charts

September 7, 2024

[13]:
```
import pandas as pd
import matplotlib.pyplot as plt

def drowApiTimeAndCpu(market_log, market_cpu):
  # Konwersja kolumny 'timestamp' na typ datetime z określonym formatem
  market_log['timestamp'] = pd.to_datetime(market_log['timestamp'],␣
 ↪format='ISO8601')
  market_cpu['timestamp'] = pd.to_datetime(market_cpu['timestamp'],␣
 ↪format='ISO8601')

  # Łączenie danych na podstawie 'timestamp'
  # Łączenie danych na podstawie 'timestamp'
  merged_data = pd.merge_asof(market_log.sort_values('timestamp'),
                              market_cpu.sort_values('timestamp'),
                              on='timestamp')

  # Podział na żądania GET i POST
  get_requests = merged_data[merged_data['apiMethod'] == 'GET']
  post_requests = merged_data[merged_data['apiMethod'] == 'POST']

  # Wykres dla żądań GET
  fig, ax1 = plt.subplots(figsize=(12, 6))

  ax1.set_xlabel('Time')
  ax1.set_ylabel('API Response Time (ms)', color='blue')
  ax1.plot(get_requests['timestamp'], get_requests['apiTime'], color='blue',␣
 ↪label='API Response Time (ms)')
  ax1.tick_params(axis='y', labelcolor='blue')

  ax2 = ax1.twinx()
  ax2.set_ylabel('CPU Usage (%)', color='red')
  ax2.plot(get_requests['timestamp'], get_requests['cpuUsage'], color='red',␣
 ↪label='CPU Usage (%)', alpha=0.7)
  ax2.tick_params(axis='y', labelcolor='red')

  plt.title('GET Requests: API Response Time and CPU Usage Over Time')
  fig.tight_layout()
```

```
plt.show()

# Wykres dla żądań POST
fig, ax1 = plt.subplots(figsize=(12, 6))

ax1.set_xlabel('Time')
ax1.set_ylabel('API Response Time (ms)', color='blue')
ax1.plot(post_requests['timestamp'], post_requests['apiTime'], color='blue',␣
↪label='API Response Time (ms)')
ax1.tick_params(axis='y', labelcolor='blue')

ax2 = ax1.twinx()
ax2.set_ylabel('CPU Usage (%)', color='red')
ax2.plot(post_requests['timestamp'], post_requests['cpuUsage'], color='red',␣
↪label='CPU Usage (%)', alpha=0.7)
ax2.tick_params(axis='y', labelcolor='red')

plt.title('POST Requests: API Response Time and CPU Usage Over Time')
fig.tight_layout()
plt.show()
```

[14]:
```
import pandas as pd
import matplotlib.pyplot as plt

def drawApplicationAndDatabaseTime(market_log, market_cpu):
    # Konwersja kolumny 'timestamp' na typ datetime
    market_log['timestamp'] = pd.to_datetime(market_log['timestamp'],␣
↪format='ISO8601')
    market_cpu['timestamp'] = pd.to_datetime(market_cpu['timestamp'],␣
↪format='ISO8601')

    # Resampling co 15 sekund, obliczanie średnich tylko dla kolumn numerycznych
    market_log_resampled = market_log.resample('5S', on='timestamp').
↪mean(numeric_only=True).reset_index()
    market_cpu_resampled = market_cpu.resample('5S', on='timestamp').
↪mean(numeric_only=True).reset_index()

    # Usuwanie wierszy z NaN po połączeniu danych
    merged_data = pd.merge_asof(market_log_resampled.sort_values('timestamp'),
                                market_cpu_resampled.sort_values('timestamp'),
                                on='timestamp').dropna()


    # Sprawdzenie, czy w danych są wartości do narysowania
    if merged_data.empty:
        print("Brak danych do narysowania wykresu.")
        return
```

```
    # Wykres dla uśrednionego applicationTime w stosunku do cpuUsage
    fig, ax1 = plt.subplots(figsize=(12, 6))

    ax1.set_xlabel('Time')
    ax1.set_ylabel('Time (ms)', color='blue')
    ax1.plot(merged_data['timestamp'], merged_data['applicationTime'],␣
 ↪color='blue', label='Average Application Time (ms)')
    ax1.tick_params(axis='y', labelcolor='blue')

    ax2 = ax1.twinx()
    ax2.set_ylabel('CPU Usage (%)', color='red')
    ax2.plot(merged_data['timestamp'], merged_data['cpuUsage'], color='red',␣
 ↪label='Average CPU Usage (%)', alpha=0.7)
    ax2.tick_params(axis='y', labelcolor='red')

    plt.title('Average Application Time and CPU Usage Over Time (15-second␣
 ↪intervals)')
    fig.tight_layout()
    ax1.legend(loc='upper left')
    ax2.legend(loc='upper right')
    plt.show()
```

[15]:
```
import pandas as pd
import matplotlib.pyplot as plt


def drawOffersAndCpu(trade_log, trade_cpu):
  # Konwersja kolumny 'timestamp' na typ datetime
  trade_log['timestamp'] = pd.to_datetime(trade_log['timestamp'],␣
 ↪format='ISO8601')
  trade_cpu['timestamp'] = pd.to_datetime(trade_cpu['timestamp'],␣
 ↪format='ISO8601')

  # Sumowanie liczby przetworzonych ofert (kupna + sprzedaży)
  trade_log['totalOffers'] = trade_log['numberOfSellOffers'] +␣
 ↪trade_log['numberOfBuyOffers']

  # Sprawdzenie, czy kolumna 'replicaId' istnieje w obu DataFrame'ach
  if 'replicaId' in trade_log.columns:
      # Łączenie danych na podstawie 'timestamp' i 'replicaId'
      merged_data = pd.merge_asof(trade_log.sort_values('timestamp'),
                                  trade_cpu.sort_values('timestamp'),
                                  on='timestamp',
                                  by='replicaId')
  else:
      # Łączenie danych tylko na podstawie 'timestamp'
```

```
    merged_data = pd.merge_asof(trade_log.sort_values('timestamp'),
                                trade_cpu.sort_values('timestamp'),
                                on='timestamp')

# Podział na różne repliki (jeśli 'replicaId' istnieje)
if 'replicaId' in merged_data.columns:
    replicas = merged_data['replicaId'].unique()
else:
    replicas = [None]  # Brak replik

# Tworzenie wykresów dla każdej repliki
for replica in replicas:
    if replica is not None:
        replica_data = merged_data[merged_data['replicaId'] == replica]
        title = f'Replica {replica}: Total Processed Offers and CPU Usage␣
↪Over Time'
    else:
        replica_data = merged_data
        title = 'Total Processed Offers and CPU Usage Over Time'

    fig, ax1 = plt.subplots(figsize=(12, 6))

    ax1.set_xlabel('Time')
    ax1.set_ylabel('Total Processed Offers', color='blue')
    ax1.plot(replica_data['timestamp'], replica_data['totalOffers'],␣
↪color='blue', label='Total Processed Offers')
    ax1.tick_params(axis='y', labelcolor='blue')

    ax2 = ax1.twinx()
    ax2.set_ylabel('CPU Usage (%)', color='red')
    ax2.plot(replica_data['timestamp'], replica_data['cpuUsage'],␣
↪color='red', label='CPU Usage (%)', alpha=0.7)
    ax2.tick_params(axis='y', labelcolor='red')

    plt.title(title)
    fig.tight_layout()
    plt.show()
```

