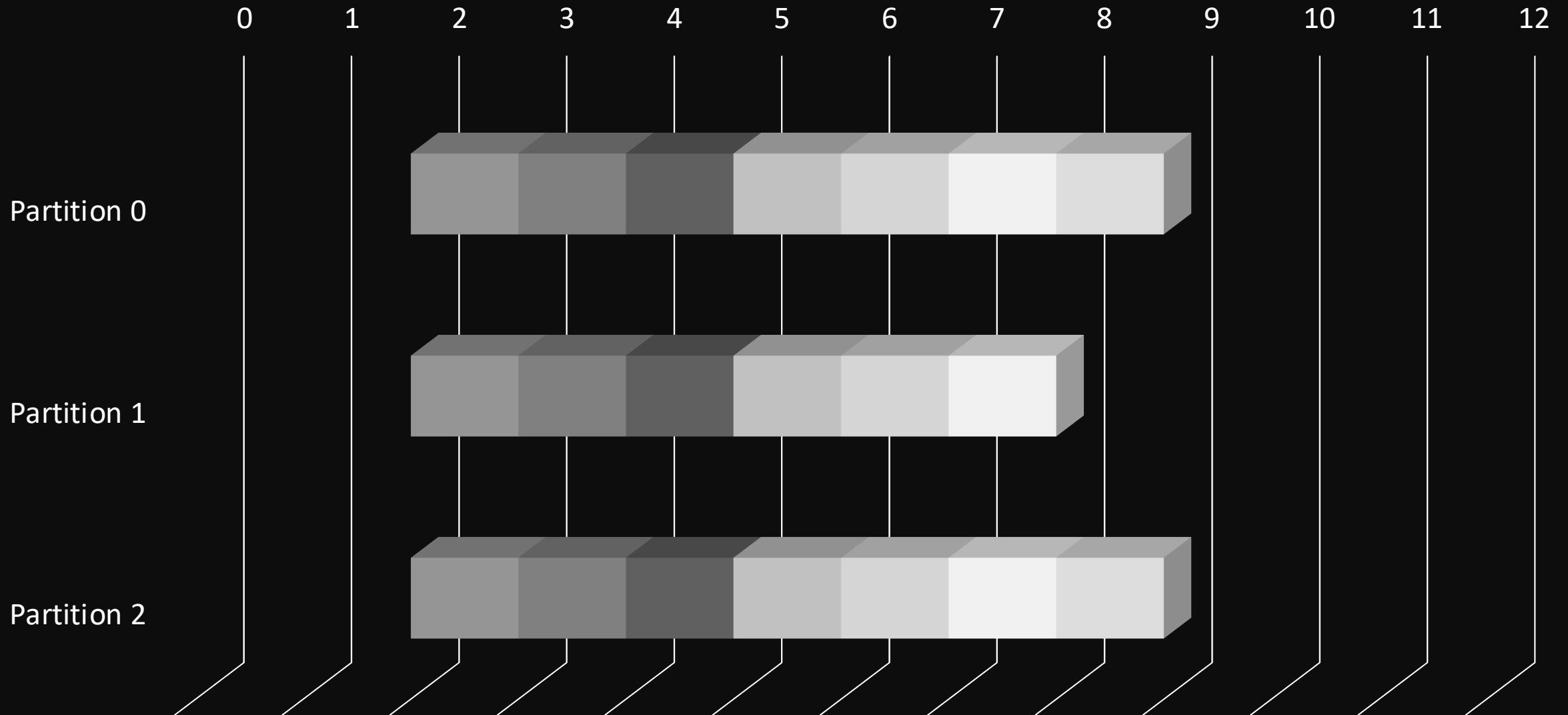
- По размеру партиции
- По времени

Config	Value
retention.ms	259200000
retention.bytes	137438953472

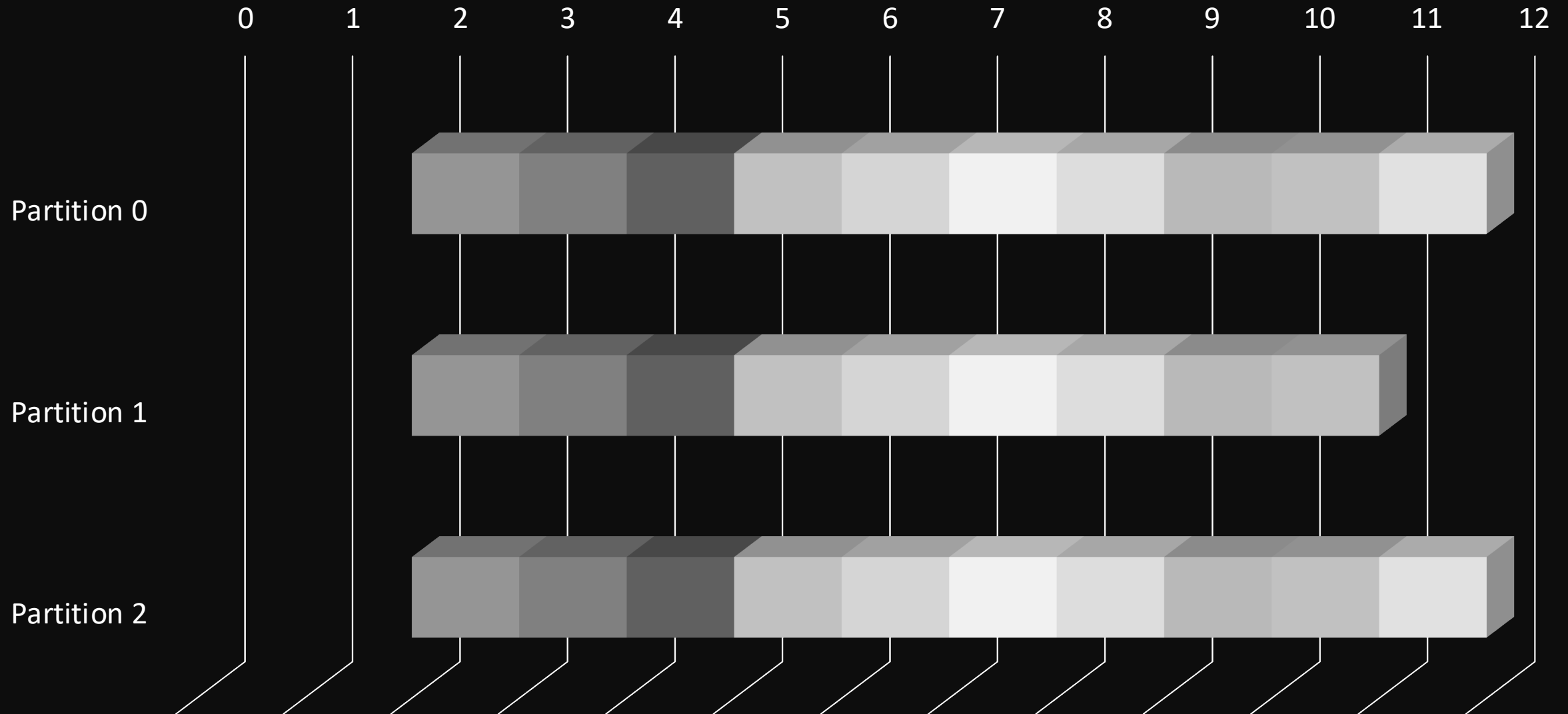
Анатомия топика. Retention



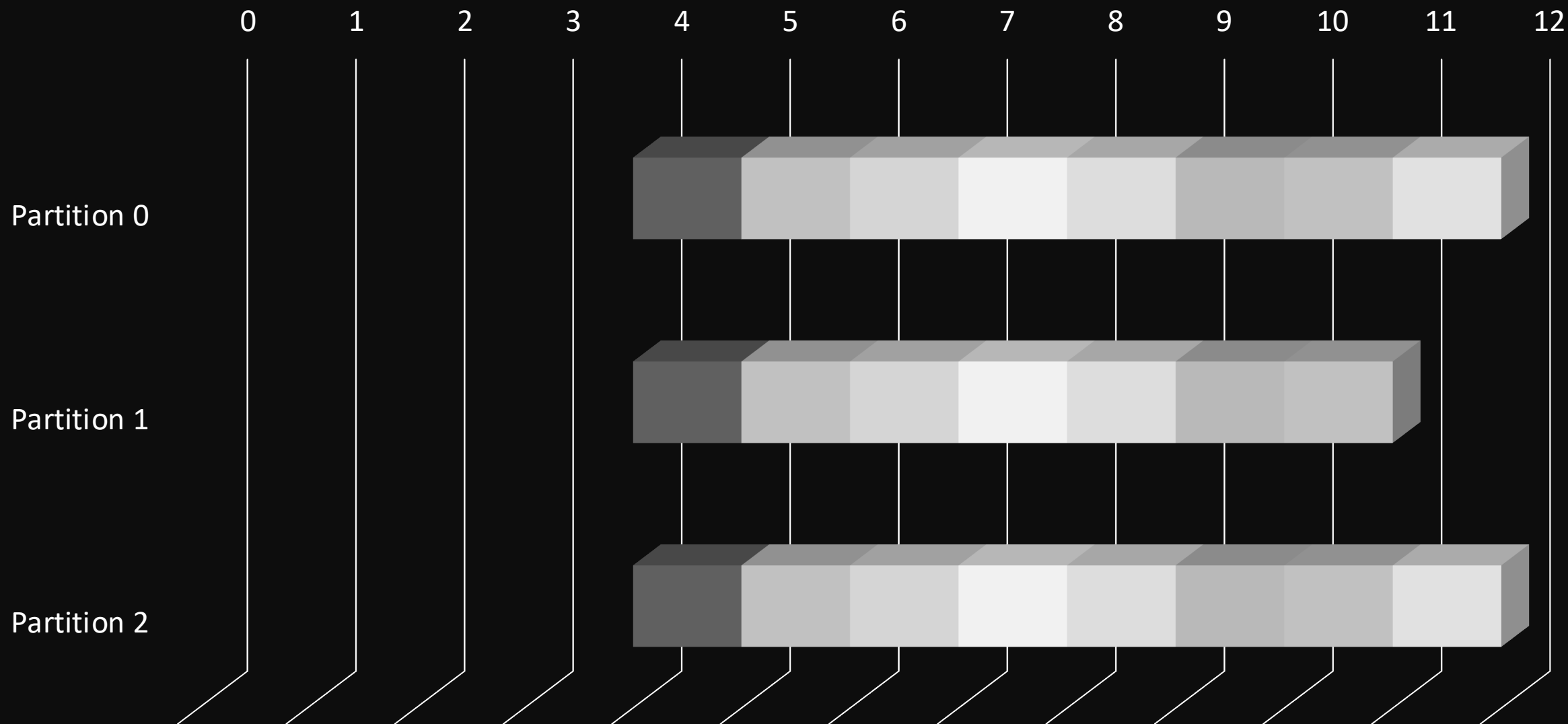
Анатомия топика. Retention



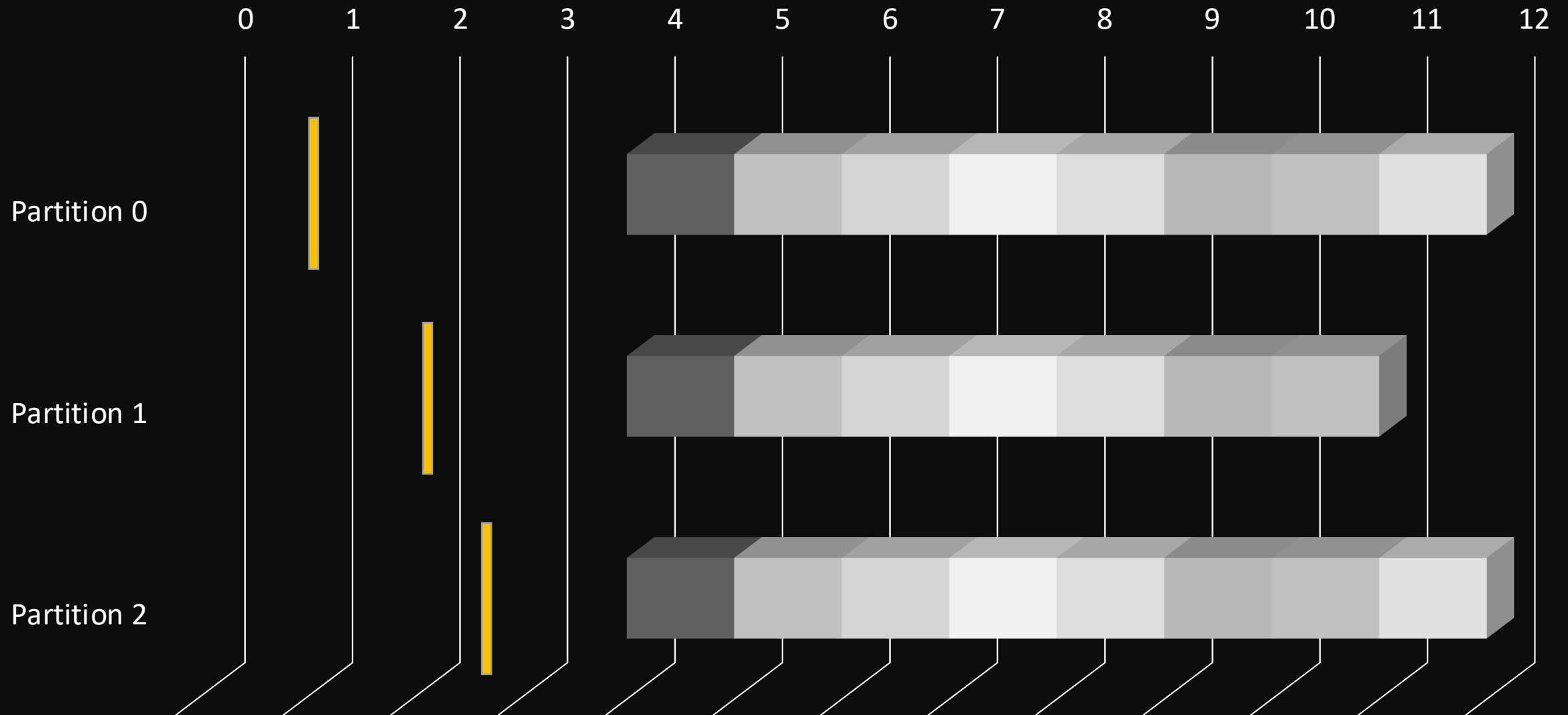
Анатомия топика. Retention



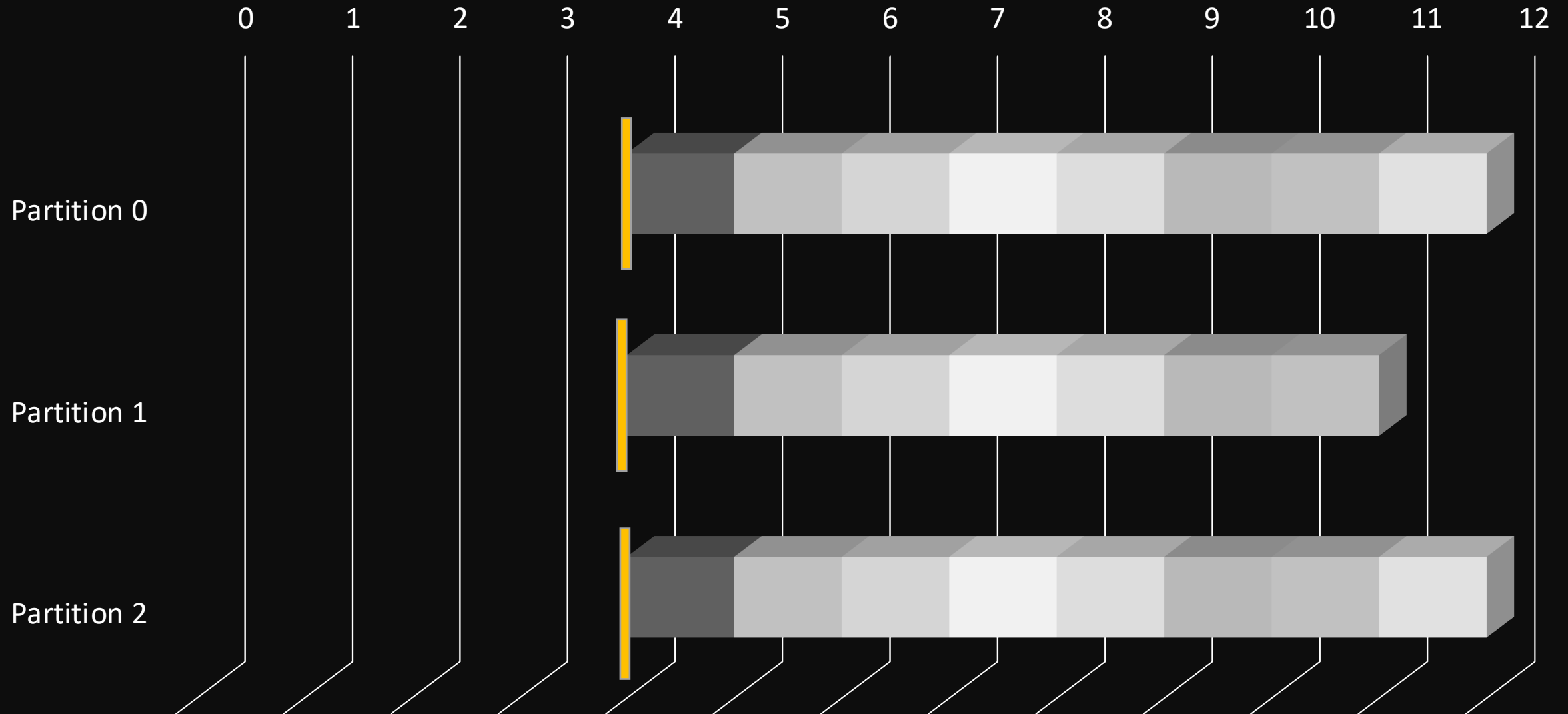
Анатомия топика. Retention

























Анатомия топика. Retention

























Анатомия топика. Retention

























Retention

/storage7/kafka/data/metrics_counters-21				
Имя	Размер	Изменено	Права	Владелец
		01.11.2019 11:30:13	rw-xr-xr-x	kafka
 00000000000661840593.log	199 572 KB	01.11.2019 11:30:16	rw-r--r--	kafka
 00000000000661840593.timeindex	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 00000000000661840593.index	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 leader-epoch-checkpoint	1 KB	01.11.2019 10:56:11	rw-r--r--	kafka
 00000000000661840593.snapshot	1 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.timeindex	1 708 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.log	1 048 575 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.index	1 203 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.snapshot	1 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.timeindex	1 593 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.log	1 048 576 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.index	1 141 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000654153878.timeindex	920 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.log	1 048 567 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.index	738 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000651573139.timeindex	1 683 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.log	1 048 571 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.index	1 177 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000648989968.timeindex	1 844 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.log	1 048 576 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.index	1 272 KB	31.10.2019 1:39:14	rw-r--r--	kafka























Retention

/storage7/kafka/data/metrics_counters-21				
Имя	Размер	Изменено	Права	Владелец
		01.11.2019 11:30:13	rw-xr-xr-x	kafka
 00000000000661840593.log	199 572 KB	01.11.2019 11:30:16	rw-r--r--	kafka
 00000000000661840593.timeindex	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 00000000000661840593.index	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 leader-epoch-checkpoint	1 KB	01.11.2019 10:56:11	rw-r--r--	kafka
 00000000000661840593.snapshot	1 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.timeindex	1 708 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.log	1 048 575 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.index	1 203 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.snapshot	1 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.timeindex	1 593 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.log	1 048 576 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.index	1 141 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000654153878.timeindex	920 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.log	1 048 567 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.index	738 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000651573139.timeindex	1 683 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.log	1 048 571 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.index	1 177 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000648989968.timeindex	1 844 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.log	1 048 576 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.index	1 272 KB	31.10.2019 1:39:14	rw-r--r--	kafka























Retention

/storage7/kafka/data/metrics_counters-21				
Имя	Размер	Изменено	Права	Владелец
		01.11.2019 11:30:13	rw-xr-xr-x	kafka
 00000000000661840593.log	199 572 KB	01.11.2019 11:30:16	rw-r--r--	kafka
 00000000000661840593.timeindex	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 00000000000661840593.index	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 leader-epoch-checkpoint	1 KB	01.11.2019 10:56:11	rw-r--r--	kafka
 00000000000661840593.snapshot	1 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.timeindex	1 708 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.log	1 048 575 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.index	1 203 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.snapshot	1 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.timeindex	1 593 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.log	1 048 576 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.index	1 141 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000654153878.timeindex	920 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.log	1 048 567 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.index	738 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000651573139.timeindex	1 683 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.log	1 048 571 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.index	1 177 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000648989968.timeindex	1 844 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.log	1 048 576 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.index	1 272 KB	31.10.2019 1:39:14	rw-r--r--	kafka

