CS1140

Computer Networks

Assignment 1

Submitted To:
Dr. R.K. Ghosh
Mr. Devendra Bhavsar

(JKLU)

Submitted By: Siddhi Nyati

Roll No: 2022btech101

Department of Computer Science Engineering

Institute of Engineering & Technology (IET)

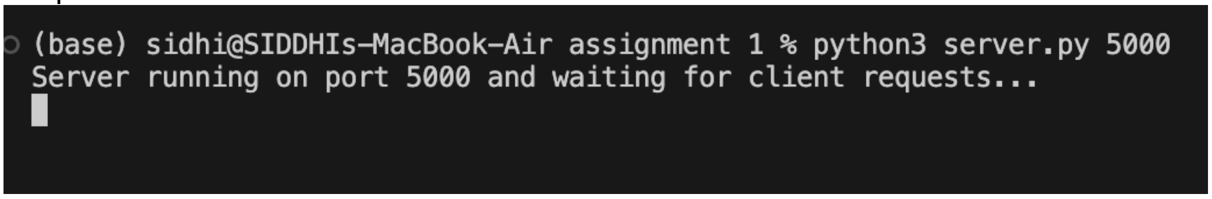
JK Lakshmipat University

Sept 2024

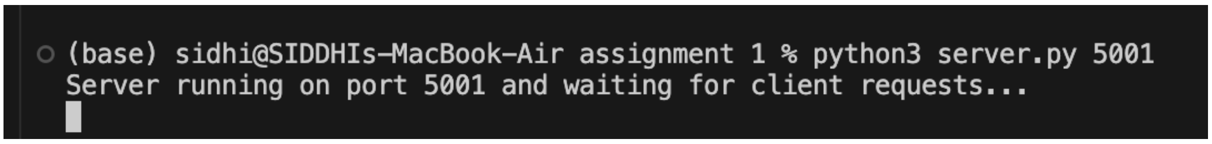
Server Code:

```
import socket

class UDPClient:
    def __init__(self, total_servers=5):
        self.total_servers = total_servers
        self.partition_sums = [0] * total_servers
        self.received_count = 0
        self.server_ports = [5000 + i for i in range(total_servers)]

    def send_request(self, partition, n, t):
        client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        start_index = partition * (n // self.total_servers)
        end_index = start_index + (n // self.total_servers) - 1

        request_message = f"Request,1,{partition},{start_index},{end_index},{n},{t}"
        client_socket.sendto(request_message.encode(), ('localhost', self.server_ports[partition]))
        print(f"Request sent for partition {partition} to server on port {self.server_ports[partition]}")

        return client_socket

    def handle_response(self, client_socket):
        data, _ = client_socket.recvfrom(1024)
        response = data.decode().split(',')
        partition_index = int(response[2])
        partition_sum = float(response[6])

        self.partition_sums[partition_index] = partition_sum
        self.received_count += 1

        print(f"Received partition {partition_index} sum: {partition_sum}")
        print(f"Current partition sums: {self.partition_sums}")

        if self.received_count == self.total_servers:
            total_sum = sum(self.partition_sums)
            print(f"Final total sum: {total_sum}")

    def compute_result(self, n, t):
        for partition in range(self.total_servers):
            client_socket = self.send_request(partition, n, t)
            self.handle_response(client_socket)
            client_socket.close()

if __name__ == "__main__":
    try:
        n = int(input("Enter the total number of terms (N): "))
        t = int(input("Enter the exponent value (T): "))

        if n <= 0 or n % 5 != 0:
            raise ValueError("N must be a positive integer and divisible by 5.")

        udp_client = UDPClient()
        udp_client.compute_result(n, t)
    except ValueError as ve:
        print(f"Input Error: {ve}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

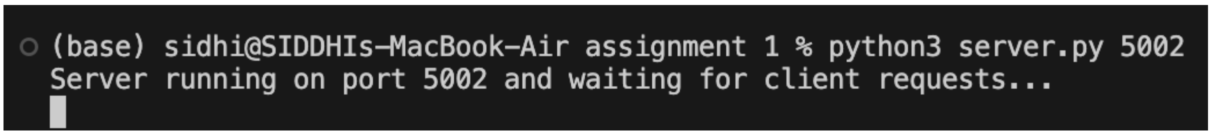
Output:



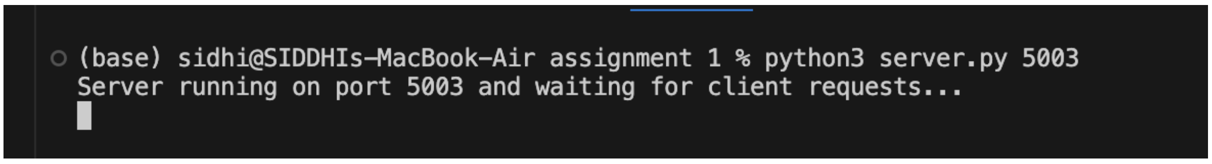
```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5000
Server running on port 5000 and waiting for client requests...
```



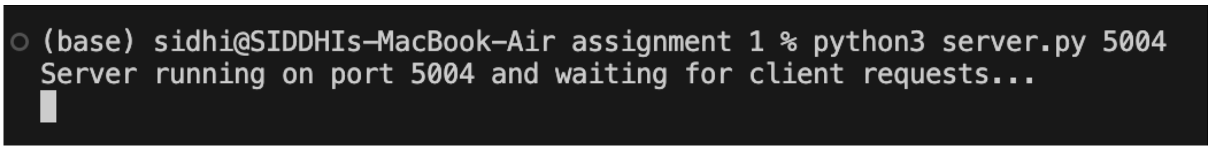
```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5001
Server running on port 5001 and waiting for client requests...
```



