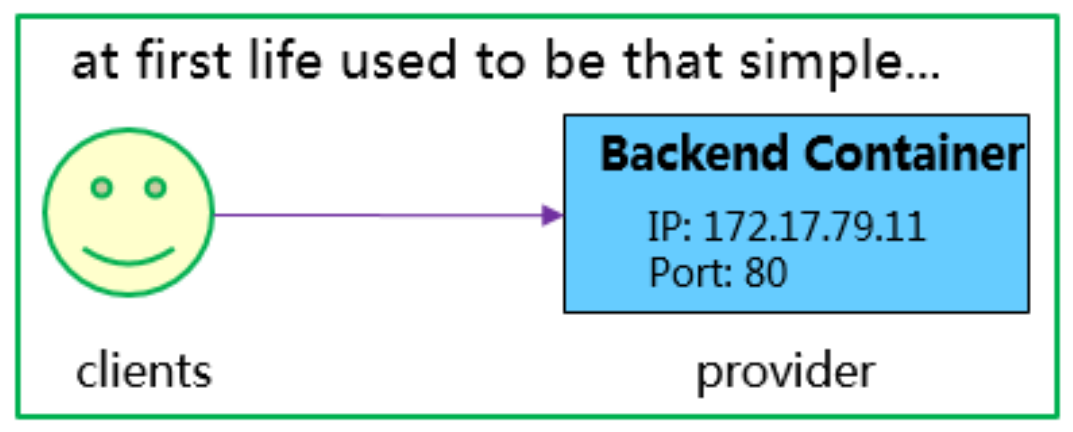
华为云在Kubernetes大规模场景下的Service性能优化实践

王泽锋 @kevin-wangzefeng

HUAWEI

目录

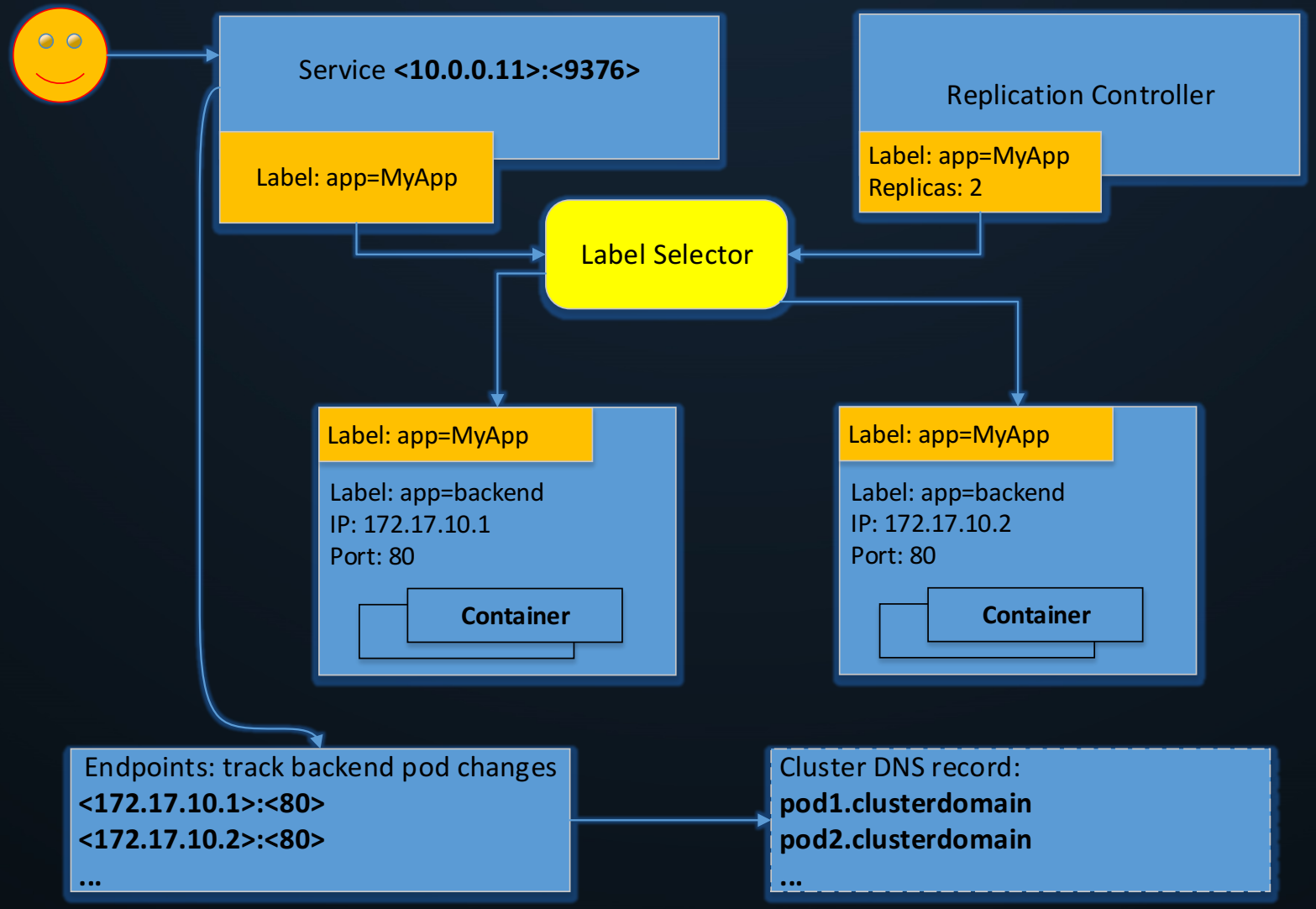
Kubernetes的Service



但，简单的生活总是暂时的：

- 多个后端实例，如何做到负载均衡？
- 如何保持会话亲和性？
- 容器迁移，IP发生变化如何访问？
- 健康检查怎么做？
- 怎么通过域名访问？

Kubernetes Service与Endpoints



