KubeCon | CloudNativeCon

North America 2018

Multi-Cloud Ingress LB: Gimbal Use Case in Actapio and Yahoo Japan

Who is Hirotaka?

Hirotaka “HIRO” Ichikawa

Actapio, Inc. (Yahoo Japan, Corp.)

Cloud Infrastructure Engineer

- Kubernetes
- OpenStack
- Networking
- Data Center

} Full Stack Infrastructure Engineer 🤔

What is Actapio?

ACTAPIO

- US subsidiary of Yahoo Japan
- 2 Data Center in WA
  - 16MW DC (next week 🎉)
- SW infrastructure acceleration
  - Load Balancing
  - Application Monitoring
  - Storage
  - etc...

Who is Ryutaro?

Ryutaro Inoue

Senior Manager @Yahoo! JAPAN

Speciality:

- Network
- Security
- HW
- OS
- OpenStack,Kubernetes

About Yahoo! JAPAN

- Joint venture of Softbank and Yahoo! Inc.
- 100+ services (news,auction,weather,etc..)
- 72billion PV per month
  - ○ 10 Data centers
  - ○ 100,000+ Servers
  - ○ 60+ OpenStack Clusters
  - ○ 100+ kubernetes Clusters

敷金やツケ 民法どう変わる?

野良猫に餌やり禁止 賛否続々

トルコ最大級の地下都市発見

皆既月食 夜桜と競演珍しく

ハリル監督の奇策 選手も驚く

ママタレ 割れる子供の顔出し

GIMBAL

What is Gimbal?

Ingress Load Balancing Platform

- built by Heptio w/ Actapio
- work on Kubernetes ecosystem
- **scalable, multi-team and API-driven**
- **support multi-cloud upstream**
  - **Kubernetes**
  - **OpenStack**

https://github.com/heptio/gimbal

Why#1: Necessity of Gimbal

BRIDGING THE GAP BETWEEN VM AND CONTAINER

Scalable Multi-Cloud Load Balancer

- support both of VM and Container cloud
- make migration from VM to Container EASY
- optimize operation cost by providing large single load balancing tier

Modernizing application release

RAPID RELEASE AT SCALE FOR ANY CLOUD

- Service Discovery
- Performance Measurement
- Canary Deployment
- Instant Rollback

