



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
第一部分： 服务器配置	5
第二部分： 集群规划	5
1. 集群配置	5
2. 架构图	6
第三部分： 组件说明	6
第四部分： 基础环境配置	7
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	10
6. 增加虚拟内存	10
第五部分： 项目部署	11
1. 获取安装包	11
2. 操作系统参数检查	11
3. 部署 mysql	12
4. 部署 jdk	14
5. 部署 python	15
6. 部署 eggroll&fate	15
7. 启动服务	26

8. 问题定位.....	27
第六部分：测试.....	28
1. Toy_example 部署验证	28
1.1 单边测试.....	28
1.2 双边测试.....	29
2. 最小化测试.....	29
2.1 上传预设数据.....	29
2.2 快速模式.....	30
2.3 正常模式.....	30
3. Fateboard testing	31
第七部分：系统运维.....	31
1. 服务管理.....	31
1.1 Eggroll 服务管理	31
1.2 Fate 服务管理.....	31
1.3 Mysql 服务管理	32
2. 查看进程和端口.....	32
2.1 查看进程.....	32
2.2 查看进程端口.....	32
3. 服务日志.....	33
第八部分：附录.....	33
1. eggroll&fate 打包构建	33
2. Eggroll 参数调优	33

版本更新表

[illegible]

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	8 core /16GB memory / 500GB 硬盘/10M 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：fl，属主：root（fl 用户需可以 sudo su root 而无需密码，创建 fl 没有创建 fl 的 group）
文件系统	1. 500G 硬盘挂载在/ data 目录下； 2.创建/ data / projects 目录，目录属主为：fl:root

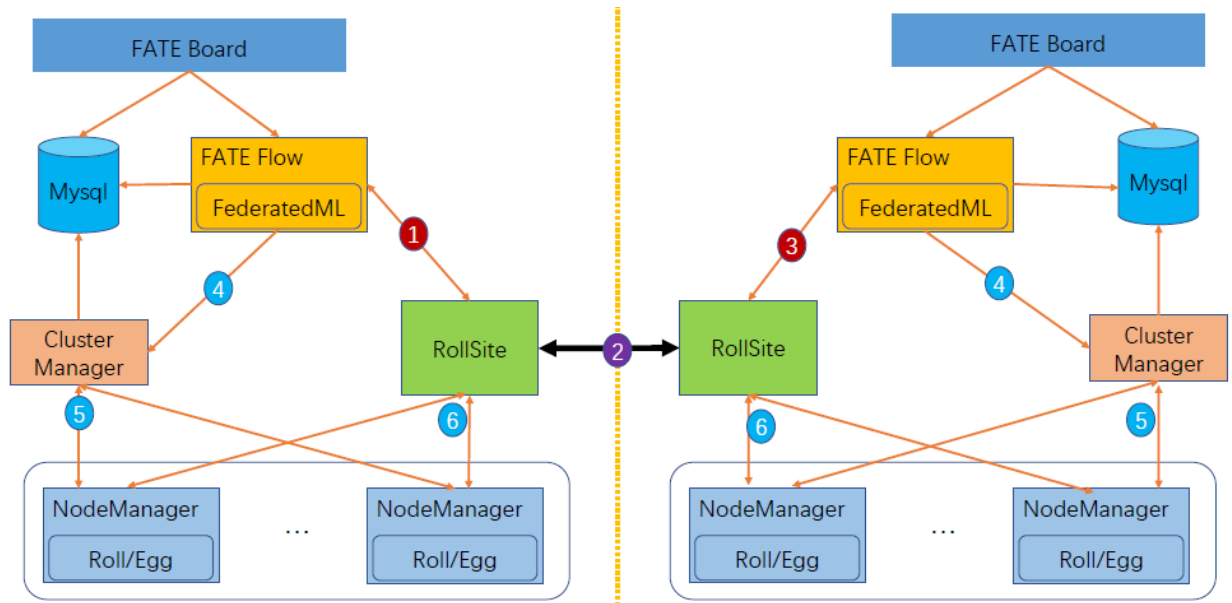
第二部分： 集群规划

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
PartyA	10000	VM_0_2_centos	10.30.68.84	CentOS 7.2/Ubuntu 16.04	fate,eggroll	nodemanager, rollsite
PartyB	9999	VM_0_3_centos	10.30.68.82	CentOS 7.2/Ubuntu 16.04	fate, eggroll, mysql	all

