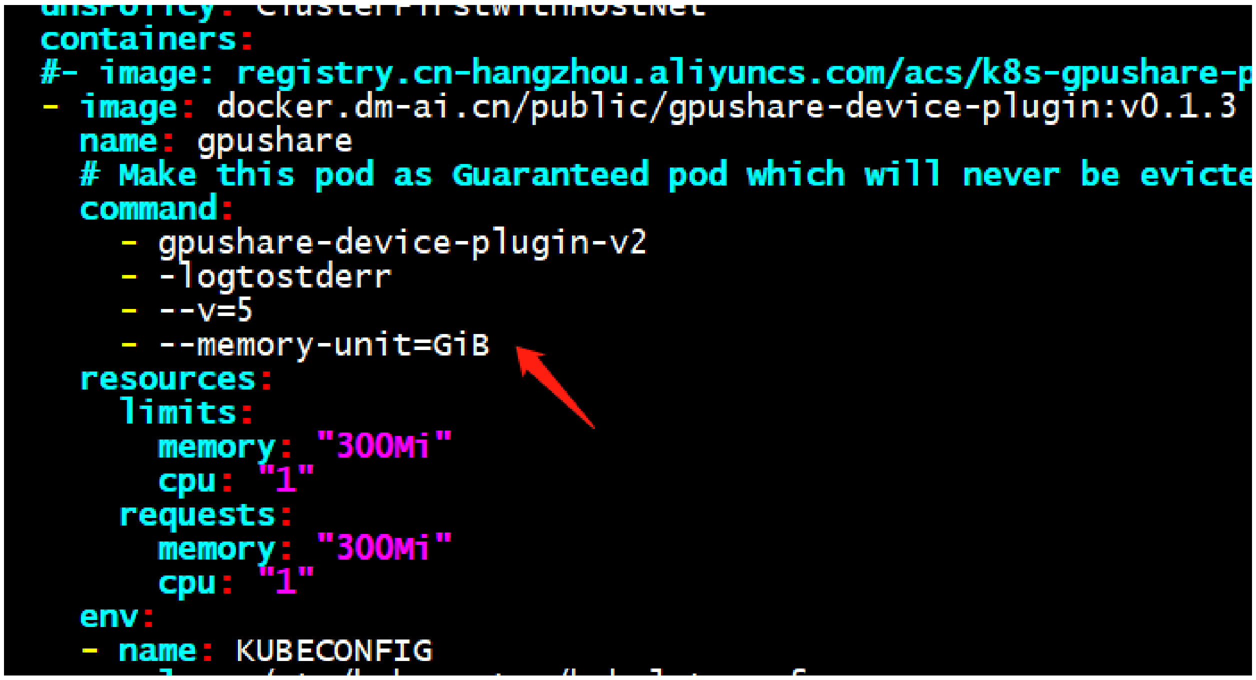
GPU 显存支持按照 MiB 分配资源

1. 背景

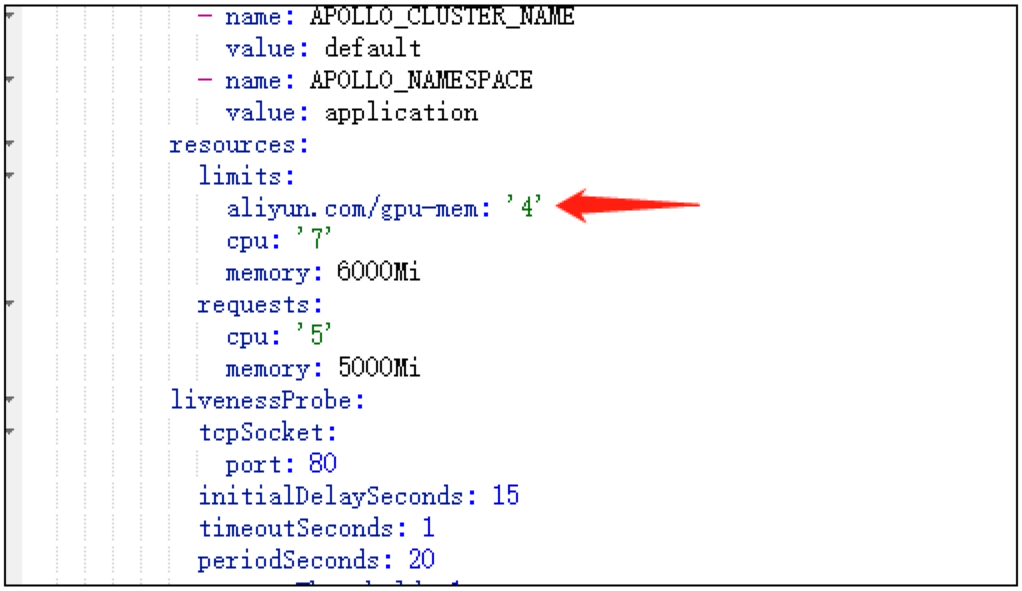
使用阿里云的插件，https://github.com/AliyunContainerService/gpushare-scheduler-extender

