



# 天翼云联邦学习产品

## 用户操作手册

天翼云科技有限公司

## 目录

版本更新表.....	4
<b>第一部分：服务器配置.....</b>	<b>5</b>
<b>第二部分：集群规划.....</b>	<b>5</b>
1. 集群配置.....	5
2. 架构图.....	6
<b>第三部分：组件说明.....</b>	<b>7</b>
<b>第四部分：基础环境配置.....</b>	<b>7</b>
1. hostname 配置(可选).....	7
2. 关闭 selinux(可选).....	8
3. 修改 Linux 系统参数.....	8
4. 关闭防火墙(可选).....	9
5. 软件环境初始化.....	9
6. 增加虚拟内存.....	10
<b>第五部分：项目部署.....</b>	<b>10</b>
1. 获取安装包.....	11
2. 操作系统参数检查.....	11
3. 部署.....	11
4. 配置 host 信息.....	13
5. 配置 guest 信息.....	15
6. 部署所有服务.....	17
7. eggroll 路由配置文件修改.....	17



8. 问题定位.....	19
<b>第六部分：测试.....</b>	<b>19</b>
1. Toy_example 部署验证.....	19
1.1 单边测试.....	19
1.2 双边测试.....	20
2. 最小化测试.....	20
2.1 上传预设数据.....	20
2.2 快速模式.....	21
2.3 正常模式.....	21
3. Fateboard testing.....	22
<b>第七部分：任务管理.....</b>	<b>22</b>
1. 任务创建.....	22
2. 查看任务.....	23
3. 取消任务.....	23
4. 查看任务日志明细.....	24
<b>第八部分：系统运维.....</b>	<b>25</b>
1. 服务管理.....	25
1.1 Eggroll 服务管理.....	25
1.2 Fate 服务管理.....	25
1.3 Mysql 服务管理.....	26
2. 查看进程和端口.....	26
2.1 查看进程.....	26



## 第一部分：服务器配置

服务器	配置
数量	>1 (根据实际情况配置)
配置	32 core /64GB memory / 2T 硬盘/100M 带宽
操作系统	CentOS linux 7.2 及以上/Ubuntu 16.04 以上
依赖包	(参见 4.5 软件环境初始化)
用户	用户: fate, 属主: fate (fate 用户需可以 sudo su root 而无需密码)
文件系统	1. 2T 硬盘挂载在/data 目录下; 2. 创建/ data/projects 目录, 目录属主为: fate:fate