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 10.30.68.84 root 用户下执行:

```
hostnamectl set-hostname VM_0_2_centos
```

在 10.30.68.82 root 用户下执行:

```
hostnamectl set-hostname VM_0_3_centos
```

2) 加入主机映射

在目标服务器 (10.30.68.83 10.30.68.84 10.30.68.82) root 用户下执行:

```
vim /etc/hosts
```

```
10.30.68.83 VM_0_1_centos
```

```
10.30.68.84 VM_0_2_centos
```

```
10.30.68.82 VM_0_3_centos 文档名称: 系统固定标准
```

2. 关闭 selinux(可选)

在目标服务器 (10.30.68.83 10.30.68.84 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 10.30.68.84 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 10.30.68.84 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 10.30.68.84 10.30.68.82) root 用户下执行

5.1 创建用户

```
<!-- groupadd -g 6000 apps
useradd -s /bin/bash -g apps -d /home/app app
passwd app -->
```

5.2 创建目录

```
mkdir -p /data/projects/fate
mkdir -p /data/projects/install
<!-- chown -R app:apps /data/projects -->
chown -R fl /data/projects
```

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-dbgsymbols liblz4-tool zlib1g zlib1g-dbgsymbols zlib1g-dev
cd /usr/lib/x86_64-linux-gnu
if [ ! -f "libssl.so.1.0" ];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 10.30.68.84 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 fl，安装时根据实际情况修改。

1. 获取安装包

在目标服务器（10.30.68.83 具备外网环境）fl 用户下执行：

```
mkdir -p /data/projects/install
cd /data/projects/install
wget https://webank-ai-1251170195.cos.ap-guangzhou.myqcloud.com/python-env-1.4.2-release.tar.gz
wget https://webank-ai-1251170195.cos.ap-guangzhou.myqcloud.com/jdk-8u192-linux-x64.tar.gz
wget https://webank-ai-1251170195.cos.ap-guangzhou.myqcloud.com/mysql-1.4.2-release.tar.gz
wget https://webank-ai-1251170195.cos.ap-guangzhou.myqcloud.com/FATE_install_1.4.2-release.tar.gz

#传输到 10.30.68.84 和 10.30.68.82
scp *.tar.gz fl@10.30.68.84:/data/projects/install
scp *.tar.gz fl@10.30.68.82:/data/projects/install
```

2. 操作系统参数检查

在目标服务器（10.30.68.83 10.30.68.84 10.30.68.82）fl 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 mysql

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

3.1 mysql 安装:

```
#建立 mysql 根目录  
mkdir -p /data/projects/fate/common/mysql  
mkdir -p /data/projects/fate/data/mysql  
  
#解压软件包  
cd /data/projects/install  
tar xzvf mysql-*.tar.gz  
cd mysql  
tar xf mysql-8.0.13.tar.gz -C /data/projects/fate/common/mysql  
  
#配置设置  
mkdir -p /data/projects/fate/common/mysql/mysql-8.0.13/{conf,run,logs}  
cp service.sh /data/projects/fate/common/mysql/mysql-8.0.13/  
cp my.cnf /data/projects/fate/common/mysql/mysql-8.0.13/conf  
  
#初始化  
cd /data/projects/fate/common/mysql/mysql-8.0.13/  
./bin/mysqld --initialize --user=fl --basedir=/data/projects/fate/common/mysql/mysql-8.0.13 --  
datadir=/data/projects/fate/data/mysql > logs/init.log 2>&1
```

```
cat logs/init.log |grep root@localhost
```

#注意输出信息中 root@localhost:后的是 mysql 用户 root 的初始密码，需要记录，后面修改密码需要用到

#启动服务

```
cd /data/projects/fate/common/mysql/mysql-8.0.13/
```

```
nohup ./bin/mysqld_safe --defaults-file=./conf/my.cnf --user=fl >>logs/mysqld.log 2>&1 &
```

