

The background features a dark blue gradient with large, overlapping, semi-transparent shapes in shades of purple, pink, and orange, creating a modern, abstract design.

AWS re:Invent

NOV. 27 – DEC. 1, 2023 | LAS VEGAS, NV

NFX304

How Netflix leverages AWS for multi-Region cache replication

Prudhviraj Karumanchi

(He/Him)
Staff Software Engineer
Netflix

Sriram Rangarajan

(He/Him)
Senior Software Engineer
Netflix

Prateek Sharma

(He/Him)
Principal Solutions Architect
AWS





Spark joy



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Introduction

Global Replication

Design and Architecture

Observability

Efficiency Improvements

Learnings from production deployments

Future

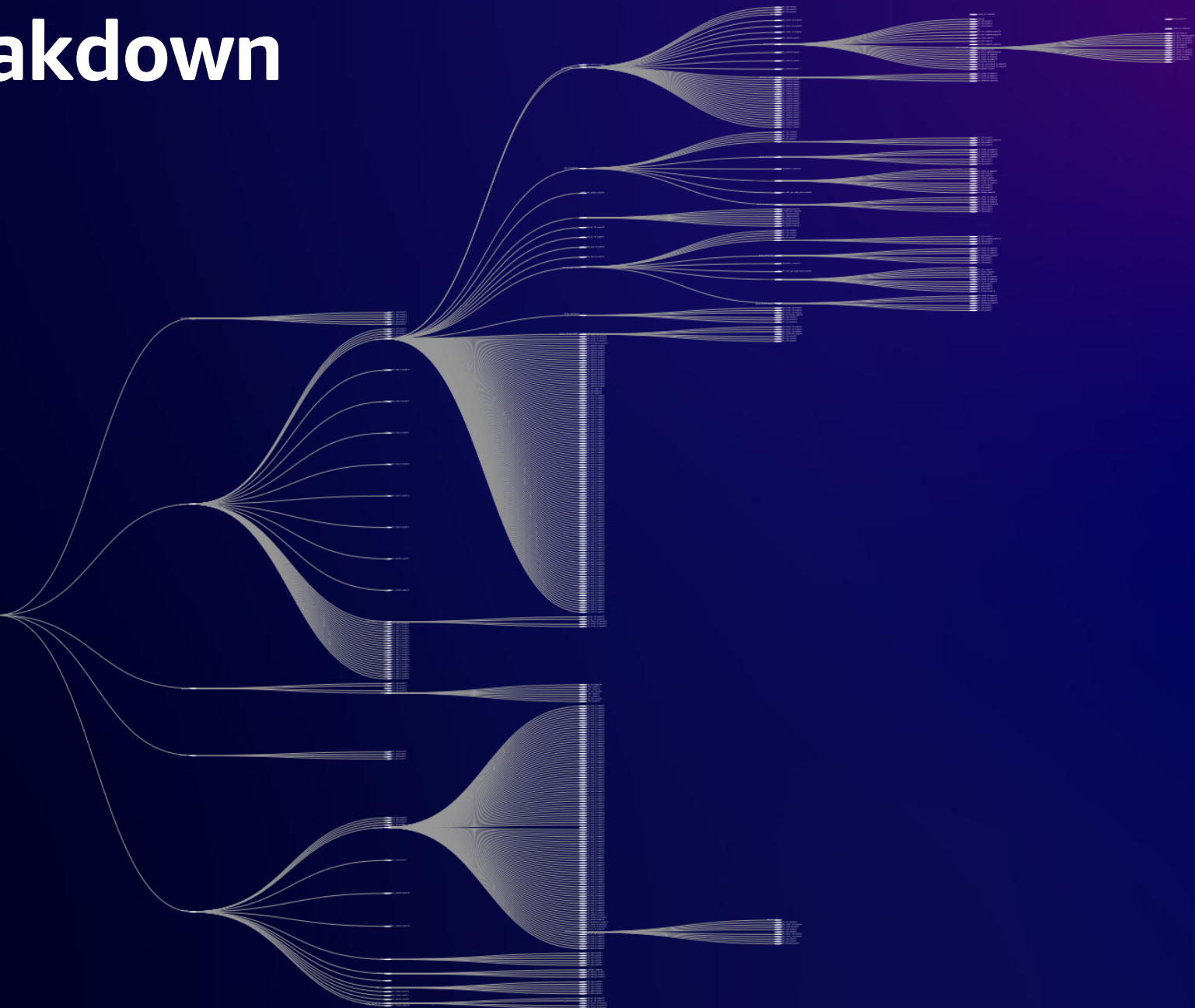


Agenda

Introduction



Homepage breakdown





Distributed, sharded, replicated key-value store

Based on Memcached

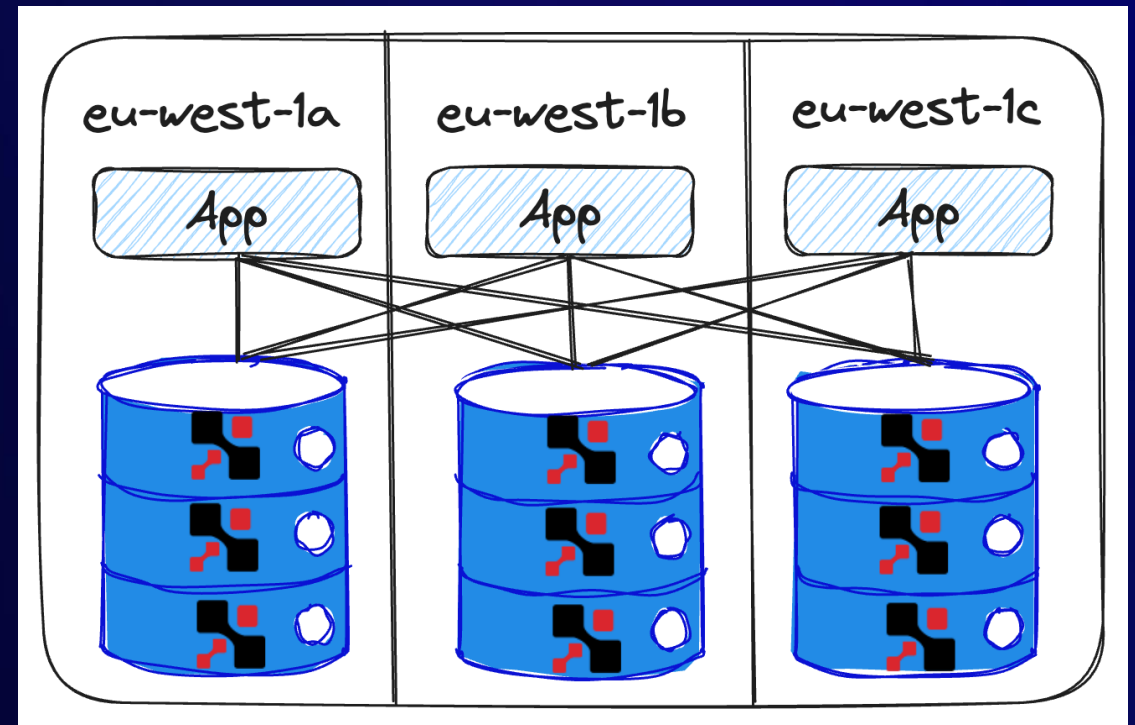
Tunable in-Region and global replication

Resilient to failures

Topology aware

Linearly scalable

Seamless deployments



EVCache

4 regions

~194 clusters

~22,000 servers

~30,000,000 replication

~400,000,000 ops/sec

~2,000,000,000,000 items

~14,300,000,000,000,000 bytes

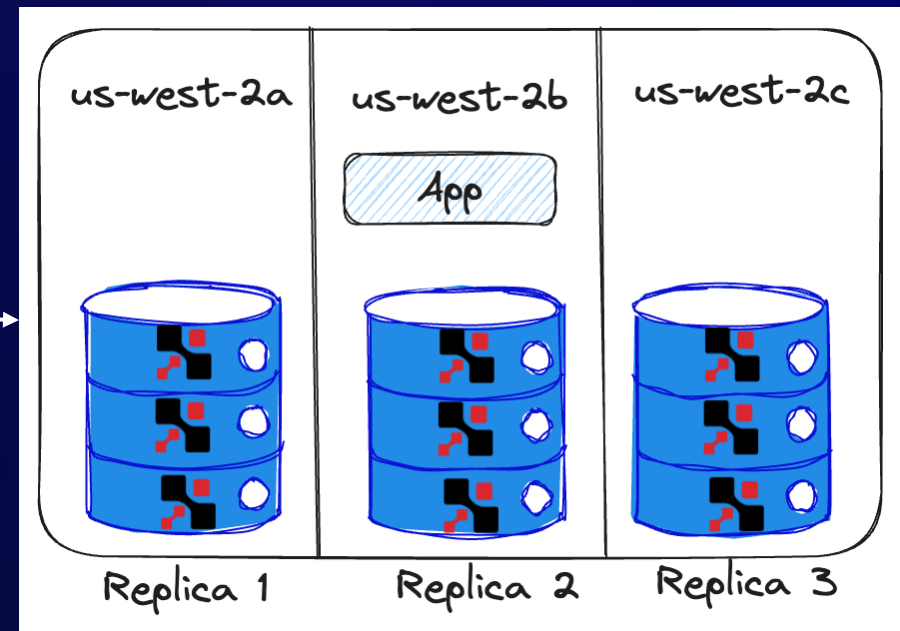
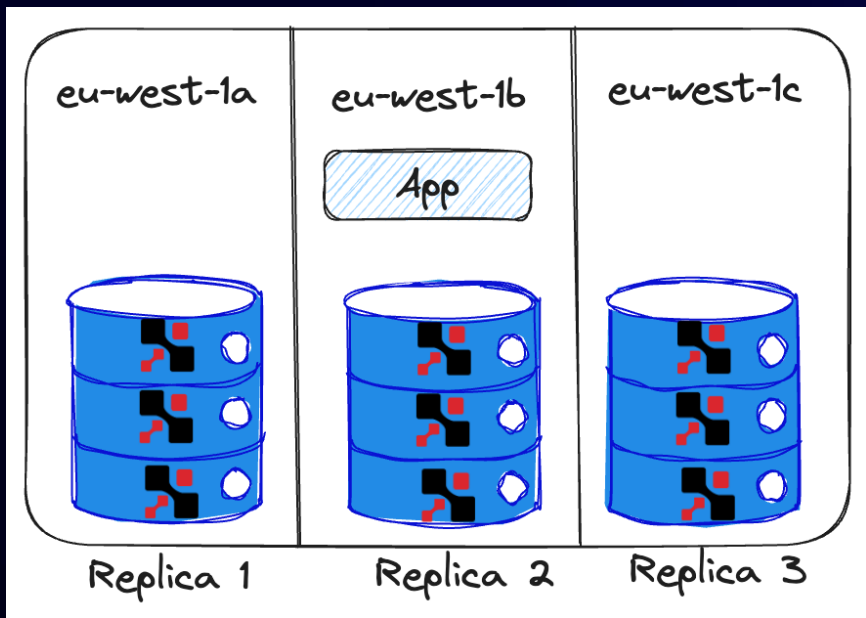
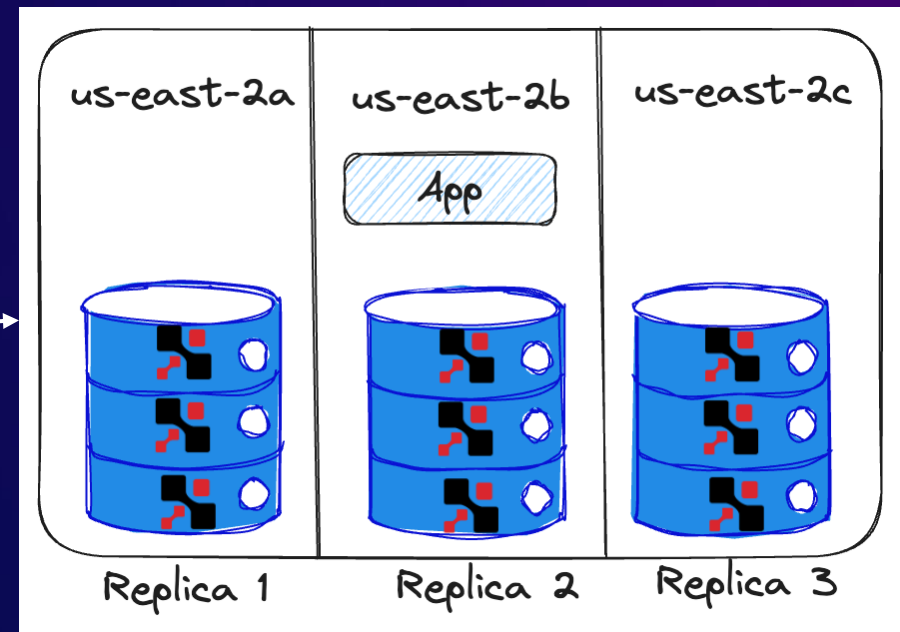
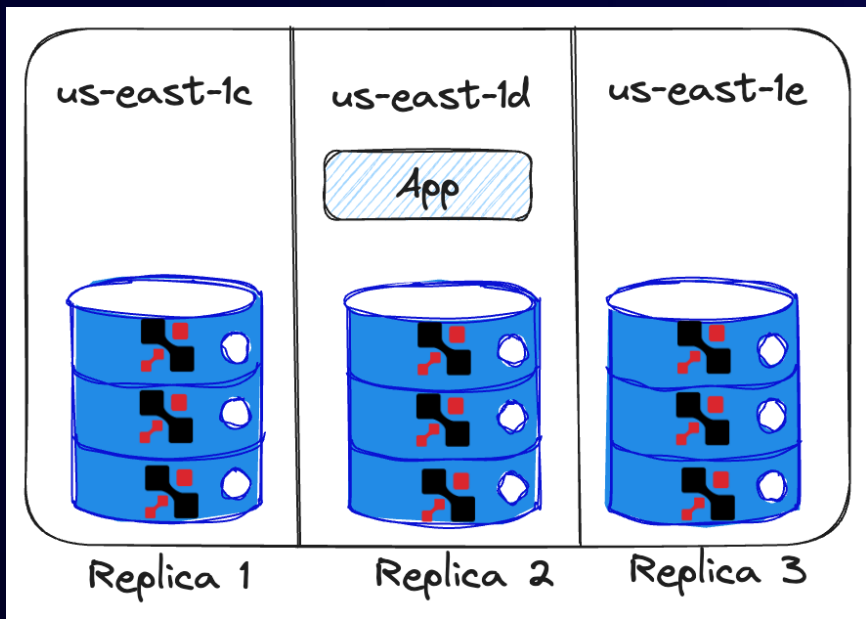


Agenda

Introduction

Global Replication

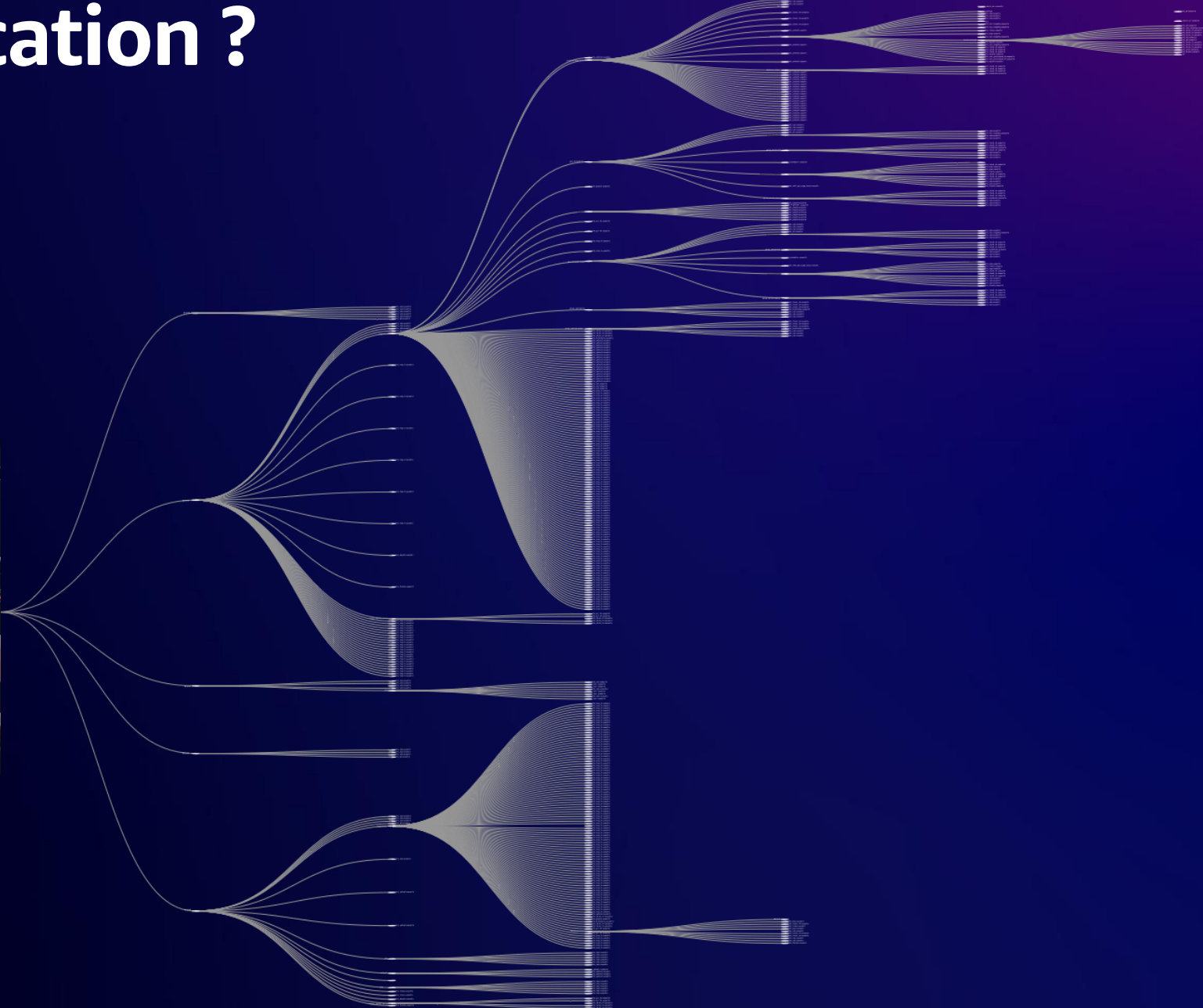
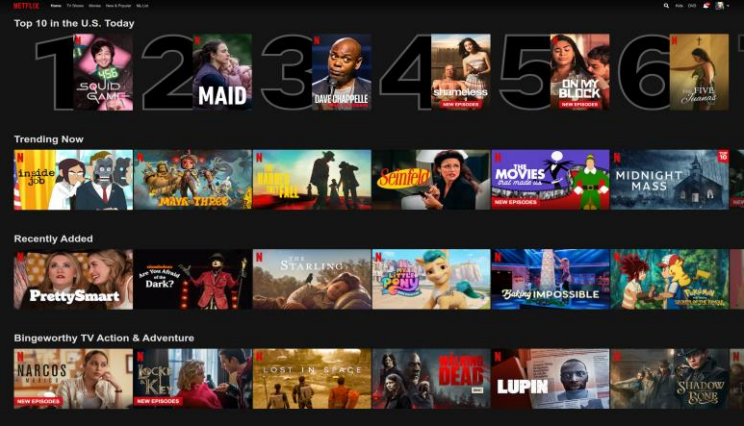




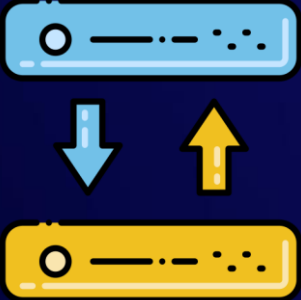
Why Global Replication ?