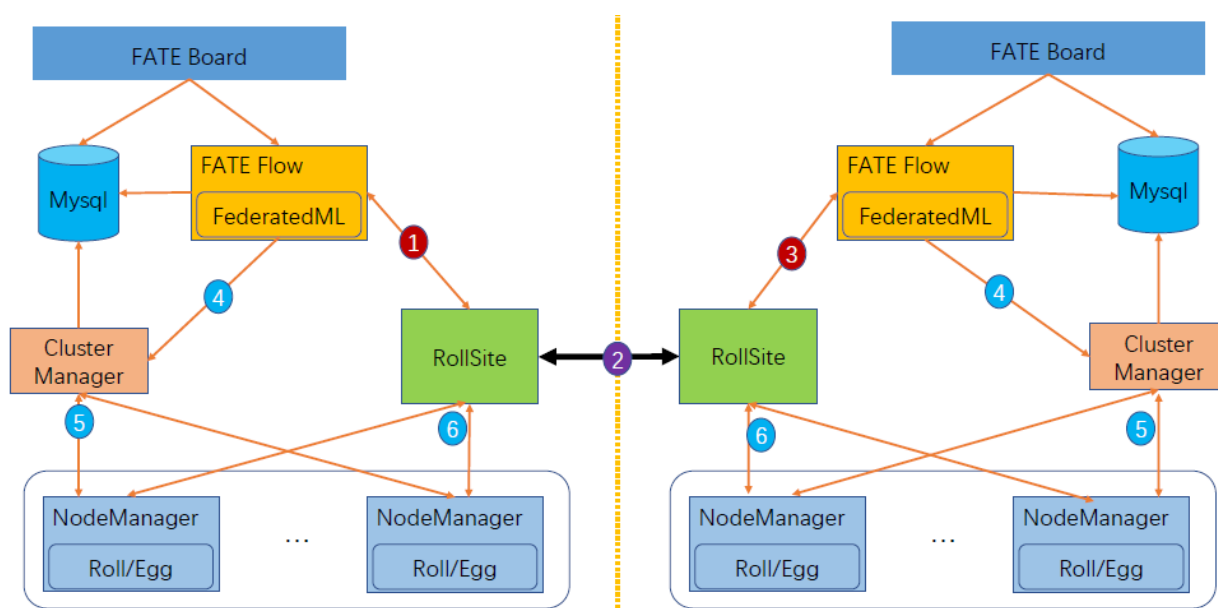
## 第二部分：集群规划

### 1. 集群配置

party	partyid	主机名	IP 地址	操作系统	安装软件	服务
PartyA	10000	VM_0_1_centos	10.30.68.83	CentOS 7.2/Ubuntu 16.04	fate, eggroll, mysql	fate_flow, fateboard, clustermanager,

						nodemanager, mysql, rollsite
PartyB	9999	VM_0_3_centos	10.30.68.82	CentOS 7.2/Ubuntu 16.04	fate, eggroll, mysql	fate_flow, fateboard, clustermanager, nodemanager, mysql, rollsite

## 2. 架构图



- 1、标识1, 2, 3, fateflow发起任务, 通过rollsite同步任务信息到所有party, 期间fateflow会检测所有方运行状态。
- 2、标识4, 5, 6, eggroll任务发起和调度计算, 并通过rollsite与对端进行通讯。

### 第三部分：组件说明

软件产品	组件	端口	说明
fate	fate_flow	9360;9380	联合学习任务流水线管理模块，每个 party 只能有一个此服务
fate	fateboard	8080	联合学习过程可视化模块，每个 party 只能有一个此服务
eggroll	clustermanager	4670	cluster manager 管理集群，每个 party 只能有一个此服务
eggroll	nodemanager	4671	node manager 管理每台机器资源，每个 party 可有多此服务，但一台服务器只能有一个
eggroll	rollsite	9370	跨站点或者说跨 party 通讯组件，相当于 proxy+federation，每个 party 只能有一个此服务
mysql	mysql	3306	数据存储，clustermanager 和 fateflow 依赖，每个 party 只需要一个此服务

### 第四部分：基础环境配置

#### 1. hostname 配置(可选)

##### 1) 修改主机名



在 10.30.68.83 root 用户下执行:

```
hostnamectl set-hostname VM_0_1_centos
```

在 10.30.68.82 root 用户下执行:

```
hostnamectl set-hostname VM_0_2_centos
```

## 2) 加入主机映射

在目标服务器 (10.30.68.83 10.30.68.82) root 用户下执行:

```
vim /etc/hosts
```

```
10.30.68.83 VM_0_1_centos
```

```
10.30.68.82 VM_0_2_centos
```

## 2. 关闭 selinux(可选)

在目标服务器 (10.30.68.83 10.30.68.82) root 用户下执行:

确认是否已安装 selinux

centos 系统执行: rpm -qa | grep selinux

ubuntu 系统执行: apt list --installed | grep selinux

如果已安装了 selinux 就执行: setenforce 0

## 3. 修改 Linux 系统参数

在目标服务器 (10.30.68.83 10.30.68.82) root 用户下执行:

1) vim /etc/security/limits.conf

\* soft nofile 65535





\* hard nofile 65535

2) vim /etc/security/limits.d/20-nproc.conf

\* soft nproc unlimited

## 4. 关闭防火墙(可选)

在目标服务器 (10.30.68.83 10.30.68.82) root 用户下执行

如果是 Centos 系统:

```
systemctl disable firewalld.service
```

```
systemctl stop firewalld.service
```

```
systemctl status firewalld.service
```

如果是 Ubuntu 系统:

```
ufw disable
```

```
ufw status
```

## 5. 软件环境初始化

在目标服务器 (10.30.68.83 10.30.68.82) root 用户下执行

### 5.1 创建用户

```
groupadd fate
```

```
useradd -s /bin/bash -g fate -d /home/ fate fate
```

```
passwd fate
```



## 5.2 创建目录

```
mkdir -pv /data/projects /data/temp /data/logs
chown -R fate:fate /data/projects /data/temp /data/logs
```