Retention

/storage7/kafka/data/metrics_counters-21				
Имя	Размер	Изменено	Права	Владелец
		01.11.2019 11:30:13	rw-xr-x	kafka
 00000000000661840593.log	199 572 KB	01.11.2019 11:30:16	rw-r--r--	kafka
 00000000000661840593.timeindex	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 00000000000661840593.index	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 leader-epoch-checkpoint	1 KB	01.11.2019 10:56:11	rw-r--r--	kafka
 00000000000661840593.snapshot	1 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.timeindex	1 708 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.log	1 048 575 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.index	1 203 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.snapshot	1 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.timeindex	1 593 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.log	1 048 576 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.index	1 141 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000654153878.timeindex	920 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.log	1 048 567 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.index	738 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000651573139.timeindex	1 683 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.log	1 048 571 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.index	1 177 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000648989968.timeindex	1 844 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.log	1 048 576 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.index	1 272 KB	31.10.2019 1:39:14	rw-r--r--	kafka

Retention

/storage7/kafka/data/metrics_counters-21				
Имя	Размер	Изменено	Права	Владелец
		01.11.2019 11:30:13	rw-xr-xr-x	kafka
 00000000000661840593.log	199 572 KB	01.11.2019 11:30:16	rw-r--r--	kafka
 00000000000661840593.timeindex	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 00000000000661840593.index	10 240 KB	01.11.2019 11:30:05	rw-r--r--	kafka
 leader-epoch-checkpoint	1 KB	01.11.2019 10:56:11	rw-r--r--	kafka
 00000000000661840593.snapshot	1 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.timeindex	1 708 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.log	1 048 575 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.index	1 203 KB	01.11.2019 10:05:50	rw-r--r--	kafka
 00000000000659287987.snapshot	1 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.timeindex	1 593 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.log	1 048 576 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000656729057.index	1 141 KB	01.11.2019 2:35:10	rw-r--r--	kafka
 00000000000654153878.timeindex	920 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.log	1 048 567 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000654153878.index	738 KB	31.10.2019 18:55:06	rw-r--r--	kafka
 00000000000651573139.timeindex	1 683 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.log	1 048 571 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000651573139.index	1 177 KB	31.10.2019 10:40:06	rw-r--r--	kafka
 00000000000648989968.timeindex	1 844 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.log	1 048 576 KB	31.10.2019 1:39:14	rw-r--r--	kafka
 00000000000648989968.index	1 272 KB	31.10.2019 1:39:14	rw-r--r--	kafka

Анатомия топика. Retention

- Лучше настроить явно (по умолчанию – 7 дней)
- Возможна настройка и по времени и по объему
- Срабатывает не мгновенно

Производительность

Зачем тестировать?

- Потолок производительности
- Оптимальные параметры
- Планирование ресурсов

Производительность

Как увеличить?

- Горизонтальное расширение
 - Увеличение количества партиций
 - Увеличение количества брокеров
- Тюнинг опций

Генерация нагрузки. Kafka-producer-perf-test

```
bin/kafka-producer-perf-test.sh \  
--producer-props bootstrap.servers=localhost:9092 \  
--topic test \  
--num-records 500000 \  
--throughput 100000 \  
--payload-file /home/mahetov/payload.txt
```

Генерация нагрузки. Kafka-producer-perf-test

```
root@vm-hercules01:/usr/lib/kafka/bin
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-producer-perf-test --topic gate_test_0 --num-records 750000 --throughput 100000 --payload-file /home/mahetov/payload.txt --producer-props bootstrap.servers=vm-hercules01:9092
```

Генерация нагрузки. Kafka-producer-perf-test

```
750000 records sent, 26810.609852 records/sec (26.03 MB/sec), 1075.51  
ms avg latency, 5683.00 ms max latency, 11 ms 50th, 3603 ms 95th, 54  
96 ms 99th, 5660 ms 99.9th.
```

Генерация нагрузки. Kafka-consumer-perf-test

```
bin/kafka-consumer-perf-test.sh \  
--broker-list localhost:9092 \  
--topic test  
--messages 500000
```

Генерация нагрузки. Kafka-consumer-perf-test

```
root@vm-hercules01:/usr/lib/kafka/bin
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-consumer-perf-test -
-broker-list vm-hercules01:9092 --topic test5p --messages 250000
start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.n
Msg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.
nMsg.sec
2019-11-11 18:33:59:242, 2019-11-11 18:34:04:354, 243.0023, 47.5357,
250301, 48963.4194, 3042, 2070, 117.3924, 120918.3575
[root@vm-hercules01 /usr/lib/kafka/bin]#
```

Генерация нагрузки. Kafka-producer-perf-test

```
root@vm-hercules01:/usr/lib/kafka/bin

[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-producer-perf-test --topic loadtestp1 --num-records 750000
--throughput 100000 --payload-file /home/mahetov/payload.txt --producer-props bootstrap.servers=vm-hercule
s01:9092
Reading payloads from: /home/mahetov/payload.txt
Number of messages read: 1
165856 records sent, 33171.2 records/sec (32.20 MB/sec), 753.7 ms avg latency, 1050.0 ms max latency.
214275 records sent, 42855.0 records/sec (41.61 MB/sec), 722.3 ms avg latency, 825.0 ms max latency.
213540 records sent, 42708.0 records/sec (41.46 MB/sec), 714.8 ms avg latency, 847.0 ms max latency.
750000 records sent, 40156.342025 records/sec (38.99 MB/sec) 723.97 ms avg latency, 1050.00 ms max latency
, 722 ms 50th, 842 ms 95th, 972 ms 99th, 1045 ms 99.9th.
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-producer-perf-test --topic loadtestp3 --num-records 750000
--throughput 100000 --payload-file /home/mahetov/payload.txt --producer-props bootstrap.servers=vm-hercule
s01:9092
Reading payloads from: /home/mahetov/payload.txt
Number of messages read: 1
369966 records sent, 73993.2 records/sec (71.84 MB/sec), 342.7 ms avg latency, 1074.0 ms max latency.
750000 records sent, 88558.271343 records/sec (85.98 MB/sec) 311.38 ms avg latency, 1074.00 ms max latency
, 154 ms 50th, 890 ms 95th, 1034 ms 99th, 1059 ms 99.9th.
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-producer-perf-test --topic loadtestp5 --num-records 750000
--throughput 100000 --payload-file /home/mahetov/payload.txt --producer-props bootstrap.servers=vm-hercule
s01:9092
Reading payloads from: /home/mahetov/payload.txt
Number of messages read: 1
434315 records sent, 86863.0 records/sec (84.33 MB/sec), 286.5 ms avg latency, 1115.0 ms max latency.
750000 records sent, 99773.845949 records/sec (96.86 MB/sec) 234.60 ms avg latency, 1115.00 ms max latency
, 70 ms 50th, 800 ms 95th, 1042 ms 99th, 1101 ms 99.9th.
[root@vm-hercules01 /usr/lib/kafka/bin]#
```


Генерация нагрузки. Kafka-consumer-perf-test

```
root@vm-hercules01:/usr/lib/kafka/bin
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-consumer-perf-test --broker-list vm-hercules01:9092 --topic loadtestp1 --messages 500000
start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.nMsg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.nMsg.sec
2019-11-11 18:53:05:993, 2019-11-11 18:53:14:706, 485.8911, 55.7662, 500485, 57441.1798, 3034, 5679, 85.5593, 88129.0720
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-consumer-perf-test --broker-list vm-hercules01:9092 --topic loadtestp3 --messages 500000
start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.nMsg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.nMsg.sec
2019-11-11 18:53:24:548, 2019-11-11 18:53:30:022, 485.4251, 88.6783, 500005, 91341.7976, 3039, 2435, 199.3532, 205340.8624
[root@vm-hercules01 /usr/lib/kafka/bin]# ./kafka-consumer-perf-test --broker-list vm-hercules01:9092 --topic loadtestp5 --messages 500000
start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.nMsg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.nMsg.sec
2019-11-11 18:53:37:945, 2019-11-11 18:53:43:418, 485.6697, 88.7392, 500257, 91404.5313, 3053, 2420, 200.6900, 206717.7686
```

Тест разного количества партиций

Количество партиций	Скорость записи (МВ/С)	Скорость чтения (МВ/С)
1	39	86
3	86	199
5	97	201

Генерация нагрузки. Gatling

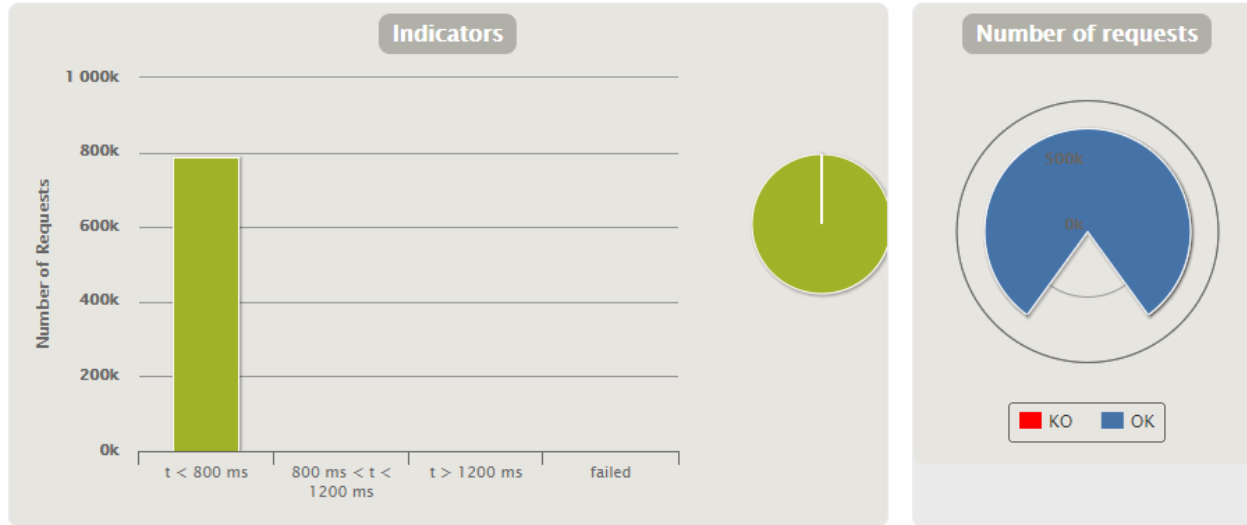
```
class BasicSimulation extends Simulation {  
  val kafkaConf: KafkaProtocol = kafka  
    .topic(name = "loadtestp5")  
    .properties(Map(  
      ProducerConfig.BOOTSTRAP_SERVERS_CONFIG -> "vm-hercules01:9092",  
      ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG -> "org.apache.kafka.common.serialization.StringSerializer",  
      ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG -> "org.apache.kafka.common.serialization.StringSerializer"))  
    )  
  
  val scn: ScenarioBuilder = scenario(scenarioName = "Basic").repeat(times = 1000) {  
    exec(kafka(requestName = "BasicRequest")  
      .send[String](payload = "foo"))  
  }  
  
  setUp(scn.inject(constantConcurrentUsers(number = 100) during (6 minutes))  
    .throttle(reachRps(target = 10000) in (120 seconds), holdFor(duration = 2 minutes), reachRps(target = 0) in (120 seconds)))  
    .protocols(kafkaConf)  
}
```

Генерация нагрузки. Gatling

```
=====
---- Global Information -----
> request count                786727 (OK=786727 KO=0    )
> min response time            34 (OK=34    KO=-    )
> max response time            334 (OK=334    KO=-    )
> mean response time           42 (OK=42    KO=-    )
> std deviation                 8 (OK=8     KO=-    )
> response time 50th percentile 39 (OK=39    KO=-    )
> response time 75th percentile 46 (OK=46    KO=-    )
> response time 95th percentile 59 (OK=59    KO=-    )
> response time 99th percentile 65 (OK=65    KO=-    )
> mean requests/sec            2191.44 (OK=2191.44 KO=-    )
---- Response Time Distribution -----
> t < 800 ms                   786727 (100%)
> 800 ms < t < 1200 ms         0 ( 0%)
> t > 1200 ms                   0 ( 0%)
> failed                         0 ( 0%)
=====
```

Генерация нагрузки. Gatling

> Global Information

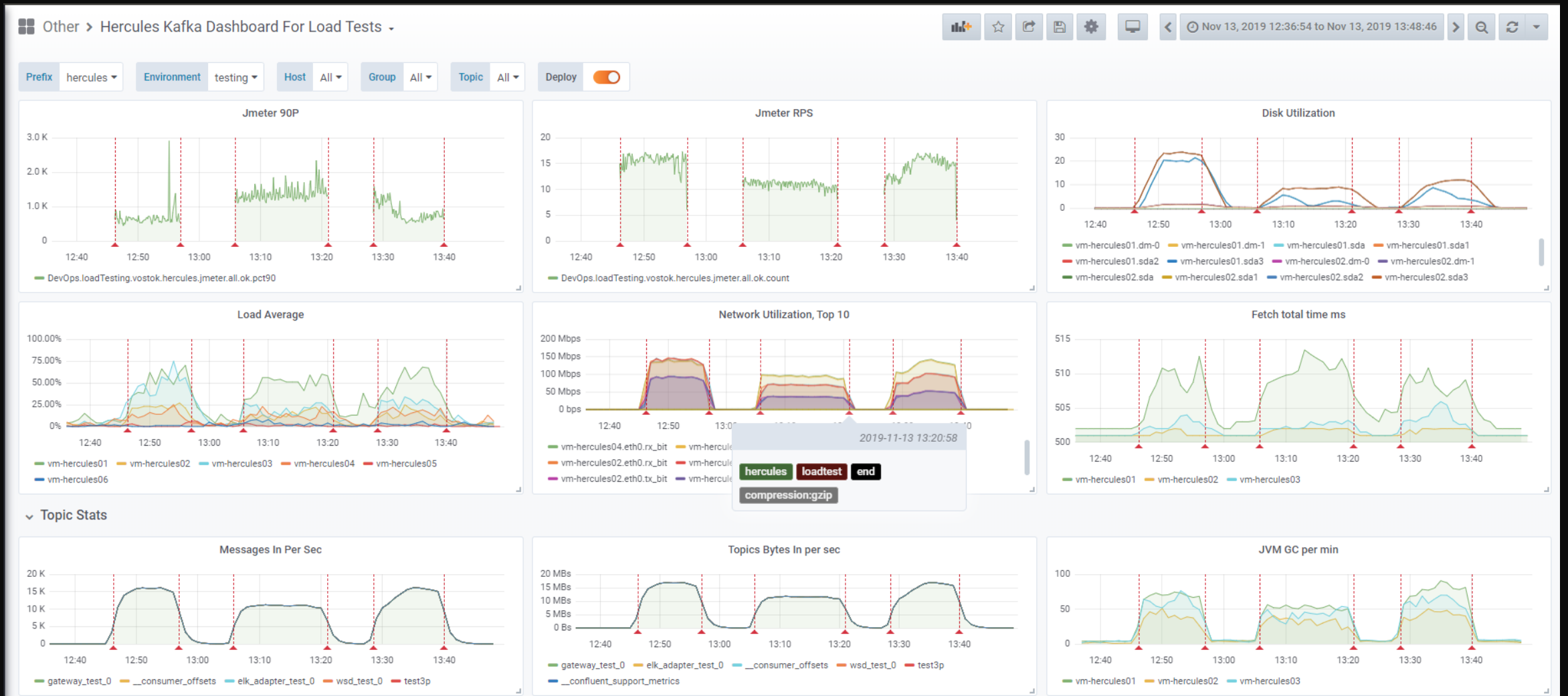


STATISTICS Expand all groups | Collapse all groups

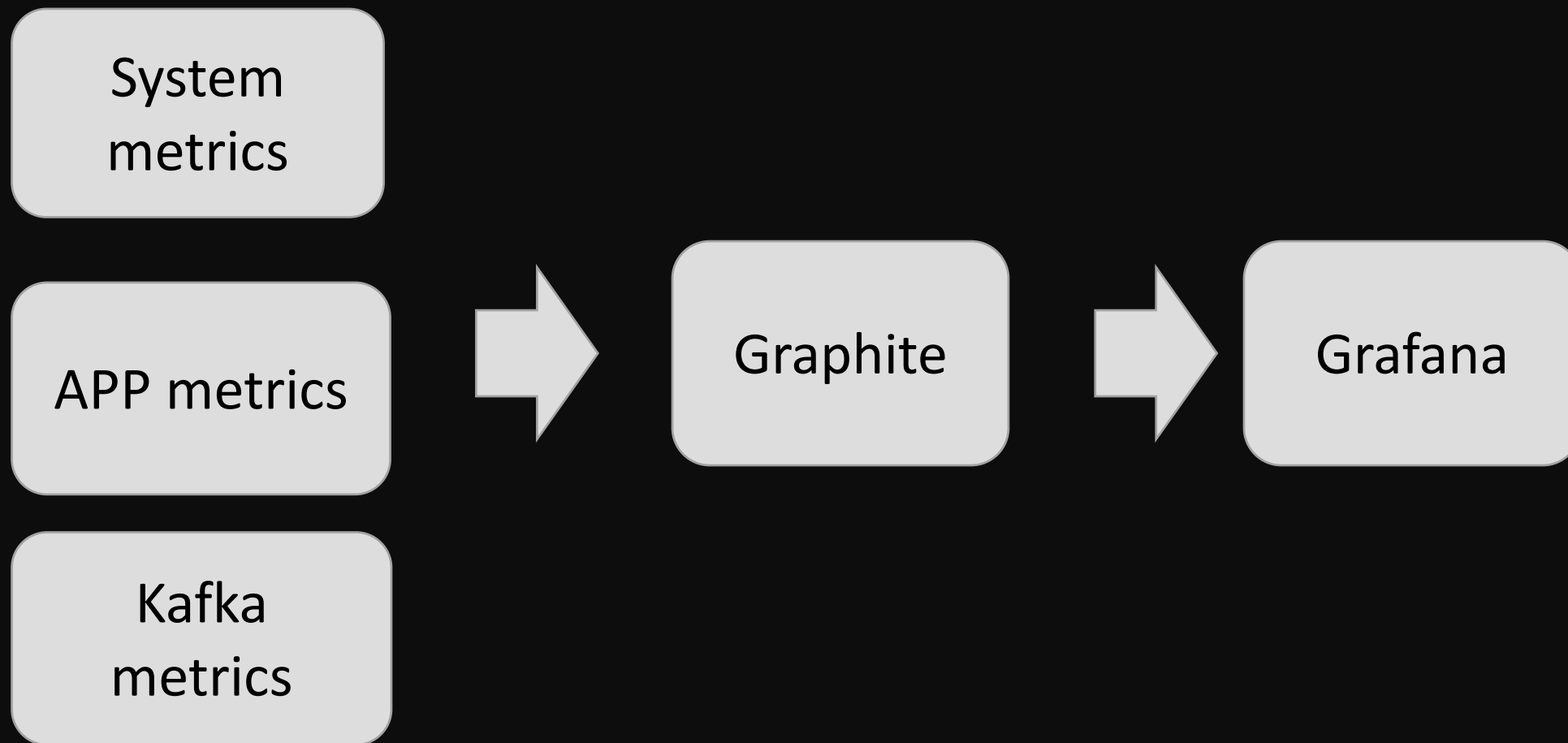
Requests ^	Executions				Response Time (ms)								
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	786727	786727	0	0%	2191.44	34	39	46	59	65	334	42	8
BasicRequest	786727	786727	0	0%	2191.44	34	39	46	59	65	334	42	8