[16]:
```
import pandas as pd
import matplotlib.pyplot as plt

def drawTimesAndOffers(trade_log):
    # Konwersja kolumny 'timestamp' na typ datetime
    trade_log['timestamp'] = pd.to_datetime(trade_log['timestamp'],␣
↪format='ISO8601')

    # Sumowanie liczby przetworzonych ofert (kupna + sprzedaży)
```

```
    trade_log['totalOffers'] = trade_log['numberOfSellOffers'] +␣
↪trade_log['numberOfBuyOffers']

    fig, ax1 = plt.subplots(figsize=(12, 6))

    ax1.set_xlabel('Time')
    ax1.set_ylabel('Time (ms)', color='blue')

    # Wykres dla applicationTime
    ax1.plot(trade_log['timestamp'], trade_log['applicationTime'],␣
↪label='Application Time', color='blue')

    # Wykres dla databaseTime
    ax1.plot(trade_log['timestamp'], trade_log['databaseTime'], label='Database␣
↪Time', color='green')

    ax1.tick_params(axis='y', labelcolor='blue')

    ax2 = ax1.twinx()
    ax2.set_ylabel('Total Processed Offers', color='red')

    # Wykres dla totalOffers
    ax2.plot(trade_log['timestamp'], trade_log['totalOffers'], label='Total␣
↪Processed Offers', color='red', alpha=0.7)
    ax2.tick_params(axis='y', labelcolor='red')

    # Dodanie legendy
    ax1.legend(loc='upper left')
    ax2.legend(loc='upper right')

    plt.title('Application Time, Database Time, and Total Processed Offers Over␣
↪Time')
    fig.tight_layout()
    plt.show()
```

[17]:
```
import pandas as pd
import matplotlib.pyplot as plt

def drawTimesAndCpu(trade_log, trade_cpu):
    # Konwersja kolumny 'timestamp' na typ datetime
    trade_log['timestamp'] = pd.to_datetime(trade_log['timestamp'],␣
↪format='ISO8601')
    trade_cpu['timestamp'] = pd.to_datetime(trade_cpu['timestamp'],␣
↪format='ISO8601')

    # Łączenie danych na podstawie 'timestamp'
```

```
    merged_data = pd.merge_asof(trade_log.sort_values('timestamp'),
                                trade_cpu.sort_values('timestamp'),
                                on='timestamp')

    fig, ax1 = plt.subplots(figsize=(12, 6))

    ax1.set_xlabel('Time')
    ax1.set_ylabel('Time (ms)', color='blue')

    # Wykres dla applicationTime
    ax1.plot(merged_data['timestamp'], merged_data['applicationTime'],␣
↪label='Application Time', color='blue')

    # Wykres dla databaseTime
    ax1.plot(merged_data['timestamp'], merged_data['databaseTime'],␣
↪label='Database Time', color='green')

    ax1.tick_params(axis='y', labelcolor='blue')

    ax2 = ax1.twinx()
    ax2.set_ylabel('CPU Usage (%)', color='red')

    # Wykres dla cpuUsage
    ax2.plot(merged_data['timestamp'], merged_data['cpuUsage'], label='CPU␣
↪Usage (%)', color='red', alpha=0.7)
    ax2.tick_params(axis='y', labelcolor='red')

    # Dodanie legendy
    ax1.legend(loc='upper left')
    ax2.legend(loc='upper right')

    plt.title('Application Time, Database Time, and CPU Usage Over Time')
    fig.tight_layout()
    plt.show()
```

[18]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def drawAverageResponseTime(complete_market_log):
  # Group data by API method and endpoints, calculate average response times
  api_performance = complete_market_log.groupby(['apiMethod', 'endpointUrl']).
↪agg({
      'applicationTime': 'mean',
      'databaseTime': 'mean',
      'apiTime': 'mean'
  }).reset_index()
```

```
    # Plot response times for different API methods
    plt.figure(figsize=(14, 7))
    sns.barplot(x='apiMethod', y='applicationTime', hue='endpointUrl',␣
    ↪data=api_performance)
    plt.title('Average Response Time for Different API Methods and Endpoints')
    plt.xlabel('API Method')
    plt.ylabel('Average Response Time (ms)')
    plt.xticks(rotation=45)
    plt.legend(title='Endpoint URL')
    plt.show()
```

[19]:
```
def drawAverageCpuReplic(trade_cpu):
    # Group data by replicaId, calculate average CPU and memory usage
    replica_performance = trade_cpu.groupby('replicaId').agg({
        'cpuUsage': 'mean',
        'memoryUsage': 'mean'
    }).reset_index()

    # Plot average CPU and memory usage for different replicas
    plt.figure(figsize=(14, 7))
    sns.barplot(x='replicaId', y='cpuUsage', data=replica_performance, color='b',␣
    ↪label='CPU Usage')
    sns.barplot(x='replicaId', y='memoryUsage', data=replica_performance,␣
    ↪color='r', label='Memory Usage')
    plt.title('Average CPU and Memory Usage for Different Replicas')
    plt.xlabel('Replica ID')
    plt.ylabel('Average Usage (%)')
    plt.legend(title='Usage Type')
    plt.show()
```

[20]:
```
def trafficCpu(traffic_cpu):
    traffic_cpu['timestamp'] = pd.to_datetime(traffic_cpu['timestamp'],␣
    ↪format='ISO8601')

    # Plot trends of CPU and memory usage over time
    plt.figure(figsize=(14, 7))
    plt.plot(traffic_cpu['timestamp'], traffic_cpu['cpuUsage'], label='CPU Usage')
    plt.plot(traffic_cpu['timestamp'], traffic_cpu['memoryUsage'], label='Memory␣
    ↪Usage')
    plt.title('Trends in CPU and Memory Usage Over Time')
    plt.xlabel('Timestamp')
    plt.ylabel('Usage (%)')
    plt.legend()
    plt.xticks(rotation=45)
    plt.show()
```

[21]:
```
import pandas as pd
import matplotlib.pyplot as plt


def hisMarket(market_cpu):
  # Plot histogram for CPU usage
  plt.figure(figsize=(10, 6))
  plt.hist(market_cpu['cpuUsage'], bins=30, color='blue', edgecolor='black')
  plt.title('Distribution of CPU Usage')
  plt.xlabel('CPU Usage (%)')
  plt.ylabel('Frequency')
  plt.show()

  # Plot histogram for Memory usage
  plt.figure(figsize=(10, 6))
  plt.hist(market_cpu['memoryUsage'], bins=30, color='green', edgecolor='black')
  plt.title('Distribution of Memory Usage')
  plt.xlabel('Memory Usage (%)')
  plt.ylabel('Frequency')
  plt.show()
```

[22]:
```
def mapCpuMem(traffic_cpu):
  # Scatter plot for CPU usage vs Memory usage
  plt.figure(figsize=(10, 6))
  plt.scatter(traffic_cpu['cpuUsage'], traffic_cpu['memoryUsage'], alpha=0.5)
  plt.title('CPU Usage vs Memory Usage')
  plt.xlabel('CPU Usage (%)')
  plt.ylabel('Memory Usage (%)')
  plt.show()
```

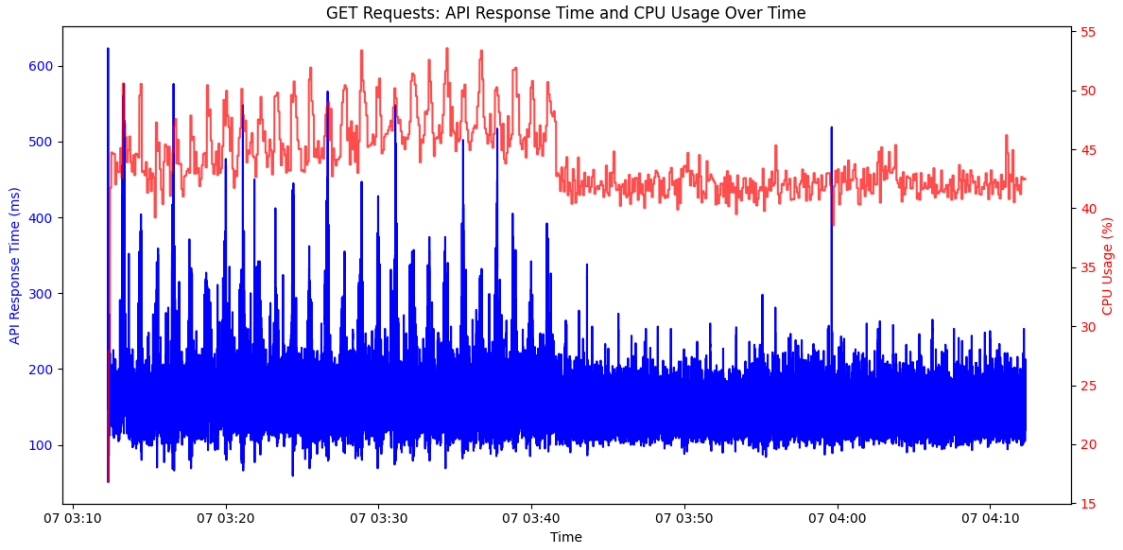
[23]:
```
import pandas as pd
import matplotlib.pyplot as plt

def offersPerMin(trade_log, market_cpu):
  # Convert timestamp to datetime in both datasets
  trade_log['timestamp'] = pd.to_datetime(trade_log['timestamp'],␣
 ↪format='ISO8601')
  market_cpu['timestamp'] = pd.to_datetime(market_cpu['timestamp'],␣
 ↪format='ISO8601')

