A survey on Blockchain Development with Colab

Author: Krishna Adatrao

Introduction:

Blockchain is an emerging technology in present world. The fundamental utilization of blockchain techniques has a scope in financial services in an organization or private sector. Blockchain is basically relies on protecting the blocks of information in a manner like immutable objects which cannot be changed or hacked. In another words, blockchain is a ledger form of any financial transactions that is duplicated and distributed across entire network of computer systems on the blockchain.

Blockchain Development:

A blockchain is a chain of blocks that contains the data as an object in those blocks. The information which is stored in the block relies on the category of the blockchain. Further, blockchain is a decentralized, distributed, digital ledger technology which is immutable in nature and mainly utilized in the scope of **financial** transactions. Blockchain has been utilizing over plenty of computers which makes in no altering of the transaction record retroactively which helps in eliminating changes in all subsequent blocks and consensus of the network. By utilization of blockchain technology there is no need of central clearing authority in the transactions.

Different Types of Blockchains:

- Public Blockchain - No central Authority
  Public Blockchains are decentralized technology which doesn't have any restrictions, the verification of the transaction of records can also be performed.

- Private Blockchain – At least one authorization is required
  Private Blockchain is not fully decentralized technique and only specific heads can perform the task in a closed network.

- Hybrid Blockchain – No permission required
  Hybrid Blockchain is a mixture of Public and private blockchains which relies on both permissions based and permissionless systems with no alter in the transactions.

- Consortium Blockchain – Controlled by a group
  Consortium Blockchain is also called as Federated Blockchain, which validates, initiates, and receives the transactions. Both Public and Private blockchains are utilized.

There are various applications of blockchain in the current world likely, Bitcoin Block Transactions. In Bitcoin Block Transactions, there is no need of huge custom central authority permissions which makes lives easier and burden free. Moreover, the transactions are more accurate and secure at the same time. The mechanism behind the block transactions is like the

information in a block will be stored as a transaction block likely sender information, receiver information, and the amount of bitcoins to be transferred. Moreover, these types of data transaction blocks will be chained into a single unit of blockchain and every block in the chain has its own hash value to maintain the integrity of the data.

Blockchain Development with Colab:

Google research foundation has given a great solution for researchers to make their work easier by providing serverless architecture around the systems. This architecture is a cloud-based foundation which results in virtual performances of RAM, ROM, and plenty other specifications. While making any projects and research there is a need of huge data and CPU performances to make the progress running. Google came up with a great solution for this as Google Collaboratory which provides virtual CPU performance. Moreover, Colab is also provided with GPU and TPU (Tensor Processing Unit) which gives high performance. However, AWS (Amazon Web Services) is also another best serverless architecture with great security services too likely, Firewall, ACL, CDN (Content Delivery Network). In addition, AWS is provided with AWS lambda services for programming purpose along with AWS Cloud Beanstalk which makes lives easier. However, Google Colab is more flexible for research perspective.

Jupyter Notebook is a localhost mechanism which runs locally on the computer. However, Google Colab is providing cloud-based runtime with the same architecture of Jupyter Notebook to make the progresses in their CPU and GPU units.

Basic Understanding of Blockchain Mechanism was built on Colab using Python Language:

Step1: Creation of Transaction (class)
Step2: Creation of Block (class) that contains transaction objects
Step3: Generate a Hash Value for a block
Step4: Create a Blockchain and add Block Objects in it
Step5: Proof-of-Work (Validation)

In python, there are pre-defined libraries which can be utilized according to the need in our code. I have imported the libraries like hashlib and Json. Hashlib is a hashing library which is utilized to convert UTF-8 string into hash format and Json is to format UTF-8 string.

```
# Importing Required Libraries
import hashlib # Hashing Library
import json    # Formatting UTF-8 string
```