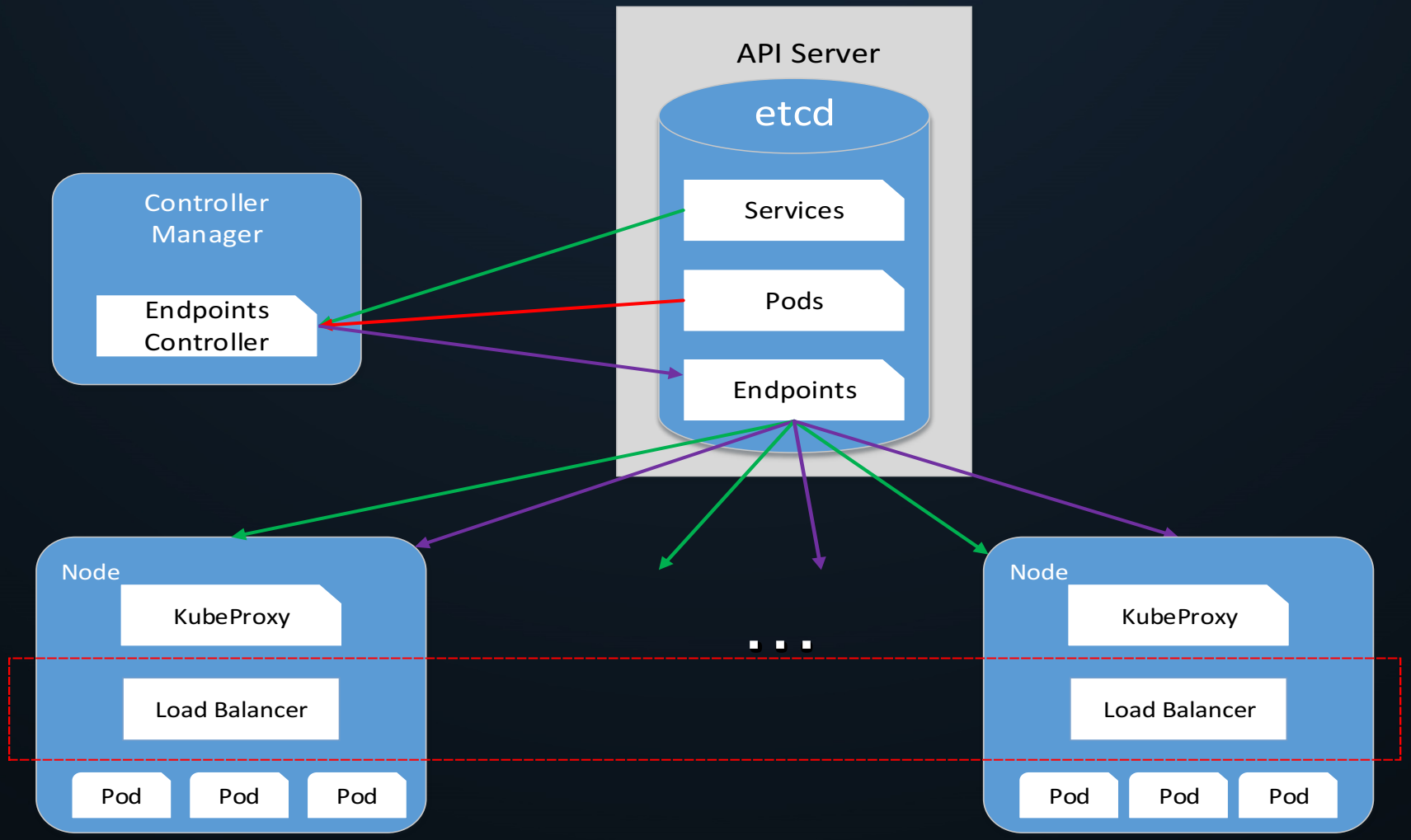
HUAWEI

Kubernetes Service与Endpoints

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: default
spec:
  clusterIP: 10.101.28.148
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 8080
  selector:
    app: nginx
```