Why Global Replication ?



Why Global Replication ?



Region Failover

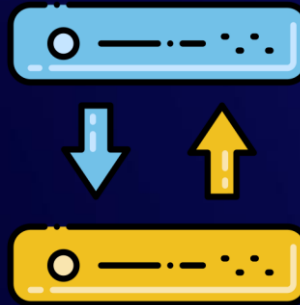
Empty Cache



Database



Why Global Replication ?



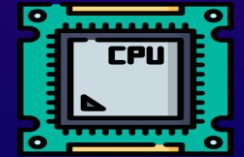
Region Failover



Empty Cache



Compute Intensive



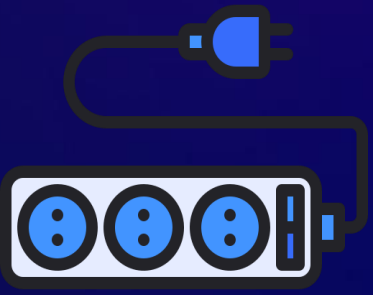
Replication Design Goals



Latency



Tunables

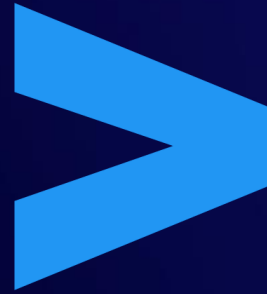


Extensibility

Replication Design Requirements



High availability

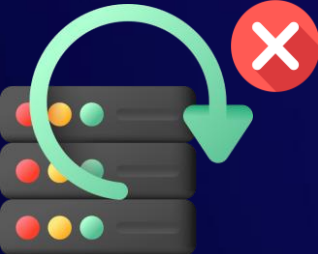


Best Effort Consistency

Design Non Requirements



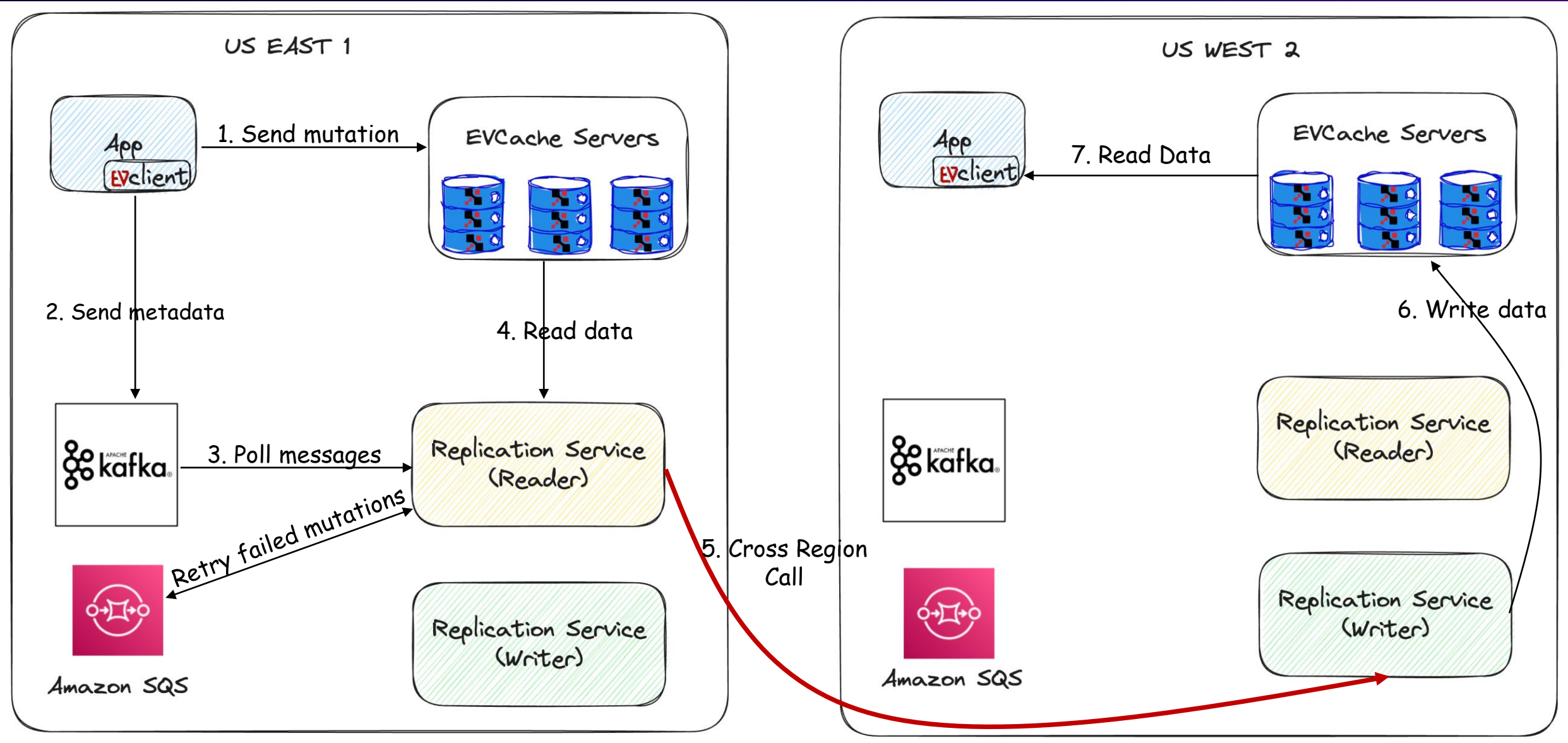
Global locking

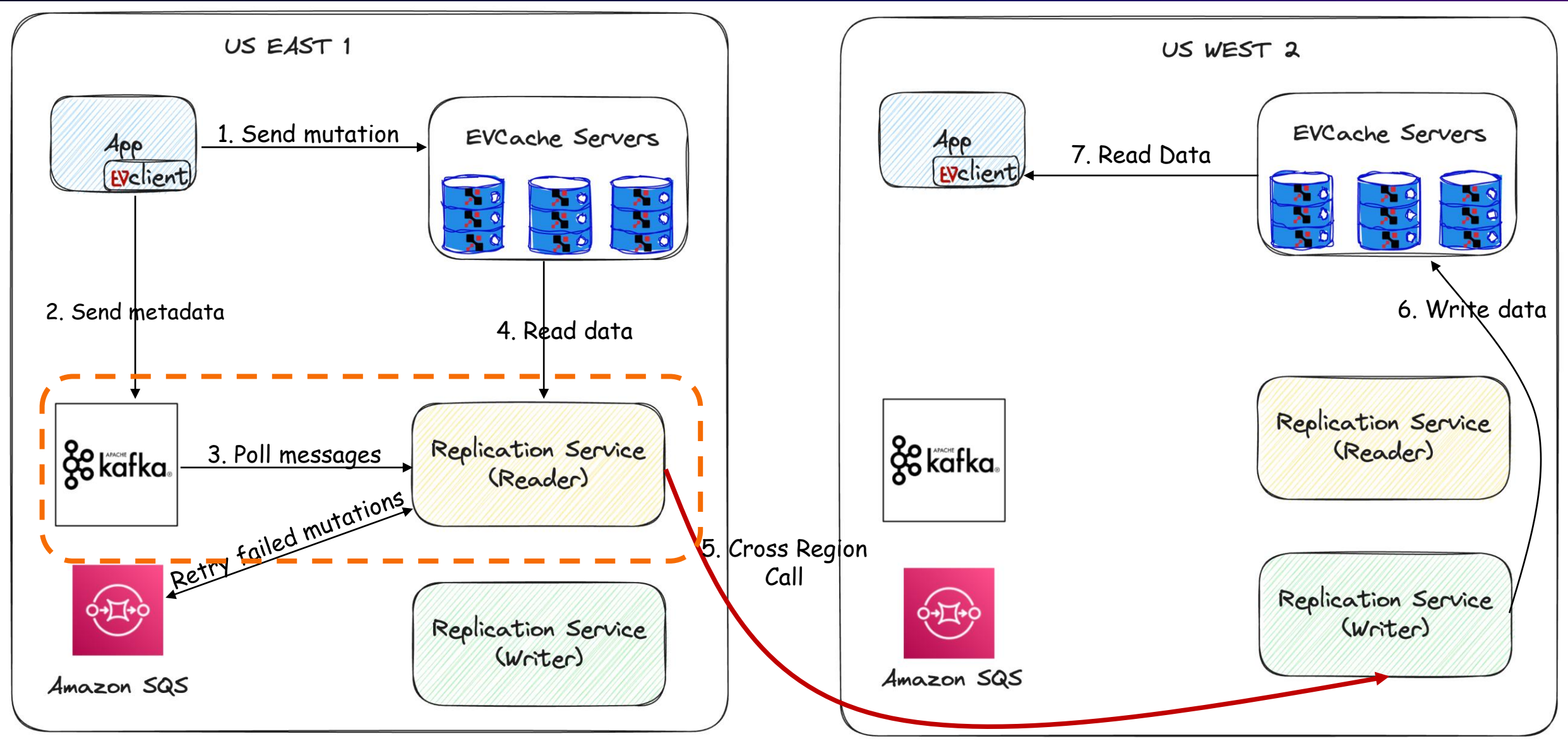


Partial-commit rollbacks



Transactional updates





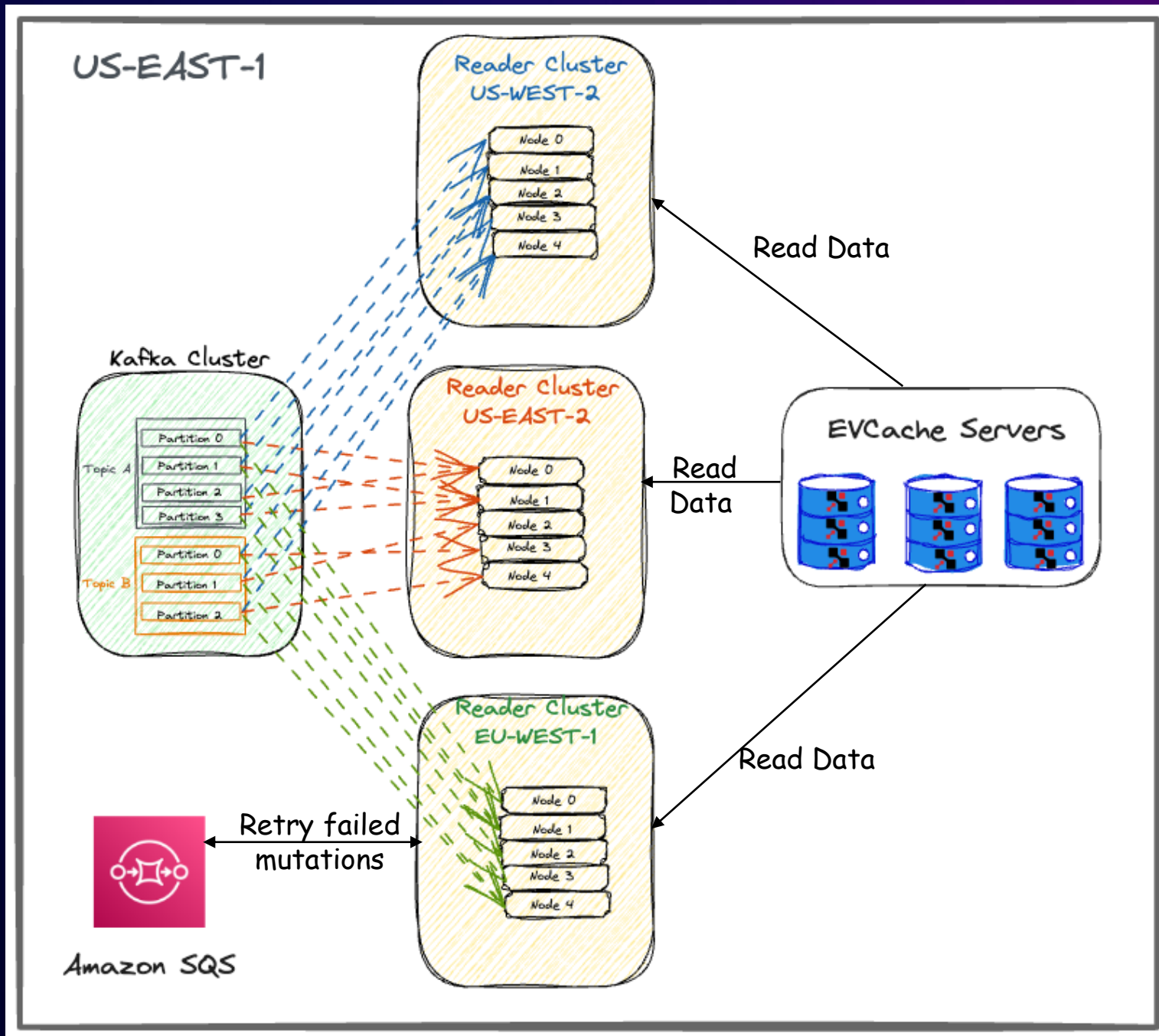
Readers

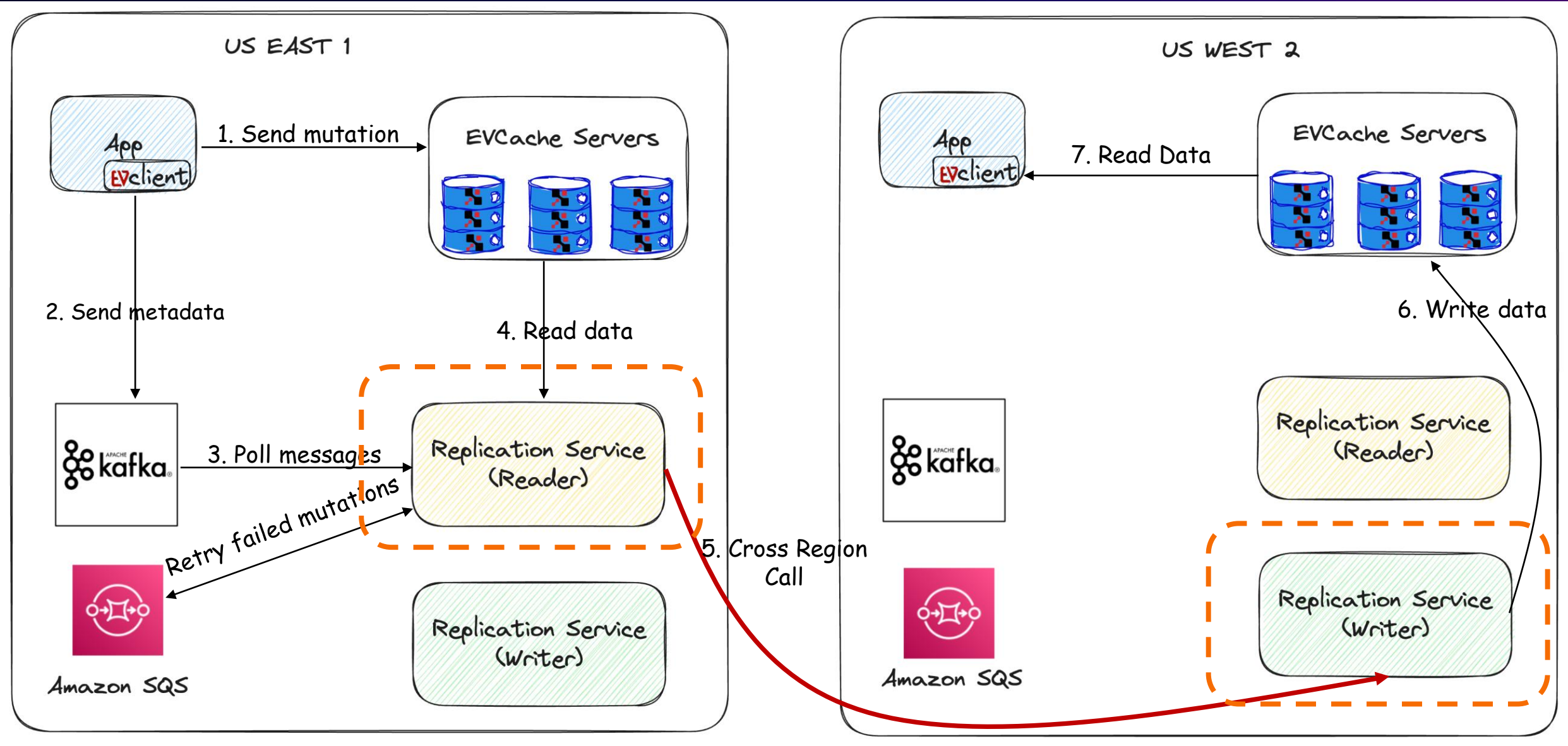
Different consumer groups reading from the Kafka cluster

