

Partisaner 用户操作手册 V1.0

WWW.XTAOTECH.COM



交互模式.....	4
运行方式.....	4
工作目录.....	4
参数说明.....	4
运行示例.....	5
服务模式.....	6
设置环境变量	6
获取环境变量	6
提交作业.....	7
查询作业状态	7
查询作业配置	8
获取作业列表	9
取消作业.....	10
获取任务日志	10
配置说明.....	11
MPICH配置.....	11
配置示例	11
详细配置项	11
OpenMPI配置.....	12
配置示例	12
详细配置项	13
Tensorflow配置.....	14
配置示例	14
详细配置项	14
启动脚本	15
Pytorch配置	16

配置示例	16
详细配置项	16
启动脚本	17
通用 (Universal) 框架配置	18
配置示例	18
详细配置项	19
启动方式	20

概述

Partisaner是极道开发的融合计算平台项目，用于快速启动和监控分布式计算框架。目前支持MPICH、OpenMPI、Tensorflow、Pytorch、MXnet和用户自定义框架（universal）等多种分布式框架，实现了一键式部署。Partisaner有两种运行模式：

- 交互模式：用户通过运行partisaner命令启动自己的框架，运行结果通过终端直接展示给用户。
- 服务模式：Partisaner作为一个服务运行在平台上，集中负责用户的分布式框架的运行、监控和管理。用户通过particli命令与Partisaner服务交互。用户的框架作为一个作业（Job）运行，用户通过particli提交、查询和管理自己的作业。服务模式具备完善的资源管理和调度功能。

下面对Partisaner这两种具体使用方式进行说明。

交互模式

交互模式下用户通过partisaner命令启动一个分布式框架（Tensorflow、Mpich、PyTorch、OpenMPI和自定义框架），partisaner命令根据特定的调度策略将任务分发到集群的不同节点上执行，并监控任务的运行状态，partisaner的运行会伴随框架运行的整个生命周期，框架中的所有任务结束后，partisaner随即运行结束。

运行方式

partisaner -t 分布式框架类型 -c 配置文件 [-i]

partisaner 会读取配置信息并根据具体配置和集群资源将计算任务调度到合适的节点上，随后监控任务的状态，同时可以动态的获取进程的输出信息和性能统计信息到指定的目录下，方便用户查看。当所有计算任务结束后，partisaner命令随之退出。

工作目录

partisaner运行会在当前目录下保存运行数据用于容错，所以不同计算任务不能复用同一目录，每次启动新的计算时都需要在一个新的工作目录下运行partisaner命令。

参数说明

- -t 必选

Type 指明要启动的分布式框架类型，当前支持mpich、 tensorflow、 caffe、 pytorch、 mxnet、 openmpi、 common，必须指定一种类型。

- -c 可选

Cluster 指定计算集群配置的描述文件，该Json格式文件说明了集群的详细配置，该

参数和-p参数必须指定其中一个。下图为一个Tensorflow集群的说明文件：

```

{
  "Roles":
  [
    {
      "Name": "ps",
      "Procs": 1,
      "Cpu": 2,
      "Mem": 1000
    },
    {
      "Name": "worker",
      "Procs": 2,
      "Cpu": 1,
      "Mem": 1000
    }
  ],
  "Cmd": "sh /tmp/bin/run.sh",
  "Image": "tensorflow",
  "Env":
  {
    "TRAIN_STEPS": "30000"
  },
  "Volmap":
  {
    "/mnt/alamo-vol1/tensorflow": "/tmp"
  },
  "LogDir": "logs",
  "LogInterval": 60,
  "AbortOnFail": true
}

```

- -p 可选

Parameter 由集群配置参数组成的字符串，该参数和-c参数必须指定其中一个，推荐使用-c参数。下面是一个Tensorflow集群配置参数的例子：

```
'Roles.0.Name=ps&Roles.0.Cpu=1.0&Roles.0.Mem=1000.0&Roles.1.Name=worker&Roles.1.Cpu=2.0&Roles.1.Mem=2000.0&cmd=/tmp/bin/run.sh
/tmp/steps&image=tensorflow&Volmap./mnt/alamo-vol1/tensorflow=/tmp&logdir=logs&LogInterval=60&AbortOnFail=true'
```

