Fundamental Distributed System Project Report Question 1

1. Vector Clocks and Causal Ordering

🔧 Implementation Breakdown

1. Node Implementation with Vector Clocks

Each server.py instance should:

- Maintain its unique ID and vector clock (e.g., {A: 0, B: 0, C: 0})
- Store local key-value pairs
- Handle read/write requests

2. Vector Clock Logic

In vector_clock.py:

- Increment your own node's clock on local write
- Attach the current clock state when sending updates to other nodes
- Merge incoming clocks using element-wise max
- Determine "happens-before" relationships for delivery decisions

3. Causal Write Propagation

In kv_store.py and buffer.py:

- If a write's vector clock is causally dependent on unseen events, buffer it
- Periodically check if conditions are met to apply buffered writes

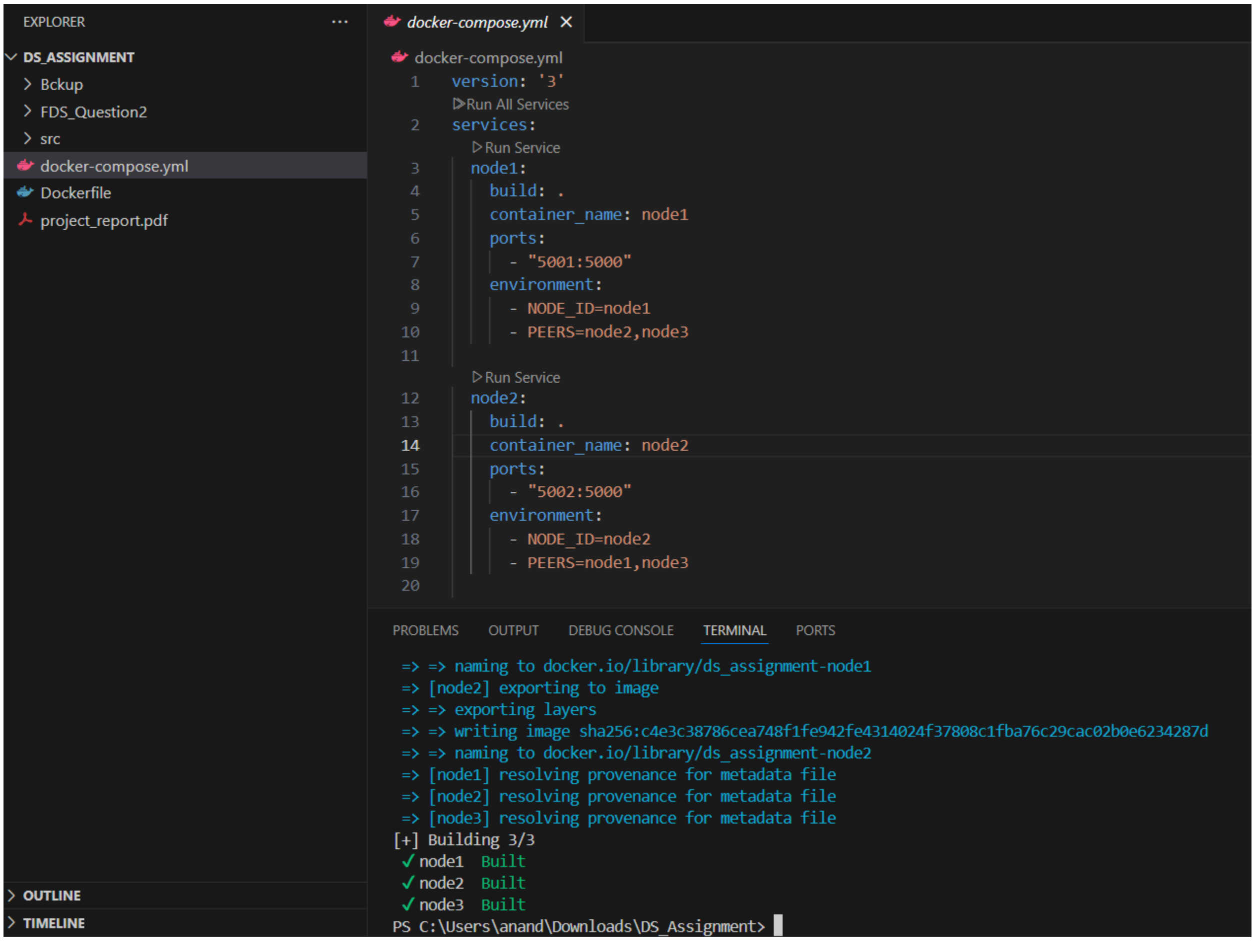
4. Containerization and Networking

- Each node runs from a shared Dockerfile, parameterized by environment variables
- docker-compose.yml will orchestrate startup and inter-node communication via network aliases

5. Verification and Scenario Testing

In client/test_scenario.py:

- Emulate reads, followed by causally dependent writes
- Introduce out-of-order message delivery (e.g., delay one node artificially using time.sleep() or iptables in the container)



```yaml
version: '3'
services:
  node1:
    build: .
    container_name: node1
    ports:
      - "5001:5000"
    environment:
      - NODE_ID=node1
      - PEERS=node2,node3

  node2:
    build: .
    container_name: node2
    ports:
      - "5002:5000"
    environment:
      - NODE_ID=node2
      - PEERS=node1,node3

