Blockchain & Cryptocurrency Lab 03

K21-4579 K21-4871

Task 2: Distributed Proof-of-Work Across Decentralized Nodes

Objective

The purpose of this task is to implement a distributed proof-of-work (POW) mechanism executed concurrently on multiple nodes. Each node processes an incoming transaction by attempting to solve a cryptographic puzzle. Only the first node to complete the POW updates the shared ledger with detailed timing information, allowing for precise comparison of processing times.

Implementation Details

Key Components

- Decentralized Nodes
  Three server nodes (Node1, Node2, and Node3) are started on separate ports (5000, 5001, and 5002, respectively). Each node is hosted within its own thread, enabling simultaneous POW computation.

- Proof-of-Work Function (pow.py)
  Implements a simple POW algorithm that iterates nonces until the resulting hash meets a specified difficulty (expressed as a requirement for leading zero pairs). The function returns the nonce, hash, and detailed time taken without rounding off the precision.

- Ledger File (ledger.json)
  A JSON file that logs the POW details only from the winning node. Each entry contains the node identifier, transaction data, nonce, resulting hash, processing time, and both start and end timestamps recorded with microsecond precision.

- Precise Time Recording
  The distributed node server records start and end times with microsecond granularity using Python's datetime module. This ensures that the timings do not round off to the nearest second, allowing for accurate performance comparisons between nodes.

File Descriptions

- distributed_pow_server.py
  - Listens for transaction data on a designated port.
  - Upon receiving a transaction, it spawns a new thread for POW processing on that node.
  - The POW process records the start and end times with full microsecond precision.
  - After the computation, the node that finishes first (i.e., the one that sets the shared `winner_event`) appends its result to the ledger.
  - Contains print statements that display only the precise start and end times (and winning announcement) in the terminal.

- pow.py
  - Contains the core POW algorithm.

  - Iterates to find a valid nonce so that the SHA-256 hash of the concatenated transaction data and nonce meets the difficulty requirement.
  - Returns results with full time precision.

- Ledger File (ledger.json)
  - Stores a record of transactions with the POW details, ensuring a historical log of each winning computation.

Commands to Run the Project

To properly test the system, execute the following commands in separate terminals one after the other:

1. Start the distributed nodes:

```
->   python3 distributed_pow_server.py
```

2. Send a transaction concurrently to all nodes:

```
->   echo "Your transaction data" | tee >(nc localhost 5000) >(nc localhost 5001) >(nc localhost 5002) > /dev/null
```