  # Resample (aggregate) the data by minute
  trade_log_minute = trade_log.resample('T', on='timestamp').sum().reset_index()
  market_cpu_minute = market_cpu.resample('T', on='timestamp').mean().
 ↪reset_index()

  # Merge the datasets on timestamp
```

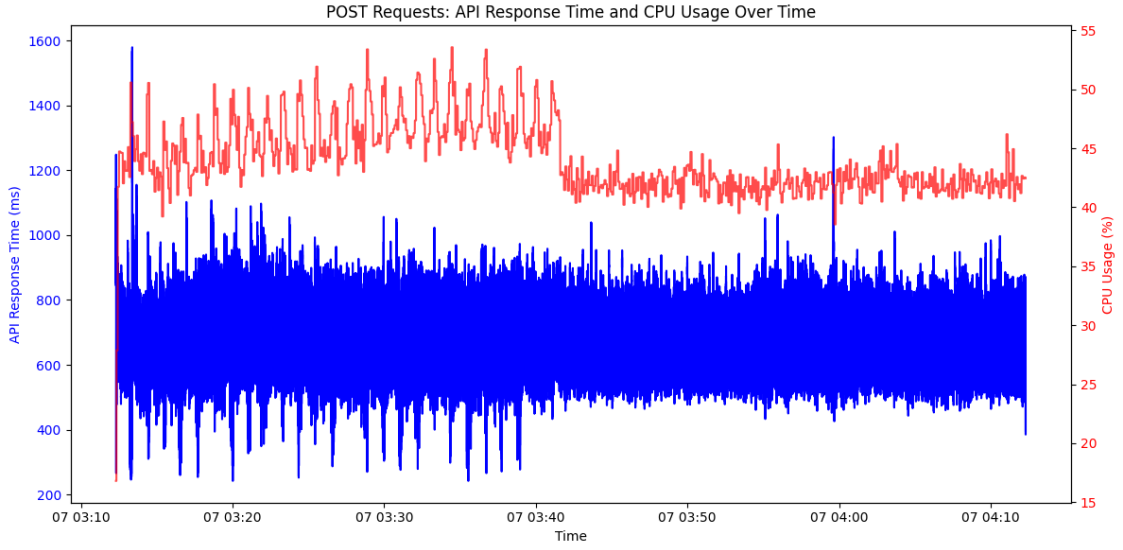
```
merged_data_minute = pd.merge(trade_log_minute, market_cpu_minute,␣
↪on='timestamp')

# Create a figure and axis
fig, ax1 = plt.subplots(figsize=(14, 7))

# Plot CPU usage on the first Y axis
ax1.set_xlabel('Timestamp')
ax1.set_ylabel('CPU Usage (%)', color='blue')
ax1.plot(merged_data_minute['timestamp'], merged_data_minute['cpuUsage'],␣
↪color='blue', label='CPU Usage')
ax1.tick_params(axis='y', labelcolor='blue')

# Create a second Y axis for the number of buy and sell offers
ax2 = ax1.twinx()
ax2.set_ylabel('Number of Offers', color='green')
ax2.plot(merged_data_minute['timestamp'],␣
↪merged_data_minute['numberOfBuyOffers'], color='orange', label='Buy Offers')
ax2.plot(merged_data_minute['timestamp'],␣
↪merged_data_minute['numberOfSellOffers'], color='red', label='Sell Offers')
ax2.tick_params(axis='y', labelcolor='green')

plt.title('CPU Usage vs Number of Buy and Sell Offers Processed Per Minute')
fig.tight_layout()  # Adjust layout
fig.legend(loc='upper left', bbox_to_anchor=(0.1,0.9))
plt.show()
```

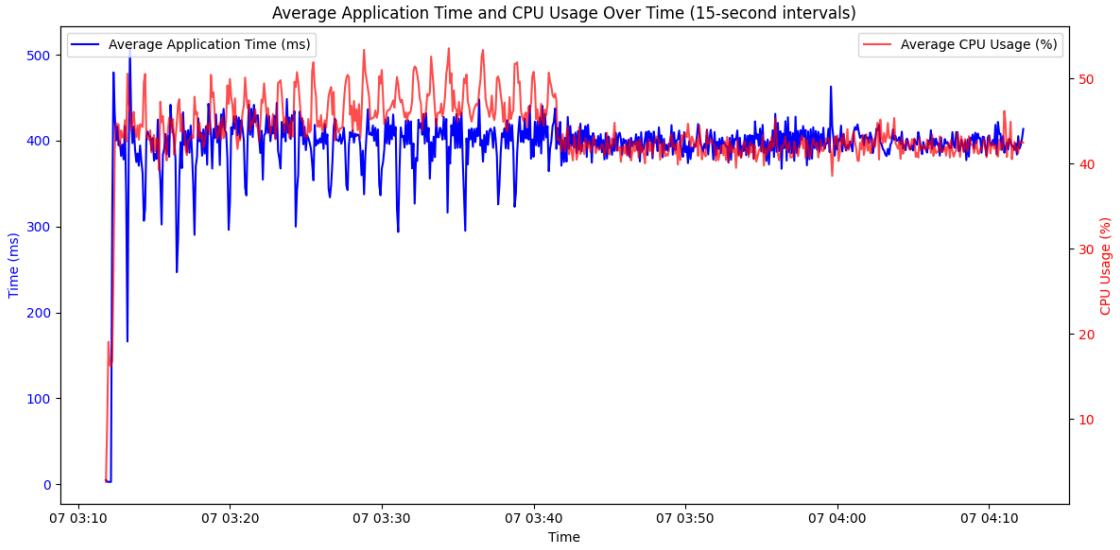
[25]:
```
import os
import pandas as pd

def drowAll():
    # Pobieramy bieżący katalog, w którym znajduje się skrypt
    current_dir = os.getcwd()

    # Iterujemy przez katalogi test1, test2, test3
    for test_dir in ['test1', 'test2', 'test3']:
        test_dir_path = os.path.join(current_dir, test_dir)

