Topology-aware Service Routing in Kubernetes Boots a Smarter Service Discovery

Jun Du, Software Engineer, Huawei Cloud
Agenda

- Topologies in Kubernetes
- Topology-aware service routing
- Solutions and prototypes
- Q&A
Topography is Arbitrary

- AZ
- Region
- Rack
- Host
- Generator
- Anything you like…
Where should I run this Pod?

Scheduling is about finding hardware to run your code.
Node Affinity

Should I run my Pod on this Node?

```yaml
# provided by Kubernetes:
k8s.io/hostname
failure-domain.beta.k8s.io/zone
failure-domain.beta.k8s.io/region
beta.k8s.io/instance-type
beta.k8s.io/os
beta.k8s.io/arch

# user-defined (cluster admin, cloud provider, etc):
rack, disktype, ...

pod:
  name: postgres-primary
  ...
  affinity:
    - node: failure-domain.beta.k8s.io/zone=us-east-1a

---

pod:
  name: postgres-standby
  ...
  affinity:
    - node: failure-domain.beta.k8s.io/zone=us-east-1b
```
Pod Affinity/Anti-affinity

- Labels identify topologies
- `topologyKey` is the key of Node Labels

Should I run my Pod in the same hostname as a web-frontend Pod?
Topologies in Pod (Anti-)Affinity
node-4
pod-a

node-1
pod-b

failure-domain.k8s.io/zone=us-east-1a

node-3

node-2

failure-domain.k8s.io/zone=us-east-1b
Supported topology-aware features in Kubernetes

- **Node level**
  - Workloads
  - Volumes

- **Within a node**
  - Hardware
Agenda

- Topologies in Kubernetes
- **Requests of topology-aware service routing**
- Solutions and prototypes
- Q&A
Kubernetes Service & Endpoints

Service `10.0.0.11:9376`
Label: `app=MyApp`

Replication Controller
Label: `app=MyApp` Replicas: 2

Label Selector

Label: `app=MyApp`
Label: `app=backend` IP: 172.17.10.1 Port: 80
Container

Label: `app=MyApp`
Label: `app=backend` IP: 172.17.10.2 Port: 80
Container

Endpoints: track backend pod changes
`172.17.10.1:80`
`172.17.10.2:80`
...

Cluster DNS record:
`pod1.clusterdomain`
`pod2.clusterdomain`
...
Topology-aware service routing: user stories

- Clear demand for node-local
  - per-node services: fluentd, aws-es-proxy
  - secure
- “Find zone-local backends for service X”? 
  - data costs
  - performance
- Extend: “locality” means same topological level
  - select a subset of endpoints based on topology
• Hard requests or soft requests?
  - try local, then go wider?
  - always want that one?
• How hard to try?
  - weight per topo
• What if multiple backends satisfy?
  - probabilities
Agenda

- Topologies in Kubernetes
- Requests of topology-aware service routing
- Solutions and prototypes
- Q&A
Solutions and prototypes

API Changes:

```yaml
kind: ServicePolicy
metadata:
  name: service-policy-example
  namespace: foo
spec:
  serviceSelector:
    matchLabels:
      app: bar
  topology:
    key: kubernetes.io/hostname # Any topology key you want
    mode: required/preferred/ignored

---

# Endpoints API changes
type EndpointAddress struct {
  // labels of node hosting the endpoint
  Topology map[string]string
}
```
Data Flow

- **ServicePolicy**
  - selector

- **Services**
  - selector

- **Pods**

- **Endpoints**

- **Kube-proxy**

- **Node**
  - NodeLabels: Topology

- **Endpoints Controller**
  - Same topological level
  - NodeLabels: Topology

- **Node**
Topology-aware service routing

- Running well in Huawei Cloud CCE
- Happy to open source the implementation
  - Proposal: https://github.com/kubernetes/community/pull/1551
Contact US!
LINUX CON
CONTAINER
CLOUD OPEN
CHINA

THINK OPEN

开放性思维