- -i 可选

Interactive 指定使用交互的方式运行partisaner命令。指定此参数时Partisaner在前台运行。不指定时在后台运行，此时标准输入、输出、错误分别绑定到当前目录下的stdin、stdout、stderr三个文件。

- -k

Kill 杀死任务，指定要杀死的任务ID（多个任务以‘，’隔开）。要求该命令在与启动命令相同的目录下运行。

- -s

Shut down 杀死整个集群。要求该命令在与启动命令相同的目录下运行。

运行示例

- 以交互的方式运行一个tensorflow集群，指定tensorflow.json为配置描述文件：

```
partisaner -t tensorflow -c tensorflow.json -i
```

- 杀死特定的任务：

```
partisaner -k tensorflow-task.a9c2af51-bc7c-4d5b-8419-1f187550cc17, tensorflow-task.9849a368-0141-4bea-b374-6982dc0e100a -i
```

上述命令示例杀死任务tensorflow-task.a9c2af51-bc7c-4d5b-8419-1f187550cc17和tensorflow-task.a9c2af51-bc7c-4d5b-8419-1f187550cc17。

- 停掉整个集群：

```
partisaner -s -i
```

服务模式

服务模式下用户通过particli客户端与Partisaner服务交互。

设置环境变量

常用的环境变量主要是：Partisaner REST server 的地址。一旦设置好环境变量，用户不需要在每一个 particli命令执行过程中携带 `-s` 参数。

命令：

```
particli env set
```

参数：

- 环境变量名字，如：server
- 环境变量值，如：partisaner.servicemgr.apc:1025

示例：

```
[root@Cc1Apc common]# particli env set server Cc1Apc:41111
[root@Cc1Apc common]# particli env get
{"server": "Cc1Apc:41111"}
```

获取环境变量

命令：particli env get

可选参数：

- `--key=环境变量的名字` 不输入可选参数，默认输出所有设置过的环境变量。

示例：

```
[root@Cc1Apc common]# particli env get  
{"server": "Cc1Apc:41111"}  
[root@Cc1Apc common]# particli env get --key=server  
Cc1Apc:41111
```

提交作业

命令 : `particli job submit <type> [<file>] [<label>] [<params>]`

参数 :

- type : 指明要启动的分布式框架类型，当前支持mpich、 tensorflow、 caffe、 pytorch、 mxnet、 openmpi、 common，必须指定一种类型。
- file : 指定计算集群配置的描述文件，该Json格式文件说明了集群的详细配置。
- label : 用户可以给一个作业指定一个label，用于后续对该作业的查询和操作。

返回值 :

返回该作业id

示例 :

```
[root@Cc1Apc common]# particli job submit common cluster1.json  
The job added success, job ID is: 4c7ad1b8-ecd4-4d9f-acb3-6e40d0b53ef4
```

查询作业状态

命令 : `particli job status [<id>] [<label>]`

参数 :

- id : 作业id
- label : 用户指定的作业label

注意 : 用户需要在id和label中指定其中一个。指定id时可以只指定原始id的子串，但当多个作业id包含该子串时命令会报错。如原始id为99eccf6e-4407-424b-a62f-d6758948b7d7，可以使用`particli job status d6758948b7d7` 获取该作业状态。

返回值 : 作业的状态信息

示例 :

```

[root@Cc1Apc common]# particli job status 4c7ad1b8-ecd4-4d9f-acb3-6e40d0b53ef4
Job 4c7ad1b8-ecd4-4d9f-acb3-6e40d0b53ef4
Label:
Type: common
Priority: 3
Status: FINISHED
SubmitTime: Mon Aug 9 11:29:31 CST 2021
StartTime: Mon Aug 9 11:30:02 CST 2021
EndTime: Mon Aug 9 11:31:42 CST 2021
FailureReasons: []
-----
task distribute-task_4c7ad1b8-ecd4-4d9f-acb3-6e40d0b53ef4_aad1c34b-561d-4da1-b718-231ddc2307f0
Status: TASK_FINISHED
Name: root.distribute-0
Cmd: sleep 100
Cpu: 1
Mem: 1000
Gpu: 0
Host: Cc2Apc
Image: partisaner_demo_mpich
WorkDir: /tmp/
FailureReason:
-----
task distribute-task_4c7ad1b8-ecd4-4d9f-acb3-6e40d0b53ef4_09c73009-2968-4075-be5c-ff392f726ab3
Status: TASK_FINISHED
Name: root.distribute-1
Cmd: sleep 100
Cpu: 1
Mem: 1000
Gpu: 0
Host: Cc4Apc
Image: partisaner_demo_mpich
WorkDir: /tmp/
FailureReason:

