東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

区块链技术与应用

v0.11.5

实验四：共识（上）

丁烨，网络空间安全学院 副教授
dingye@dgut.edu.cn

分布式账本回顾

分布式账本工作流程

1. 本地创建交易记录（Transaction）
2. 传送给分布式账本，存放于记录池（Pool）
3. 分布式账本节点之间通过共识不断同步记录池
4. 不断尝试让记录池满足一定条件（例如，完成写入证明）
5. 校验记录池中的交易记录并生成区块（Block）
6. 加入新的区块并生成新的区块链
7. 分布式账本节点之间通过共识不断同步区块链

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持区块链的记录池

- app.py

```
from ..blockchain.ledger import Ledger
from ..blockchain.transaction import Transaction
```

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持区块链的记录池

- app.py

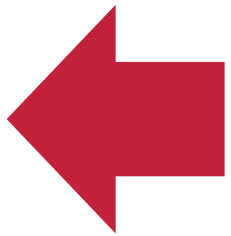
```
def on_message(self, message):
    ...
    elif message['op'] == 'pool':
        self.write_message(json.dumps({
            'status': 200,
            'error': 'OK',
            'response': {
                'pool': [json.loads(str(i)) for i in list(pickle.loads(db.get('pool')))]
            }
        }))
```

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持区块链的记录池

- app.py

```
elif message['op'] == 'merge':
    if 'args' in message and 'pool' in message['args']:
        pool = pickle.loads(db.get('pool'))
        for i in message['args']['pool']:
            pool.add(Transaction(
                sender=i['sender'],
                receiver=i['receiver'],
                amount=i['amount']
            ))
        db.set('pool', pickle.dumps(pool))
```



实现支持共识的分布式账本

修改 P2P 网络服务并使其支持区块链的记录池

- app.py

```
elif message['op'] == 'merge':
    if 'args' in message and 'pool' in message['args']:
        ...
        self.write_message(json.dumps({
            'status': 202,
            'error': 'Accepted'
        }))
```

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持区块链的记录池

- app.py

```
elif message['op'] == 'merge':
    if 'args' in message and 'pool' in message['args']:
        ...
    else:
        self.write_message(json.dumps({
            'status': 500,
            'error': 'Operation "merge" requires the following "args": "pool"',
            'response': None
        }))
```

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持区块链的记录池

- app.py

```
if __name__ == '__main__':
    ...
    db = Redis(args.redis)
    if b'peers' not in db.keys():
        db.set('peers', pickle.dumps(set([])))
    if b'pool' not in db.keys():
        db.set('pool', pickle.dumps(set([])))
    ...
```

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持区块链的记录池

```
~/Workspace/foxchain(master*) » python3 -u -m foxchain.app.app
[2021-04-26 23:36:05,378] Tornado is listening on port: 9000
```

HISTORY

Clear

- Request #36
- Request #35
- Request #34
- Request #33
- Request #32
- Request #31
- Request #30
- Request #29

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "pool"
}
```

```
{
  "status": 200,
  "error": "OK",
  "response": {
    "pool": []
  }
}
```

HISTORY

Clear

- Request #37
- Request #36
- Request #35
- Request #34
- Request #33
- Request #32
- Request #31
- Request #30
- Request #29

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "merge",
  "args": {
    "pool": [{
      "sender": "00000000-0000-0000-0000-000000000000",
      "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf2",
      "amount": 10000
    }]
  }
}
```

```
{
  "status": 202,
  "error": "Accepted"
}
```

HISTORY

Clear

- Request #38
- Request #37
- Request #36
- Request #35
- Request #34
- Request #33
- Request #32
- Request #31
- Request #30
- Request #29

Everything is ok, ready to go!

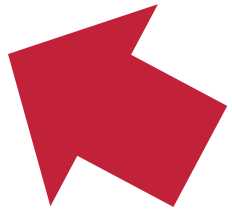
ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "pool"
}
```

```
{
  "status": 200,
  "error": "OK",
  "response": {
    "pool": [{
      "sender": "00000000-0000-0000-0000-000000000000",
      "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf2",
      "amount": 10000,
      "t": 1619451754.012226,
      "prev_hash": null
    }]
  }
}
```



实现支持共识的分布式账本

生成区块链

- builder.py

```
import argparse
import logging
import pickle
import time
import traceback

from redislite import Redis
from wrenchbox.logging import setup_log

from ..blockchain.blockchain import Block
from ..blockchain.ledger import GENESIS
```

实现支持共识的分布式账本

生成区块链

- builder.py

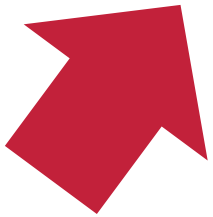
```
class Builder:
    def __init__(self, url):
        self.db = Redis(url)
```

实现支持共识的分布式账本

生成区块链

- builder.py