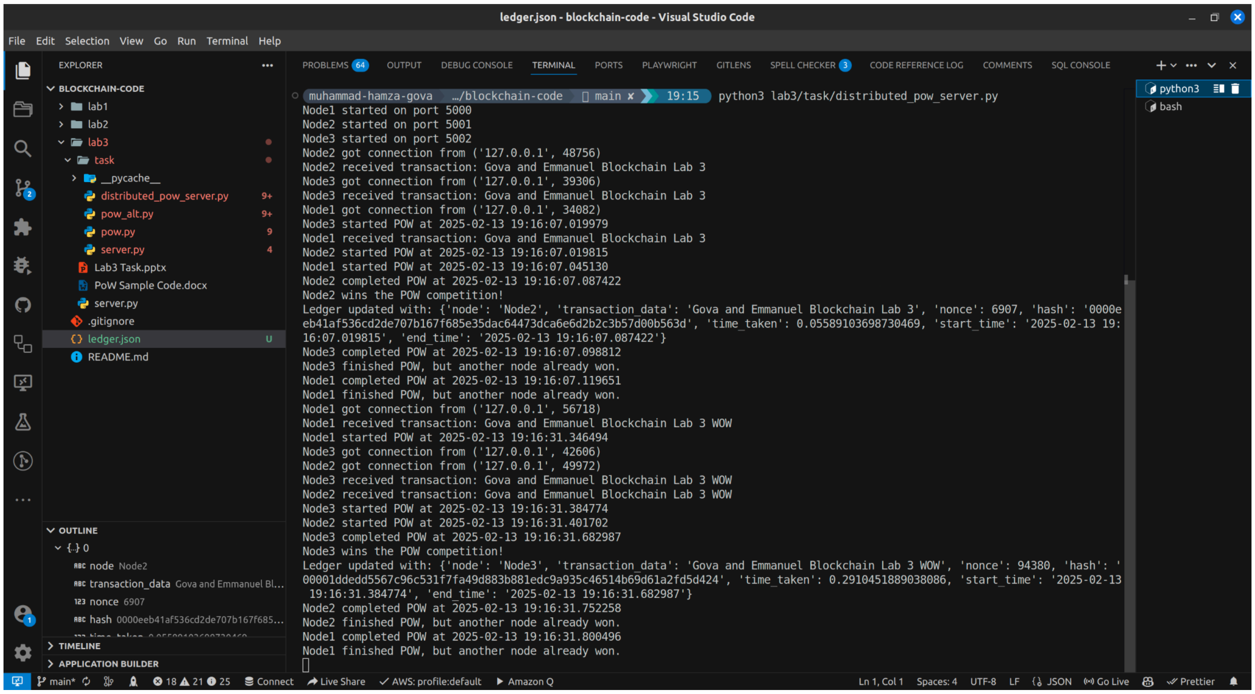
Results

- All three nodes start concurrently and listen for incoming transactions.
- When the transaction is sent, each node begins the POW process concurrently.
- The terminal output displays only the starting and ending times (with detailed millisecond and microsecond precision) for each node’s POW attempt.
- The shared ledger (ledger.json) logs the POW details of the winning node, ensuring that only one node’s result (the fastest) is recorded for each transaction.

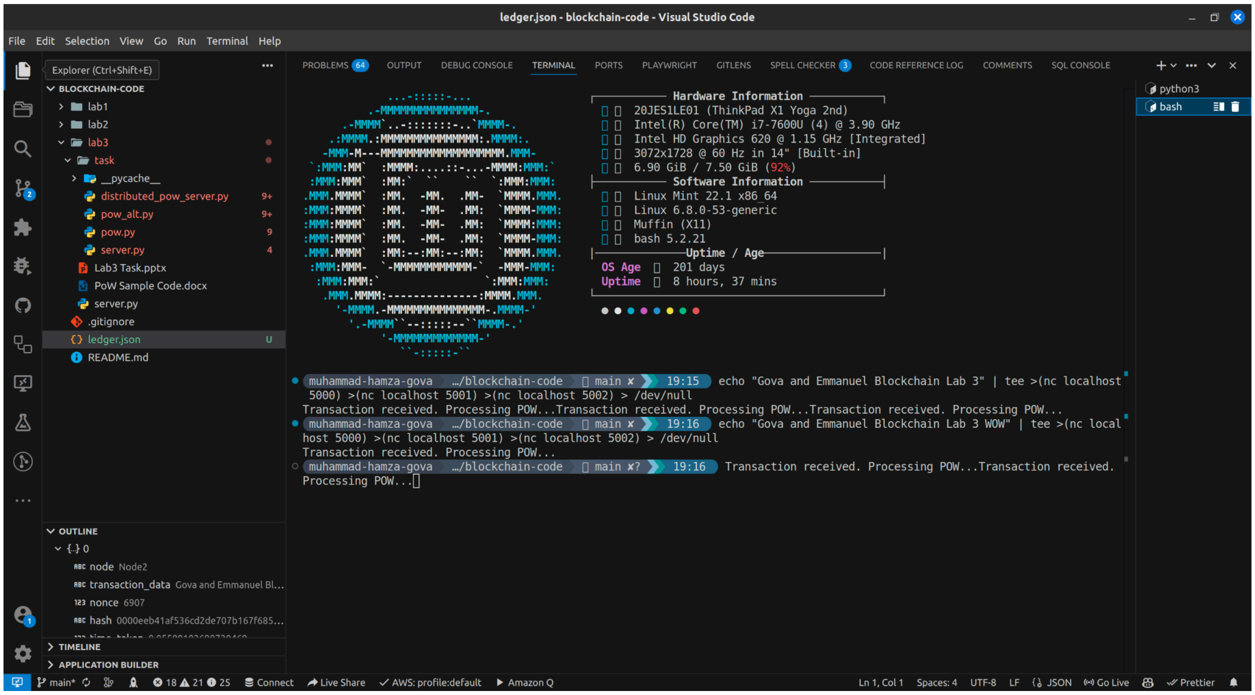
Conclusion

This task successfully demonstrates a distributed approach to executing proof-of-work. By leveraging concurrent processing across decentralized nodes and capturing high-precision timing data, the system ensures fairness and provides granular performance metrics. The implementation forms a foundational component for more complex distributed consensus and blockchain-based applications.

