Research and application of sports health system based on Block Chain Technology

Minghao Wang¹,², Shaofei Wu¹,², *

¹Hubei Province Key Laboratory of Intelligent Robots, Wuhan Institute of Technology, Wuhan, P.R.China
²School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan, P.R.China

*Corresponding author: 04005047@wit.edu.cn

Abstract. With the continuous development of block chain technology, people realize that this trusted underlying mechanism has important practical significance as the foundation of various activities. This paper combines block chain technology with sports health system, and improves it based on Ethereum system. In the traditional block chain system, the time to generate blocks is too slow and the centralization problem is caused by the large asset gap. We propose the concept of surplus advantage of account to determine which account has more power to generate the new block in the block chain. In this way, users of the system can maintain the system, which can further ensure the security of the whole system and the authenticity of data information, and greatly improve the efficiency of the whole system. And we write smart contracts to make sure that things always follow the agreed rules. This will also make the whole sports health system has better security and credibility. With this block chain technology as the guarantee of the underlying technology, the sports health system can be more accurate, the relevant departments for the implementation and supervision of sports health subsidies, but also ensure the authenticity and security of the data information.

Keywords: Block chain, Consensus mechanism, Sports health.

1. Introduction
The core revolutionary feature of block chain is to change the backward trust mechanism between people [1]. Block chain may completely change the way of value transmission of the whole human society and become an extremely important core technology of the next generation [2].

Block chain is a special distributed database [3]. Anyone can join the block chain network and become a node. In the block chain system each node is equal and stores the database information of the whole database. You can write data to any node, and all nodes will be synchronized to ensure the data information consistency of each node in the block chain.
2. Core technologies in block chain
Consensus mechanism [4]: in a simple way, each node in the block chain system reaches a consensus according to certain rules to decide which participating node should be used for bookkeeping, so as to ensure that all transactions in the block chain can be successfully completed. Common consensus mechanisms are as follows: PoW, PoS, PBFT, DPoS, etc. In this study, based on Ethereum system, the consensus mechanism combining PoW and PoS is adopted.

Smart contract: in fact, it is the underlying code written into the block chain to ensure that transactions and other processes are automatically executed according to the contract. And for the Ethereum system based on this research, there are two kinds of accounts: one is the external account used by the user to operate directly. The other is the contract account controlled by the released smart contract code. Smart contract is not only a computer program that can do automatically executed, but also a participant of the block chain system [5].

3. Sports health system based on block chain
With the rise of sports health industry, the proportion of people’s sports consumption is increasing. Every gymnasium needs a formal booking system to serve users and ensure the normal consumption process. At the same time, it also needs a management system to serve and supervise the health sports subsidies issued by the government to each gymnasium. Combining these with block chain technology can create a more intelligent, convenient and credible sports health system. The whole system can be managed and maintained by the relevant government departments.

4. The appropriate improvement of consensus algorithm in the sports health system
The earliest well-known network consensus is the mining consensus algorithm for creating new blocks in Bitcoin [1]. The main idea is to check that:

$$\text{hash(block head)} \leq \text{target}$$  \hspace{1cm} (1)

4.1. More optimized algorithm
However, if the above algorithm is directly applied to the sports health system in this study, there will be some problems. For example, it takes a long time to complete the transaction and the mining operation is becoming more and more professional and centralized.

To counteract the centralization trend of creating blocks, we can increase the need for memory access, such as opening up a large area in memory. And to ensure that each node produces the same array, the data in the array needs to be pseudo-random from a seed data. We consider splitting a randomly generated array stored in memory into two randomly generated arrays, one smaller array for validation and the other larger array for competing for new block publishing rights. Any location in a larger array can be generated from the corresponding data in a smaller array. The validation array is Cache and the mining array is Dataset.

4.1.1. Generate Dataset from Cache. A simple description of this process is as follows: when requesting the ith element in the Dataset, first get the content of the i%cache_size element in the Cache, take power i, and then hash it to get a temporary value mix=hash(cache[i%cache_size]). Find the next location cache_index = get_int_from_item(mix) to look for based on the result mix. Find a new mix=make_item(mix, cache[cache_index%cache_size]) based on the content at that location in the cache combined with the mix that has been calculated. Loop 256 times to ensure a certain amount of work and randomness. Hash the last mix to get Dataset[i] = hash(mix). from this all the data in the mining array Dataset can be obtained.