```yaml
apiVersion: v1
kind: Endpoints
metadata:
  name: nginx-service
  namespace: default
subsets:
- addresses:
  - ip: 172.17.0.2
    nodeName: 100-106-179-237.node
    targetRef:
      kind: Pod
      name: nginx-rc-c8tw2
      namespace: default
  - ip: 172.17.0.3
    nodeName: 100-106-179-238.node
    targetRef:
      kind: Pod
      name: nginx-rc-x14tv
      namespace: default
  ports:
  - name: http
    port: 8080
    protocol: TCP
```

Service 内部逻辑

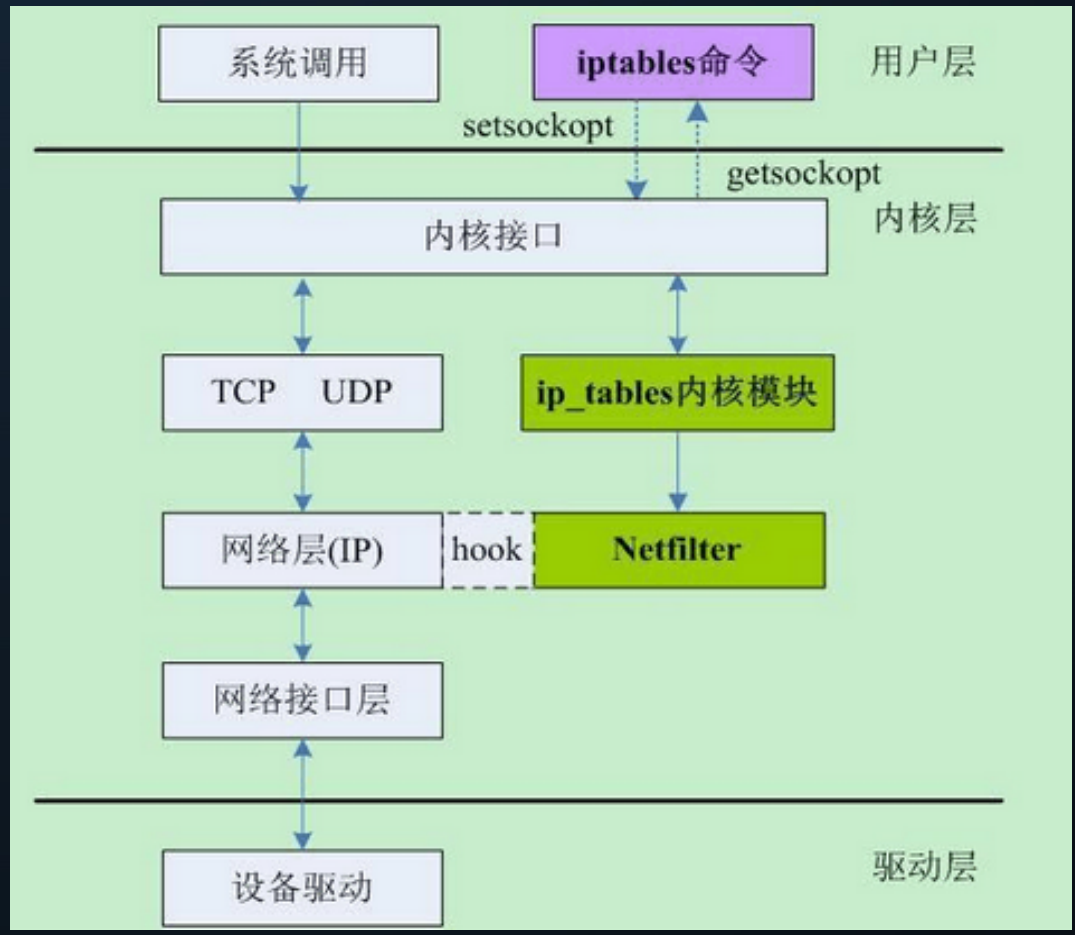


HUAWEI

目录

什么是Iptables？

- Iptables是一个用户态程序，通过配置Netfilter规则表（ Xtables ）来构建linux内核防火墙。
- Netfilter是Linux内核的网络包管理框架，支持自定义包过滤、NAT、端口映射等操作。

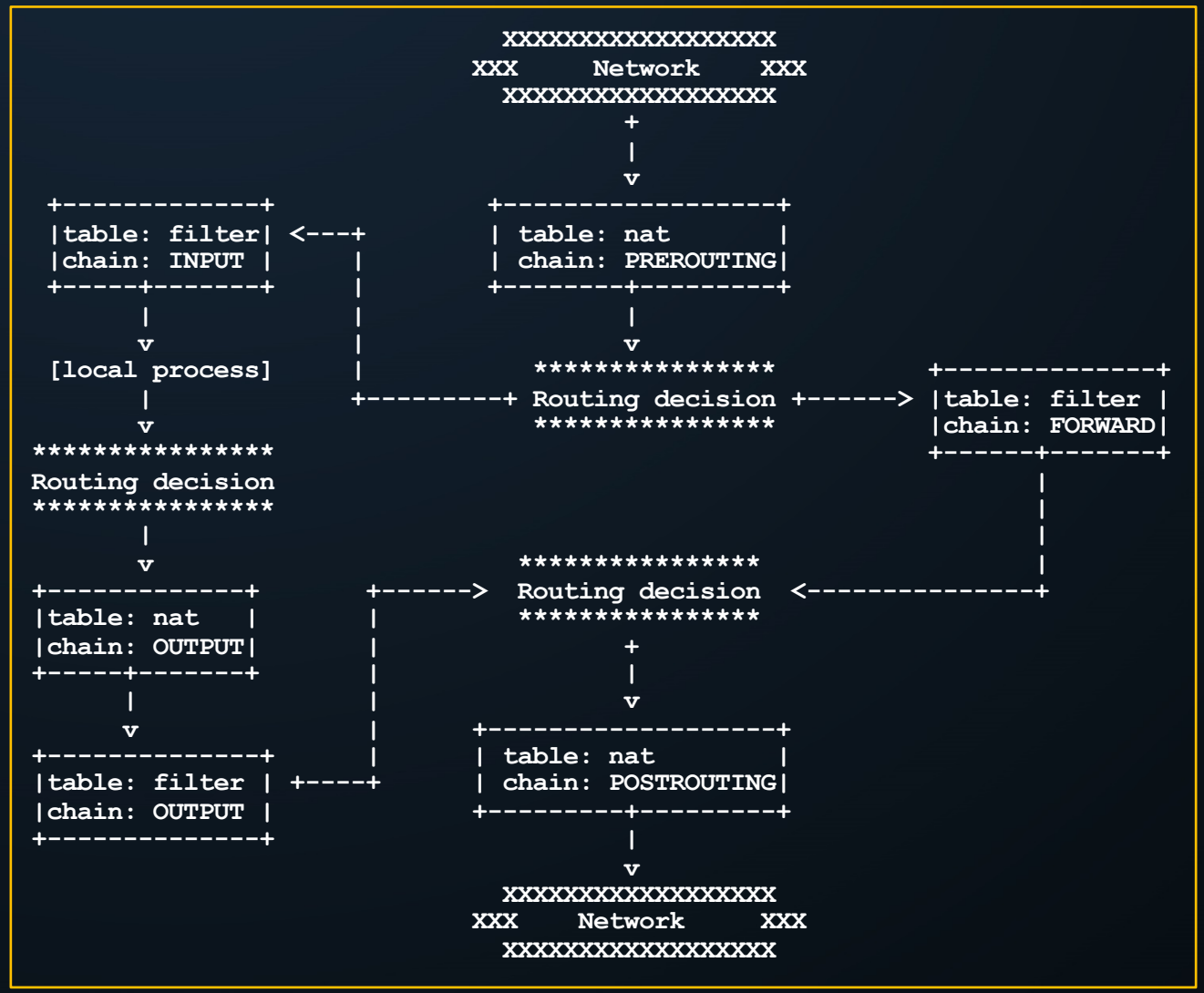


HUAWEI

网络数据包通过 iptables 的过程

从任何网络端口进来的每一个 IP 数据包都要从上到下的穿过这张图：

- 用于本地进程的数据包——从顶端进入，到 <Local Process> 停止；
- 由本地进程生成的数据包——从 <Local Process> 发出，一直向下穿过该流程图。



HUAWEI

Iptables实现流量转发与负载均衡

- **Iptables 如何做流量转发？**

—— DNAT 实现IP地址和端口映射