Apply transformations of the incoming mutations

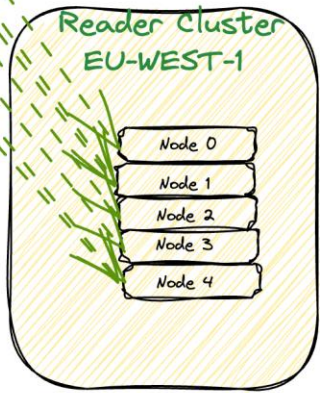
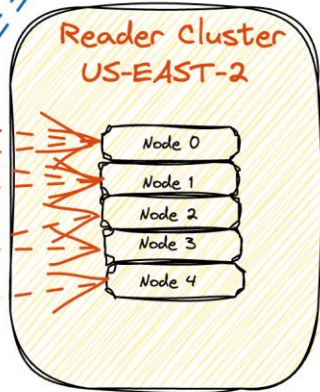
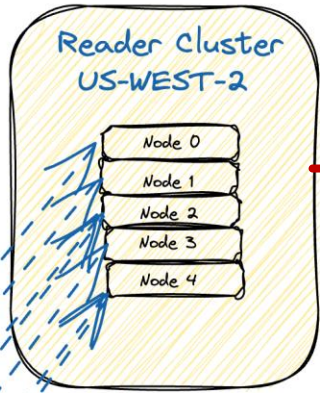
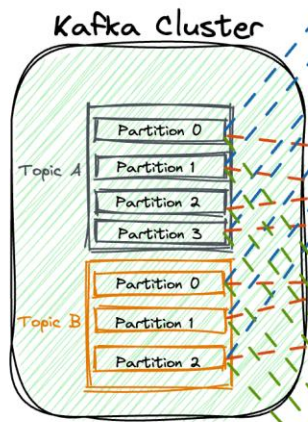
Perform local read

SQS



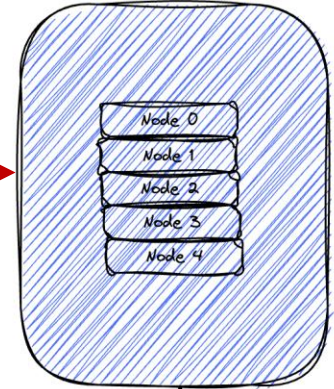


US-EAST-1

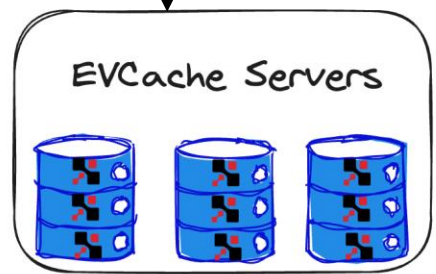


Cross-Region
traffic

US-WEST-2



Native Call



Backend cluster
(EVCache)



Extensibility

Generic Replication Service

Write Ahead Log

Delay Queues



Extensibility

Kafka Configurations	Caching	Write Ahead Log
	High Availability	High Durability
unclean.leader.election.enabled	true	false
default.replication.factor	2	3
min.insync.replicas	1	2
acks	1	all

Design Advantages



Independent scalability



Predictable end-to-end latency



Tunables

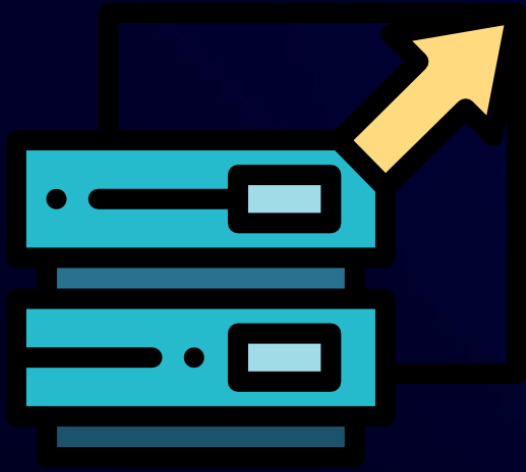


Cost efficient



Back pressure

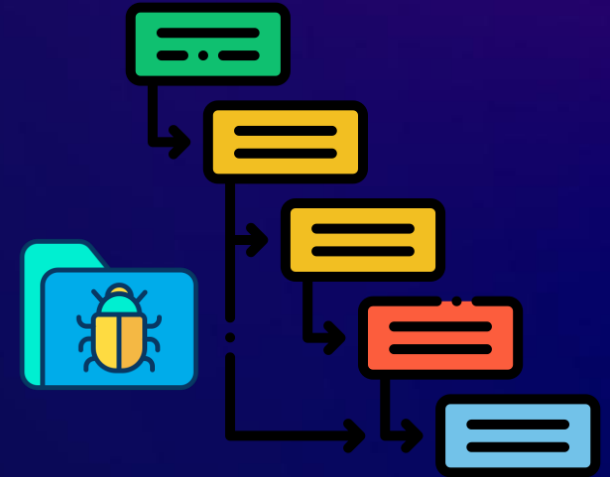
Design Pitfalls



Scaling policies



Message duplication



Cascading failures

Agenda

Introduction

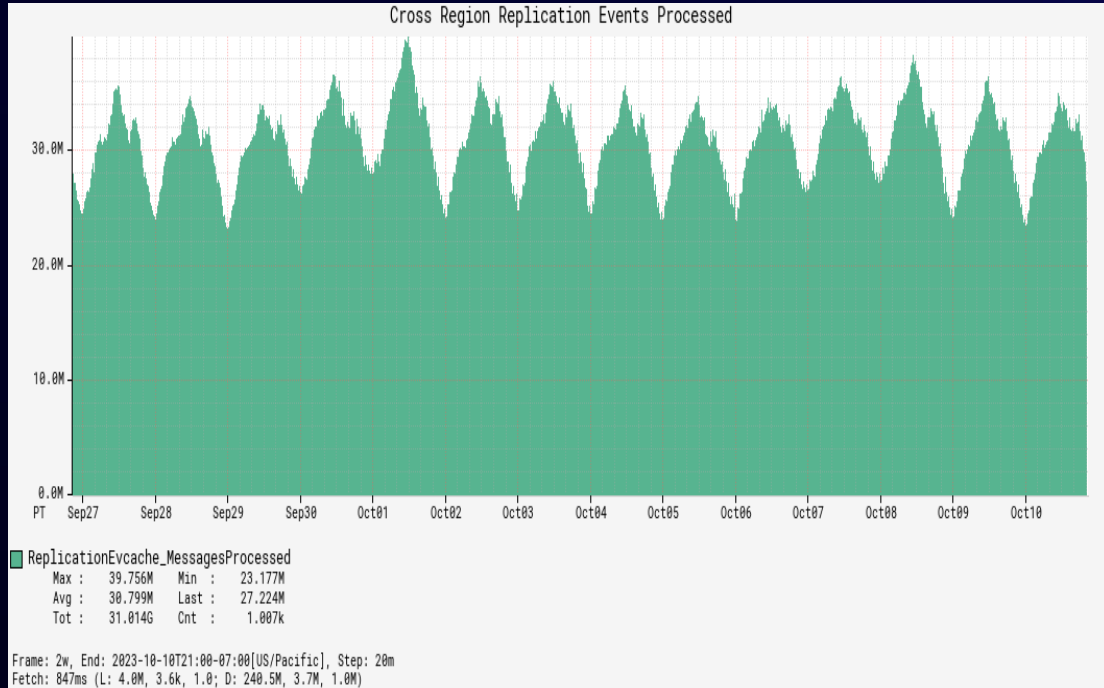
Global Replication

Design and Architecture

Observability

Observability & Alerts

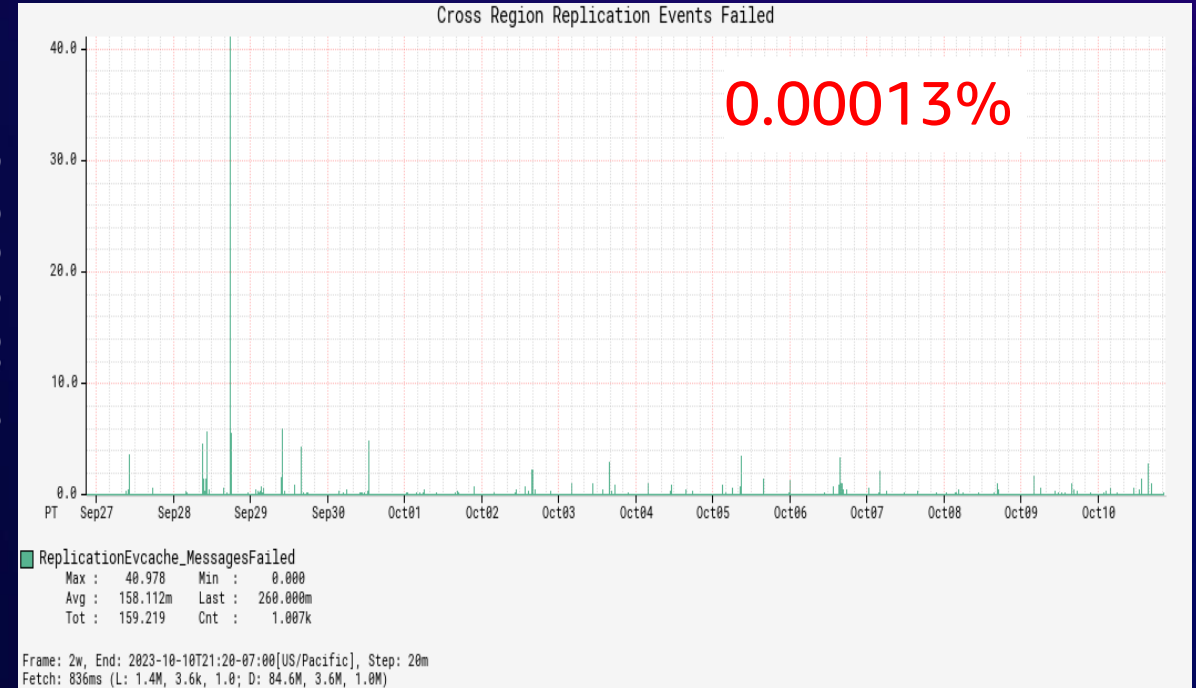
Events Processed



Time

Max : ~40 Million Replication Events
Avg: ~31 Million Replication Events

Events Failed

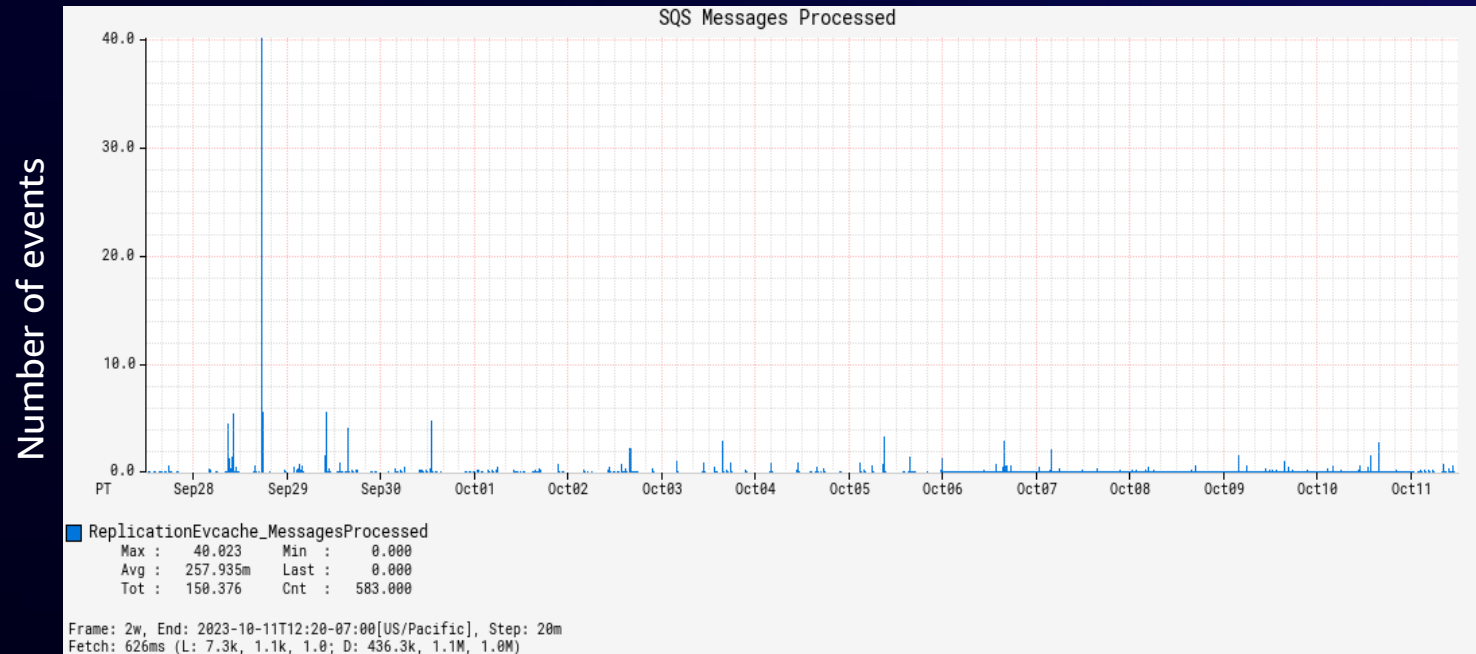


Time

Max : ~40 Failure

Alerts & Observability

Events Retried with SQS



Max : ~40 Failure



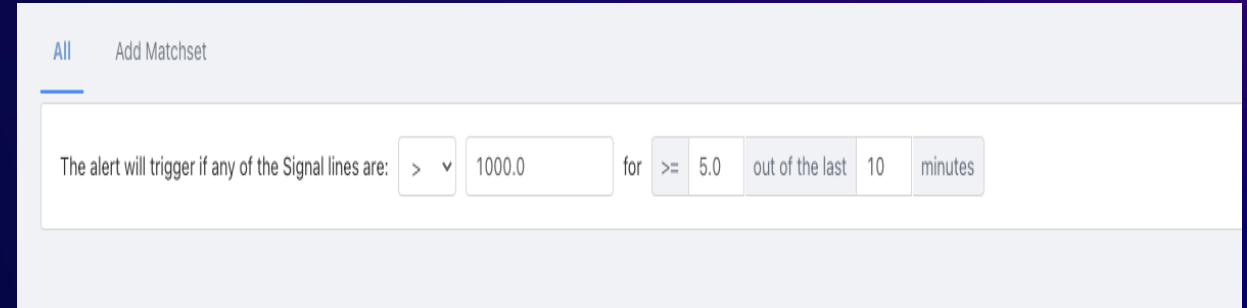
~ 0% Failure

Observability & Alerts

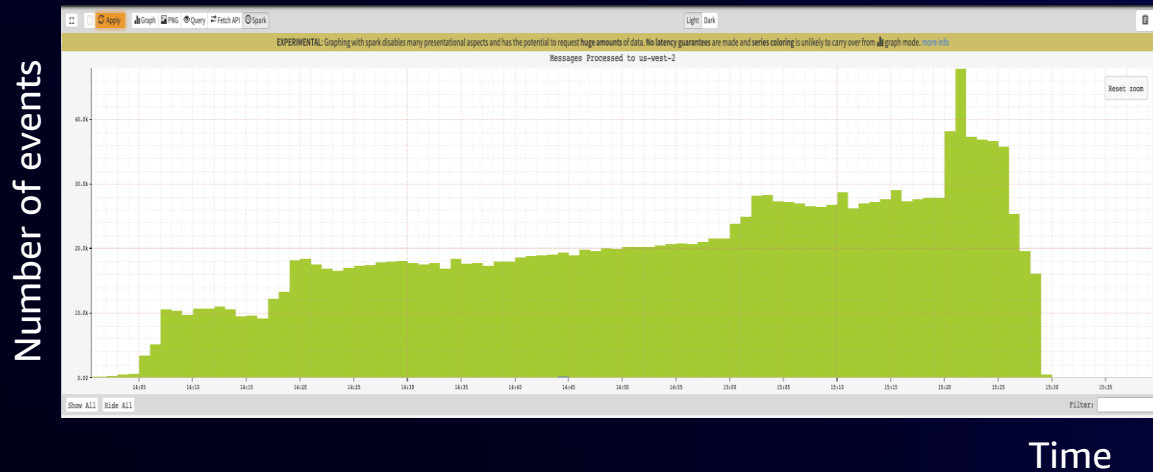


CRR Event Failure

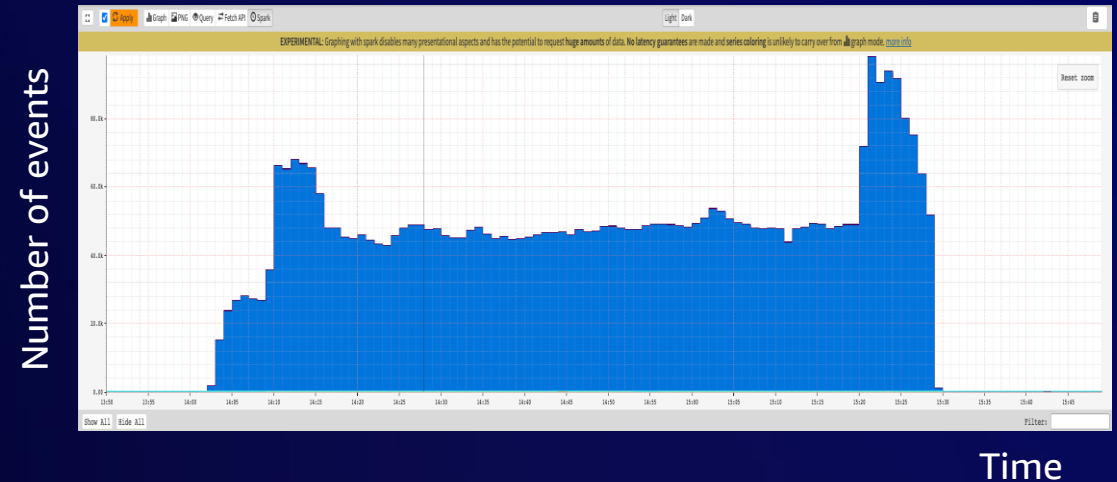
Number of failures > 1000 per minute
5 minutes in a 10 min window