## 5.3 安装依赖

```
#centos
yum -y install gcc gcc-c++ make openssl-devel gmp-devel mpfr-devel libmpc-devel libaio numactl
autoconf automake libtool libffi-devel snappy snappy-devel zlib zlib-devel bzip2 bzip2-devel lz4-
devel libasan lsof sysstat telnet psmisc
#ubuntu
apt-get install -y gcc g++ make openssl supervisor libgmp-dev libmpfr-dev libmpc-dev libaio1
libaio-dev numactl autoconf automake libtool libffi-dev libssl1.0.0 libssl-dev liblz4-1 liblz4-dev
liblz4-1-dbg liblz4-tool zlib1g zlib1g-dbg zlib1g-dev
cd /usr/lib/x86_64-linux-gnu
if [ ! -f "libssl.so.10" ];then
    ln -s libssl.so.1.0.0 libssl.so.10
    ln -s libcrypto.so.1.0.0 libcrypto.so.10
fi
```

## 6. 增加虚拟内存

在目标服务器（10.30.68.83 10.30.68.82）root 用户下执行

生产环境使用时，因内存计算需要增加 128G 虚拟内存，参考：

```
cd /data
dd if=/dev/zero of=/data/swapfile128G bs=1024 count=134217728
mkswap /data/swapfile128G
swapon /data/swapfile128G
cat /proc/swaps
echo '/data/swapfile128G swap swap defaults 0 0' >> /etc/fstab
```

## 第五部分：项目部署



注：此指导安装目录默认为/data/projects/install，执行用户为 fate，安装时根据具体实际情况修改。

## 1. 获取安装包

在目标服务器（10.30.68.83 10.30.68.82 具备外网环境） fate 用户下执行：

```
mkdir -p /data/projects/install
cd /data/projects/install
wget
https://webank-ai-1251170195.cos.ap-guangzhou.myqcloud.com/AnsibleFATE\_1.7.2\_release-offline.tar.gz
```

## 2. 操作系统参数检查

在目标服务器（10.30.68.83 10.30.68.82） fate 用户下执行

#虚拟内存，size 不低于 128G，如不满足需参考 4.6 章节重新设置

```
cat /proc/swaps
Filename                                Type      Size      Used      Priority
/data/swapfile128G                      file      134217724 384      -1
```

#文件句柄数，不低于 65535，如不满足需参考 4.3 章节重新设置

```
ulimit -n
65535
```

#用户进程数，不低于 64000，如不满足需参考 4.3 章节重新设置

```
ulimit -u
65535
```

## 3. 部署

在目标服务器（10.30.68.83 10.30.68.82） fate 用户下执行



### 3.1 进入部署根目录:

```
cd 【部署包根目录】
```

### 3.2 初始化配置

步骤 1: 使用辅助脚本产生初始化配置, 分别在 10.30.68.83, 10.30.68.82 上执行

```
sh deploy/deploy.sh init -h="10000: 10.30.68.83" (-g="9999: 10.30.68.82") //括号中 在 10.30.68.82 执行
```

步骤 2: 按需修改配置

```
vim deploy/conf/setup.conf
```

```
env: prod
```

```
pname: fate
```

```
ssh_port: 22
```

```
deploy_user: fate
```

```
deploy_group: fate
```

```
deploy_mode: deploy
```

```
modules:
```

```
- mysql
```

```
- eggroll
```

```
- fate_flow
```

```
- fateboard
```

```
roles:
```

```
- host:10000 (-guest:9999)
```

```
ssl_roles: []
```

```
polling: {}
```

```
host_ips(guest_ips):
```

```
- default: 10.30.68.83 (10.30.68.82)
```

```
host_special_routes (guest _special_routes):
```

```
- default:[] //可以设置额外路由指向 exchange
```

```
guest_ips(host_ips): []
```

```
guest_special_routes(host_special_routes): []
```

```
exchange_ips: []
```

```
exchange_special_routes: []
```

```
default_engines: eggroll
```

