区块链技术与应用期中大作业报告

一、 作业内容

利用 Python 实现一个 PoW 的仿真程序，模拟一定数量的节点生成区块链的状态。

1 设置参数包括：节点数量和每个轮次出块的成功率，测量区块链的增长速度。

2 设置一定数量的恶意节点实施攻击。1）测量不同恶意节点比例（10%-40%）条件下，统计分叉攻击成功的长度测量 2）不同恶意节点比例条件下，自私挖矿收益比例

二、 代码解析

1 程序入口是 simulate_pow.py，simulate_pow 函数参数为诚实节点数量，恶意节点数量和出块难度。默认参数设置如下：

```
def simulate_pow(honest_nodes_num=10, evil_node_num=0, difficulty="00000"):
```

修改

```
if __name__ == "__main__":
    simulate_pow(honest_node_num=10, evil_node_num=0, difficulty="000000")
```

2 仿真结果保存在 log 目录下的日志中。

3 令恶意节点攻击第一个区块(genesis 块)

```
# attackers attack the first block
evil_chain.chain[0].hack()
hack_flag = False
```

三、 实验内容

第一轮仿真

参数：honest node number = 10, evil node number = 0, **difficulty = 000000**

仿真结果：

```
13:48:54 PM: One block is dug out. Its mining time is: 91. Its miner is 7.
13:49:04 PM: One block is dug out. Its mining time is: 9. Its miner is 9.
13:49:28 PM: One block is dug out. Its mining time is: 24. Its miner is 8.
13:49:46 PM: One block is dug out. Its mining time is: 17. Its miner is 2.
13:50:33 PM: One block is dug out. Its mining time is: 47. Its miner is 5.
13:52:18 PM: One block is dug out. Its mining time is: 104. Its miner is 4.
13:55:40 PM: One block is dug out. Its mining time is: 202. Its miner is 9.
13:56:59 PM: One block is dug out. Its mining time is: 78. Its miner is 2.
13:58:31 PM: One block is dug out. Its mining time is: 92. Its miner is 5.
13:58:38 PM: One block is dug out. Its mining time is: 6. Its miner is 3.
14:01:02 PM: One block is dug out. Its mining time is: 144. Its miner is 4.
14:02:12 PM: One block is dug out. Its mining time is: 70. Its miner is 1.
14:02:56 PM: One block is dug out. Its mining time is: 42. Its miner is 6.
14:02:59 PM: One block is dug out. Its mining time is: 3. Its miner is 1.
14:04:08 PM: One block is dug out. Its mining time is: 68. Its miner is 4.
14:06:18 PM: One block is dug out. Its mining time is: 130. Its miner is 9.
14:07:13 PM: One block is dug out. Its mining time is: 54. Its miner is 1.
14:07:31 PM: One block is dug out. Its mining time is: 17. Its miner is 8.
14:07:52 PM: One block is dug out. Its mining time is: 20. Its miner is 1.
14:08:29 PM: One block is dug out. Its mining time is: 36. Its miner is 9.
```

```
14:08:29 PM: block chain length is: 21
14:08:29 PM: total evil blocks is: 0
14:08:29 PM: average mining time is: 66
14:08:29 PM: maximum mining time is: 202
14:08:29 PM: minimum mining time is: 3
14:08:29 PM: The miner account is:
14:08:29 PM: {7: 1, 9: 4, 8: 2, 2: 2, 5: 2, 4: 3, 3: 1, 1: 4, 6: 1}
```

平均出块时间: 66s，最短出块时间 3s，最长出块时间 202s

第二轮仿真：

参数：honest node number = 10, evil node number = 0, **difficulty = 00000**

仿真结果：

```
14:19:08 PM: One block is dug out. Its mining time is: 2. Its miner is 5.
14:19:13 PM: One block is dug out. Its mining time is: 4. Its miner is 1.
14:19:19 PM: One block is dug out. Its mining time is: 5. Its miner is 9.
14:19:20 PM: One block is dug out. Its mining time is: 0. Its miner is 1.
14:19:23 PM: One block is dug out. Its mining time is: 3. Its miner is 1.
14:19:28 PM: One block is dug out. Its mining time is: 4. Its miner is 9.
14:19:30 PM: One block is dug out. Its mining time is: 2. Its miner is 6.
14:19:31 PM: block chain length is: 23
14:19:31 PM: total evil blocks is: 0
14:19:31 PM: average mining time is: 3
14:19:31 PM: maximum mining time is: 7
14:19:31 PM: minimum mining time is: 0
14:19:31 PM: The miner account is:
14:19:31 PM: {3: 3, 1: 7, 5: 4, 7: 1, 4: 3, 9: 3, 6: 1}
```