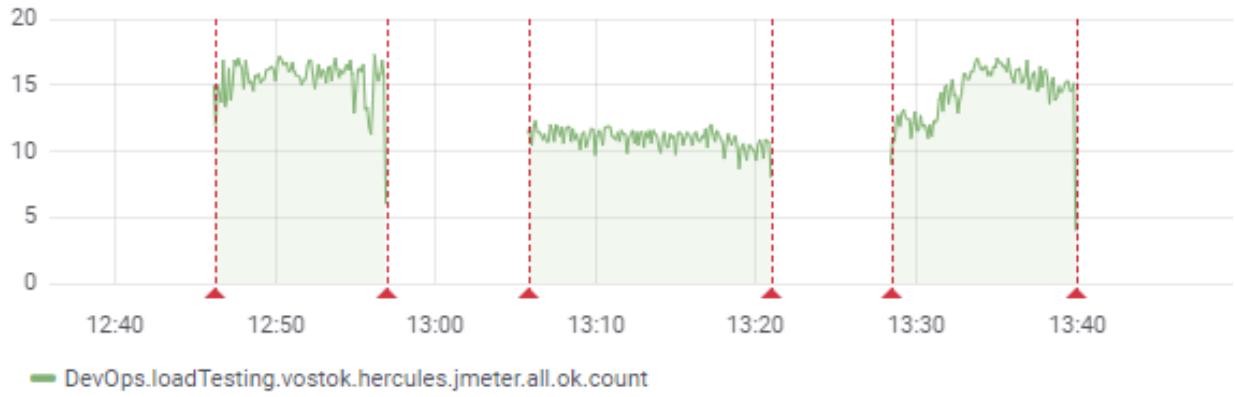
Мониторинг. Grafana



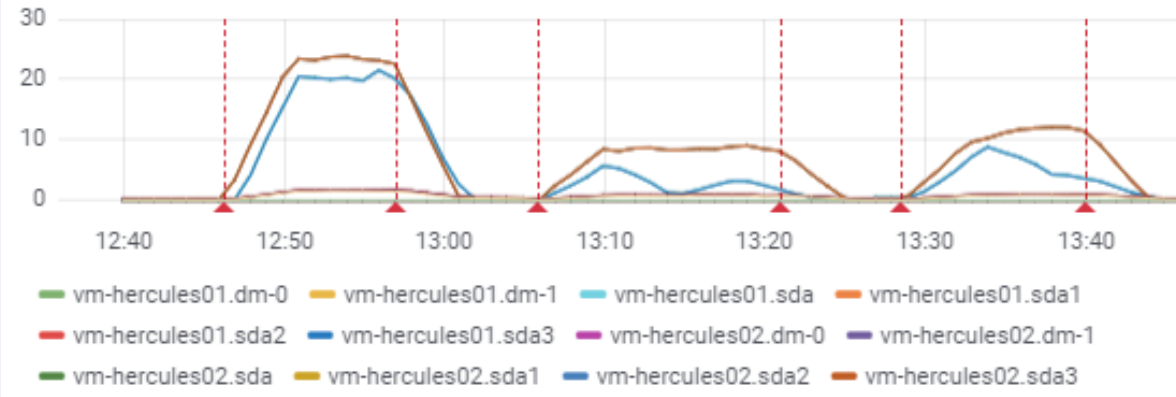
Сбор метрик



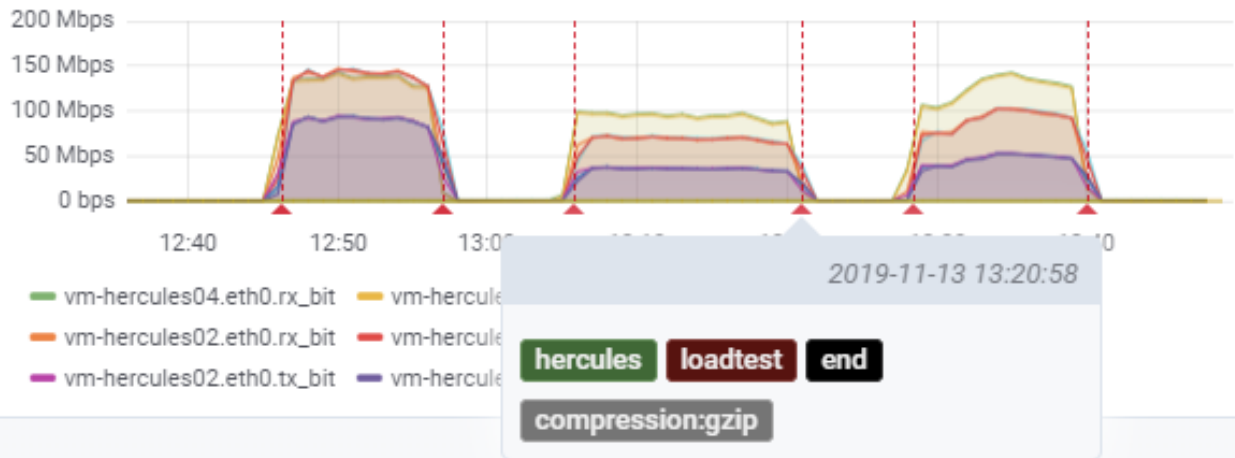
Jmeter RPS



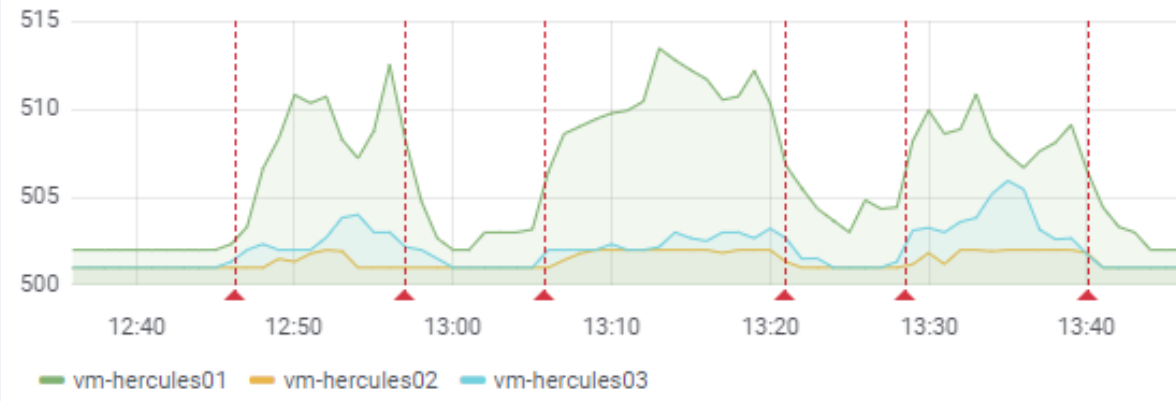
Disk Utilization



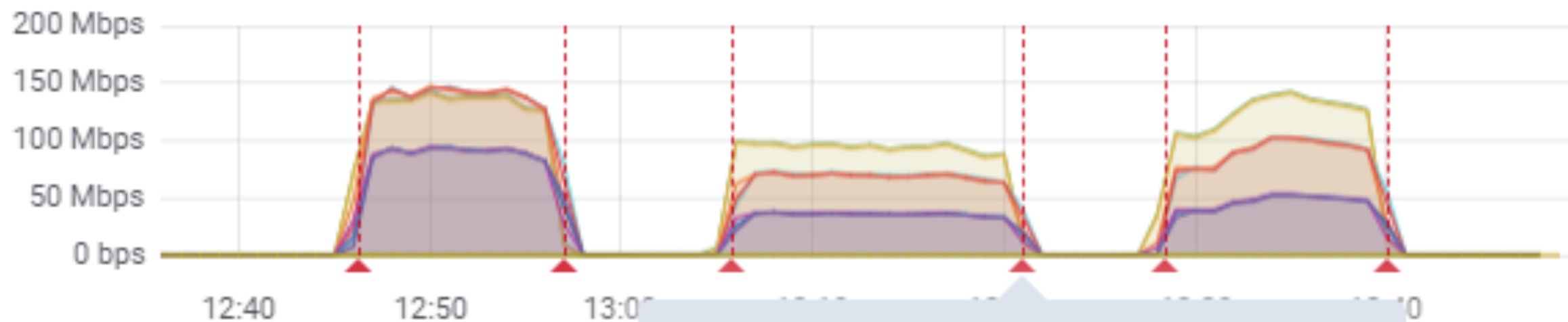
Network Utilization, Top 10



Fetch total time ms



Network Utilization, Top 10



vm-hercules04.eth0.rx_bit
vm-hercules02.eth0.rx_bit
vm-hercules02.eth0.tx_bit

vm-hercule
vm-hercule
vm-hercule

2019-11-13 13:20:58

hercules

loadtest

end

compression:gzip

Кафка. Конфигурирование

Объект

- Кластер
- Брокер
- Топик
- Producer
- Consumer

Источник

- Значение по-умолчанию
- Команда на создание
- Файл `.properties`
- Динамически

Кafka. Конфигурирование

```
[2019-11-11 18:57:45,298] INFO KafkaConfig values:
advertised.host.name = null
advertised.listeners = null
advertised.port = null
alter.config.policy.class.name = null
alter.log.dirs.replication.quota.window.num = 11
alter.log.dirs.replication.quota.window.size.seconds = 1
authorizer.class.name =
auto.create.topics.enable = false
auto.leader.rebalance.enable = true
background.threads = 10
broker.id = 1
broker.id.generation.enable = true
broker.interceptor.class = class org.apache.kafka.server.interceptor.DefaultBrokerInterceptor
broker.rack = null
client.quota.callback.class = null
compression.type = producer
connection.failed.authentication.delay.ms = 100
connections.max.idle.ms = 600000
connections.max.reauth.ms = 0
control.plane.listener.name = null
controlled.shutdown.enable = true
controlled.shutdown.max.retries = 3
controlled.shutdown.retry.backoff.ms = 5000
controller.socket.timeout.ms = 30000
create.topic.policy.class.name = null
default.replication.factor = 3
delegation.token.expiry.check.interval.ms = 3600000
delegation.token.expiry.time.ms = 86400000
delegation.token.master.key = null
delegation.token.max.lifetime.ms = 604800000
```

Кафка. Конфигурирование

- Настройки брокеров
 - server.properties
- Писателей
 - producer.properties
- Читателей
 - consumer.properties

Имени	Размера
connect-console-sink.properties	1 КБ
connect-console-source.properties	1 КБ
connect-distributed.properties	6 КБ
connect-file-sink.properties	1 КБ
connect-file-source.properties	1 КБ
connect-log4j.properties	2 КБ
connect-standalone.properties	3 КБ
consumer.properties	2 КБ
log4j.properties	5 КБ
producer.properties	2 КБ
server.properties	7 КБ
tools-log4j.properties	2 КБ
trogdor.conf	2 КБ
zookeeper.properties	1 КБ

Кafka. Конфигурирование

```
##### Log Retention Policy #####  
  
# The following configurations control the disposal of log segments. The policy can  
# be set to delete segments after a period of time, or after a given size has accumulated.  
# A segment will be deleted whenever *either* of these criteria are met. Deletion always happens  
# from the end of the log.  
  
# The minimum age of a log file to be eligible for deletion due to age  
log.retention.hours=168  
  
# A size-based retention policy for logs. Segments are pruned from the log unless the remaining  
# segments drop below log.retention.bytes. Functions independently of log.retention.hours.  
#log.retention.bytes=1073741824  
  
# The maximum size of a log segment file. When this size is reached a new log segment will be created.  
log.segment.bytes=1073741824  
  
# The interval at which log segments are checked to see if they can be deleted according  
# to the retention policies  
log.retention.check.interval.ms=300000
```

Кafka. Конфигурирование

```
##### Log Flush

# Messages are immediately written to
# the OS cache lazily. The following
# There are a few important trade-offs:
#   1. Durability: Unflushed data may
#   2. Latency: Very large flush intervals
#   3. Throughput: The flush is generally
# The settings below allow one to control
# every N messages (or both). This can

# The number of messages to accept before
#log.flush.interval.messages=10000

# The maximum amount of time a message
#log.flush.interval.ms=1000

##### Log Retention
```

Кafka. Конфигурирование

```
##### Log Flush

# Messages are immediately written to
# the OS cache lazily. The following
# There are a few important trade-offs:
#   1. Durability: Unflushed data may
#   2. Latency: Very large flush intervals
#   3. Throughput: The flush is generally
# The settings below allow one to control
# every N messages (or both). This can be
# The number of messages to accept before
#log.flush.interval.messages=10000
# The maximum amount of time a message
#log.flush.interval.ms=1000
##### Log Retention
```

```
log.cleaner.io.max.bytes.per.second = 1.7976931348624158e+31
log.cleaner.min.cleanable.ratio = 0.5
log.cleaner.min.compaction.lag.ms = 0
log.cleaner.threads = 1
log.cleanup.policy = [delete]
log.dir = /tmp/kafka-logs
log.dirs = /var/lib/kafka
log.flush.interval.messages = 9223372036854775807
log.flush.interval.ms = null
log.flush.offset.checkpoint.interval.ms = 60000
log.flush.scheduler.interval.ms = 9223372036854775807
log.flush.start.offset.checkpoint.interval.ms = 60000
log.index.interval.bytes = 4096
log.index.size.max.bytes = 10485760
log.message.downconversion.enable = true
log.message.format.version = 2.2
log.message.timestamp.difference.max.ms = 9223372036854775807
```

Кafka. Конфигурирование

log.flush.interval.messages	The number of messages accumulated on a log partition before messages are flushed to disk	long	9223372036854775807	[1,...]	high	cluster-wide
log.flush.interval.ms	The maximum time in ms that a message in any topic is kept in memory before flushed to disk. If not set, the value in log.flush.scheduler.interval.ms is used	long	null		high	cluster-wide
log.flush.offset.checkpoint.interval.ms	The frequency with which we update the persistent record of the last flush which acts as the log recovery point	int	60000	[0,...]	high	read-only
log.flush.scheduler.interval.ms	The frequency in ms that the log flusher checks whether any log needs to be flushed to disk	long	9223372036854775807		high	read-only
log.flush.start.offset.checkpoint.interval.ms	The frequency with which we update the persistent record of log start offset	int	60000	[0,...]	high	read-only

Producer, cluster, consumer, topic

```
build.sbt x BasicSimulation.scala x KafkaProducerExample.scala x logback.xml x gatling.conf x recorder.conf x
1 import java.util.Properties
2 import org.apache.kafka.clients.producer.KafkaProducer
3 import org.apache.kafka.clients.producer.ProducerRecord
4
5 object KafkaProducerExample extends App {
6   val props = new Properties()
7   props.put("bootstrap.servers", "vm-hercules01:9092")
8   props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer")
9   props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer")
10  val producer = new KafkaProducer[String, String](props)
11
12  for (x <- 1 to 100) producer.send(new ProducerRecord[String, String](topic = "test", x.toString))
13 }
14
15
16
```

Динамическое конфигурирование

```
./kafka-configs \  
--zookeeper localhost:2181 \  
--entity-type topics \  
--entity-name test \  
--alter --add-config retention.ms=3600000
```

Динамическое конфигурирование

Topic

legacy_logs_elk_c2

cleanup.policy

A string that is either "delete" or "compact" or both. This string designates the retention policy to use on old log segments. The default policy ("delete") will discard old segments when their retention time or size limit has been reached. The "compact" setting will enable [log compaction](#) on the topic.

compression.type

gzip

Specify the final compression type for a given topic. This configuration accepts the standard compression codecs ('gzip', 'snappy', 'lz4', 'zstd'). It additionally accepts 'uncompressed' which is equivalent to no compression; and 'producer' which means retain the original compression codec set by the producer.

delete.retention.ms

Конфигурация топика. Количество партиций

+ увеличение
производительности за счет
параллелизации

+ более равномерное
использование ресурсов
кластера

- много сущностей –
файловые дескрипторы,
потоки и т.п.

- долгие выборы лидера

Конфигурация. Пакетная обработка

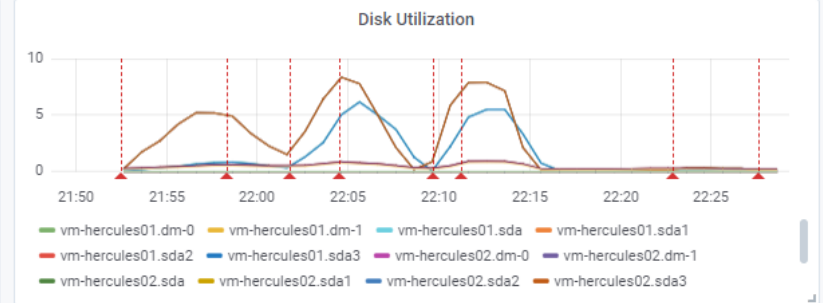
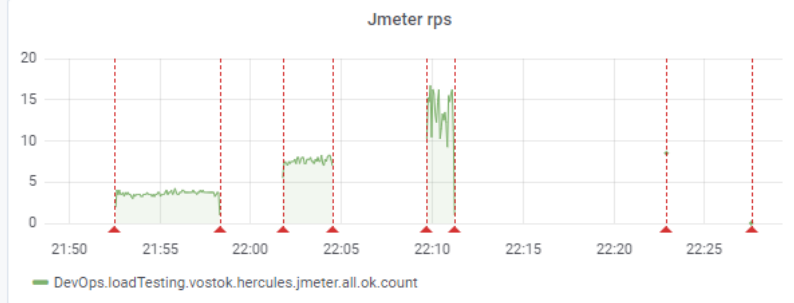
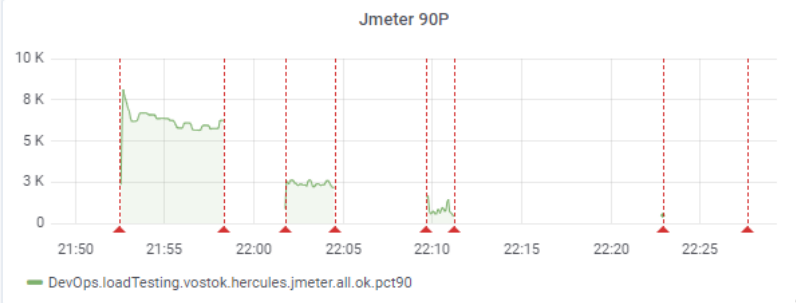
- Batch.size=16384
- + увеличение производительности за счет пакетной обработки
 - Латентность при доставке
 - Нужно выделение памяти

Конфигурация. Пакетная обработка

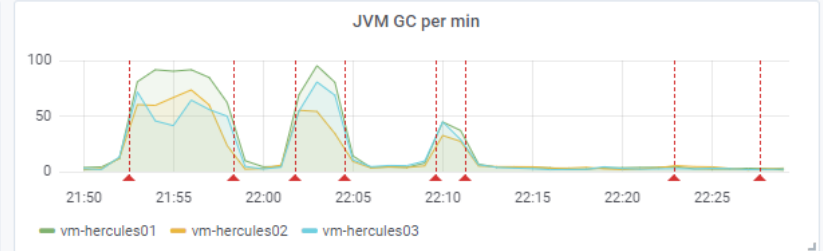
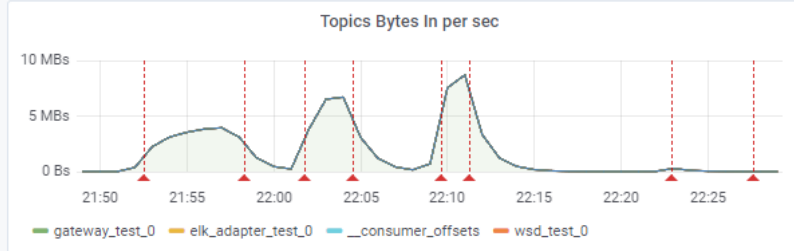
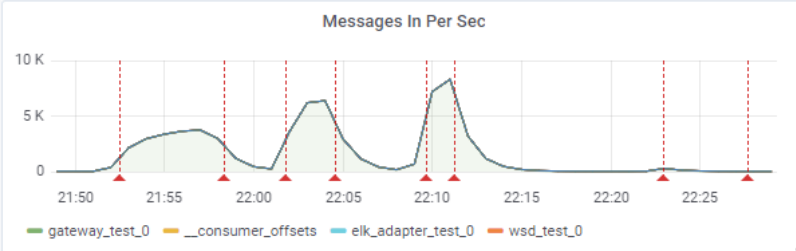
Other > Hercules Kafka Dashboard For Load Tests