4.1.2. Calculate to get the result. More hash operations are still required when nodes are competing for bookkeeping right. The hash function used is Keccak256. the process is simply described as: block head and random number nonce take hash, and then a temporary value mix= Keccak256(head, nonce)
is obtained. Find the next location dataset_index based on what’s in the resulting mix, dataset_index=get_int_from_item(mix)%full_size. Since the array needed to compete for accounting right in the block chain is the mining array Dataset, the target locations in the process are all locations in the Dataset, and the nodes that compete for accounting right should put the Dataset in memory. Combine the contents of this location in the Dataset with the calculated temporary value mix to get the temporary value mix1=make_item(mix, Dataset[dataset_index]). And combine the contents of the next location in the Dataset with the calculated temporary value mix to get another temporary value mix2=make_item(mix, Dataset[dataset_index+1]). A new temporary value mix=mix(mix1, mix2) is obtained from the two temporary values mix1 and mix2 obtained above. The operation is cycled 64 times to ensure a certain amount of workload and randomness. Hash for the last mix yields result=Keccak256(mix). this result is the proof of each node’s own workload. If the result <= target is satisfied by the node’s result, the node gains accounting right for new blocks in the block chain. This block broadcasts locally assembled blocks to other nodes in the block chain for validation, and after the vast majority of nodes have been validated, you can ensure that new blocks have been successfully added to the block chain system.

4.2. Introduction of surplus_advantage

The traditional block chain serves various monetary systems, and its account mainly includes account balance. In these systems, block chain is mainly to protect the users who have more funds, so PoS appear gradually. We get inspiration from it. The sports health system is to protect the interests of users who often make an appointment to use the system. We can consider making such accounts have some advantages when competing for bookkeeping right. The system in the study can not only include the account balance in each account, but also add a new content, surplus_advantage, in order to make the user accounts who often use the system obtain certain advantages in the competition. Each time the user makes an appointment through the system and uses it, the surplus_advantage in his account will be increased by one, so that his account can gain an advantage in the competition. The surplus_advantage in each account is used to adjust the actual target that need to be met. The adjustment process is as follows: extract the first value that is not 0 in target and the two zeros in front of it. Add this value to surplus_advantage2 to get a three digit number. Replace the original position and get the actual target that needs to be meet. For example, the original target was ‘0000000012345678’, and the surplus_advantage of account is 20, then get ‘00a’ in the target and increase 202 to generate a new three digit ‘19a’. then the account needs to meet the requirement of being smaller than the target ‘0000019a12345678’ to obtain the bookkeeping right. When a certain target is lowered for the node with larger surplus_advantage in the system, the node will gain a greater advantage in competing for bookkeeping right. At the same time, the running speed of the system can be further improved by reducing the target. Every time an account gets the accounting right, the balance in this node account will be reduced to ensure the corresponding fairness of the competition. At the same time, even if there are enough residual advantages, the target can not be infinitely increased. It is stipulated that when the target is increased, the maximum increase is 602, and the account with surplus_advantage greater than 60 can only be calculated according to 602, so as to avoid serious system centralization caused by nodes having excessive surplus_advantage and always having bookkeeping right.

5. Simulation experiment

5.1. Simulation: More optimized algorithm

getCache(){
var seed = sha256(this.getBlockHeader()).toString();
var thedata = seed;
for(let i = 0; i < 10; i++){
  var theKeccak256Result = this.Keccak256(thedata);
}
this.cache.push(theKeccak256Result);
    thedata = theKeccak256Result;
  }
}
Keccak256(thedata){
  var theResult = keccak256('keccak256').update(thedata).digest('hex');
  return theResult;
}
newBlock = new Block();
newBlock.getCache();
console.log('the content of cache in newBlock:', newBlock.cache);

Figure 1. Cache obtained.

5.1.1. Simulation: Generate Dataset from Cache.
getDataset(){
  for(let i = 0; i < 20; i++){
    this.dataset.push(this.calculationDatasetI(i));
  }
}
calculationDatasetI(i){
  var cache_index = i % this.cache.length;
  var mix;
  mix = this.Keccak256(this.cache[cache_index]);
  for (let number = 0; number < 256; number++){
    cache_index = this.get_int_from_item(mix);
    mix = this.make_item(this.cache[cache_index], mix);
  }
  var dataSet_i = this.Keccak256(mix);
  return dataSet_i;
}
newBlock.getDataset();
console.log('the content of dataset in newBlock:', newBlock.dataset);
5.1.2. Simulation: Calculate to get the result.

```javascript
function generateResult() {
    var mix;
    var mix1;
    var mix2;
    var blockHeader = this.getBlockHeader();
    var dataset_index;
    mix = this.Keccak256(blockHeader + this.nonce);
    for (let i = 0; i < 64; i++) {
        dataset_index = this.getIntFromItem(mix, this.dataset.length);
        mix1 = this.makeItem(this.dataset[dataset_index], mix);
        mix2 = this.makeItem(this.dataset[dataset_index + 1], mix);
        mix = this.newMix(mix1, mix2);
    }
    var result = this.Keccak256(mix);
    console.log('the result of calculation is:', result);
    return result;
}

function mine() {
    while (true) {
        var result = this.generateResult();
        if (result < this.target) {
            break;
        }
        this.nonce ++;
    }
    return result;
}

const theResult = newBlock.mine();
console.log('need to meet the target value less than:', newBlock.target);
console.log('the result satisfying the condition after calculation is: ');
console.log(theResult);
```

