大作业项目：基于区块链的供应链金融平台前期报告

姓名	年级专业	学号
冼子婷	19级软件工程	
廖雨轩	19级软件工程	
胡文浩	19级软件工程	18346019

实验环境

实验中使用到的是华为云的云服务器，具体配置如下。具体的 FISCO-BCOS 部署参照 官方文档 给出的部署流程即可

依赖安装

首先执行 `sudo apt install -y openssl curl` 获取所需要的依赖文件，然后执行一下流程：

```
## 创建操作目录
cd /home && mkdir -p fisco && cd fisco

## 下载脚本
curl -#LO https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/v2.8.0/build_chain.sh && chmod u+x build_chain.sh
```

```
root@hecs-268613:/home# ll
total 28
drwxr-xr-x  7 root root 4096 Dec 30 21:40 ./
drwxr-xr-x 25 root root 4096 Dec 29 16:46 ../
drwxr-xr-x  4 root root 4096 Dec 30 21:40 data/
drwxr-xr-x  4 root root 4096 Dec 23 16:19 DownloadPackage/
drwxr-xr-x  4 root root 4096 Dec 17 21:11 fisco/
drwxr-xr-x  3 root root 4096 Dec 30 21:13 goPro/
drwxr-xr-x 19 root root 4096 Dec 26 00:22 go-sdk/
root@hecs-268613:/home#
```

```
root@hecs-268613:/home/fisco# curl -#LO https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/v2.8.0/build_chain.sh
 && chmod u+x build_chain.sh
######################################################################################################################## 100.0%
######################################################################################################################## 100.0%
root@hecs-268613:/home/fisco# ll
total 42248
drwxr-xr-x 4 root root      4096 Dec 17 21:11 ./
drwxr-xr-x 7 root root      4096 Dec 30 21:40 ../
-rwxr--r-- 1 root root     67645 Dec 30 21:44 build_chain.sh*
drwxr-xr-x 9  501 staff     4096 Dec 24 19:11 console/
-rw-r--r-- 1 root root  43168632 Dec 17 16:55 console.tar.gz
-rw-r--r-- 1 root root      3659 Dec 17 16:33 download_console.sh
drwxr-xr-x 4 root root      4096 Dec 17 16:28 nodes/
root@hecs-268613:/home/fisco#
```

搭建 4 节点联盟链

节点配置生成

由于此报告在实验完成后补做，已经部署的联盟链中存在一些重要信息不便删除，因此使用另外一个目录 `/home/fiscoTemp` 进行节点的创建演示。执行以下命令创建节点信息，并获得如图的结果

```
1  bash build_chain.sh -l 127.0.0.1:4 -p 30300,20200,8545
```

```
root@hecs-268613:/home/fiscoTemp# bash build_chain.sh -l 127.0.0.1:4 -p 30300,20200,8545
[INFO] Downloading fisco-bcos binary from https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/v2.8.0/fisco-bcos.t
ar.gz ...
######################################################################################################################## 100.0%
###                                                                                                                        3.4%
curl: (28) Operation too slow. Less than 102400 bytes/sec transferred the last 20 seconds
[INFO] Download speed is too low, try https://osp-1257653870.cos.ap-guangzhou.myqcloud.com/FISCO-BCOS/FISCO-BCOS/release
s/v2.8.0/fisco-bcos.tar.gz
######################################################################################################################## 100.0%
============================================================
Generating CA key...
============================================================
Generating keys and certificates ...
Processing IP=127.0.0.1 Total=4 Agency=agency Groups=1
============================================================
Generating configuration files ...
Processing IP=127.0.0.1 Total=4 Agency=agency Groups=1
============================================================
[INFO] Start Port      : 30300 20200 8545
[INFO] Server IP       : 127.0.0.1:4
[INFO] Output Dir      : /home/fiscoTemp/nodes
[INFO] CA Path         : /home/fiscoTemp/nodes/cert/
============================================================
[INFO] Execute the download_console.sh script in directory named by IP to get FISCO-BCOS console.
e.g.  bash /home/fiscoTemp/nodes/127.0.0.1/download_console.sh -f
============================================================
[INFO] All completed. Files in /home/fiscoTemp/nodes
root@hecs-268613:/home/fiscoTemp#
```