Blockchain & Cryptocurrency Lab 03
K21-4579 K21-4871



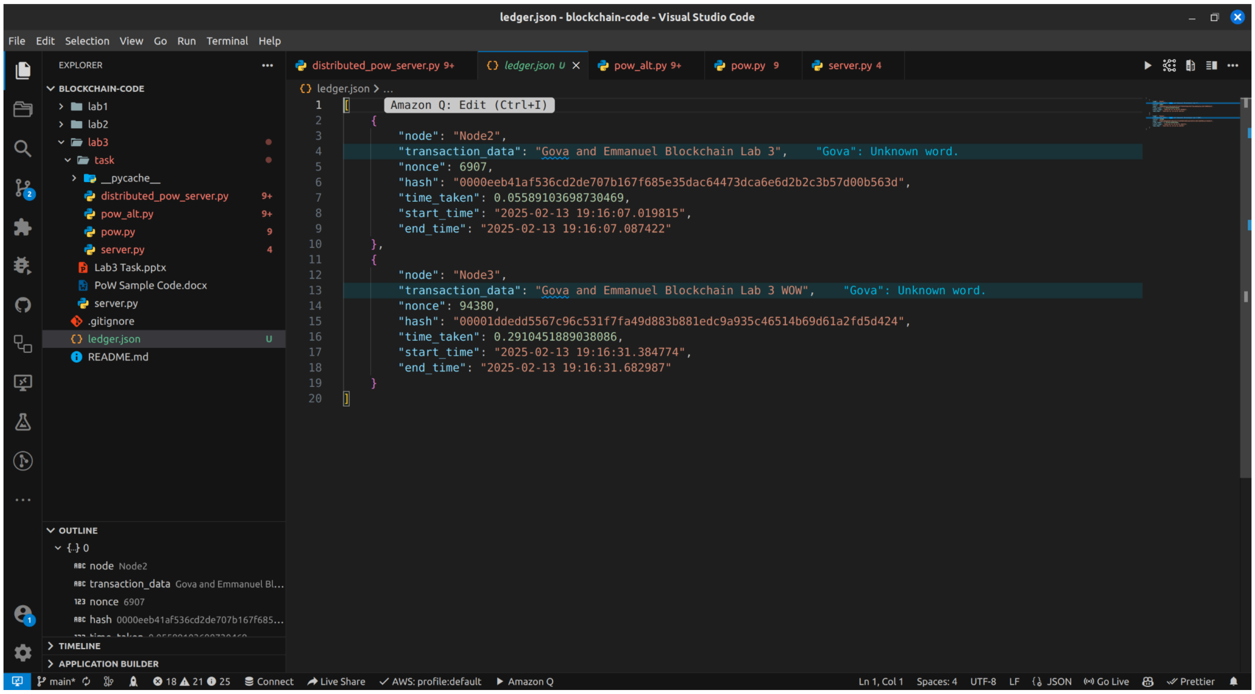
```
muhammad-hamza-gova  …/blockchain-code   main ✘  19:15  python3 lab3/task/distributed_pow_server.py
Node1 started on port 5000
Node2 started on port 5001
Node3 started on port 5002
Node2 got connection from ('127.0.0.1', 48756)
Node2 received transaction: Gova and Emmanuel Blockchain Lab 3
Node3 got connection from ('127.0.0.1', 39306)
Node3 received transaction: Gova and Emmanuel Blockchain Lab 3
Node1 got connection from ('127.0.0.1', 34082)
Node3 started POW at 2025-02-13 19:16:07.019979
Node1 received transaction: Gova and Emmanuel Blockchain Lab 3
Node2 started POW at 2025-02-13 19:16:07.019815
Node1 started POW at 2025-02-13 19:16:07.045130
Node2 completed POW at 2025-02-13 19:16:07.087422
Node2 wins the POW competition!
Ledger updated with: {'node': 'Node2', 'transaction_data': 'Gova and Emmanuel Blockchain Lab 3', 'nonce': 6907, 'hash': '0000e
eb41af536cd2de707b167f685e35dac64473dca6e6d2b2c3b57d00b563d', 'time_taken': 0.05589103698730469, 'start_time': '2025-02-13 19:
16:07.019815', 'end_time': '2025-02-13 19:16:07.087422'}
Node3 completed POW at 2025-02-13 19:16:07.098812
Node3 finished POW, but another node already won.
Node1 completed POW at 2025-02-13 19:16:07.119651
Node1 finished POW, but another node already won.
Node1 got connection from ('127.0.0.1', 56718)
Node1 received transaction: Gova and Emmanuel Blockchain Lab 3 WOW
Node1 started POW at 2025-02-13 19:16:31.346494
Node3 got connection from ('127.0.0.1', 42606)
Node2 got connection from ('127.0.0.1', 49972)
Node3 received transaction: Gova and Emmanuel Blockchain Lab 3 WOW
Node2 received transaction: Gova and Emmanuel Blockchain Lab 3 WOW
Node3 started POW at 2025-02-13 19:16:31.384774
Node2 started POW at 2025-02-13 19:16:31.401702
Node3 completed POW at 2025-02-13 19:16:31.682987
Node3 wins the POW competition!
Ledger updated with: {'node': 'Node3', 'transaction_data': 'Gova and Emmanuel Blockchain Lab 3 WOW', 'nonce': 94380, 'hash': '
00001ddedd5567c96c531f7fa49d883b881edc9a935c46514b69d61a2fd5d424', 'time_taken': 0.2910451889038086, 'start_time': '2025-02-13
 19:16:31.384774', 'end_time': '2025-02-13 19:16:31.682987'}
Node2 completed POW at 2025-02-13 19:16:31.752258
Node2 finished POW, but another node already won.
Node1 completed POW at 2025-02-13 19:16:31.800496
Node1 finished POW, but another node already won.
```



```
           ...-:::::-...                  ┌──────── Hardware Information ────────┐
        .-MMMMMMMMMMMMMMM-.                 20JES1LE01 (ThinkPad X1 Yoga 2nd)