#修改 mysql root 用户密码

```
cd /data/projects/fate/common/mysql/mysql-8.0.13/
```

```
./bin/mysqladmin -h 127.0.0.1 -P 3306 -S ./run/mysql.sock -u root -p password "fate_dev"
```

Enter Password: 【输入 root 初始密码】

#验证登陆

```
cd /data/projects/fate/common/mysql/mysql-8.0.13/
```

```
./bin/mysql -u root -p -S ./run/mysql.sock
```

Enter Password: 【输入 root 修改后密码:fate_dev】

3.2 建库授权和业务配置

```
cd /data/projects/fate/common/mysql/mysql-8.0.13/
```

```
./bin/mysql -u root -p -S ./run/mysql.sock
```

Enter Password: 【fate_dev】

#创建 eggroll 库表

```
mysql>source /data/projects/install/mysql/create-eggroll-meta-tables.sql;
```

#创建 fate_flow 库

```
mysql>CREATE DATABASE IF NOT EXISTS fate_flow;
```

#创建远程用户和授权

1) 10.30.68.83 执行

```
mysql>CREATE USER 'fate'@'%' IDENTIFIED BY 'fate_dev';
```

```
mysql>GRANT ALL ON *.* TO 'fate'@'%';
```

```
mysql>CREATE USER 'fate'@'%' IDENTIFIED BY 'fate_dev';
```

```
mysql>GRANT ALL ON *.* TO 'fate'@'%';
```

```
mysql>flush privileges;
```

2) 10.30.68.82 执行

```
mysql>CREATE USER 'fate'@'%' IDENTIFIED BY 'fate_dev';
```

```
mysql>GRANT ALL ON *.* TO 'fate'@'%';
```

```
mysql>flush privileges;
```

```
#insert 配置数据
1) 10.30.68.83 执行
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.83', '4670',
'CLUSTER_MANAGER', 'HEALTHY');
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.83', '4671',
'NODE_MANAGER', 'HEALTHY');
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.84', '4671',
'NODE_MANAGER', 'HEALTHY');

---新增
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.85', '4671',
'NODE_MANAGER', 'HEALTHY');
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.86', '4671',
'NODE_MANAGER', 'HEALTHY');
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.89', '4671',
'NODE_MANAGER', 'HEALTHY');

2) 10.30.68.82 执行
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.82', '4670',
'CLUSTER_MANAGER', 'HEALTHY');
mysql>INSERT INTO server_node (host, port, node_type, status) values ('10.30.68.82', '4671',
'NODE_MANAGER', 'HEALTHY');

#校验
mysql>select User,Host from mysql.user;
mysql>show databases;
mysql>use eggroll_meta;
mysql>show tables;
mysql>select * from server_node;
```

4. 部署 jdk

在目标服务器 (10.30.68.83 10.30.68.84 10.30.68.82) fl 用户下执行:

```
#创建 jdk 安装目录
mkdir -p /data/projects/fate/common/jdk
#解压缩
cd /data/projects/install
tar xzf jdk-8u192-linux-x64.tar.gz -C /data/projects/fate/common/jdk
cd /data/projects/fate/common/jdk
```

```
mv jdk1.8.0_192 jdk-8u192
```

5. 部署 python

在目标服务器 (10.30.68.83 10.30.68.84 10.30.68.82) fl 用户下执行:

```
#创建 python 虚拟化安装目录
mkdir -p /data/projects/fate/common/python

#安装 miniconda3
cd /data/projects/install
tar xvf python-env-*.tar.gz
cd python-env
sh Miniconda3-4.5.4-Linux-x86_64.sh -b -p /data/projects/fate/common/miniconda3

#安装 virtualenv 和创建虚拟化环境
/data/projects/fate/common/miniconda3/bin/pip install virtualenv-20.0.18-py2.py3-none-any.whl -f . --no-index

/data/projects/fate/common/miniconda3/bin/virtualenv -p
/data/projects/fate/common/miniconda3/bin/python3.6 --no-wheel --no-setuptools --no-download /data/projects/fate/common/python/venv

#安装依赖包
tar xvf pip-packages-fate-*.tar.gz
source /data/projects/fate/common/python/venv/bin/activate
pip install setuptools-42.0.2-py2.py3-none-any.whl
pip install -r pip-packages-fate-1.4.2/requirements.txt -f ./pip-packages-fate-1.4.2 --no-index
pip list | wc -l
#结果应为 161
```