Events Retried with SQS

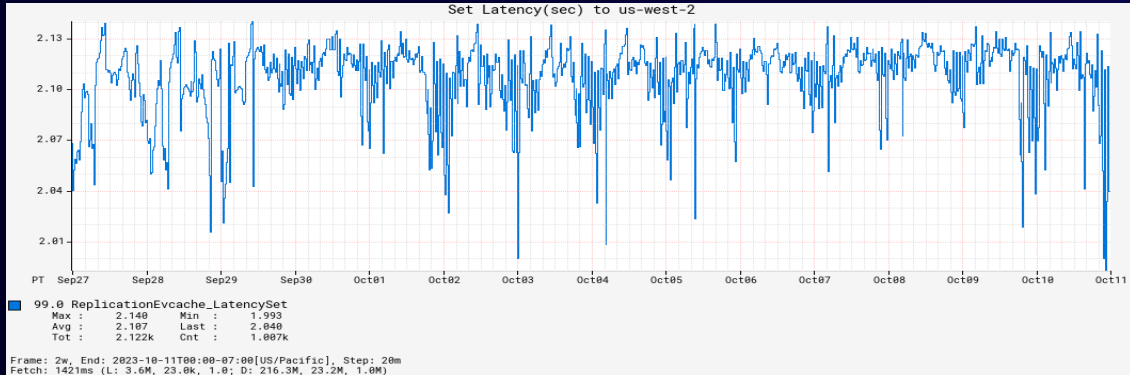


Failed Replication events



Observability – Latencies

Seconds



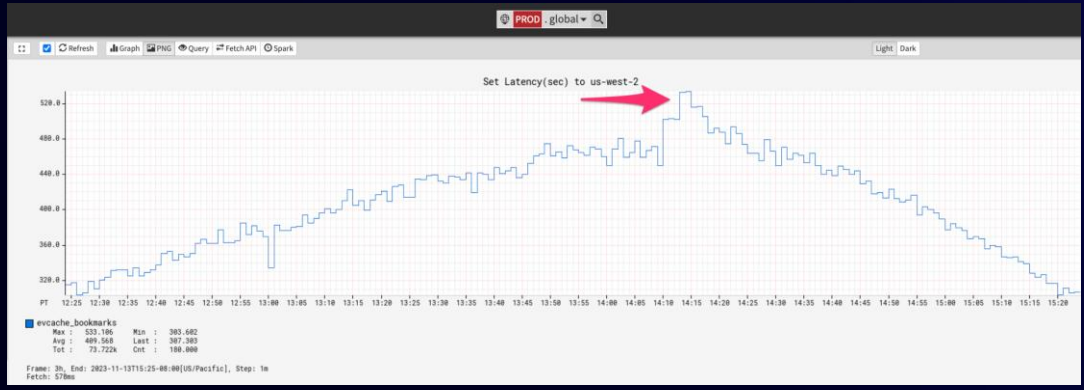
Time

Set Latency
Avg: ~2.10 seconds
Max: ~2.14 seconds

Latency
Alert



Seconds



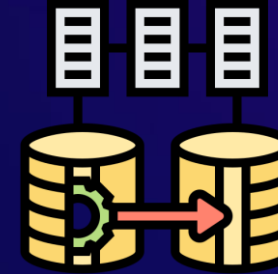
Time

Max: ~500 seconds



Observability – Autoscaling

• LIVE

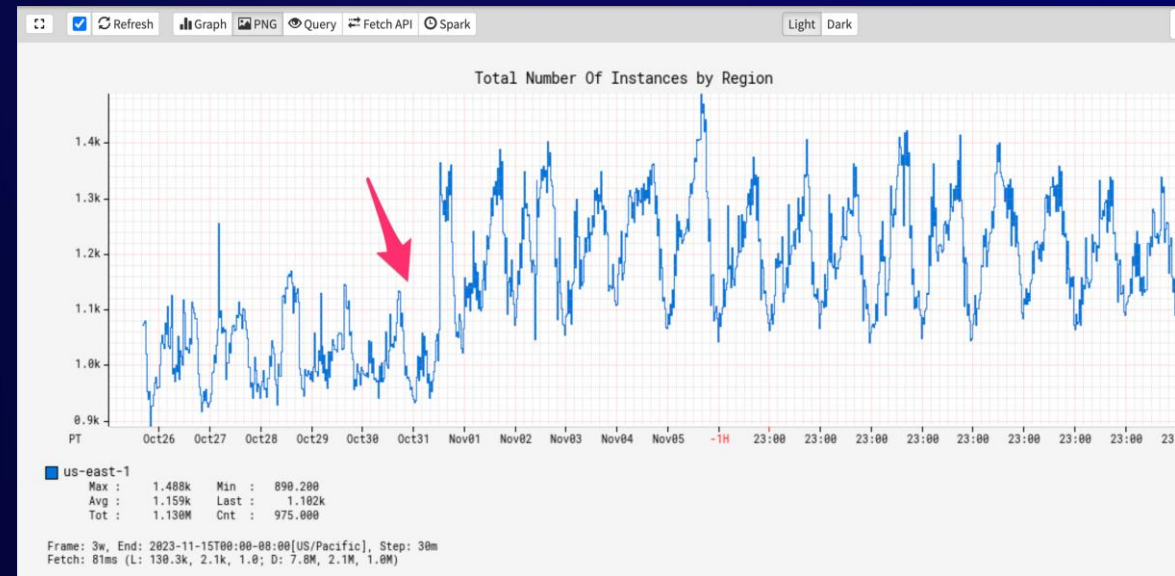


Number of events



Time

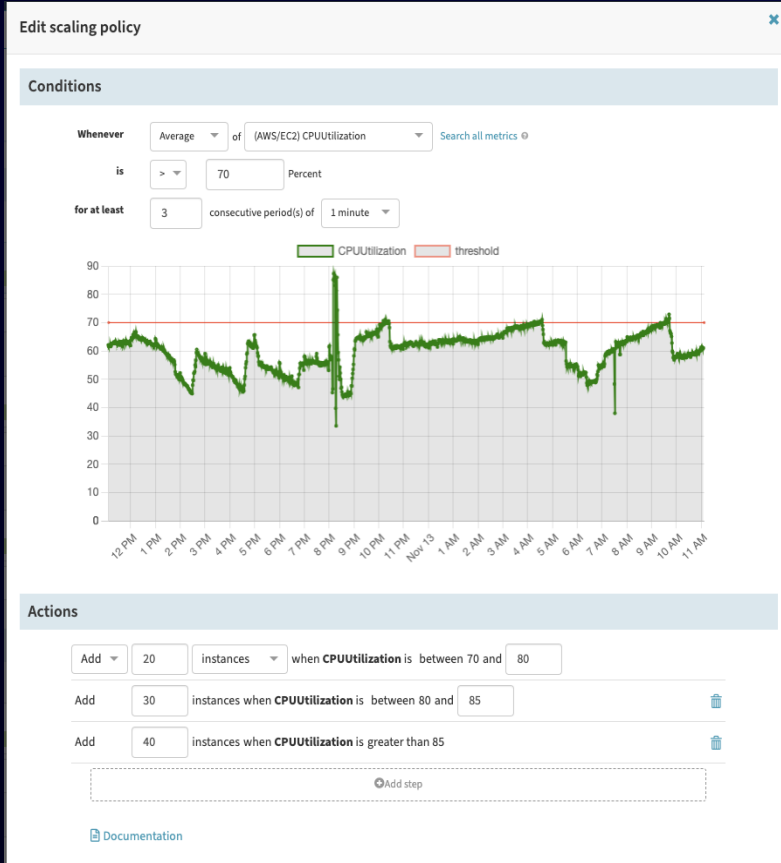
Number of instances



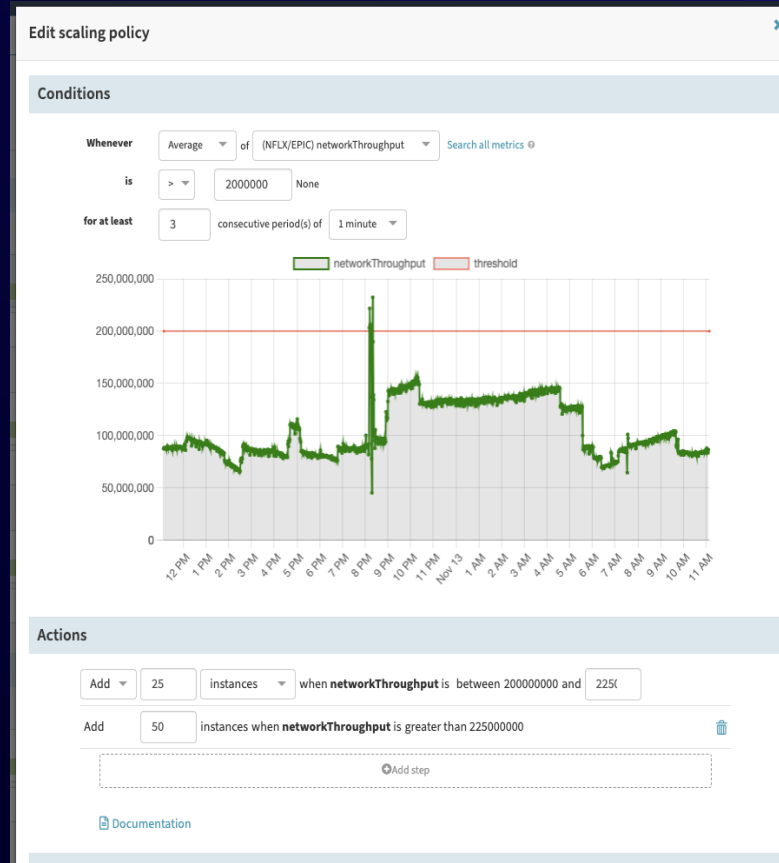
Time



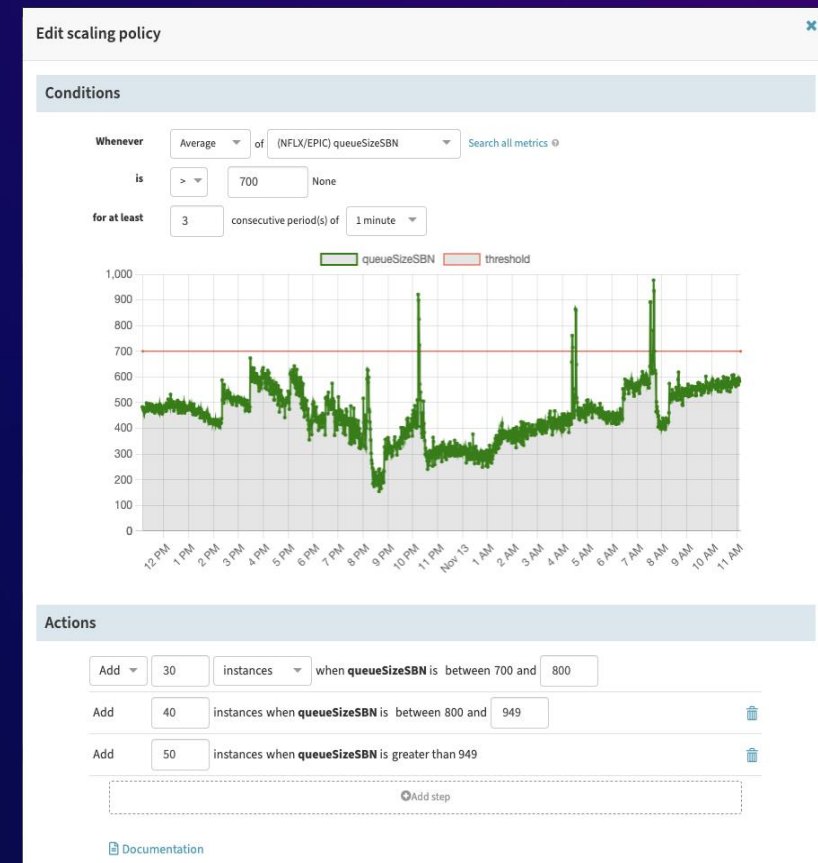
Observability – Autoscaling



CPU > 70%



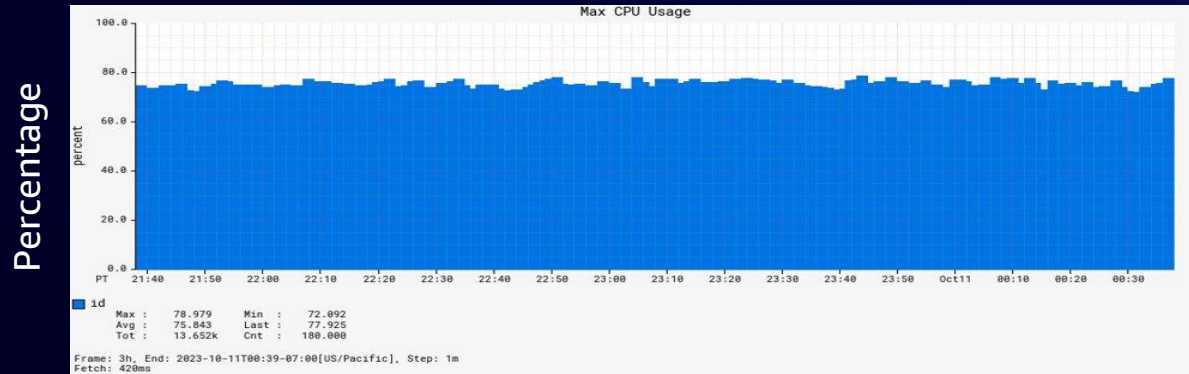
