天数智芯

天数智芯加速卡 K8s 插件使用指南

版本：V3.1 及以上

日期：2024.11.8

适用产品：天垓 100| 天垓 150| 智铠 50| 智铠 100

1 声明

1.1 版权声明

1.2 免责声明

天数智芯可以随时对本文档或本文档中描述的产品进行改进和/或更改。本文档包括与天数智芯产品有关的信息，作为说明典型应用的一种方式, 因此，不一定提供足以进行生产设计的完整信息。对于本文档中内容的准确性或完整性，天数智芯不做任何陈述或保证。

1.3 联系方式

地址：上海市闵行区陈行公路 2168 号 3 幢

电话：021-68886607

网址：www.iluvatar.com

Contents

2 天数智芯加速卡 K8s 插件使用指南

2.1 修订记录

- COREX-K8SPLG10-UG01-06：2024/11/8
  - 天数智芯加速卡 K8s 设备插件使用指南 中的” 更新部署”，补充对执行 delete 命令的说明
- COREX-K8SPLG10-UG01-05：2024/9/24
  - 更新概述 关于 K8s 插件的相关描述
  - 更新 K8s 设备插件 yaml 配置文件名 `iluvatar-device-plugin.yaml` 为 `ix-device-plugin.yaml`

    yaml 配置文件名更新从软件栈 V4.1.0 开始
- COREX-K8SPLG10-UG01-04：2024/8/30
  - 删除 K8s 设备插件 yaml 配置文件 `iluvatar-device-plugin.yaml` 和 K8s 集群资源监控插件 yaml 配置文件 `ix-exporter.yaml` 的具体内容，建议用户直接在 天数智芯官网 进行下载
- COREX-K8SPLG10-UG01-03：2024/7/9
  - K8s 设备插件新增单节点加速卡分配优选功能
  - 部署天数智芯加速卡 K8s 设备插件，区分不同软件栈版本适用的 `iluvatar-device-plugin.yaml` 配置文件内容
- COREX-K8SPLG10-UG01-02：2024/4/19
  - 创建含天数智算软件栈容器的 pod 更新用于创建 pod 使用的 yaml 文件的内容，提醒用户”image” 指的是包含软件栈容器的 docker image 的名称
- COREX-K8SPLG10-UG01-01：2024/4/12

  文档本次发布与上一次发布 (COREX-K8SPLG10-UG01-00) 相比，有以下更新：
  - 更新查询 K8s 插件版本的地址：
    * 软件栈版本为 V3.4.0 及以上时，查询地址为 https://hub.docker.com/r/iluvatarcorex/ix-device-plugin/tags
    * 软件栈版本低于 V3.4.0 时，查询地址为 https://hub.docker.com/r/iluvatarcorex/k8s-device-plugin/tags
  - 更新 `iluvatar-device-plugin.yaml` 配置文件内容

2.2 概述

如果您已有 K8s 集群及管理平台，在集群内节点上安装天数智芯加速卡硬件后，可安装天数智芯加速卡 K8s 插件，将天数智芯加速卡的管理集成到您现有的 K8s 集群管理平台中。在您的 K8s 集群管理平台现有功能支持下，插件可支持如下应用场景：

- 天数智芯加速卡 K8s 设备插件，可根据用户的请求动态地分配天数智芯加速卡资源
- 天数智芯加速卡 K8s 集群资源监控插件，支持集群用户远程实时获取天数智芯加速卡的多项指标

2.3 天数智芯加速卡 K8s 设备插件使用指南

天数智芯加速卡 K8s 设备插件提供以下功能：

1. 建立与 kubelet 的连接并且为天数智芯加速卡注册 iluvatar.ai/gpu 资源
2. 自动发现和监听天数智芯加速卡信息
3. 根据 kubelet 的请求动态地分配天数智芯加速卡资源
4. 检查天数智芯加速卡健康状态并对已注册的天数智芯加速卡状态进行更新
5. 监听 kubelet 状态并对设备插件的状态进行更新
6. 分配 kubernetes 的时候对加速卡进行优选，加速同一个 pod 下，加速卡之间互相通讯的速度

使用开启天数智芯加速卡 K8s 设备插件的天数智算软件栈 Docker 环境，您需要完成以下主要步骤：

1. 部署天数智芯加速卡 K8s 设备插件：您使用天数提供的 `ix-device-plugin.yaml` 文件，在您的 K8s 集群环境中创建天数智芯加速卡 K8s 设备插件
2. 创建含天数智算软件栈容器的 pod：您根据需要修改天数提供的模板 yaml 文件，如修改 pod 名称或容器名称，并使用该 yaml 文件创建 pod