```

查询作业配置

命令 : `particli job config [<id>] [<label>]`

参数 :

- id : 作业id
- label : 用户指定的作业label

注意 : 用户需要在id和label参数中指定一个。指定id时可以只指定原始id的子串 , 但当多个作业id包含该子串时命令会报错。如原始id为99eccf6e-4407-424b-a62f-d6758948b7d7 , 可以使用`particli job config d6758948b7d7` 获取该作业配置。

返回值 : 作业的配置信息

示例 :


```
[root@CclApc common]# particli job config 4c7ad1b8-ecd4-4d9f-acb3-6e40d0b53ef4
{
  "Procs": 2,
  "Cpu": 1,
  "Mem": 1000,
  "WorkDir": "/tmp/",
  "Cmd": "sleep 100",
  "Priority":3,
  "ScheduleDomain":"hg-test",
  "Image": "partisaner_demo_mpich"
}
```

获取作业列表

命令 : `particli job list`

参数 :

- `-a` : 列举所有的作业
- `-t` : 列举指定状态的作业 , 可以取的值为:
waiting/running/finished/failed/canceled (不区分大小写)

返回值 : 满足条件的所有作业

示例 :

```
[root@Cc1Apc common]# particli job list -t FINISHED
Get 63 jobs:
Job b58d2639-2451-4e29-8c5c-c5e841962968:
  Label:
  Status: FINISHED
  SubmitTime: Wed Aug  4 11:32:29 CST 2021
  StartTime: Wed Aug  4 11:32:44 CST 2021
  EndTime: Wed Aug  4 11:33:53 CST 2021

Job b089a3dc-7d6a-499a-bbbb-a311ff98f14e:
  Label:
  Status: FINISHED
  SubmitTime: Wed Aug  4 11:40:14 CST 2021
  StartTime: Wed Aug  4 11:40:53 CST 2021
  EndTime: Wed Aug  4 11:41:25 CST 2021

Job f127a70d-1eaa-4057-ba91-9b056380dcea:
  Label:
  Status: FINISHED
  SubmitTime: Wed Aug  4 11:40:13 CST 2021
  StartTime: Wed Aug  4 11:40:53 CST 2021
  EndTime: Wed Aug  4 11:41:46 CST 2021
```

取消作业

命令 : `particli job cancel [<id>] [<label>]`

参数 :

- id : 作业id (可以为all或者以‘,’分隔的id列表, 指定all时取消所有未运行或正在运行的作业)
- label : 用户指定的作业label

注意 : id和label指定其中一个。指定id时可以只指定原始id的子串, 但当多个作业id包含该子串时命令会报错。如原始id为99eccf6e-4407-424b-a62f-d6758948b7d7, 可以使用`particli job cancel d6758948b7d7` 取消该作业。

示例 :

```
[root@Cc1Apc common]# particli job cancel 4c7ad1b8-ecd4-4d9f-acb3-6e40d0b53ef4
Succeed to cancel job
```

获取任务日志

命令：particli job log <dir> [<job_id>] [<task_id>]