步骤 3: 执行辅助脚本产生配置

```
sh deploy/deploy.sh render
```



## 4. 配置 host 信息

在目标服务器 (10.30.68.83) fate 用户下执行:

```
vi var_files/prod/fate_host
```

内容如下:

```
host:
```

```
partyid: 10000
```

```
rollsite:
```

```
enable: True
```

```
coordinator: fate
```

```
ips:
```

```
- 10.30.68.83
```

```
port: 9370
```

```
secure_port: 9371
```

```
server_secure: False
```

```
client_secure: False
```

```
polling:
```

```
enable: False
```

```
route_tables:
```

```
- id: default
```

```
routes:
```

```
- name: default
```

```
ip: 10.30.68.82
```

```
port: 9370
```

```
is_secure: False
```

```
- id: 10000
```

```
routes:
```

```
- name: default
```

```
ip: 10.30.68.83
```

```
port: 9370
```

```
is_secure: false
```

```
- name: fateflow
```

```
ip: 10.30.68.83
```

```
port: 9360
```

```
clustermanager:
```

```
enable: True
```

```
ips:
```

```
- 10.30.68.83
```

```
port: 4670
```

```
cores_per_node: 16
```

```
nodemanager:
```



```
enable: True
ips:
- 10.30.68.83
port: 4671
eggroll:
  dbname: "eggroll_meta"
  egg: 4
fate_flow:
  enable: True
  ips:
  - 10.30.68.83
  grpcPort: 9360
  httpPort: 9380
  dbname: "fate_flow"
  proxy: rollsite
  http_app_key:
  http_secret_key:
  use_deserialize_safe_module: false
  default_engines: eggroll
fateboard:
  enable: True
  ips:
  - 10.30.68.83
  port: 8080
  dbname: "fate_flow"
mysql:
  enable: True
  type: inside
  ips:
  - 10.30.68.83
  port: 3306
  dbuser: "fate"
  dbpasswd: "xxx"//设置强密码
zk:
  enable: False
  lists:
  - ip: 192.168.0.1
    port: 2181
  use_acl: false
  user: "fate"
  passwd: "fate"
servings:
  ips:
  - 192.168.0.1
```

## 5. 配置 guest 信息

在目标服务器 (10.30.68.82) fate 用户下执行:

```
vi var_files/prod/fate_host
```

内容如下:

```
host:
```

```
partyid: 9999
```

```
rollsite:
```

```
enable: True
```

```
coordinator: fate
```

```
ips:
```

```
- 10.30.68.82
```

```
port: 9370
```

```
secure_port: 9371
```

```
server_secure: False
```

```
client_secure: False
```

```
polling:
```

```
enable: False
```

```
route_tables:
```

```
- id: default
```

```
routes:
```

```
- name: default
```

```
ip: 10.30.68.83
```

```
port: 9370
```

```
is_secure: False
```

```
- id: 9999
```

```
routes:
```

```
- name: default
```

```
ip: 10.30.68.82
```

```
port: 9370
```

```
is_secure: false
```

```
- name: fateflow
```

```
ip: 10.30.68.82
```

```
port: 9360
```

```
clustermanager:
```

```
enable: True
```

```
ips:
```

```
- 10.30.68.82
```

```
port: 4670
```

```
cores_per_node: 16
```



```
nodemanager:
  enable: True
  ips:
    - 10.30.68.82
  port: 4671
eggroll:
  dbname: "eggroll_meta"
  egg: 4
fate_flow:
  enable: True
  ips:
    - 10.30.68.82
  grpcPort: 9360
  httpPort: 9380
  dbname: "fate_flow"
  proxy: rollsite
  http_app_key:
  http_secret_key:
  use_deserialize_safe_module: false
  default_engines: eggroll
fateboard:
  enable: True
  ips:
    - 10.30.68.82
  port: 8080
  dbname: "fate_flow"
mysql:
  enable: True
  type: inside
  ips:
    - 10.30.68.82
  port: 3306
  dbuser: "fate"
  dbpasswd: "xxx"//设置强密码
zk:
  enable: False
  lists:
    - ip: 192.168.0.1
      port: 2181
  use_acl: false
  user: "fate"
  passwd: "fate"
servings:
  ips:
```