Nov 13, 2019 21:49:38 to Nov 13, 2019 22:28:17

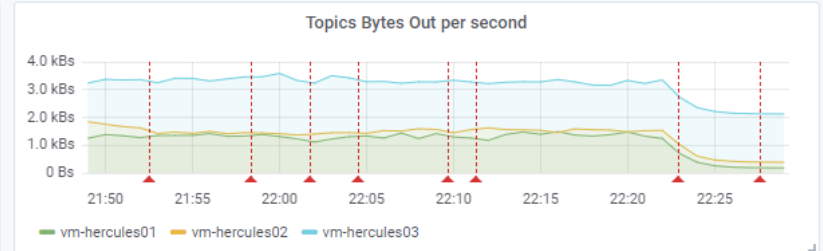
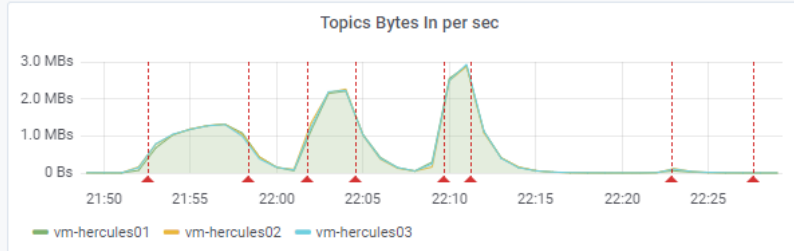
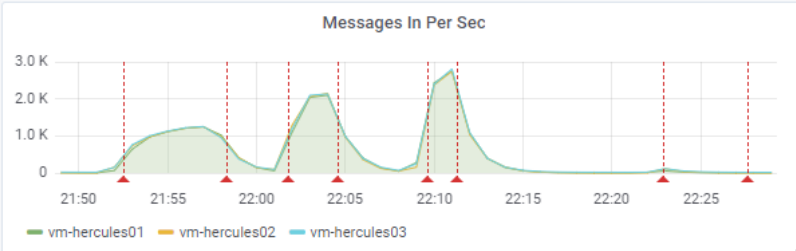
Prefix hercules Environment testing Host All Group All Topic All Deploy

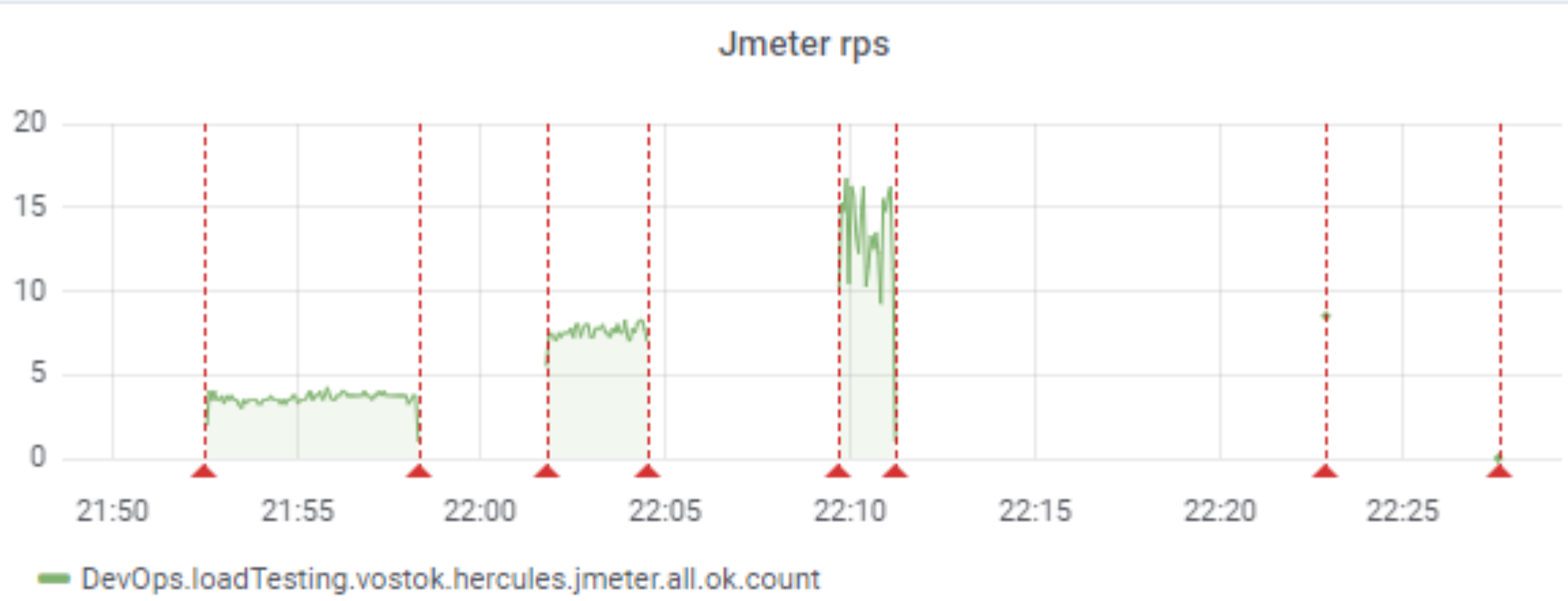


Topic Stats

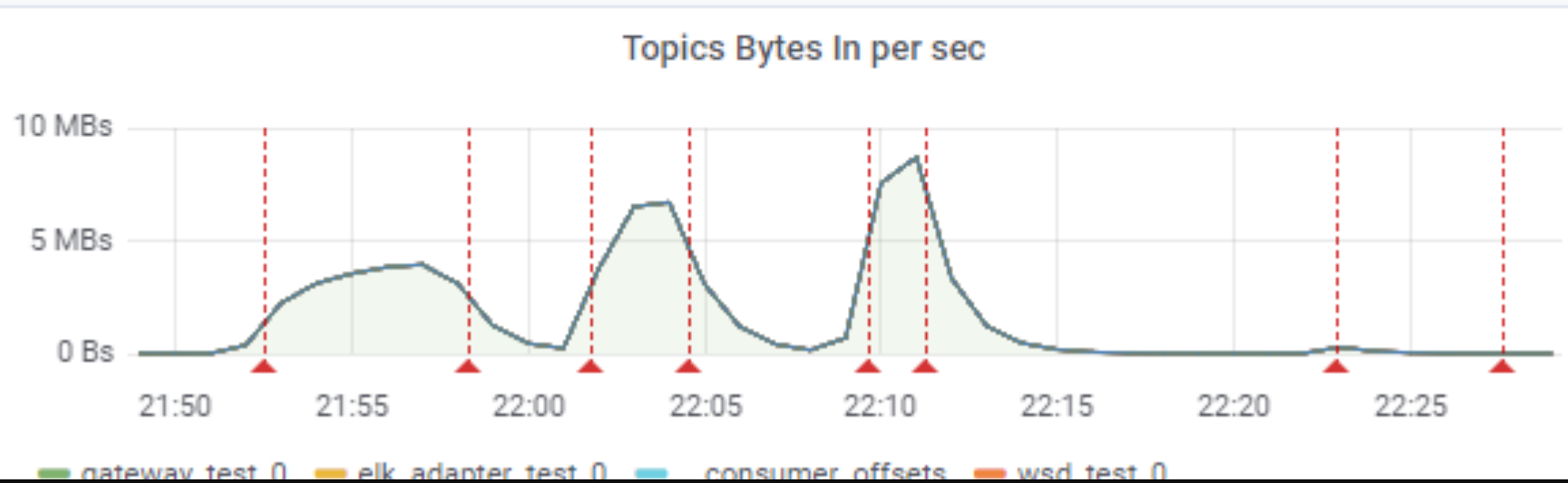


Node Stats





8192	16384	65536	1048576
------	-------	-------	---------



Конфигурация. Пакетная обработка

Other > Hercules Kafka Dashboard For Load Tests



Prefix hercules

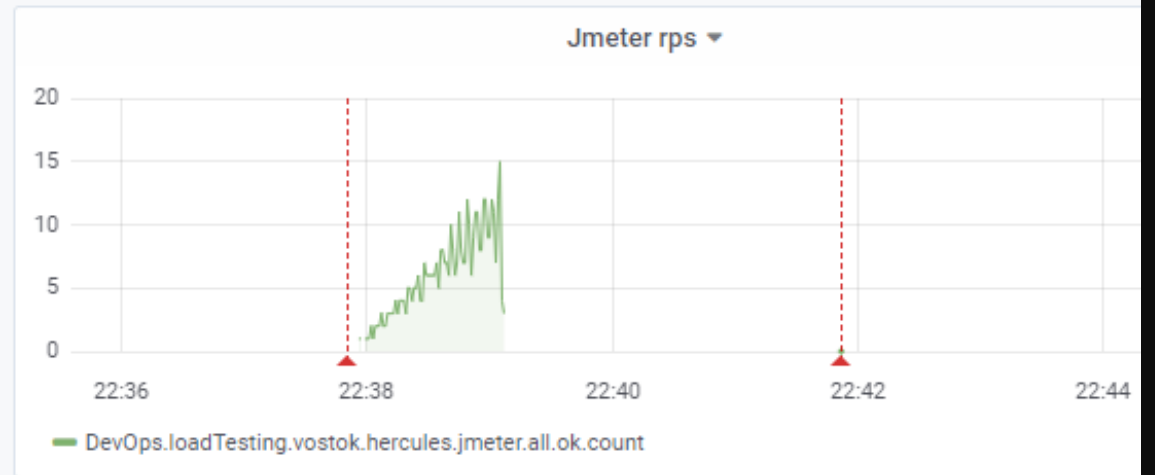
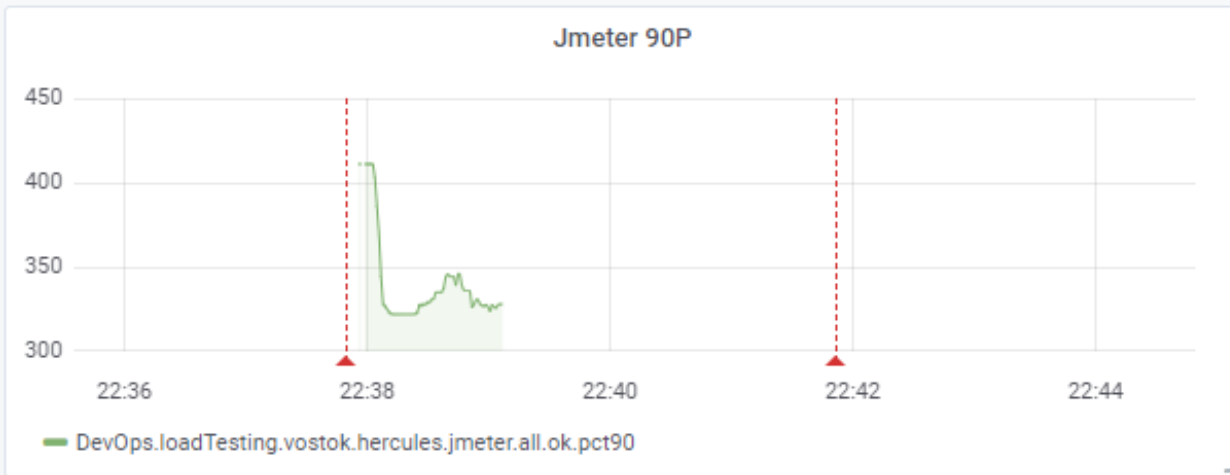
Environment testing

Host All

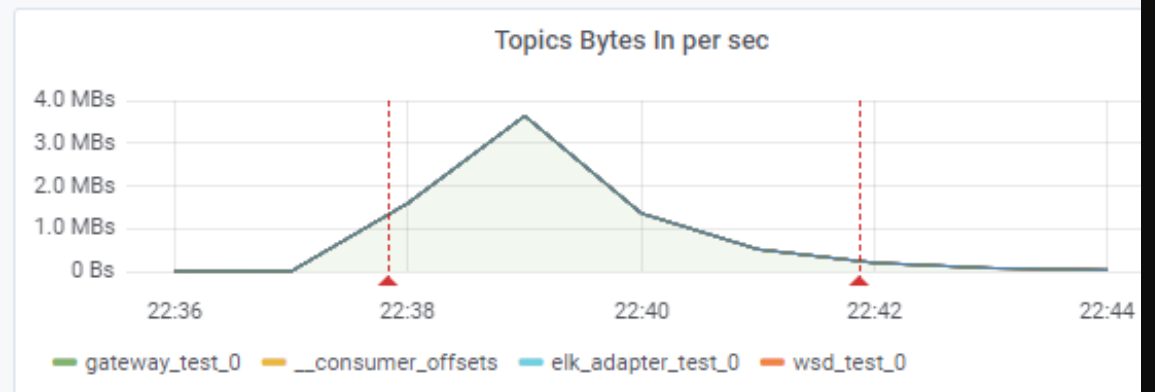
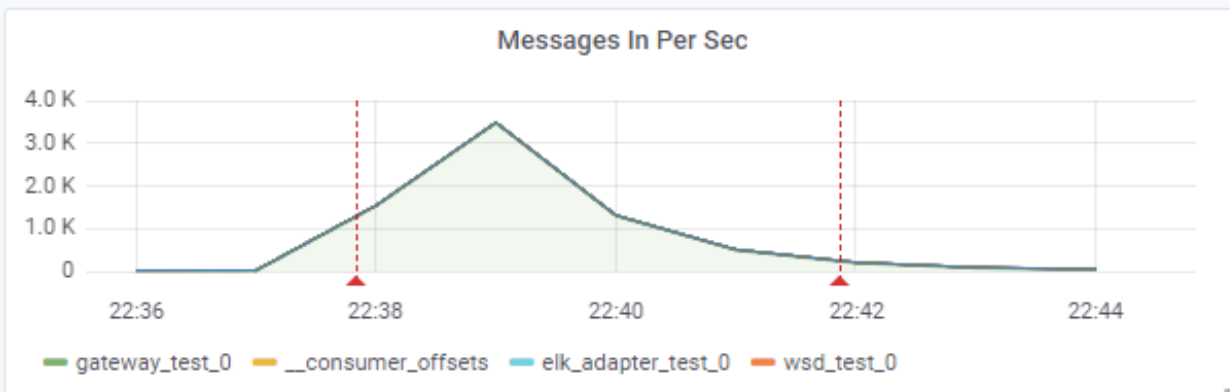
Group All

Topic All

Deploy



Topic Stats



Конфигурация. Пакетная обработка

```
ay_test_0-2, splitting and retrying (10 attempts left). Error: MESSAGE_TOO_LARGE
2019-11-13T17:23:00.198Z [kafka-producer-network-thread | producer-1] WARN o.a.k.c.producer.internals.Sender - [Producer clientId=producer-1] Got error produce response in correlation id 615 on topic-partition gateway_test_0-1, splitting and retrying (10 attempts left). Error: MESSAGE_TOO_LARGE
2019-11-13T17:23:00.209Z [kafka-producer-network-thread | producer-1] WARN o.a.k.c.producer.internals.Sender - [Producer clientId=producer-1] Got error produce response in correlation id 616 on topic-partition gateway_test_0-0, splitting and retrying (10 attempts left). Error: MESSAGE_TOO_LARGE
2019-11-13T17:23:00.218Z [kafka-producer-network-thread | producer-1] WARN o.a.k.c.producer.internals.Sender - [Producer clientId=producer-1] Got error produce response in correlation id 618 on topic-partition gateway_test_0-1, splitting and retrying (10 attempts left). Error: MESSAGE_TOO_LARGE
2019-11-13T17:23:00.449Z [kafka-producer-network-thread | producer-1] ERROR o.a.kafka.common.utils.KafkaThread - Uncaught exception in thread 'kafka-producer-network-thread | producer-1':
java.lang.OutOfMemoryError: Java heap space
```

```
2019-11-13T17:23:00.449Z [kafka-producer-network-thread | producer-1] ERROR o.a.kafka.common.utils.KafkaThread - Uncaught exception in thread 'kafka-producer-network-thread | producer-1':
java.lang.OutOfMemoryError: Java heap space
    at java.nio.HeapByteBuffer.<init>(HeapByteBuffer.java:57)
    at java.nio.ByteBuffer.allocate(ByteBuffer.java:335)
    at org.apache.kafka.clients.producer.internals.ProducerBatch.createBatchOffAccumulatorForRecord(ProducerBatch.java:296)
    at org.apache.kafka.clients.producer.internals.ProducerBatch.split(ProducerBatch.java:265)
    at org.apache.kafka.clients.producer.internals.RecordAccumulator.splitAndReenqueue(RecordAccumulator.java:345)
    at org.apache.kafka.clients.producer.internals.Sender.completeBatch(Sender.java:593)
    at org.apache.kafka.clients.producer.internals.Sender.handleProduceResponse(Sender.java:557)
    at org.apache.kafka.clients.producer.internals.Sender.access$100(Sender.java:74)
    at org.apache.kafka.clients.producer.internals.Sender$1.onComplete(Sender.java:786)
    at org.apache.kafka.clients.ClientResponse.onComplete(ClientResponse.java:109)
    at org.apache.kafka.clients.NetworkClient.completeResponses(NetworkClient.java:557)
    at org.apache.kafka.clients.NetworkClient.poll(NetworkClient.java:549)
    at org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:311)
    at org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:235)
    at java.lang.Thread.run(Thread.java:748)
```

Конфигурация. Пакетная обработка

- `Batch.size=16384`
- `max.message.bytes=1000012`
- + увеличение производительности за счет пакетной обработки
- Латентность при доставке
- Нужно выделение памяти

