Running Envoy as an Edge Proxy @ eBay

Qiu Yu (QY)
Bala Madhavan

Agenda

- eBay's Edge
- SW Architecture of Edge proxy
  - L4 Load Balancer
  - Managing Envoy in Kubernetes
  - Data Plane Journey
  - Caching on the Edge with Envoy
- Observability
- Deploy management
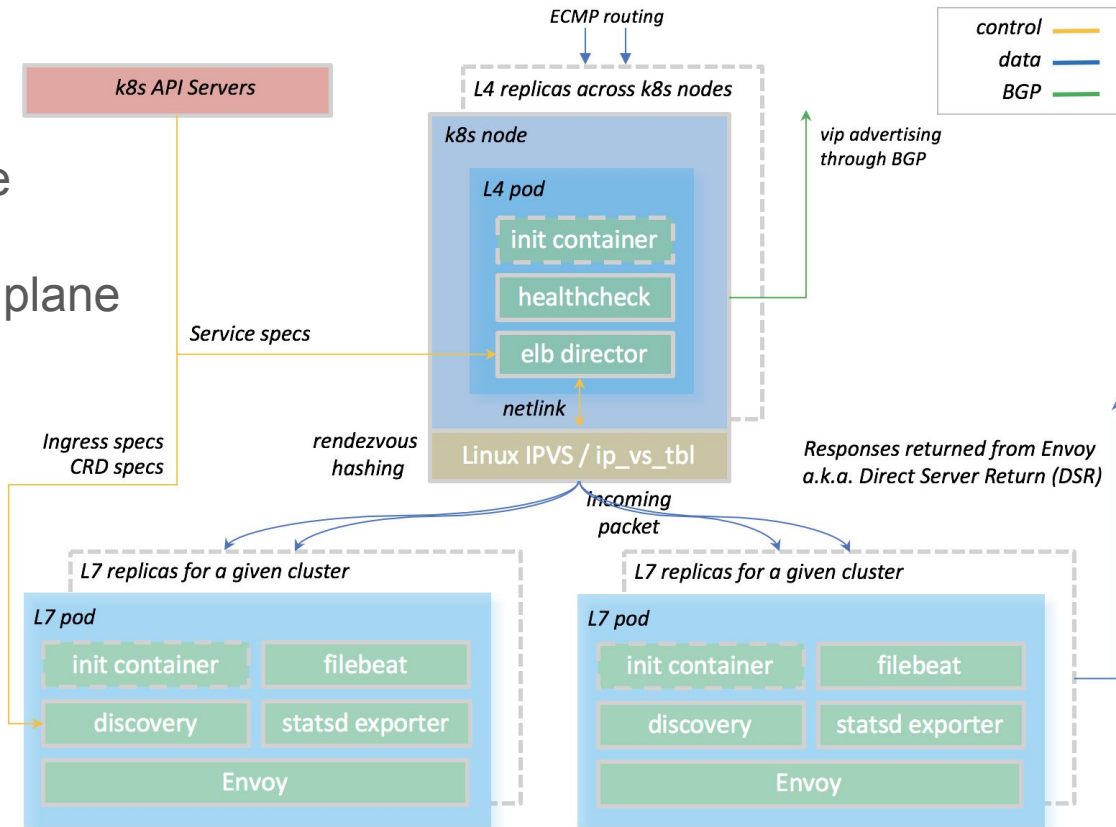- Learnings and Challenges
- Q&A