启动节点

配置生成成功后即可启动节点，由于使用的已经部署好的节点，因此先将节点停止，再启动节点。

```
root@hecs-268613:/home/fisco# ./nodes/127.0.0.1/stop_all.sh
try to stop node0
try to stop node1
try to stop node2
try to stop node3
 stop node0 success.
 stop node1 success.
 stop node3 success.
 stop node2 success.
root@hecs-268613:/home/fisco# ./nodes/127.0.0.1/start_all.sh
try to start node0
try to start node1
try to start node2
try to start node3
 node3 start successfully
 node2 start successfully
 node1 start successfully
 node0 start successfully
root@hecs-268613:/home/fisco#
```

使用 `ps -ef | grep -v grep | grep fisco` 验证节点是否成功启动

```
root@hecs-268613:/home/fisco# ps -ef | grep -v grep | grep fisco-bcos
root      9249     1  0 21:56 pts/1    00:00:00 /home/fisco/nodes/127.0.0.1/node3/../fisco-bcos -c config.ini
root      9251     1  0 21:56 pts/1    00:00:00 /home/fisco/nodes/127.0.0.1/node2/../fisco-bcos -c config.ini
root      9253     1  0 21:56 pts/1    00:00:00 /home/fisco/nodes/127.0.0.1/node1/../fisco-bcos -c config.ini
root      9255     1  0 21:56 pts/1    00:00:00 /home/fisco/nodes/127.0.0.1/node0/../fisco-bcos -c config.ini
root@hecs-268613:/home/fisco#
```

检查日志输出

使用 `tail -f nodes/127.0.0.1/node0/log/log*  | grep connected` 可看见日志不断输出

```
info|2021-12-29 14:59:57.746749|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 15:59:57.771042|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 16:59:57.506952|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 17:59:57.544667|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 18:59:57.570027|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 19:59:57.595472|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 20:59:57.618987|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 21:59:57.644476|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 22:59:57.670267|[P2P][Service] heartBeat,connected count=3
info|2021-12-29 23:59:57.697684|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 00:59:57.720355|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 01:59:57.744007|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 02:59:57.767741|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 03:59:57.791645|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 04:59:57.814270|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 05:59:57.837776|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 06:59:57.863148|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 07:59:57.887486|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 08:59:57.910502|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 09:59:57.935778|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 10:59:57.959976|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 11:59:57.984132|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 12:59:58.011644|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 13:59:58.036493|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 14:59:58.060723|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 15:59:58.084078|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 16:59:58.109361|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 17:59:58.137982|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 18:59:58.164356|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 19:59:58.189380|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 20:59:58.213899|[P2P][Service] heartBeat,connected count=3
info|2021-12-30 21:58:33.540120|[P2P][Service] heartBeat,connected count=3
```

使用 `tail -f nodes/127.0.0.1/node0/log/log*  | grep +++` 检查节点是否正在共识

```
info|2021-12-30 00:59:59.318016|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=a
a4c492d...
info|2021-12-30 01:59:59.910055|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=5
d9ee63e...
info|2021-12-30 02:59:59.918715|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=b
4ab46d6...
info|2021-12-30 05:59:57.468336|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=1
3243201...
info|2021-12-30 06:59:58.455977|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=b
22c8e36...
info|2021-12-30 07:59:58.693947|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=a
4803944...
info|2021-12-30 08:59:59.922725|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=6
6f735ff...
info|2021-12-30 10:59:57.791309|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=4
e5d4854...
info|2021-12-30 11:59:59.109900|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=d
558b06c...
info|2021-12-30 14:59:58.256499|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=3
683f2ea...
info|2021-12-30 15:59:58.333154|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=a
84f7327...
info|2021-12-30 16:59:59.082439|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=720,tx=0,nodeIdx=2,hash=c
c5e15a3...
info|2021-12-30 17:59:58.415151|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=754,tx=0,nodeIdx=2,hash=a
918aa60...
info|2021-12-30 19:59:57.652395|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=754,tx=0,nodeIdx=2,hash=d
3431aca...
info|2021-12-30 20:59:58.984865|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=754,tx=0,nodeIdx=2,hash=3
60d4701...
info|2021-12-30 21:59:11.131366|[g:1][CONSENSUS][SEALER]++++++++++++++++ Generating seal on,blkNum=754,tx=0,nodeIdx=2,hash=d
d38dda1...
```