        # Przechodzimy przez wszystkie podkatalogi w test1, test2, test3
        for root, dirs, files in os.walk(test_dir_path):
            # Sprawdzamy, czy w podkatalogu znajdują się wymagane pliki CSV
            complete_market_log_csv_path = os.path.join(root,␣
↪'complete_market_log_csv.csv')
            market_cpu_path = os.path.join(root, 'market_cpu.csv')
            market_log_path = os.path.join(root, 'market_log.csv')
            traffic_cpu_path = os.path.join(root, 'traffic_cpu.csv')
            sum_trade_log_path = os.path.join(root, 'sum_trade_log.csv')
```

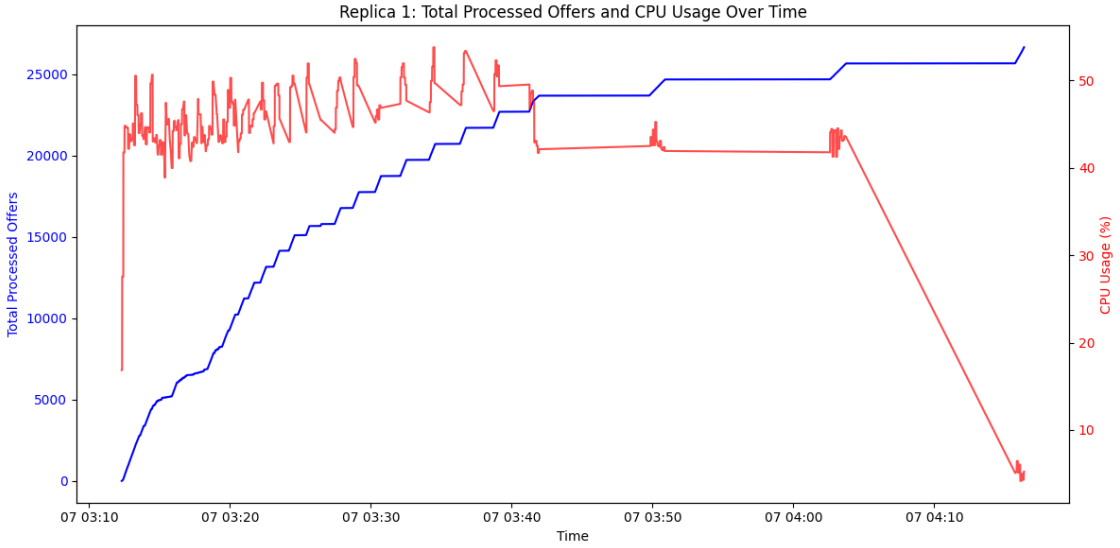
```
            trade_cpu_path = os.path.join(root, 'trade_cpu.csv')
            trade_log_path = os.path.join(root, 'trade_log.csv')

            # Sprawdzamy czy wszystkie pliki istnieją
            if all(os.path.exists(path) for path in␣
 ↪[complete_market_log_csv_path, market_cpu_path, traffic_cpu_path,␣
 ↪sum_trade_log_path, trade_cpu_path, market_log_path, trade_log_path]):
                # Wczytujemy pliki CSV do DataFrame
                complete_market_log_csv = pd.
 ↪read_csv(complete_market_log_csv_path)
                market_cpu = pd.read_csv(market_cpu_path)
                market_log = pd.read_csv(market_log_path)
                traffic_cpu = pd.read_csv(traffic_cpu_path)
                sum_trade_log = pd.read_csv(sum_trade_log_path)
                trade_cpu = pd.read_csv(trade_cpu_path)
                trade_log = pd.read_csv(trade_log_path)
                print(f"Loaded data from {root} successfully.")

                drowApiTimeAndCpu(complete_market_log_csv, market_cpu)
                drawApplicationAndDatabaseTime(market_log, market_cpu)
                drawOffersAndCpu(sum_trade_log, trade_cpu)
                drawTimesAndOffers(sum_trade_log)
                drawTimesAndCpu(trade_log, trade_cpu)
                drawAverageResponseTime(complete_market_log_csv)
                drawAverageCpuReplic(trade_cpu)
                trafficCpu(traffic_cpu)
                hisMarket(market_cpu)
                # mapCpuMem(traffic_cpu)
                offersPerMin(trade_log, market_cpu)

            else:
                print(f"One or more CSV files are missing in directory: {root}")