Initially, The class of Transaction_Block creates a transaction with the user response on providing the data/information of the sender, recipient, and amount. Something which gives an illustration like transferring a bitcoin block transaction. This block of transaction information will be stored for further usage. Hence, for creating the transaction I have implemented a class called Transaction_Block which creates a transaction block.

```
blockchain = []
work_proof = []

class Transaction_Block:
    # Block that contains transaction objects
    def block(sender_response, recipient_response, amount_response):
        block_creation = {'Sender': sender_response, 'Recipient': recipient_response, 'Amount': amount_response}
        blockchain.append(block_creation) # adding the block object to blockchain
```

```
class User_Responses:
    # Transaction units or data given by user for a transaction block
    def Transaction_Units():
        sender_response = input('\n\nWho is the Sender? : ')
        recipient_response = input('\nWhat is the transaction Recipient ID? : ')
        amount_response = float(input('\nHow much is the transaction amount? : '))
        return sender_response, recipient_response, amount_response
```

The block is generated which contains transaction objects. For generating hash value for a block of transaction objects the class of Hash_Value with a function hash(block) is implemented.

```
class Hash_Value:
    # Generating a Hash value for a block
    def hash(block):
        return hashlib.sha256(json.dumps(block).encode()).hexdigest()
```

Logic:

```
temp_block = [{'previous_hash': '', 'index': 0, 'transaction': [], 'nonce': 23}]
again = True
while again == True:
    sender, recipient, amount = User_Responses.Transaction_Units()
    Transaction_Block.block(sender, recipient, amount)
    print("\n\nHash Value of the block {} is: \n".format(blockchain[-1]), "Hash Value - '{}'".format(Hash_Value.hash(blockchain[-1])))
    repeat = input("\n\nDo you want to add some more transaction blocks (Y/N)? ")
    if repeat.lower() == 'y':
      again = True
    else:
      again = False

print("\n\nCreation of a blockchain with block objects: \n", blockchain, "\n\n")
```

Blockchain with block objects is created as blockchain list.

Output:

Transaction Block and Hash Value (Steps 1 - 4):

```
Who is the Sender? : Krishna

What is the transaction Recipient ID? : Partner1

How much is the transaction amount? : 100


Hash Value of the block {'Sender': 'Krishna', 'Recipient': 'Partner1', 'Amount': 100.0} is:
 Hash Value - '512e726779bf24d9193f8d6a29ce4c903457ef7c0fd05c5f51a82c1c10e11eb1'


Do you want to add some more transaction blocks (Y/N)? Y
```

```
Do you want to add some more transaction blocks (Y/N)? Y


Who is the Sender? : Adatrao

What is the transaction Recipient ID? : Partner2

How much is the transaction amount? : 200


Hash Value of the block {'Sender': 'Adatrao', 'Recipient': 'Partner2', 'Amount': 200.0} is:
 Hash Value - '04b4015056c5666291967c0b09c1426fcb1cd0cc30976ce702faad5c4eeccbe5'


Do you want to add some more transaction blocks (Y/N)? Y


Who is the Sender? : Rohin

What is the transaction Recipient ID? : Friend

How much is the transaction amount? : 20


Hash Value of the block {'Sender': 'Rohin', 'Recipient': 'Friend', 'Amount': 20.0} is:
 Hash Value - 'c811217bcd4e249ac232403fdbd7f38c386a62badbbd0653099854dc423ef922'


Do you want to add some more transaction blocks (Y/N)? N


Creation of a blockchain with block objects:
 [{'Sender': 'Krishna', 'Recipient': 'Partner1', 'Amount': 100.0}, {'Sender': 'Adatrao', 'Recipient': 'Partner2', 'Amount': 200.0}, {'Sender': 'Rohin', 'Re
```

Proof-of-Work / Validation (Step 5):

```python
class Proof_of_Work:
    def proof(block_transactions, last_hash, nonce):
        predict_hash = hashlib.sha256((str(block_transactions) + str(last_hash) + str(nonce)).encode()).hexdigest()
        print("\nChecking Hash Value: ", predict_hash)
        return predict_hash[0:2] == '00'
    def func():
        nonce = 0
        while not Proof_of_Work.proof(blockchain, Hash_Value.hash(temp_block[-1]), nonce):
            nonce += 1
        return nonce
    def check():
        work_proof.append({'previous_hash': Hash_Value.hash(blockchain[-1]), 'index': len(blockchain), 'transaction': blockchain,
                           'nonce': Proof_of_Work.func()})

Proof_of_Work.check()
print("\n\nProof of Work: \n\n", work_proof[0], "\n\n")
```