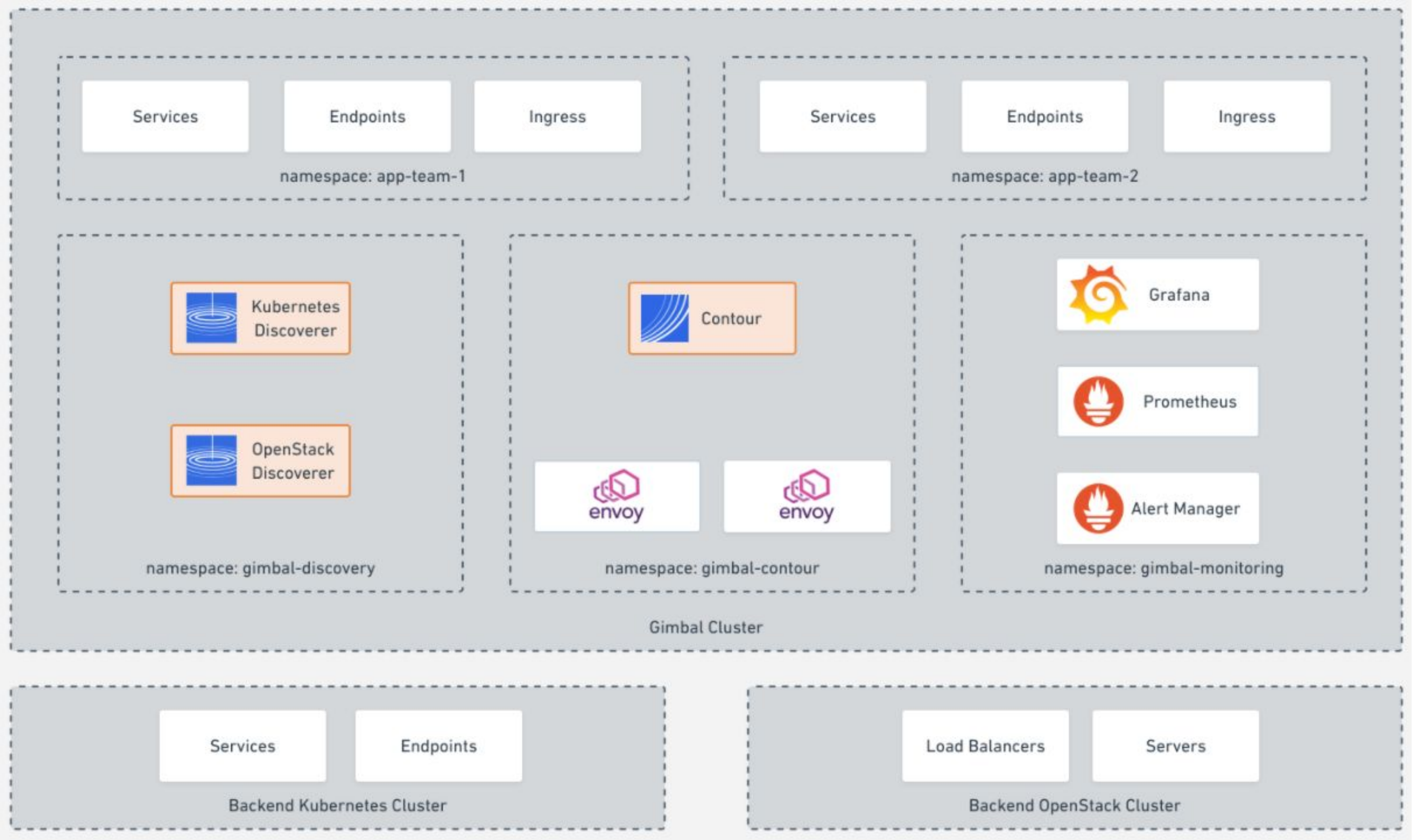
Why#2: Collaborating with Heptio

heptio

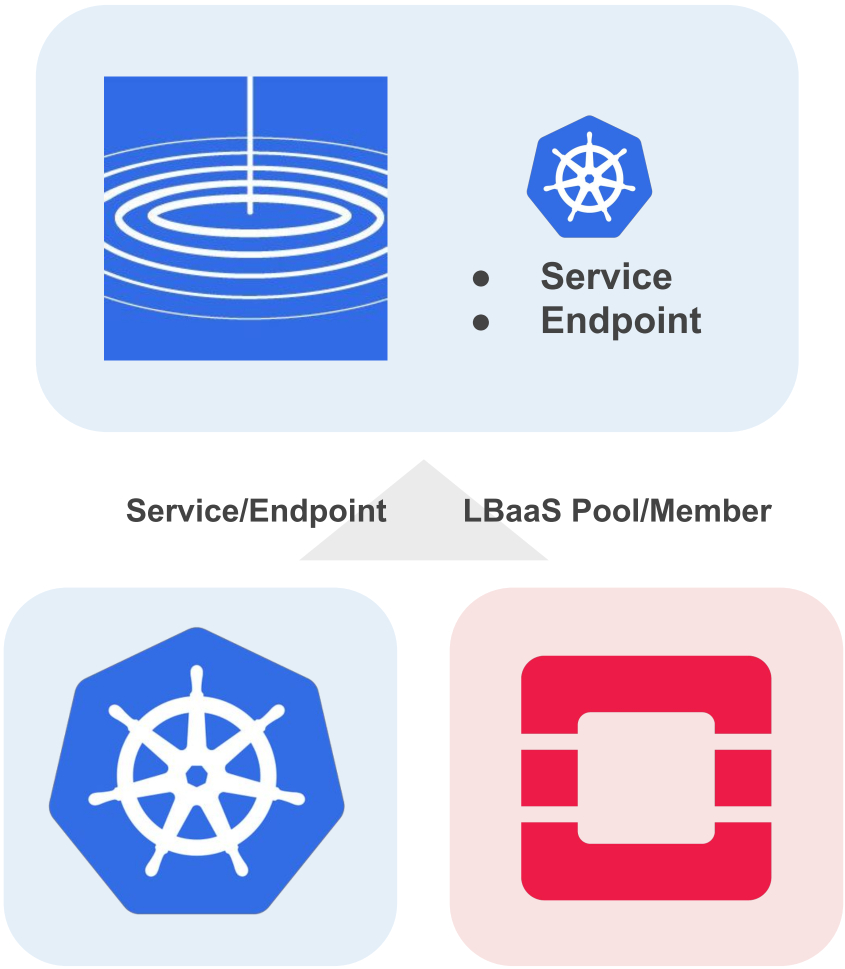
- Deep Expertise around K8S & Cloud Native
- Tight Connection to OSS Community

Technical Overview

Gimbal Architecture



Service Discovery



Gimbal discovers upstream cluster services. Developers can see their services at gimbal cluster and operate routing rules according to discovered services.

Supported Upstream Cloud

- Kubernetes
  - Service & Endpoint
- OpenStack
  - LBaaSv2 Pool & Member

K8S Service Discovery

# Gimbal K8S Cluster

```
$ kubectl get svc,ep  --all-namespaces
NAMESPACE        NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
team01           service/up01-team01-app   ClusterIP   None           <none>        80/TCP           136m
team01           service/up02-<LB_UUID>    ClusterIP   None           <none>        80/TCP           20h
team02           service/up01-team02-app   ClusterIP   None           <none>        80/TCP           42s

NAMESPACE        NAME                                ENDPOINTS                                    AGE
team01           endpoints/up01-team01-app           10.1.0.10:80,10.1.0.5:80,10.1.0.8:80         136m
team01           endpoints/up02-<LB_UUID>            172.17.165.214:80,172.17.165.215:80          20h
team02           endpoints/up01-team02-app           10.1.0.12:80                                 42s
```

# Upstream K8S Cluster

```
$ kubectl get svc,ep --all-namespaces
NAMESPACE     NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)         AGE
team01        service/team01-app     ClusterIP   10.96.236.146    <none>        80/TCP          15h
team02        service/team02-app     ClusterIP   10.98.14.219     <none>        80/TCP          7m51s

NAMESPACE     NAME                                ENDPOINTS                                         AGE
team01        endpoints/team01-app                10.1.0.10:80,10.1.0.5:80,10.1.0.8:80              15h
team02        endpoints/team01-app                10.1.0.12:80                                      7m51s
```

K8S Service Discovery

# Gimbal K8S Cluster

```
$ kubectl get svc,ep  --all-namespaces
NAMESPACE        NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
team01           service/up01-team01-app   ClusterIP   None           <none>        80/TCP           136m
team01           service/up02-<LB_UUID>    ClusterIP   None           <none>        80/TCP           20h
team02           service/up01-team02-app   ClusterIP   None           <none>        80/TCP           42s

NAMESPACE        NAME                                    ENDPOINTS                                    AGE
team01           endpoints/up01-team01-app               10.1.0.10:80,10.1.0.5:80,10.1.0.8:80         136m
team01           endpoints/up02-<LB_UUID>                172.17.165.214:80,172.17.165.215:80          20h
team02           endpoints/up01-team02-app               10.1.0.12:80                                 42s
```

# Upstream K8S Cluster

```
$ kubectl get svc,ep --all-namespaces
NAMESPACE     NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)         AGE
team01        service/team01-app     ClusterIP   10.96.236.146    <none>        80/TCP          15h
team02        service/team02-app     ClusterIP   10.98.14.219     <none>        80/TCP          7m51s

NAMESPACE     NAME                                ENDPOINTS                                         AGE
team01        endpoints/team01-app                10.1.0.10:80,10.1.0.5:80,10.1.0.8:80              15h
team02        endpoints/team01-app                10.1.0.12:80                                      7m51s
```

K8S Service Discovery

# Gimbal K8S Cluster

```
$ kubectl get svc,ep  --all-namespaces
NAMESPACE        NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
team01           service/up01-team01-app   ClusterIP   None           <none>        80/TCP           136m
team01           service/up02-<LB_UUID>    ClusterIP   None           <none>        80/TCP           20h
team02           service/up01-team02-app   ClusterIP   None           <none>        80/TCP           42s

NAMESPACE        NAME                                ENDPOINTS                                    AGE
team01           endpoints/up01-team01-app           10.1.0.10:80,10.1.0.5:80,10.1.0.8:80         136m
team01           endpoints/up02-<LB_UUID>            172.17.165.214:80,172.17.165.215:80          20h
team02           endpoints/up01-team02-app           10.1.0.12:80                                 42s
```

# Upstream K8S Cluster

```
$ kubectl get svc,ep --all-namespaces
NAMESPACE     NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)         AGE
team01        service/team01-app     ClusterIP   10.96.236.146    <none>        80/TCP          15h
team02        service/team02-app     ClusterIP   10.98.14.219     <none>        80/TCP          7m51s

NAMESPACE     NAME                                ENDPOINTS                                         AGE
team01        endpoints/team01-app                10.1.0.10:80,10.1.0.5:80,10.1.0.8:80              15h
team02        endpoints/team01-app                10.1.0.12:80                                      7m51s
```