参数：

- dir：指定保存日志文件的目录
- job_id：作业id
- task_id：任务id

注意：只指定作业id会获取该作业的所有任务的日志，如果指定了任务id则只会获取指定任务的日志

返回值：在指定目录下生成目标任务的日志文件

示例：

```
[root@Cc1Apc common]# particli job submit common cluster1.json
The job added success, job ID is: 8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5
[root@Cc1Apc common]# particli job log log 8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5
get log for task distribute-task_8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5_c8bb0278-40b2-4dc1-9555-c993a939f86a
get log for task distribute-task_8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5_5f69f2b9-99cf-4de5-8f76-99df6b436f85
Written logs to log
[root@Cc1Apc common]# ls log/
distribute-task_8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5_5f69f2b9-99cf-4de5-8f76-99df6b436f85.cmd_stderr
distribute-task_8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5_5f69f2b9-99cf-4de5-8f76-99df6b436f85.cmd_stdout
distribute-task_8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5_c8bb0278-40b2-4dc1-9555-c993a939f86a.cmd_stderr
distribute-task_8e9ce9ce-6deb-4c40-bee7-a05cb010a7c5_c8bb0278-40b2-4dc1-9555-c993a939f86a.cmd_stdout
```

配置说明

由于各种计算框架都有其结构的特殊性，所以每种计算框架的配置信息不同，下面分别进行说明。

MPICH配置

配置示例

```
{
  "Procs": 5,
  "Cpu": 10,
  "Mem": 1000,
  "WorkDir": "/tmp/",
  "Cmd": "/home/mpi/mpich-install/run/calculator-PI steps 50000",
  "Image": "partisaner_demo_mpich"
}
```

详细配置项

名称	类型	必填/默认	说明
Procs	int	必填	需要启动的进程数
Cpu	float	必填	单个进程运行分配的CPU资源
Mem	float	必填	单个进程运行分配的内存资源

Gpu	float	0	单个进程运行分配的GPU资源（当前仅支持整数）
Infiniband	bool	False	是否使用IB卡
Workdir	string	空	进程运行的工作目录
Cmd	string	必填	进程运行命令
Image	string	必填	进程启动的docker镜像
Volmap	map[string]string	空	启动容器时的卷映射配置，Key为主机路径，Value为容器内路径
Env	map[string]string	空	传入容器的环境变量
LogDir	string	空	保存任务输出日志信息的目录(只在交互模式下生效)
LogInterval	int	60	获取任务输出日志信息的频率，单位秒(只在交互模式下生效)
ProfilingDir	string	空	保存进程性能统计信息的目录(只在交互模式下生效)
ProfilingInterval	int	180	获取进程性能统计信息的频率，单位秒(只在交互模式下生效)
AbortOnFail	bool	False	当其中一个或多个任务失败时，是否杀死所有任务
MatchPolicy	int	0	启动任务时的调度策略。0：尽量将进程启动在不同的节点上；1：尽量将进程启动在相同的节点上；2：使用GPU优化调度，在满足GPU资源的情况下尽量将进程启动在同一个节点上，减少进程间的通信延迟。
ScheduleDomain	string	空	调度域，任务只会在指定调度域的资源上运行，不指定时使用默认调度域。

OpenMPI配置

配置示例

```

{
  "Procs": 5,
  "Cpu": 1,
  "Mem": 128,
  "Gpu": 1,
  "WorkDir": "/tmp",
  "Cmd": "/tmp/openmpi/openmpi-4.0.2/examples/hello_c",
  "Image": "openmpi",
  "User": "root",
  "Volmap" : {
    "/mnt/registry" : "/tmp"
  },
  "Env": {}
}

```

详细配置项

名称	类型	必填/默认	说明
Procs	int	必填	需要启动的进程数
Cpu	float	必填	单个进程运行分配的CPU资源
Mem	float	必填	单个进程运行分配的内存资源
Gpu	float	0	单个进程运行分配的GPU资源（当前仅支持整数）
Infiniband	bool	False	是否使用IB卡
Workdir	string	空	进程运行的工作目录
Cmd	string	必填	进程运行命令
Image	string	必填	进程启动的docker镜像
Volmap	map[string]string	空	启动容器时的卷映射配置，Key为主机路径，Value为容器内路径
Env	map[string]string	空	传入容器的环境变量
LogDir	string	空	保存任务输出日志信息的目录
LogInterval	int	60	获取任务输出日志信息的频率，单位秒
ProfilingDir	string	空	保存进程性能统计信息的目录
ProfilingInterval	int	180	获取进程性能统计信息的频率，单位秒
AbortOnFail	bool	False	当其中一个或多个任务失败时，是否杀死所有任务
MatchPolicy	int	0	启动任务时的调度策略。0：尽量将进程启动在不同的节点上；1：尽量将进程启动在相同的节点上；2：使用GPU优化调度，在满足GPU资源的情况下尽量将进程启动在同一个节点上，减少进程间的通信延迟。