安装插件的 yaml 中，显存的使用是 GiB 为单位



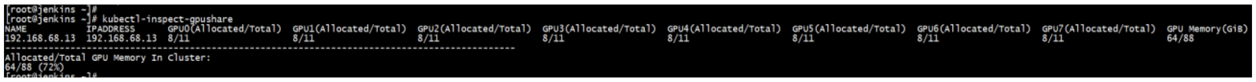
```
  containers:
  #- image: registry.cn-hangzhou.aliyuncs.com/acs/k8s-gpushare-p
  - image: docker.dm-ai.cn/public/gpushare-device-plugin:v0.1.3
    name: gpushare
    # Make this pod as Guaranteed pod which will never be evicte
    command:
      - gpushare-device-plugin-v2
      - -logtostderr
      - --v=5
      - --memory-unit=GiB
    resources:
      limits:
        memory: "300Mi"
        cpu: "1"
      requests:
        memory: "300Mi"
        cpu: "1"
    env:
    - name: KUBECONFIG
```

此时资源申请如下，表示使用 4GB 显存



```
            - name: APOLLO_CLUSTER_NAME
              value: default
            - name: APOLLO_NAMESPACE
              value: application
          resources:
            limits:
              aliyun.com/gpu-mem: '4'
              cpu: '7'
              memory: 6000Mi
            requests:
              cpu: '5'
              memory: 5000Mi
          livenessProbe:
            tcpSocket:
              port: 80
            initialDelaySeconds: 15
            timeoutSeconds: 1
            periodSeconds: 20
```

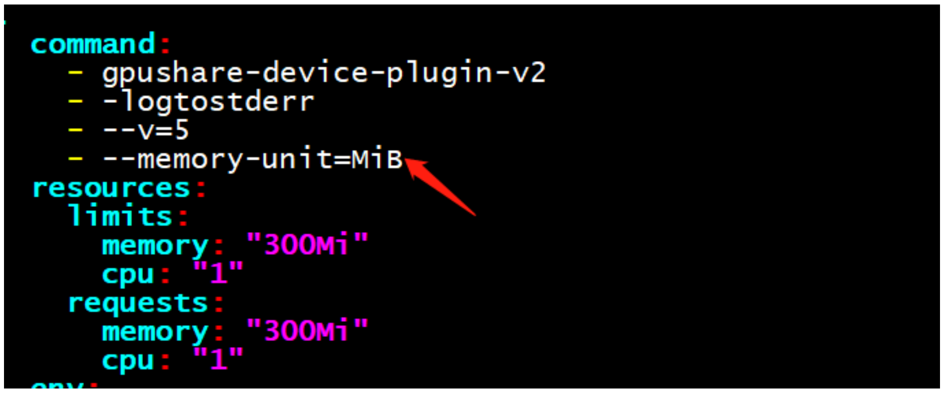
查看显存大小时，也是以 GiB 为单位，效果类似如下：



```
[root@jenkins ~]#
[root@jenkins ~]# kubectl-inspect-gpushare
NAME           IPADDRESS      GPU0(Allocated/Total)  GPU1(Allocated/Total)  GPU2(Allocated/Total)  GPU3(Allocated/Total)  GPU4(Allocated/Total)  GPU5(Allocated/Total)  GPU6(Allocated/Total)  GPU7(Allocated/Total)  GPU Memory(GiB)
192.168.68.13  192.168.68.13  8/11                   8/11                   8/11                   8/11                   8/11                   8/11                   8/11                   8/11                   64/88
-----------------------------------------------------------------------------------------------
Allocated/Total GPU Memory In Cluster:
64/88 (72%)
[root@jenkins ~]#
```