控制台的使用

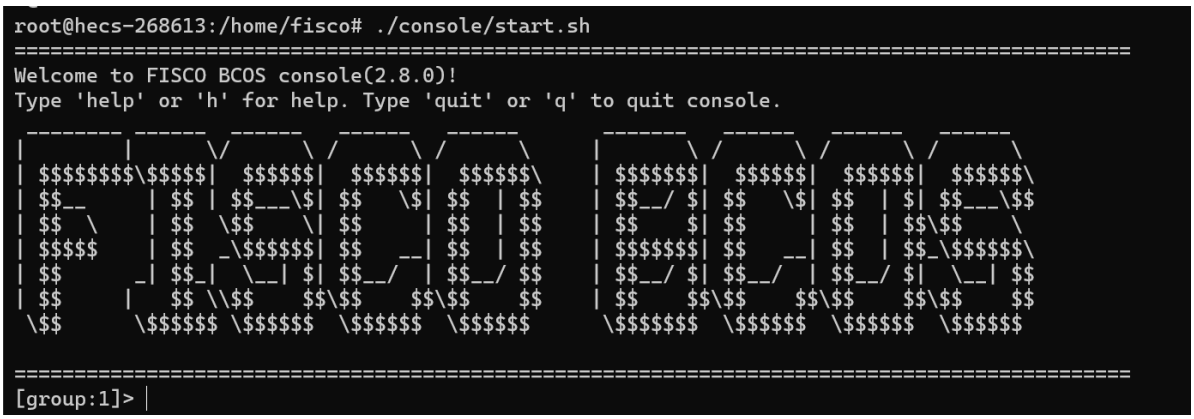
控制台环境

控制台的环境只需要简单的安装 java，然后安装按照以下命令即可配置完成

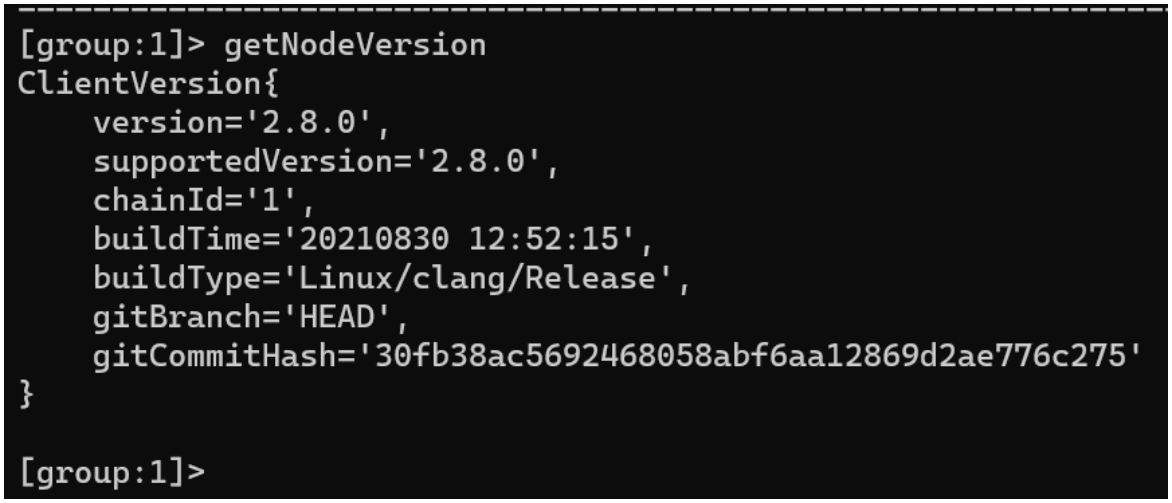
```
sudo apt install -y default-jdk

cd /home/fisco && curl -LO https://github.com/FISCO-BCOS/console/releases/download/v2.8.0/download_console.sh && bash download_console.sh

# 最新版本控制台使用如下命令拷贝配置文件
cp -n console/conf/config-example.toml console/conf/config.toml

cp -r nodes/127.0.0.1/sdk/* console/conf/
```