drowAll()
```

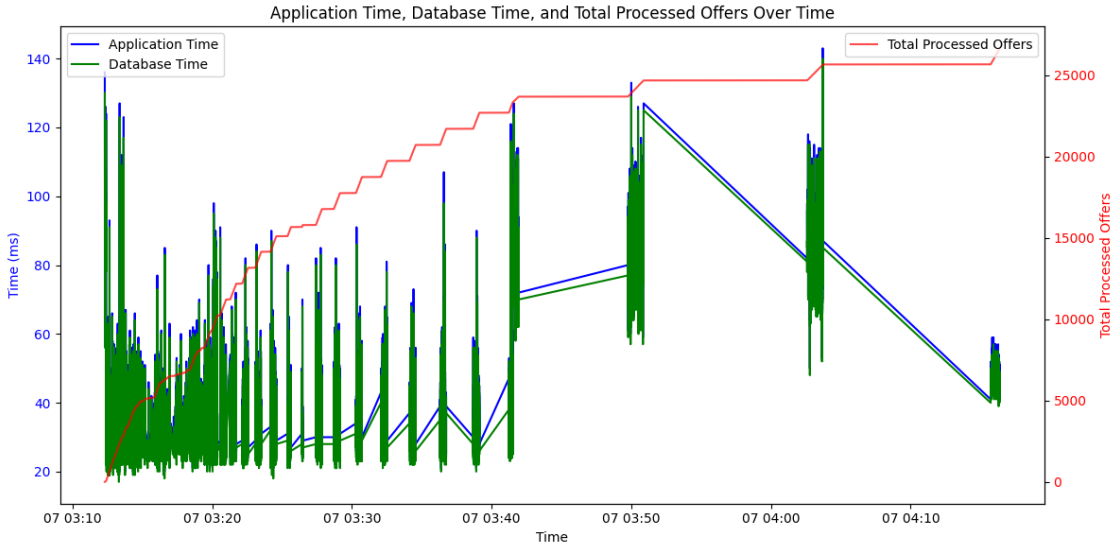
```
One or more CSV files are missing in directory:
c:\Users\luki_\Desktop\logs\test1
Loaded data from c:\Users\luki_\Desktop\logs\test1\1 10 200 500 4 500 100
successfully.
```

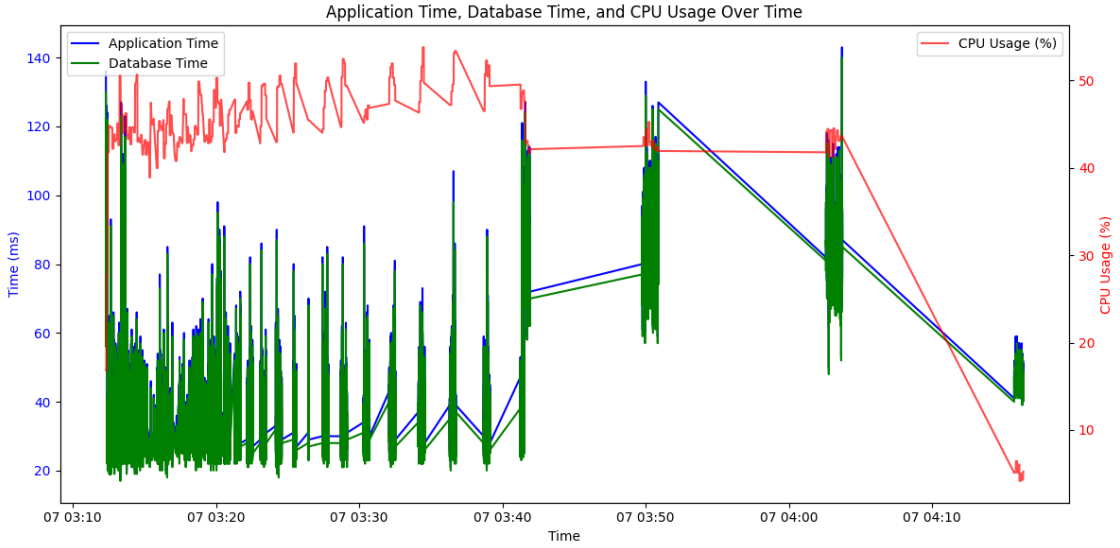

GET Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
07 03:10
07 03:20
07 03:30
07 03:40
07 03:50
07 04:00
07 04:10

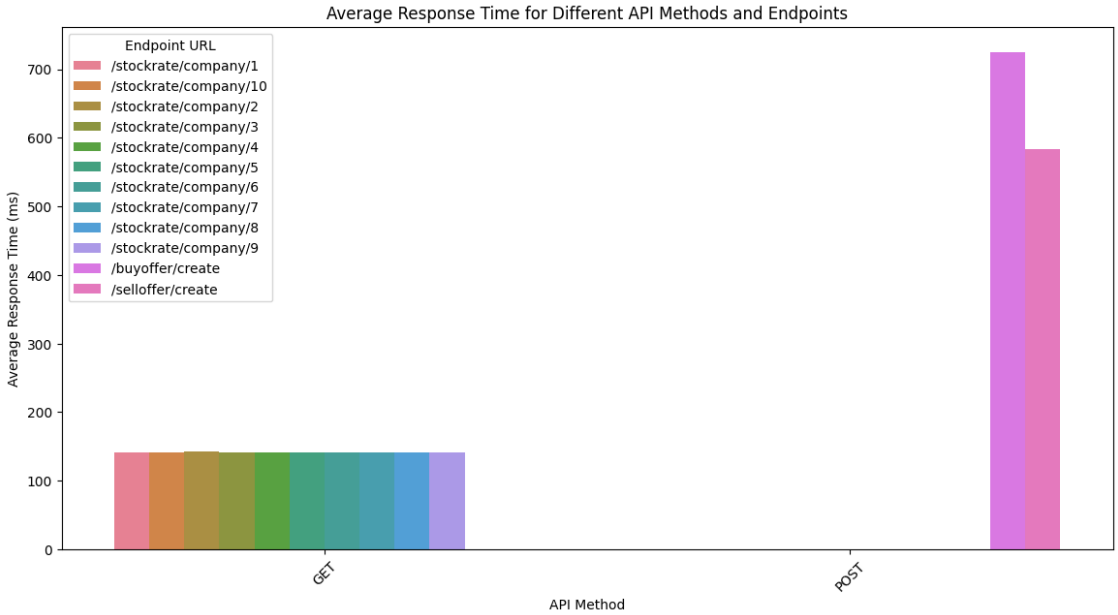

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
07 03:10
07 03:20
07 03:30
07 03:40
07 03:50
07 04:00
07 04:10

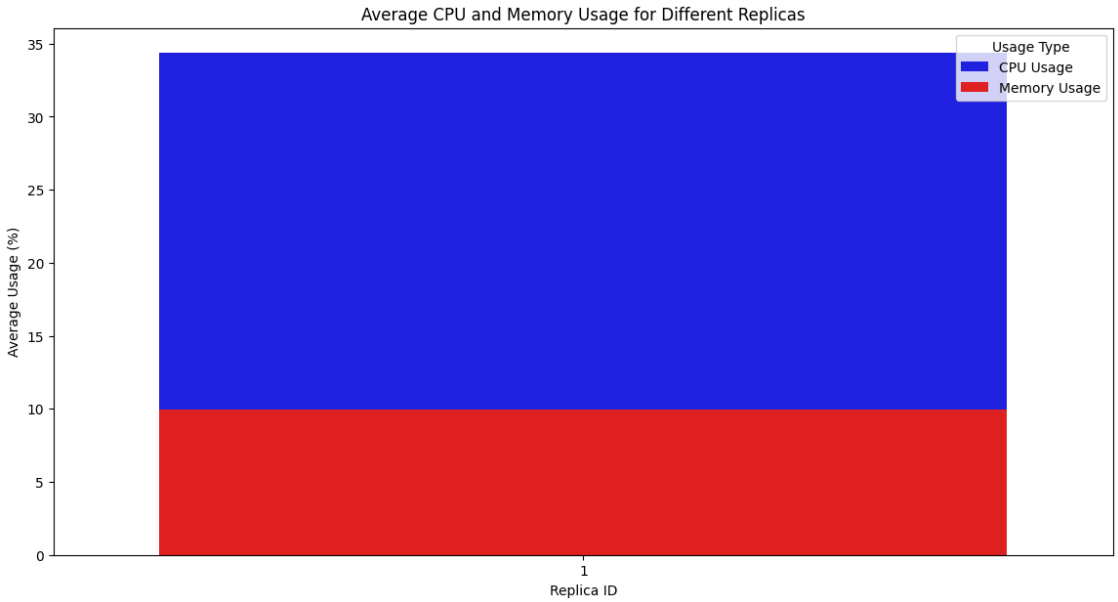

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