Routing and Policy

heptio
Contour

envoy

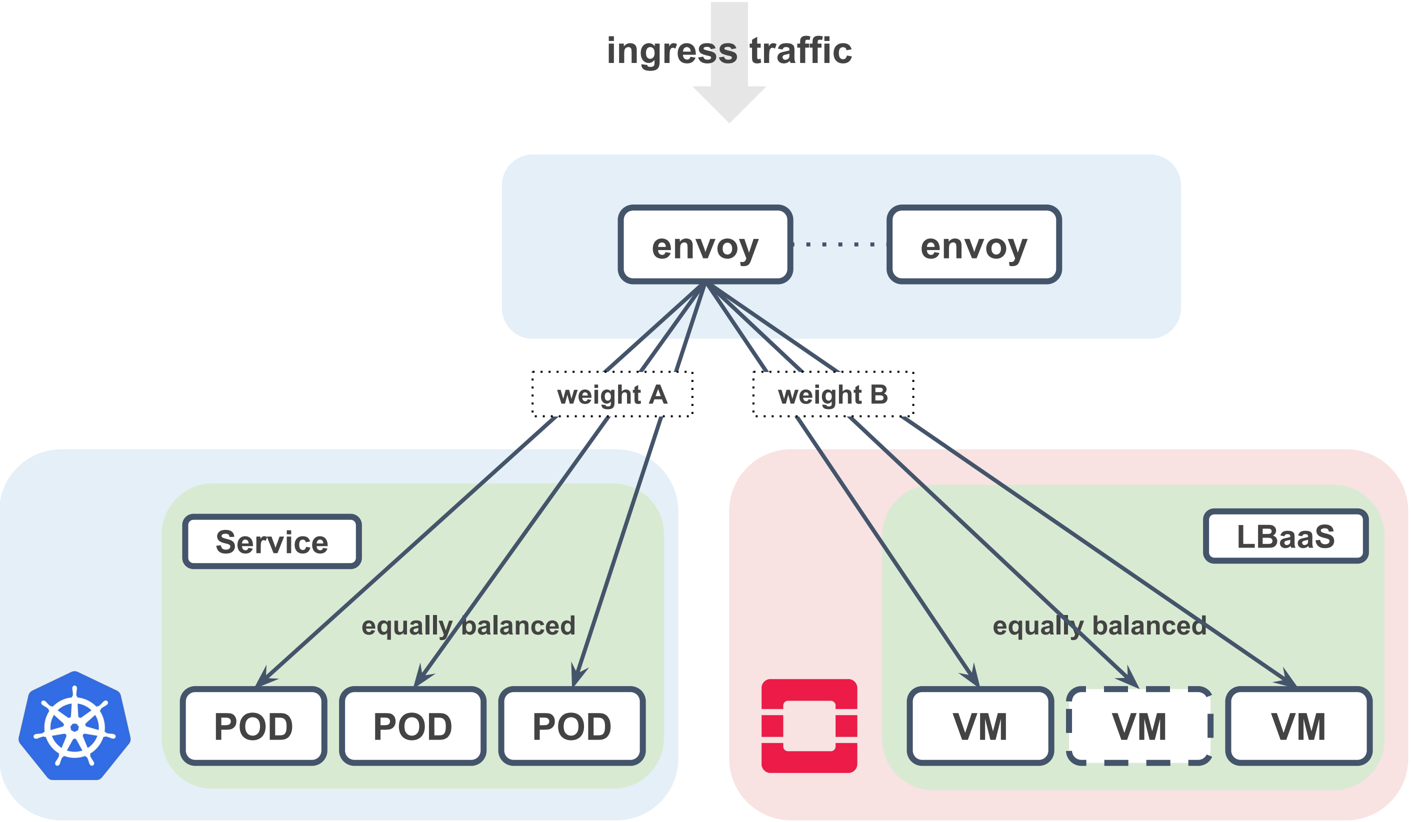
- Ingress controller for envoy
- Advanced Capability
  - Weight-shifting
  - Load balancing method
  - cross-cluster backends

https://github.com/heptio/contour

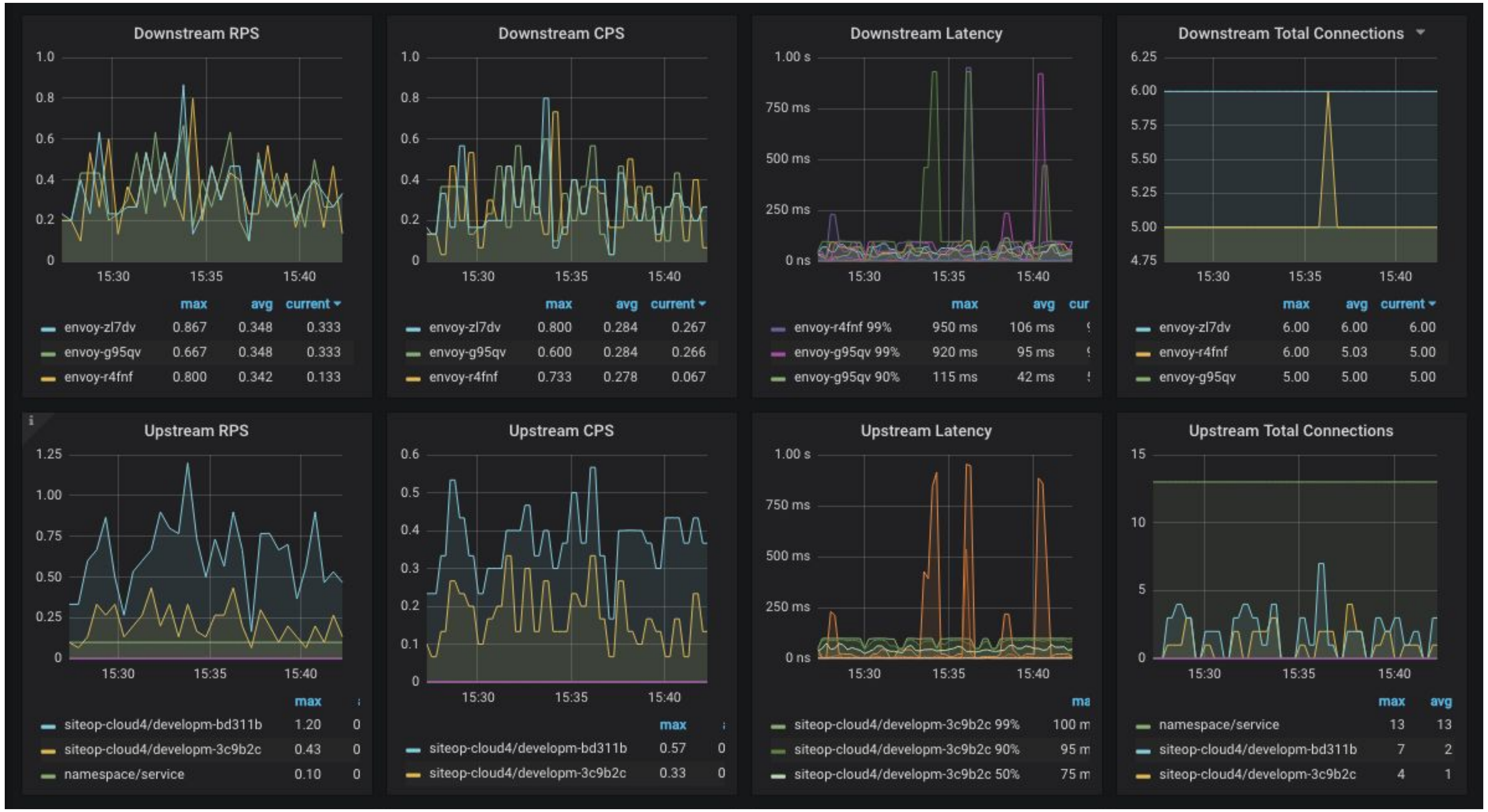
Ingress Routing

```
$ kubectl get ingressroute --namespace team01
NAME         FQDN   ALIASES   TLS SECRET   FIRST ROUTE   STATUS   STATUS DESCRIPTION
app01                                      /             valid    valid IngressRoute
```

```yaml
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: team01-app
spec:
  virtualhost:
    fqdn: team01-app.example.com
  routes:
    - match: /
      services:
        - name: up01-team01-app
          port: 80
          weight: 75
        - name: up02-team01-<UUID>
          port: 80
          weight: 25
          healthCheck:
            path: /status.html
            intervalSeconds: 60
```