```
iptables -t nat -A PREROUTING -d 1.2.3.4/32 --dport 80 -j DNAT --to-destination 10.20.30.40:8080
```

- **Iptables 如何做负载均衡？**

—— statistic 模块为每个后端设置权重

```
iptables -t nat -A PREROUTING -d 1.2.3.4 --dport 80 -m statistic --mode random --probability .25
-j DNAT --to-destination 10.20.30.40:8080
```

- **Iptables 如何做会话保持？**

—— recent 模块设置会话保持时间

```
iptables -t nat –A FOO -m recent --rcheck --seconds 3600 --reap --name BAR -j BAR
```

Iptables在Kubernetes的应用举例

VIP:Port -> PREROUTING(OUTPUT) -> KUBE-SERVICES -> KUBE-SVC-XXX

-> KUBE-SEP-XXX -> RIP:Port

1
```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
KUBE-SERVICES  all  --  0.0.0.0/0            0.0.0.0/0
```

2
```
Chain KUBE-SERVICES (2 references)
target     prot opt source               destination
KUBE-SVC-6IM33IEVEEV7U3GP  tcp  --  0.0.0.0/0     10.20.30.40  tcp dpt:80
```

3
```
Chain KUBE-SVC-6IM33IEVEEV7U3GP (1 references)
target     prot opt source               destination
KUBE-SEP-Q3UCPZ54E6Q2R4UT  all  --  0.0.0.0/0            0.0.0.0/0
```

4
```
Chain KUBE-SEP-Q3UCPZ54E6Q2R4UT (1 references)
target     prot opt source               destination
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0       tcp to:172.17.0.2:8080
```

目录

- Kubernetes的Service机制
- Iptables实现Service负载均衡
- 当前Iptables实现存在的问题
- IPVS实现Service负载均衡
- Iptables VS. IPVS

Iptables做负载均衡的问题

- **规则线性匹配时延**

  KUBE-SERVICES链挂了一长串KUBE-SVC-*链；访问每个service，要遍历每条链直到匹配，时间复杂度**O(N)**

- **规则更新时延**

  非增量式

- **可扩展性**

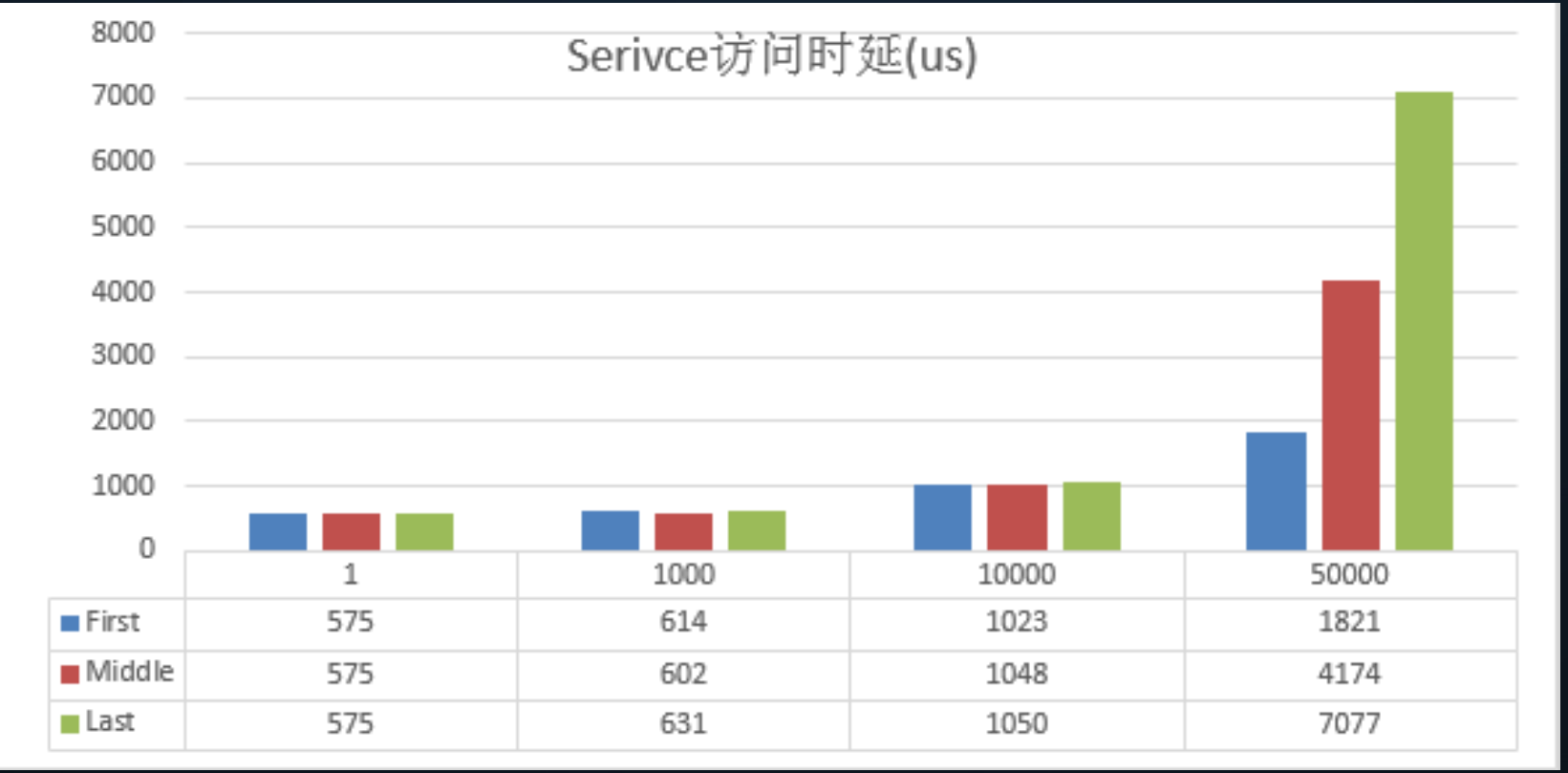
  当系统存在大量iptables规则链时，增加/删除规则会出现kernel lock

  Another app is currently holding the xtables lock. Perhaps you want to use the -w option?