Конфигурация. Сжатие

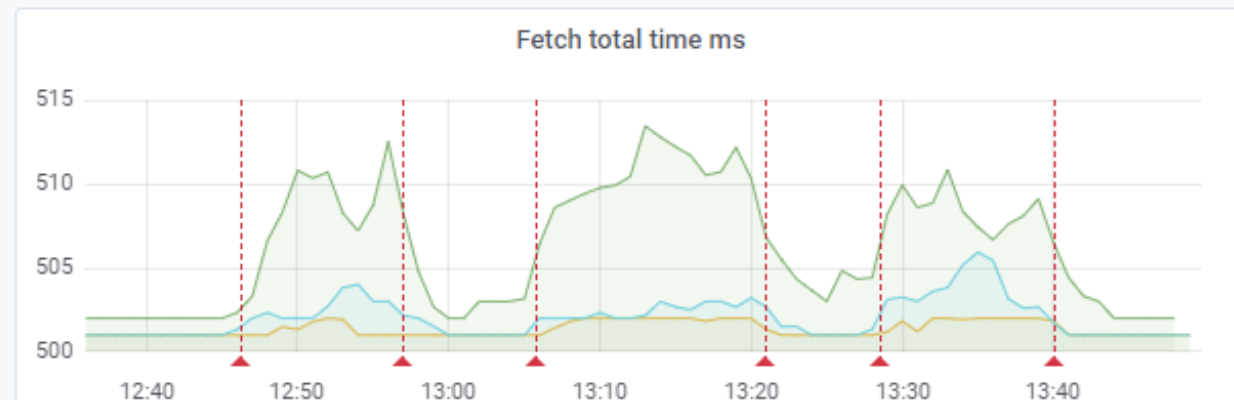
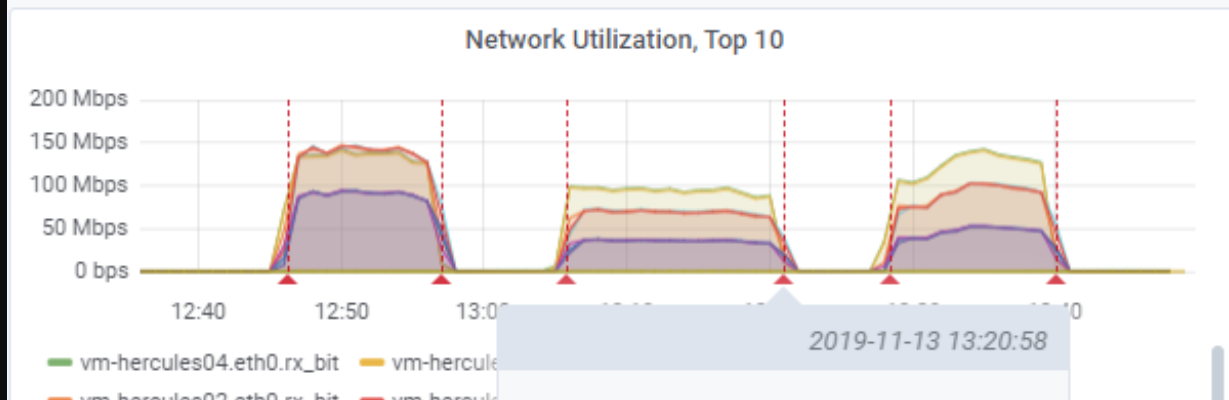
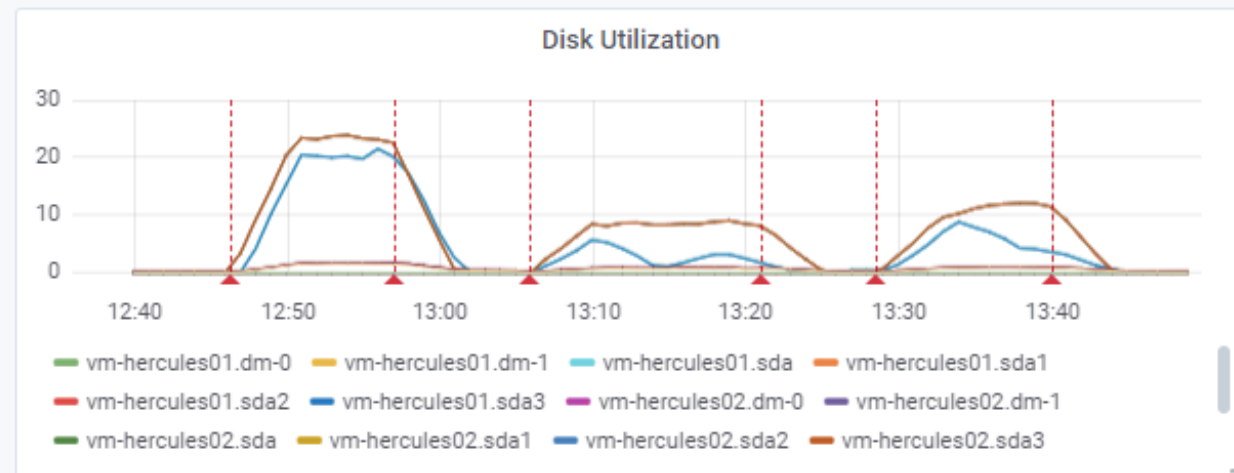
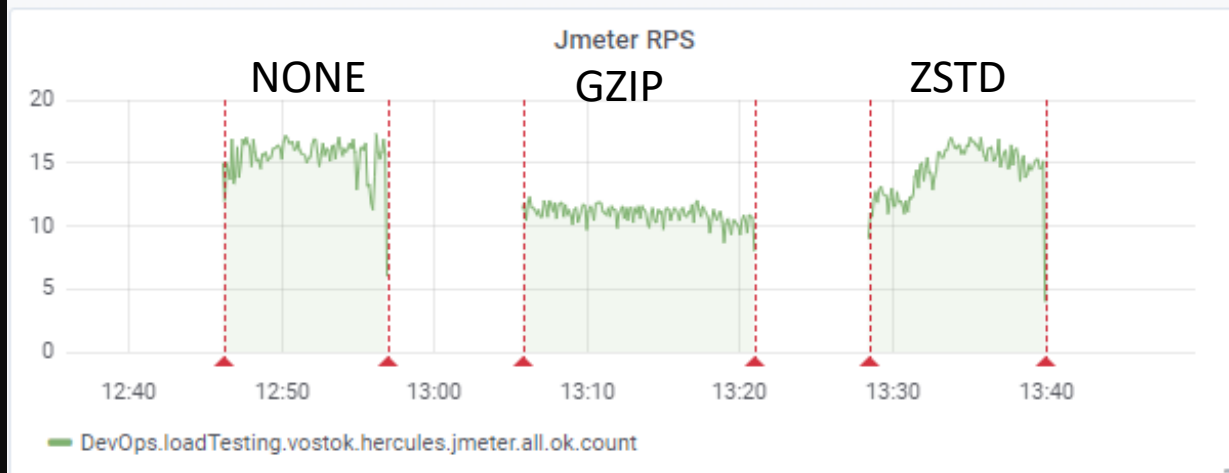
- Gzip, Snappy, lz4, zstd

- + уменьшение сетевого трафика
- + уменьшение занимаемого на диске места

-Более высокая утилизация процессора на стороне писателей и читателей

-Перехват трафика или чтение файла топика не покажет актуальные записи

Конфигурация. Сжатие



Конфигурация. Threads, buffers

- num.network.threads
- num.io.threads
- num.replica.fetchers
- num.recovery.threads.per.data.dir
- log.cleaner.threads
- background.threads
- send.buffer.bytes
- receive.buffer.bytes
- socket.send.buffer.bytes
- socket.receive.buffer.bytes
- buffer.memory
- replica.socket.receive.buffer.bytes

Конфигурация. Threads, buffers

- + Увеличение производительности
- + Более полное использование ресурсов

- Сложно воспроизводить, нельзя экстраполировать
- Непредсказуемое поведение при неправильных настройках

Порядок действий

- Определяем параметры для изменения
- Читаем по ним документацию
- Запускаем тест с текущими значениями
- Запускаем тест с новыми значениями
- Сравниваем результаты
- Делаем выводы

Пример отчета



Цыпаев Владимир Николаевич • commented 25/06/19 12:24

Было проведено тестирование чтения записей из кафки через кафка-консьюмеры.

Номера стримов:

- 1 - стрим состоящий из 48 партиций, все данные пишутся в одну из 48 партиций;
- 2 - стрим состоящий из 48 партиций, все данные пишутся равномерно по всем партициям;
- 3 - стрим состоящий из 1 партиции, все данные пишутся в одну партицию.

Номер стрима	Значение <code>max.poll.records</code>	Количество прочитанных мб/с.
1	1_000	16
1	10_000	104
2	1_000	220
2	10_000	200
3	1_000	102
3	10_000	103

При дальнейшем увеличении параметра `max.poll.records`, значительное увеличение количества прочитанных из кафки мб/с. не наблюдается.

Увеличение количества прочитанных мб/с. в стриме номер 1 происходит резким скачком начиная со значения примерно равного 9950.

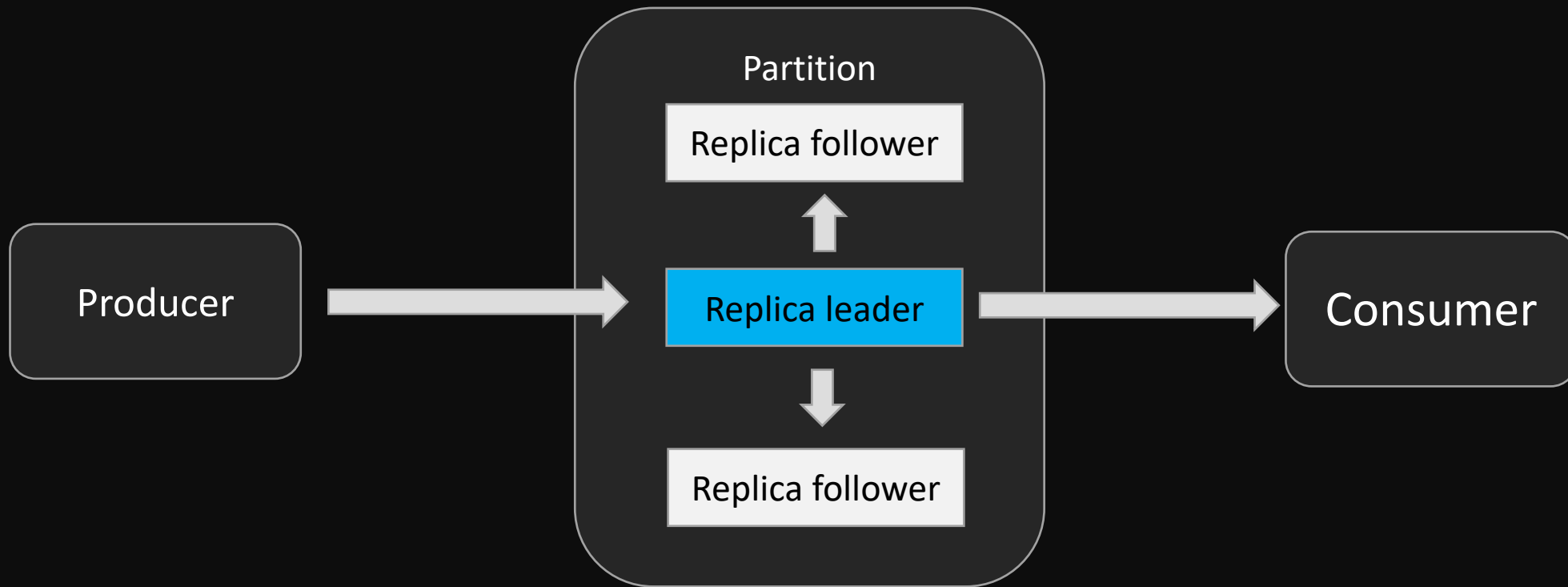
Тестирование в эксплуатации

- Отказоустойчивость
- Возможность обслуживания
- Гарантии доставки данных

Виды операций/инцидентов

- Выпадение ноды кафки из кластера
- Запланированное выключение ноды
- Рост лага при отставании записи в целевую систему
- Сдвиг офсета для консьюмера
- Перераспределение партиций

Репликация

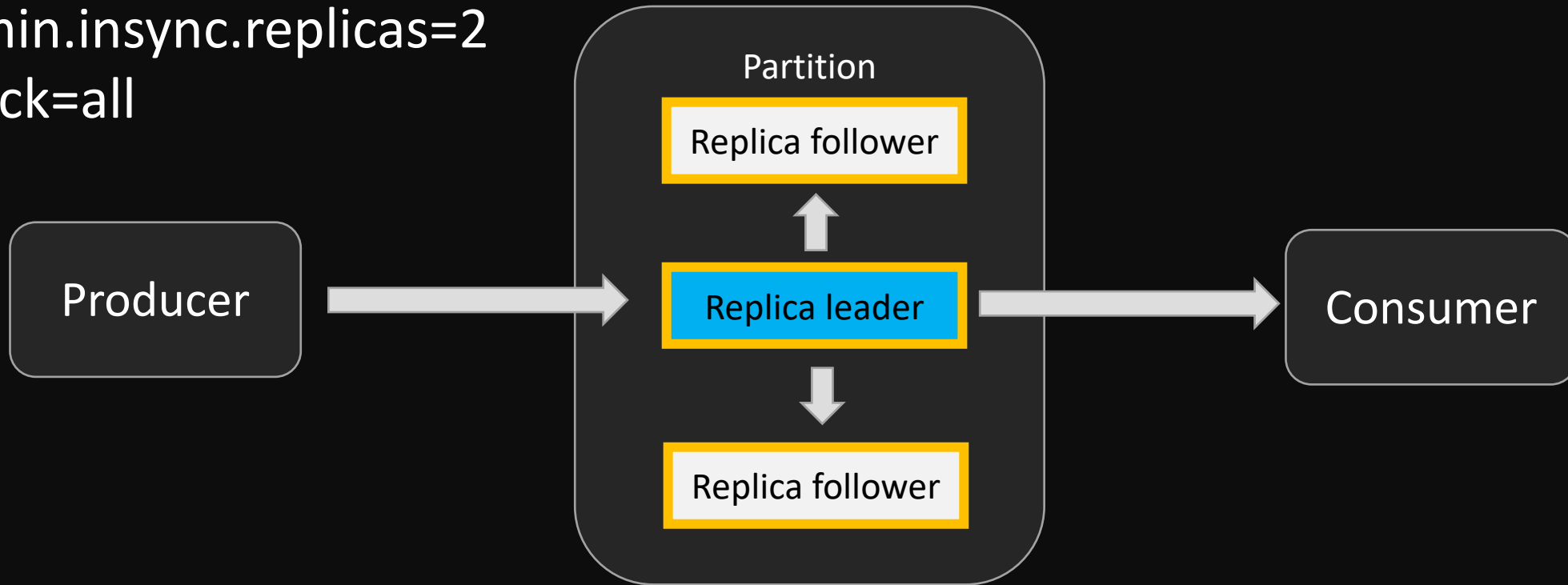


Репликация

rf=3

min.insync.replicas=2

ack=all

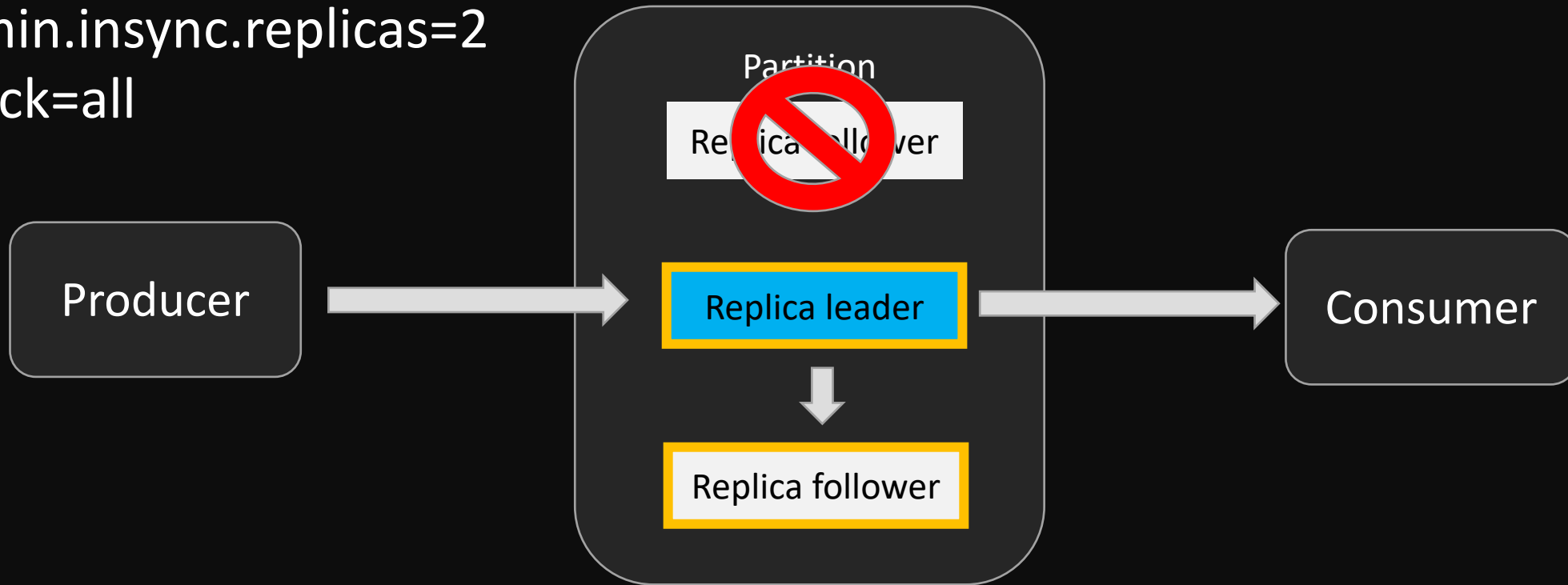


Репликация

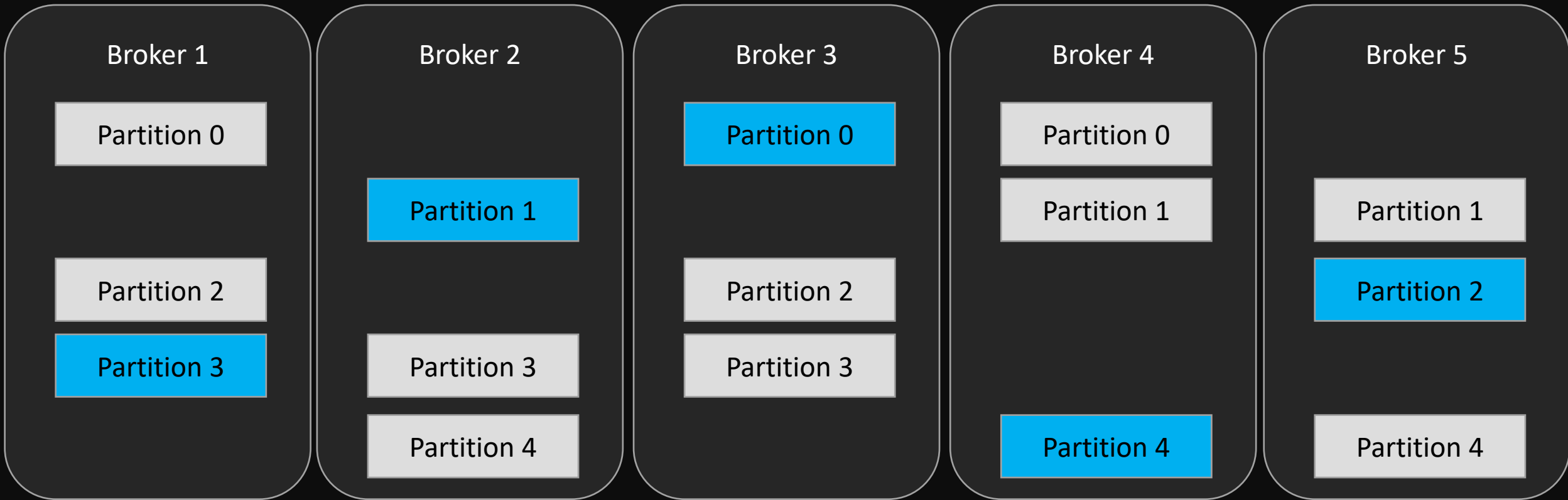
rf=3

min.insync.replicas=2

ack=all



Топик, партиция, реплика. p5r3



Исключение брокера

Broker 1

Partition 0

Partition 2

Partition 3

Broker 2

Partition 1

Partition 3

Partition 4

Broker 3

Partition 0

Partition 2

Partition 3

Broker 4

Partition 0

Partition 1

Partition 4

Broker 5

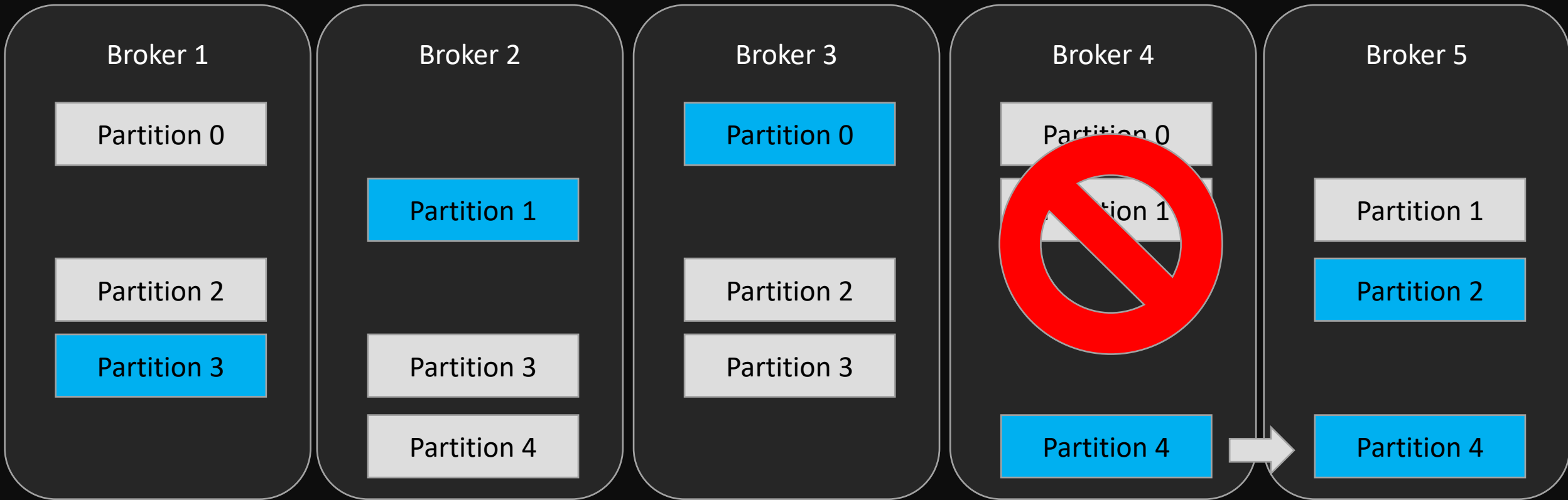
Partition 1

Partition 2

Partition 4



Исключение брокера



Топик, партиция, реплика

Partition	Leader	Replicas	ISR
0	3	3,1,4	3,1,4
1	2	2,4,5	2,4,5
2	5	5,3,1	5,3,1
3	1	1,3,2	1,3,2
4	4	4,5,2	4,5,2