ebay™

eBay's Edge

- Why Edge
  - Data Centers in US
  - Getting closer to the end user

- Edge PoPs
  - Point of Presences around the Globe
  - Connect eBay network to the Internet
  - Software powered infrastructure
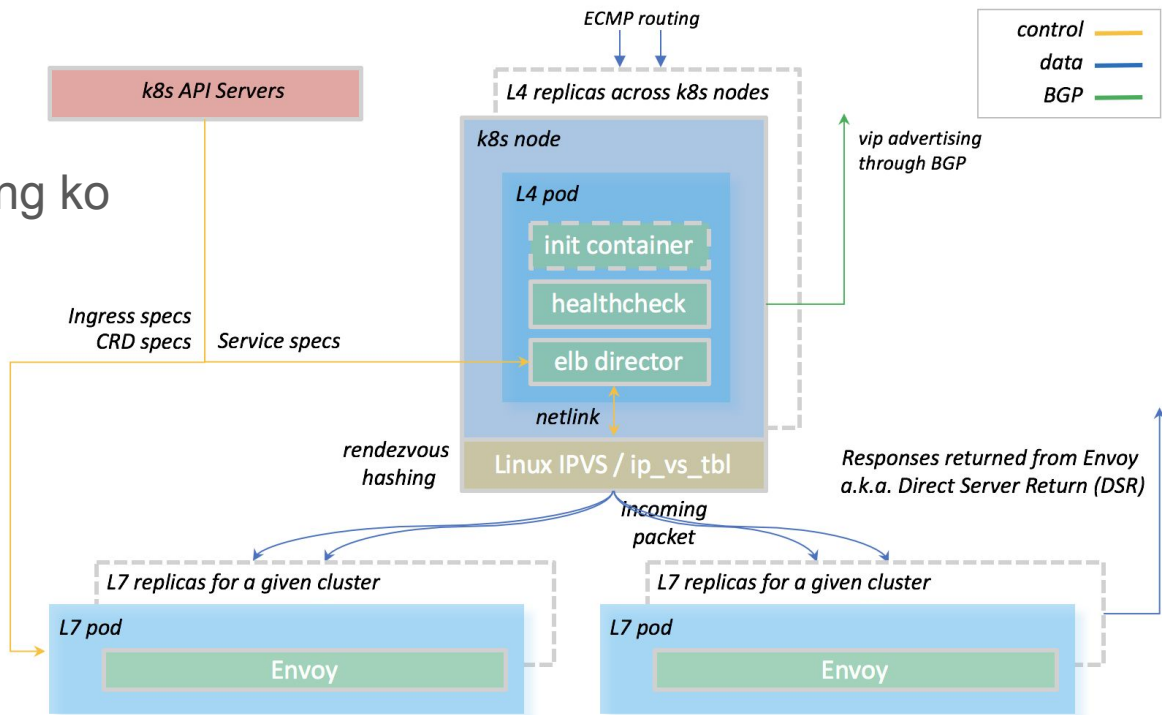   - k8s cluster in each PoP

- Beyond Speed

ebay™

Edge Proxy Architecture

- ## North-South gateway on Edge

- ## L4 + L7: Control plane + data plane

- ## Specs realized by controllers

- ## K8s native deployment

- ## Horizontally scalable

ECMP routing

control
data
BGP

k8s API Servers

L4 replicas across k8s nodes

k8s node

L4 pod

init container

healthcheck

elb director

vip advertising through BGP

Service specs

netlink

Ingress specs
CRD specs

rendezvous hashing

Linux IPVS / ip_vs_tbl

Incoming packet

Responses returned from Envoy a.k.a. Direct Server Return (DSR)

L7 replicas for a given cluster

L7 pod

init container	filebeat
discovery	statsd exporter
Envoy	

L7 replicas for a given cluster

L7 pod

| init container | filebeat |
| discovery | statsd exporter |
| Envoy | |

ebay™

L4 Load Balancer

- IPVS based data plane

- In-house rendezvous hashing ko

- BGP for VIP advertising

- Direct Server Return (DSR)

ECMP routing

control
data
BGP

L4 replicas across k8s nodes

k8s API Servers

vip advertising
through BGP

k8s node

L4 pod

init container

healthcheck

elb director

Ingress specs
CRD specs

Service specs

netlink

rendezvous
hashing

Linux IPVS / ip_vs_tbl

Responses returned from Envoy
a.k.a. Direct Server Return (DSR)

