

AEROSPIKE

SUMMIT '19



# Hurdling Operational Scenarios at the Nielsen Marketing Cloud

Henry Snow  
VP, Cloud Operations

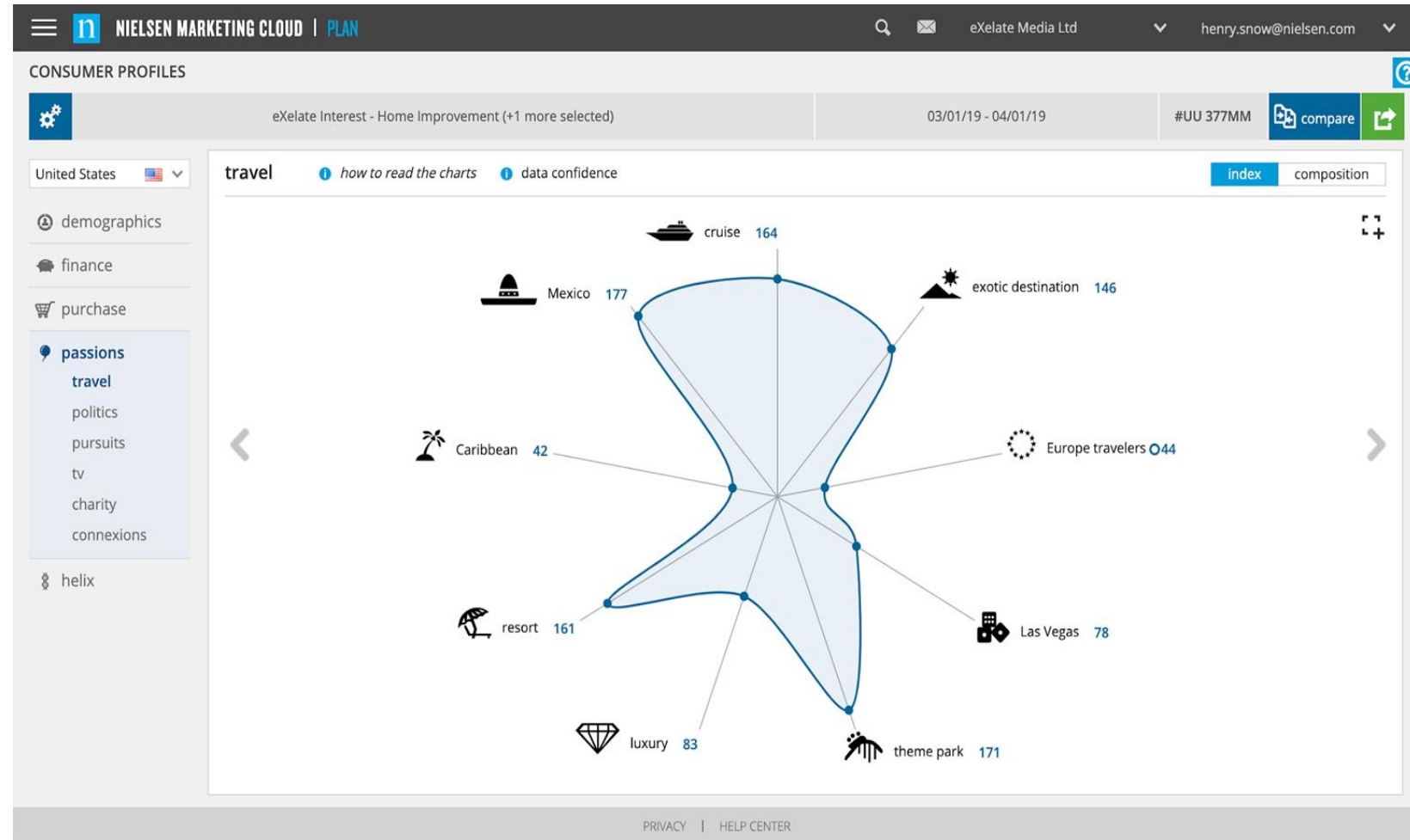
# Agenda

- Nielsen Marketing Cloud Intro
- Our Tech Stack
- Aerospike Architecture
- Configuration Deployment & Monitoring
- Notable Hurdles
- Questions