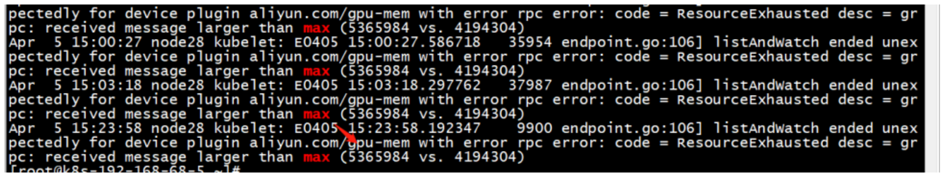
2. 变更

将脚本中 GiB 改为 MiB 后



```
command:
  - gpushare-device-plugin-v2
  - -logtostderr
  - --v=5
  - --memory-unit=MiB
resources:
  limits:
    memory: "300Mi"
    cpu: "1"
  requests:
    memory: "300Mi"
    cpu: "1"
```

无法识别显卡，对应节点的 kubelet 报错：



```
pectedly for device plugin aliyun.com/gpu-mem with error rpc error: code = ResourceExhausted desc = gr
pc: received message larger than max (5365984 vs. 4194304)
Apr  5 15:00:27 node28 kubelet: E0405 15:00:27.586718   35954 endpoint.go:106] listAndWatch ended unex
pectedly for device plugin aliyun.com/gpu-mem with error rpc error: code = ResourceExhausted desc = gr
pc: received message larger than max (5365984 vs. 4194304)
Apr  5 15:03:18 node28 kubelet: E0405 15:03:18.297762   37987 endpoint.go:106] listAndWatch ended unex
pectedly for device plugin aliyun.com/gpu-mem with error rpc error: code = ResourceExhausted desc = gr
pc: received message larger than max (5365984 vs. 4194304)
Apr  5 15:23:58 node28 kubelet: E0405 15:23:58.192347    9900 endpoint.go:106] listAndWatch ended unex
pectedly for device plugin aliyun.com/gpu-mem with error rpc error: code = ResourceExhausted desc = gr
pc: received message larger than max (5365984 vs. 4194304)
```

```
Apr  5 15:03:18 node28 kubelet: E0405 15:03:18.297762   37987 endpoint.go:106] listAndWatch ended unexpectedly for device plugin aliyun.com/gpu-mem with error rpc error: code = ResourceExhausted desc = grpc: received message larger than max (5365984 vs. 4194304)
```

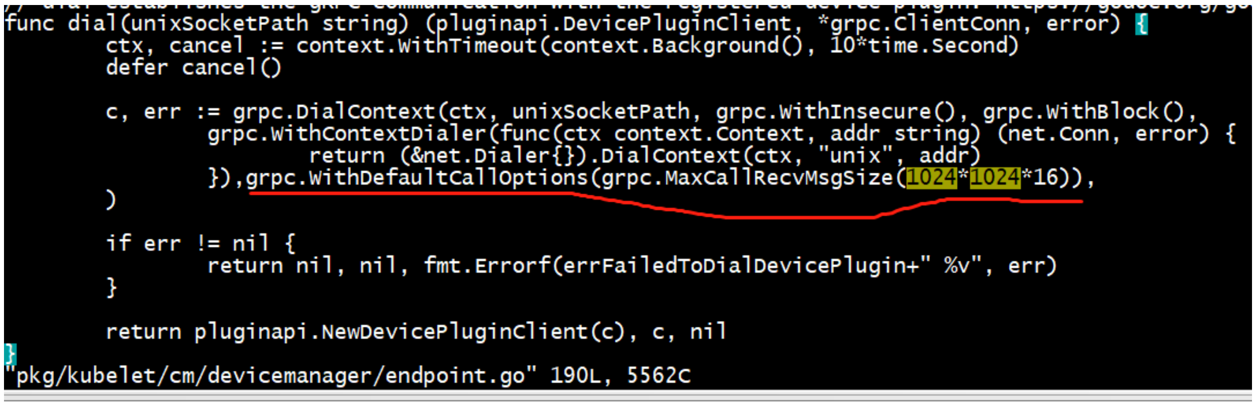
这是因为 GRPC 源码中，接收的 grpc 消息最大报文是 1024*1024*4=4MiB，消息体超过了最大大小 4MB。

3. 解决办法

重新编译 kubelet 源码，替换原有 kubelet。修改代码的地方是 https://github.com/kubernetes/kubernetes/blob/master/pkg/kubelet/cm/devicemanager/endpoint.go 中，

dial 方法中添加参数 grpc.WithDefaultCallOptions(grpc.MaxCallRecvMsgSize(1024*1024*16))