Incoming
packet

L7 replicas for a given cluster

L7 pod

Envoy

L7 replicas for a given cluster

L7 pod

Envoy

ebay™

Managing Envoy in Kubernetes

- Workloads: L7 PODs (envoy runs as container) & Deployment controller
  - Rolling updates
  - Scaling
  - Better control (Rollback, Pause, Live Status etc.)
- Listener configured via Ingress spec
  - Annotations for custom features specifications.
- L7 routes managed by Custom Resource Definition (CRDs)

ebay™
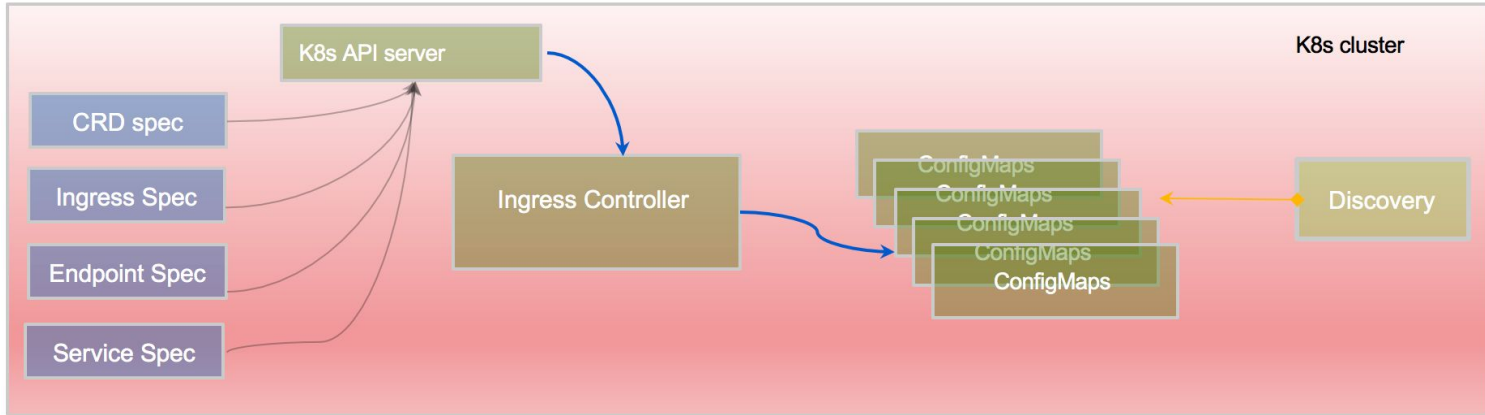
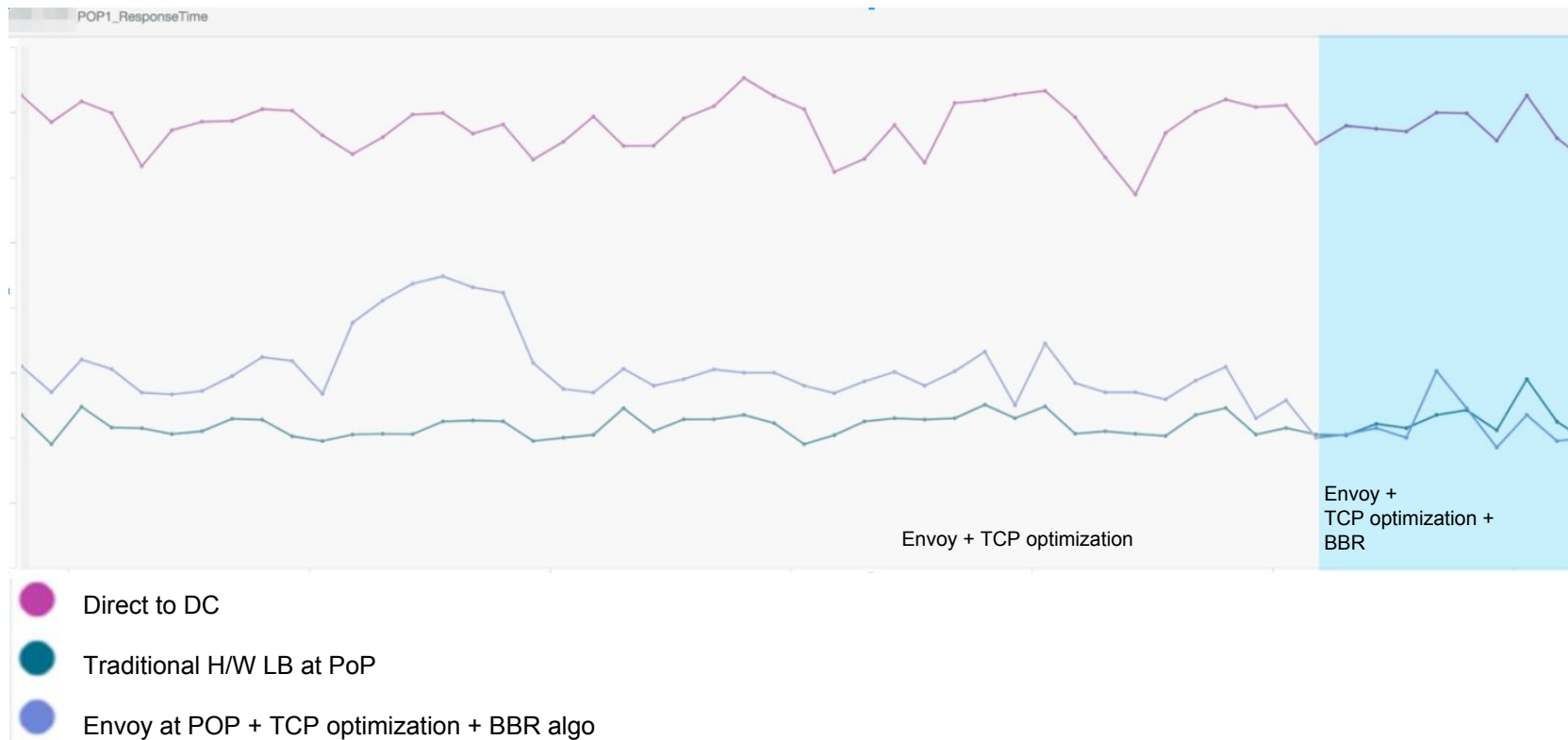# Managing Envoy in Kubernetes - Cont.
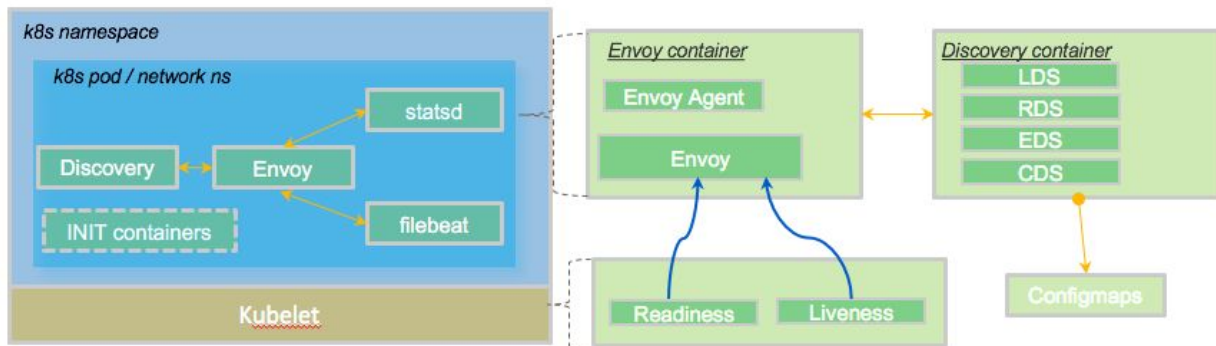
- Custom 'discovery' service handles RDS, LDS, EDS & CDS
- Readiness probe ensures dynamic discovery is complete
- Liveness probe ensures container is running