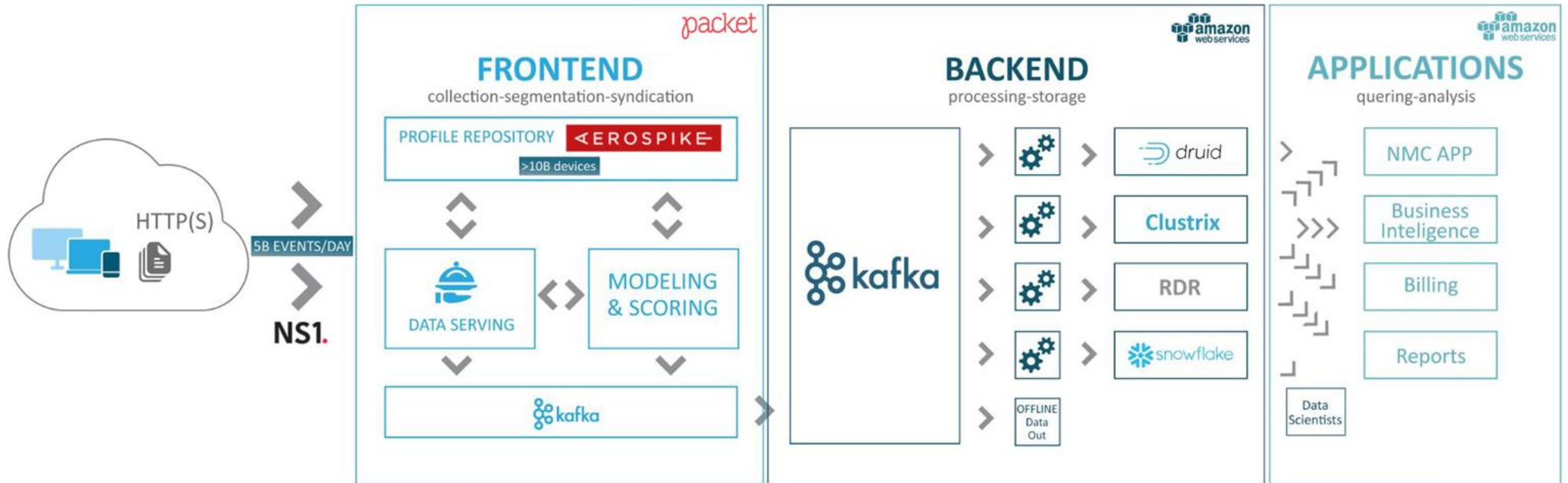


# What is the Nielsen Marketing Cloud

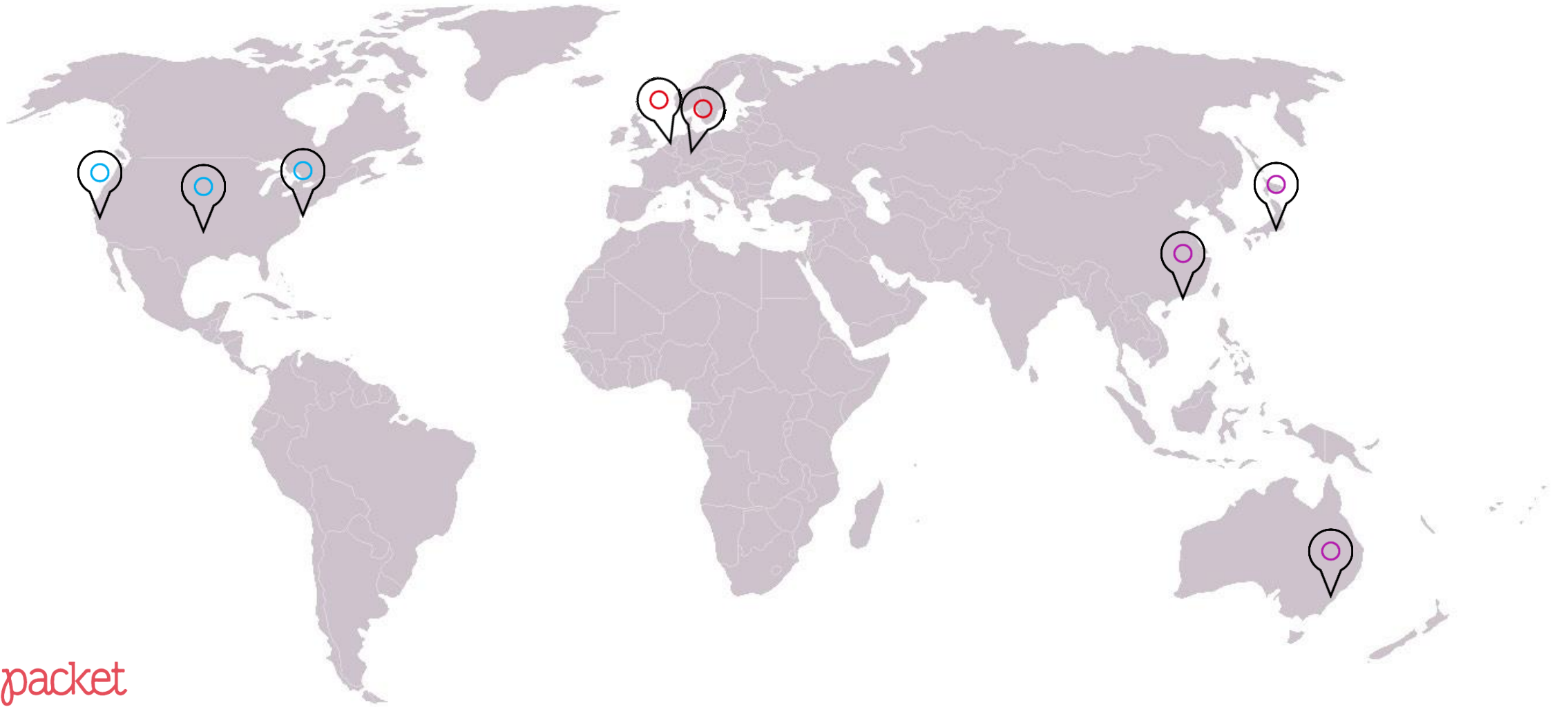
- Born from eXelate
- Digital data group within Nielsen
- Data Enrichment
- Insights
- Activation
- Targeting



# NMC High-Level Architecture



# Data Ingestion Facilities



packet



# Data Ingestion Infrastructure

## eXpresso

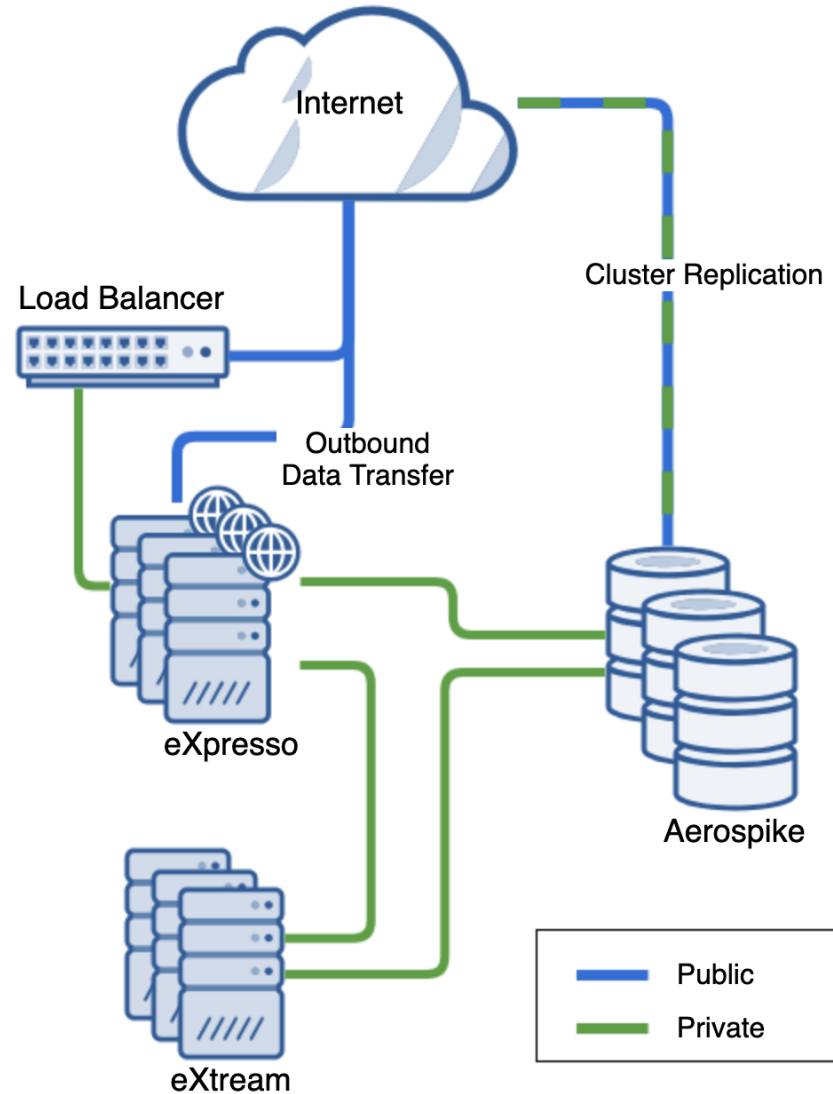
- Web App Server
- Online / Offline Data Ingestion