控制台启动和使用

然后执行 `./console/start.sh` 启动控制台



获取节点版本



获取 peers，从这个命令中可以得知控制台当前连接的是 node0

```
[group:1]> getPeers
[
    PeerInfo{
        nodeID='107a3cd4042264ac7d1e51bc9134c16bc64e8cd11b41917698ba579d4d13dfd61289ba0dbd68174a0480bbf289cff1a455a974f4
1e1a195e2e923954861886cd',
        iPAndPort='127.0.0.1:30302',
        node='node2',
        agency='agency',
        topic='[

        ]'
    },
    PeerInfo{
        nodeID='7bc8f08fdfebe142bf6fde6e3b1aa834a1b7b49a7bba43eb7ef3a5c929eeafcf77291e4f048bd157f989a3a19276914f4b1d0d40
ba95d05965de0f33b6c9f305',
        iPAndPort='127.0.0.1:30303',
        node='node3',
        agency='agency',
        topic='[

        ]'
    },
    PeerInfo{
        nodeID='05c8c60e09f0c9449340c2814395abc3364c901e70859aa0477ea2f75608f488ee35ab518f720c1d04bf54582cc7648659169cff
7cf1fb39a950edc41bfb9ad9',
        iPAndPort='127.0.0.1:30301',
        node='node1',
        agency='agency',
        topic='[
            _block_notify_1
        ]'
    }
]

[group:1]>
```

合约的部署与调用

合约部署

合约的部署比较简单，通过 `deploy` 命令即可找到在 fisco bcos 目录下 `console/contracts/solidity/` 存在的合约，只需要选择需要的合约名称，然后对应给出合约 `constructer()` 方法所需要的参数即可，这里演示我们所编写的合约的部署：

通过输入 `deploy SupplyChain 0x54e42c4648a236e9557a0f30484155d7b2cf94b1 test90` 即可完成合约的部署，其中 deploy 的第一个参数为具体合约的名称，后面两个参数则对应于我们所编写的合约中系统的管理员地址以及对应 Table.sol 的表名后缀（后缀是为了每次部署调试的时候能够使用全新的表，不和之前的冲突）

`transaction hash` 表示的是在区块链上部署合约这个操作所执行的交易地址，`currentAccount` 则为连上控制台的账号。最重要的是 `contract address`，合约地址是用户调用对应智能合约的地址。

```
[group:1]> deploy SupplyChain 0x54e42c4648a236e9557a0f30484155d7b2cf94b1 test90
transaction hash: 0x9eda765be56bcf8aa6cfae43e5ecf8f72b7c089ede18abc4ae96e3a0a55153ca
contract address: 0x8e73eca132c10a45541e6ad08783a03caa6ad500
currentAccount: 0xe150a93b790ce8495306a8589e896dce1f91dc81

[group:1]>
```

合约调用

控制台中可以使用 `call` 对合约进行调用，`call` 的第一个参数为需要调用的合约名称，第二参数则是合约具体的地址（同一份合约可以多次部署，每次部署都能得到不同的地址，调用时使用对应地址才能调用对应合约），第三个参数以及后续参数则分别为调用合约中的具体函数名以及该函数所需要的参数。

```
[group:1]> call SupplyChain 0x8e73eca132c10a45541e6ad08783a03caa6ad500 getAdmin
-------------------------------------------------------------------------------------------
Return code: 0
description: transaction executed successfully
Return message: Success
-------------------------------------------------------------------------------------------
Return value size:1
Return types: ([ADDRESS, UINT] )
Return values:([0x54e42c4648a236e9557a0f30484155d7b2cf94b1, 0] )
-------------------------------------------------------------------------------------------

[group:1]>
```

