（五）Kubernetes上深度学习最佳实践

2018-11-08

目标

- 熟悉Arena的基本使用，包括创建任务，查看任务列表，查看任务信息以及日志

准备工作

- 安装和设置kubectl客户端，请参考不同的操作系统，如果已经安装请忽略：
  - macos

    ```
    curl -LO https://kubectl.oss-cn-hangzhou.aliyuncs.com/macos/kubectl
    chmod +x ./kubectl
    sudo mv ./kubectl /usr/local/bin/kubectl
    kubectl --help
    ```

  - linux

    ```
    curl -LO https://kubectl.oss-cn-hangzhou.aliyuncs.com/linux/kubectl
    chmod +x ./kubectl
    sudo mv ./kubectl /usr/local/bin/kubectl
    kubectl --help
    ```

  - windows
    把https://kubectl.oss-cn-hangzhou.aliyuncs.com/windows/kubectl.exe 放到系统PATH路径下

    ```
    kubectl --help
    ```

- 配置kubectl连接Kubernetes集群的配置，可参考文档通过kubectl连接Kubernetes集群

- 移除ingress controller使用的负载均衡

```
# kubectl delete svc -n kube-system nginx-ingress-lb
```

实验步骤

1. 部署arena命令行控制台

以下为arena命令行控制台的部署yaml:

```
apiVersion: v1
kind: Pod
metadata:
  name: arena
  namespace: kube-system
  annotations:
    scheduler.alpha.kubernetes.io/critical-pod: ""
spec:
  serviceAccountName: admin
  hostPID: true
  hostNetwork: true
  containers:
  - args:
    - tail
    - -f
    - /dev/null
    env:
    - name: useLoadBlancer
      value: "true"
    image: registry.cn-hangzhou.aliyuncs.com/acs/arena:0.1.0-35dd9e4
    imagePullPolicy: IfNotPresent
    name: arena
    resources:
      requests:
        cpu: 100m
```

将该模板保存到 `arena.yaml` 文件，并且创建

```
kubectl create -f arena.yaml
```

2. 创建完成后，查看arena的部署日志

```
kubectl logs -n kube-system arena
```

3. 通过 `kubectl exec` 登录arena的命令行控制台

```
kubectl exec -it -n kube-system arena bash
```

注意以下命令都是在登录到arena的命令行控制台后执行

4. 在arena的控制台中查看基础组件是否安装成功, 比如pod是否运行，service是否绑定

```
# kubectl get all -n arena-system
NAME                                          READY     STATUS    RESTARTS   AGE
pod/mpi-operator-5c686b86b-4xb8b              1/1       Running   0          14h
pod/tf-job-dashboard-7dc786b7fb-qthf6         1/1       Running   0          14h
pod/tf-job-operator-v1alpha2-98bfbfc4-fvjdw   1/1       Running   0          14h

NAME                       TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/tf-job-dashboard   NodePort   172.19.23.126   <none>        80:32543/TCP   14h

NAME                                       DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mpi-operator               1         1         1            1           14h
deployment.apps/tf-job-dashboard           1         1         1            1           14h
deployment.apps/tf-job-operator-v1alpha2   1         1         1            1           14h

NAME                                                DESIRED   CURRENT   READY     AGE
replicaset.apps/mpi-operator-5c686b86b              1         1         1         14h
replicaset.apps/tf-job-dashboard-7dc786b7fb         1         1         1         14h
replicaset.apps/tf-job-operator-v1alpha2-98bfbfc4   1         1         1         14h
```

5. 查看节点状态

```
# arena top node
NAME                                IPADDRESS      ROLE    GPU(Total)  GPU(Allocated)
cn-hangzhou.i-bp1ic9ggky8i9aygvtwg  192.168.1.187  <none>  0           0
cn-hangzhou.i-bp1ic9ggky8i9aygvtwh  192.168.1.186  <none>  0           0
-----------------------------------------------------------------------------------------
Allocated/Total GPUs In Cluster:
0/0 (0%)
```

到这里可以看到

6. 提交分布式训练任务

```
arena submit tf \
             --name=tf-estimator-dist \
             --chief \
             --evaluator \
             --workers=1 \
             --ps=1 \
             --env=TF_CPP_MIN_LOG_LEVEL=2 \
             --image=registry.cn-hangzhou.aliyuncs.com/tensorflow-samples/tf-estimator:cpu \
             --tensorboard \
             "python /models/official/mnist/mnist.py"
```