2.3.1 前置条件

项目	条件
硬件	配备天数智芯加速卡的 K8s 集群
软件	Kubernetes 1.10 或以上版本 天数智芯加速卡驱动及软件栈 V1.1.0 或以上版本

2.3.2 部署天数智芯加速卡 K8s 设备插件

1. 在 天数智芯官网 的 **客户支持** > **资源中心**页面下载 `ix-device-plugin.yaml` 文件
   - 该文件中提到的版本号需要与当前使用的天数智算软件栈版本匹配
   - 软件栈版本为 V3.4.0 及以上，请在 https://hub.docker.com/r/iluvatarcorex/ix-device-plugin/tags 页面查询天数智芯提供的 K8s 设备插件的版本
   - 软件栈版本低于 V3.4.0，请在 https://hub.docker.com/r/iluvatarcorex/k8s-device-plugin/tags 页面查询天数智芯提供的 K8s 设备插件的版本
2. 执行以下命令部署天数智芯加速卡 K8s 设备插件

```
$ kubectl create -f ix-device-plugin.yaml
```

3. 检查天数智芯加速卡 K8s 设备插件的部署状态

```
$ kubectl get po -n kube-system -o wide | grep iluvatar
```

比如，可查看到如下状态：

```
iluvatar-device-plugin-b578g                    1/1     Running   0          3h12m
 ↪  10.244.7.21    corex-worker1       <none>           <none>
iluvatar-device-plugin-cvp2q                    1/1     Running   0          3h12m
 ↪  10.244.8.19    corex-worker2       <none>           <none>
iluvatar-device-plugin-sptgs                    1/1     Running   0          3h12m
 ↪  10.244.6.30    corex-worker3       <none>           <none>
```

4. 在设备插件部署完成后，检查 K8s 是否可以枚举出 iluvatar.ai/gpu 资源信息

   以检查步骤 3 里给出的 corex-worker1 为例，执行以下命令：

```
$ kubectl describe node corex-worker1
...
Capacity:
cpu:                80
ephemeral-storage:  1584285364Ki
hugepages-1Gi:      0
hugepages-2Mi:      0
iluvatar.ai/gpu:    2
memory:             263617400Ki
pods:               110
Allocatable:
cpu:                80
ephemeral-storage:  1460077389045
hugepages-1Gi:      0
hugepages-2Mi:      0
iluvatar.ai/gpu:    2
memory:             263515000Ki
pods:               110
...
```

   Tip

   iluvatar.ai/gpu 资源后面的数值应该与 host 机器上的天数智芯加速卡的数量相同，例如，host 上有 8 张天数智芯加速卡，这个数值也应该是 8。

2.3.3 创建含天数智算软件栈容器的 pod

1. 创建 yaml 文件定义一个 pod，并且给这个 pod 指定 iluvatar.ai/gpu 资源

   以下示例文件 `corex-example.yaml` 以"corex-pod" 为 pod 名称，"corex-container" 为容器名称，"corex:{v.r.m}" 为 docker image 名称：

```
apiVersion: v1
kind: Pod
metadata:
  name: corex-pod # 您可自定义 pod 名称
spec:
```

```
  containers:
  - name: corex-container # 您可自定义 Docker 容器名称
    image: corex:{v.r.m} # 创建含天数智算软件栈容器的 pod
    command: ["sleep"]
    args: ["3600"]
    resources:
      limits:
        iluvatar.ai/gpu: 1 # 申请占用一张天数智芯加速卡
```

Important

”image” 指的是含有天数智算软件栈容器的 docker image 的名称。如果您直接使用 `corex-docker-installer-{v.r.m}-10.2-centos7.8.2003-py3.10-x86_64.run` 生成包含软件栈的容器，则 docker image 的名称为”corex:{v.r.m}”；如果您使用自己的 image 并在其中生成了软件栈容器，则需要改成您自己的 docker image 的名称。

2. 使用该 yaml 文件部署 pod

   以步骤 1 中的 `corex-example.yaml` 为例：

```
$ kubectl create -f corex-example.yaml
```

2.3.4 检查 pod 状态

1. 检查 pod 状态

```
$ kubectl get po -o wide
```

以上述步骤创建的 corex-pod 为例，得到以下信息：

```
NAME          READY    STATUS    RESTARTS   AGE     IP             NODE
 ↪  NOMINATED  NODE   READINESS  GATES
corex-pod      1/1      Running   0          61s     10.244.6.31    corex-worker1
 ↪  <none>                       <none>
```