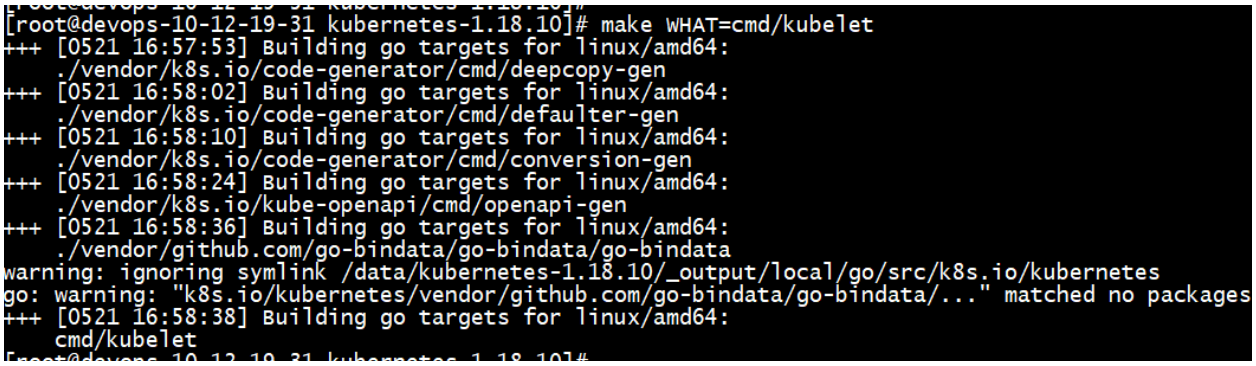
如下图：



```
func dial(unixSocketPath string) (pluginapi.DevicePluginClient, *grpc.ClientConn, error) {
        ctx, cancel := context.WithTimeout(context.Background(), 10*time.Second)
        defer cancel()

        c, err := grpc.DialContext(ctx, unixSocketPath, grpc.WithInsecure(), grpc.WithBlock(),
                grpc.WithContextDialer(func(ctx context.Context, addr string) (net.Conn, error) {
                        return (&net.Dialer{}).DialContext(ctx, "unix", addr)
                }),grpc.WithDefaultCallOptions(grpc.MaxCallRecvMsgSize(1024*1024*16)),
        )

        if err != nil {
                return nil, nil, fmt.Errorf(errFailedToDialDevicePlugin+" %v", err)
        }

        return pluginapi.NewDevicePluginClient(c), c, nil
}
"pkg/kubelet/cm/devicemanager/endpoint.go" 190L, 5562C
```

重新编译 kubelet

make WHAT=cmd/kubelet



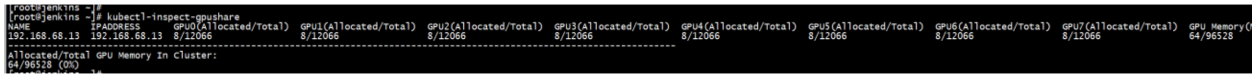
```
[root@devops-10-12-19-31 kubernetes-1.18.10]# make WHAT=cmd/kubelet
+++ [0521 16:57:53] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/deepcopy-gen
+++ [0521 16:58:02] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/defaulter-gen
+++ [0521 16:58:10] Building go targets for linux/amd64:
    ./vendor/k8s.io/code-generator/cmd/conversion-gen
+++ [0521 16:58:24] Building go targets for linux/amd64:
    ./vendor/k8s.io/kube-openapi/cmd/openapi-gen
+++ [0521 16:58:36] Building go targets for linux/amd64:
    ./vendor/github.com/go-bindata/go-bindata/go-bindata
warning: ignoring symlink /data/kubernetes-1.18.10/_output/local/go/src/k8s.io/kubernetes
go: warning: "k8s.io/kubernetes/vendor/github.com/go-bindata/go-bindata/..." matched no packages
+++ [0521 16:58:38] Building go targets for linux/amd64:
    cmd/kubelet
```

编译成功的二进制文件路径：_output/bin/kubelet

将生成的 kubelet 二进制文件进行替换，重启 kubelet

4. 效果

上述步骤搞完后，查看 gpu 情况：



```
[root@jenkins ~]# kubectl-inspect-gpushare
NAME           IPADDRESS      GPU0(Allocated/Total)  GPU1(Allocated/Total)  GPU2(Allocated/Total)  GPU3(Allocated/Total)  GPU4(Allocated/Total)  GPU5(Allocated/Total)  GPU6(Allocated/Total)  GPU7(Allocated/Total)  GPU Memory(
192.168.68.13  192.168.68.13  8/12066                8/12066                8/12066                8/12066                8/12066                8/12066                8/12066                8/12066                64/96528
----------------------------------------------------------------------------------------------------------
Allocated/Total GPU Memory In Cluster:
64/96528 (0%)
```