## eXtream

- Real-Time Modeling Engine
- Online Learning

## Aerospike

- Device History Storage
- Cross-Datacenter Replication



# What We Store in Aerospike

- **Device User ID Objects**
  - General Variables
    - Device Type
    - Geolocation
    - Timezone
  - Segment History
  - Delivery History
  - Household IDs
- **ID Mappings**
  - 3rd Party IDs to User IDs
  - Hashed IPs to User ID
  - Households to User IDs
- **Object Storage**
  - Single Bin namespaces
  - zstandard compression
  - Blob types
  - Maps



# Aerospike by the Numbers

## US Clusters:

- 12 nodes
- 140TB used storage
- 14B objects
- 100k read TPS
- 75k write TPS
- 15Gbps XDR throughput

## EU/APAC Clusters:

- 4 nodes
- 10TB used storage
- 3.5B objects
- 25k read TPS
- 15k write TPS
- 2Gbps XDR throughput





# Configuration Deployment

- **Ansible**
  - Playbooks or Ad-Hoc commands
  - Idempotent Modules
  - Rolling Updates
  - Group Variables
  - Tags
  - Jinja2 Templating
- **Common Tasks**
  - Building New Server
  - Aerospike Version Upgrade
  - Static Configuration Change

```
### roles/aerospike/tasks/main.yml
```

```
- name: template configuration file
  template:
    src: aerospike.conf.j2
    dest: /etc/aerospike/aerospike.conf
  tags:
    - template_config
```

```
- name: test if aerospike is running
  shell: 'pgrep asd || true'
  register: asd_pid
  tags:
    - aerospike_upgrade
    - aerospike_restart
```

```
- name: include service restart playbook
  import_tasks: restart.nodes.yml
  when: (asd_pid.stdout and asd_version.changed) or
        "aerospike_restart" in ansible_run_tags
  tags:
    - aerospike_restart
    - aerospike_upgrade
```



# Monitoring Aerospike

- **Prometheus**

- Time-series, dimensional data
- Queryable via PromQL
- Ad-hoc graphs
- Independent Clusters (per region)
- Alert management

- **Prometheus Exporters**

- `node_exporter`
  - system metrics
- `aerospike_exporter`
  - namespace statistics
  - node statistics
  - cluster latency/tps metrics

```
# TYPE aerospike_latency_write gauge
aerospike_latency_write{namespace="lookup_eu",threshold=">1ms"} 0.33
aerospike_latency_write{namespace="lookup_eu",threshold=">2ms"} 0.31
aerospike_latency_write{namespace="lookup_eu",threshold=">8ms"} 0
```

```
# TYPE aerospike_node_client_connections gauge
aerospike_node_client_connections 876
```

```
# TYPE aerospike_node_cluster_size gauge
aerospike_node_cluster_size 4
```



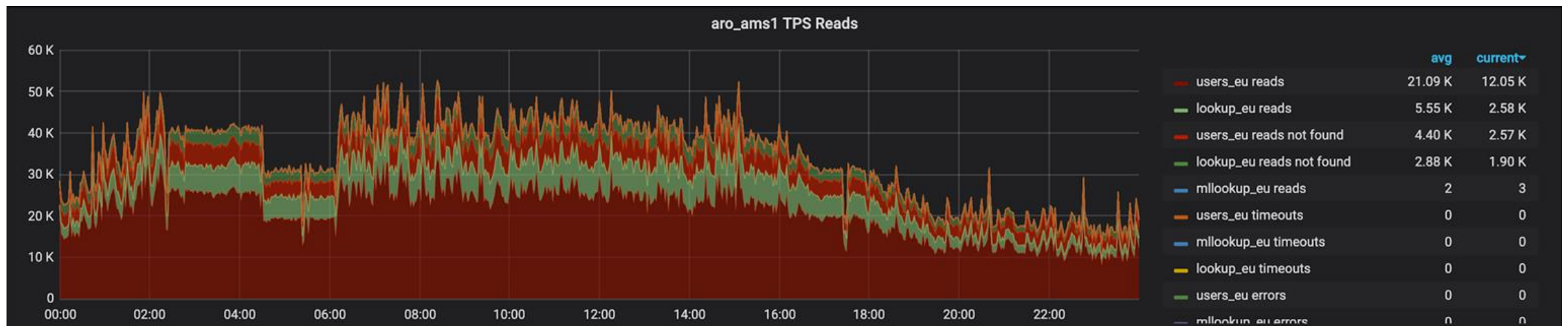
# Monitoring Aerospike

- **Thanos**

- Global query view for metrics
- Prometheus API-compatible
- Unlimited retention via S3/Object storage
- Metric Downsampling

- **Grafana**

- Customizable Dashboards
- Multiple Datasources
- Alert Management
- Annotations