0
100
200
300
400
500
10
20
30
40
50
07 03:10
07 03:20
07 03:30
07 03:40
07 03:50
07 04:00
07 04:10
Time

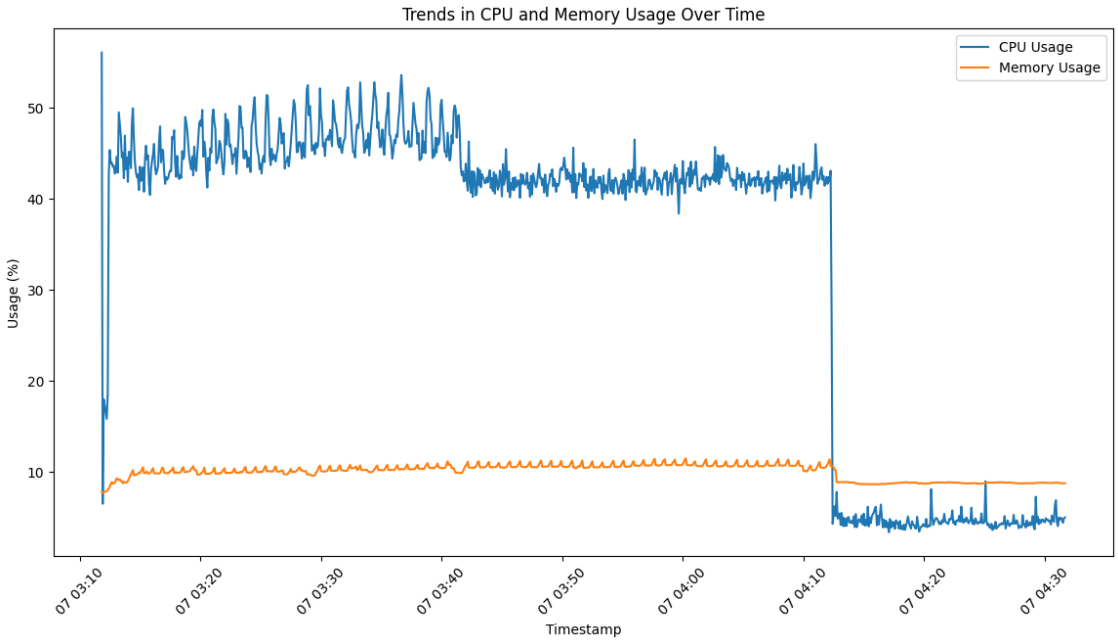

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
0
5000
10000
15000
20000
25000
10
20
30
40
50
07 03:10
07 03:20
07 03:30
07 03:40
07 03:50
07 04:00
07 04:10
Time

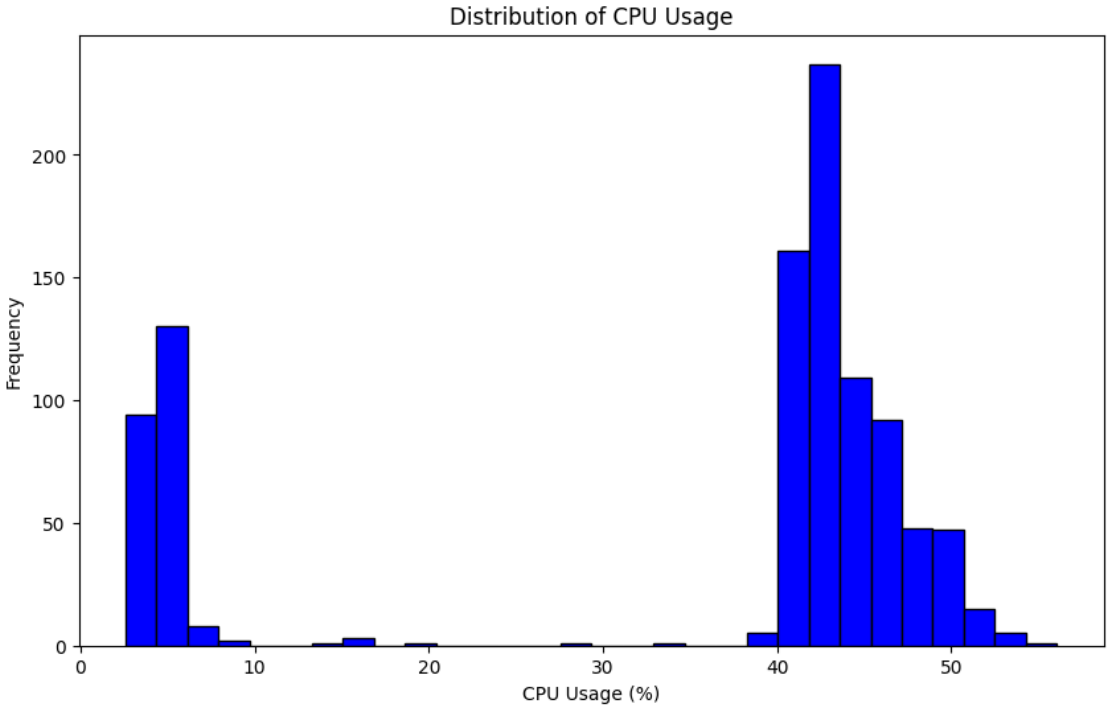

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
07 03:10
07 03:20
07 03:30
07 03:40
07 03:50
07 04:00
07 04:10

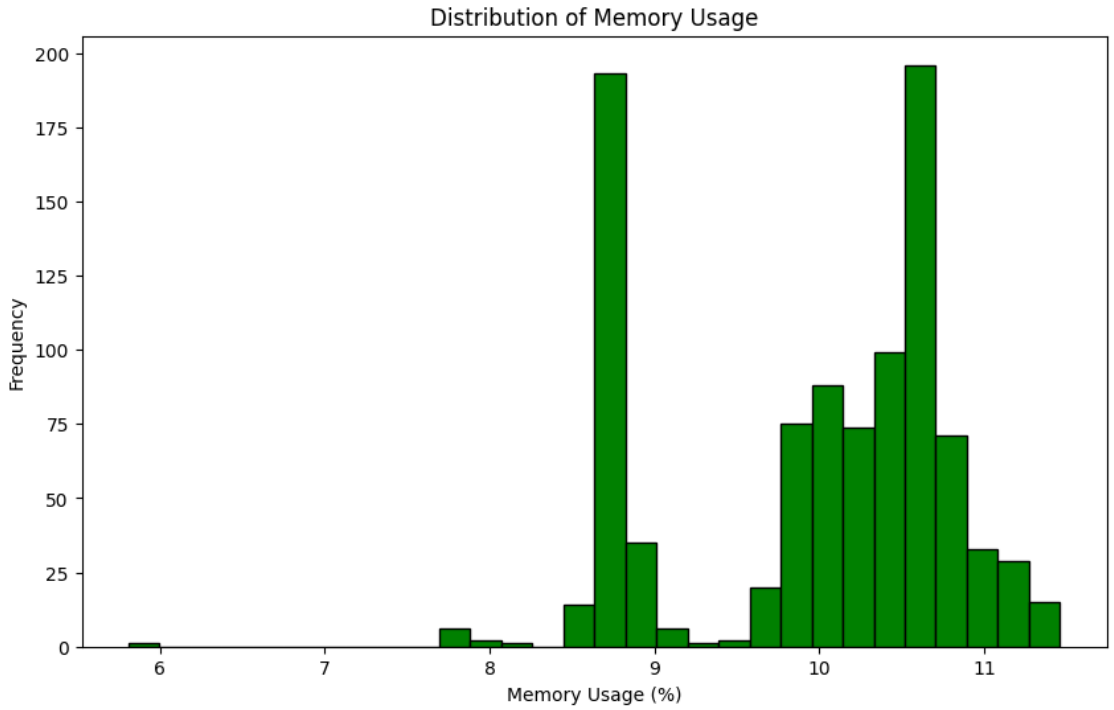

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
07 03:10
07 03:20
07 03:30
07 03:40
07 03:50
07 04:00
07 04:10