- 192.168.0.1

port: 8000

## 6. 部署所有服务

```
sh deploy/deploy.sh deploy
```

查看部署日志: `tailf logs/deploy-?.log`

## 7. eggroll 路由配置文件修改

此配置文件 `rollsite` 使用, 配置路由信息, 可以参考如下例子手工配置, 也可以使用以下指令完成. 配置文件: `/data/projects/fate/eggroll/conf/route_table.json`

#在目标服务器 (10.30.68.83) fate 用户下修改执行

```
cat > /data/projects/fate/eggroll/conf/route_table.json << EOF
```

```
{
  "route_table":
  {
    "10000":
    {
      "default":[
        {
          "port": 9370,
          "ip": "10.30.68.83"
        }
      ],
      "fateflow":[
        {
          "port": 9360,
          "ip": "10.30.68.83"
        }
      ]
    },
    "9999":
    {
      "default":[
        {
          "port": 9370,
          "ip": "10.30.68.82"
        }
      ]
    }
  }
}
```



```
},
"permission":
{
  "default_allow": true
}
}
EOF

#在目标服务器（10.30.68.82） fate 用户下修改执行
cat > /data/projects/fate/eggroll/conf/route_table.json << EOF
{
  "route_table":
  {
    "9999":
    {
      "default":[
        {
          "port": 9370,
          "ip": "10.30.68.82"
        }
      ],
      "fateflow":[
        {
          "port": 9360,
          "ip": "10.30.68.82"
        }
      ]
    },
    "10000":
    {
      "default":[
        {
          "port": 9370,
          "ip": "10.30.68.83"
        }
      ]
    }
  },
  "permission":
  {
    "default_allow": true
  }
}
EOF
```



```
#在目标服务器（10.30.68.83、10.30.68.82） fate 用户下执行：  
/bin/bash /data/projects/common/supervisord/service.sh restart fate-rollsites
```

## 8. 问题定位

### 1) eggroll 日志

```
/data/logs/fate/eggroll/eggroll/bootstrap.clustermanager.err
```

```
/data/logs/fate/eggroll/eggroll/eggroll/clustermanager.jvm.err.log
```

```
/data/logs/fate/eggroll/eggroll/nodemanager.jvm.err.log
```

```
/data/logs/fate/eggroll/eggroll/bootstrap.nodemanager.err
```

```
/data/logs/fate/eggroll/eggroll/bootstrap.rollsite.err
```

```
/data/logs/fate/eggroll/eggroll/rollsite.jvm.err.log
```

### 2) fateflow 日志

```
/data/logs/fate/fateflow/error.log
```

### 3) fateboard 日志

```
/data/logs/fate/fateboard/
```

## 第六部分：测试

### 1. Toy\_example 部署验证

此测试您需要设置 3 个参数： guest\_partyid, host\_partyid, work\_mode。

#### 1.1 单边测试

1) 10.30.68.83 上执行， guest\_partyid 和 host\_partyid 都设为 10000：



```
source /data/projects/fate/bin/init_env.sh
flow test toy -gid 10000 -hid 10000
```

类似如下结果表示成功:

```
"2020-04-28 18:26:20,789 - secure_add_guest.py[line:126] - INFO: success to calculate
secure_sum,it is 1999.9999999999998"
```

2) 10.30.68.82 上执行, guest\_partyid 和 host\_partyid 都设为 9999:

```
source /data/projects/fate/bin/init_env.sh
flow test toy -gid 9999 -hid 9999
```

类似如下结果表示成功:

```
"2020-04-28 18:26:20,789 - secure_add_guest.py[line:126] - INFO: success to calculate
secure_sum, it is 1999.9999999999998"
```

## 1.2 双边测试

选定 9999 为 guest 方, 在 10.30.68.82 fate 用户上执行:

```
source /data/projects/fate/init_env.sh
flow test toy -gid 9999 -hid 10000
```

类似如下结果表示成功:

```
"2020-04-28 18:26:20,789 - secure_add_guest.py[line:126] - INFO: success to calculate
secure_sum, it is 1999.9999999999998"
```