运行结果：

```
NAME:   tf-estimator-dist
LAST DEPLOYED: Wed Nov  7 03:02:50 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Service
NAME                     TYPE          CLUSTER-IP   EXTERNAL-IP  PORT(S)         AGE
tf-estimator-dist-tfjob  LoadBalancer  172.19.43.8  <pending>    6006:32013/TCP  1s

==> v1beta1/Deployment
NAME                     DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
tf-estimator-dist-tfjob  1        1        1           0          1s

==> v1alpha2/TFJob
NAME                     AGE
tf-estimator-dist-tfjob  1s

==> v1/Pod(related)
NAME                                     READY  STATUS   RESTARTS  AGE
tf-estimator-dist-tfjob-8f7569b54-jvzz2  0/1    Pending  0         0s
```

> 这个命令单机训练的命令是基本相同的，唯一的区别是增加了分布式训练需要的各个拓扑结构参数

```
--chief \
--evaluator \
--workers=1 \
--ps=1 \
```

> 应用代码：https://github.com/cheyang/models/blob/master/official/mnist/mnist.py
> 镜像对应的Dockerfile： https://github.com/cheyang/models/blob/master/Dockerfile.cpu

7. 查看任务列表, 发现该任务处于PENDING的状态

```
# arena list
NAME                     STATUS   TRAINER  AGE  NODE
tf-estimator-standalone  PENDING  TFJOB    0s   N/A
```

8. 查看任务列表, 发现该任务处于RUNNING的状态

```
# arena list
NAME               STATUS   TRAINER  AGE  NODE
tf-estimator-dist  RUNNING  TFJOB    46s  192.168.1.186
```

9. 检查Pending原因, 通过Events可以看到是由于镜像没有完成下载导致任务Pending

```
# # arena get tf-estimator-dist  -e
NAME               STATUS   TRAINER  AGE  INSTANCE                            NODE
tf-estimator-dist  PENDING  TFJOB    0s   tf-estimator-dist-tfjob-chief-0      N/A
tf-estimator-dist  PENDING  TFJOB    0s   tf-estimator-dist-tfjob-evaluator-0  N/A
tf-estimator-dist  PENDING  TFJOB    0s   tf-estimator-dist-tfjob-ps-0         N/A
tf-estimator-dist  PENDING  TFJOB    0s   tf-estimator-dist-tfjob-worker-0     N/A

Your tensorboard will be available on:
47.110.179.110:6006

Events:
INSTANCE                         TYPE    AGE  MESSAGE
---------                        ----    ---  --------
tf-estimator-dist-tfjob-chief-0  Normal  37s  [Scheduled] Successfully assigned default/tf-estimator-
tf-estimator-dist-tfjob-chief-0  Normal  33s  [Pulling] pulling image "registry.cn-hangzhou.aliyuncs.
```

10. 查看运行的日志，请留意一下每100个step花费的平均时间

```
# arena logs tf-estimator-dist
2018-11-08T10:00:18.633268103Z I1108 10:00:18.632817 140601684506368 tf_logging.py:115] TF_CONFIG environm
2018-11-08T10:00:18.633519861Z I1108 10:00:18.633331 140601684506368 tf_logging.py:115] Using default conf
2018-11-08T10:00:18.634059354Z I1108 10:00:18.633665 140601684506368 tf_logging.py:115] Using config: {'_s
2018-11-08T10:00:18.635321752Z I1108 10:00:18.635102 140601684506368 tf_logging.py:115] Start Tensorflow s
2018-11-08T10:00:18.751716432Z I1108 10:00:18.751452 140601684506368 tf_logging.py:115] Calling model_fn.
```

11. 大约两三分钟左右之后， 再次查看任务执行状态。

```
# arena list
NAME                     STATUS     TRAINER  AGE  NODE
tf-estimator-standalone  SUCCEEDED  TFJOB    3m   N/A
```

12. 此时可以通过 `arena get` 命令查看TensorBoard的访问链接

```
# arena get tf-estimator-dist
NAME               STATUS     TRAINER  AGE  INSTANCE                         NODE
tf-estimator-dist  SUCCEEDED  TFJOB    6m   tf-estimator-dist-tfjob-chief-0  N/A

Your tensorboard will be available on:
47.110.179.110:6006
```

13. 这时可以通过tensorboard的链接查看

清理实验环境

在arena控制台执行

```
# arena delete tf-estimator-dist
# kubectl delete ns arena-system
```

About

This theme was developed by Jonathan Klughertz. The source code is available on Github. Create Websites. Make Magic.

Recent Posts

Kubernetes动手实践沙龙
（一）Kubernetes弹性扩缩容实践
（二）基于Kubernetes的三种发布策略 -
（三）通过Serverless Kubernete

@Untitled. All right reserved | Design & Hexo Jonathan Klughertz