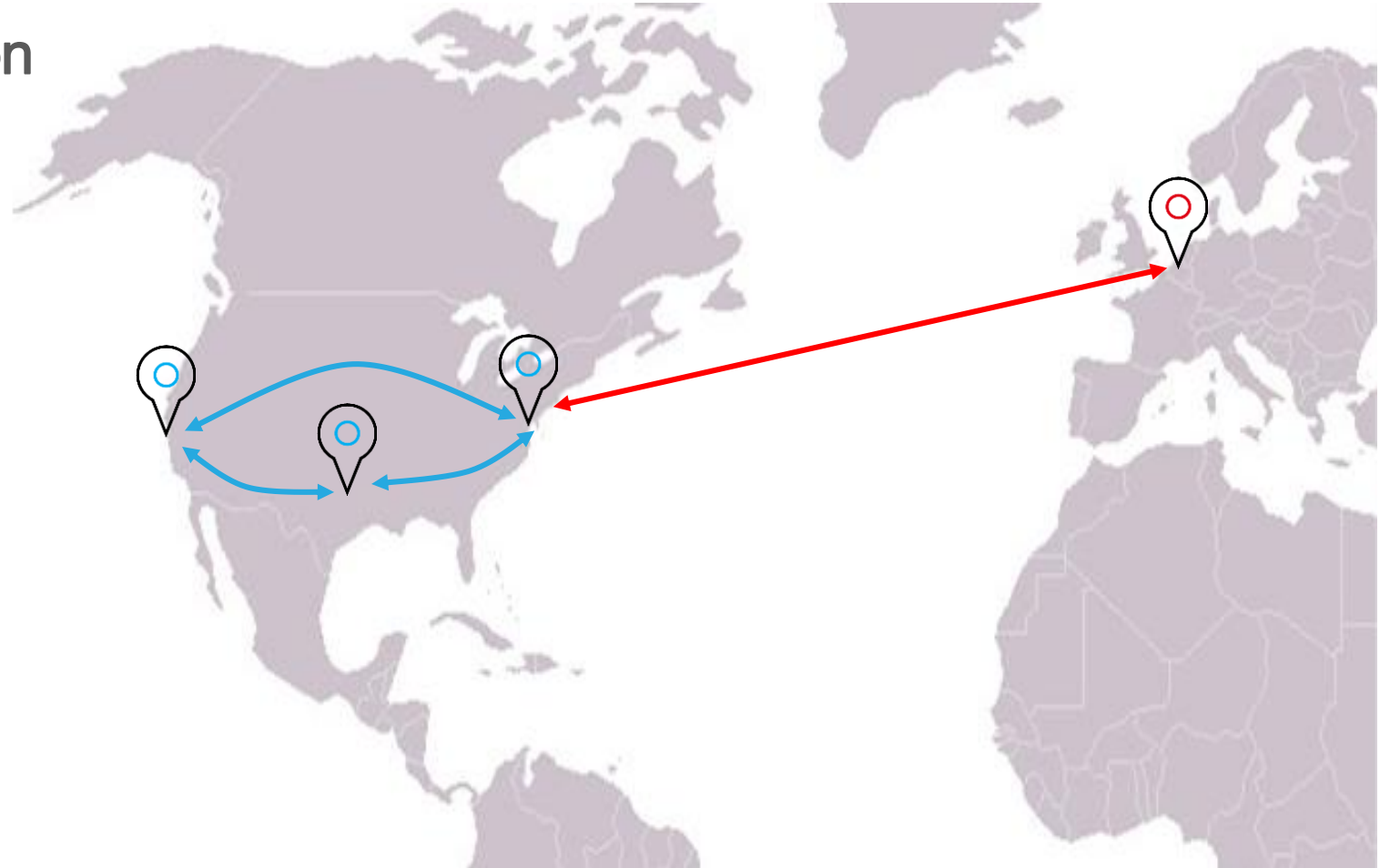
## Second EU Facility 2017



# Second EU Facility

## XDR Namespace Replication

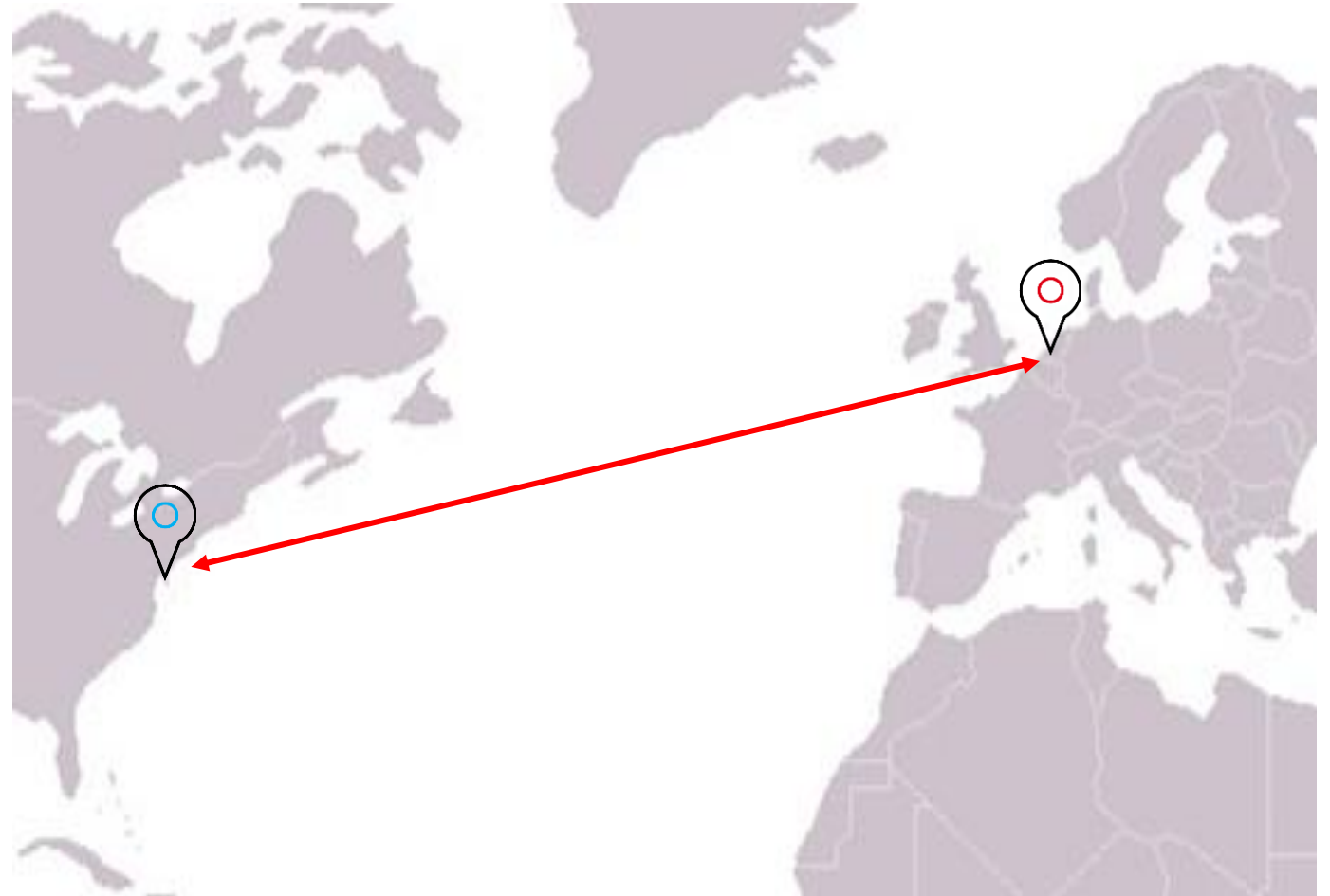
- users\_us (NJ, TX, CA)
- lookup (NJ, TX, CA)
- users\_eu (AM, NJ)
- lookup\_eu (AM, NJ)