Топик, партиция, реплика

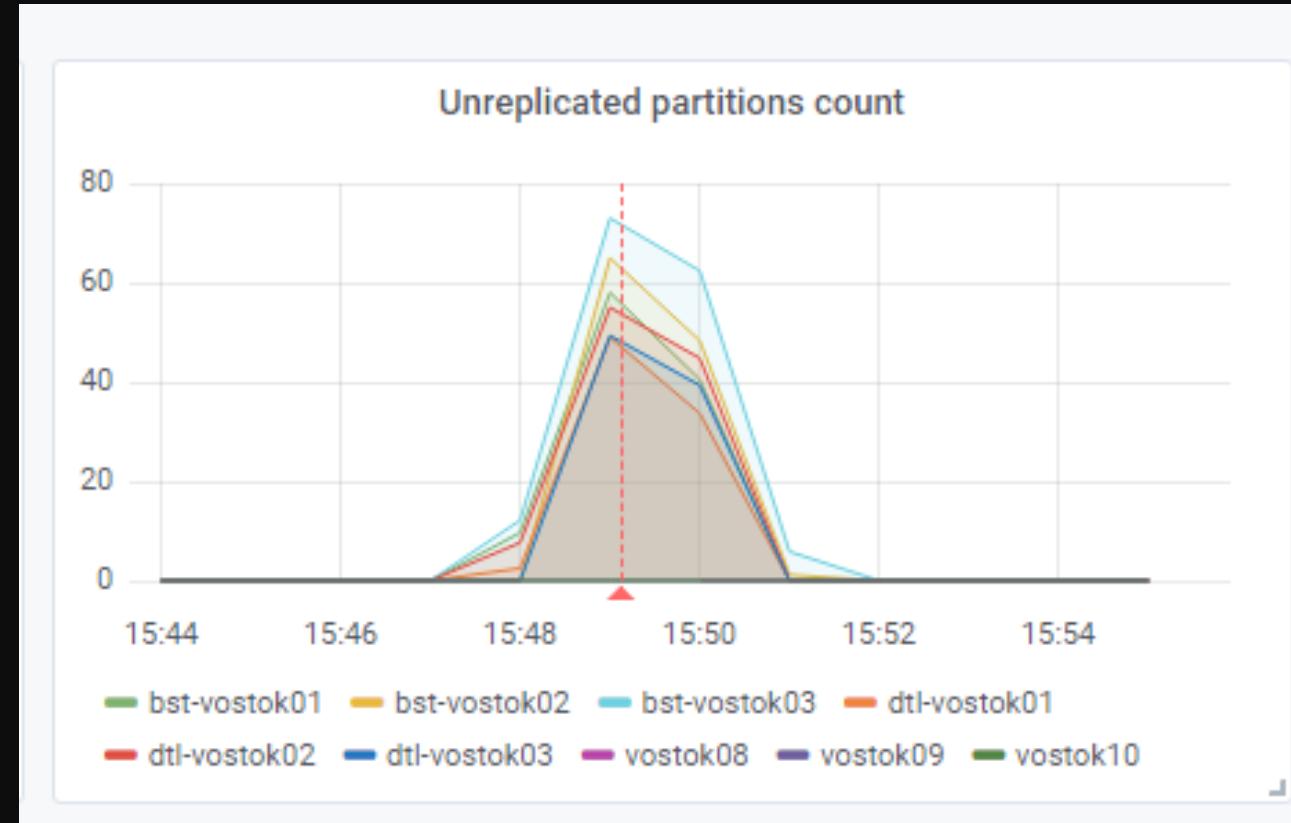
Partition	Leader	Replicas	ISR
0	3	3,1,4	3,1,4
1	2	2,4,5	2,4,5
2	5	5,3,1	5,3,1
3	1	1,3,2	1,3,2
4	4	4,5,2	4,5,2

Топик, партиция, реплика

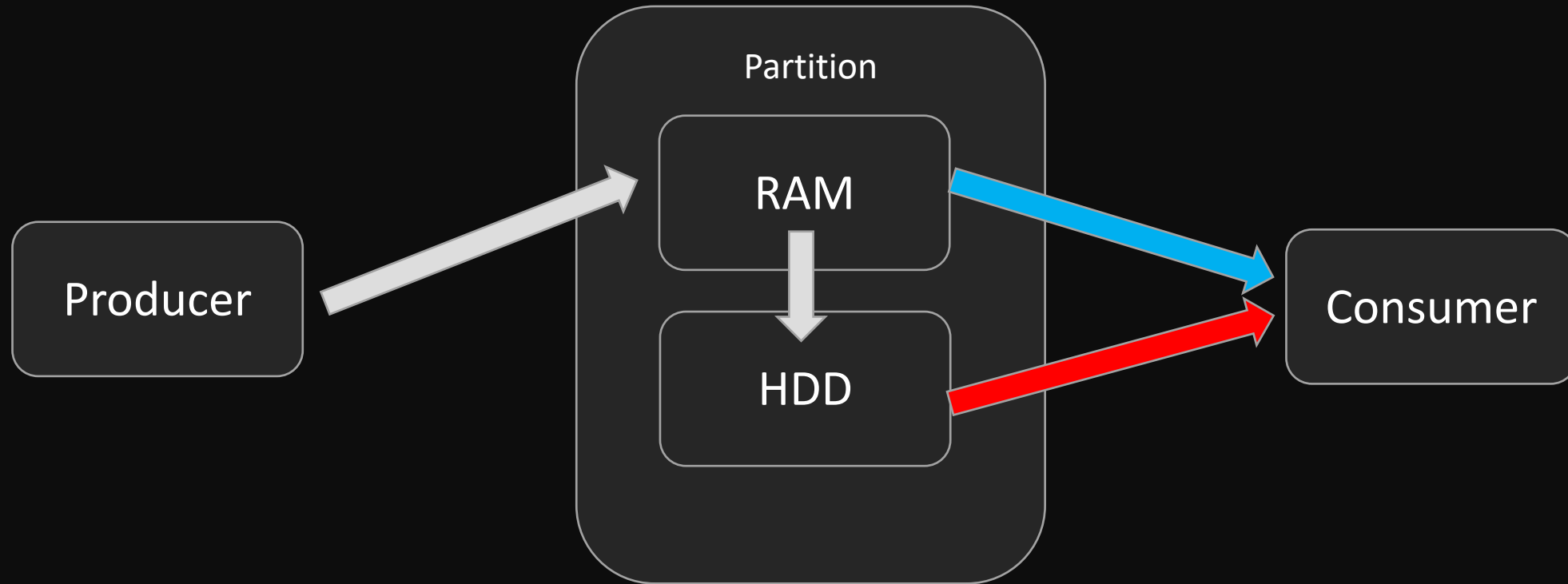
Partition	Leader	Replicas	ISR
0	3	3,1,4	3,1
1	2	2,4,5	2,5
2	5	5,3,1	5,3,1
3	1	1,3,2	1,3,2
4	5	4,5,2	5,2

Перевыборы лидера. Параметры продюсера

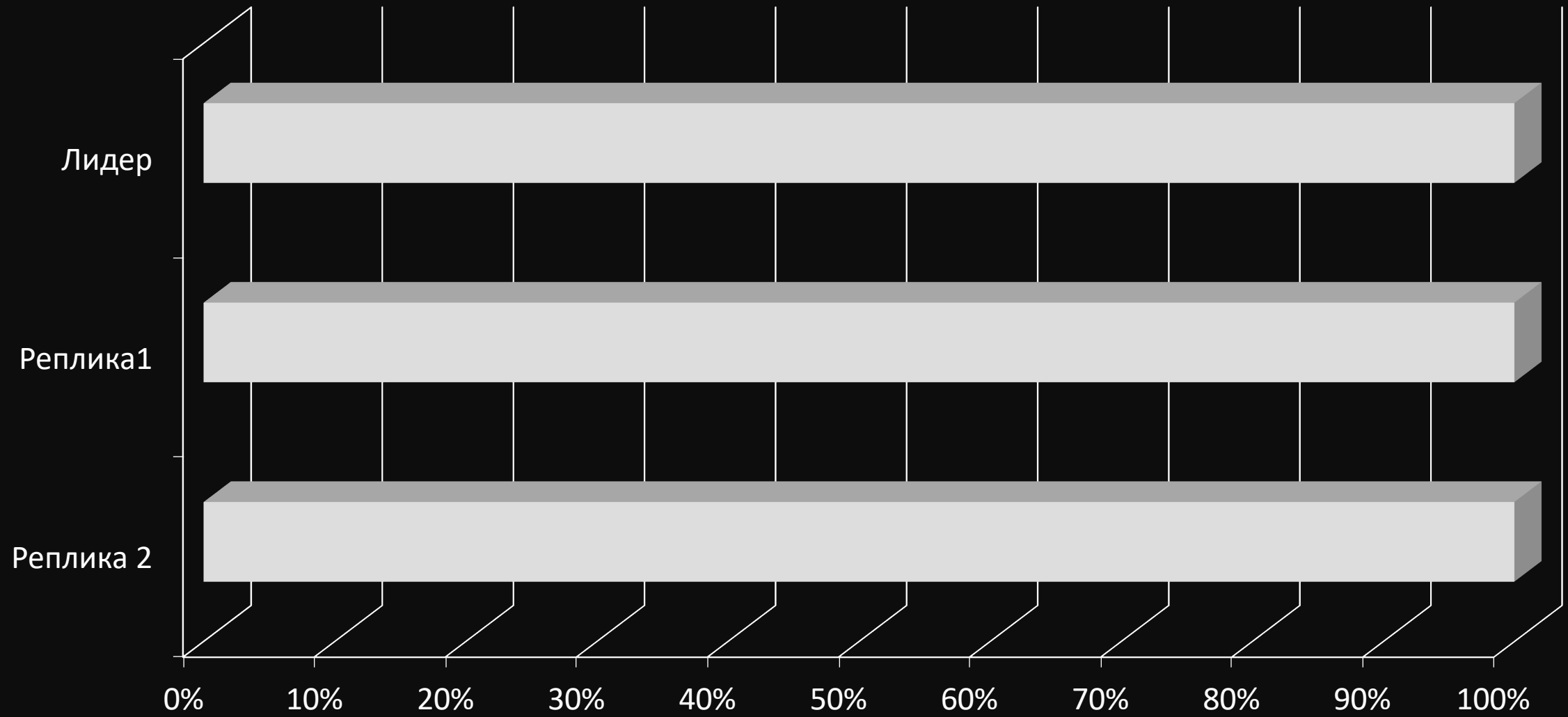
- `rf=3`
- `min.insync.replicas=2`
- `producer.acks=all`
- `producer.retries=10`
- `producer.retry.backoff.ms=250`