6. 部署 eggroll&fate

6.1 软件部署

```
#部署软件
#在目标服务器 (10.30.68.83 10.30.68.84 10.30.68.82) fl 用户下执行:
cd /data/projects/install
tar xf FATE_install_*.tar.gz
cd FATE_install_*
```

```
tar xvf python.tar.gz -C /data/projects/fate/
tar xvf eggroll.tar.gz -C /data/projects/fate

#在目标服务器 (10.30.68.83 10.30.68.82) fl 用户下执行:
tar xvf fateboard.tar.gz -C /data/projects/fate

#设置环境变量文件
#在目标服务器 (10.30.68.83 10.30.68.84 10.30.68.82) fl 用户下执行:
cat >/data/projects/fate/init_env.sh <<EOF
export PYTHONPATH=/data/projects/fate/python:/data/projects/fate/eggroll/python
export EGGROLL_HOME=/data/projects/fate/eggroll/
venv=/data/projects/fate/common/python/venv
source \${venv}/bin/activate
export JAVA_HOME=/data/projects/fate/common/jdk/jdk-8u192
export PATH=\$PATH:\$JAVA_HOME/bin
EOF
```

6.2 eggroll 系统配置文件修改

此配置文件 rollsite, clustermanager, nodemanager 共用, 每端 party 多台主机保持一致, 需修改内容:

- 数据库驱动, 数据库对应 party 用的连接 IP, 端口, 用户名和密码, 端口一般默认即可。

eggroll.resourcemanager.clustermanager.jdbc.driver.class.name

eggroll.resourcemanager.clustermanager.jdbc.username

eggroll.resourcemanager.clustermanager.jdbc.password

- 对应 party clustermanager 的 IP、端口, nodemanager 端口, 进程 tag, 端口一般默认即可。

eggroll.resourcemanager.clustermanager.host

eggroll.resourcemanager.clustermanager.port

eggroll.resourcemanager.nodemanager.port

eggroll.resourcemanager.process.tag

- Python 虚拟环境路径、业务代码 pythonpath、JAVA Home 路径修改，如果相关路径无变化，保持默认即可。

eggroll.resourcemanager.bootstrap.egg_pair.venv

eggroll.resourcemanager.bootstrap.egg_pair.pythonpath

eggroll.resourcemanager.bootstrap.roll_pair_master.javahome

- 对应 party rollsite 的 IP、端口、本 party 的 Party Id 修改，rollsite 的端口一般默认即可。

eggroll.rollsite.host eggroll.rollsite.port eggroll.rollsite.party.id

以上参数调整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/eggroll.properties

```
#在目标服务器（10.30.68.83 10.30.68.84）fl 用户下修改执行
cat > /data/projects/fate/eggroll/conf/eggroll.properties <<EOF
[eggroll]
#db connect inf
eggroll.resourcemanager.clustermanager.jdbc.driver.class.name=com.mysql.cj.jdbc.Driver
eggroll.resourcemanager.clustermanager.jdbc.url=jdbc:mysql://10.30.68.83:3306/eggroll_meta?
useSSL=false&serverTimezone=UTC&characterEncoding=utf8&allowPublicKeyRetrieval=true
eggroll.resourcemanager.clustermanager.jdbc.username=fate
eggroll.resourcemanager.clustermanager.jdbc.password=fate_dev
eggroll.data.dir=data/
eggroll.logs.dir=logs/
#clustermanager & nodemanager
eggroll.resourcemanager.clustermanager.host=10.30.68.83
eggroll.resourcemanager.clustermanager.port=4670
eggroll.resourcemanager.nodemanager.port=4671
eggroll.resourcemanager.process.tag=fate-host
eggroll.bootstrap.root.script=bin/eggroll_boot.sh
eggroll.resourcemanager.bootstrap.egg_pair.exepath=bin/roll_pair/egg_pair_bootstrap.sh
#python env
eggroll.resourcemanager.bootstrap.egg_pair.venv=/data/projects/fate/common/python/venv
#pythonpath, very import, do not modify.
```

```
eggroll.resourcemanager.bootstrap.egg_pair.pythonpath=/data/projects/fate/python:/data/projects/fate/eggroll/python
eggroll.resourcemanager.bootstrap.egg_pair.filepath=python/eggroll/roll_pair/egg_pair.py
eggroll.resourcemanager.bootstrap.roll_pair_master.exepath=bin/roll_pair/roll_pair_master_bootstrap.sh
#javahome
eggroll.resourcemanager.bootstrap.roll_pair_master.javahome=/data/projects/fate/common/jdk/jdk-8u192
eggroll.resourcemanager.bootstrap.roll_pair_master.classpath=conf:/lib/*
eggroll.resourcemanager.bootstrap.roll_pair_master.mainclass=com.webank.eggroll.rollpair.RollPairMasterBootstrap
eggroll.resourcemanager.bootstrap.roll_pair_master.jvm.options=
# for roll site. rename in the next round
eggroll.rollsite.coordinator=webank
eggroll.rollsite.host=10.30.68.84
eggroll.rollsite.port=9370
eggroll.rollsite.party.id=10000
eggroll.rollsite.route.table.path=conf/route_table.json

eggroll.session.processors.per.node=4
eggroll.session.start.timeout.ms=180000
eggroll.rollsite.adapter.sendbuf.size=1048576
eggroll.rollpair.transferpair.sendbuf.size=4150000
EOF

#在目标服务器（10.30.68.82） fl 用户下修改执行
cat > /data/projects/fate/eggroll/conf/eggroll.properties <<EOF
[eggroll]
#db connect inf
eggroll.resourcemanager.clustermanager.jdbc.driver.class.name=com.mysql.cj.jdbc.Driver
eggroll.resourcemanager.clustermanager.jdbc.url=jdbc:mysql://10.30.68.82:3306/eggroll_meta?
useSSL=false&serverTimezone=UTC&characterEncoding=utf8&allowPublicKeyRetrieval=true
eggroll.resourcemanager.clustermanager.jdbc.username=fate
eggroll.resourcemanager.clustermanager.jdbc.password=fate_dev
eggroll.data.dir=data/
eggroll.logs.dir=logs/
#clustermanager & nodemanager
eggroll.resourcemanager.clustermanager.host=10.30.68.82
eggroll.resourcemanager.clustermanager.port=4670
eggroll.resourcemanager.nodemanager.port=4671
eggroll.resourcemanager.process.tag=fate-guest
eggroll.bootstrap.root.script=bin/eggroll_boot.sh
eggroll.resourcemanager.bootstrap.egg_pair.exepath=bin/roll_pair/egg_pair_bootstrap.sh
#python env
```

```
eggroll.resourcemanager.bootstrap.egg_pair.venv=/data/projects/fate/common/python/venv
#pythonpath, very import, do not modify.
eggroll.resourcemanager.bootstrap.egg_pair.pythonpath=/data/projects/fate/python:/data/projects/fate/eggroll/python
eggroll.resourcemanager.bootstrap.egg_pair.filepath=python/eggroll/roll_pair/egg_pair.py
eggroll.resourcemanager.bootstrap.roll_pair_master.exeopath=bin/roll_pair/roll_pair_master_bootstrap.sh
#javahome
eggroll.resourcemanager.bootstrap.roll_pair_master.javahome=/data/projects/fate/common/jdk/jdk-8u192
eggroll.resourcemanager.bootstrap.roll_pair_master.classpath=conf:/lib/*
eggroll.resourcemanager.bootstrap.roll_pair_master.mainclass=com.webank.eggroll.rollpair.RollPairMasterBootstrap
eggroll.resourcemanager.bootstrap.roll_pair_master.jvm.options=
# for roll site. rename in the next round
eggroll.rollsite.coordinator=webank
eggroll.rollsite.host=10.30.68.82
eggroll.rollsite.port=9370
eggroll.rollsite.party.id=9999
eggroll.rollsite.route.table.path=conf/route_table.json

eggroll.session.processors.per.node=4
eggroll.session.start.timeout.ms=180000
eggroll.rollsite.adapter.sendbuf.size=1048576
eggroll.rollpair.transferpair.sendbuf.size=4150000
EOF
```

