Anolis OS 23 envoy_accel 镜像使用文档

简介

Anolis OS 23 envoy_accel 镜像是一个预先安装了 envoy 的 Anolis OS 23 系统镜像，并且支持 Intel QAT/IAA/DLB 加速 envoy 的 TLS，GZIP 等功能，带来远超于传统 envoy 的性能体验。

Intel QuickAssist Technology（QAT），Intel Analytic Accelerator（IAA）和 Intel Dynamic Load Balancer（DLB）是集成进英特尔 Sapphire Rapids 至强处理器的专用加速器。

- Intel QAT 通过硬件卸载主流加解密、哈希、压缩解压缩算法，Intel QAT 用以显著提升云计算、网络、大数据和存储解决方案的性能和效率。
- Intel IAA 为数据查询分析提供了高吞吐的数据压缩解压缩和数据过滤能力。
- Intel DLB 提供了无锁的硬件队列，可以实现负载均衡的能力，内部工作队列可以原子的、有序的分发负载到最多 64 个消费者处理。

本文档将向您介绍如何使用 Anolis OS 23 envoy_accel 镜像以及如何利用加速器加速 envoy 性能。

系统要求

建议您的 ecs 主机满足如下要求：

- ecs intel 8代系列实例
- 至少拥有 QAT，IAA，DLB 一款加速器

驱动加载

```Bash
modprobe vfio uio mdev udma_drv intel_qat dlb2 qat_4xxx qat_4xxxvf qat_vqat qat_vdcm
```

加速器配置

QAT 查看

1. 查看拥有的 QAT 设备

`adf_ctl status`

```
[root@iZbp1dv0l0t3s3iemyxg8nZ ~]# adf_ctl status
Checking status of all devices.
There is 2 QAT acceleration device(s) in the system:
 qat_dev0 - type: vqat-adi,  inst_id: 0,  node_id: 0,  bsf: 0000:00:0e.0,  #accel: 1 #engines: 1 state: down
 qat_dev1 - type: vqat-adi,  inst_id: 1,  node_id: 0,  bsf: 0000:00:0f.0,  #accel: 1 #engines: 1 state: down
```

如上图所示，当前系统有两个 QAT 设备，其中 `qat_dev0` 用作非对称加解密， `qat_dev1` 用作压缩解压缩（该设定是规定的，在 VM 中无法更改）。

QAT 配置

1. 执行命令，这里以查看到的 `qat_dev0` 和 `qat_dev1` 为例，分别为其添加配置文件（配置必须与设备规定类型一致， `qat_dev0` 用作非对称加解密， `qat_dev1` 用作压缩解压缩）。

```
cp /usr/bin/QAT/build/vqat-adi_dev0.conf.dc /etc/vqat-adi_dev1.conf
cp /usr/bin/QAT/build/vqat-adi_dev0.conf.asym /etc/vqat-adi_dev0.conf
sed -i 's/Dc0IsPolled = 1/Dc0IsPolled = 2/g' /etc/vqat-adi_dev1.conf
sed -i 's/Cy0IsPolled = 1/Cy0IsPolled = 2/g' /etc/vqat-adi_dev0.conf
sed -i 's/\[SSL\]/\[SHIM\]/g' /etc/vqat-adi_dev*
```

2. 执行 `adf_ctl restart`

```
[root@iZbp1dv0l0t3s3iemyxg8nZ ~]# adf_ctl restart
Restarting all devices.
Processing /etc/vqat-adi_dev0.conf
Processing /etc/vqat-adi_dev1.conf
```

3. `adf_ctl status` 查看设备状态，如果显示 up，则配置成功

```
[root@iZbp1dv0l0t3s3iemyxg8nZ ~]# adf_ctl status
Checking status of all devices.
There is 2 QAT acceleration device(s) in the system:
 qat_dev0 - type: vqat-adi,  inst_id: 0,  node_id: 0,  bsf: 0000:00:0e.0,  #accel: 1 #engines: 1 state: up
 qat_dev1 - type: vqat-adi,  inst_id: 1,  node_id: 0,  bsf: 0000:00:0f.0,  #accel: 1 #engines: 1 state: up
```

IAA 查看

1. 查看拥有的 IAA 设备

`ls /sys/bus/dsa/devices | grep -e "iax[0-9]"`

```
[root@iZbp1dv0l0t3s3iemyxg8nZ ~]# ls /sys/bus/dsa/devices | grep -e "iax[0-9]"
iax4
iax5
iax6
iax7
```