```

```
 => => naming to docker.io/library/ds_assignment-node1
 => [node2] exporting to image
 => => exporting layers
 => => writing image sha256:c4e3c38786cea748f1fe942fe4314024f37808c1fba76c29cac02b0e6234287d
 => => naming to docker.io/library/ds_assignment-node2
 => [node1] resolving provenance for metadata file
 => [node2] resolving provenance for metadata file
 => [node3] resolving provenance for metadata file
[+] Building 3/3
 ✓ node1  Built
 ✓ node2  Built
 ✓ node3  Built
PS C:\Users\anand\Downloads\DS_Assignment> 
```

```
PS C:\Users\anand\Downloads\DS_Assignment> docker-compose build --no-cache

time="2025-06-19T00:27:31-04:00" level=warning msg="C:\\Users\\anand\\Downloads\\DS_Assignment\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"

Compose can now delegate builds to bake for better performance.

 To do so, set COMPOSE_BAKE=true.

[+] Building 31.8s (19/21)                                   docker:desktop-linux

 => [node1 internal] load build definition from Dockerfile                  0.1s

 => => transferring dockerfile: 111B                               0.0s

 => [node3 internal] load build definition from Dockerfile                  0.1s

 => => transferring dockerfile: 111B                               0.0s

 => [node2 internal] load build definition from Dockerfile                  0.1s

 => => transferring dockerfile: 111B                               0.0s

 => [node2 internal] load metadata for docker.io/library/python:3.9                  0.3s
 => [node3 internal] load .dockerignore                                                          0.1s

 => => transferring context: 2B                                                            0.0s

 => [node1 internal] load .dockerignore                                                          0.0s

 => => transferring context: 2B                                                            0.0s

 => [node2 internal] load .dockerignore                                                          0.1s

 => => transferring context: 2B                                                            0.0s

 => [node2 1/3] FROM
docker.io/library/python:3.9@sha256:e2d6f8be31a35665d3c39561ac2e96ece5b847e49ff84c462ab1d6850900ba7d
30.1s  => => resolve
docker.io/library/python:3.9@sha256:e2d6f8be31a35665d3c39561ac2e96ece5b847e49ff84c462ab1d6850900ba7d
0.1s  => => sha256:e2d6f8be31a35665d3c39561ac2e96ece5b847e49ff84c462ab1d6850900ba7d 10.35kB / 10.35kB
0.0s

 => => sha256:94fc7903e767dcafefeade7815b96b3d1b8e19c648b24141736c8434ee21aa59 2.32kB / 2.32kB
0.0s

 => => sha256:3b1eb73e993990490aa137c00e60ff4ca9d1715bafb8e888dbb0986275edb13f 24.02MB / 24.02MB
2.9s

 => => sha256:ce13473f18a0d44f7a5c56cd283075b1b66de98cf51459acbce5756e31b3a18d 6.18kB / 6.18kB
0.0s

 => => sha256:0c01110621e0ec1eded421406c9f117f7ae5486c8f7b0a0d1a37cc7bc9317226 48.49MB / 48.49MB
5.2s

 => => sha256:b1b8a0660a31403a35d70b276c3c86b1200b8683e83cd77a92ec98744017684a 64.40MB / 64.40MB
8.6s

 => => sha256:48b8862a18fa961ebfbac8484877dd4894e96ee88177d8c4f1f54d9727262b7d 211.37MB / 211.37MB
14.6s

 => => sha256:a7d2ea3f33a8c566c6e27c0e123d043ab74d29b26c2b22f01fc65131ccb90bf8 6.16MB / 6.16MB
6.1s