ScheduleDomain	string	空	调度域，任务只会在指定调度域的资源上运行，不指定时使用默认调度域。
----------------	--------	---	-----------------------------------

Tensorflow配置

配置示例

```

{
  "Roles":
  [
    {
      "Name": "ps",
      "Procs": 1,
      "Cpu": 2,
      "Mem": 1000
    },
    {
      "Name": "worker",
      "Procs": 2,
      "Cpu": 1,
      "Mem": 1000
    }
  ],
  "Cmd": "sh /tmp/bin/run.sh",
  "Image": "tensorflow",
  "Env":
  {
    "TRAIN_STEPS": "30000"
  },
  "Volmap":
  {
    "/mnt/alamo-vol1/tensorflow": "/tmp"
  },
  "LogDir": "logs",
  "LogInterval": 60,
  "AbortOnFail": true
}

```

详细配置项

名称	类型	必填/默认	说明
Roles	[]struct	必填	分布式程序中的角色列表，一般为PS、Worker
Role.Name	string	必填	角色类型，一般为PS、Worker
Role.Procs	int	必填	对应角色的进程数
Role.Cpu	float	必填	对应角色的进程运行分配的CPU资源
Role.Mem	float	必填	对应角色的进程运行分配的内存资源
Role.Gpu	float	0	对应角色的进程运行分配的GPU资源（当前仅支持整数）
Visualization	struct	空	可视化服务的配置信息。Partisaner支持Tensorflow的可视化界面，当指定Visualization时，Partisaner在启动计算任务的同时会启动一个TensorBoard可视化服务，并提供访问方式。

Visualization.DataDir	string	空	DataDir为存放Tensorflow summary、meta等数据的目录，一般可以是训练的checkpoint目录。注意TensorBoard服务不会随着计算任务的结束而停止，需要用户手动执行partisaner -s命令结束服务。
Infiniband	bool	False	是否使用IB卡
Workdir	string	空	进程运行的工作目录
Cmd	string	必填	进程运行命令
Image	string	必填	进程启动的docker镜像
Volmap	map[string]string	空	启动容器时的卷映射配置，Key为主机路径，Value为容器内路径
Env	map[string]string	空	传入容器的环境变量
LogDir	string	空	保存任务输出日志信息的目录(只在交互模式下生效)
LogInterval	int	60	获取任务输出日志信息的频率，单位秒(只在交互模式下生效)
ProfilingDir	string	空	保存进程性能统计信息的目录(只在交互模式下生效)
ProfilingInterval	int	180	获取进程性能统计信息的频率，单位秒(只在交互模式下生效)
AbortOnFail	bool	False	当其中一个或多个任务失败时，是否杀死所有任务
MatchPolicy	int	0	启动任务时的调度策略。0：尽量将进程启动在不同的节点上；1：尽量将进程启动在相同的节点上；2：使用GPU优化调度，在满足GPU资源的情况下尽量将进程启动在同一个节点上，减少进程间的通信延迟。
ScheduleDomain	string	空	调度域，任务只会在指定调度域的资源上运行，不指定时使用默认调度域。

启动脚本

Partisaner在启动 Tensorflow的任务时会将各个任务所在的地址信息以环境变量的形式传入启动的Docker容器中，用户需要自行准备启动脚本获取这些环境变量。所有的环境变量包括

- TASK_ROLE: 当前进程属于哪一个角色，一般为Ps或Worker。
- TASK_INDEX: 当前进程在所属角色中的ID号，一般从0开始。
- HOSTLIST_{role}: 每个角色中所有进程所在节点的地址列表，{role}为用户所配置角色名的大写形式，一般为PS或WORKER。地址列表的取值形式为 host1:port,host2:port,host3:port