Network > 200000000 bytes



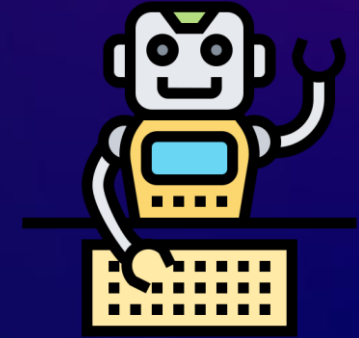
Queue Size > 700

Observability – Resource Utilization

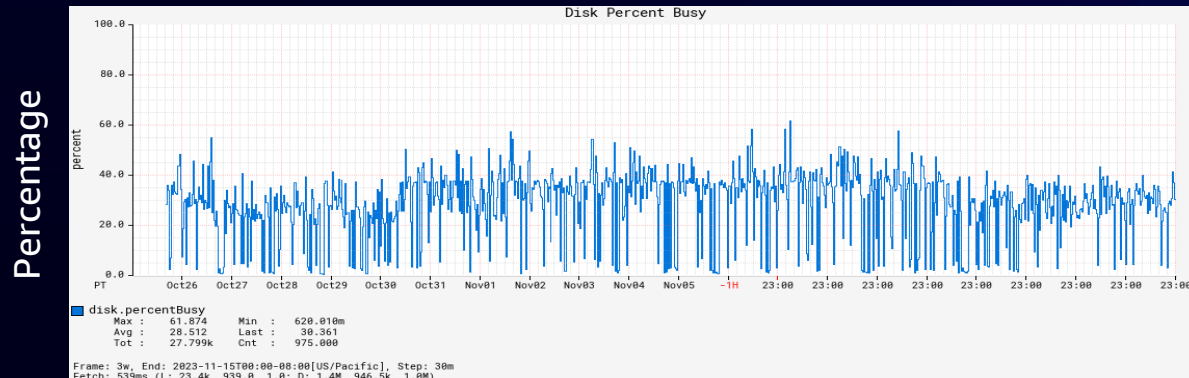
Max CPU Utilization ~ 70%



Time



Max Disk Utilization ~ 40%



Time



Agenda

Introduction

Global Replication

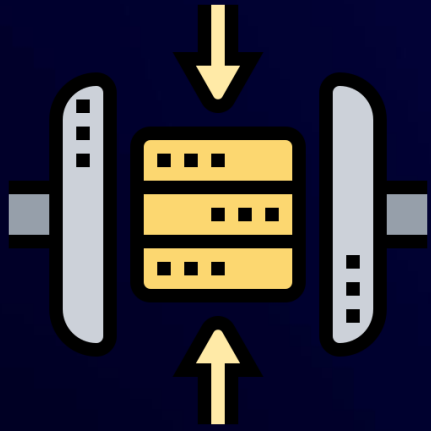
Design and Architecture

Observability

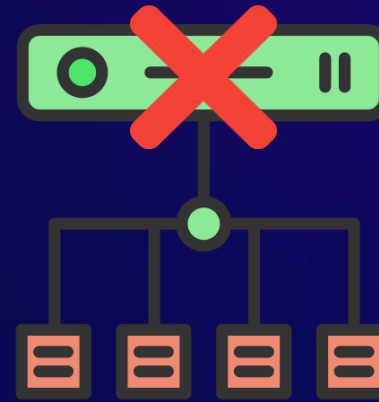
Efficiency Improvements



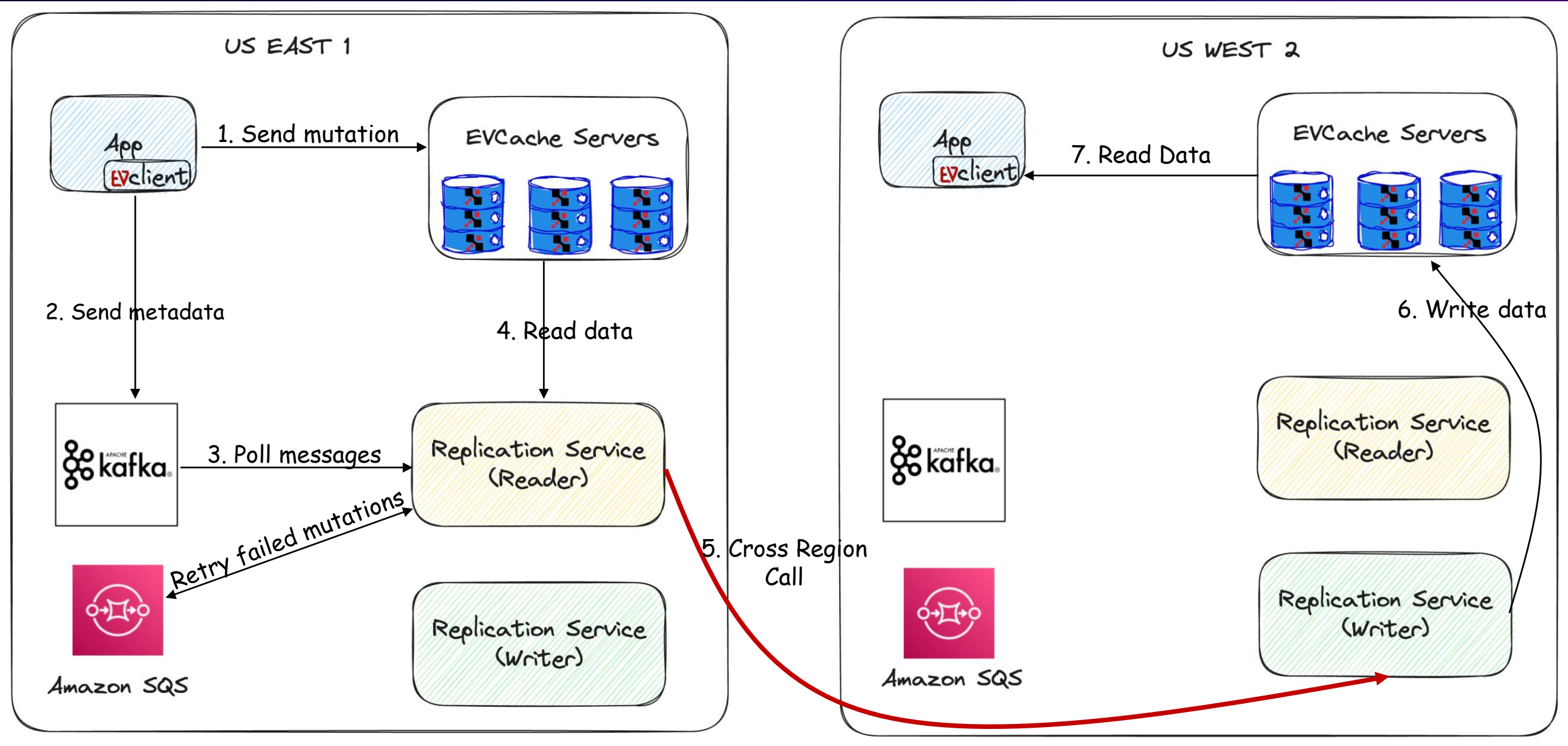
Efficiency Improvements



Batch compression

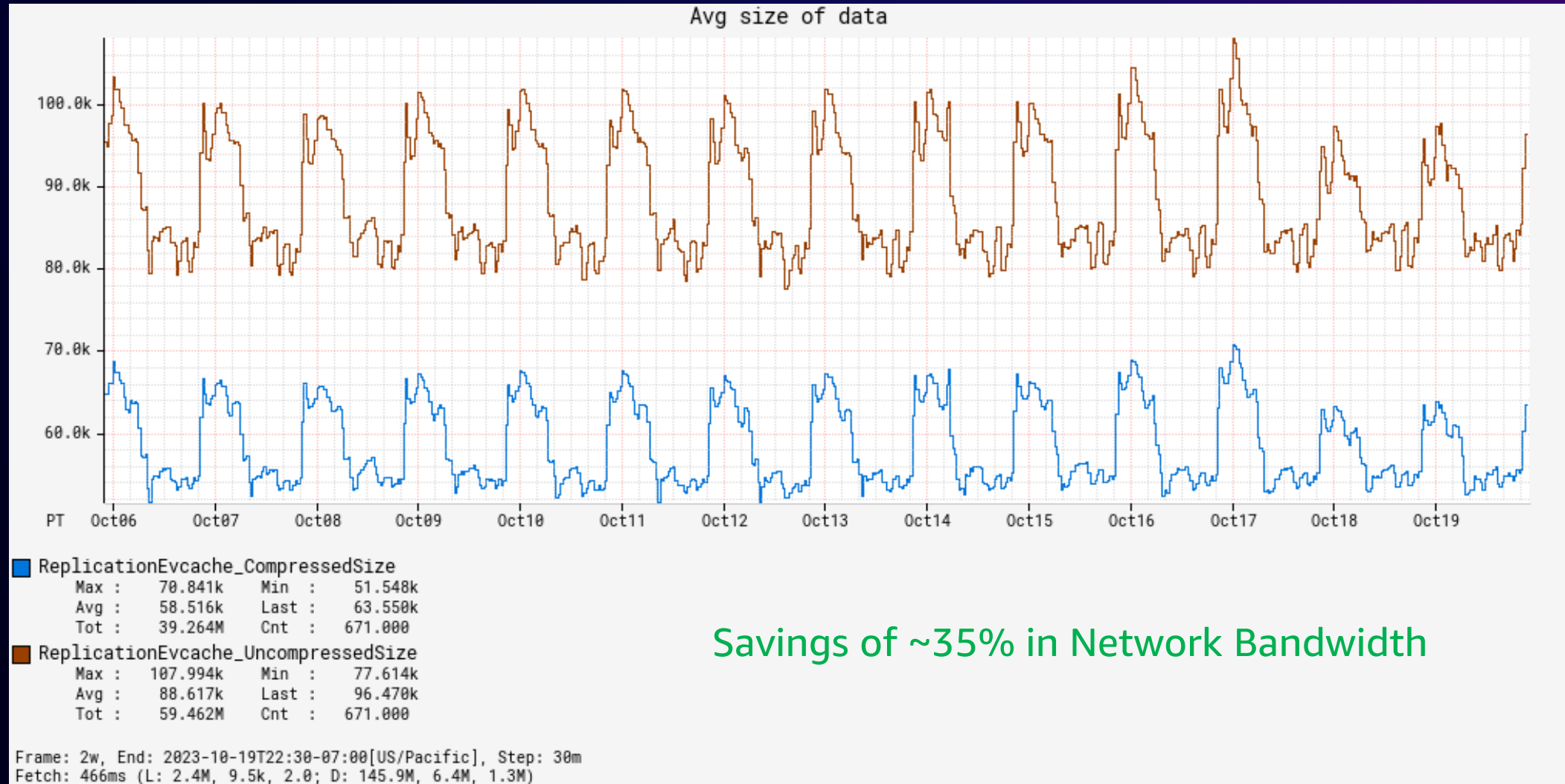


Removing NLBs



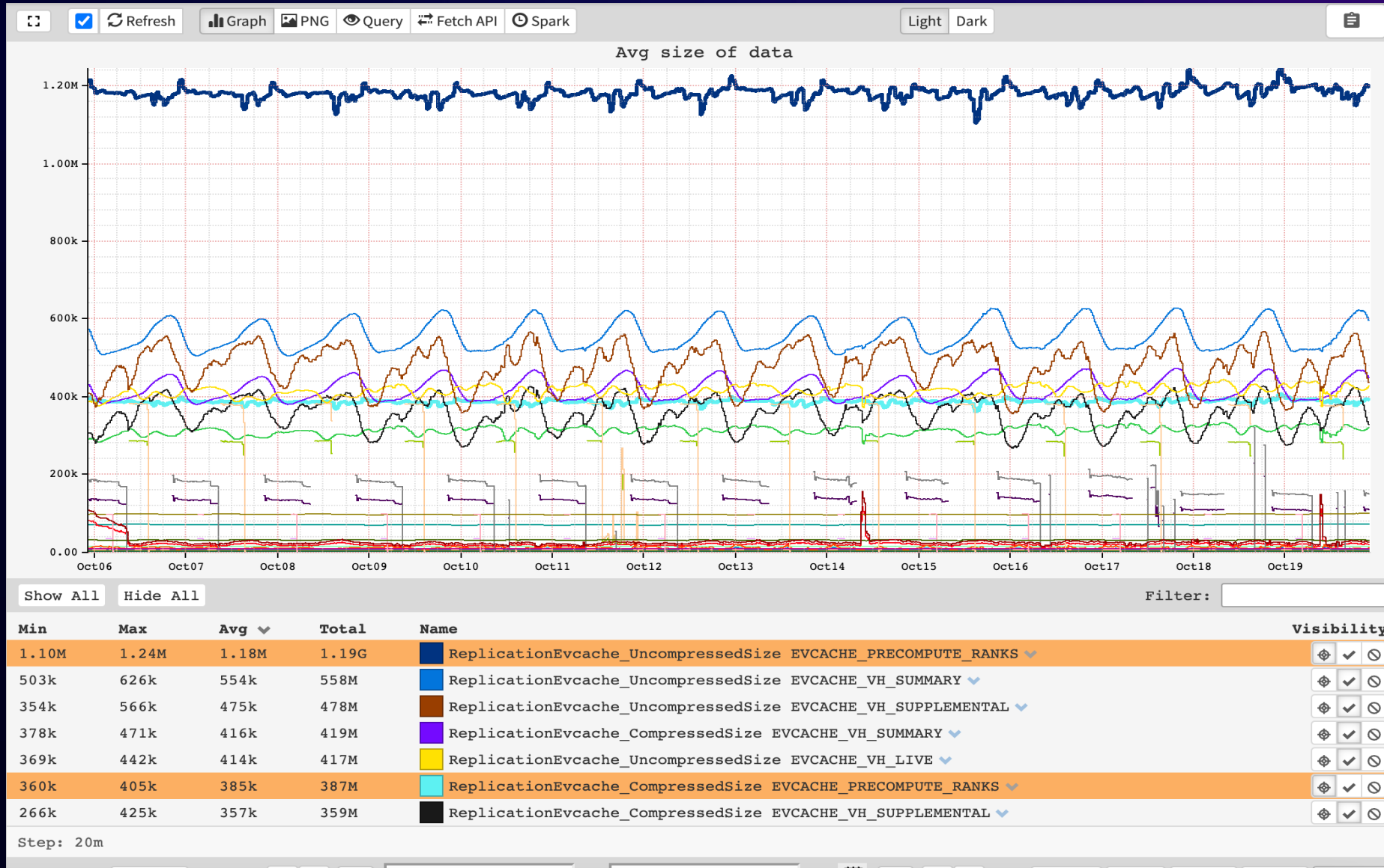
Batch compression – Across all the use cases