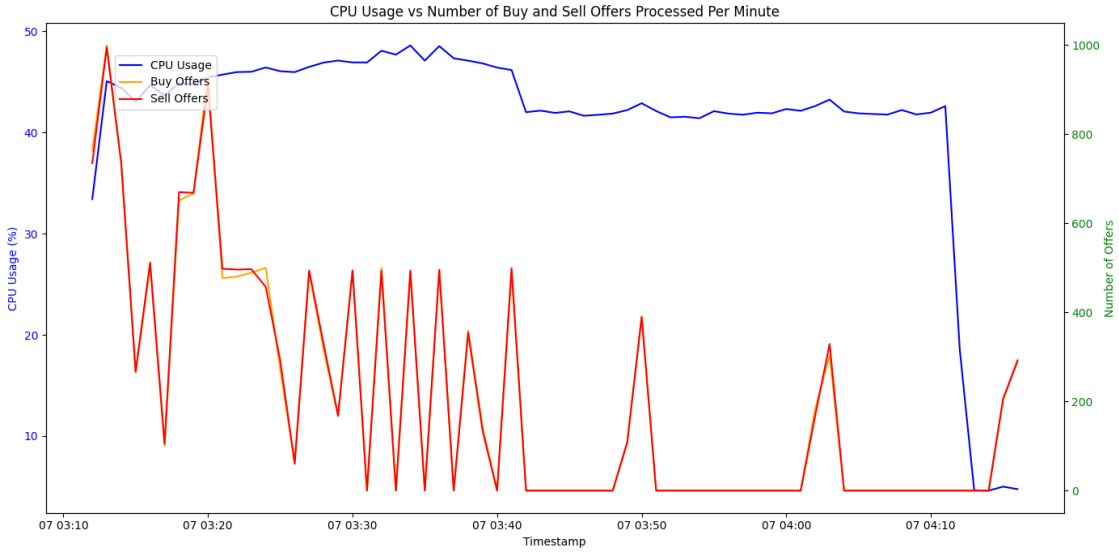

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
100
200
300
400
500
600
700
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
5
10
15
20
25
30
35
1
Replica ID

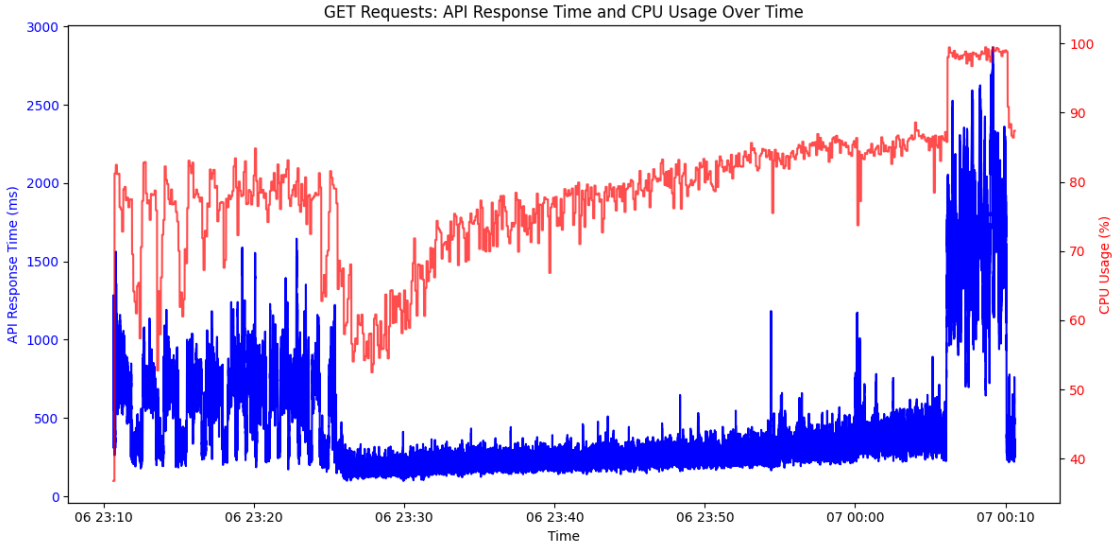

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
50
40
30
20
10
07 03:10
07 03:20
07 03:30
07 03:40
07 03:50
07 04:00
07 04:10
07 04:20
07 04:30
Timestamp

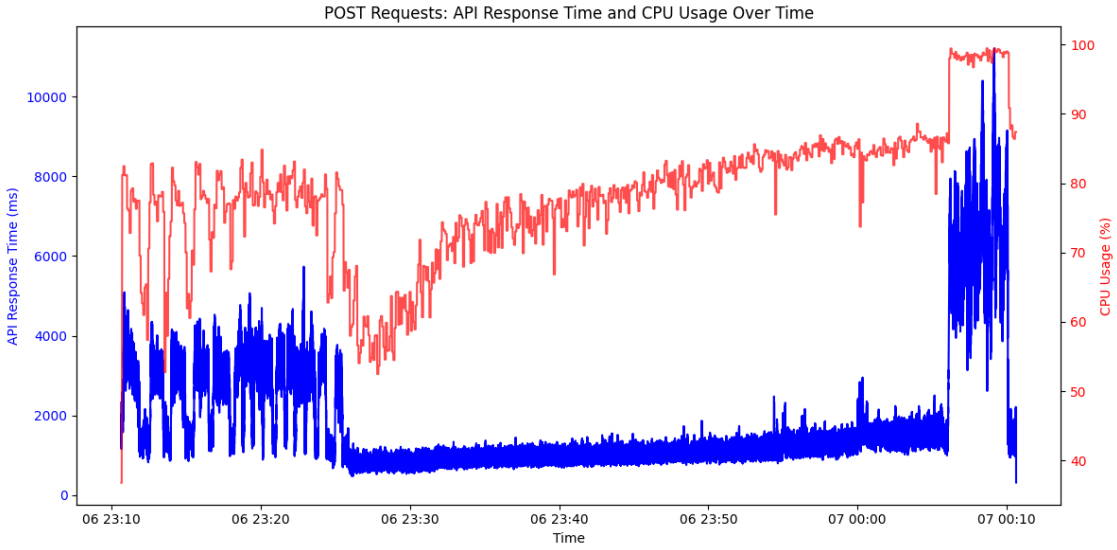

Distribution of CPU Usage
Frequency
200
150
100
50
0
0
10
20
30
40
50
CPU Usage (%)

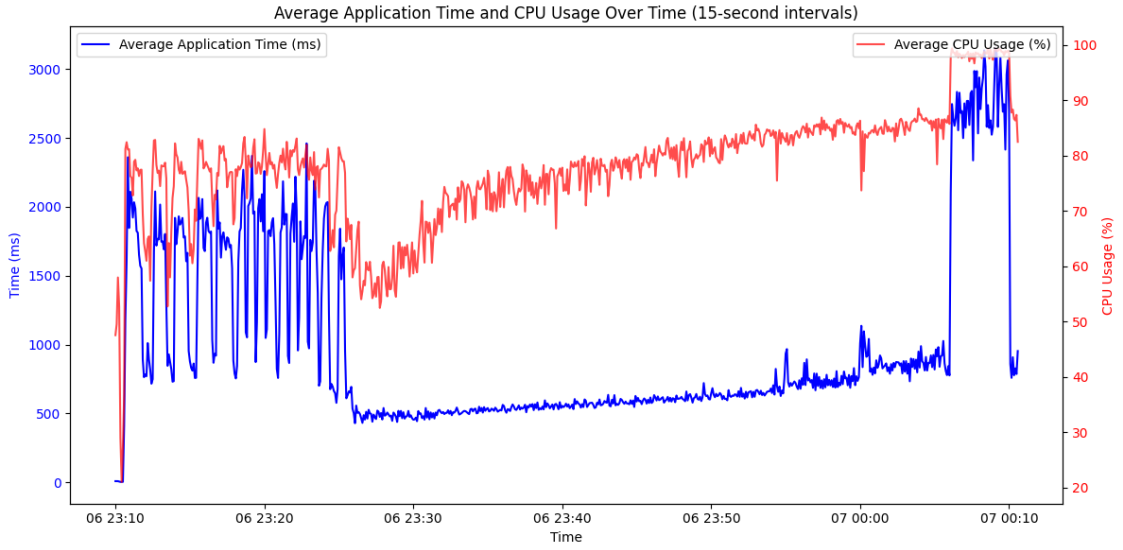


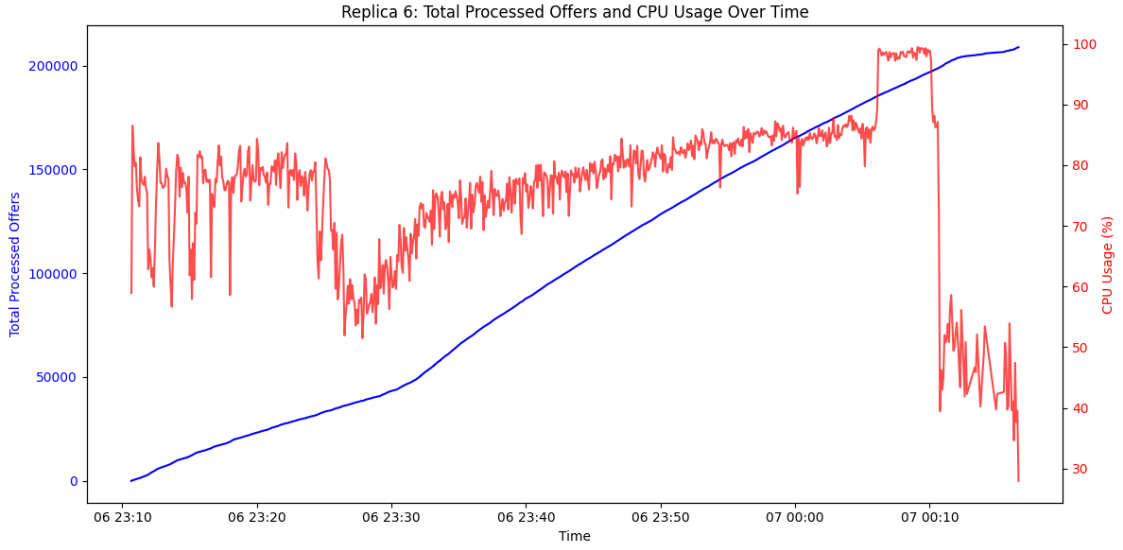


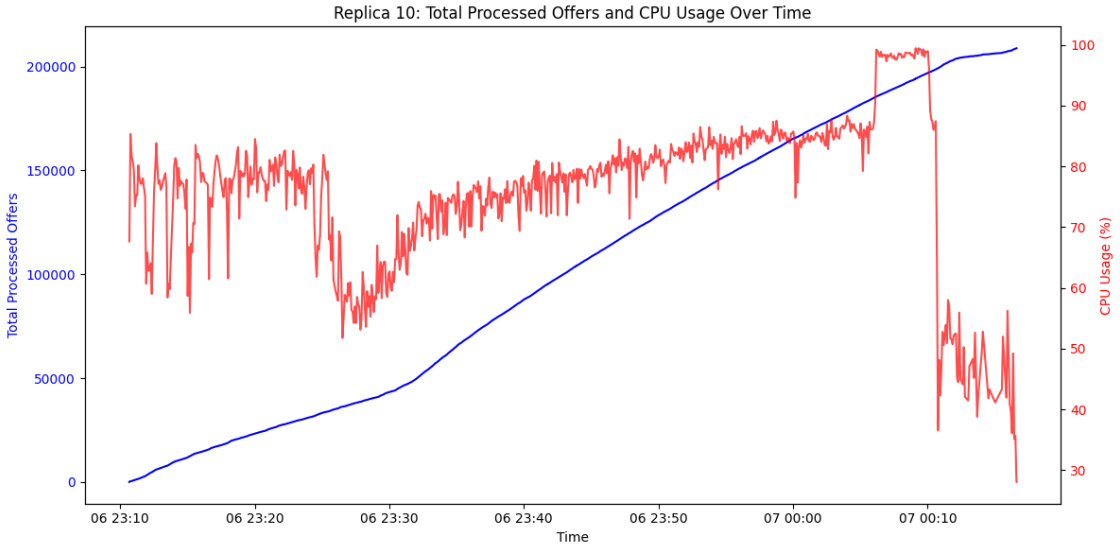