下面是一个启动脚本的示例：

```
echo "Start to run tensorflow cluster"
rm -rf /tmp/checkpoint/*
python /tmp/bin/mnist.py --job_name=$TASK_ROLE --task_index=$TASK_INDEX --ps_hosts=$HOSTLIST_PS --worker_hosts=$HOSTLIST_WORKER --train_steps=$step
```

Pytorch配置

配置示例

```
{
  "WorldSize": 2,
  "NeedInfiniband": true,
  "Cpu": 1,
  "Gpu": 1,
  "Mem": 1000,
  "Cmd": "run.sh",
  "Image": "pytorch",
  "WorkDir": "/pytorch",
  "Env": {
    "BACKEND": "nccl",
    "INIT_METHOD": "env://"
  },
  "Volmap": {
    "/mnt/anna-fuse/pytorch": "/pytorch"
  }
}
```

详细配置项

名称	类型	必填/默认	说明
WorldSize	int	必填	启动的进程数
Cpu	float	必填	单个进程运行分配的CPU资源
Mem	float	必填	单个进程运行分配的内存资源
Gpu	float	0	单个进程运行分配的GPU资源（当前仅支持整数）
Visualization	struct	空	可视化服务的配置信息。Partisaner支持Pytorch的可视化界面，当指定Visualization时，Partisaner在启动计算任务的同时会启动一个TensorBoard可视化服务，并提供访问方式。

Visualization.DataDir	string	空	DataDir为存放Pytorch summary、meta等数据的目录，一般可以是训练的checkpoint目录。注意TensorBoard服务不会随着计算任务的结束而停止，需要用户手动执行partisaner -s命令结束服务。
Infiniband	bool	False	是否使用IB卡
Workdir	string	空	进程运行的工作目录
Cmd	string	必填	进程运行命令
Image	string	必填	进程启动的docker镜像
Volmap	map[string]string	空	启动容器时的卷映射配置，Key为主机路径，Value为容器内路径
Env	map[string]string	空	传入容器的环境变量
LogDir	string	空	保存任务输出日志信息的目录(只在交互模式下生效)
LogInterval	int	60	获取任务输出日志信息的频率，单位秒(只在交互模式下生效)
ProfilingDir	string	空	保存进程性能统计信息的目录(只在交互模式下生效)
ProfilingInterval	int	180	获取进程性能统计信息的频率，单位秒(只在交互模式下生效)
AbortOnFail	bool	False	当其中一个或多个任务失败时，是否杀死所有任务
MatchPolicy	int	0	启动任务时的调度策略。0: 尽量将进程启动在不同的节点上；1: 尽量将进程启动在相同的节点上；2: 使用GPU优化调度，在满足GPU资源的情况下尽量将进程启动在同一个节点上，减少进程间的通信延迟。
ScheduleDomain	string	空	调度域，任务只会在指定调度域的资源上运行，不指定时使用默认调度域。

启动脚本

Partisaner在启动 Pytorch的任务时会把Pytorch所需的信息以环境变量的形式传入启动的Docker容器中，用户需要自行准备启动脚本获取这些环境变量。所有的环境变量包括：

- RANK: 当前进程的rank号。
- WORLD_SIZE: 所有任务数。
- INIT_METHOD: Rank0(Master)进程的访问地址。

下面是一个启动脚本的示例：

```
#!/bin/bash  
python /home/mnist_nets.py --rank=$RANK --world-size=$WORLD_SIZE --init-method=$INIT_METHOD
```

通用 (Universal) 框架配置

“universal”类型的框架指的是用户自定义的框架。Partisaner支持通用分布式框架，帮助用户启动自定义的框架。

配置示例

配置1：

```
{  
  "Roles":  
  [  
    {  
      "Name": "role1",  
      "Procs": 2,  
      "Cpu": 15,  
      "Gpu": 0,  
      "Mem": 1000,  
      "Port": 3  
    },  
    {  
      "Name": "role2",  
      "Procs": 5,  
      "Cpu": 3,  
      "Gpu": 0,  
      "Mem": 1000,  
      "Port": 3  
    }  
  ],  
  "Cmd": "env",  
  "WorkDir": "/tmp",  
  "Image": "common-img",  
  "Env":  
  {  
    "TRAIN_STEPS": "60000",  
    "TRAIN_DATA_DIR": "/tmp/mnist"  
  },  
  "Volmap":  
  {  
    "/autofs/vol6/hg-demo/partisaner": "/tmp"  
  },  
  "AbortOnFail": true  
}
```