```
def run(self, k_size, cool_down):
    while True:
        pool = pickle.loads(self.db.get('pool'))
        if len(pool) >= k_size:
            self.db.set('pool', pickle.dumps(set([])))
            logging.info('Pool is cleared.')
            logging.info('Packing %d transactions...', len(pool))
            ...
        else:
            logging.debug('Currently %d records, require: %d', len(pool), k_size)
        time.sleep(cool_down)
```



实现支持共识的分布式账本

生成区块链

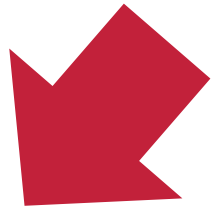
- builder.py

```
ledger = pickle.loads(self.db.get('ledger'))
block = Block()
for transaction in pool:
    if transaction.sender == GENESIS \
            or ledger.balance(transaction.sender) >= transaction.amount:
        block.add(transaction)
    else:
        logging.warning('Dropped transaction due to not enough balance: %s', transaction)
logging.info('Block is created with %d records.', len(block.items))
```

实现支持共识的分布式账本

生成区块链

- builder.py



```
if len(block.items):
    try:
        block.validate()
    except AssertionError:
        logging.error('Block is invalid and dropped.')
        if args.debug:
            traceback.print_exc()
    else:
        ...
```

实现支持共识的分布式账本

生成区块链

- builder.py

```
ledger.add(block)
try:
    ledger.validate()
except AssertionError:
    logging.error('Ledger is invalid and dropped.')
    if args.debug:
        traceback.print_exc()
else:
    self.db.set('ledger', pickle.dumps(ledger))
    logging.info('Block is added to the blockchain.')
```

实现支持共识的分布式账本

生成区块链

- builder.py

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument(
        '--debug',
        action='store_true', default=False,
        help='show debug information'
    )
    ...
    args, _ = parser.parse_known_args()
    setup_log(level=logging.DEBUG if args.debug else logging.INFO)
    Builder(args.redis).run(args.size, args.sleep)
```

实现支持共识的分布式账本

生成区块链

- builder.py

```
parser.add_argument(
    '-r', '--redis',
    type=str, default='redis.db',
    help='redis database file, default: redis.db'
)
```

实现支持共识的分布式账本

生成区块链

- builder.py

```
parser.add_argument(
    '-k', '--size',
    type=int, default=3,
    help='# of minimum packed transactions, default: 3'
)
```

实现支持共识的分布式账本

生成区块链

- builder.py

```
parser.add_argument(
    '-t', '--sleep',
    type=int, default=3,
    help='refresh rate in seconds, default: 3'
)
```

实现支持共识的分布式账本

生成区块链

```
~/Workspace/foxchain(master*) » python3 -u -m foxchain.app.builder
```

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持查询区块链

- app.py

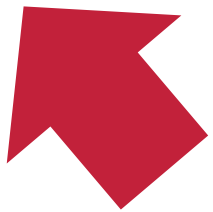
```
elif message['op'] == 'blocks':
    blocks = json.loads(str(pickle.loads(db.get('ledger'))))['blocks']
    if 'args' in message and 'start' in message['args']:
        blocks = blocks[int(message['args']['start']):]
    self.write_message(json.dumps({
        'status': 200,
        'error': 'OK',
        'response': {
            'blocks': blocks
        }
    }))
```

实现支持共识的分布式账本

修改 P2P 网络服务并使其支持查询区块链

- app.py

```
if __name__ == '__main__':
    ...
    db = Redis(args.redis)
    if b'peers' not in db.keys():
        db.set('peers', pickle.dumps(set([])))
    if b'pool' not in db.keys():
        db.set('pool', pickle.dumps(set([])))
    if b'ledger' not in db.keys():
        db.set('ledger', pickle.dumps(Ledger()))
    ...
```



实现支持共识的分布式账本

修改 P2P 网络服务并使其支持查询区块链

```
~/Workspace/foxchain(master*) » python3 -u -m foxchain.app.app
[2021-04-26 23:36:05,378] Tornado is listening on port: 9000
```