 => => extracting sha256:0c01110621e0ec1eded421406c9f117f7ae5486c8f7b0a0d1a37cc7bc9317226
6.1s

 => => sha256:b6754ffd126059cf8be2be19089f13392b63ecd646f6dfb411c015fce35a15d2 19.85MB / 19.85MB
8.4s

 => => sha256:f57f9892d27393f8c8cea049f776d6cd4eb831a8be16b3e89db6ef64efb93fdb 251B / 251B
8.6s

 => => extracting sha256:3b1eb73e993990490aa137c00e60ff4ca9d1715bafb8e888dbb0986275edb13f
1.8s

 => => extracting sha256:b1b8a0660a31403a35d70b276c3c86b1200b8683e83cd77a92ec98744017684a
4.0s

 => => extracting sha256:48b8862a18fa961ebfbac8484877dd4894e96ee88177d8c4f1f54d9727262b7d
9.5s

 => => extracting sha256:a7d2ea3f33a8c566c6e27c0e123d043ab74d29b26c2b22f01fc65131ccb90bf8
0.4s

 => => extracting sha256:b6754ffd126059cf8be2be19089f13392b63ecd646f6dfb411c015fce35a15d2
1.1s
```

```
 => => extracting sha256:f57f9892d27393f8c8cea049f776d6cd4eb831a8be16b3e89db6ef64efb93fdb
0.0s

 => [node1 internal] load build context                                          0.1s

 => => transferring context: 1.91kB                                             0.0s

 => [node3 internal] load build context                                          0.1s

 => => transferring context: 1.91kB                                             0.0s

 => [node2 internal] load build context                                          0.1s

 => => transferring context: 1.91kB                                             0.0s

 => [node2 2/3] WORKDIR /app                                                    0.3s

 => [node1 3/3] COPY node.py .                                                 0.1s

 => [node3] exporting to image                                                 0.3s

 => => exporting layers                                                      0.1s

 => => writing image sha256:c7590b18fea2fba0f543575b6e502780dda5f35499191d11d2fd8278e261de0a
0.0s

 => => naming to docker.io/library/ds_assignment-node3                                   0.0s

 => [node1] exporting to image                                                 0.3s

 => => exporting layers                                                      0.1s

 => => writing image sha256:f6bac17d0c0c6df0aac6f3ddc9c0a5ded37cb17766caac7ee84aee4e01c91440
0.0s

 => => naming to docker.io/library/ds_assignment-node1                                   0.0s

 => [node2] exporting to image                                                 0.3s

 => => exporting layers                                                     0.1s

 => => writing image sha256:c4e3c38786cea748f1fe942fe4314024f37808c1fba76c29cac02b0e6234287d
0.0s  => => naming to docker.io/library/ds_assignment-node2                                   0.1s

 => [node1] resolving provenance for metadata file                                   0.2s

 => [node2] resolving provenance for metadata file                                   0.1s

 => [node3] resolving provenance for metadata file                                   0.1s

[+] Building 3/3

 ✓ node1  Built                                                        0.0s

 ✓ node2  Built                                                        0.0s

 ✓ node3  Built                                                        0.0s

PS C:\Users\anand\Downloads\DS_Assignment>
```