Output:

```
Checking Hash Value:  19840627f296253265fa92870b93d7f18516c867adcff035ed11f51f5faf6f39

Checking Hash Value:  53d2779e05928f1b0b79bf0a799a07bee4535b27c9a6ea0ac8154571a1df1894

Checking Hash Value:  0ce7332baace988bd849f2bdda934b5f1ee6983ba54150225447d414ee846fd1

Checking Hash Value:  9250660f1cc1f70b9f03586f417c47737aaa528e0a18d93c811094977baa9ba7

Checking Hash Value:  9167f12bf1728048ced9c41e472e4b50c08fa8d44994f4b3ce4b2446ff1f529d

Checking Hash Value:  6610fdda98018c2be7f50736e43822aadbdbefd50c8c3a3e112d8d342330b2e1

Checking Hash Value:  4302a22e7fd9e2888e22941d29c18662fc9d532f13e777b560ce384f7501520e

Checking Hash Value:  ace914558f231499cfaf674e824a6f2472db2c01fc839db5d8ed6fc2392f3704

Checking Hash Value:  e77f5193eb04374f5bfab0d7b39a17230201220ccff46f284fbe532ec212a57c

Checking Hash Value:  83fff7605d2321b7327ace118918fc24a9bfaa7167a2c1835287b33feee33f0b

Checking Hash Value:  247e45419bf3bc35e2bc5bef1ac10b36e4fd8ee59d5ca65c18bad0d03535edab

Checking Hash Value:  be1e440f2458619c003c09c50762a3e31420a0748d04ea90d1106c4911d7e6e4

Checking Hash Value:  9f9a8a14966bbf9d22e0b3c38cbc0737604ba5800be7c6e3a7cefd774a94885d

Checking Hash Value:  9c455ba188b5c0ad8c5444ea048d7fd25d323eb2f4e55173fece6176b9067f57

Checking Hash Value:  b998d64f7244760b425ebbf2175c6f9d89a6c6b76dbf08679d380d5dbc5abc80
```

```
Checking Hash Value:  c7ac10cd9b4594acbdc4ca3dfe7d91661ecbb9e09cd5058fed9bc1167f62bedd
Checking Hash Value:  85fa0f444c4d9c463383645e96131262f0eb80e5ab1650ff21067709d1eae005
Checking Hash Value:  d0c61bff39312ddb54a22cfa5452aae629801d9056eeecbc4d62eca9b6164b65
Checking Hash Value:  52977aff07b19a88763c69f23dd262c876dafa939a84eb7d4d3d897dac0738fd
Checking Hash Value:  af110d647d046bcfd151a11df2255650767bce40e39aedf8f69f07c4f2b8a4a6
Checking Hash Value:  d3f3404feb515536cfa58c4fe28c41d192bb8734a5111378a07c8c7f491b3cd9
Checking Hash Value:  37bc0c60259f055fcfe11da8d9fc6cba07c8ece1a12bcf9c09cb9042fe3e326f
Checking Hash Value:  79bbf55ba7675d61bdcb9b365e0dae2cc6ac842db0e92fbdbb700bc145726761
Checking Hash Value:  f4b47ff95ab22362f5f9298754c4d1861fd31fb740b91d120b266dbb15fb6d10
Checking Hash Value:  872487c141e56ea76fed4d1114696df52bf6596e909fa8165ce17e04c7afe707
Checking Hash Value:  54fcee706d4ff44d0a6211c30a14ea80f0acd6690ac60e3794f31562c6afad4b
Checking Hash Value:  2b33ed6fd9d0a1f7dc82d7be2b0fcde954d9640c65836211e11cca4ec51279c8
Checking Hash Value:  f956b4811567a7b182bdfbb227462f1cd78891b1b3f988f8158be0b6e2e15702
Checking Hash Value:  8a55d7034b92ddc157f712211bb86cdc9a43adcb2c8ef0fecda516e35906eb04
Checking Hash Value:  42522ef79f8e13b18dabf5c6516c9481a08c7ed464a74f915d87b3d5047ac762
Checking Hash Value:  57e3cc033b92430cae82cd7c8591d82e81900ba4632cbd93cbc1bab383ee37f6
```

```
Checking Hash Value:  a13a3176dc6af65b58c23a7d227afc754c1e46b652f99825fdd49c34466dd17c