# Second EU Facility

## NJ as EU Failover

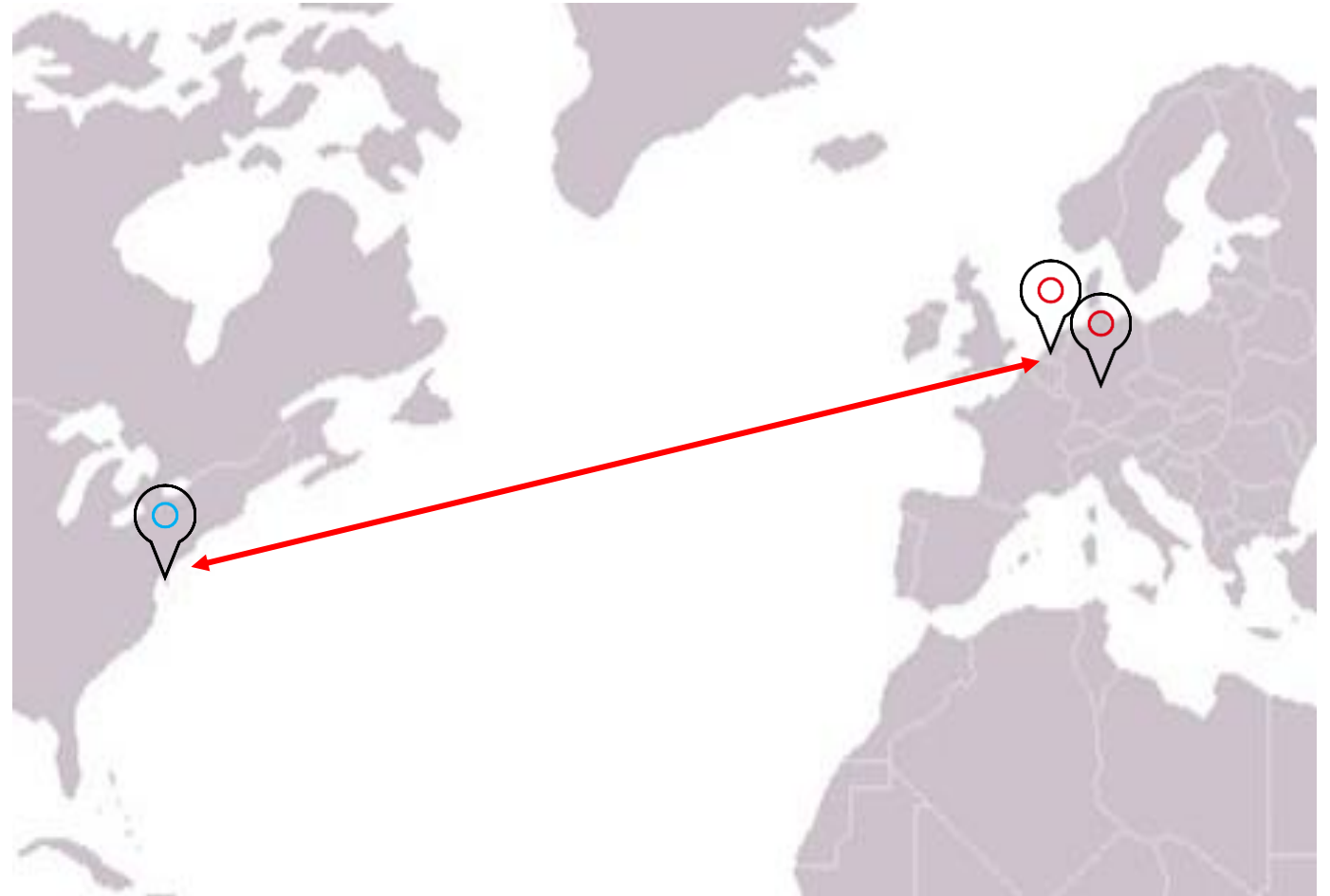
- 1Gbps Link
- ~94ms RTT
- Single XDR Queue
- Undersea cable cuts
- Storage and RAM imbalance



# Second EU Facility

## Introducing Frankfurt, DE

- Packet as cloud vendor
- Micron NVMe 9200s
- Cross-cloud networking
- 5TB of live objects



# Second EU Facility

## Wireguard

- High-performance VPN
  - Open source kernel module
  - Easy configuration
  - >2Gbps throughput
- 
- No out of box HA



### Amsterdam

[Interface]