## 2. 最小化测试

### 2.1 上传预设数据

分别在 10.30.68.83 和 10.30.68.82 fate 用户上执行:

```
source /data/projects/fate/bin/init_env.sh
cd /data/projects/fate/examples/scripts/
python upload_default_data.py
```



## 2.2 快速模式

请确保 guest 和 host 两方均已分别通过给定脚本上传了预设数据。

快速模式下，最小化测试脚本将使用一个相对较小的数据集，即包含了 569 条数据的 breast 数据集。

选定 9999 为 guest 方，在 10.30.68.82 fate 用户上执行：

```
source /data/projects/fate/bin/init_env.sh
cd /data/projects/fate/examples/min_test_task/
#单边测试
python run_task.py -gid 9999 -hid 9999 -aid 9999 -f fast
#双边测试
python run_task.py -gid 9999 -hid 10000 -aid 10000 -f fast
```

其他一些可能有用的参数包括：

1. -f: 使用的文件类型. "fast" 代表 breast 数据集, "normal" 代表 default credit 数据集.
2. --add\_sbt: 如果被设置为 1, 将在运行完 lr 以后, 启动 secureboost 任务, 设置为 0 则不启动 secureboost 任务, 不设置此参数系统默认为 1.

若数分钟后在结果中显示了“success”字样则表明该操作已经运行成功了。若出现“FAILED”或者程序卡住，则意味着测试失败。

## 2.3 正常模式

只需在命令中将“fast”替换为“normal”，其余部分与快速模式相同。



### 3. Fateboard testing

Fateboard 是一项 Web 服务。如果成功启动了 fateboard 服务，则可以通过访问 <http://10.30.68.83:8080> 和 <http://10.30.68.82:8080> 来查看任务信息。

## 第七部分：任务管理

### 1. 任务创建

在目标服务器（10.30.68.83 10.30.68.82）fate 用户下执行

1、guest(10.30.68.82)和 host(10.30.68.83)双方都需先执行数据上传任务，执行以下命令：

```
source /data/projects/fate/bin/init_env.sh
```

host 上传数据任务，执行以下命令：

```
flow data upload -c examples/dsl/v2/upload/upload_conf.json
# upload_conf.json 内容
{
  "file": "/data/projects/fate/examples/data/breast_hetero_host.csv",
  "table_name": "breast_hetero_host ",
  "namespace": "experiment",
  "head": 1,
  "partition": 8
}
```

guest 数据上传，执行以下命令：

```
flow data upload -c examples/dsl/v2/upload/upload_conf.json
# upload_conf.json 内容
{
  "file": "/data/projects/fate/examples/data/breast_hetero_guest.csv",
  "table_name": "breast_hetero_guest",
  "namespace": "experiment",
  "head": 1,
  "partition": 8
}
```

2、模型训练、预测任务执行，在 guest(10.30.68.82) fate 用户下执行以下命令：



以 secureboost train 为例。

```
flow job submit-c ./examples/dsl/v2/hetero_secureboost/test_secureboost_train_binary_conf.json -  
d ./examples/dsl/v2/hetero_secureboost/test_secureboost_train_dsl.json
```