配置2

```
{
  "Procs": 2,
  "Cpu": 1,
  "Mem": 1000,
  "WorkDir": "/tmp/",
  "Cmd": "echo dddddddd && sleep 100"
  "Image": "dddd"
}
```

类似于Tensorflow，universal类型也引入了role的概念，当框架中的任务具有异构性时可以使用role，类似配置1，否则可以使用如配置2的普通形式。

详细配置项

名称	类型	必填/默认	说明
Roles	[]struct	必填	分布式程序中的角色列表，一般为PS、Worker
Role.Name	string	必填	角色类型，一般为PS、Worker
Role.Procs	int	必填	对应角色的进程数
Role.Cpu	float	必填	对应角色的进程运行分配的CPU资源
Role.Mem	float	必填	对应角色的进程运行分配的内存资源
Role.Gpu	float	0	对应角色的进程运行分配的GPU资源（当前仅支持整数）
Role.Port	int	0	需要分配的端口数量
Procs	int	必填	需要启动的进程数
Cpu	float	必填	单个进程运行分配的CPU资源
Mem	float	必填	单个进程运行分配的内存资源
Gpu	float	0	单个进程运行分配的GPU资源（当前仅支持整数）
Visualization	struct	空	可视化服务的配置信息。Partisaner支持Tensorflow的可视化界面，当指定Visualization时，Partisaner在启动计算任务的同时会启动一个TensorBoard可视化服务，并提供访问方式。
Visualization.DataDir	string	空	DataDir为存放Tensorflow summary、meta等数据的目录，一般可以是训练的checkpoint目录。注意TensorBoard服务不会随着计算任务的结束而停止，需要用户手动执行partisaner -s命令结束服务。

Infiniband	bool	False	是否使用IB卡
Workdir	string	空	进程运行的工作目录
Cmd	string	必填	进程运行命令
Image	string	必填	进程启动的docker镜像
Volmap	map[string]string	空	启动容器时的卷映射配置，Key为主机路径，Value为容器内路径
Env	map[string]string	空	传入容器的环境变量
LogDir	string	空	保存任务输出日志信息的目录(只在交互模式下生效)
LogInterval	int	60	获取任务输出日志信息的频率，单位秒(只在交互模式下生效)
ProfilingDir	string	空	保存进程性能统计信息的目录(只在交互模式下生效)
ProfilingInterval	int	180	获取进程性能统计信息的频率，单位秒(只在交互模式下生效)
AbortOnFail	bool	False	当其中一个或多个任务失败时，是否杀死所有任务
MatchPolicy	int	0	启动任务时的调度策略。0：尽量将进程启动在不同的节点上；1：尽量将进程启动在相同的节点上；2：使用GPU优化调度，在满足GPU资源的情况下尽量将进程启动在同一个节点上，减少进程间的通信延迟。
ScheduleDomain	string	空	调度域，任务只会在指定调度域的资源上运行，不指定时使用默认调度域。

启动方式

Partisaner在启动universal类型框架的任务时会将各个任务所在的地址信息以环境变量的形式传入启动的Docker容器中，用户需要自行准备启动脚本获取这些环境变量。所有的环境变量包括：

- TASK_ROLE: 当前进程属于哪一个角色，如配置1图中的role1/role2，使用如图配置2的普通形式时不提供该环境变量。
- TASK_INDEX: 当前进程在所属角色中的ID号，一般从0开始，使用如图配置2的普通形式时不提供该环境变量。
- HOSTLIST_{role}: 每个角色中所有进程所在节点的地址列表，{role}为用户所配置角色名的大写形式，使用如图配置2的普通形式时该环境变量为HOSTLIST。地址列表的取值形式为host1:port1|port2,host2:port1|port2,host3:port1|port2|port3