PrivateKey = yAnz5TF+lXXJte14tji3zIMNq+hd2rYUIgJBgB3fBmk=

ListenPort = 51820

[Peer]

Endpoint = <frankfurt public ip>:51820

PublicKey = xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=

AllowedIPs = 10.0.0.0/24

### Frankfurt

[Interface]

PrivateKey = gl6EdUSYvn8ugXOt8QQD6Yc+JyiZxIhp3GlnSWRfWGE=

ListenPort = 51820

[Peer]

Endpoint = <amsterdam public ip>:51820

PublicKey = Hlgo9xNzJMWLKASShiTqlybxZ0U3wGLiUeJ1PKf8ykw=

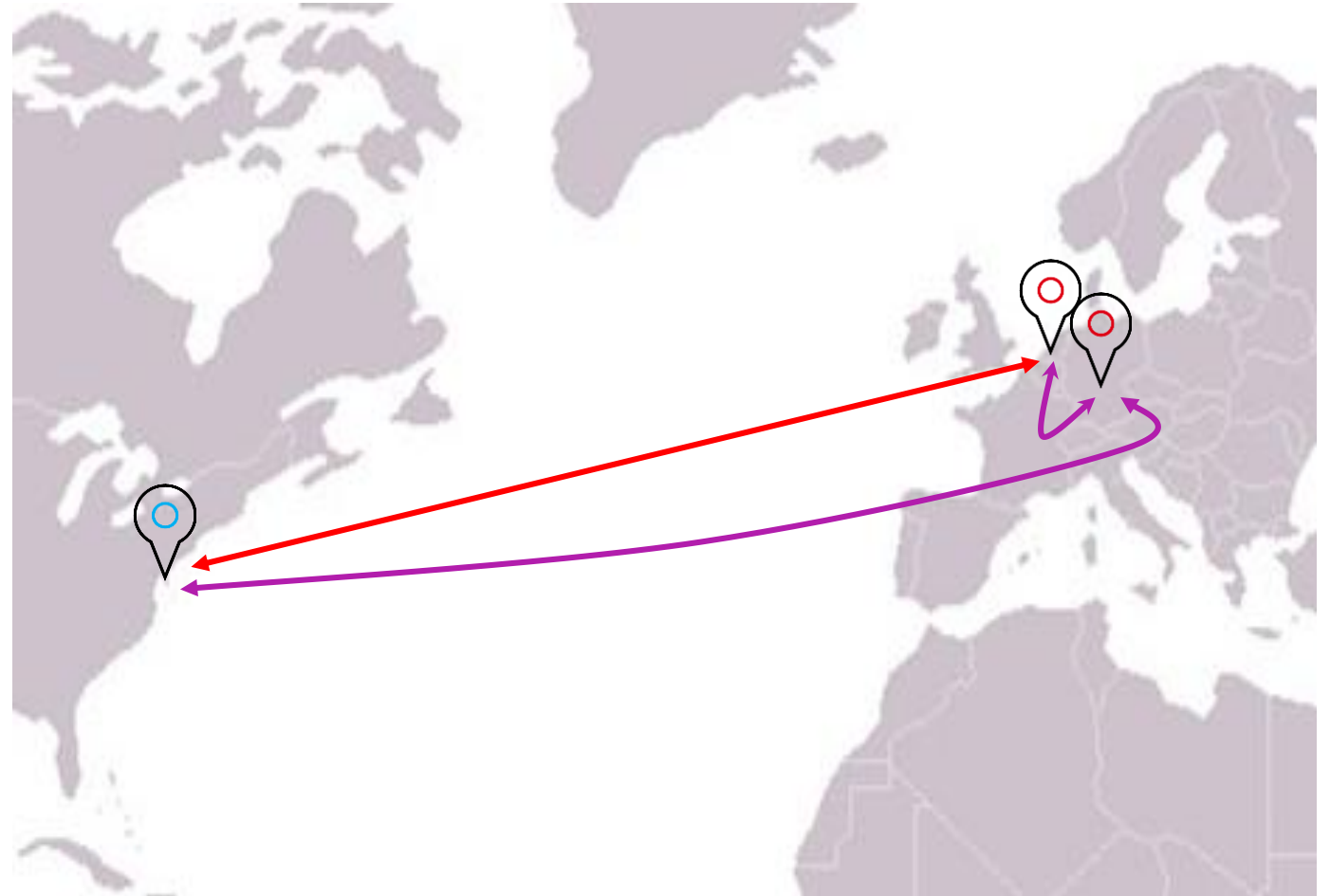
AllowedIPs = 10.0.1.0/24



# Second EU Facility

## Migration Steps

1. Configure XDR star
2. asbackup
3. asrestore (--unique)
4. Update GSLB DNS
5. Remove NJ from star



# US NVMe Refresh 2017-2018



# US NVMe Refresh

## NVMe Drive Selection

- Aerospike Certification Tool (ACT)
  - Stress-tests Drives
  - Adjustable Read/Write ratio
  - Adjustable Object Size
  - Goal of reads %>1ms under 5%
  - Run for > 24 hours
  - Run with multiple drives

slice	reads %>(ms)			device-reads %>(ms)		
	1	8	64	1	8	64
1	1.67	0.00	0.00	1.63	0.00	0.00
2	1.38	0.00	0.00	1.32	0.00	0.00
3	1.80	0.14	0.00	1.56	0.08	0.00
4	1.43	0.00	0.00	1.39	0.00	0.00
5	1.68	0.00	0.00	1.65	0.00	0.00
6	1.37	0.00	0.00	1.33	0.00	0.00
7	1.44	0.00	0.00	1.41	0.00	0.00
8	1.41	0.00	0.00	1.35	0.00	0.00
9	2.70	0.73	0.00	1.91	0.08	0.00
10	1.54	0.00	0.00	1.51	0.00	0.00
11	1.53	0.00	0.00	1.48	0.00	0.00
12	1.47	0.00	0.00	1.43	0.00	0.00
avg	1.62	0.07	0.00	1.50	0.01	0.00
max	2.70	0.73	0.00	1.91	0.08	0.00