## 2. 查看任务

登录 fateboard 任务看板，单击“任务(jobs)”，进入任务中心页面。

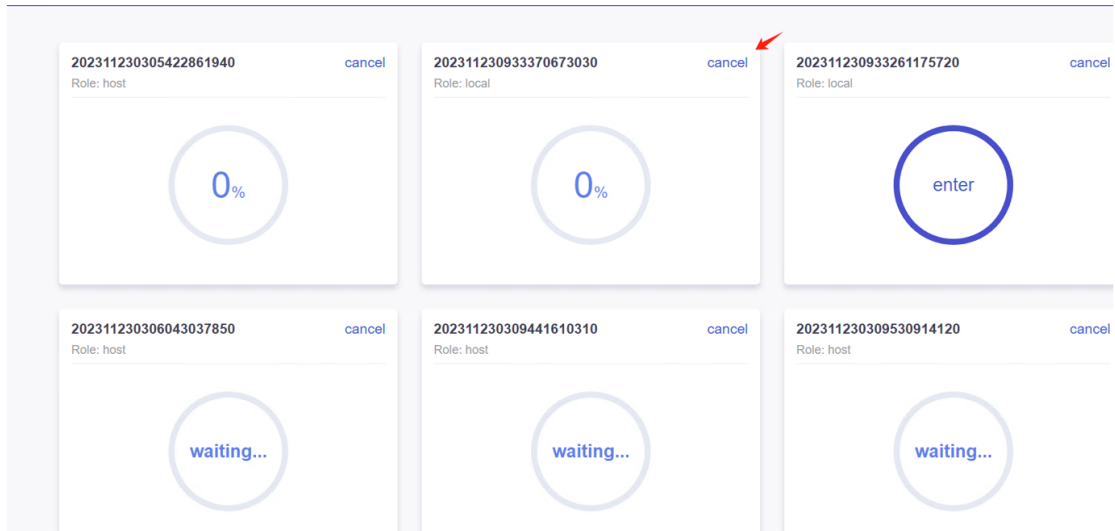
可进行任务 ID、任务角色、任务 Party ID、任务状态、任务类型进行筛选。

ID	Role	Party ID	Partner	Start Time	End Time	Duration
202311231014472761850	host	20000	9994	2023-11-23 10:14:49	2023-11-23 10:16:22	00:01:32
202311231013270542610	host	20000	9994	2023-11-23 10:13:32	2023-11-23 10:14:25	00:00:53
202311231011534296720	host	20000	9994	2023-11-23 10:11:56	2023-11-23 10:13:18	00:01:22
202311231010258490910	host	20000	9994	2023-11-23 10:10:28	2023-11-23 10:12:08	00:01:39
202311231008365725230	host	20000	9994	2023-11-23 10:08:39	2023-11-23 10:09:32	00:00:53
202311231007176262400	host	20000	9994	2023-11-23 10:07:20	2023-11-23 10:08:07	00:00:47

## 3. 取消任务

对于不再需要执行的任务，支持通过“任务”进行任务的取消。取消任务仅取消执行中的任务，不会删除任务。

- 1、登录 fateboard 任务看板。
- 2、在“运行(running)”页面，选择目标任务，操作列的“取消 (cancel)”，在弹出框中点击“确定”，取消任务。



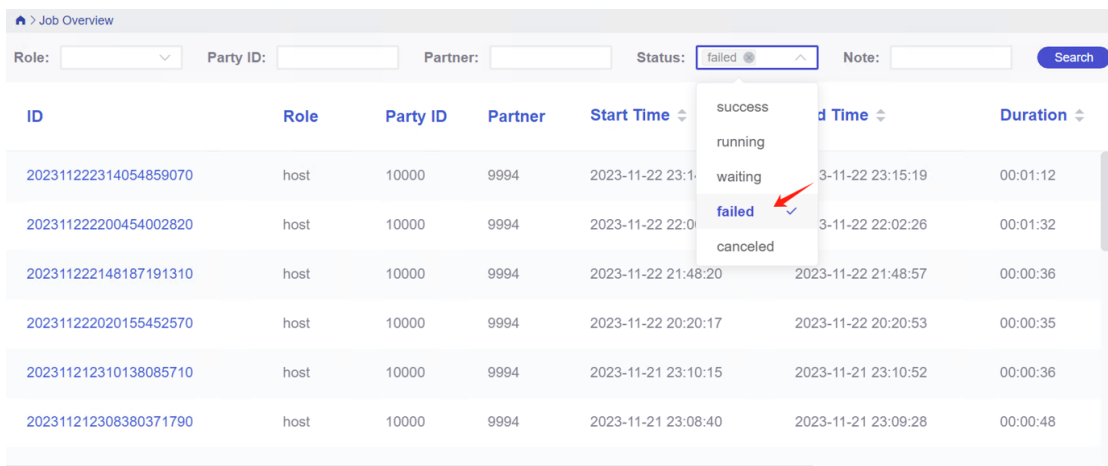
#### 4. 查看任务日志明细

产品服务的日志管理功能支持查看算法级别的日志，包括算法运行的错误信息，有助于您分析系统中存在的问题。

运行日志记录了算法运行时的 warning、error、debug 和 info 级别的日志。您可以通过运行日志分析系统中存在的问题。

1. 登录 fateboard 看板。

2. 单击任务(jobs),再过滤状态(Status) failed 的任务。



3. 单击查看失败任务的 ID。

4. 选择任务失败的算法，再单击 view the outputs 查看错误日志的详细信息。



Dashboard > Job detail

status: failed

notes:

role: host

party\_ID: 10000

dataset: experime...