2. 检查天数智芯加速卡资源分配情况

   以天垓 150 加速卡和上述步骤创建的 corex-pod 为例，得到以下信息：

```
$ kubectl exec -it corex-pod /bin/bash
$ ixsmi
Timestamp    Day MM DD HH:mm:ss YYYY
+-----------------------------------------------------------------------------+
|  IX-ML: v.r.m       Driver Version: v.r.m       CUDA Version: 10.2          |
|-------------------------------+----------------------+----------------------|
| GPU  Name                     | Bus-Id               | Clock-SM  Clock-Mem  |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage    | GPU-Util  Compute M. |
|===============================+======================+======================|
| 0    Iluvatar BI-V150         | 00000000:00:05.0     | 1500MHz   1600MHz    |
```

```
| 0%   33C   P0    N/A / N/A    | 114MiB / 32768MiB    | 0%        Default    |
+-------------------------------+----------------------+----------------------+
| 1    Iluvatar BI-V150         | 00000000:00:06.0     | 1500MHz   1600MHz    |
| 0%   32C   P0    87W / 300W   | 114MiB / 32768MiB    | 0%        Default    |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU        PID      Process name                                Usage(MiB) |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

2.3.5 更新部署

对于以下两种情况，您需要重启这个节点上的 pod 部署：

- 节点上的天数智芯加速卡数量有增减
- 节点上的天数智芯加速卡驱动有更新

使用以下命令更新 pod 部署：

```
kubectl delete pod iluvatar-device-plugin-<id> -n kube-system
```

以上文创建的天数智芯加速卡 K8s 设备插件的部署为例：

如 corex-worker1（iluvatar-device-plugin-b578g）节点上的天数智芯加速卡数量有增减，或天数智芯加速卡驱动有更新，您需要执行以下命令更新 pod 部署：

```
kubectl delete pod iluvatar-device-plugin-b578g -n kube-system
```

Note

执行 delete 后 pod 会自动重启。

2.3.6 删除部署

2.3.6.1 删除 pod

您可以通过以下任一方法删除 pod：

- `$ kubectl delete -f <pod_yaml>.yaml`

- `$ kubectl delete po <pod_name>`

以上文使用 `corex-example.yaml` 创建的 corex-pod 为例：

- `$ kubectl delete -f corex-example.yaml`
- `$ kubectl delete po corex-pod`

2.3.6.2 删除天数智芯加速卡 K8s 设备插件

```
$ kubectl delete -f ix-device-plugin.yaml
```

2.3.7 常见问题

1. 默认情况下，K8s 不推荐将设备插件部署到 master 节点上，这也是 `ix-device-plugin.yaml` 的设置。但是，如果您想将设备插件部署到 master 节点上，您可以使用以下任一方法修改默认设置。
   - 方法一：删除 master 节点上的 taint

     ```
     $ kubectl taint node <Master-node-name> node-role.kubernetes.io/master-
     ```

   - 方法二：在 `ix-device-plugin.yaml` 里面加上 toleration。如果已经部署好设备插件，您需要重新部署。

     ```
     tolerations:
     - key: node-role.kubernetes.io/master
       operator: Exists
       effect: NoSchedule
     ```

2.4 天数智芯加速卡 K8s 集群资源监控插件使用指南

天数智芯加速卡 K8s 集群资源监控插件 ix-Exporter 是一个 http 后台应用程序，使得集群用户可以远程实时获取天数智芯加速卡的以下指标：

监控指标	说明
ix_temperature	GPU 温度（C）
ix_fan_speed	GPU 风扇速度
ix_sm_clock	SM 的时钟频率（MHz）

监控指标	说明
ix_mem_clock	Memory 的时钟频率（MHz）
ix_mem_total	Memory 总的大小值（MiB）
ix_mem_used	Memory 已使用值（MiB）
ix_mem_free	Memory 可使用值（MiB）
ix_mem_utilization	Memory 已使用百分比（%）
ix_gpu_utilization	GPU 已使用百分比（%）
ix_power_usage	GPU 使用能耗（W）

2.4.1 前置条件

项目	条件
硬件	配备天数智芯加速卡的 K8s 集群
软件	Kubernetes 1.10 或以上版本 天数智芯加速卡驱动及软件栈 V3.1.0 或以上版本

2.4.2 部署 ix-Exporter

1. 在 天数智芯官网 的 **客户支持** > **资源中心**页面下载 `ix-exporter.yaml` 文件
   - 该文件中提到的版本号需要与当前使用的天数智算软件栈版本匹配
   - 软件栈版本为 V3.4.0 及以上，请在 https://hub.docker.com/r/iluvatarcorex/ix-device-plugin/tags 页面查询天数智芯提供的 K8s 集群资源监控插件的版本
   - 软件栈版本低于 V3.4.0，请在 https://hub.docker.com/r/iluvatarcorex/k8s-device-plugin/tags 页面查询天数智芯提供的 K8s 集群资源监控插件的版本
2. 执行以下命令部署 ix-Exporter

