Cloud Native Batch Computing Platform Volcano

Yang Wang, Contributor of Volcano Community

Volcano 安装部署

1. **安装k8s集群：**
   - git clone git@github.com:volcano-sh/volcano.git
   - kind create cluster --name my-k8s –config hack/e2e-kind-config.yaml
   - Kind的安装与使用参见：https://kind.sigs.k8s.io/

2. **安装Volcano:**

   使用helm安装：
   - 添加Volcano helm仓：helm repo add volcano-sh https://volcano-sh.github.io/helm-charts
   - 安装Volcano：helm install volcano volcano-sh/volcano -n volcano-system --create-namespace

   通过yaml安装：
   - kubectl apply -f https://raw.githubusercontent.com/volcano-sh/volcano/master/installer/volcano-development.yaml

01 运行TensorFlow 作业

02 运行Pytorch作业

03 运行Spark作业

VOLCANO
CLOUD NATIVE COMPUTING FOUNDATION
THE LINUX FOUNDATION 开源软件学园

TensorFlow 作业配置

```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: tensorflow-benchmark
  labels:
    "volcano.sh/job-type": "Tensorflow"

spec:
  minAvailable: 3
  schedulerName: volcano
  plugins:
    env: []
    svc: []

  policies:
    - event: PodEvicted
      action: RestartJob
```

根据业务需求设置作业类型

开启SSH免密码认证

用于创建headless service解决ps和worker之间的通信问题

配置作业重启策略

TensorFlow 作业配置

```
tasks:
    - replicas: 1
      name: ps
      template:
        spec:
          imagePullSecrets:
            - name: default-secret
          containers:
            - command:
                - sh
                - -c
                - |
                  PS_HOST=`cat /etc/volcano/ps.host | sed 's/$/&:2222/g' | tr "\n" ","`;
                  WORKER_HOST=`cat /etc/volcano/worker.host | sed 's/$/&:2222/g' | tr "\n" ","`;
                  python tf_cnn_benchmarks.py --batch_size=32 --model=resnet50 --
variable_update=parameter_server --flush_stdout=true --num_gpus=1 --
local_parameter_device=cpu --device=cpu --data_format=NHWC --job_name=ps --
task_index=${VK_TASK_INDEX} --ps_hosts=${PS_HOST} --worker_hosts=${WORKER_HOST}
              image: volcanosh/example-tf:0.0.1
              name: tensorflow
              ports:
                - containerPort: 2222
                  name: tfjob-port
              workingDir: /opt/tf-benchmarks/scripts/tf_cnn_benchmarks
          restartPolicy: OnFailure
```

PS的运行配置

暴露2222端口

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

TensorFlow 作业配置

```
- replicas: 2
    name: worker
    policies:
      - event: TaskCompleted
        action: CompleteJob
    template:
      spec:
        imagePullSecrets:
          - name: default-secret
        containers:
          - command:
              - sh
              - -c
              - |
                PS_HOST=`cat /etc/volcano/ps.host | sed 's/$/&:2222/g' | tr "\n" ","`;
                WORKER_HOST=`cat /etc/volcano/worker.host | sed 's/$/&:2222/g' | tr "\n" ","`;
                python tf_cnn_benchmarks.py --batch_size=32 --model=resnet50 --
variable_update=parameter_server --flush_stdout=true --num_gpus=1 --
local_parameter_device=cpu --device=cpu --data_format=NHWC --job_name=worker --
task_index=${VK_TASK_INDEX} --ps_hosts=${PS_HOST} --worker_hosts=${WORKER_HOST}
            image: volcanosh/example-tf:0.0.1
            name: tensorflow
            ports:
              - containerPort: 2222
                name: tfjob-port
            workingDir: /opt/tf-benchmarks/scripts/tf_cnn_benchmarks
        restartPolicy: OnFailure
```

配置作业终止策略

Worker运行信息

暴露2222端口

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

TensorFlow 演示

Tensorflow作业状态：

```
NAME                            READY   STATUS    RESTARTS   AGE
tensorflow-benchmark-ps-0       1/1     Running   0          6s
tensorflow-benchmark-worker-0   1/1     Running   0          6s
tensorflow-benchmark-worker-1   1/1     Running   0          6s
```

Tensorflow 作业调度完成，开始运行

Tensorflow作业结果：

```
2023-07-03 11:25:20.462920: I tensorflow/core/distributed_runtime/rpc/grpc_server_lib.cc:324] Started server with tar
get: grpc://localhost:2222
TensorFlow:  1.5
Model:       resnet50
Mode:        training
SingleSess:  False
Batch size:  32 global
             32 per device
Devices:     ['/job:worker/task:0/cpu:0']
Data format: NHWC
Optimizer:   sgd
Variables:   parameter_server
Sync:        True
==========
Running parameter server 0
```

作业输出

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

01 运行TensorFlow 作业

02 运行Pytorch作业

03 运行Spark作业

VOLCANO
CLOUD NATIVE COMPUTING FOUNDATION
THE LINUX FOUNDATION 开源软件学园

Pytorch 作业配置

```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: pytorch-job
  labels:
    "volcano.sh/job-type": “Pytorch“

spec:
  minAvailable: 1
  schedulerName: volcano

  plugins:
    pytorch: ["--master=master","--worker=worker","--port=23456"]

  policies:
    - event: PodEvicted
      action: RestartJob
```