- **可用性**

  后端实例扩容，服务会话保持时间更新等都会导致连接断开。

Iptables规则匹配时延



	1	1000	10000	50000
First	575	614	1023	1821
Middle	575	602	1048	4174
Last	575	631	1050	7077

注：上面测试中，每个service在kube-services对应1条chain

HUAWEI

Iptables规则的更新时延

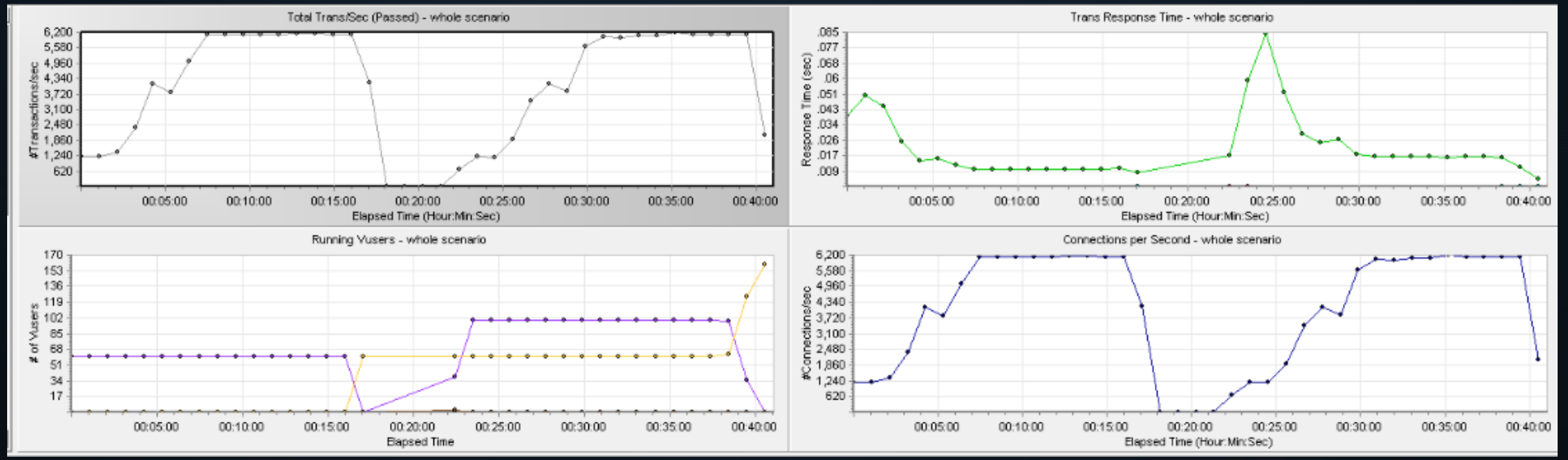
时延出现在哪？

- **非增量式，即使加上--no-flush(iptables-restore)选项**
- **Kube-proxy定期同步iptables状态：**
  - ✓ 拷贝所有规则 iptables-save
  - ✓ 在内存中更新规则
  - ✓ 在内核中修改规则 iptables-restore
  - ✓ 规则更新期间存在kernel lock

5K service（40K 规则），增加一条iptables规则，耗时11min

20K service（160K 规则），增加一条iptables规则，耗时5h

Iptables周期性刷新导致TPS抖动



HUAWEI

K8S Scalability

5000 Nodes

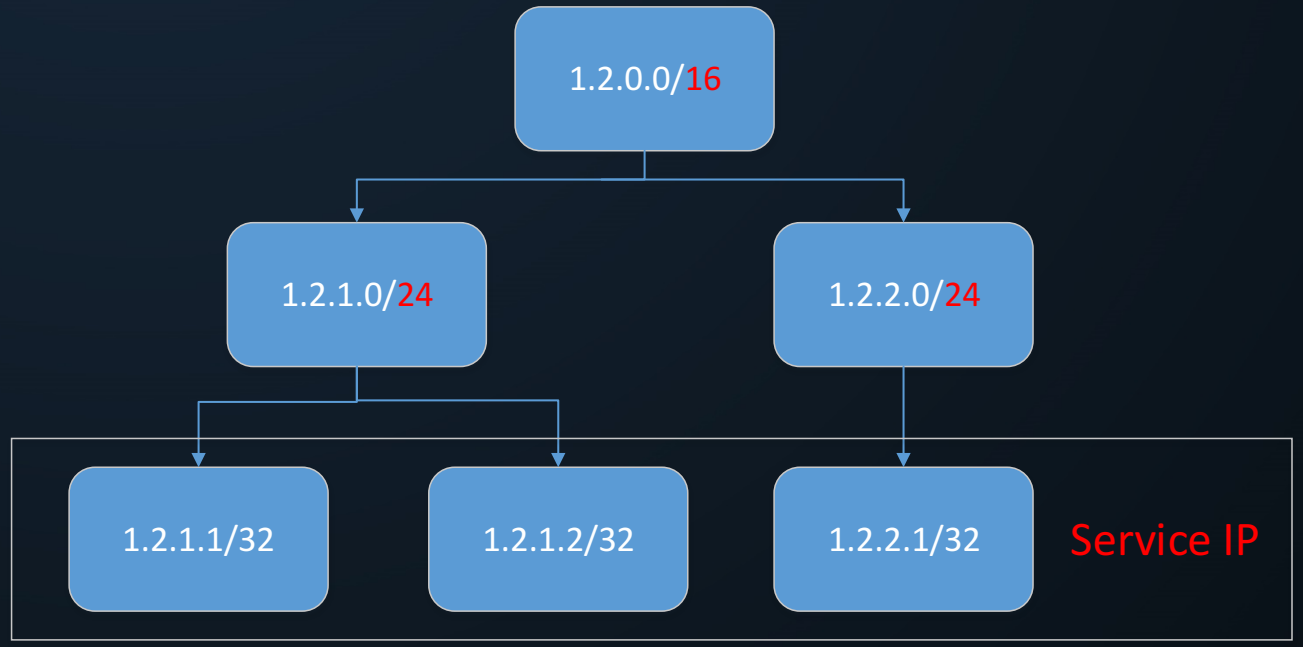
1000 Services ？？

K8S Scalability

5000 Nodes

1000 Services ？？

使用树形结构组织iptables规则 =>>



路由时间复杂度取决于搜索树的高度(m)，时间复杂度O($\sqrt[m]{N}$)