```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5002
Server running on port 5002 and waiting for client requests...
```



```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5003
Server running on port 5003 and waiting for client requests...
```

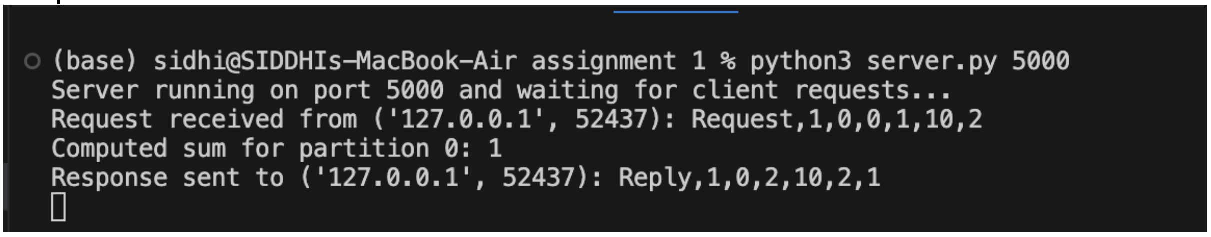


```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5004
Server running on port 5004 and waiting for client requests...
```

Client's Code:

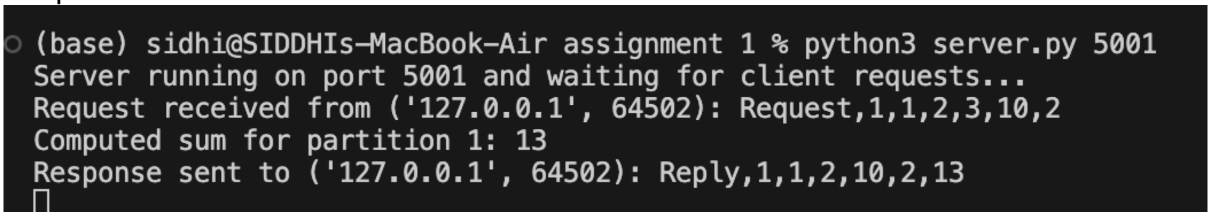
```
import socket

class UDPClient:
    def __init__(self, total_servers=5):
        self.total_servers = total_servers
        self.partition_sums = [0] * total_servers
        self.received_count = 0
        self.server_ports = [5000 + i for i in range(total_servers)]

    def send_request(self, partition, n, t):
        client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        start_index = partition * (n // self.total_servers)
        end_index = start_index + (n // self.total_servers) - 1