HISTORY

Clear

Request #51

Request #50

Request #49

Request #48

Request #47

Request #46

Request #45

Request #44

Request #43

Request #42

Request #41

Request #40

Request #39

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "blocks"
}
```

```
{
  "status": 200,
  "error": "OK",
  "response": {
    "blocks": []
  }
}
```

HISTORY

Clear

- Request #38
- Request #37
- Request #36
- Request #35
- Request #34
- Request #33
- Request #32
- Request #31
- Request #30
- Request #29

Everything is ok, ready to go!

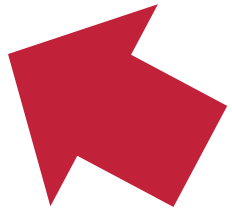
ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "pool"
}
```

```
{
  "status": 200,
  "error": "OK",
  "response": {
    "pool": [{
      "sender": "00000000-0000-0000-0000-000000000000",
      "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf2",
      "amount": 10000,
      "t": 1619451754.012226,
      "prev_hash": null
    }]
  }
}
```



HISTORY

Clear

- Request #39
- Request #38
- Request #37
- Request #36
- Request #35
- Request #34

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "merge",
  "args": {
    "pool": [{
      "sender": "00000000-0000-0000-0000-000000000000",
      "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf1",
      "amount": 8000
    }]
  }
}
```

```
{
  "status": 202,
  "error": "Accepted"
}
```

HISTORY

Clear

- Request #40
- Request #39
- Request #38
- Request #37
- Request #36
- Request #35
- Request #34

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "merge",
  "args": {
    "pool": [{
      "sender": "00000000-0000-0000-0000-000000000000",
      "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf0",
      "amount": 5000
    }]
  }
}
```

```
{
  "status": 202,
  "error": "Accepted"
}
```

HISTORY

Clear

- Request #48
- Request #47
- Request #46
- Request #45
- Request #44
- Request #43
- Request #42
- Request #41
- Request #40
- Request #39
- Request #38
- Request #37
- Request #36

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "pool"
}
```


```
{
  "status": 200,
  "error": "OK",
  "response": {
    "pool": [{
      "sender": "00000000-0000-0000-0000-000000000000",
      "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf2",
      "amount": 10000,
      "t": 1619452890.405376,
      "prev_hash": null
    }, {
      "sender": "00000000-0000-0000-0000-000000000000",
      "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf1",
      "amount": 8000,
      "t": 1619452895.812331,
```

HISTORY

Clear

Request #49

Request #48

Request #47

Request #46

Request #45

Request #44

Request #43

Request #42

Request #41

Request #40

Request #39

Request #38

Request #37

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "pool"
}
```

```
{
  "status": 200,
  "error": "OK",
  "response": {
    "pool": []
  }
}
```

HISTORY

Clear

- Request #50
- Request #49
- Request #48
- Request #47
- Request #46
- Request #45
- Request #44
- Request #43
- Request #42
- Request #41
- Request #40
- Request #39
- Request #38

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
{
  "op": "blocks"
}
```


```
{
  "status": 200,
  "error": "OK",
  "response": {
    "blocks": [{
      "items": [{
        "sender": "00000000-0000-0000-0000-000000000000",
        "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf1",
        "amount": 8000,
        "t": 1619452895.812331,
        "prev_hash": null
      }, {
        "sender": "00000000-0000-0000-0000-000000000000",
        "receiver": "bc70b89d-4521-454c-bebc-6c2456074bf2",
        "amount": 10000,
```

实现支持共识的分布式账本

生成区块链

```
~/Workspace/foxchain(master*) » python3 -u -m foxchain.app.builder
[2021-04-27 00:06:25,074] Pool is cleared.
[2021-04-27 00:06:25,075] Packing 3 transactions...
[2021-04-27 00:06:25,075] Block is created with 3 records.
[2021-04-27 00:06:25,075] Block is added to the blockchain.
```

作业

- 参考实验教程，实现一个支持共识的分布式账本
- 修改“写入证明”共识，将“Pool 中记录数量 ≥ 3”修改为：
- “Pool 中最早记录和最新记录的时间跨度 ≥ 3 分钟”

- 提供完整的测试代码及截图

作业

- 在作业系统中下载并完成本实验课对应实验报告
- https://hw.css.dgut.edu.cn/
- 注意：所有标识为 * 的地方都需要填写
- 请务必在截止时间之前提交实验报告

课程名称：区块链技术与应用　　学期：2021 年春季

实验名称	分布式账本			实验序号	1
姓　名	***	学　号	***	班　级	***
实验地点	***	实验日期	***	指导老师	丁烨
教师评语	-			实验成绩	-
				百分制	100
同组同学					

四、　实验作业及分析

4.1　实验过程

1) *** 请将详细实验过程填写在此处 ***

4.2　实验结果

*** 请将实验结果截图填写在此处 ***

五、　实验总结

*** 请撰写一段 200 字左右的实验总结 ***


GOOD
LUCK!