```
$ kubectl create -f ix-exporter.yaml
```

3. 检查检查部署后的状态

```
$ kubectl get svc
```

比如，可查看到如下状态：

```
NAME          TYPE        CLUSTER-IP        EXTERNAL-IP   PORT(S)           AGE
ix-exporter   NodePort    169.169.228.153   <none>        32021:32021/TCP   23s
```

2.4.3 通过 http 获取天数智芯加速卡信息

通过 http 获取天数智芯加速卡信息，以端口 32021 为例：

```
$ curl http://<master-node-ip>:32021/metrics
```

您将得到天数智芯加速卡 K8s 集群资源信息，例如：

```
# HELP ix_fan_speed Fan speed of iluvatar GPU
# TYPE ix_fan_speed gauge
ix_fan_speed{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 0
ix_fan_speed{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 0
# HELP ix_gpu_utilization The utilization of iluvatar GPU (%).
# TYPE ix_gpu_utilization gauge
ix_gpu_utilization{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 4
ix_gpu_utilization{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 0
# HELP ix_mem_clock Mem clock of iluvatar GPU (MHz).
# TYPE ix_mem_clock gauge
ix_mem_clock{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 1200
ix_mem_clock{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 1200
# HELP ix_mem_free The free physical memory of iluvatar GPU (MiB).
# TYPE ix_mem_free gauge
ix_mem_free{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 32252
ix_mem_free{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 32255
# HELP ix_mem_total The total physical memory of iluvatar GPU (MiB).
# TYPE ix_mem_total gauge
ix_mem_total{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 32768
ix_mem_total{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 32768
# HELP ix_mem_used The used physical memory of iluvatar GPU (MiB).
# TYPE ix_mem_used gauge
ix_mem_used{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 515
ix_mem_used{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 513
# HELP ix_mem_utilization The memory utilization of iluvatar GPU (%).
# TYPE ix_mem_utilization gauge
```

```
ix_mem_utilization{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 2
ix_mem_utilization{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 2
# HELP ix_power_usage The power usage of iluvatar GPU.
# TYPE ix_power_usage gauge
ix_power_usage{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 34
ix_power_usage{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 35
# HELP ix_sm_clock Sm clock of iluvatar GPU (MHz).
# TYPE ix_sm_clock gauge
ix_sm_clock{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 1000
ix_sm_clock{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 1000
# HELP ix_temperature The temperature of iluvatar GPU (C).
# TYPE ix_temperature gauge
ix_temperature{container="",gpu="0",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-79a0ab98-5b7b-4e9f-99c1-e79f2c885554"} 38
ix_temperature{container="",gpu="1",name="Iluvatar BI-
↪  V150",namespace="",pod="",uuid="GPU-118a11b9-6d38-41ff-8fd8-de945ad18a08"} 39
```

2.4.4 更新部署

对于以下两种情况，您需要重启这个节点上的 pod 部署：

- 节点上的天数智芯加速卡数量有增减
- 节点上的天数智芯加速卡驱动有更新

使用以下命令更新 pod 部署：

```
kubectl delete pod ix-exporter-<id> -n kube-system
```

以上文创建的天数智芯加速卡 K8s 设备插件的部署为例：

如 ix-exporter-nlsm7 节点上的天数智芯加速卡数量有增减，或天数智芯加速卡驱动有更新，您需要执行以下命令更新 pod 部署：

```
kubectl delete pod ix-exporter-nlsm7 -n kube-system
```

3 商标声明

- 天数智芯、天数智芯 logo、Iluvatar CoreX 等商标、标识、组合商标为上海天数智芯半导体有限公司之注册商标或商标，受法律保护。
- 除了天数智芯的注册商标外，本内容中使用的其他产品名称及标志仅用于识别目的，该等名称及标志可能是归属于其各自公司的商标。我们否认对该等名称及标志的所有权利。
- CentOS 标识为 Red Hat 公司的商标。
- Docker 为 Docker 公司在美国和其他国家的商标或注册商标。
- Linux 为 Linus Torvalds 在美国和其它国家的注册商标。
- NVIDIA 和 CUDA 为 NVIDIA 公司在美国和/或其它国家的商标和/或注册商标。
- PyTorch 为 Facebook 公司的商标。
- TensorFlow 为 Google 公司的商标。
- Ubuntu 为 Canonical 公司的注册商标。