Observability



Downstream RPS

	max	avg	current ▾
envoy-zl7dv	0.867	0.348	0.333
envoy-g95qv	0.667	0.348	0.333
envoy-r4fnf	0.800	0.342	0.133

Downstream CPS

	max	avg	current ▾
envoy-zl7dv	0.800	0.284	0.267
envoy-g95qv	0.600	0.284	0.266
envoy-r4fnf	0.733	0.278	0.067

Downstream Latency

	max	avg	cur
envoy-r4fnf 99%	950 ms	106 ms	[illegible]
envoy-g95qv 99%	920 ms	95 ms	[illegible]
envoy-g95qv 90%	115 ms	42 ms	[illegible]

Downstream Total Connections ▾

	max	avg	current ▾
envoy-zl7dv	6.00	6.00	6.00
envoy-r4fnf	6.00	5.03	5.00
envoy-g95qv	5.00	5.00	5.00

Upstream RPS

	max	
siteop-cloud4/developm-bd311b	1.20	0
siteop-cloud4/developm-3c9b2c	0.43	0
namespace/service	0.10	0

Upstream CPS

	max	
siteop-cloud4/developm-bd311b	0.57	0
siteop-cloud4/developm-3c9b2c	0.33	0

Upstream Latency

	ma
siteop-cloud4/developm-3c9b2c 99%	100 m
siteop-cloud4/developm-3c9b2c 90%	95 m
siteop-cloud4/developm-3c9b2c 50%	75 m

Upstream Total Connections

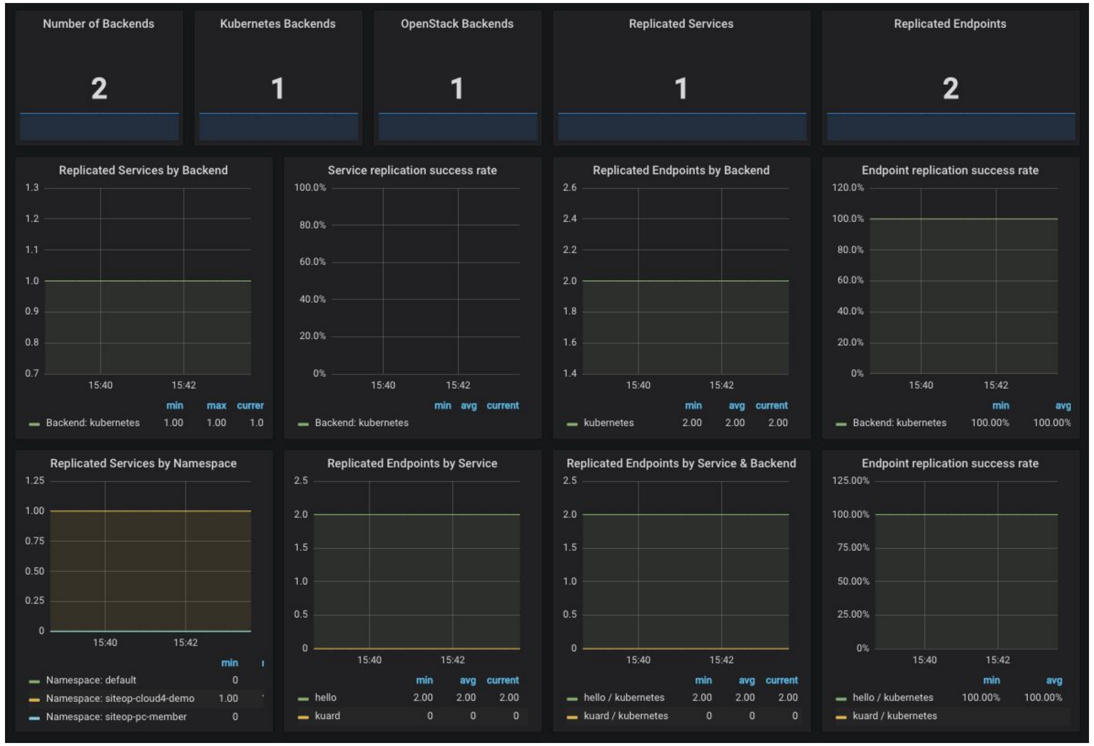
	max	avg
namespace/service	13	13
siteop-cloud4/developm-bd311b	7	2
siteop-cloud4/developm-3c9b2c	4	1