6.3 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：

令完成：

配置文件: /data/projects/fate/eggroll/conf/route_table.json

```
#在目标服务器（10.30.68.84）fl 用户下修改执行
cat > /data/projects/fate/eggroll/conf/route_table.json << EOF
{
  "route_table":
  {
    "10000":
```

```

{
  "default":[
    {
      "port": 9370,
      "ip": "10.30.68.84"
    }
  ],
  "fateflow":[
    {
      "port": 9360,
      "ip": "10.30.68.83"
    }
  ]
},
"default":
{
  "default":[
    {
      "port": 9370,
      "ip": "10.30.68.82"
    }
  ]
}
},
"permission":
{
  "default_allow": true
}
}
EOF

```

#在目标服务器（10.30.68.82） fl 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"
  }
],
"default":
{
  "default":[
    {
      "port": 9370,
      "ip": "10.30.68.84"
    }
  ]
}
},
"permission":
{
  "default_allow": true
}
}
EOF
```

6.4 依赖服务配置文件修改

- fateflow

fateflow IP , host: 10.30.68.83, guest: 10.30.68.82

grpc 端口: 9360

http 端口: 9380

- fateboard

fateboard IP, host: 10.30.68.83, guest: 10.30.68.82

fateboard 端口: 8080

- proxy

proxy IP, host: 10.30.68.84, guest: 10.30.68.82---rollsite 组件对应 IP

proxy 端口：9370

此文件要按照 json 格式进行配置，不然会报错，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/python/arch/conf/server_conf.json

```
#在目标服务器（10.30.68.83 10.30.68.84） fl 用户下修改执行
cat > /data/projects/fate/python/arch/conf/server_conf.json << EOF
```

```
{
  "servers": {
    "fateflow": {
      "host": "10.30.68.83",
      "grpc.port": 9360,
      "http.port": 9380
    },
    "fateboard": {
      "host": "10.30.68.83",
      "port": 8080
    },
    "proxy": {
      "host": "10.30.68.84",
      "port": 9370
    },
    "servings": [
      "127.0.0.1:8000"
    ]
  }
}
EOF
```

```
#在目标服务器（10.30.68.82） fl 用户下修改执行
cat > /data/projects/fate/python/arch/conf/server_conf.json << EOF
```

```
{
  "servers": {
    "fateflow": {
      "host": "10.30.68.82",
      "grpc.port": 9360,
      "http.port": 9380
    },
    "fateboard": {
      "host": "10.30.68.82",
      "port": 8080
    }
  }
}
```

```

    },
    "proxy": {
      "host": "10.30.68.82",
      "port": 9370
    },
    "servings": [
      "127.0.0.1:8000"
    ]
  }
}
EOF

```

6.5 数据库信息配置文件修改

- work_mode(为 1 表示集群模式，默认)
- db 的连接 ip、端口、账号和密码
- redis IP、端口、密码（redis 暂使用不需要配置）

此配置文件格式要按照 yaml 格式配置，不然解析报错，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/python/arch/conf/base_conf.yaml

```

#在目标服务器（10.30.68.83）fl 用户下修改执行
cat > /data/projects/fate/python/arch/conf/base_conf.yaml <<EOF
work_mode: 1
fate_flow:
  host: 0.0.0.0
  http_port: 9380
  grpc_port: 9360
database:
  name: fate_flow
  user: fate
  passwd: fate_dev
  host: 10.30.68.83
  port: 3306
  max_connections: 100
  stale_timeout: 30
redis:
  host: 127.0.0.1

```

```
port: 6379
password: WEBANK_2014_fate_dev
max_connections: 500
db: 0
default_model_store_address:
  storage: redis
  host: 127.0.0.1
  port: 6379
  password: fate_dev
  db: 0
EOF