Size in KB

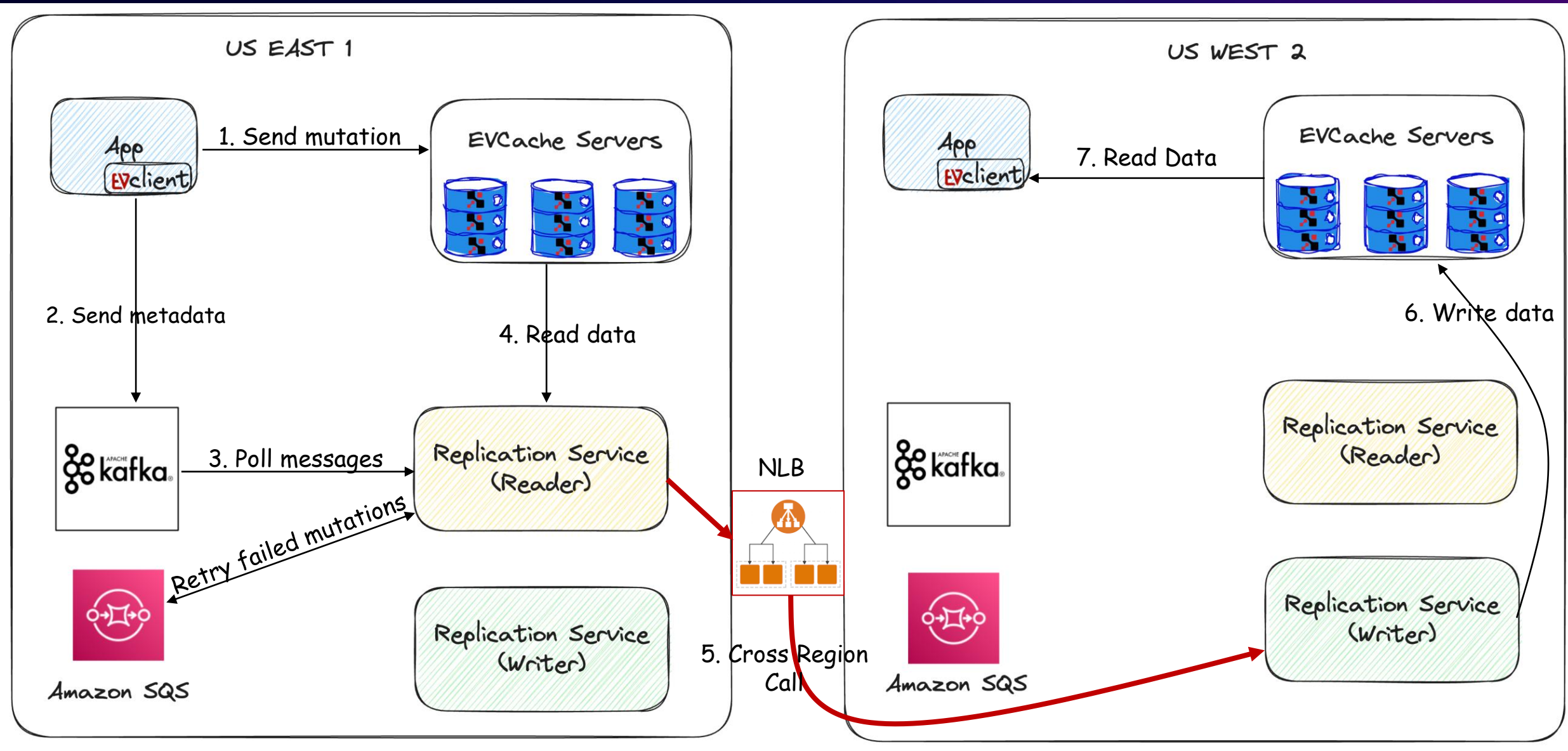


Batch compression – Per use case distribution

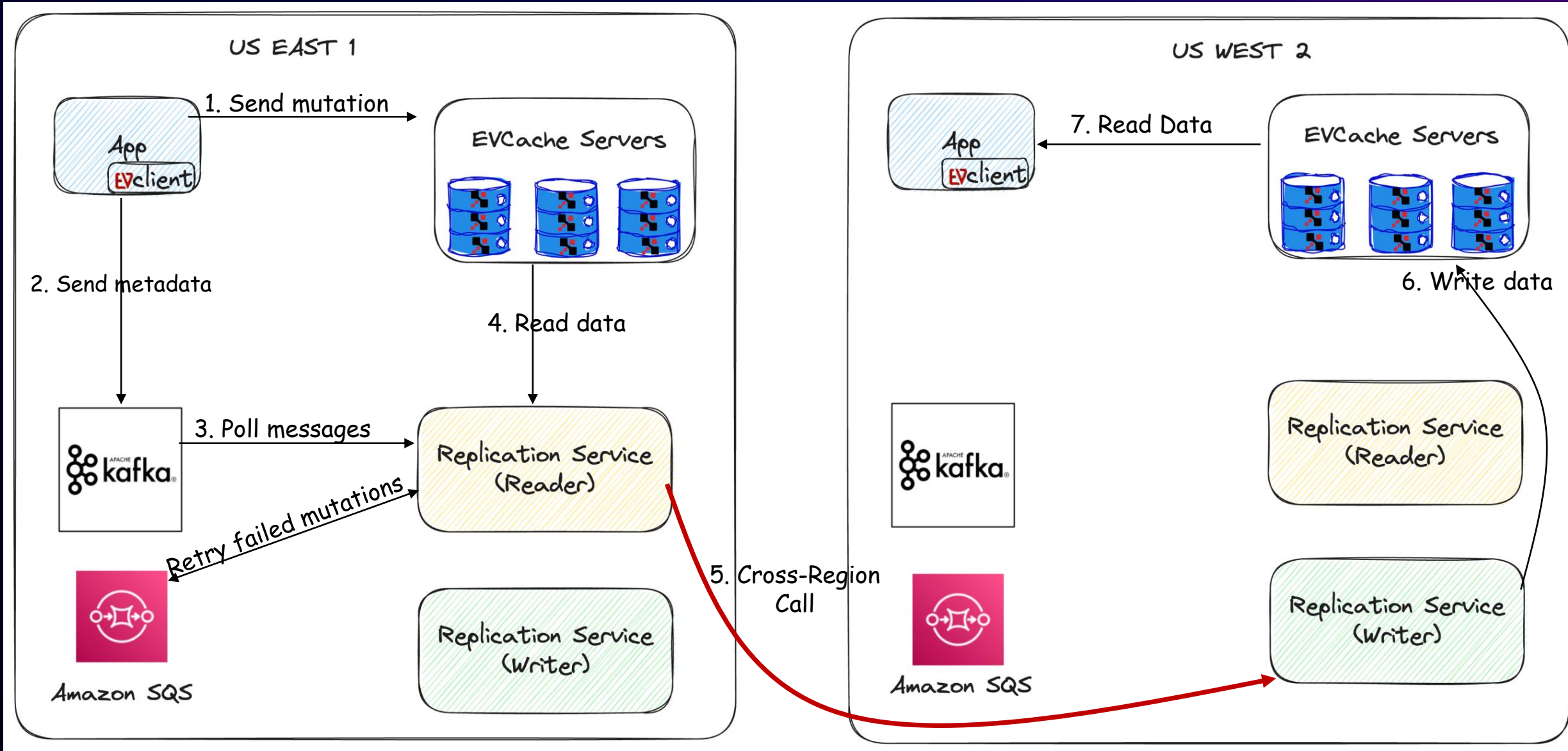
Size in KB



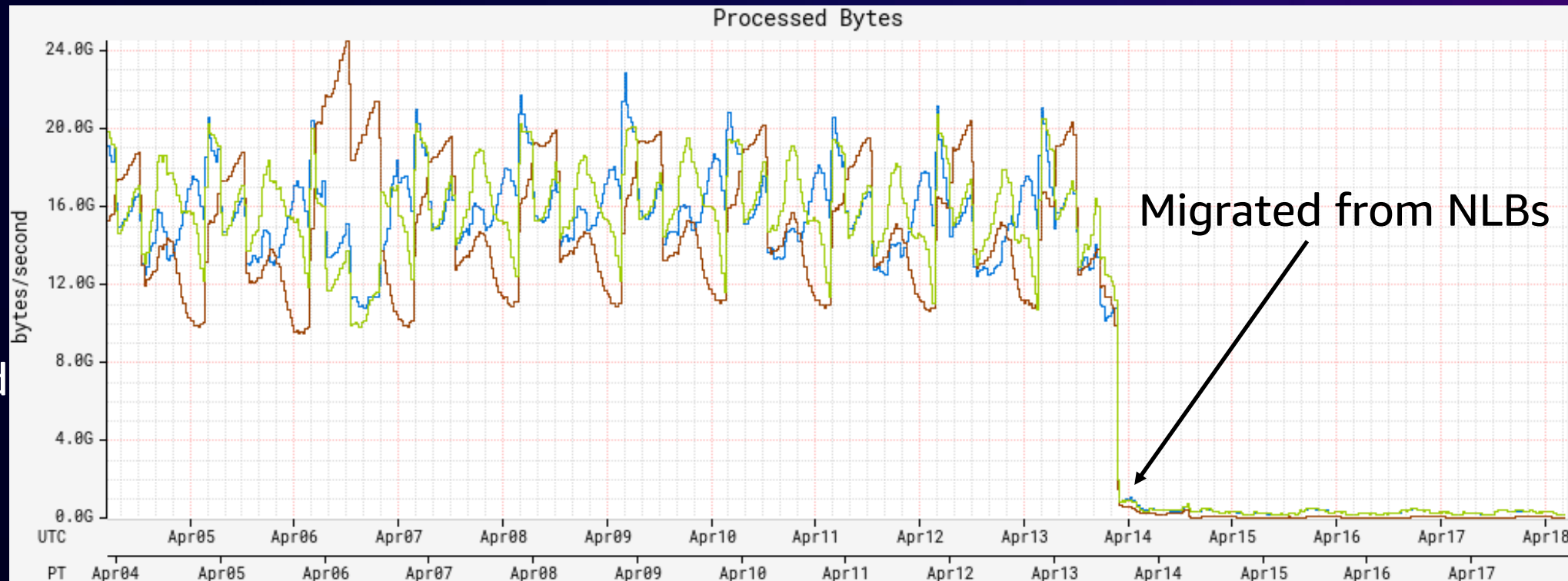
Network Load Balancer



Client Side Load Balancing



NLB Processed Bytes



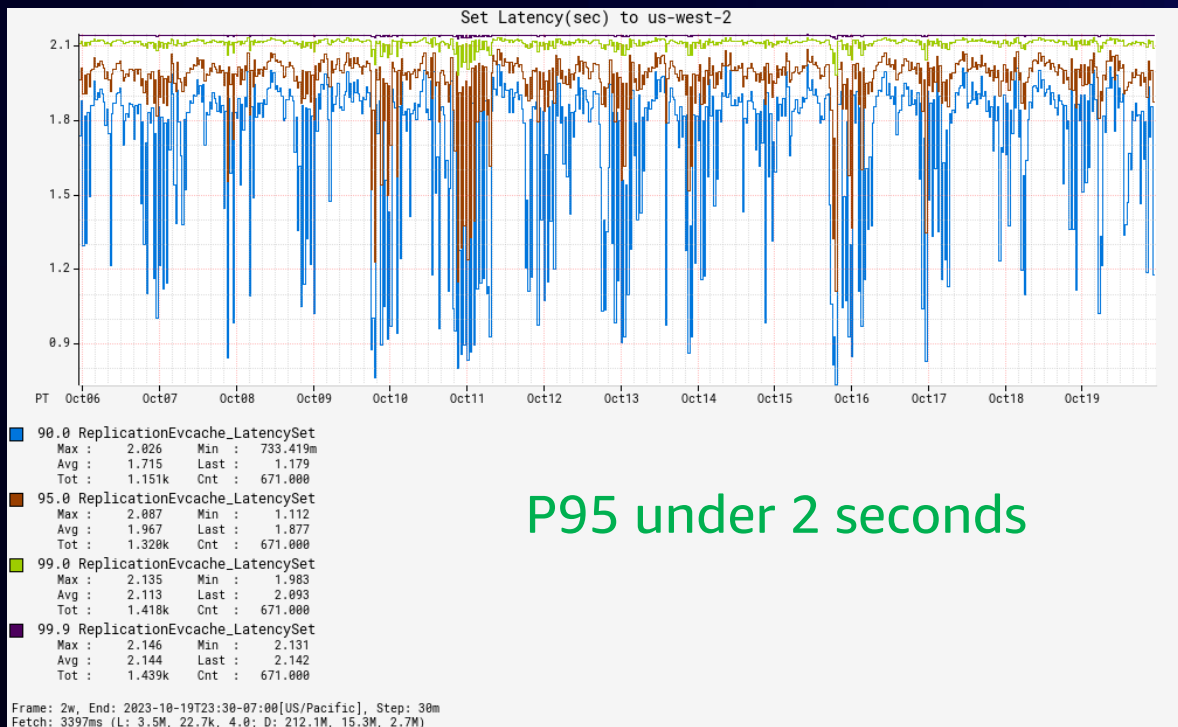
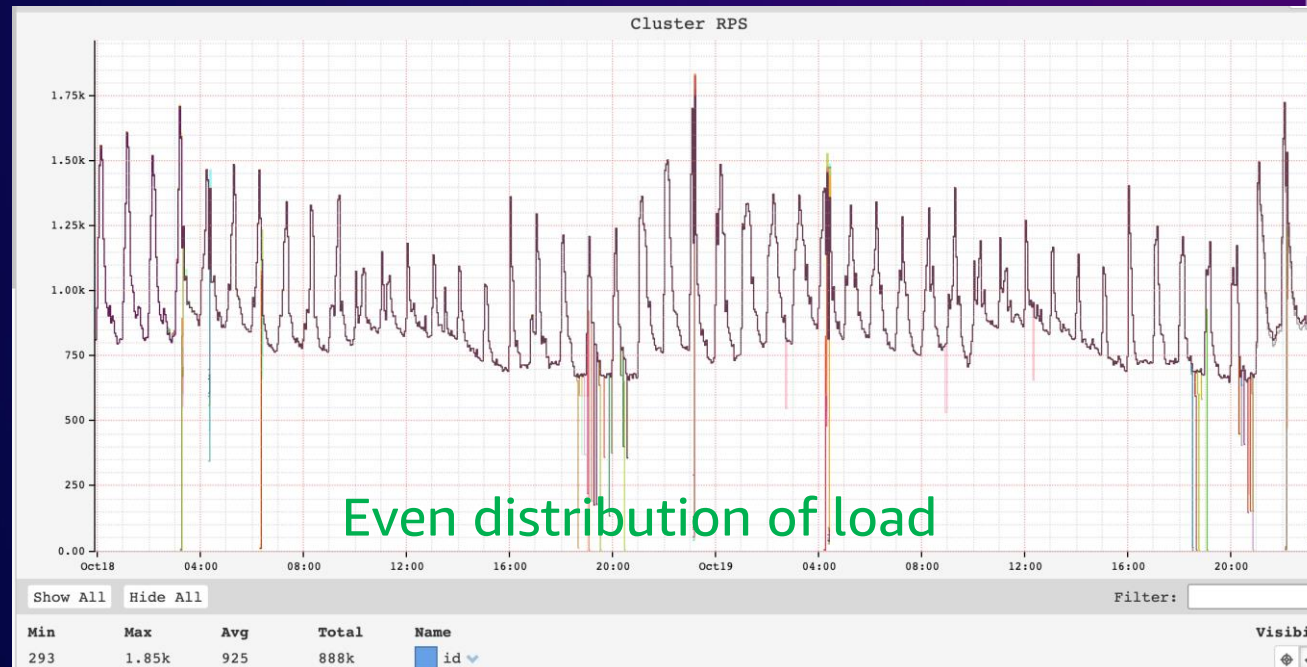
■ eu-west-1				
Max :	22.832G	Min :	173.751M	
Avg :	10.994G	Last :	184.839M	
Tot :	7.377T	Cnt :	671.000	
■ us-east-1				
Max :	24.462G	Min :	18.248M	
Avg :	10.444G	Last :	19.186M	
Tot :	7.008T	Cnt :	671.000	
■ us-west-2				
Max :	20.788G	Min :	174.743M	
Avg :	11.131G	Last :	187.263M	
Tot :	7.469T	Cnt :	671.000	

Frame: 2w, End: 2023-04-18T05:30Z[UTC], Step: 30m
Fetch: 76ms (L: 22.7k, 120.0, 3.0; D: 1.4M, 80.6k, 2.0M)

Savings of ~50% in network transfer costs

Did it work?

RPS
On
Writer Side



End to End
latencies
in seconds



Agenda

Introduction

Global Replication

Design and Architecture

Observability

Efficiency Improvements

Life of an EVCache Engineer



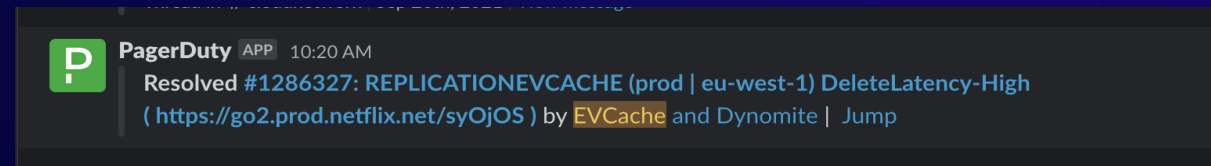
Life of an EVCache Engineer

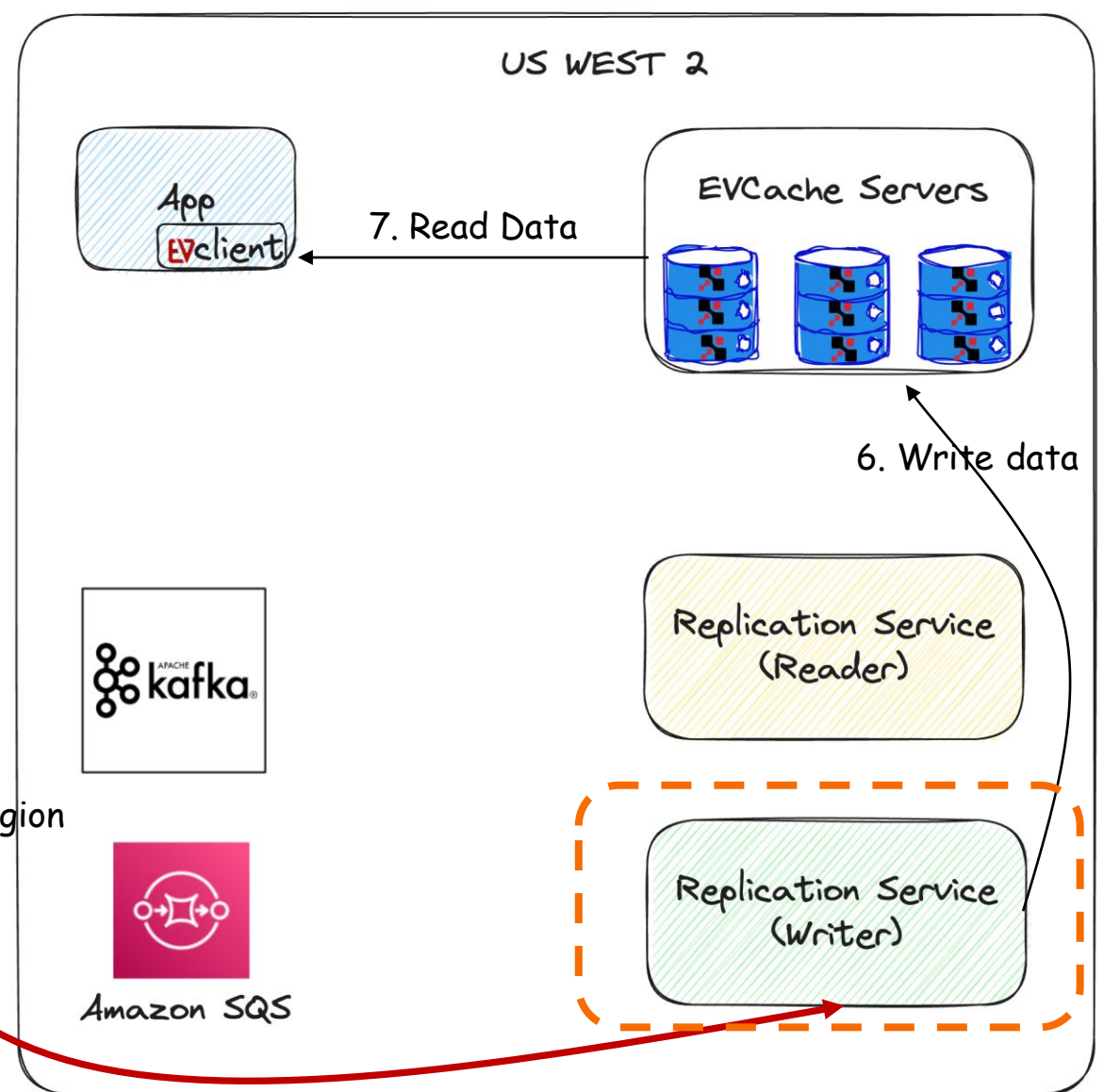
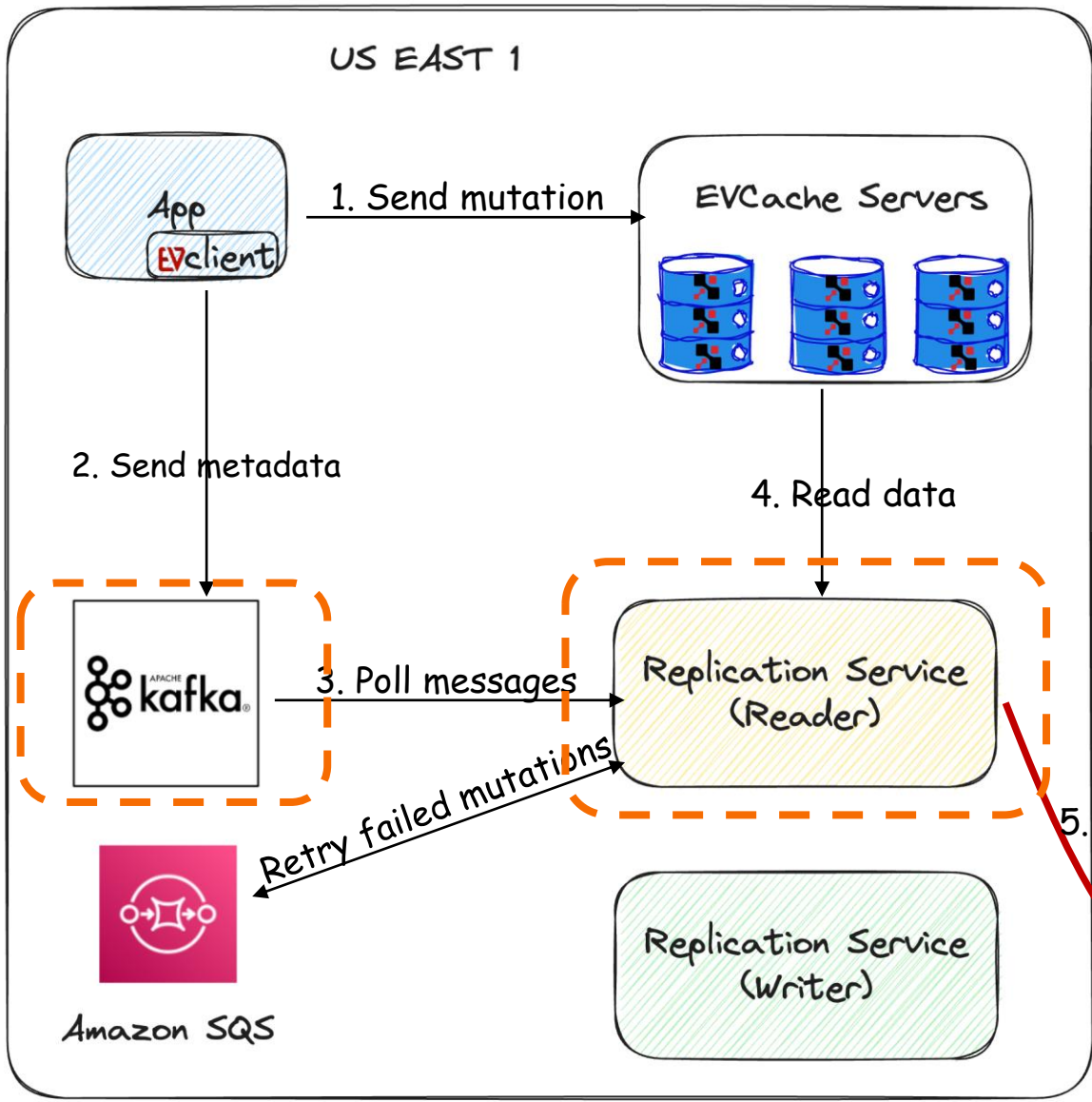
Latency Page