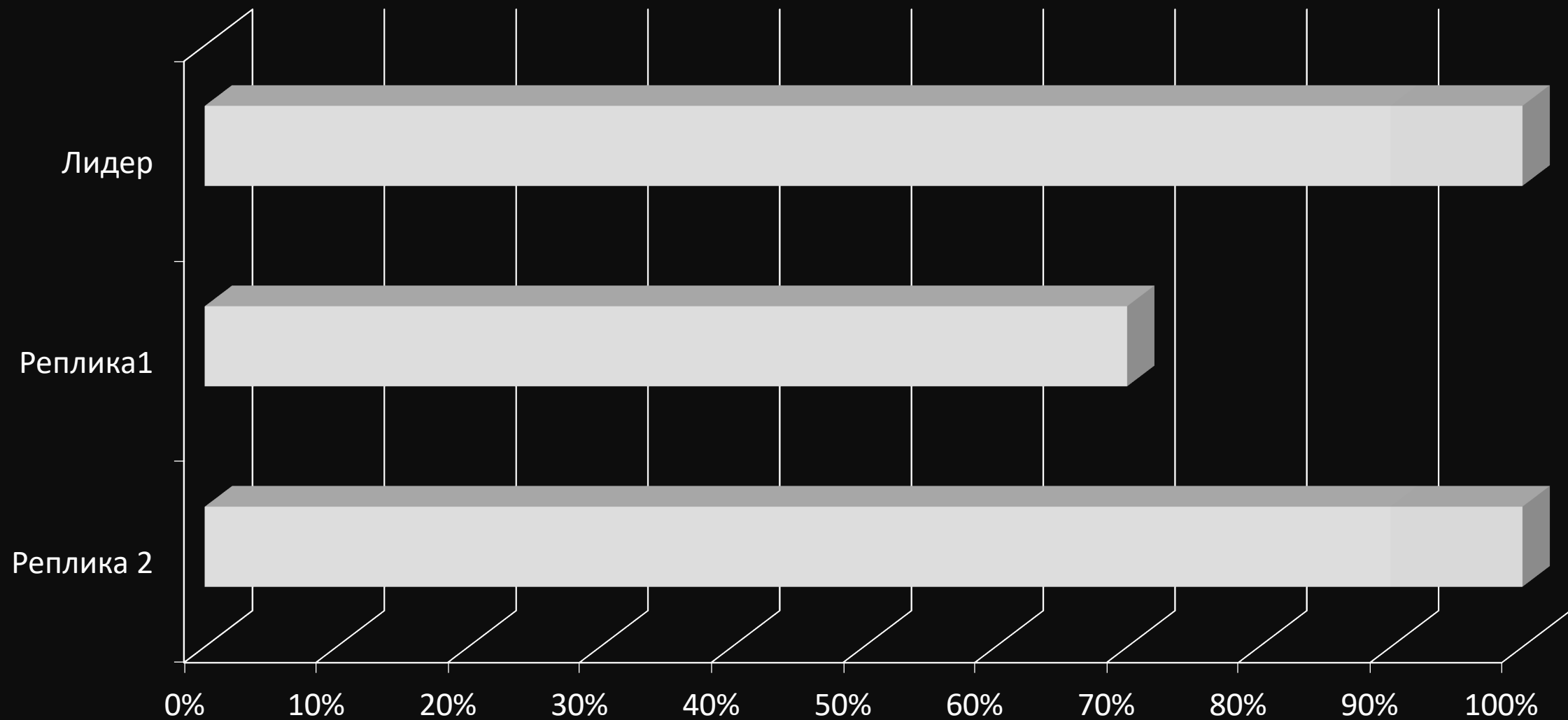
Page cache



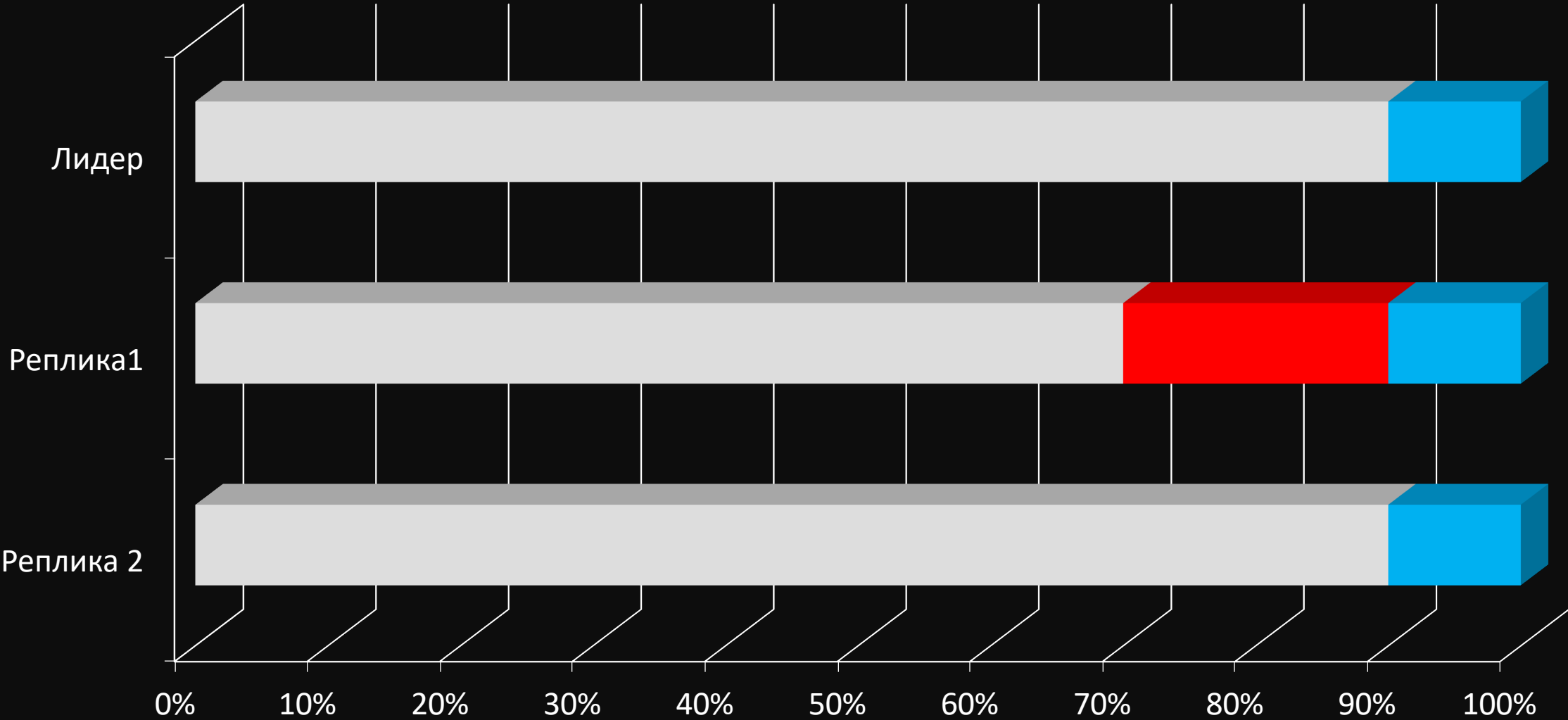
Репликация



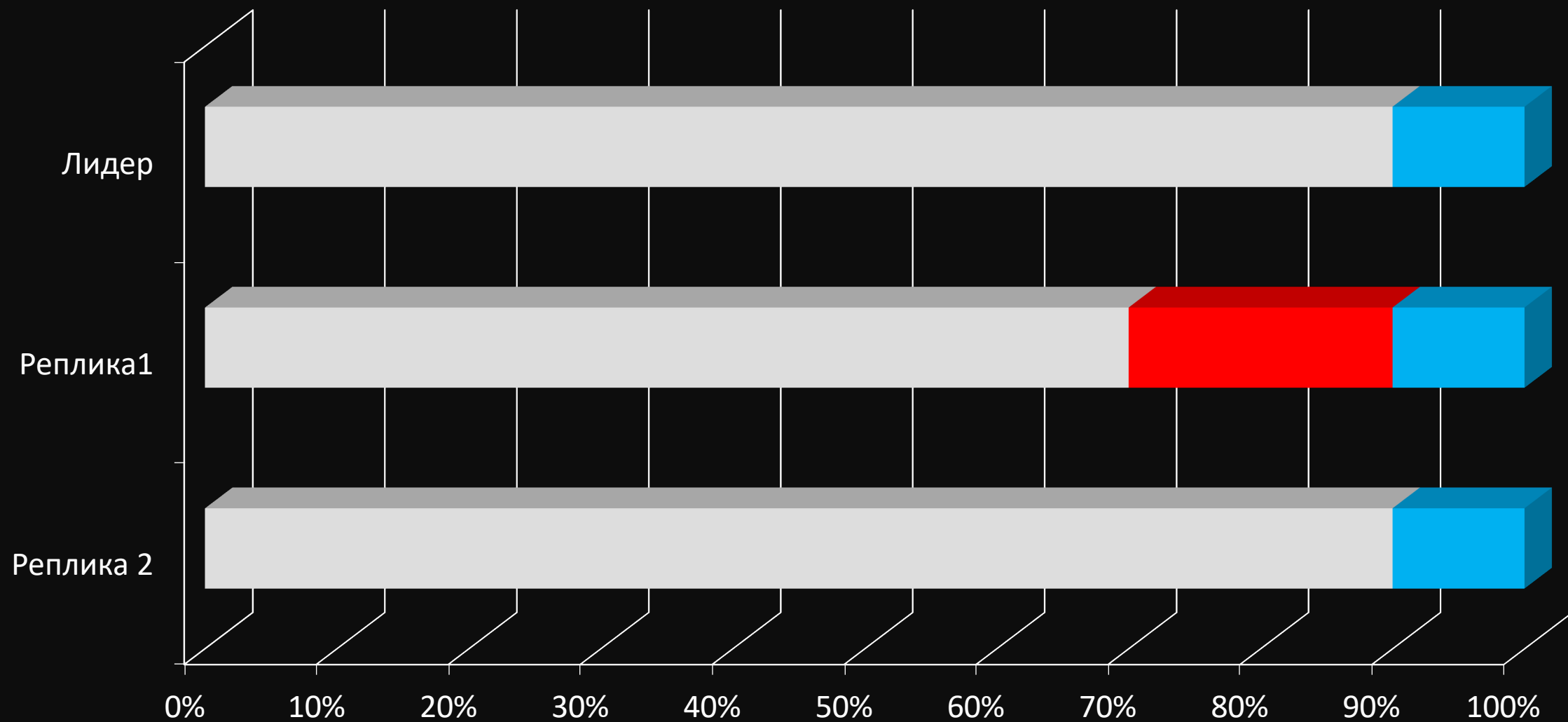
Репликация. Page cache



Репликация. Page cache



Репликация. Page cache



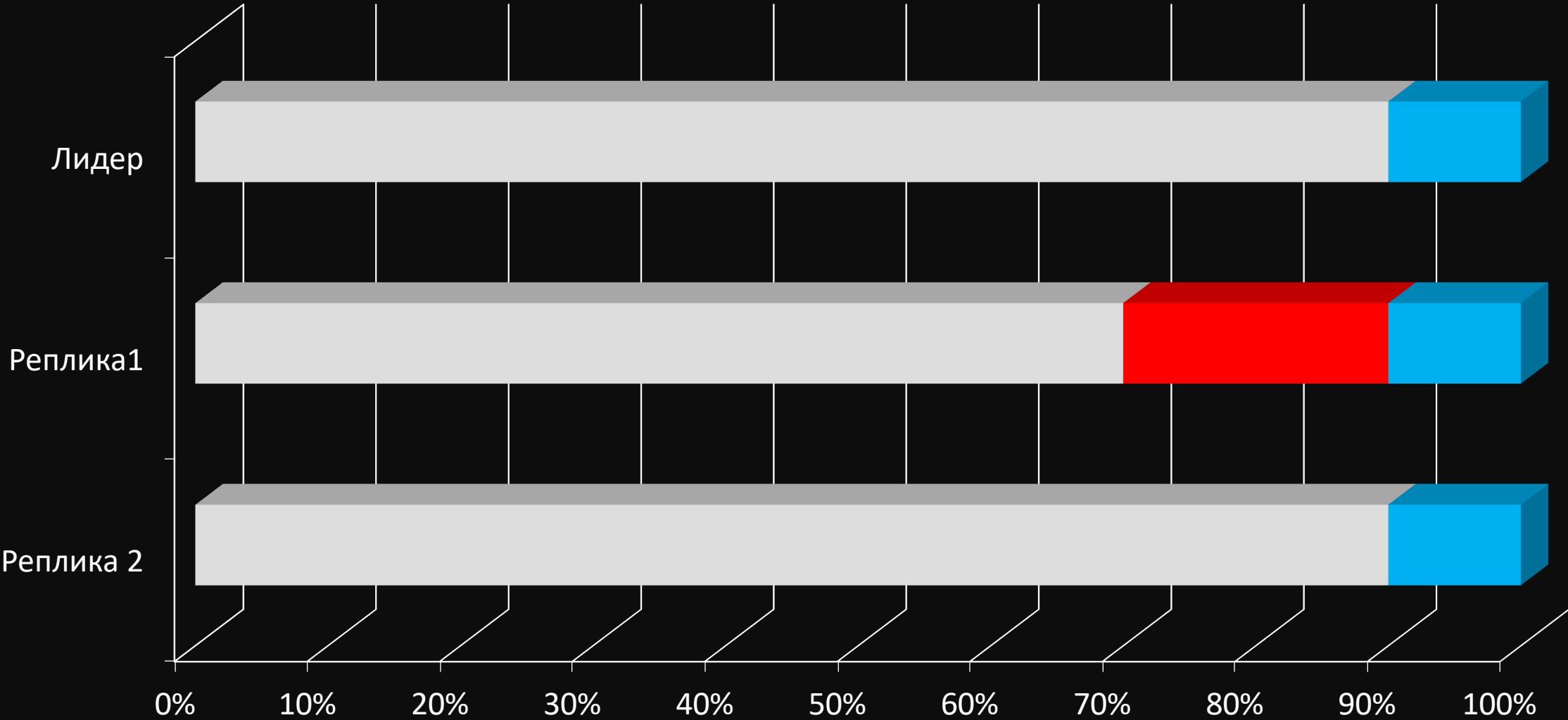
Перезагрузка ноды. Page cache

- 200 MB/S – скорость записи в брокер
- 120 GB – объем оперативной памяти брокера под кеш
- Время в кеше = $120000/200 = 600$ секунд

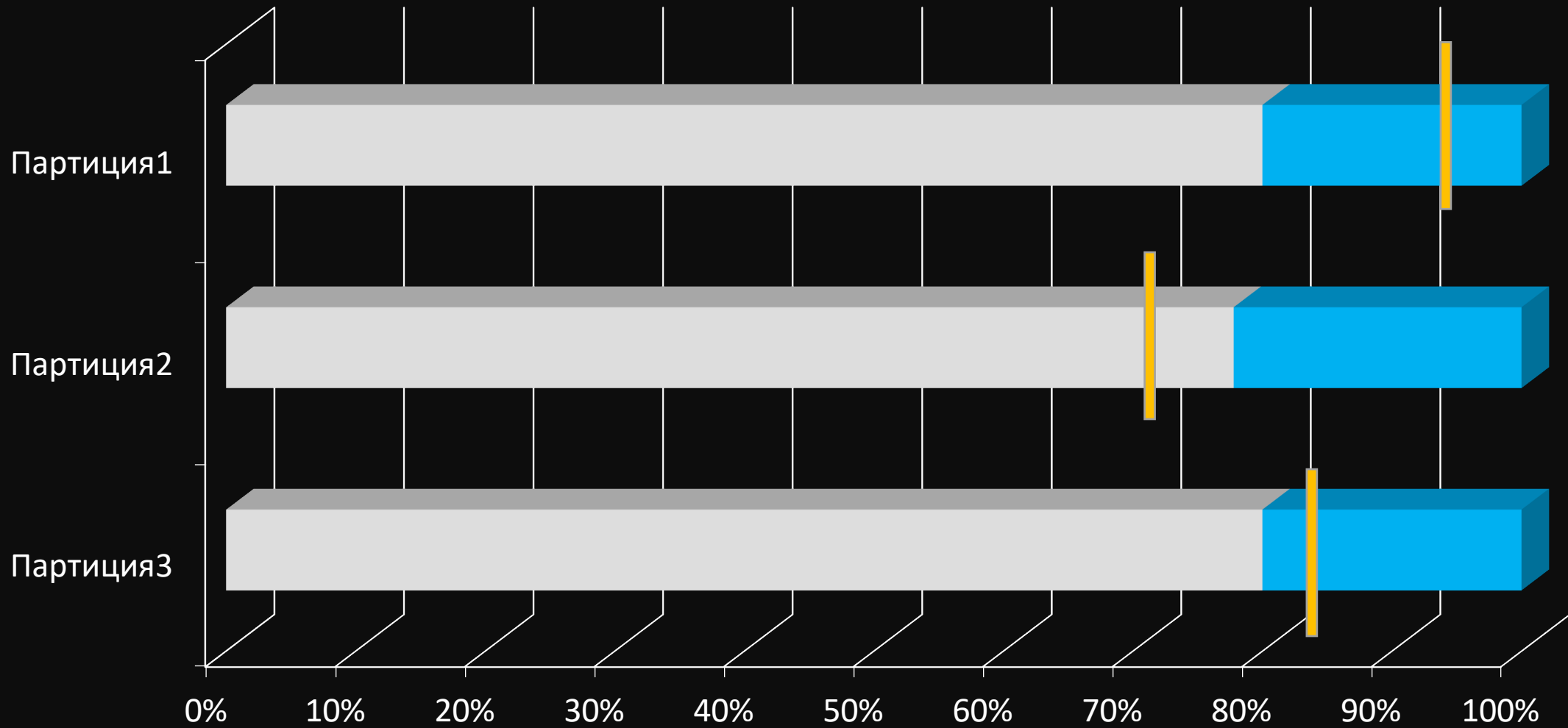
Перезагрузка ноды

- Понимать, как система ведет себя при этой операции
- Правильно оценивать время восстановления
 - Выборы лидеров
 - Индексация файлов
 - Репликация
- Не забывать про кеш

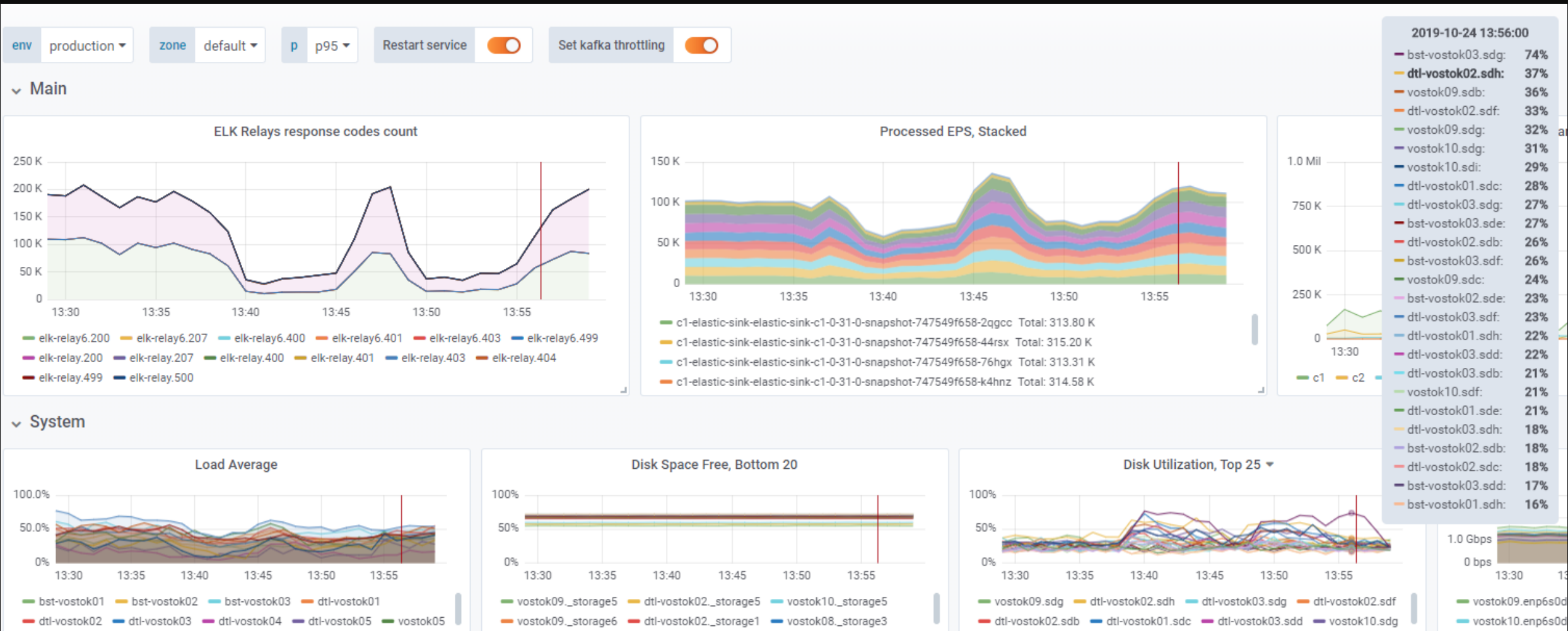
Репликация. Page cache



Лаг чтения. Сдвиг офсета



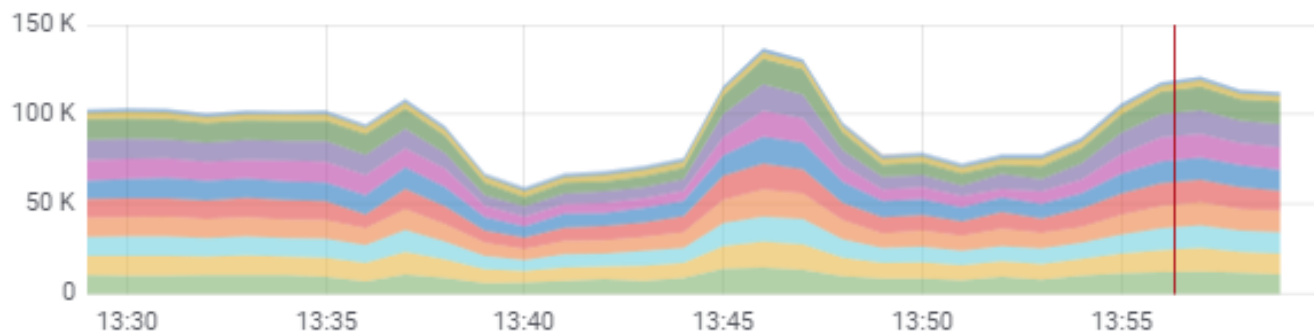
Лаг чтения



rottling



Processed EPS, Stacked



- c1-elastic-sink-elastic-sink-c1-0-31-0-snapshot-747549f658-2qgcc Total: 313.80 K
- c1-elastic-sink-elastic-sink-c1-0-31-0-snapshot-747549f658-44rsx Total: 315.20 K
- c1-elastic-sink-elastic-sink-c1-0-31-0-snapshot-747549f658-76hgx Total: 313.31 K
- c1-elastic-sink-elastic-sink-c1-0-31-0-snapshot-747549f658-k4hnz Total: 314.58 K

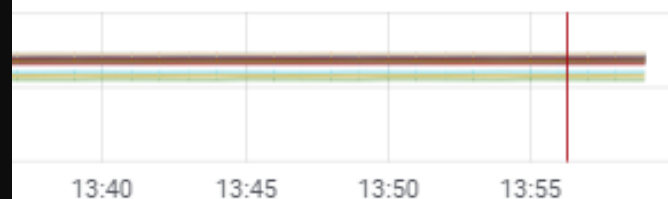


c1 c2

2019-10-24 13:56:00

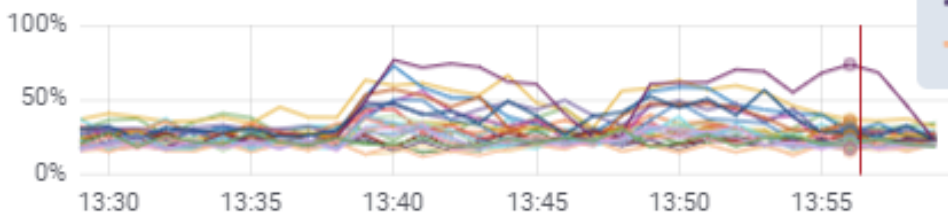
- bst-vostok03.sdg: 74%
- dtl-vostok02.sdh: 37%
- vostok09.sdb: 36%
- dtl-vostok02.sdf: 33%
- vostok09.sdg: 32%
- vostok10.sdg: 31%
- vostok10.sdi: 29%
- dtl-vostok01.sdc: 28%
- dtl-vostok03.sdg: 27%
- bst-vostok03.sde: 27%
- dtl-vostok02.sdb: 26%
- bst-vostok03.sdf: 26%
- vostok09.sdc: 24%
- bst-vostok02.sde: 23%
- dtl-vostok03.sdf: 23%
- dtl-vostok01.sdh: 22%
- dtl-vostok03.sdd: 22%
- dtl-vostok03.sdb: 21%
- vostok10.sdf: 21%
- dtl-vostok01.sde: 21%
- dtl-vostok03.sdh: 18%
- bst-vostok02.sdb: 18%
- dtl-vostok02.sdc: 18%
- bst-vostok03.sdd: 17%
- bst-vostok01.sdh: 16%

Disk Space Free, Bottom 20

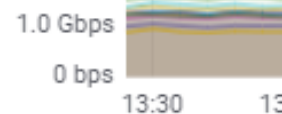


- dtl-vostok02_storage5
- vostok10_storage5
- dtl-vostok02_storage1
- vostok08_storage3

Disk Utilization, Top 25

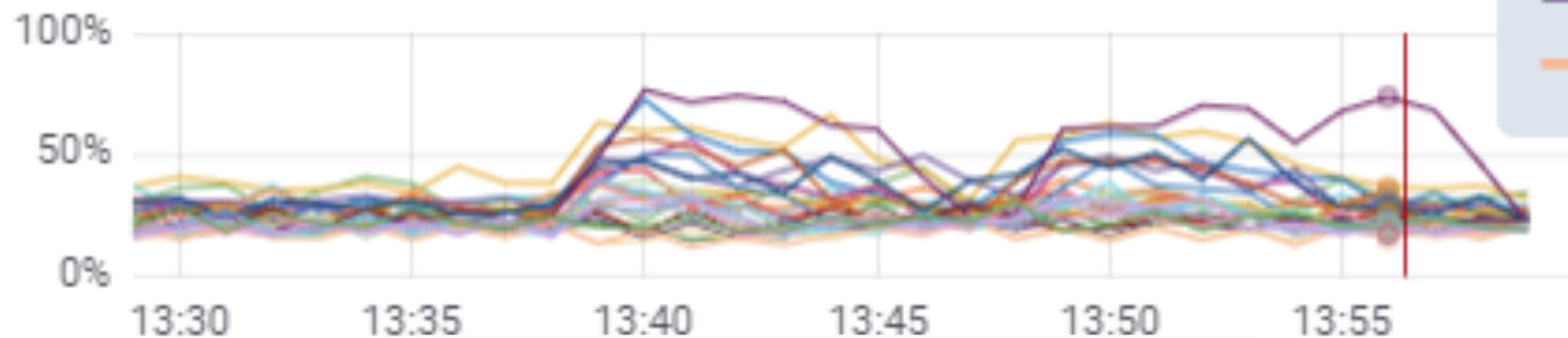


- vostok09.sdg
- dtl-vostok02.sdh
- dtl-vostok03.sdg
- dtl-vostok02.sdf
- dtl-vostok02.sdb
- dtl-vostok01.sdc
- dtl-vostok03.sdd
- vostok10.sdg