#在目标服务器（10.30.68.82）fl 用户下修改执行
cat > /data/projects/fate/python/arch/conf/base_conf.yaml <<EOF
work_mode: 1
fate_flow:
  host: 0.0.0.0
  http_port: 9380
  grpc_port: 9360
database:
  name: fate_flow
  user: fate
  passwd: fate_dev
  host: 10.30.68.82
  port: 3306
  max_connections: 100
  stale_timeout: 30
redis:
  host: 127.0.0.1
  port: 6379
  password: WEBANK_2014_fate_dev
  max_connections: 500
  db: 0
default_model_store_address:
  storage: redis
  host: 127.0.0.1
  port: 6379
  password: fate_dev
  db: 0
EOF
```


6.6 配置文件修改

1) application.properties

- 服务端口

server.port---默认

- fateflow 的访问 url

fateflow.url, host: <http://10.30.68.83:9380>, guest: <http://10.30.68.82:9380>

- 数据库连接串、账号和密码

fateboard.datasource.jdbc-url, host: mysql://10.30.68.83:3306, guest:

mysql://10.30.68.82:3306

fateboard.datasource.username: fate

fateboard.datasource.password: fate_dev

以上参数调整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/fateboard/conf/application.properties

```
#在目标服务器（10.30.68.83）fi 用户下修改执行
cat > /data/projects/fate/fateboard/conf/application.properties <<EOF
server.port=8080
fateflow.url=http://10.30.68.83:9380
spring.datasource.driver-Class-Name=com.mysql.cj.jdbc.Driver
spring.http.encoding.charset=UTF-8
spring.http.encoding.enabled=true
server.tomcat.uri-encoding=UTF-8
```

```

fateboard.datasource.jdbc-
url=jdbc:mysql://10.30.68.83:3306/fate_flow?characterEncoding=utf8&characterSetResults=utf8
&autoReconnect=true&failOverReadOnly=false&serverTimezone=GMT%2B8
fateboard.datasource.username=fate
fateboard.datasource.password=fate_dev
server.tomcat.max-threads=1000
server.tomcat.max-connections=20000
EOF

#在目标服务器（10.30.68.82）fl 用户下修改执行
cat > /data/projects/fate/fateboard/conf/application.properties <<EOF
server.port=8080
fateflow.url=http://10.30.68.82:9380
spring.datasource.driver-Class-Name=com.mysql.cj.jdbc.Driver
spring.http.encoding.charset=UTF-8
spring.http.encoding.enabled=true
server.tomcat.uri-encoding=UTF-8
fateboard.datasource.jdbc-
url=jdbc:mysql://10.30.68.82:3306/fate_flow?characterEncoding=utf8&characterSetResults=utf8
&autoReconnect=true&failOverReadOnly=false&serverTimezone=GMT%2B8
fateboard.datasource.username=fate
fateboard.datasource.password=fate_dev
server.tomcat.max-threads=1000
server.tomcat.max-connections=20000
EOF

```

2) service.sh

```

#在目标服务器（10.30.68.83 10.30.68.82）fl 用户下修改执行
cd /data/projects/fate/fateboard
vi service.sh
export JAVA_HOME=/data/projects/fate/common/jdk/jdk-8u192

```

7. 启动服务

在目标服务器（10.30.68.84）fl 用户下执行

```

#启动 eggroll 服务
source /data/projects/fate/init_env.sh
cd /data/projects/fate/eggroll
sh ./bin/eggroll.sh rollsite start
sh ./bin/eggroll.sh nodemanager start

```

在目标服务器（10.30.68.83）fl 用户下执行

```
#启动 eggroll 服务
source /data/projects/fate/init_env.sh
cd /data/projects/fate/eggroll
sh ./bin/eggroll.sh clustermanager start
sh ./bin/eggroll.sh nodemanager start