Checking Hash Value:  edbfb7e752f5f42ed42188281d84671abbd11a9e53f7a5752713a4ab9e0de9cd
Checking Hash Value:  c4e2addddfdeea4a9f76017510a911491fd535438c28123d40ad362007132830
Checking Hash Value:  b96e5c526fedee9b1da080a6b046a6efa484ea8259c55c7e900d6ef0958672f8
Checking Hash Value:  aa63598685c2b6e30fc7a6c24850a8dc14b9631ac98de56f937e903ff978a7a1
Checking Hash Value:  033e0f6a1e28f7b2ef339b6baea1d476b5051d9dbf6e8ac001900305c1bb7ec4
Checking Hash Value:  18e62d9b76c5d681eaa1df11f5dd00564b7498571488d2697c96c4786d98a37d
Checking Hash Value:  25cbbb985b4079257e0d7f7953e50e800e268e1ab12e9ee03a0a79ac39b8d837
Checking Hash Value:  880b64e6e9d99a150ced28fa0b645e4e418736792881e97c6b3265883e6bf618
Checking Hash Value:  bb158623992189a80cba03bc6b1f6cdd55f1ce0bd29016a5c7ce2728813fb144
Checking Hash Value:  ac719be9d499f87d1c5e0c2e47f0955019e6f0a8ad24d1c6df75358805827d81
Checking Hash Value:  b738f9033ed9e7c0bc087412674e0f0162439556c11905ea9fae5ddb3e1c31ac
Checking Hash Value:  452752dc2410f4e8a07068f504192dc117624c86feed60e7efc883022fd9e507
Checking Hash Value:  9a35dbc17418e9cbd5fc091a0a2004c1a5c92160d311538a8baffddefae4b25a
Checking Hash Value:  a52e52c2b3b838c37d24b263963963154460689c3423b0a2841bc4e2ce7aff56
Checking Hash Value:  b59303015cb500b9fa9161b39429e1ade869c2c2747acb9eceac29fcfdcbb9fc
Checking Hash Value:  eda917ed2ae6533592847ce9ea3c2e63d6f7f7d593cc1b9711ce4cecff540ec7
Checking Hash Value:  38576bbad23fd62265622a4db2a18b04ba105aba44279c53d7b037df7852666c
Checking Hash Value:  3764d50f3e20b1e618607fc808b1c1d15d27851ea1db52e3b921416779ffa6b0
Checking Hash Value:  00adc1e2f1068a6f77ba882a7c46c61482e73372c797bb357cf9eceea4db15d9
```

```
Proof of Work:
 {'previous_hash': 'c811217bcd4e249ac232403fdbd7f38c386a62badbbd0653099854dc423ef922', 'index': 3, 'transaction': [{'Sender': 'Krishna', 'Recipient': 'Par
```

Advantages, Disadvantages, and Applications of Blockchain:

Advantages	• Provides Security and Integrity for the data/information • Build trust among user transactions • Minimizes the production cost • More economical • Quick and fast transactions • Immutable in nature
Disadvantages	• No modifications can take place • Requires huge storage space • Even the Authorized user can't access the data if he forgets the private key
Applications	• Voting Management System • Chain Management System • Healthcare • Real Estate • NFT Marketplace • Patent Right Balancing • Immutable Storage • IoT

Conclusion:

This survey is based on blockchain technology with the integration of the concept utilization on cloud-based architecture which is Google Colab with python language. The implementation of blockchain with block objects of transactions has been implemented and the transaction objects are generated into hash values to make immutable structure of blockchain. Finally, the validation of this hash values is also illustrated with the running implementation of the program.