Example ACT Report



# US NVMe Refresh

## Old Layout:

18 Server Nodes

12x 1.2TB SSD Drives

512GB RAM

Total Storage: 302.4TB

Total RAM: 9TB

## New Layout:

12 Server Nodes

8x 3.2TB NVMe Drives

786GB RAM

Total Storage: 307.2TB

Total RAM: 9TB



# US NVMe Refresh

## Typical Upgrade Path

1. Add all new nodes to cluster
2. Remove old nodes by looping:
  - a. Wait for migrations
  - b. Remove 1 old node

## Issues

1. Migrations take hours to complete
2. Last old nodes run out of space

NVMe Nodes	SSD Nodes	Required TB/Node	NVMe Storage (8x 3.2TB)	SSD Storage (14x 1.2TB)
	18	13.89		16.8
12	18	8.33	25.6	16.8
12	17	8.62	25.6	16.8
12	16	8.93	25.6	16.8
12	15	9.26	25.6	16.8
...	...			
12	4	15.63	25.6	16.8
12	3	16.67	25.6	16.8
12	2	17.86	25.6	16.8
12	1	19.23	25.6	16.8
12	0	20.83	25.6	16.8

Hypothetical 250TB of Used Storage



# US NVMe Refresh

## The Better Solution: Aerospike Rack Awareness

### Upgrade Path:

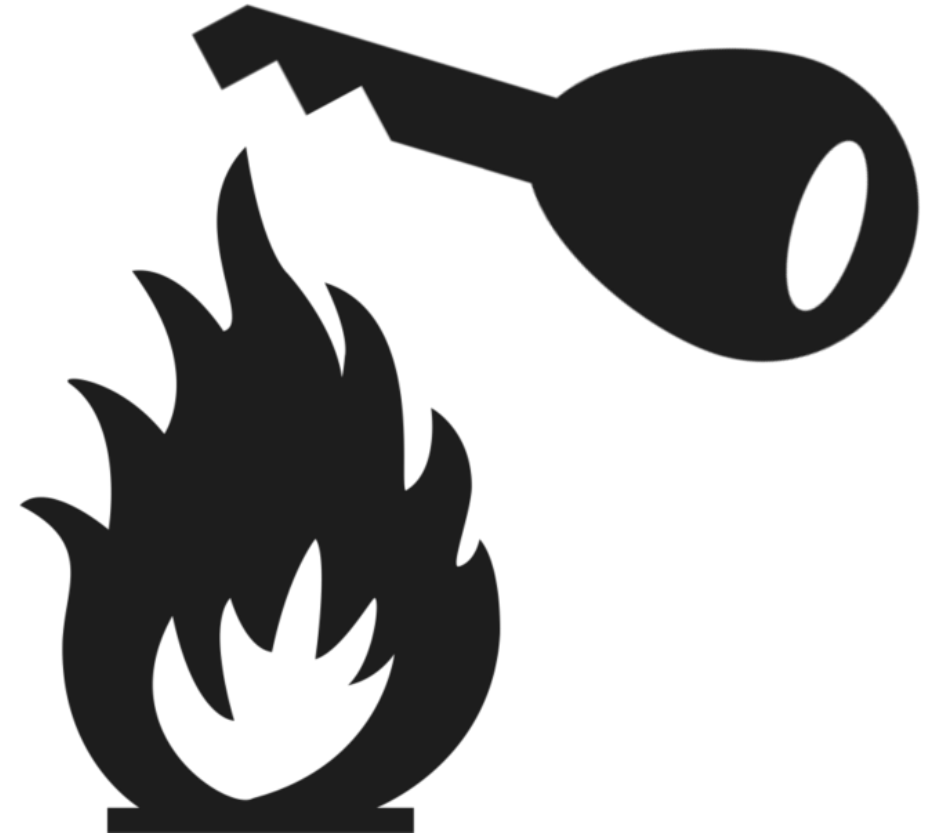
1. Set old nodes to `rack-id: 0`
2. Add new nodes with `rack-id: 1`
3. Wait for migrations to complete
4. Stop all old Aerospike nodes
5. Wait for migrations to complete

### Results:

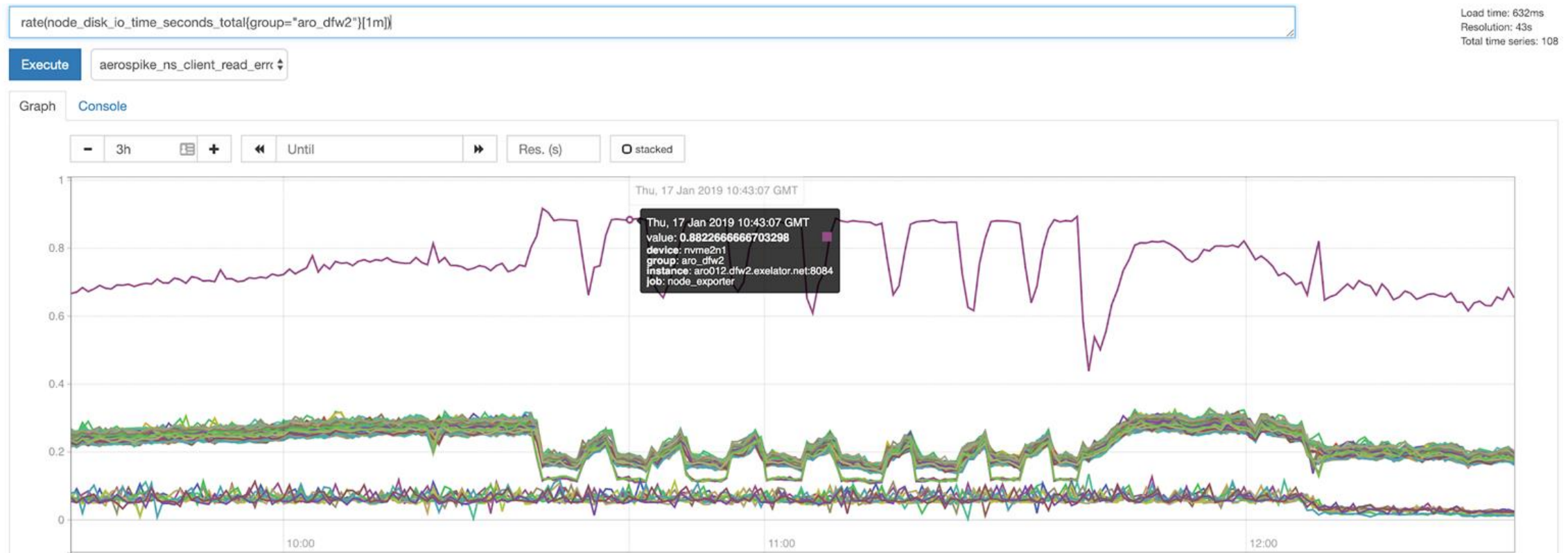
- Writes  $\%>1\text{ms}$  from 20%  $\rightarrow$  5%
- Reads latency ~ same
- Less latency impact from migrations
- ~Equal \$/node/month