平均出块时间: 3s，最短出块时间 0s（精确到个位），最长出块时间 7s

第三轮仿真

参数：honest node number = 10, **evil node number = 1**, difficulty = 00000

仿真结果：

```
14:24:33 PM: One block is dug out. Its mining time is: 0. Its miner is 3.
14:24:34 PM: One block is dug out. Its mining time is: 0. Its miner is 3.
14:24:34 PM: One block is dug out. Its mining time is: 0. Its miner is 7.
14:24:35 PM: One block is dug out. Its mining time is: 0. Its miner is 3.
14:24:35 PM: One block is dug out. Its mining time is: 0. Its miner is 2.
14:24:35 PM: One block is dug out. Its mining time is: 0. Its miner is 2.
14:24:36 PM: One block is dug out. Its mining time is: 0. Its miner is 2.
14:24:36 PM: block chain length is: 33
14:24:36 PM: total evil blocks is: 0
14:24:36 PM: average mining time is: 0
14:24:36 PM: maximum mining time is: 8
14:24:36 PM: minimum mining time is: 0
14:24:36 PM: The miner account is:
14:24:36 PM: {10: 1, 3: 7, 6: 2, 2: 5, 7: 5, 4: 1, 5: 4, 8: 4, 9: 2, 1: 2}
```

恶意节点攻击失败

第四轮仿真

参数：honest node number = 10, **evil node number = 4**, difficulty = 00000

1 到 10 号 miner 为 honest node，11 号到 14 号为 evil node

```
14:45:40 PM: node 9 starts mining.
14:45:41 PM: One block is dug out. Its mining time is: 1. Its miner is 4.
14:45:41 PM: evil node 10 is ready.
14:45:41 PM: node 10 starts mining.
14:45:42 PM: evil node 11 is ready.
14:45:42 PM: node 11 starts mining.
14:45:42 PM: One block is dug out. Its mining time is: 0. Its miner is 11.
14:45:42 PM: evil node 12 is ready.
14:45:42 PM: node 12 starts mining.
14:45:43 PM: evil node 13 is ready.
14:45:43 PM: node 13 starts mining.
14:45:43 PM: One block is dug out. Its mining time is: 0. Its miner is 13.
14:45:43 PM: Attackers hack the chain at block num 2.
14:45:43 PM: EVIL CHAIN WINS.
```

可以看到，诚实节点在领先一个区块的情况下被恶意节点瞬间反超，我认为这和 python 多线程的机制有关。python 多线程并不是真正意义上的并行，并且会先调度后创建的线程。为了更真实地模拟分叉攻击，我决定让诚实节点领先一个区块后再让恶意节点开始攻击，并且提高出块难度。

第四轮仿真

参数：honest node number = 10, **evil node number = 4**, **difficulty = 000000**

0 到 9 号 miner 为 honest node，10 号到 13 号为 evil node

```
15:34:33 PM: node 10 starts mining.
15:34:33 PM: evil node 11 is ready.
15:34:33 PM: node 11 starts mining.
15:34:33 PM: evil node 12 is ready.
15:34:33 PM: node 12 starts mining.
15:34:33 PM: evil node 13 is ready.
15:34:33 PM: node 13 starts mining.
15:34:34 PM: One block is dug out. Its mining time is: 0. Its miner is 11.
15:34:34 PM: Attackers hack the chain at block num 1.
15:34:34 PM: EVIL CHAIN WINS.
15:34:35 PM: One block is dug out. Its mining time is: 1. Its miner is 11.
15:34:35 PM: One block is dug out. Its mining time is: 0. Its miner is 13.
15:34:35 PM: One block is dug out. Its mining time is: 0. Its miner is 11.
15:34:36 PM: One block is dug out. Its mining time is: 0. Its miner is 11.
15:34:37 PM: One block is dug out. Its mining time is: 1. Its miner is 10.
15:34:39 PM: One block is dug out. Its mining time is: 1. Its miner is 11.
15:34:40 PM: One block is dug out. Its mining time is: 0. Its miner is 11.
15:34:40 PM: One block is dug out. Its mining time is: 0. Its miner is 11.
15:34:40 PM: One block is dug out. Its mining time is: 0. Its miner is 11.
15:34:40 PM: One block is dug out. Its mining time is: 0. Its miner is 12.
15:34:40 PM: One block is dug out. Its mining time is: 0. Its miner is 12.
15:34:42 PM: One block is dug out. Its mining time is: 1. Its miner is 10.
15:34:43 PM: One block is dug out. Its mining time is: 1. Its miner is 10.
```

可以看到，python 多线程优先调度后创建的线程，因此恶意节点攻击成功。

四、 实验心得

由于 python 多线程的机制，本次仿真并不能很好地模拟分叉攻击。攻击成功与否极度依赖于线程调度顺序。通过本次实验，我理解了 Prove of Work 共识协议，与 Raft 和 Paxos 达成共识的方法完全不同，POW 用计算时间来达成共识，这令我感到非常新奇。