[Job DSL](#)

[Runtime config](#)

submission time: 2023-11-22 23:14:05

start time: 2023-11-22 23:14:06

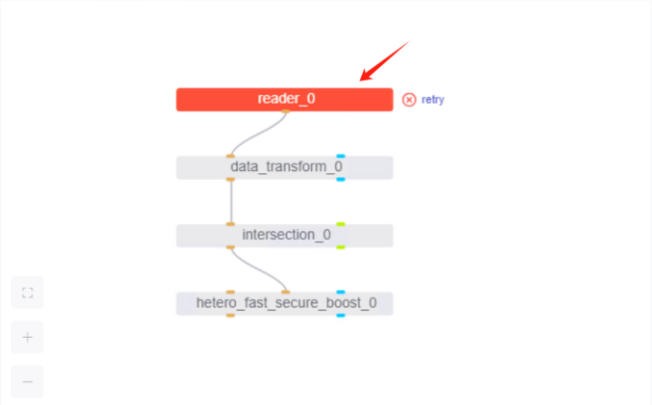
end time: 2023-11-22 23:15:19

duration: 00:01:12

[dashboard](#)

### Outputs From Job

Main Graph Click component to view details



Parameter(4) unfold all

module: Reader

> ComponentParam

[view the outputs](#)

## 第八部分：系统运维

### 1. 服务管理

#### 1.1 Eggroll 服务管理

```
cd /data/projects/common/supervisord
```

启动/关闭/查看/重启单个模块(可选: clustermanager, nodemanager, rollsite):

```
sh service.sh start/stop/status/restart fate-clustermanager
```

```
sh service.sh start/stop/status/restart fate- nodemanager
```

```
sh service.sh start/stop/status/restart fate- rollsite
```

#### 1.2 Fate 服务管理

1. 启动/关闭/查看/重启 fate\_flow 服务

```
sh service.sh start/stop/status/restart fate-fateflow
```

2. 启动/关闭/重启 fateboard 服务

```
sh service.sh start/stop/status/restart fate-fateboard
```



## 1.3 Mysql 服务管理

启动/关闭/查看/重启 mysql 服务

```
sh service.sh start/stop/status/restart fate-mysql
```

## 2. 查看进程和端口

在目标服务器 (10.30.68.83 10.30.68.82) fate 用户下执行

### 2.1 查看进程

```
ps aux|grep fate  
/bin/bash /data/projects/common/supervisord/service.sh status all
```

### 2.2 查看进程端口

```
#根据部署规划查看进程端口是否存在  
#clustermanager  
netstat -tlnp | grep 4670  
#nodemanager  
netstat -tlnp | grep 4671  
#rollsite  
netstat -tlnp | grep 9370  
#fate_flow_server  
netstat -tlnp | grep 9360  
#fateboard  
netstat -tlnp | grep 8080  
#mysql  
netstat -tlnp | grep 3306
```

## 3. 服务日志

服务	日志路径
eggroll	/data/logs/fate/eggroll/



fate_flow	/data/logs/fate/fateflow/
fateboard	/data/logs/fate/fateboard/
mysql	/data/logs/mysql/

## 第九部分：附录

### 1. Eggroll 参数调优

配置文件路径：/data/projects/fate/eggroll/conf/eggroll.properties

配置参数：eggroll.session.processors.per.node

假定 CPU 核数 (cpu cores) 为  $c$ , Nodemanager 的数量为  $n$ , 需要同时运行的任务数为  $p$ , 则:

$\text{egg\_num} = \text{eggroll.session.processors.per.node} = c * 0.8 / p$

$\text{partitions (roll pair 分区数)} = \text{egg\_num} * n$