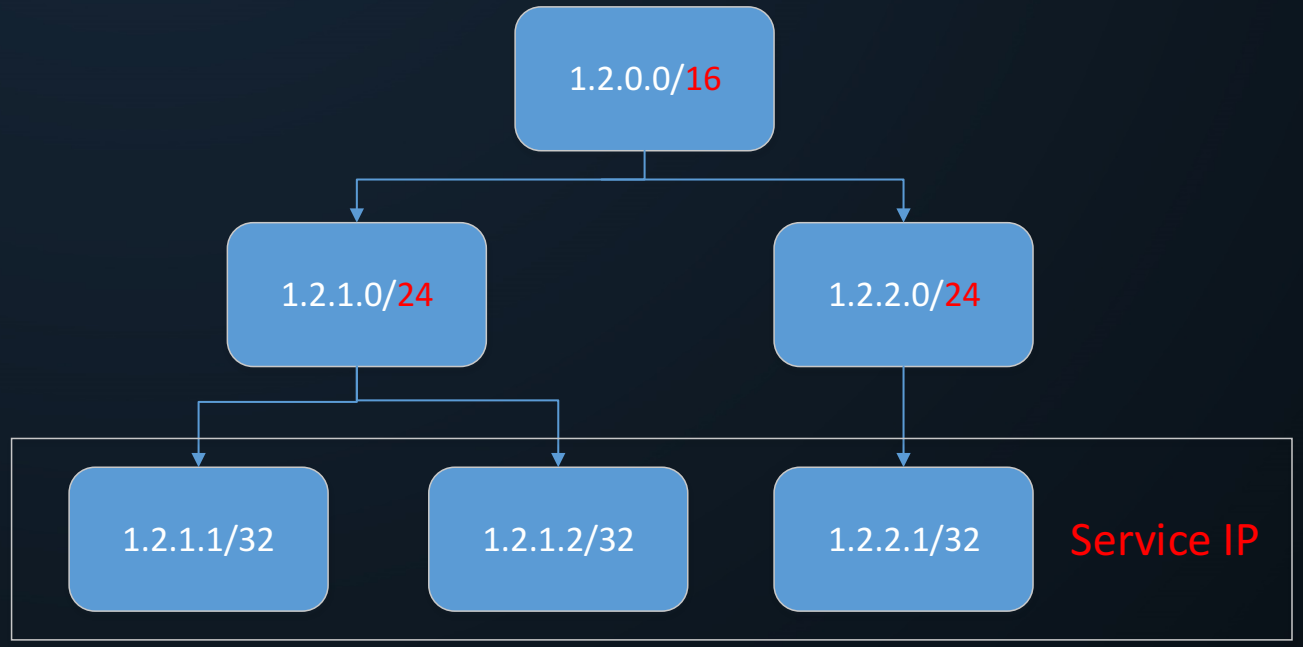
HUAWEI

K8S Scalability

5000 Nodes

1000 Services ？？

使用树形结构组织iptables规则 =>>



路由时间复杂度取决于搜索树的高度(m)，时间复杂度O($\sqrt[m]{N}$)

或者，**IPVS**

HUAWEI

目录

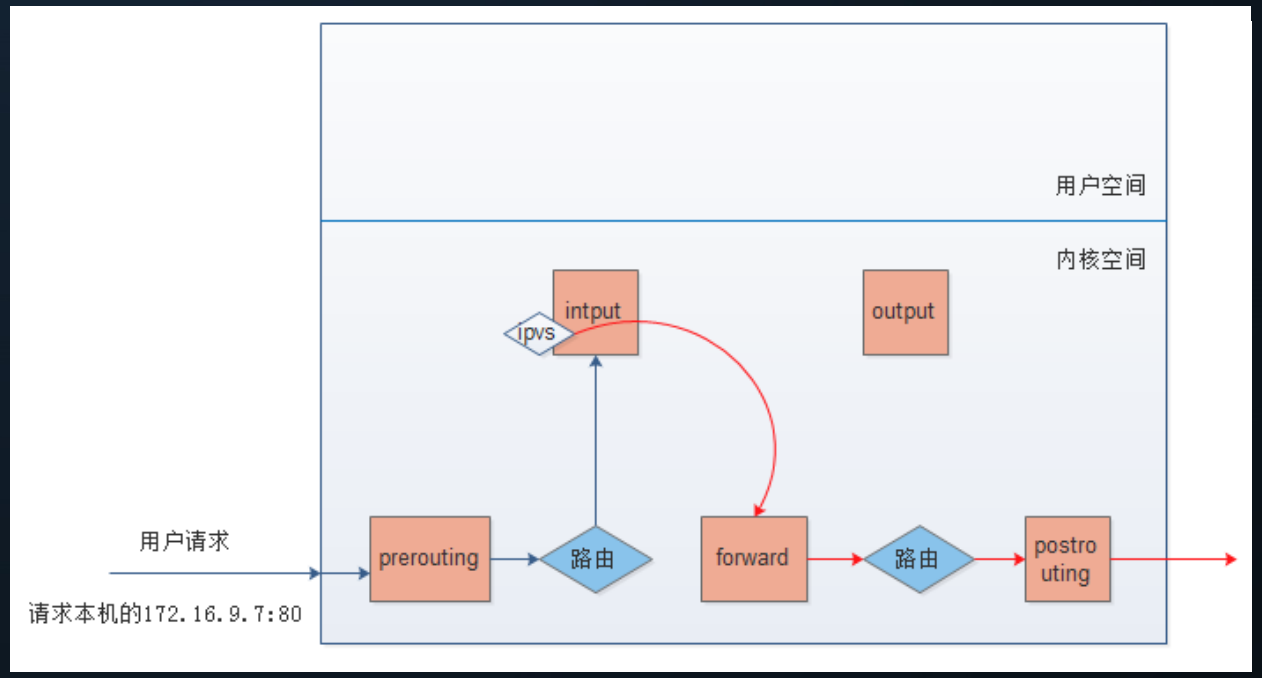
- Kubernetes的Service机制
- Iptables实现Service负载均衡
- 当前Iptables实现存在的问题
- IPVS实现Service负载均衡
- Iptables VS. IPVS

什么是IPVS（IP Virtual Server）

- 传输层Load Balancer，LVS负载均衡器的实现
- 同样基于Netfilter，但使用的是hash表
- 支持TCP, UDP，SCTP协议，IPV4，IPV6
- 支持多种负载均衡策略

  rr, wrr, lc, wlc, sh, dh, lblc...
- 支持会话保持

  persistent connection调度算法



HUAWEI

IPVS的三种转发模式

支持三种LB模式: Direct Routing(DR), Tunneling, NAT

- DR模式工作在L2，最快，但不支持端口映射
- Tunneling模式用IP包封装IP包，不支持端口映射
- DR和Tunneling模式，回程报文不会经过IPVS Director
- NAT模式支持端口映射，回程报文经过IPVS Director - 内核原生版本只做DNAT，不做SNAT

使用IPVS实现流量转发

- **绑定VIP**

  dummy网卡