Figure 2. Dataset obtained.
5.2. Simulation: Introduction of surplus_advantage

```javascript
update_target()
{
    var i;
    for(i = 0; i < this.target.length; i ++){
        if(this.target[i] !== '0'){
            break;
        }
    }
    var theThree = '00' + this.target[i];
    var threeNumber10 = parseInt(theThree, 16);
    var newThreeNumber16 = this.update_Three(surplus_advantage, threeNumber10);
    this.target = this.target.replace(theThree, newThreeNumber16);
    return 'target updated successfully';
}
update_Three(surplus_advantage, threeNumber10)
{
    var the_surplus_advantage = surplus_advantage[this.account];
    if(the_surplus_advantage > 60){
        the_surplus_advantage = 60;
    }
    var newThreeNumber10 = threeNumber10 + Math.pow(the_surplus_advantage, 2);
    var newThreeNumber16 = newThreeNumber10.toString(16);
    while(true){
        if(newThreeNumber16.length < 3){
            newThreeNumber16 = '0' + newThreeNumber16;
        }else{
            break;
        }
    }
    return newThreeNumber16;
}
function creatKey()
{
    const keyPairSender = ec.genKeyPair();
    const privateKeySender = keyPairSender.getPrivate('hex');
    const publicKeySender = keyPairSender.getPublic('hex');
    privateKey.push(privateKeySender);
    publicKey.push(publicKeySender);
}
```

Figure 3. The results satisfying the conditions are obtained
for(let i = 0; i < 5; i ++){
  creatKey();
  thenewblock = new Block();
  thenewblock.account = i;
  thenewblock.getCache();
  thenewblock.getDataset();
  thenewblock.update_target();
  var thenewresult = thenewblock.testMine();
  real_target.push(thenewblock.target);
  nonce_number.push(thenewblock.nonce);
  result.push(thenewresult);
}

for(let i = 0 ; i < 5 ; i ++){
  console.log('which account:', i + 1);
  console.log('the private key of this account:', privateKey[i]);
  console.log('the public key of this account:', publicKey[i]);
  console.log('the balance of this account:', balance[i]);
  console.log('the surplus_advantage of this account:', surplus_advantage[i]);
  console.log('the real target of this account:', real_target[i]);
  console.log('the number of attempts require to find the result of this account:', nonce_number[i]);
  console.log('the result found by this account:', result[i]);
}

Figure 4. Content in the generated three accounts.
Figure 5. Content in the generated two accounts.

For different target and different surplus_advantage, the number of operations within the system to make blocks contain different data is compared as follows: When the original threshold target=00a1bf4800000000000000000000000000000000000000000000000000000000, eleven situations were set up for the experimentation. 1. surplus_advantage is not introduced, 2. when surplus_advantage=10, 3. when surplus_advantage=20, 4. when surplus_advantage=25, 5. when surplus_advantage=30, 6. when surplus_advantage=35, 7. when surplus_advantage=40, 8. when surplus_advantage=45, 9. when surplus_advantage=50, 10. when surplus_advantage=55, 10. when surplus_advantage=60. Write different data information within a block and then get different nonces. The results are as follows:

The results of all the experimental groups were summarized as follows:

| Surplus_advantage | Sum(nonce) | Avg(nonce) | Median(nonce) |
|-------------------|------------|------------|---------------|
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 7261 | 363.05 | 252 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 981 | 49.05 | 26 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 154 | 7.7 | 6 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 106 | 5.3 | 3 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 88 | 4.4 | 2 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 48 | 2.4 | 2 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 56 | 2.8 | 2 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 46 | 2.3 | 1 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 32 | 1.6 | 1 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 29 | 1.45 | 1 |
| 00a1bf4800000000000000000000000000000000000000000000000000000000 | 21 | 1.05 | 1 |

And change the original result to 000a1bf480000000000000000000000000000000000000000000000000000000 and 00000a1bf48000000000000000000000000000000000000000000000000000000 and 000000a1bf480000000000000000000000000000000000000000000000000000000000000, then we get the following results:
Figure 9. Trend of nonce 1.

Figure 10. Trend of nonce 2.

Figure 11. Trend of nonce 3.

Figure 12. Trend of nonce 4.

Therefore, the introduction of surplus_advantage can shorten the calculation times of corresponding accounts and speed up the overall system. At the same time, users can maintain the whole system, which can more protect the rights of users.