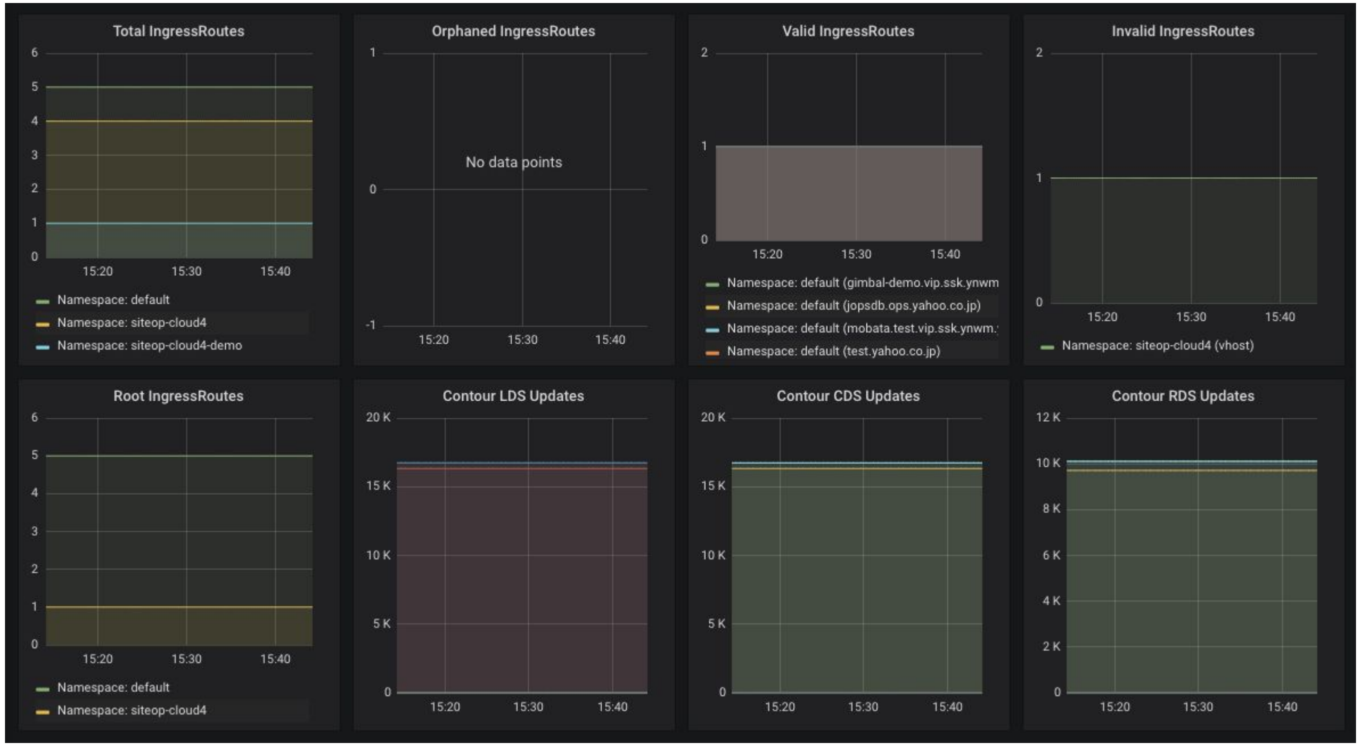
Activity Monitoring

- Up/Downstream
  - Request
  - Connection
  - Latency

Developers can control balancing weight watching each service upstream situation.

Observability


Number of Backends
2
Kubernetes Backends
1
OpenStack Backends
1
Replicated Services
1
Replicated Endpoints
2
Replicated Services by Backend
Backend: kubernetes min 1.00 max 1.00 curre 1.0
Service replication success rate
Backend: kubernetes min avg current
Replicated Endpoints by Backend
kubernetes min 2.00 avg 2.00 current 2.00
Endpoint replication success rate
Backend: kubernetes min 100.00% avg 100.00%
Replicated Services by Namespace
Namespace: default 0
Namespace: siteop-cloud4-demo 1.00
Namespace: siteop-pc-member 0
Replicated Endpoints by Service
hello 2.00 2.00 2.00
kuard 0 0 0
Replicated Endpoints by Service & Backend
hello / kubernetes 2.00 2.00 2.00
kuard / kubernetes 0 0 0
Endpoint replication success rate
hello / kubernetes 100.00% 100.00%
kuard / kubernetes
15:40
15:42