#启动 fate 服务，fateflow 依赖 rollsite 和 mysql 的启动，等所有节点的 eggroll 都启动后再启动
fateflow，
否则会卡死报错
source /data/projects/fate/init_env.sh
cd /data/projects/fate/python/fate_flow
sh service.sh start
cd /data/projects/fate/fateboard
sh service.sh start(在 service.sh 中添加 JAVA_HOME: /data/projects/fate/common/jdk/jdk-8u192)
```

在目标服务器 (10.30.68.82) fl 用户下执行

```
#启动 eggroll 服务
source /data/projects/fate/init_env.sh
cd /data/projects/fate/eggroll
sh ./bin/eggroll.sh all start

#启动 fate 服务
source /data/projects/fate/init_env.sh
cd /data/projects/fate/python/fate_flow
sh service.sh start
cd /data/projects/fate/fateboard
sh service.sh start(在 service.sh 中添加 JAVA_HOME: /data/projects/fate/common/jdk/jdk-8u192)
```

8. 问题定位

1) eggroll 日志

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

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

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

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

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

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

2) fateflow 日志

/data/projects/fate/python/logs/fate_flow/

3) fateboard 日志

/data/projects/fate/fateboard/logs

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/init_env.sh
```

```
cd /data/projects/fate/python/examples/toy_example/
```

```
python run_toy_example.py 10000 10000 1
```

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/init_env.sh
```

```
cd /data/projects/fate/python/examples/toy_example/
```

```
python run_toy_example.py 9999 9999 1
```

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 1999.99999999999998"
```

1.2 双边测试

选定 9999 为 guest 方, 在 10.30.68.82 上执行:

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

```
cd /data/projects/fate/python/examples/toy_example/
```

```
python run_toy_example.py 9999 10000 1
```

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 1999.99999999999998"
```

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 上执行:

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

```
cd /data/projects/fate/python/examples/scripts/
```

```
python upload_default_data.py -m 1
```

更多细节信息，敬请参考[脚本 README](#)

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 上执行：

```
source /data/projects/fate/init_env.sh
cd /data/projects/fate/python/examples/min_test_task/
python run_task.py -m 1 -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，如果有防火墙需开通。如果 fateboard 和 fateflow 没有部署再同一台服务器，需在 fateboard 页面设置 fateflow 所部署主机的登陆信息：页面右上侧齿轮按钮--> add--> 填写 fateflow 主机 ip, os 用户, ssh 端口, 密码。

第七部分：系统运维

1. 服务管理

1.1 Eggroll 服务管理

```
source /data/projects/fate/init_env.sh
cd /data/projects/fate/eggroll
```

启动/关闭/查看/重启所有：

```
sh ./bin/eggroll.sh all start/stop/status/restart
```

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

```
sh ./bin/eggroll.sh clustermanager start/stop/status/restart
```

1.2 Fate 服务管理

1. 启动/关闭/查看/重启 fate_flow 服务

```
source /data/projects/fate/init_env.sh
cd /data/projects/fate/python/fate_flow
sh service.sh start|stop|status|restart
```

如果逐个模块启动，需要先启动 eggroll 再启动 fateflow，fateflow 依赖 eggroll 的启动。

2. 启动/关闭/重启 fateboard 服务

```
cd /data/projects/fate/fateboard
sh service.sh start|stop|status|restart
```

1.3 Mysql 服务管理

启动/关闭/查看/重启 mysql 服务

```
cd /data/projects/fate/common/mysql/mysql-8.0.13
sh ./service.sh start|stop|status|restart
```

2. 查看进程和端口

在目标服务器（10.30.68.83 10.30.68.84 10.30.68.82） fl 用户下执行

2.1 查看进程

```
#根据部署规划查看进程是否启动
ps -ef | grep -i clustermanager
ps -ef | grep -i nodemanager
ps -ef | grep -i rollsite
ps -ef | grep -i fate_flow_server.py
ps -ef | grep -i fateboard
```

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
```

3. 服务日志

服务	日志路径
eggroll	/data/projects/fate/eggroll/logs
fate_flow& 任务日志	/data/projects/fate/python/logs
fateboard	/data/projects/fate/fateboard/logs
mysql	/data/projects/fate/common/mysql/mysql- 8.0.13/logs

第八部分：附录

1. eggroll&fate 打包构建

参见 [build 指导](#)

2. 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:

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

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