ebay™

Data Plane Journey

- ● Migration from H/W LB to envoy
  - ○ Identified and implemented feature gaps between `envoy` & H/W LB (& contributed upstream)
  - ○ Automation for migrating H/W LB policies to envoy config.
  - ○ Policy Validation - Auto generation of test requests & RUM
- ● Integration with Propriety Cert Management system
- ● Data Plane customizations
  - ○ Connection prefetching
  - ○ Drop or RST requests to certain routes based on need
  - ○ Default Upstream Cluster for graceful error handling
- ● BBR algorithm
- ● SSL & TCP optimization

ebay™

Data Plane Journey - Cont. - TCP optimization

POP1_ResponseTime

Envoy + TCP optimization

Envoy +
TCP optimization +
BBR

● Direct to DC

● Traditional H/W LB at PoP

● Envoy at POP + TCP optimization + BBR algo

ebay™

Caching on the Edge with Envoy

- Created a custom HTTP filter to support dynamic object caching (Yes. You can!!)
  - Based on URL path, cookies, and request headers
  - Unique for each website user
- Using external cache stores
- Used Envoy's RingHash LB to shard cached data on multiple cache stores
- Better TTFB by order of magnitude (10x)
  - Provides enhanced User Experience
 - Engages User quickly & increases conversion

Future Roadmap
- Enhanced prefetch filter to facilitate Just-In-Time caching

ebay

DC vs Edge Proxy vs Caching

Observability

- Prometheus in each PoP

- Central Grafana for visualization

- Envoy specifics
  - Statsd exporter for histogram data
  - FELK for logging data

- Alerting
  - Alertmanager and external checks
  - Static and Dynamic threshold (Anomaly Detection)

ebay™
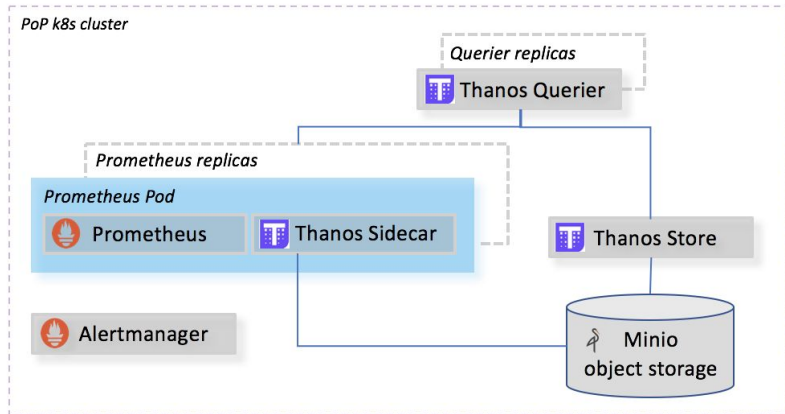
Helm based deploy management

- Repeatable deployment across all PoPs

- Helm Charts for the entire PoP stack

- Single source of truth of everything

```
[                        (⚙ |          :l4)]# helm ls
NAME                       REVISION      UPDATED                 STATUS      CHART                              NAMESPACE
alertmanager-prod                                  :19:40 2018  DEPLOYED    alertmanager-0.1.0
ats-exporter-prod                                  :13:38 2018  DEPLOYED    trafficserver-exporter-0.1.0
elasticsearch-prod                                 :07:56 2018  DEPLOYED    elasticsearch-0.1.3
ingress-controller-prod                            :56:51 2018  DEPLOYED        ingress-controller-0.1.1
ksm-prod                                           :30:55 2018  DEPLOYED    kube-state-metrics-0.1.0
l4-prod                                            :02:13 2018  DEPLOYED        l4-0.1.6
l7-         prod                                   :07:24 2018  DEPLOYED        l7-0.1.14
l7-prod                                            :02:06 2018  DEPLOYED        l7-0.1.14
logstash-prod                                      :27:20 2018  DEPLOYED    logstash-0.1.7
minio-prod                                         :30:13 2018  DEPLOYED    minio-1.6.0
prometheus-prod                                    :17:47 2018  DEPLOYED    prometheus-0.2.2
```

Learnings and Challenges

- Testing is the key
  - HW / SW cross validation
  - RUM to ensure no perf degradation
  - DC and PoP failover test
  - Chaos test

- For smooth workload migration
  - Traffic shift in controlled manner
  - Integration with Legacy systems
  - Cross-function planning and verification

- Multi PoP cluster operation
  - Scaling ops with increasing PoPs
  - Drift detection across PoPs
  - Tooling support for cluster management

- Programmable Edge
  - Build Vs. Integrate
  - Howto: act on data
  - Howto: debug and troubleshoot

ebay™

Thank You !

We are hiring !!!!

bmadhavan@ebay.com
qiuyu@ebay.com

ebay™