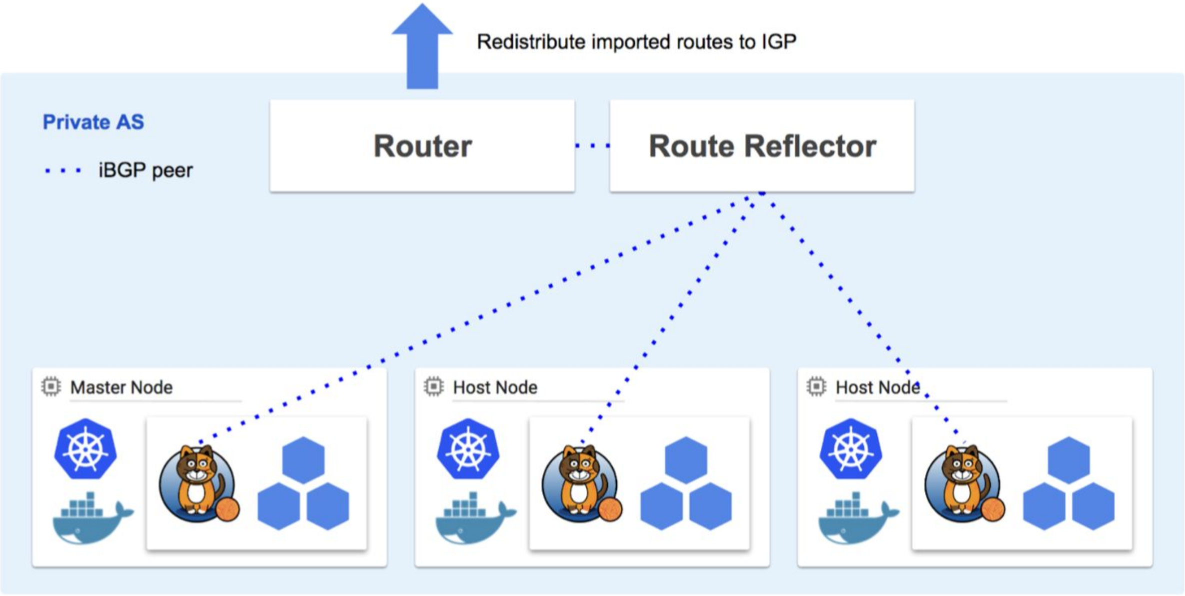


For Operators:
Backend Clusters / Replication Status / IngressRoutes

Current Limitations

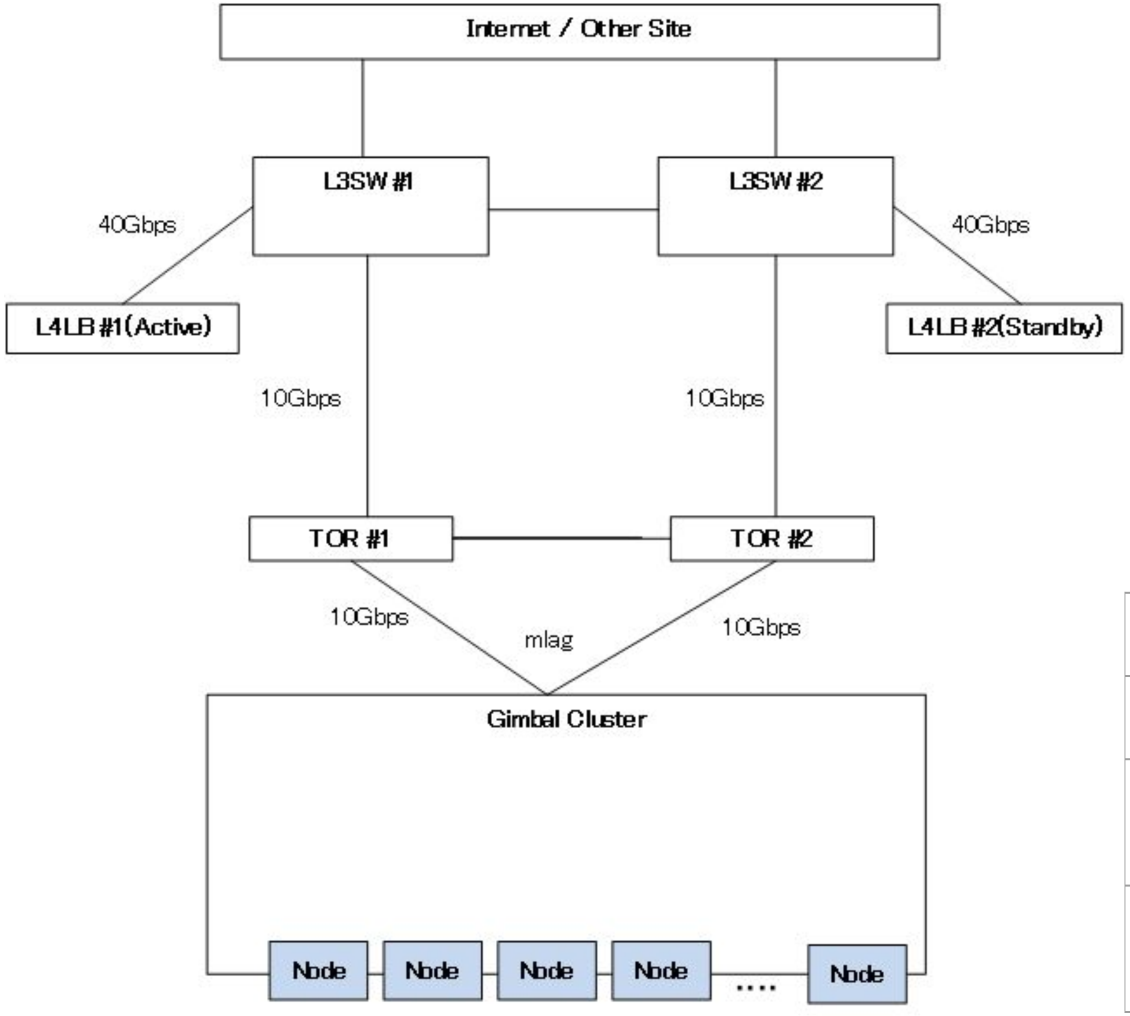
- POD should be reachable from outside cluster
  - use "`hostNetwork: true`"
  - advertise pod network with Calico

These methods are not applicable to all K8S users. We need future improvement.



Case study at Yahoo! JAPAN

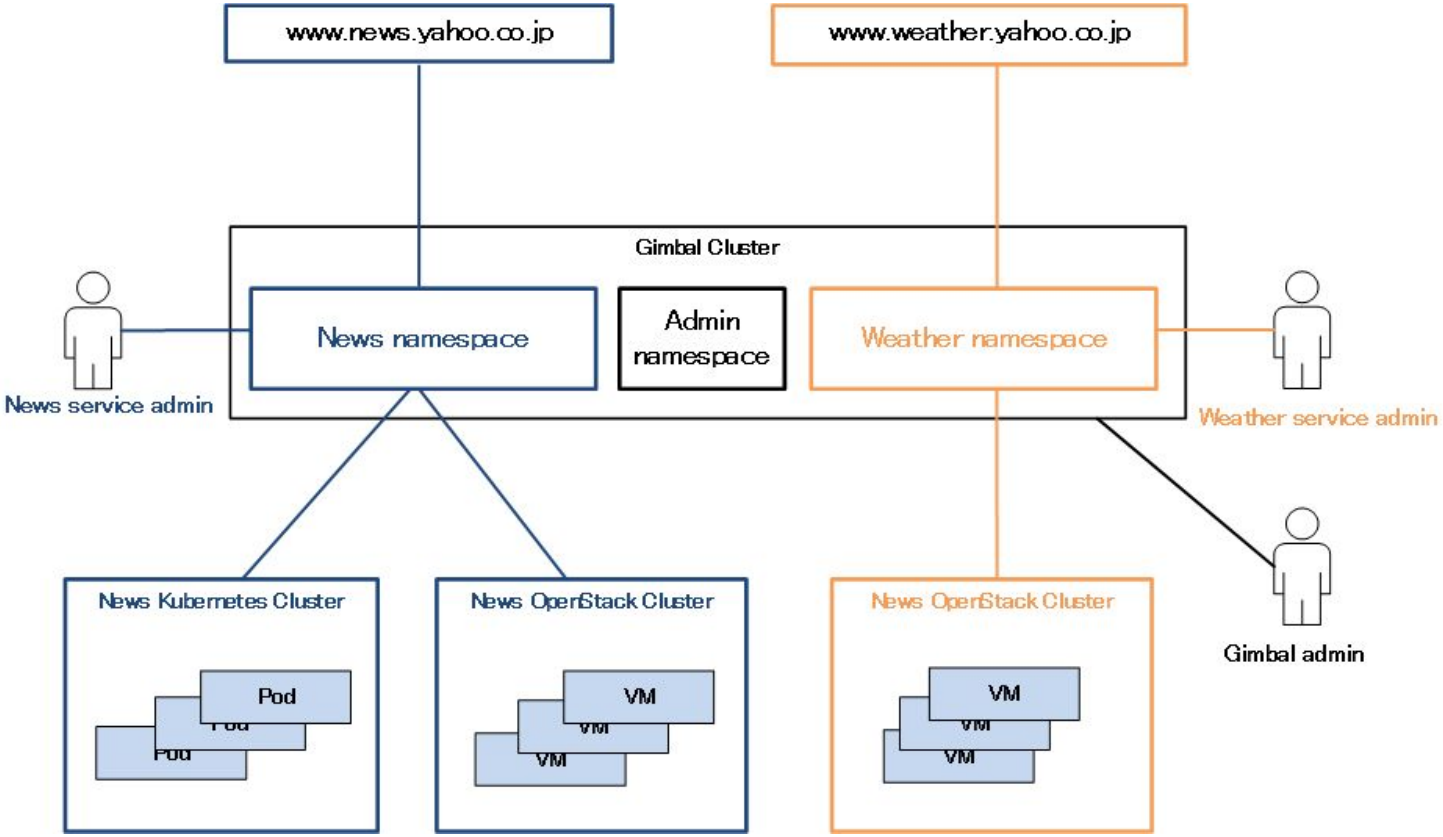
Gimbal cluster structure in Yahoo! JAPAN production env



- 1 Cluster per datacenter
- L4LB uses DSR (Direct Server Return)
- Envoy pods in Gimbal cluster uses hostNetwork
  - 1 Envoy pod per node

	HW/Spec	OS	number
L4LB	A10 Thunder 3030S	ACOS 2.7.2-P12-SP1	2(HA pair)
Worker Nodes	Xeon E5-2683v4 * 2 / 512GBmem/ 10G NIC * 2	Ubuntu 16.04.5 LTS	10
Master & etcd Nodes	Xeon E5-2683v4 * 2 / 512GBmem / 10G NIC * 2	Ubuntu 16.04.5 LTS	3

Using Gimbal



- The application is developed for each service
  - Requires namespace for each service

Service admin:

- Managing endpoint(vm/pod)
- Managing Gimbal config in their namespace

Gimbal admin:

- Managing entire Gimbal cluster
  - Admin Namespace,Nodes, L4LB,etc...