Instances going out of discovery

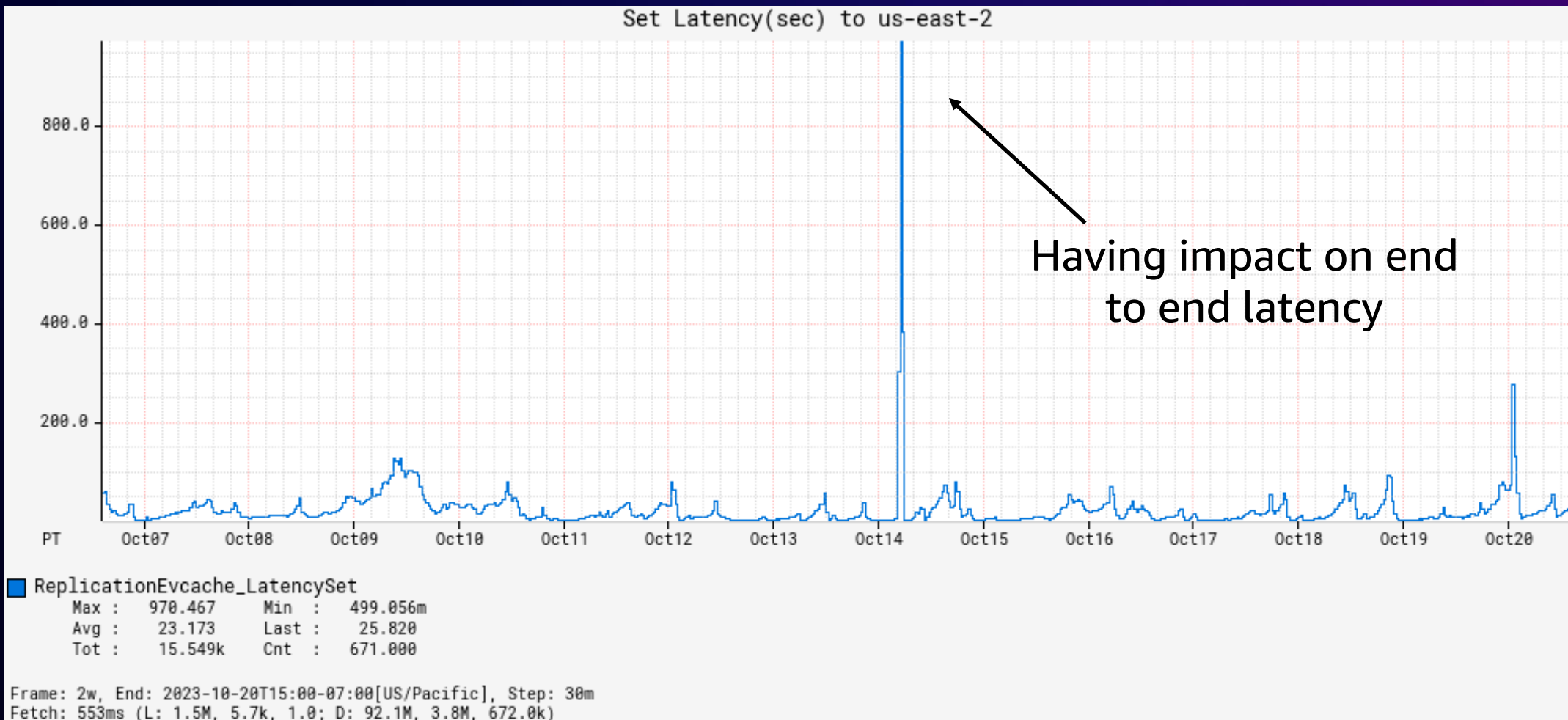
Latency Page

- Kafka
 - Consumer lag
 - # of partitions per topic
 - Join rate
- Reader
 - Bottlenecked by thread pool
 - Control message polling
- Writer
 - Slow backend

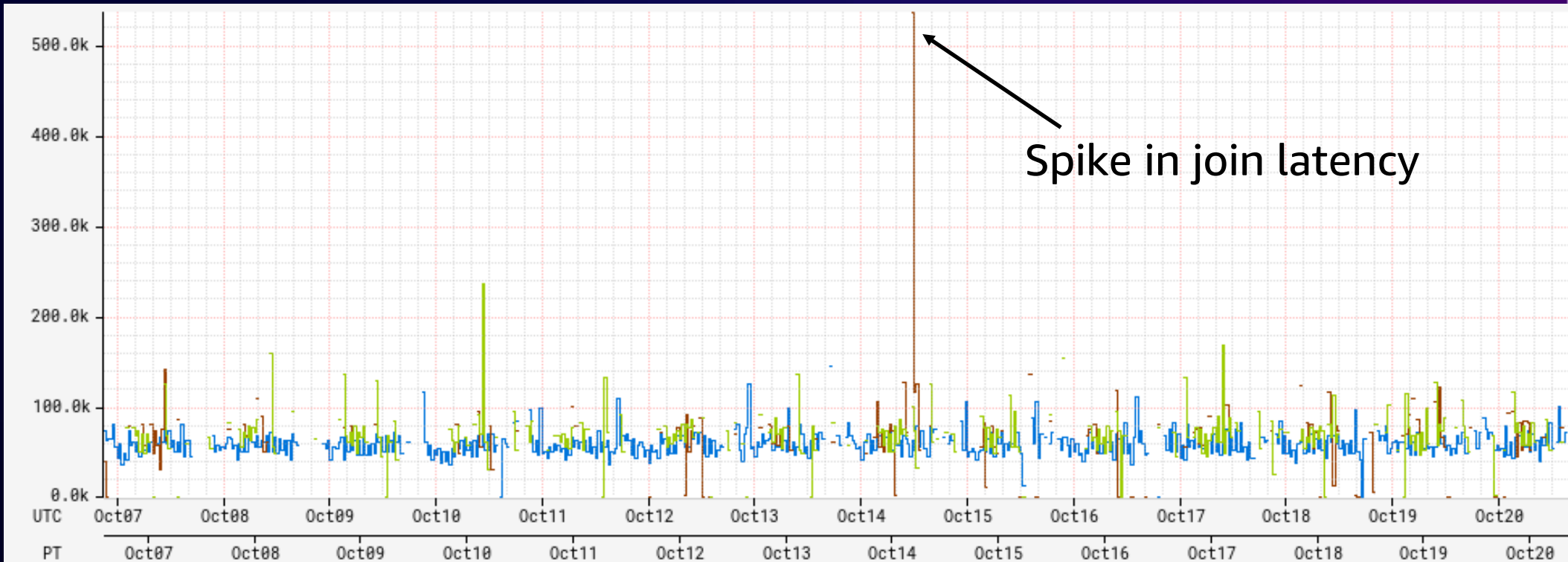




SET latency in Seconds



Join latency in milliseconds

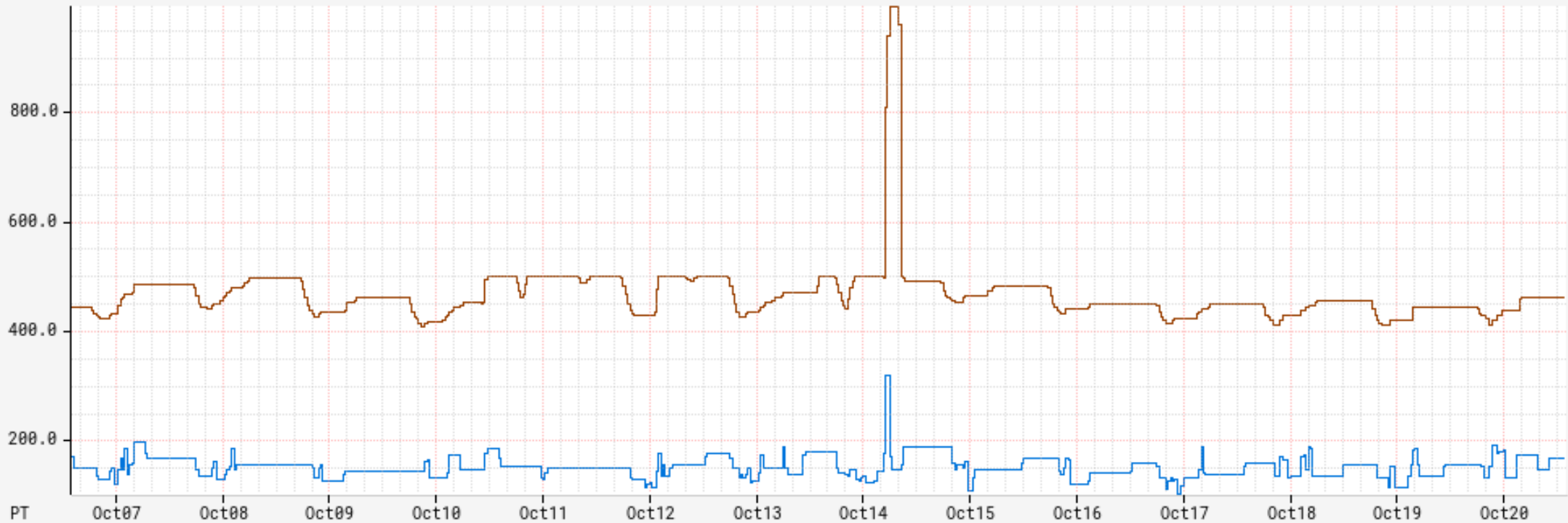


■ replicationevcache-sbnnfkafka-euwest1-mtron-v014	
Max : 145.219k	Min : 0.000
Avg : 59.841k	Last : 3.990k
Tot : 34.947M	Cnt : 584.000
■ replicationevcache-sbnnfkafka-useast2-mtron-v011	
Max : 535.946k	Min : 0.000
Avg : 68.729k	Last : 83.899k
Tot : 10.859M	Cnt : 158.000
■ replicationevcache-sbnnfkafka-uswest2-mtron-v006	
Max : 235.888k	Min : 0.000
Avg : 69.757k	Last : 38.050k
Tot : 19.811M	Cnt : 284.000

Frame: 2w, End: 2023-10-20T21:30Z[UTC], Step: 30m
Fetch: 1325ms (L: 294.9k, 5.6k, 3.0; D: 17.7M, 3.7M, 2.0M)

Surge in instances

Number Of Reader vs Writer Instances



■ us-east-1 m5.2xlarge replicationevcache-sbnnfkafka-useast2-mtron

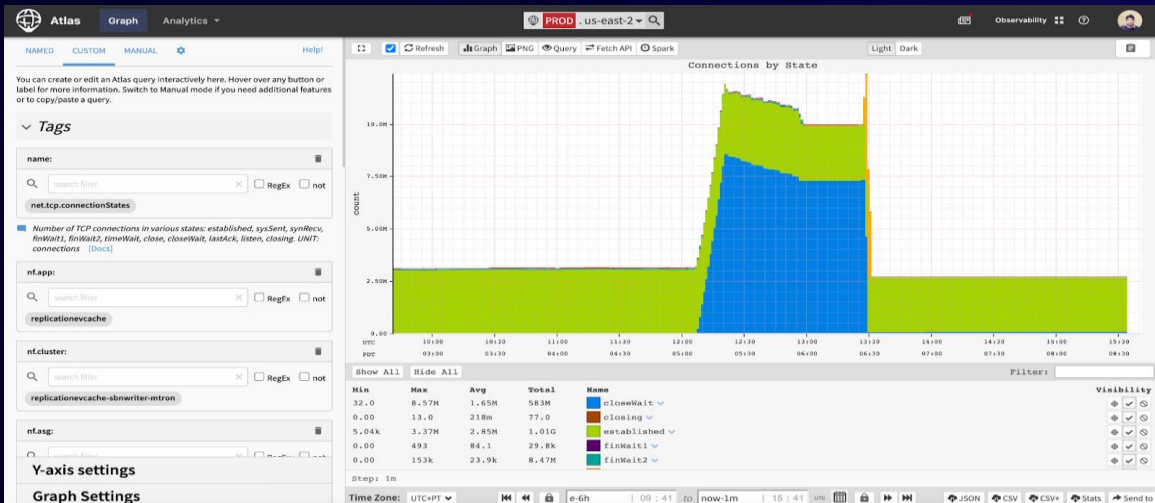
Max : 321.133 Min : 102.000
Avg : 152.484 Last : 168.000
Tot : 102.317k Cnt : 671.000

■ us-east-2 m5.2xlarge replicationevcache-sbnwriter-mtron

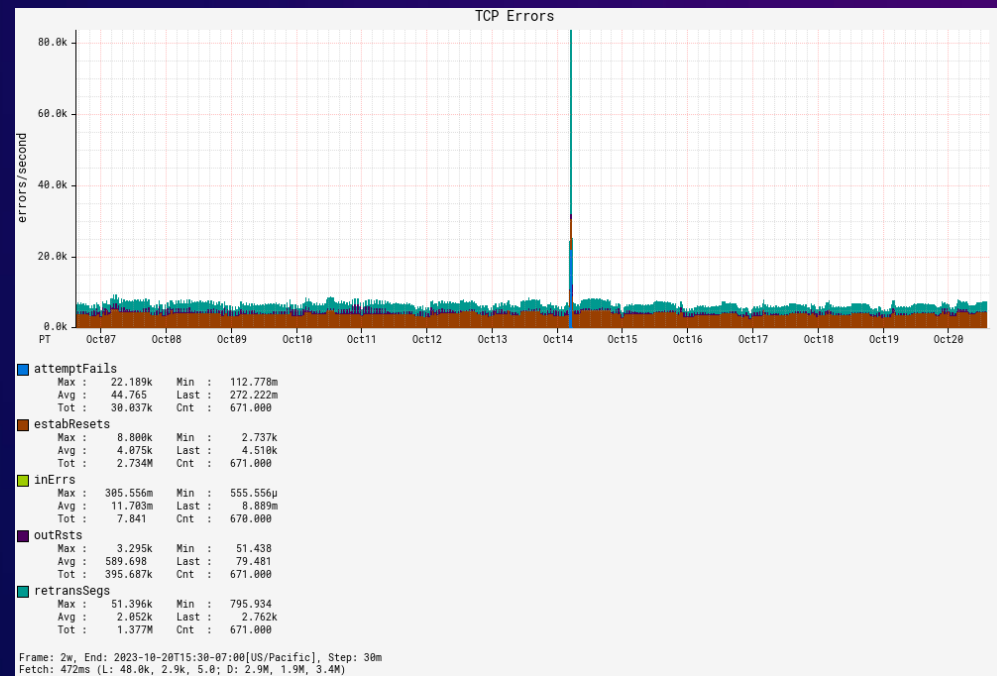
Max : 992.000 Min : 408.000
Avg : 467.062 Last : 463.000
Tot : 313.399k Cnt : 671.000