        request_message = f"Request,1,{partition},{start_index},{end_index},{n},{t}"
        client_socket.sendto(request_message.encode(), ('localhost', self.server_ports[partition]))
        print(f"Request sent for partition {partition} to server on port {self.server_ports[partition]}")

        return client_socket

    def handle_response(self, client_socket):
        data, _ = client_socket.recvfrom(1024)
        response = data.decode().split(',')
        partition_index = int(response[2])
        partition_sum = float(response[6])

        self.partition_sums[partition_index] = partition_sum
        self.received_count += 1

        print(f"Received partition {partition_index} sum: {partition_sum}")
        print(f"Current partition sums: {self.partition_sums}")

        if self.received_count == self.total_servers:
            total_sum = sum(self.partition_sums)
            print(f"Final total sum: {total_sum}")

    def compute_result(self, n, t):
        for partition in range(self.total_servers):
            client_socket = self.send_request(partition, n, t)
            self.handle_response(client_socket)
            client_socket.close()

if __name__ == "__main__":
    try:
        n = int(input("Enter the total number of terms (N): "))
        t = int(input("Enter the exponent value (T): "))

        if n <= 0 or n % 5 != 0:
            raise ValueError("N must be a positive integer and divisible by 5.")

        udp_client = UDPClient()
        udp_client.compute_result(n, t)
    except ValueError as ve:
        print(f"Input Error: {ve}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

Output in Server 1 Port Number 5000:



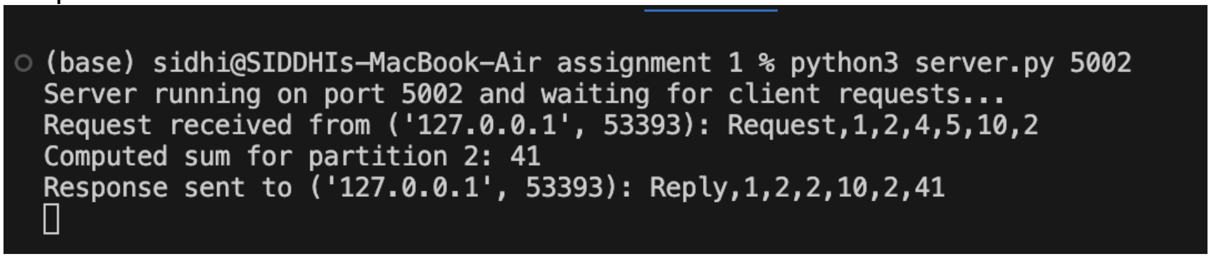
```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5000
Server running on port 5000 and waiting for client requests...
Request received from ('127.0.0.1', 52437): Request,1,0,0,1,10,2
Computed sum for partition 0: 1
Response sent to ('127.0.0.1', 52437): Reply,1,0,2,10,2,1
```

Output in Server 1 Port Number 5001:



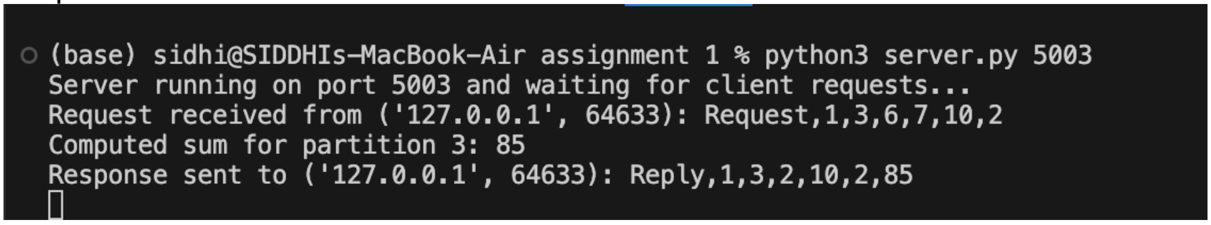
```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5001
Server running on port 5001 and waiting for client requests...
Request received from ('127.0.0.1', 64502): Request,1,1,2,3,10,2
Computed sum for partition 1: 13
Response sent to ('127.0.0.1', 64502): Reply,1,1,2,10,2,13
```