Using Gimbal at Yahoo! JAPAN

Beginning of use

1. Service admin sends a request Gimbal admin via ticketing system(JIRA)
   - Namespace & FQDN
2. Gimbal admin makes a kubeconfig file with authority for service namespace
3. Gimbal admin create the FQDN in admin(default) namespace
4. Gimbal admin sends the kubeconfig to service admin
5. Service admin modifies gimbal config using their kubeconfig
6. Service admin modifies their DNS record to gimbal cluster's ip

Modifying IngressRoutes

1. Service admin modifies Gimal cluster config

Adding or deleting Endpoints

1. Service admin modifies their backend (openstack or kubernetes)
   a. Gimbal discovers automatically applies the changes to ingressroute

Config sample (delegation)

admin namespace(default)

```
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: root-news-yahoo-co-jp
  namespace: default
spec:
  routes:
  - delegate:
      name: news-yahoo-co-jp
      namespace: news-team
    match: /
  virtualhost:
    fqdn: news.yahoo.co.jp
```

service namespace(news-team)

```
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: news-yahoo-co-jp
  namespace: news-team
spec:
  routes:
    - match: /
      services:
        - name: os1-6f671e13-1fbd-4c6d-a1db-31a565bb019c
          port: 80
          weight: 60
        - name: k8s1-7877d3b4-a471-4f38-9459-69659a593b38
          port: 80
          weight: 40
```

Performance Testing

Performance Testing -environment

Workload cluster:

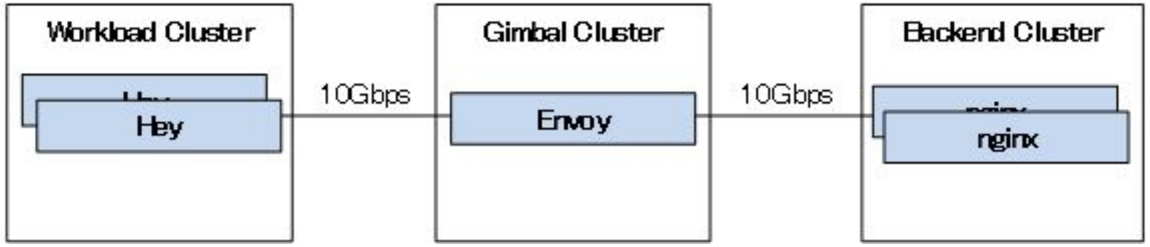
- kubernetes on vm
- VM(4vCPU/8Gmem) * 10 nodes
- load generator: hey (https://github.com/rakyll/hey)

Gimbal cluster:

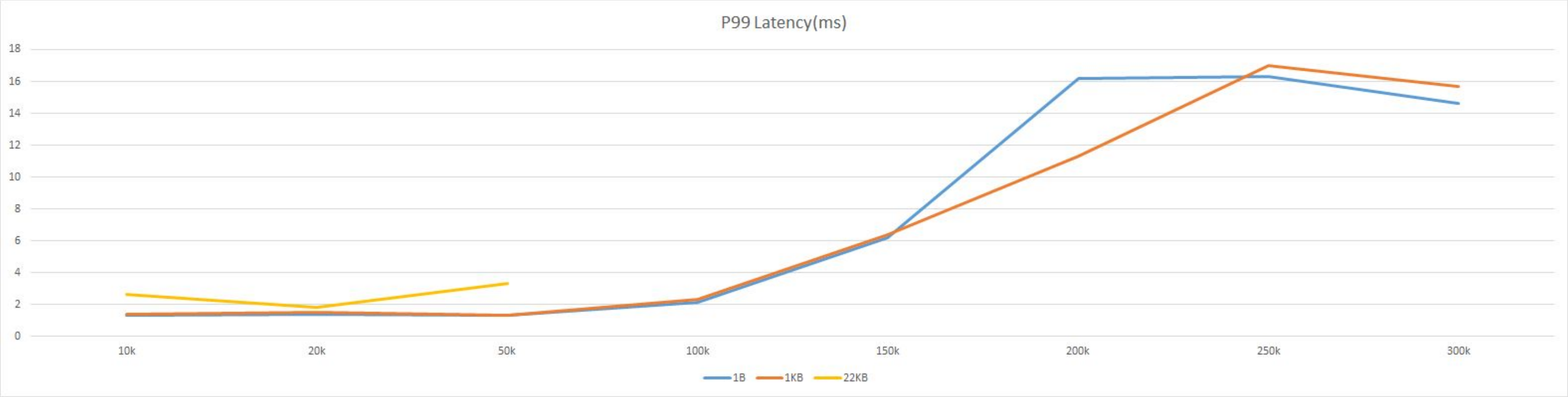
- kubernetes on physical machine
- spec:Xeon E5-2683v4 * 2 (64 thread) / 512GBmem
- 1 Enovy pod,
- Envoy 1.6.0