根据上图所示，本示例中拥有 IAA 设备为 `iax4-iax7` 。

2. 查看设备可用 work queue

这里以 `iax4` 为例：`ls /sys/bus/dsa/devices/iax4 | grep -e "wq[0-9]"`

```
[root@iZbp1dv0l0t3s3iemyxg8nZ ~]# ls /sys/bus/dsa/devices/iax4 | grep -e "wq[0-9]"
wq4.0
```

根据上图所示，本示例中 `iax4` 拥有一个work queue：`wq4.0`

IAA 配置

1. 执行配置命令，这里以 `iax4` 的 `wq4.0` 为例

```
accel-config config-wq iax4/wq4.0 --type=user -d user -n envoy
accel-config enable-device iax4
accel-config enable-wq iax4/wq4.0
```

2. 如果显示如下信息，则配置成功

```
enabled 1 wq(s) out of 1
```

DLB 查看

1. 查看拥有的 DLB 设备

`ls /dev | grep dlb`

```
[kube@ecs-spr ~]$ ls /dev | grep dlb
dlb0
dlb1
```

根据上图所示，本示例中拥有 DLB 设备为 `dlb0` 和 `dlb1`。

DLB 配置

无需手动配置

envoy 使用加速器说明

QAT TLS

```Bash
transport_socket:
  name: envoy.transport_sockets.tls
  typed_config:
    "@type": type.googleapis.com/envoy.extensions.transport_sockets.tls.v3.DownstreamTlsContext
    common_tls_context:
      tls_certificates:
      - certificate_chain: { "filename": "/etc/envoy/tls/server.crt" }
        private_key_provider:
          provider_name: qat
          typed_config:
            "@type": "type.googleapis.com/envoy.extensions.private_key_providers.qat.v3alpha.QatPrivateKeyMethodConfig"
            poll_delay: 0.002s
            private_key: { filename: "/etc/envoy/tls/server.key" }
```

```Bash
{
  "poll_delay": ...,
  "private_key": ...
}
```

- poll_delay：硬件 poll 延迟，建议设置为 0.002
- private_key：私钥

QAT ZIP （GZIP compress）

```Bash
compressor_library:
  name: text_optimized
  typed_config:
    "@type": type.googleapis.com/envoy.extensions.compression.qatzip.compressor.v3alpha.Qatzip
    hardware_buffer_size: SZ_4K
```

```Bash
{
  "compression_level": ...,
  "hardware_buffer_size": ...,
  "chunk_size": ...,
  "fallback_policy": ...
}
```

- compression_level：同 GZIP
- hardware_buffer_size：默认 SZ_32K（对应window bits 15）（SZ_4K~SZ_512K）
- chunk_size：默认 4096
- fallback_policy：
  - NO_FALLBACK：默认，QATZIP 启动失败后 crash
  - USE_GZIP：QATZIP 启动失败后使用软件 GZIP 代替

IAA ZIP（GZIP decompress）

注意：IAA仅支持 windows bit <= 12 的解压

```Bash
decompressor_library:
  name: upload
  typed_config:
    "@type": "type.googleapis.com/envoy.extensions.compression.iaazip.decompressor.v3alpha.Iaazip"
    chunk_size: 4096

```

```Bash
{
  "chunk_size": ...,
  "fallback_policy": ...
}
```

- chunk_size：默认4096
- fallback_policy：
  - NO_FALLBACK：默认，IAA 启动失败后 crash
  - USE_GZIP：IAA 启动失败后使用软件 GZIP 代替

DLB

注意：使用 DLB 需要 --concurrency 指定 worker 数量，worker 数量不能超过 DLB 设备的 port 数量的一半

Bash

```
connection_balance_config:
  extend_balance:
    name: envoy.network.connection_balance.dlb
    typed_config:
      "@type": type.googleapis.com/envoy.extensions.network.connection_balance.dlb.v3alpha.Dlb
      max_retries: 10
      fallback_policy: NopConnectionBalance
```

Bash

```
{
  "id": ...,
  "max_retries": ...,
  "fallback_policy": ...
}
```

- id：设备号，如果不填写，默认从 0 开始找（这个不需要设定）
- max_retries：最大重试次数，默认为 10
- fallback_policy：
  - None：默认，DLB 启动失败后会触发 crash
  - NopConnectionBalance：DLB 启动失败后 Fallback 到无负载均衡情况
  - ExactConnectionBalance：DLB 启动失败后 Fallback 到软件负载均衡

注意事项

- 请确保您的硬件符合系统要求，以获得最佳的性能和用户体验。
- 请确保您已充分阅读 envoy 的使用许可，并严格按照许可协议使用 envoy 。
- envoy 仍然保持原有软件能力，如果不希望使用硬件加速器加速，按原有 envoy 配置仍可继续享受 envoy 功能。