根据业务需求设置作业类型

通过Volcano提供的通用插件配置Pytorch作业的角色和端口信息

配置作业重启策略

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

Pytorch 作业配置

```
tasks:
  - replicas: 1
    name: master
    policies:
      - event: TaskCompleted
        action: CompleteJob
    template:
      spec:
        containers:
          - image: gcr.io/kubeflow-ci/pytorch-dist-sendrecv-test:1.0
            imagePullPolicy: IfNotPresent
            name: master
        restartPolicy: OnFailure
  - replicas: 2
    name: worker
    template:
      spec:
        containers:
          - image: gcr.io/kubeflow-ci/pytorch-dist-sendrecv-test:1.0
            imagePullPolicy: IfNotPresent
            name: worker
            workingDir: /home
        restartPolicy: OnFailure
```

配置作业终止策略。当pytorch master完成时，整个Pytorch作业被视为完成。

Worker角色配置

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

Pytorch 演示

Pytorch作业状态：

```
[root@ecs-4b42-0002 demo-kind-test]# kubectl get pod
NAME                   READY   STATUS    RESTARTS   AGE
pytorch-job-master-0   1/1     Running   0          4s
pytorch-job-worker-0   1/1     Running   0          4s
pytorch-job-worker-1   1/1     Running   0          4s
```

Pytorch 作业调度完成

```
[root@ecs-4b42-0002 demo-kind-test]# kubectl get pod
NAME                   READY   STATUS      RESTARTS   AGE
pytorch-job-master-0   0/1     Completed   0          36s
pytorch-job-worker-0   0/1     Completed   0          36s
pytorch-job-worker-1   0/1     Completed   0          36s
```

Pytorch 作业运行完成

Pytorch作业结果：

```
[root@ecs-4b42-0002 demo-kind-test]# kubectl logs pytorch-job-master-0
INFO:root:Torch version: 1.0.0
INFO:root:MASTER_PORT: 23456
INFO:root:MASTER_ADDR: pytorch-job-master-0.pytorch-job
INFO:root:WORLD_SIZE: 3
INFO:root:RANK: 0
INFO:root:Result from worker 1 : tensor([[0.4862, 0.4732],
        [0.7807, 1.2848]])
INFO:root:Result from worker 2 : tensor([[0.4862, 0.4732],
        [0.7807, 1.2848]])
```

作业输出

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

01 运行TensorFlow 作业

02 运行Pytorch作业

03 运行Spark作业

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

Spark 作业配置

Podgroup config:

```
apiVersion: scheduling.volcano.sh/v1beta1
kind: PodGroup
spec:
  minMember: 1

  minResources:
    cpu: "4"
    memory: "5Gi“

  priorityClassName: system-node-critical

  queue: default
```

Specify minMember to 1 to make a driver pod

Specify minResources to support resource reservation (the driver pod resource and executors pod resource should be considered)

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

Spark 作业配置

```
./spark-3.3.1/bin/spark-submit \
--master k8s://https://127.0.0.1:40883 \
--deploy-mode cluster \
--driver-cores 1 \
--driver-memory 2G \
--num-executors 1 \
--executor-cores 1 \
--executor-memory 1G \
--name spark-volcano-wy1 \
--class org.apache.spark.examples.SparkPi \
--conf spark.executor.instances=3 \
--conf spark.kubernetes.container.image=wangyang0616/spark:3.3.1-volcano.v1 \
--conf spark.kubernetes.authenticate.driver.serviceAccountName=spark \
--conf spark.kubernetes.scheduler.name=volcano \
--conf spark.kubernetes.scheduler.volcano.podGroupTemplateFile=…/podgroup-template.yaml \
--conf spark.kubernetes.driver.pod.featureSteps=org.apache.spark.deploy.k8s.features.VolcanoFeatureStep \
--conf spark.kubernetes.executor.pod.featureSteps=org.apache.spark.deploy.k8s.features.VolcanoFeatureStep \
local:///opt/spark/examples/jars/spark-examples_2.12-3.3.1.jar
```

1. Specify custom scheduler
2. Specify scheduler hints (podgroup template)
3. Specify custom feature step

VOLCANO CLOUD NATIVE COMPUTING FOUNDATION THE LINUX FOUNDATION 开源软件学园

Spark 演示

Spark 作业状态：

```
NAME                                        READY   STATUS    RESTARTS   AGE
spark-pi-43882d891f8c6886-exec-1            1/1     Running   0          7s
spark-pi-43882d891f8c6886-exec-2            1/1     Running   0          7s
spark-pi-43882d891f8c6886-exec-3            1/1     Running   0          7s
spark-volcano-wy1-b98f3d891f8c5504-driver   1/1     Running   0          12s
```

拉起spark driver pod
Driver pod 拉起3个executor pod

```
NAME                                        READY   STATUS      RESTARTS   AGE
spark-volcano-wy1-b98f3d891f8c5504-driver   0/1     Completed   0          35s
```

Spark 作业运行完成

Thanks