  ```
  # ip link add dev dummy0 type dummy
  # ip addr add 192.168.2.2/32 dev dummy0
  ```

  本地路由表

  ```
  # ip route add to local 192.168.2.2/32 dev eth0 proto kernel
  ```

  网卡别名

  ```
  # ifconfig eth0:1 192.168.2.2 netmask 255.255.255.255 up
  ```

- **IPVS Virtual Server**

  ```
  # ipvsadm -A -t 192.168.60.200:80 -s rr -p 600
  ```

- **IPVS Real Server**

  ```
  # ipvsadm -a -t 192.168.60.200:80 -r 172.17.1.2:80 –m
  # ipvsadm -a -t 192.168.60.200:80 -r 172.17.2.3:80 –m
  ```

IPVS实现Kubernetes Service

```
Name:              nginx-service
Type:              ClusterIP
IP:                10.102.128.4
Port:              http    80/TCP
Endpoints:         10.244.0.235:8080,10.244.1.237:8080
Session Affinity:   10800

# ip addr
73: kube-ipvs0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 1a:ce:f5:5f:c1:4d brd ff:ff:ff:ff:ff:ff
    inet 10.102.128.4/32 scope global kube-ipvs0
       valid_lft forever preferred_lft forever

# ipvsadm -ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.102.128.4:80 rr persistent 10800
  -> 10.244.0.235:8080            Masq    1      0          0
  -> 10.244.1.237:8080            Masq    1      0          0
```

目录

Iptables vs. IPVS 规则增加时延

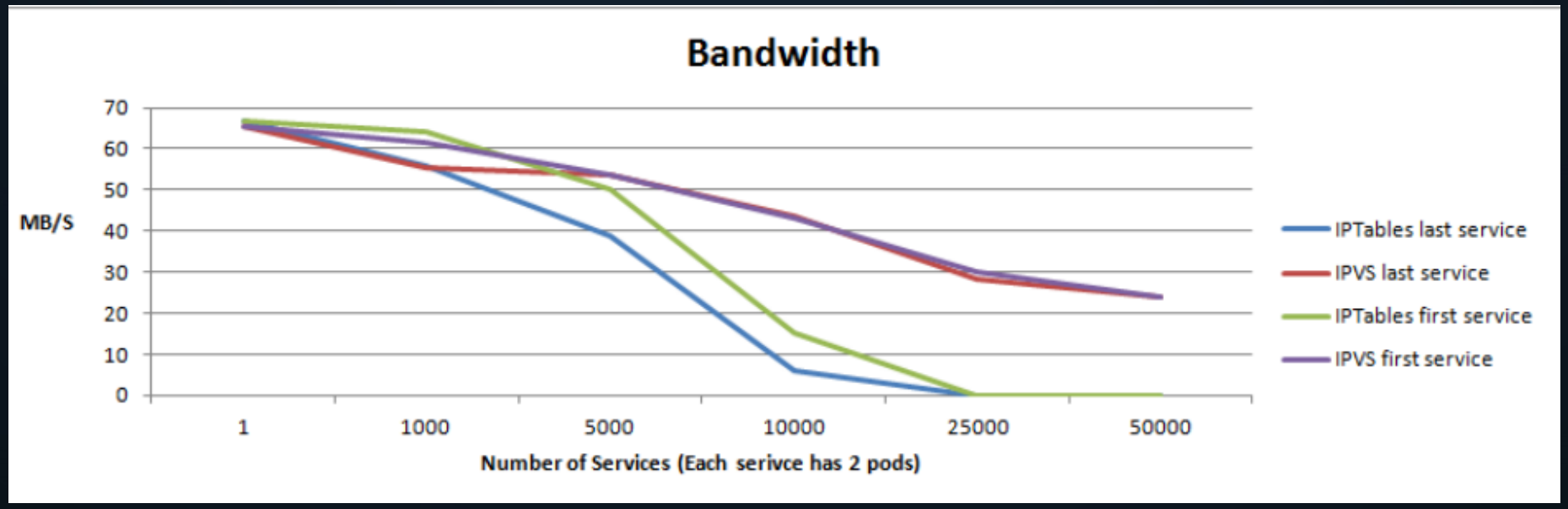
Service基数	1	5,000	20,000
Rules基数	8	40,000	160,000
增加1条Iptables规则	50 us	11 min	5 hours
增加1条IPVS规则	30 us	50 us	70 us

观察结果：

- Iptables模式下：增加一条Iptables的时延，随规则数的增加“指数”上升
- IPVS模式下：增加一条IPVS的时延，几乎不受规则基数影响

Iptables vs. IPVS 网络带宽

- 使用iperf测量
- 每个Service暴露4个端口（KUBE-SERVICES下挂4条KUBE-SVC-*）



service数	1	1000	5000	10000	25000	50000
带宽,iptables,first Service	66.6	64	50	15	0	0
带宽,iptables,last Service	66.6	56	38.6	6	0	0
带宽,IPVS,first Service	65.3	61.7	53.5	43	30	24
带宽,IPVS,last Service	65.3	55.3	53.8	43.5	28.5	23.8

HUAWEI

Iptables vs. IPVS CPU/内存消耗

Metrics	number of services	IPVS	Iptables
Memory Usage	1000	386 MB	1.1 G
	5000	N/A	1.9 G
	10000	542 MB	2.3 G
	15000	N/A	OOM
	50000	1272 MB	OOM
CPU Usage	1000	0%	N/A
	5000		50% - 85%
	10000		50%-100%
	15000		N/A
	50000		N/A

特性社区状态

成熟度：Alpha in 1.8, Beta in v1.9, 即将GA

Maintained by：华为云K8S开源团队

兼容性：

- 支持ClusterIP，NodePort，External IP，Load Balancer...类型Service
- iptables模式的特性，IPVS模式都已支持
- 兼容Network Policy
- SNAT和访问控制仍依赖于iptables实现

抢先体验：

- Kubeadm（需手动加载内核），操作指导

华为云，让客户业务更简单、更高效

HUAWEI