6. Overall design of sports health system based on block chain
The government should assist the operation of sports health system, issue a reasonable smart contract. When the gymnasium publishes the information, it needs to provide the information such as the appointment time to the relevant government departments for audit and store in the block chain, so as to ensure the authenticity of the information. When the user wants to make an appointment, the user also needs to execute the corresponding code in the published smart contract to realize the reservation operation, and store the successful result of the user reservation in the block chain network. When users pay, they can enjoy different discounts according to different score. When the user makes an appointment and uses it successfully the amount of gymnasiumSubsidy in the corresponding gymnasium account will increase, and this part of subsidy can be withdrawn directly.

7. Key information of sports health system
The entity types in the system can be roughly divided into user entity and gymnasium entity.

User entity mainly includes the following contents: IDNumber(user’s ID number), userName, userAccountAddress, score(user points).

The gymnasium entity mainly includes the following contents: gymnasiumNumber(gymnasium record number, convenient to find whether it has been registered), gymansiumName, gymnasiumAccountAddress, gymnasiumRevenueAvailable(gymnasium income to be collected),
gymnasiumSubsidy (gymnasium to receive subsidy), price (single booking price of gymnasium), appointmentNumber (booking information serial number of the gymnasium), appointmentInformationList (the gymnasium booking list collection).

And it is necessary to create an appointment information structure to save the reservation records: gymnasiumNumber, gymnasiumName, appointmentPerson (ID card number of the person making the appointment), appointmentStartTime, userAccountAddress, score (reservation user points), duration (duration of use), isAppointed (have the time period been reserved), discount (subsidy discount for the subscriber).

And it is necessary to create a gymnasium withdrawal record structure to save the withdrawal record: gymnasiumNumber, gymnasiumName, gymnasiumAccountAddress, getRevenue (withdrawal amount of gymnasium’s income), getSubsidy (withdrawal amount of gymnasium’s subsidy), time (withdrawal time of the gymnasium), payee (gymnasium teller), businessHandler (issuer of withdrawal business).

The smart contract should also store the following information: gymnasiumList, userList, getMoneyRecordList (information collection of gymnasium withdrawal records) , gymnasiumTotal, userTotal , relevantDepartments (account address of relevant government departments) , subsidyAmountAvailable (the amount of subsidy remaining in the smart contract) , etc.

8. Implementation of smart contract

In order to ensure the stable operation of the sports health system based on block chain, the relevant departments need to transfer enough sports health subsidies to the contract account.

When the gymnasium carries on the account registration, it needs to go to the relevant departments for filing, which is conducive to the later review. Judge whether the operation is carried out by relevant departments, require (msg.sender == relevantDepartments); and search all registered gymnasiums in the smart contract to avoid duplicate registration. Users can register directly and search in smart contract to avoid duplicate registration.

When the gymnasium wants to publish the information available for appointment in the sports health system, they can publish it by themselves. It needs to judge whether the account address is the registered account address of the target gymnasium, require (theGymnasium.gymnasiumAccountAddress == msg.sender); to avoid misoperation by others.

When a user makes an appointment. He should first judge whether the account address initiating the reservation call is consistent with the registered account address of the user entity that wants to make an appointment to ensure the security of user accounts. At the same time, if the user wants to make a successful appointment, he should also ensure that the current appointment period has not been reserved by others, and that the amount paid by the user is enough to pay for the reservation fee. The user information of successful appointment is written into the reservation information to ensure the authenticity of the reservation transaction, and the discount subsidy of the user is automatically returned.

9. Conclusion

In this study, a sports health system based on block chain technology is implemented. It adopts a consensus mechanism which is more suitable for the system, which is maintained by users themselves to make the system more credible. And smart contracts regulate the behavior of all parties. When the system is running, manual intervention is reduced.

With the development of the times, the era of block chain is bound to come. I speculate that any industry related to trust needs upgrading and transformation. Through the use of block chain technology as the underlying architecture to establish a decentralized system, so that people can trust each other without a third party, further strengthen cooperation and accelerate social development.

References

[1] Nakamoto. S. Bitcoin: A peer-to-peer electronic cash system, information on
https://bitcoin.org/bitcoin.pdf

[2] Zheng Zibin, Xie Shaoan, Dai Hongning. Blockchain challenges and opportunities: a survey. International Journal of Web and Grid Services, 2018, 14 (4): 352-375.

[3] Pilkington. M. Blockchain Technology: Principles and Applications, J. Social Science Electronic Publishing, 2016.

[4] Liu. Y. Z, Liu. J. W, Zhang. Z. Y, Xu. T. G, Yu. H. Overview on Blockchain consensus mechanisms, J. Journal of Cryptologic Research, 2019, 6 (4): 395-432.

[5] HusamIbrahim, A Next-Generation Smart Contract and Decentralized Application Platform, information on https://github.com/ethereum/wiki/White-Paper.