- vostok09.enp6s0d
- vostok10.enp6s0d

Disk Utilization, Top 25



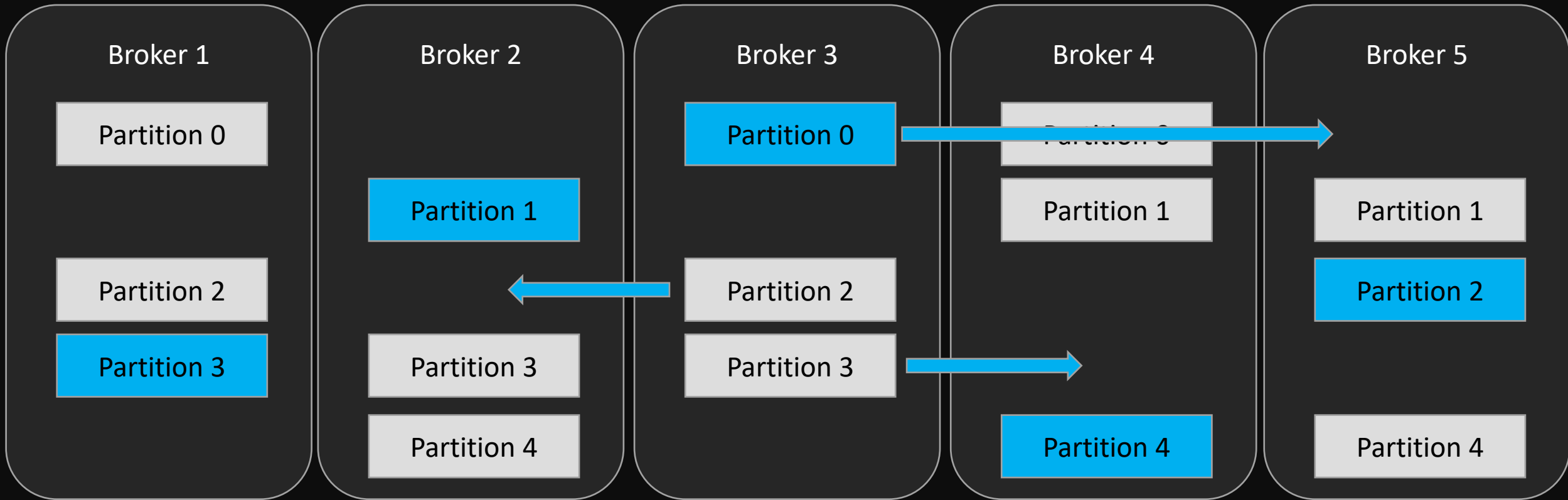
vostok09.sdg
dtl-vostok02.sdb

2019-10-24 13:56:00

bst-vostok03.sdg: 74%
dtl-vostok02.sdh: 37%

dtl-vostok02.sdf
vostok10.sdg

Перераспределение партиций. p5r3



Перераспределение партиций. p5r3

Broker 1

Partition 0

Partition 2

Partition 3

Broker 2

Partition 1

Partition 2

Partition 3

Partition 4

Broker 3

Broker 4

Partition 0

Partition 1

Partition 3

Partition 4

Broker 5

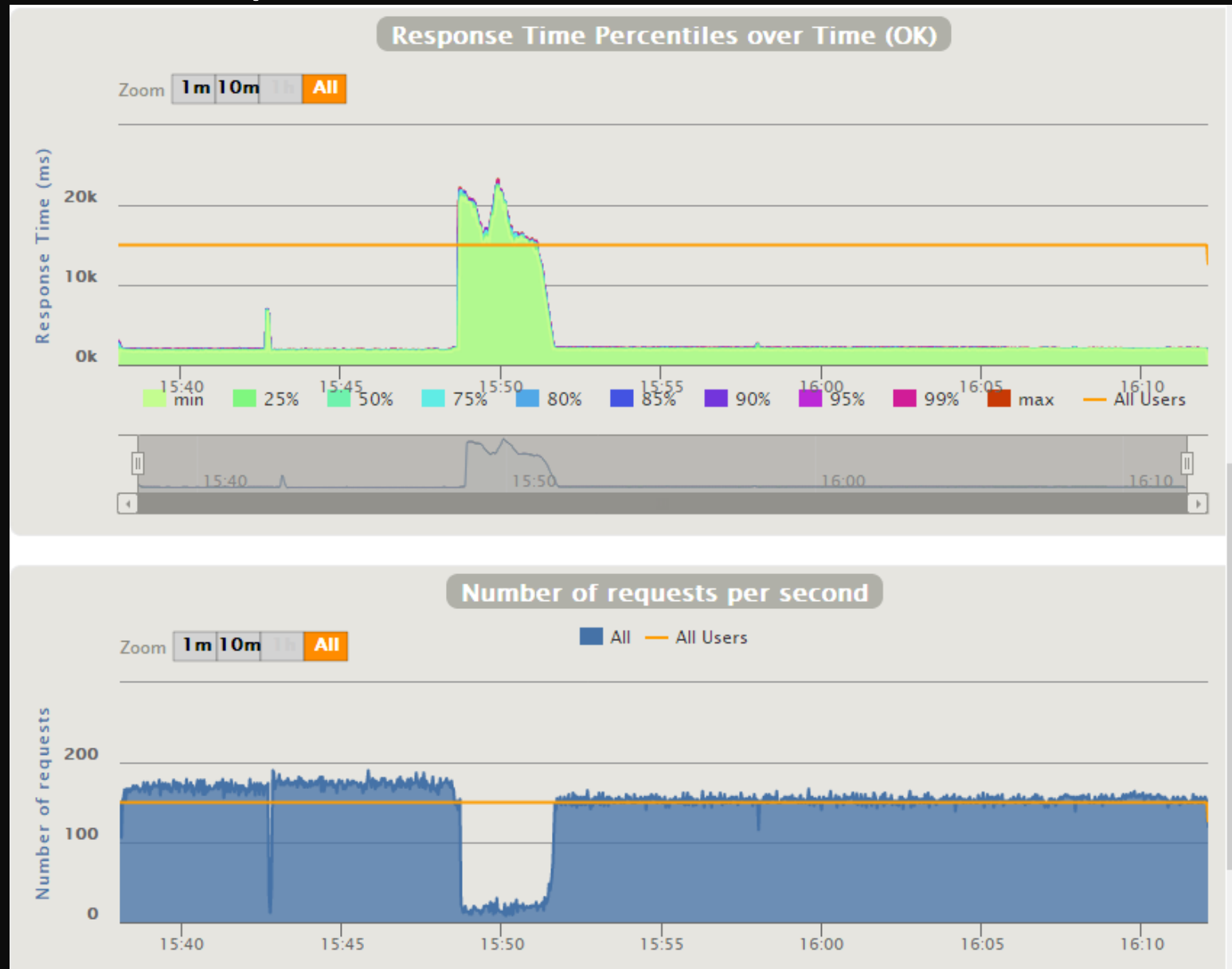
Partition 0

Partition 1

Partition 2

Partition 4

Репликация. Троттлинг



Троттлинг репликации

Репликация партиций сжирает ресурсы – Network, HDD –
Урезаем скорость

```
follower.replication.throttled.rate=100000000
```

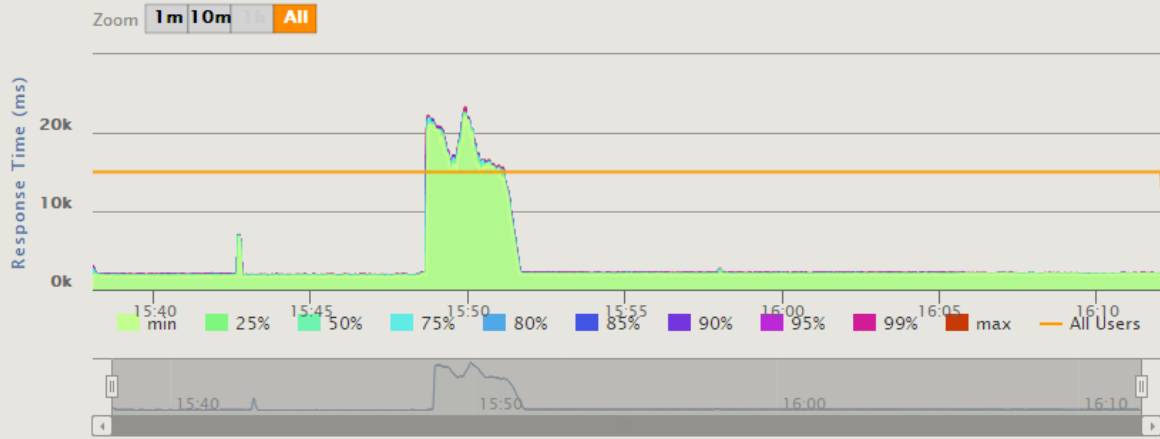
```
leader.replication.throttled.rate=100000000
```

Много – не успеваем писать текущий поток

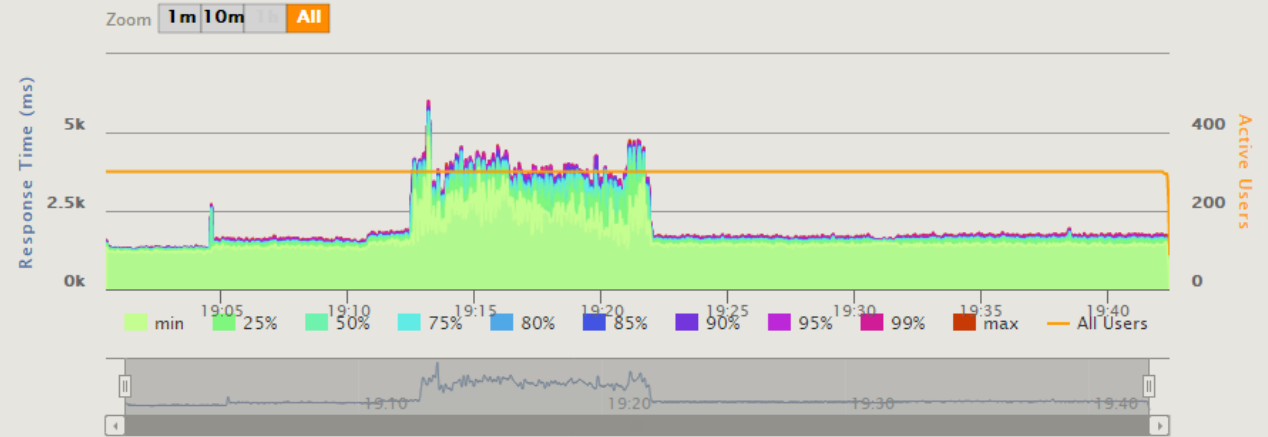
Мало – долго дочитываем – можем не дочитать никогда

Перезагрузка ноды. Троттлинг

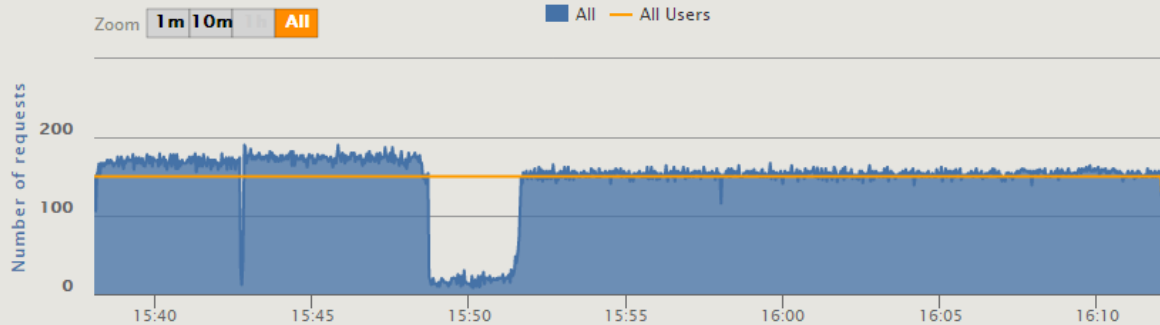
Response Time Percentiles over Time (OK)



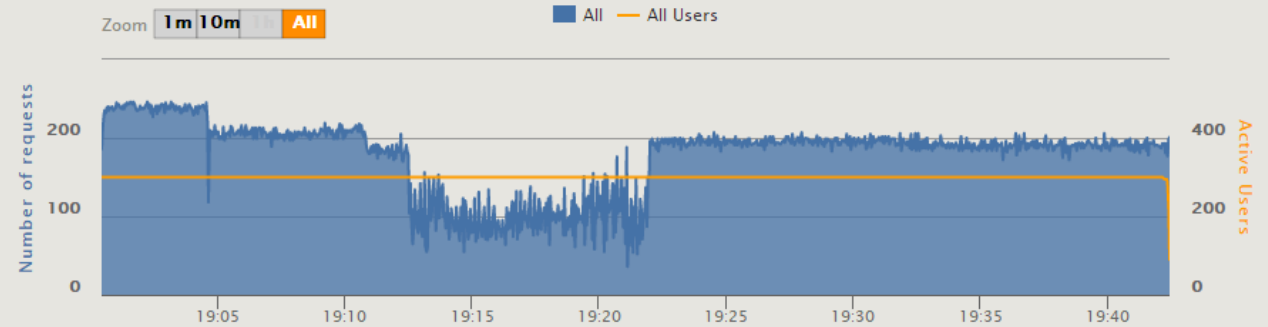
Response Time Percentiles over Time (OK)

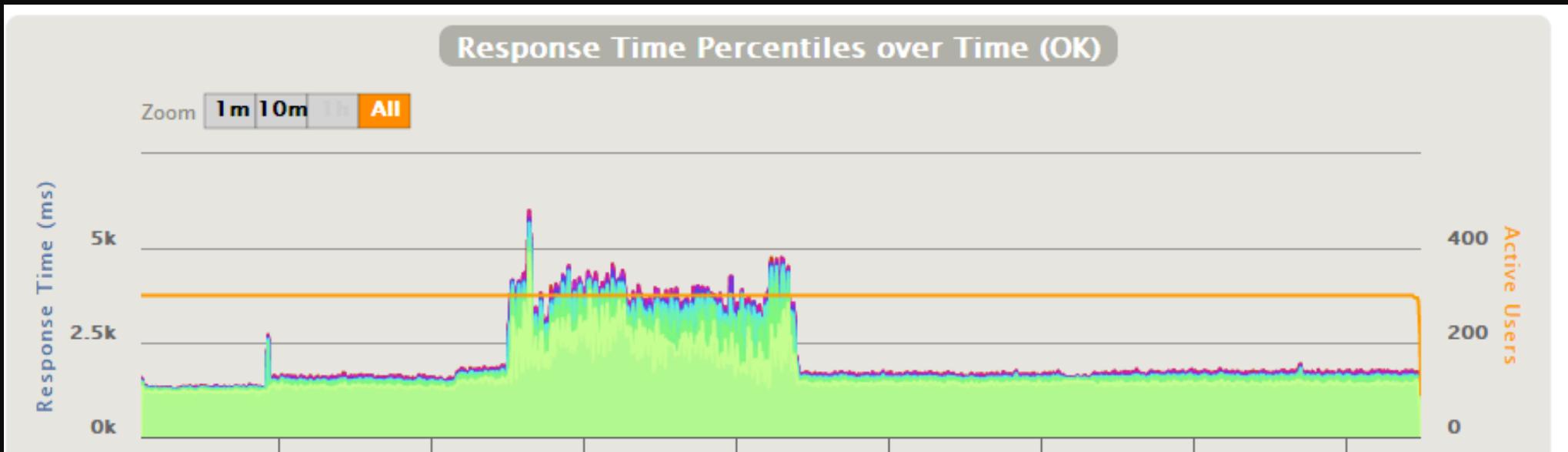
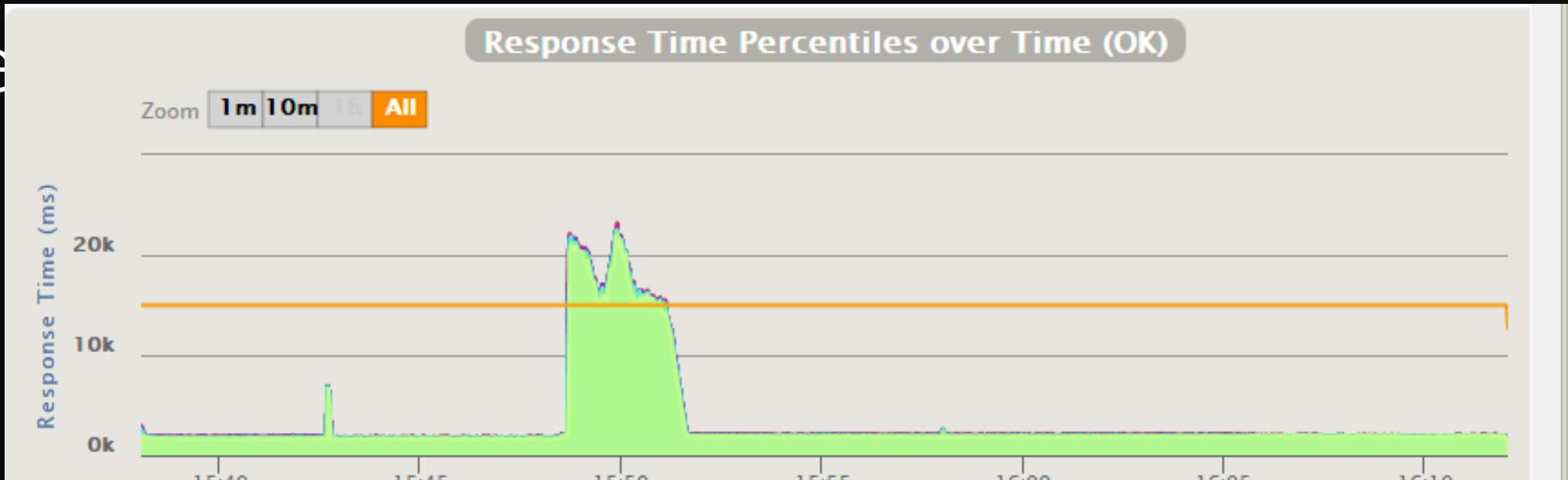


Number of requests per second

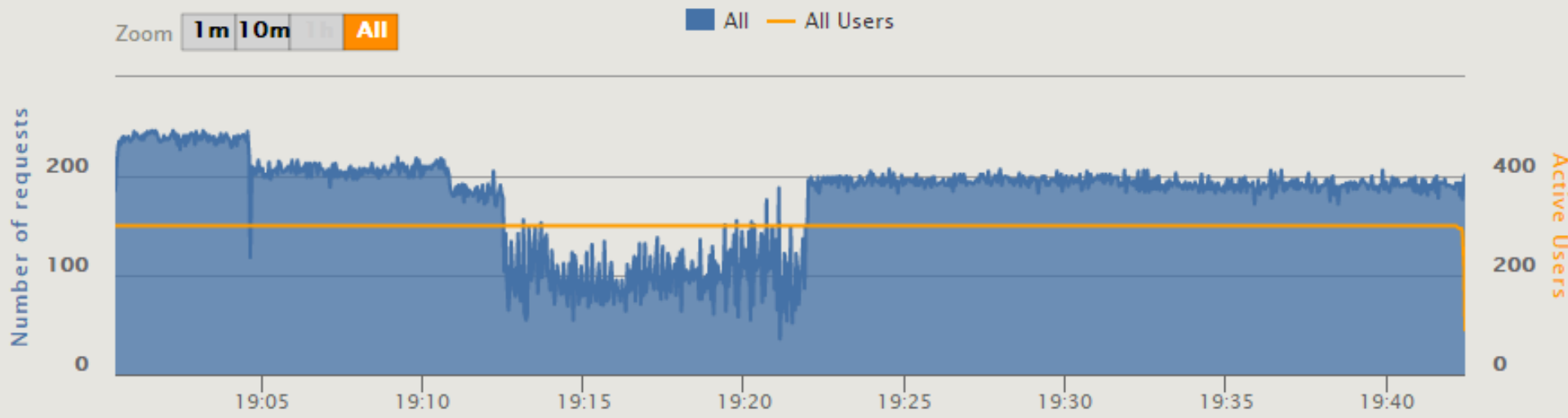


Number of requests per second

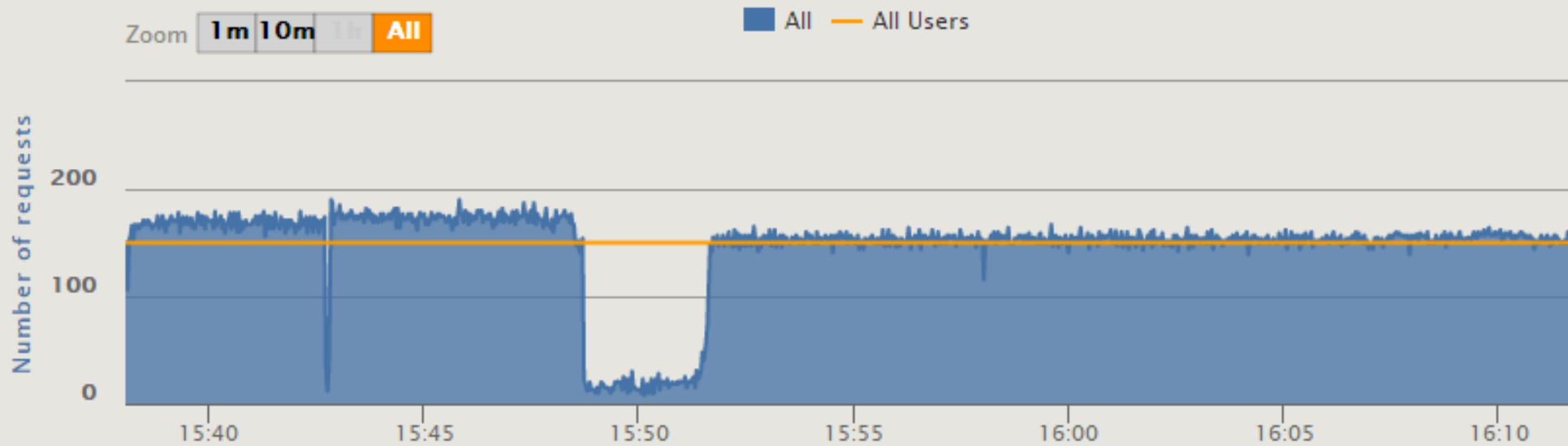




Number of requests per second



Number of requests per second



О чем еще хотелось бы поговорить

- Особенности работы с разными сценариями
- Поведение при проблемах с сетью
- Партиционирование при записи
- Особенности работы consumer group
- Тестирование обновлений
- ...

Немного рецептов для успешной работы с Кафкой

- Понимаем концепций топик, партиция, реплика, retention, лаг чтения
- Понимаем процессы, происходящие в системе, в какие ресурсы упираемся в различных ситуациях
- Помним о работе с памятью(PageCache)
- Оставляем запас ресурсов для обслуживания и обеспечения отказоустойчивости

Спасибо за внимание!

- Махетов Сергей
- E-mail: Profitfx@mail.ru
- Telegram: @Mahetovs
- <https://tech.kontur.ru/>