Backend cluster

- kubernetes on vm
- VM(4vCPU/8Gmem) * 10nodes
- application: nginx

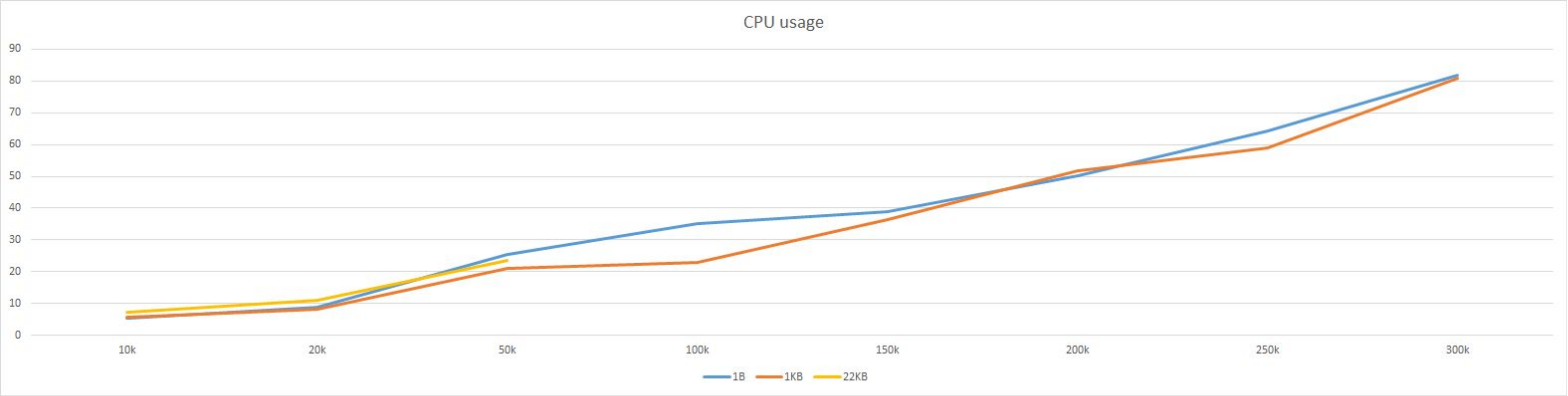


Performance Testing - Latency



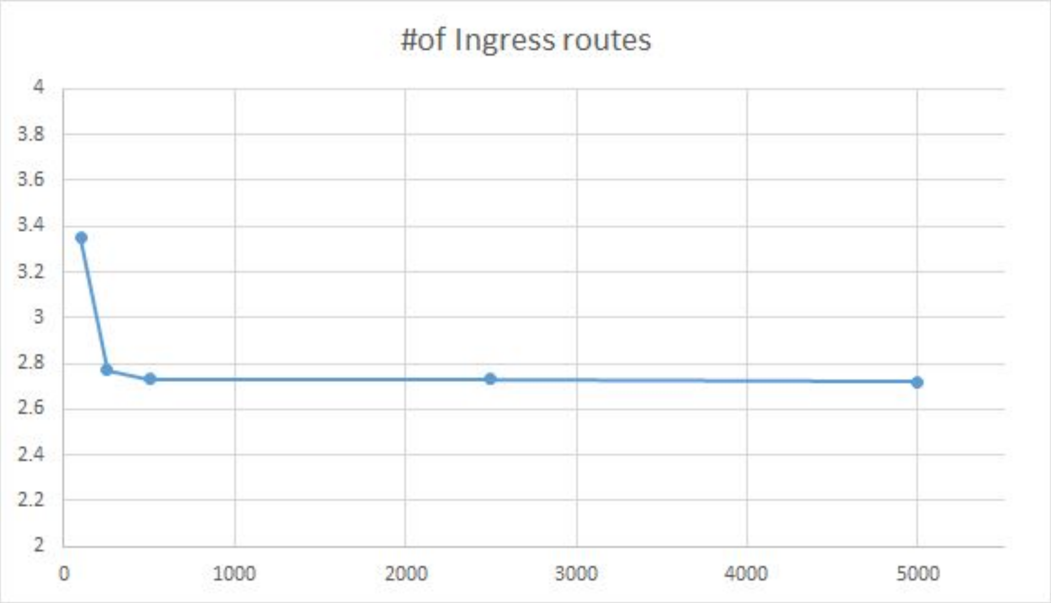
	Request per second								comment
Data Size	10k	20k	50k	100k	150k	200k	250k	300k	
1B	1.3ms	1.4ms	1.3ms	2.1ms	6.2ms	16.2ms	16.3ms	14.6ms	
1KB	1.4ms	1.5ms	1.3ms	2.3ms	6.4ms	11.3ms	17.0ms	15.7ms	
22KB	3.8ms	5.4ms	12.5ms	-	-	-	-	-	saturation of bandwidth(10Gbps)

Performance Testing - CPU usage

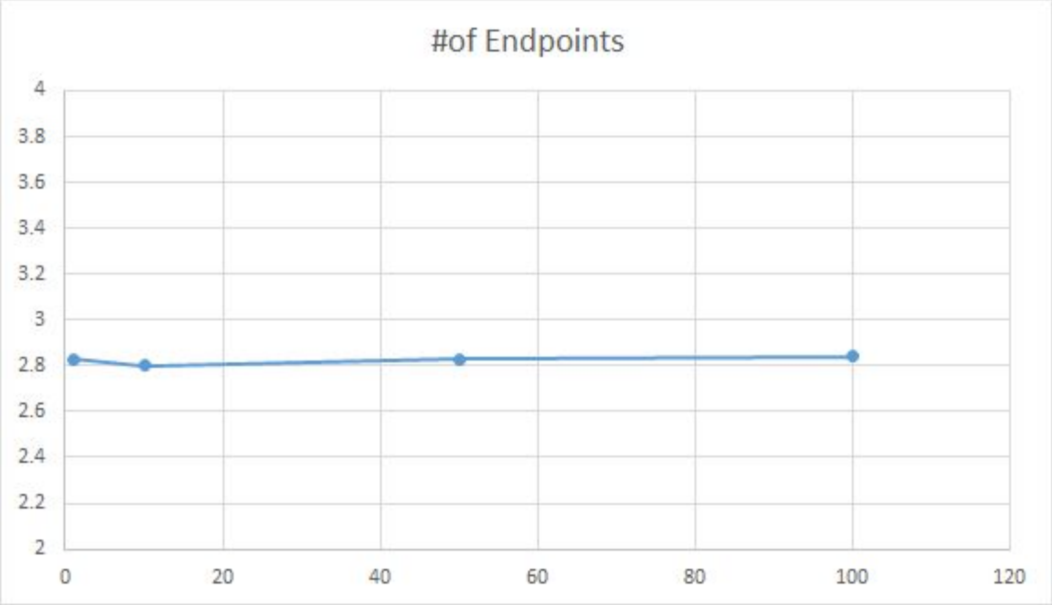


	Request per second								comment
Data Size	10k	20k	50k	100k	150k	200k	250k	300k	
1B	5.33%	8.66%	25.33%	35.10%	38.85%	50.25%	64.22%	81.88%	
1KB	5.66%	8.16%	20.88%	22.84%	36.28%	51.71%	59.05%	80.89%	
22KB	7.17%	11.02%	23.45%	-	-	-	-	-	saturation of bandwidth(10Gbps)

Performance Testing -
# of Ingress routes / Endpoints



# of Ingress routes	P99 Latency (ms)
100	3.35
250	2.77
500	2.73
2500	2.73
5000	2.72



# of Endpoints	P99 Latency (ms)
1	2.83
10	2.80
50	2.83
100	2.84

@22KB,45krps

Performance Testing -Conclusion

- An Envoy pod achieves 300k req/s @1KB data size
  - Might achieve higher?
- It scales linearly in terms of structure, 10 envoy pods will achieve 3M req/s
- No degradation of latency was observed when increasing the number of ingressroutes or endpoints

Future activities

heptio
Gimbal

- Expansion of usage in Yahoo JAPAN production environment
- Development of faster openstack discover
- Change Hardware L4LB into SWLB in upstream of Gimbal cluster
- Development of GUI

Conclusion

- Gimbal Overview
  - Bridging the gap between VM and Container
  - Modernize traditional application release
- Production ready performance at scale
- Yahoo Japan will expand Gimbal in production
- PR is welcome

Thank you

heptio
Gimbal

ACTAPIO

YAHOO!
JAPAN

Check it out https://github.com/heptio/gimbal

Office Hour: 12/13 12:15 - 13:45 @Heptio booth

KubeCon | CloudNativeCon

North America 2018