# Identifying Read Hot-Keys 2019



# Identifying Read Hot-Keys



IO Utilization by Drive



# Identifying Read Hot-Keys

Enable query history

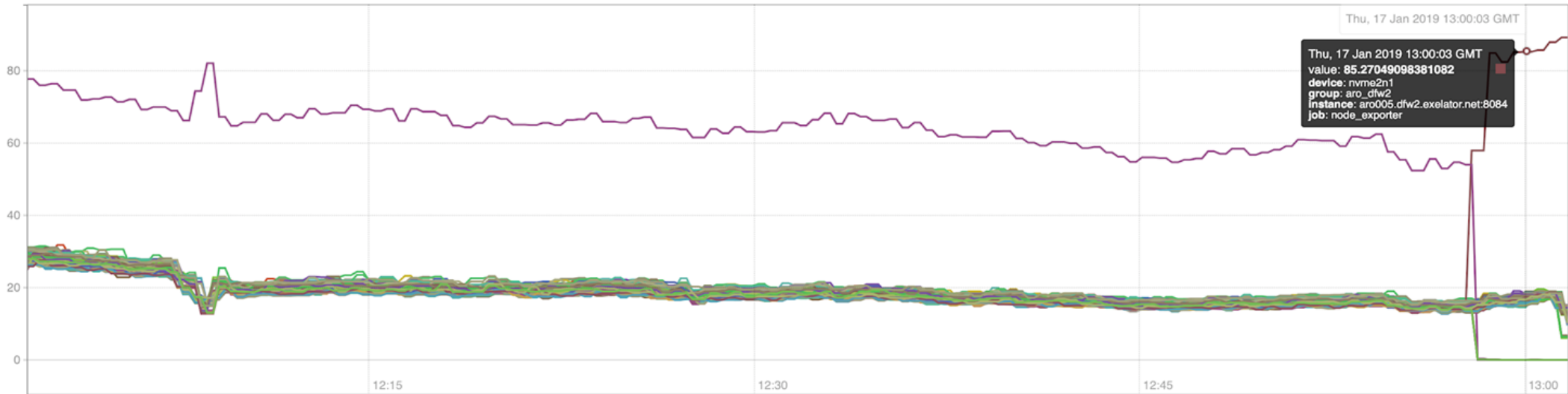
```
rate(node_disk_io_time_seconds_total{group="aro_dfw2",device=~"nvme.*"}[1m]) *100
```

Load time: 768ms  
Resolution: 14s  
Total time series: 9

Execute

Graph Console

1h ⏪ Until ⏩ Res. (s) stacked



IO Utilization by Drive



# Identifying Read Hot-Keys

## Enabling Debug Logging

```
asinfo -v 'set-log:id=0;rw-client=detail'
```

### Log output:

```
4 {users_us} <Digest>:0x43ba01edd9d3970b3bcda17f151b28946cc77994  
4 {users_us} <Digest>:0x71511f45f9b2e228df31420435d95f54889e7a8b  
4 {users_us} <Digest>:0x9d7988b0c8f9921f09d63e3fff890997a02bb696  
5 {lookup} <Digest>:0x3755a296ad9f9b418dbf8f779b3297d87e8d6f6a  
6 {lookup} <Digest>:0x3dcc95323c5725c2cc4b0fbb5292b2f1208a82d2  
6 {lookup} <Digest>:0xa256468db9041d4b8322925e4ddcd5cebd6d9f05  
8 {lookup} <Digest>:0xa03f550f1e0b3add3bf4f1126c22e19e35ce172f  
12 {users_us} <Digest>:0x65819c2a66df2eb18cbe9c9bd4b2c89bb19f1517  
15 {lookup} <Digest>:0x7d5f43a239034a32dee4dc8ec498d5a864bdc9a5  
421 {lookup} <Digest>:0xaeaa3f9c4263e4cc22e8faafd5a53fa96283e5f1
```



# Identifying Read Hot-Keys

So what is '{lookup} 0xeaea3f9c4263e4cc22e8faafd5a53fa96283e5f1'?

- Aerospike does not store the 'key' (by default)
- The digest is a one-way hash (RIPEMD160)
- Can perform reads from Aerospike by digest

## Client application logs

```
com.exelate.servingutils.aerospike.AerospikeWorker - wasn't able to complete the action 'get' from aerospike.  
namespace lookup key: IP_MAIN_.Exception is com.aerospike.client.AerospikeException$Connection: Error Code  
-8: java.net.SocketTimeoutException: connect timed out
```



# Identifying Read Hot-Keys

## The Key “IP\_MAIN\_”

- It's a IP address → user device lookup key
- Typically formatted as “IP\_MAIN\_< md5sum of IP addr >”
- Application attempting to read ‘IP\_MAIN\_’ + str(null)
  
- Application bug!



# Identifying Read Hot-Keys

## Possible Solutions

### Enabling read-page-caching

```
asinfo -v 'set-config:context=namespace;id=lookup;read-page-cache=true'
```

- Added in v4.3.1
- Enables Linux page cache
- Repeat reads hit RAM instead of disk

### Update Client Application

- Should it be aggressively reading the same key?

### Scale Up Hardware

- More IOPS (more \$)



# NMC's Takeaways

- Monitor Everything
- Alert on Anomalies
- Capacity Plan for Failure
- Trust in Aerospike Support