Output in Server 1 Port Number 5002:



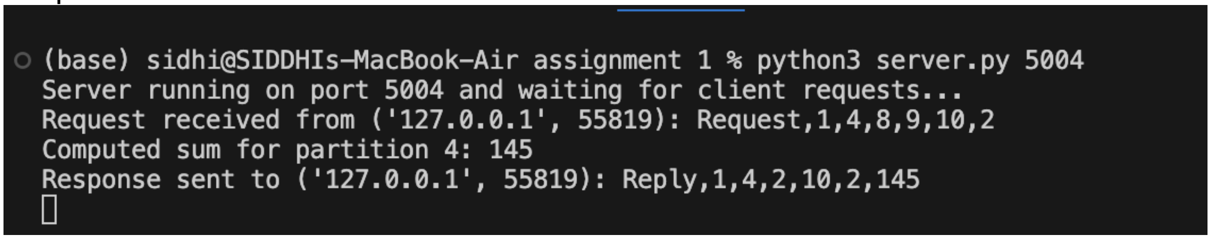
```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5002
Server running on port 5002 and waiting for client requests...
Request received from ('127.0.0.1', 53393): Request,1,2,4,5,10,2
Computed sum for partition 2: 41
Response sent to ('127.0.0.1', 53393): Reply,1,2,2,10,2,41
```

Output in Server 1 Port Number 5003:



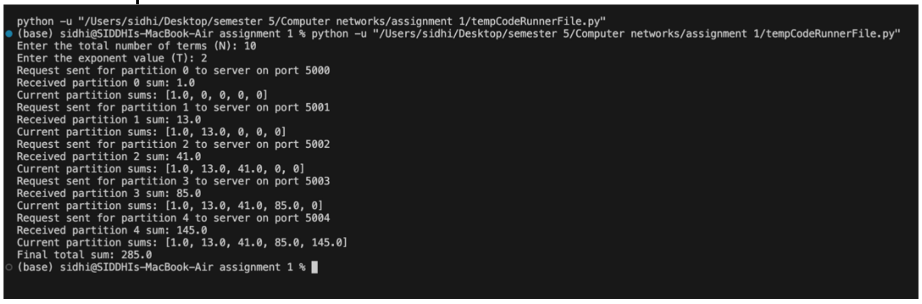
```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5003
Server running on port 5003 and waiting for client requests...
Request received from ('127.0.0.1', 64633): Request,1,3,6,7,10,2
Computed sum for partition 3: 85
Response sent to ('127.0.0.1', 64633): Reply,1,3,2,10,2,85
```

Output in Server 1 Port Number 5004:



```
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python3 server.py 5004
Server running on port 5004 and waiting for client requests...
Request received from ('127.0.0.1', 55819): Request,1,4,8,9,10,2
Computed sum for partition 4: 145
Response sent to ('127.0.0.1', 55819): Reply,1,4,2,10,2,145
```

Final Output:



```
python -u "/Users/sidhi/Desktop/semester 5/Computer networks/assignment 1/tempCodeRunnerFile.py"
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 % python -u "/Users/sidhi/Desktop/semester 5/Computer networks/assignment 1/tempCodeRunnerFile.py"
Enter the total number of terms (N): 10
Enter the exponent value (T): 2
Request sent for partition 0 to server on port 5000
Received partition 0 sum: 1.0
Current partition sums: [1.0, 0, 0, 0, 0]
Request sent for partition 1 to server on port 5001
Received partition 1 sum: 13.0
Current partition sums: [1.0, 13.0, 0, 0, 0]
Request sent for partition 2 to server on port 5002
Received partition 2 sum: 41.0
Current partition sums: [1.0, 13.0, 41.0, 0, 0]
Request sent for partition 3 to server on port 5003
Received partition 3 sum: 85.0
Current partition sums: [1.0, 13.0, 41.0, 85.0, 0]
Request sent for partition 4 to server on port 5004
Received partition 4 sum: 145.0
Current partition sums: [1.0, 13.0, 41.0, 85.0, 145.0]
Final total sum: 285.0
(base) sidhi@SIDDHIs-MacBook-Air assignment 1 %
```