     .-MMMM`..-:::::::-..`MMMM-.            Intel(R) Core(TM) i7-7600U (4) @ 3.90 GHz
   .:MMMM.:MMMMMMMMMMMMMMM:.MMMM:.          Intel HD Graphics 620 @ 1.15 GHz [Integrated]
  -MMM-M---MMMMMMMMMMMMMMMMMMM.MMM-         3072x1728 @ 60 Hz in 14" [Built-in]
`:MMM:MM`  :MMMM:....::-...-MMMM:MMM:`      6.90 GiB / 7.50 GiB (92%)
:MMM:MMM`  :MM:`  ``    ``  `:MMM:MMM:    ├──────── Software Information ────────┤
.MMM.MMMM`  :MM.  -MM.  .MM-  `MMMM.MMM.      Linux Mint 22.1 x86_64
:MMM:MMMM`  :MM.  -MM-  .MM:  `MMMM-MMM:      Linux 6.8.0-53-generic
:MMM:MMMM`  :MM.  -MM-  .MM:  `MMMM:MMM:      Muffin (X11)
:MMM:MMMM`  :MM.  -MM-  .MM:  `MMMM-MMM:      bash 5.2.21
.MMM.MMMM`  :MM:--:MM:--:MM:  `MMMM.MMM.  |──────────Uptime / Age──────────────|
 :MMM:MMM-  `-MMMMMMMMMMMM-`  -MMM-MMM:     OS Age   201 days
  :MMM:MMM:`                `:MMM:MMM:      Uptime   8 hours, 37 mins
   .MMM.MMMM:--------------:MMMM.MMM.     └──────────────────────────────────────┘
     '-MMMM.-MMMMMMMMMMMMMMM-.MMMM-'
       '.-MMMM``--:::::--``MMMM-.'          ● ● ● ● ● ● ● ●
            '-MMMMMMMMMMMMM-'
               ``-:::::-``

● muhammad-hamza-gova  …/blockchain-code   main ✘  19:15  echo "Gova and Emmanuel Blockchain Lab 3" | tee >(nc localhost
 5000) >(nc localhost 5001) >(nc localhost 5002) > /dev/null
Transaction received. Processing POW...Transaction received. Processing POW...Transaction received. Processing POW...
● muhammad-hamza-gova  …/blockchain-code   main ✘  19:16  echo "Gova and Emmanuel Blockchain Lab 3 WOW" | tee >(nc local
host 5000) >(nc localhost 5001) >(nc localhost 5002) > /dev/null
Transaction received. Processing POW...
○ muhammad-hamza-gova  …/blockchain-code   main ✘?  19:16  Transaction received. Processing POW...Transaction received.
Processing POW...
```

Blockchain & Cryptocurrency Lab 03
K21-4579   K21-4871



```json
[
    {
        "node": "Node2",
        "transaction_data": "Gova and Emmanuel Blockchain Lab 3",
        "nonce": 6907,
        "hash": "0000eeb41af536cd2de707b167f685e35dac64473dca6e6d2b2c3b57d00b563d",
        "time_taken": 0.05589103698730469,
        "start_time": "2025-02-13 19:16:07.019815",
        "end_time": "2025-02-13 19:16:07.087422"
    },
    {
        "node": "Node3",
        "transaction_data": "Gova and Emmanuel Blockchain Lab 3 WOW",
        "nonce": 94380,
        "hash": "00001ddedd5567c96c531f7fa49d883b881edc9a935c46514b69d61a2fd5d424",
        "time_taken": 0.2910451889038086,
        "start_time": "2025-02-13 19:16:31.384774",
        "end_time": "2025-02-13 19:16:31.682987"
    }
]
```