查看区块

使用 `getBlockByNumber` 命令进行区块的获取，可以得到以下结果：

```
[group:1]> getBlockByNumber 1
{
    transactions=[
        TransactionHash{
            value='0x7d019a09b655a532c598b8ee0e250c71063a56df1ccd4f1c07f7afd423343363'
        }
    ],
    number='0x1',
    hash='0xae2d9e448677320dad43563f6072f7e62199459939dcdaaa280d44dc50cff8f1',
    parentHash='0x5d4b1db5e0c217b4e68b8c600fd8eaa62ea4737dd9a33e42f7faa7316d2aaaeb',
    logsBloom='0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000',
    transactionsRoot='0x87d1af00e984e8508181f7040c084a0c974abeab72cbb2ef146288e5b379f88e',
    receiptsRoot='0x3f4afc77a829d34829971bfe9236c6228eb352dd5a69af5d43735a499f7d0db6',
    dbHash='0x06b991d10c6a80dd5d39186b893ad3687112838358cfdc9e8e7b975e0fbcf8ea',
    stateRoot='0x06b991d10c6a80dd5d39186b893ad3687112838358cfdc9e8e7b975e0fbcf8ea',
    sealer='0x3',
    sealerList=[
        05c8c60e09f0c9449340c2814395abc3364c901e70859aa0477ea2f75608f488ee35ab518f720c1d04bf54582cc7648659169cff7cf1fb39
a950edc41bfb9ad9,
        107a3cd4042264ac7d1e51bc9134c16bc64e8cd11b41917698ba579d4d13dfd61289ba0dbd68174a0480bbf289cff1a455a974f41e1a195e
2e923954861886cd,
        672bd545a734972b6eaec99c9e8bb31c8ced49179cc2ac73354f0d1761ef9825f002d9adb31968f0ba98ef7b526a6cd03719dcc4c2a844fe
e9ea3e312f6355d0,
        7bc8f08fdfebe142bf6fde6e3b1aa834a1b7b49a7bba43eb7ef3a5c929eeafcf77291e4f048bd157f989a3a19276914f4b1d0d40ba95d059
65de0f33b6c9f305
    ],
    extraData=[

    ],
    gasLimit='0x0',
    gasUsed='0x0',
    timestamp='0x17dc7d7851a',
    signatureList=null
}

[group:1]>
```

字段	类型	作用	RLP index
transactions	vector	包含区块中所有交易的 Hash	-
number	int64_t	本区块的块号，块号从0号开始计算	6
hash	h256	区块头前13个字段RLP编码后的哈希值，FISCO BCOS新增字段	-
parentHash	h256	父区块的 hash	0
logsBloom	LogBloom	区块中的 logs 布隆过滤器，用于快速筛选某个日志一定不存在或者可能存在在区块中	5
transactionRoot	h256	交易树根，类似于比特币中的 Merkle root，可以向轻节点提供 Merkle Proof	2
receiptsRoot	h256	收据树根。收据树上的节点和交易树的节点一一对应，每个收据具体记录对应交易的相关信息，收据树根可用于验证和防篡改	3
stateRoot	h256	状态树根。以太坊是基于账户的账本，需要对账户地址和账户状态进行映射，同理与交易树一一对应，也可以用于验证和防篡改	1
dbHash	h256	分布式存储通过计算哈希值来记录一区块中写入的数据，FISCO BCOS新增字段	4
sealer	u256	打包区块的节点在共识节点列表中的索引，FISCO BCOS新增字段	11
sealerList	vector	区块的共识节点列表（不含观察节点），FISCO BCOS新增字段	12
extraData	vector<byte>	预留字段，记录交易信息，内部使用“#”分割信息	-
gasLimit	u256	本区块中所有交易消耗的Gas上限	7
gasUsed	u256	本区块中所有交易消耗的Gas之和	8
timestamp	int64_t	打包区块的 unix 时间戳	9
signatureList	vector	PBFT 共识的签名列表	-