Frame: 2w, End: 2023-10-20T14:30-07:00[US/Pacific], Step: 30m
Fetch: 281ms (L: 44.9k, 1.2k, 2.0; D: 2.7M, 772.8k, 1.3M)





TCP Errors Writer



TCP Errors Reader

```

tcp6 6017 0 10.168.214.104:7004 52.71.231.25:13048 CLOSE_WAIT
tcp6 6016 0 10.168.214.104:7004 34.220.255.118:19838 CLOSE_WAIT
tcp6 6017 0 10.168.214.104:7004 34.222.159.118:11690 CLOSE_WAIT
tcp6 6015 0 10.168.214.104:7004 54.229.224.111:27374 CLOSE_WAIT
tcp6 6015 0 10.168.214.104:7004 52.12.163.46:51954 CLOSE_WAIT
tcp6 289 0 10.168.214.104:7004 3.253.46.21:17584 CLOSE_WAIT
tcp6 6015 0 10.168.214.104:7004 54.77.132.0:52174 CLOSE_WAIT
tcp6 6017 0 10.168.214.104:7004 3.249.35.102:33114 CLOSE_WAIT
tcp6 6016 0 10.168.214.104:7004 54.78.216.110:17370 CLOSE_WAIT
tcp6 6015 0 10.168.214.104:7004 35.165.82.30:42180 CLOSE_WAIT
tcp6 6016 0 10.168.214.104:7004 3.81.224.186:16600 CLOSE_WAIT
tcp6 6015 0 10.168.214.104:7004 35.175.244.246:26752 CLOSE_WAIT
tcp6 6016 0 10.168.214.104:7004 54.152.147.6:62618 CLOSE_WAIT
tcp6 6015 0 10.168.214.104:7004 44.210.123.30:63946 CLOSE_WAIT
tcp6 6016 0 10.168.214.104:7004 34.242.221.23:17468 CLOSE_WAIT
tcp6 6015 0 10.168.214.104:7004 54.224.241.199:38580 CLOSE_WAIT
tcp6 6016 0 10.168.214.104:7004 3.249.123.94:56310 CLOSE_WAIT
tcp6 289 0 10.168.214.104:7004 3.83.41.125:14794 CLOSE_WAIT
tcp6 6016 0 10.168.214.104:7004 35.88.109.167:19962 CLOSE_WAIT

prod replicationevcache-sbnwriter-mtron-v008 us-east-2 i-05d6b6ad38d419498
(nfsuper) ~ $ netstat -ltan | grep 7004 | wc -l
94149

prod replicationevcache-sbnwriter-mtron-v008 us-east-2 i-05d6b6ad38d419498
(nfsuper) ~ $
  
```

```

2023-10-14 12:19:27.679 ERROR 4659 --- [replication-369] c.n.e.r.r.RemoteHttpRequestEventListener : Connection Failed to : /18.191.27.128:7004; Protocol : null
java.net.SocketTimeoutException: Read timed out
    at java.base/sun.nio.ch.NioSocketImpl.timedRead(NioSocketImpl.java:288)
    at java.base/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:314)
    at java.base/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:355)
    at java.base/sun.nio.ch.NioSocketImpl$.read(NioSocketImpl.java:808)
    at java.base/java.net.Socket$SocketInputStream.read(Socket.java:976)
    at java.base/sun.security.ssl.SSLSocketInputRecord.read(SSLSocketInputRecord.java:484)
    at java.base/sun.security.ssl.SSLSocketInputRecord.readHeader(SSLSocketInputRecord.java:478)
    at java.base/sun.security.ssl.SSLSocketInputRecord.decode(SSLSocketInputRecord.java:160)
    at java.base/sun.security.ssl.SSLTransport.decode(SSLTransport.java:111)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1510)
    at java.base/sun.security.ssl.SSLSocketImpl.readHandshakeRecord(SSLSocketImpl.java:1425)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:455)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:426)
    at com.netflix.metatron.ipc.security.DelegatingSSLSocket.startHandshake(DelegatingSSLSocket.java:88)
    at okhttp3.internal.connection.RealConnection.connectTls(RealConnection.kt:379)
    at okhttp3.internal.connection.RealConnection.establishProtocol(RealConnection.kt:337)
    at okhttp3.internal.connection.RealConnection.connect(RealConnection.kt:209)
    at okhttp3.internal.connection.ExchangeFinder.findConnection(ExchangeFinder.kt:226)
  
```



Instances going out of discovery

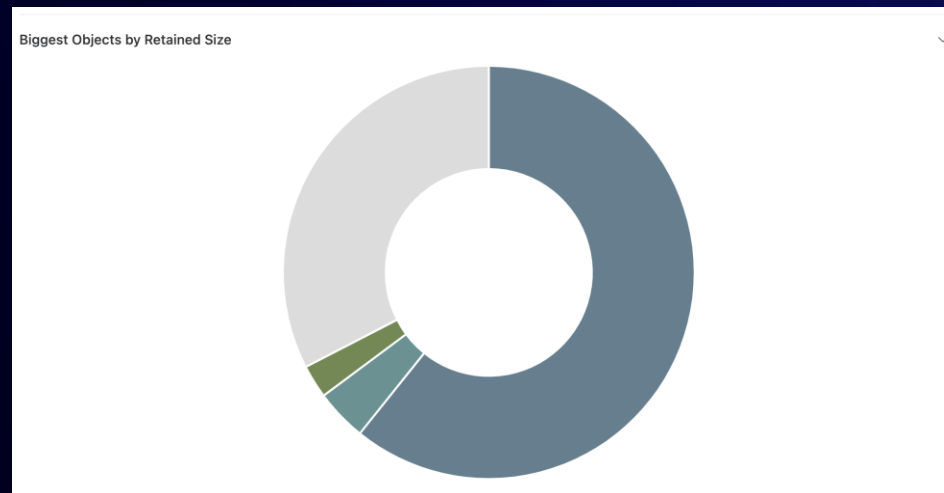
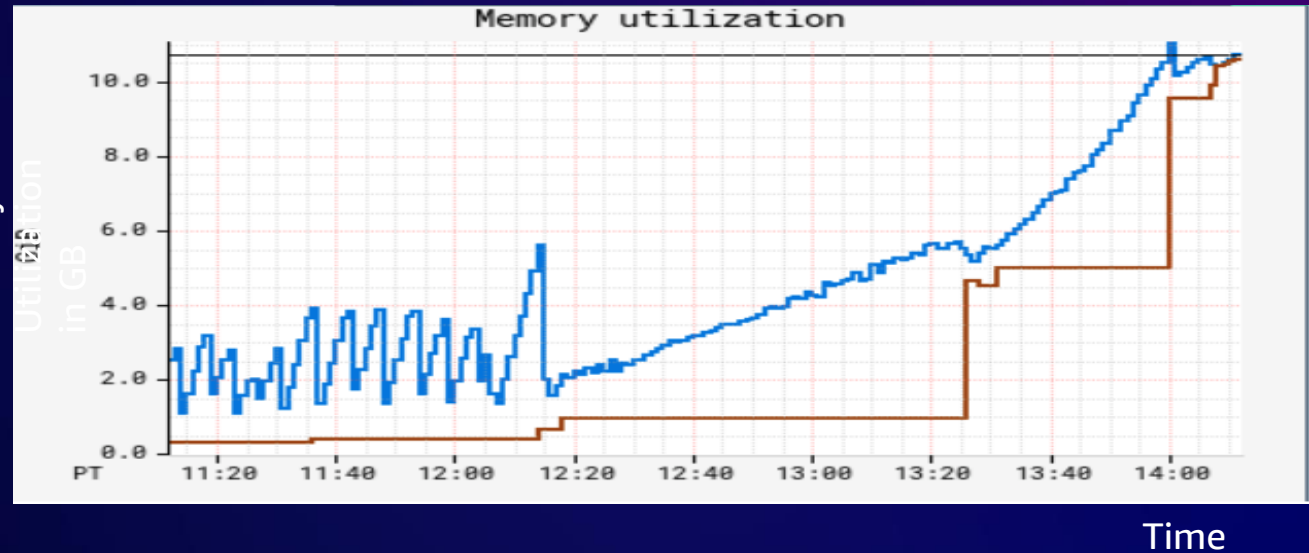
 V037: Build: #41 



~50% of instances going out of discovery

```
2023-11-01 22:41:07.268 ERROR 3989 --- [tor-scheduled-2] c.n.e.p.EVCacheClientPool : Exception while
refreshing the Server List
java.lang.OutOfMemoryError: Java heap space
2023-11-01 22:41:25.951 WARN 3989 --- [nfoReplicator-0] c.n.s.c.ArchaiusPollerHealthIndicator : Archaius poller failed
to load remote layer for 520 cycles
2023-11-01 22:41:27.250 WARN 3989 --- [nfoReplicator-0] c.n.d.InstanceInfoReplicator : There was a problem with
the instance info replicator
java.lang.OutOfMemoryError: Java heap space
2023-11-01 22:41:57.890 WARN 3989 --- [nfoReplicator-0] c.n.s.c.ArchaiusPollerHealthIndicator : Archaius poller failed
to load remote layer for 521 cycles
```

Out of memory (OOM)



Heap Dump

Class Name	Shallow Heap	Retained Heap	Percentage
> com.netflix.util.concurrent.NFExecutorPool @ 0x54aface40	88B	1.38G	15.14%
> com.netflix.util.concurrent.NFExecutorPool @ 0x54afad170	88B	1.27G	13.96%
> com.netflix.util.concurrent.NFExecutorPool @ 0x54ae73510	88B	1.27G	13.93%
> com.netflix.util.concurrent.NFExecutorPool @ 0x54afad4a0	88B	1.23G	13.54%
> com.netflix.util.concurrent.NFExecutorPool @ 0x54afac6c8	88B	1.19G	13.04%

700K items in some of the queues, and items are ~KBs in size, so that means ~GBs of memory taken per queue

at java.util.concurrent.ArrayBlockingQueue.put(Ljava/lang/Object;)V (ArrayBlockingQueue.java:353)
> <local> java.util.concurrent.ArrayBlockingQueue @ 0x53afa70f0
> <class> class java.util.concurrent.ArrayBlockingQueue @ 0x54aba7570 System Class
> items java.lang.Object[250] @ 0x53afa7120
> <class> class java.lang.Object[] @ 0x539038b68
> [189] com.netflix.evcache.evcommon.replmessage.ReplicationMessage @ 0x54b262000
> <class> class com.netflix.evcache.evcommon.replmessage.ReplicationMessage @ 0x53b0e4808
> eventType com.netflix.evcache.EVCache\$Call @ 0x54ac74e80 SET
> topic java.lang.String @ 0x54b261fa8 evcache_precompute_ranks
> ext java.lang.Object @ 0x54b262040
> key java.lang.String @ 0x54b262050 pvr:data_198372525195606295_2cf8f8386a712312a59c170da
> length java.lang.Integer @ 0x54b262100 63653
> checksum java.lang.String @ 0x54b262110 3765088820

Atlas **Graph** Analytics PROD .global Observability

NAMED CUSTOM MANUAL Help!

You can create or edit an Atlas query interactively here. Hover over any button or label for more information. Switch to Manual mode if you need additional features or to copy/paste a query.

Tags: name: search filter RegEx not
 ReplicationEvcache_LatencySet

Metrics Glossary entry not found for metric 'ReplicationEvcache_LatencySet'. If you'd like to contribute a glossary entry please follow the [Contributing Guide](#)

nf.cluster: replicationevcache-dup-uswest2.* RegEx not

nf.asg: search filter RegEx not
 replicationevcache-dup-uswest2-v033

nf.app:

Y-axis settings

Graph meet.google.com is sharing your screen.

Refresh Graph PNG Query Fetch API Spark

Set Latency(max sec) to us-west-2

Filter:

Min	Max	Avg	Total	Name	Visibility
0.00	992	288	49.3k	dup evcache_ocp_flink_am_nc	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
0.00	528	66.5	11.5k	dup evcache_search_models	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
0.00	59.1	7.12	1.07k	dup evcache_blacklight	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
0.00	45.6	6.33	1.05k	dup evcache_vayu	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
1.02	34.6	9.27	1.59k	dup evcache_pbc	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
0.00	30.2	2.17	374	dup evcache_obelix	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
0.00	24.2	5.41	926	dup evcache_ccs	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
0.00	18.6	1.78	308	dup evcache_geooverrides	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

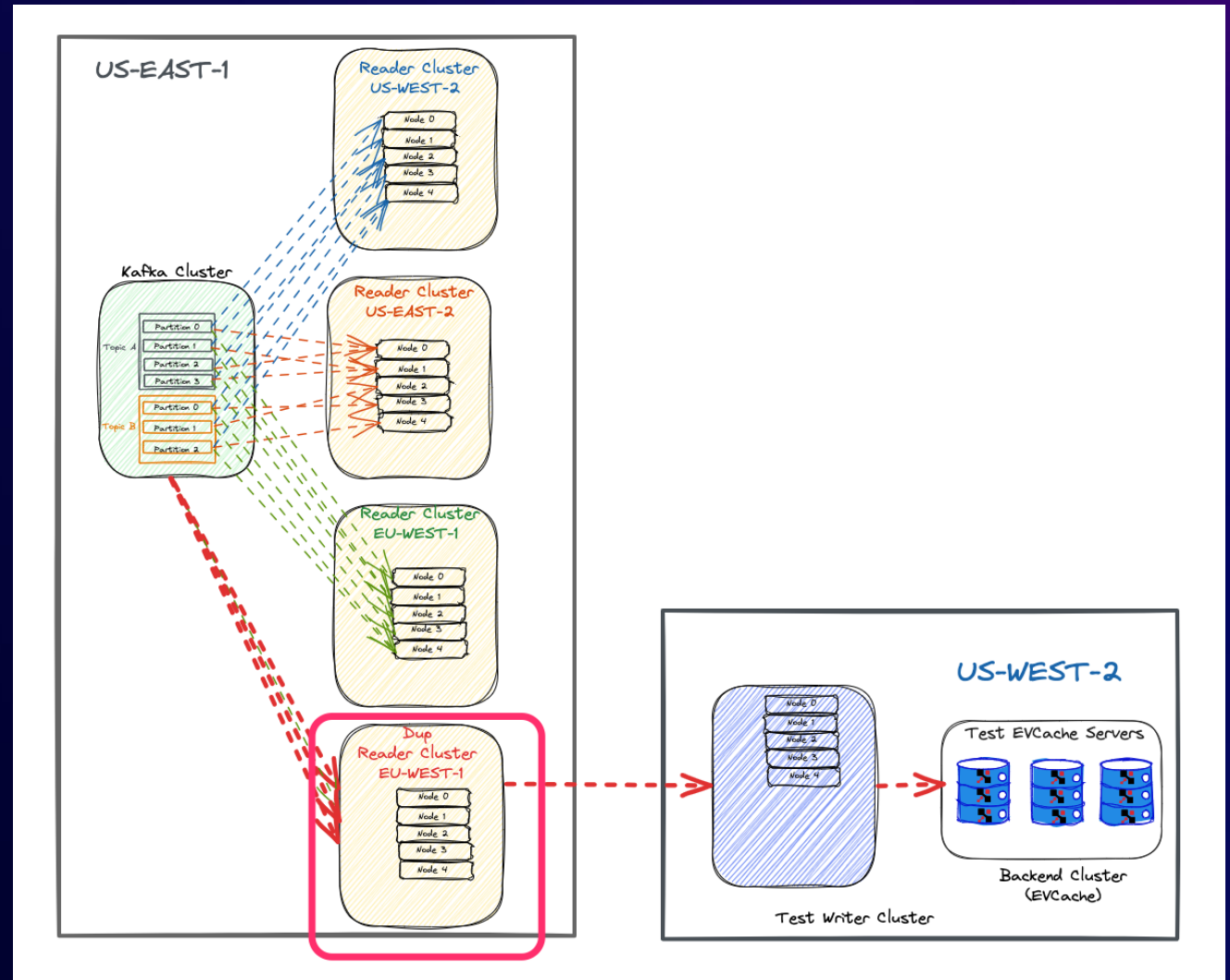
Step: 1m

Time Zone: Pacific e-3h | 11:14 to now-5m | 14:20 PST

Apply backpressure

Shadow Testing

- Route X % of traffic via staging cluster
- Build confidence, faster development cycle



Agenda

Introduction

Global Replication

Design and Architecture

Observability

Efficiency Improvements

Life of an EVCache Engineer

Future



Future



IPv6



Container Migration



Generic Service

Thank you!



Please complete the session survey in the mobile app

Prudhviraj Karumanchi

 <https://www.linkedin.com/in/prudhviraj9/>

Sriram Rangarajan

 <https://www.linkedin.com/in/sriram-rangarajan-35169715/>

Previous Re:Invent Talk - AWS re:Invent 2021 - How Netflix operates mission-critical data stores on AWS

<https://www.youtube.com/watch?v=MtGYgilaPgo>