```
Loaded data from c:\Users\luki_\Desktop\logs\test1\10 1 200 500 4 500 100
successfully.
```

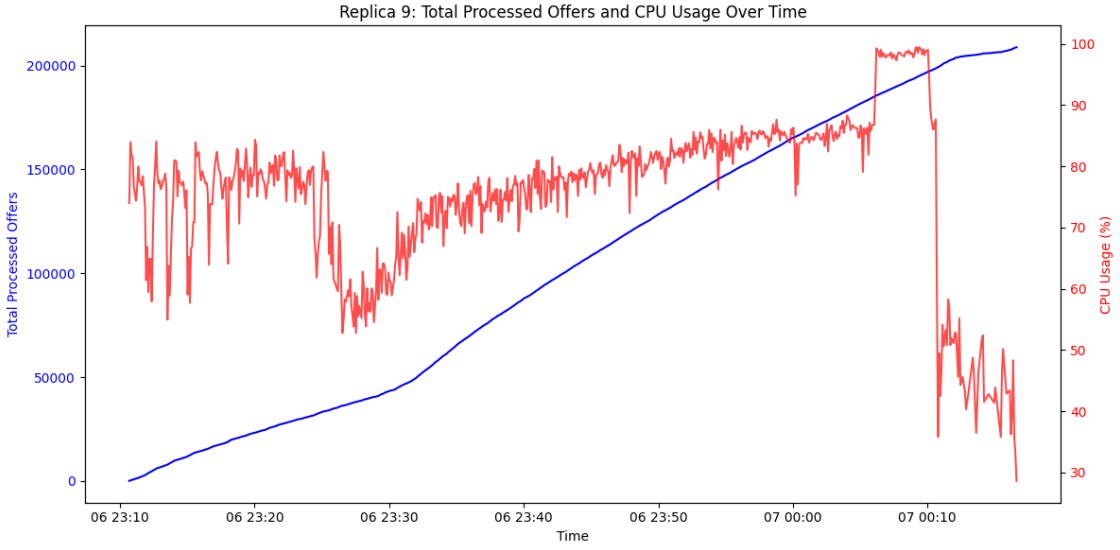

GET Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
500
1000
1500
2000
2500
3000
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

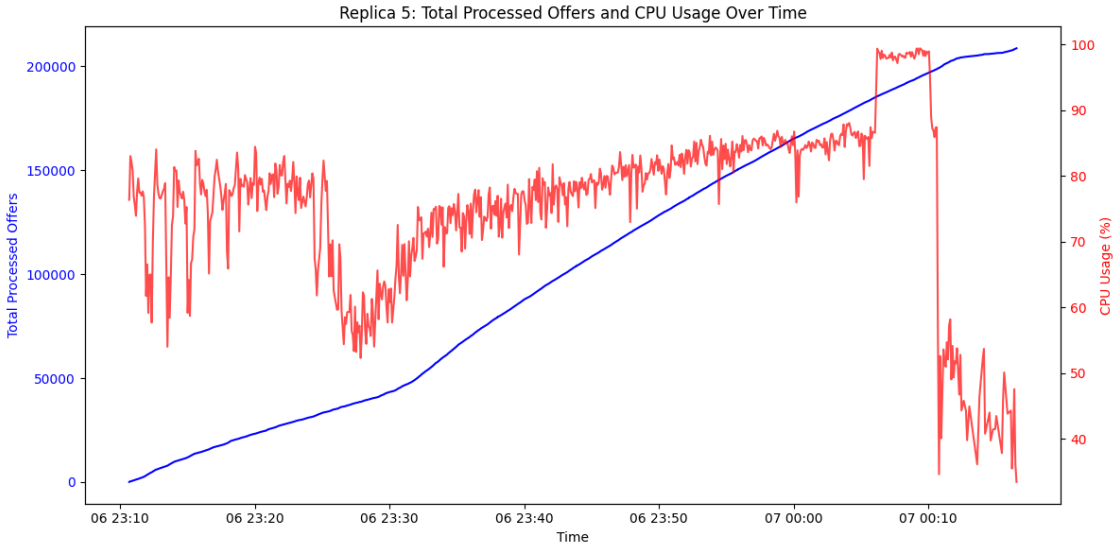

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
2000
4000
6000
8000
10000
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

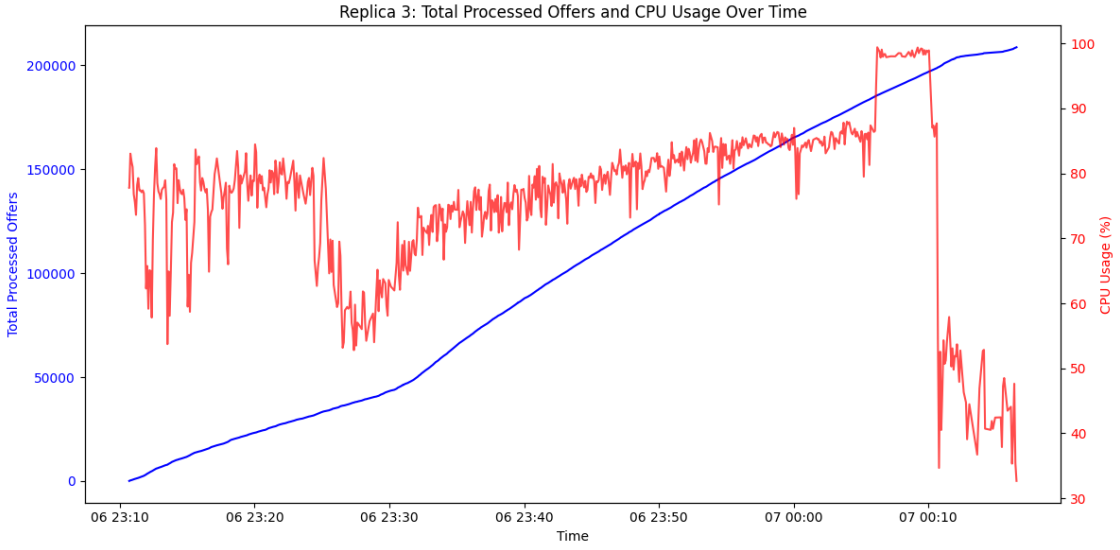

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

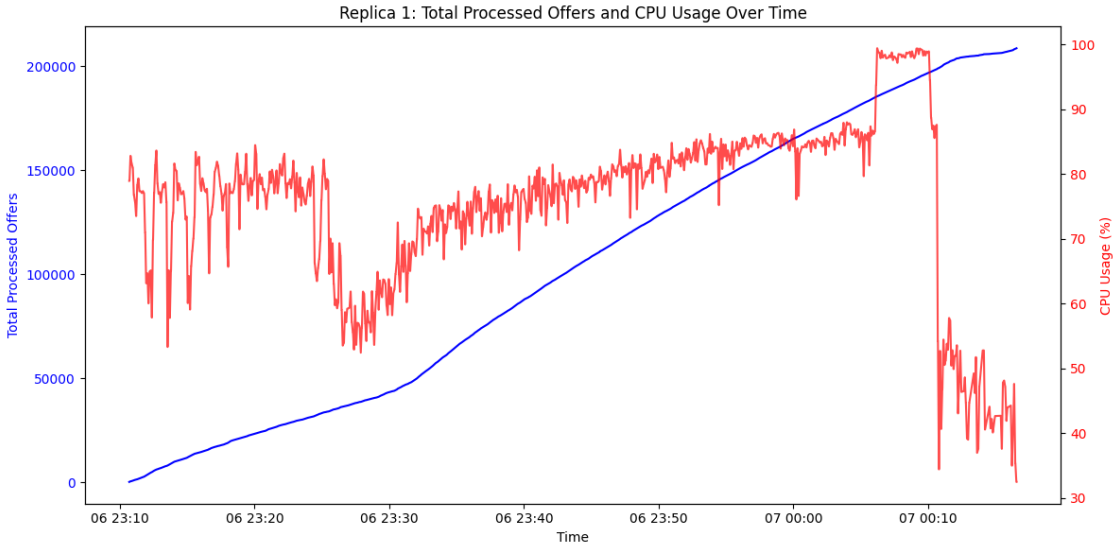

Replica 6: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

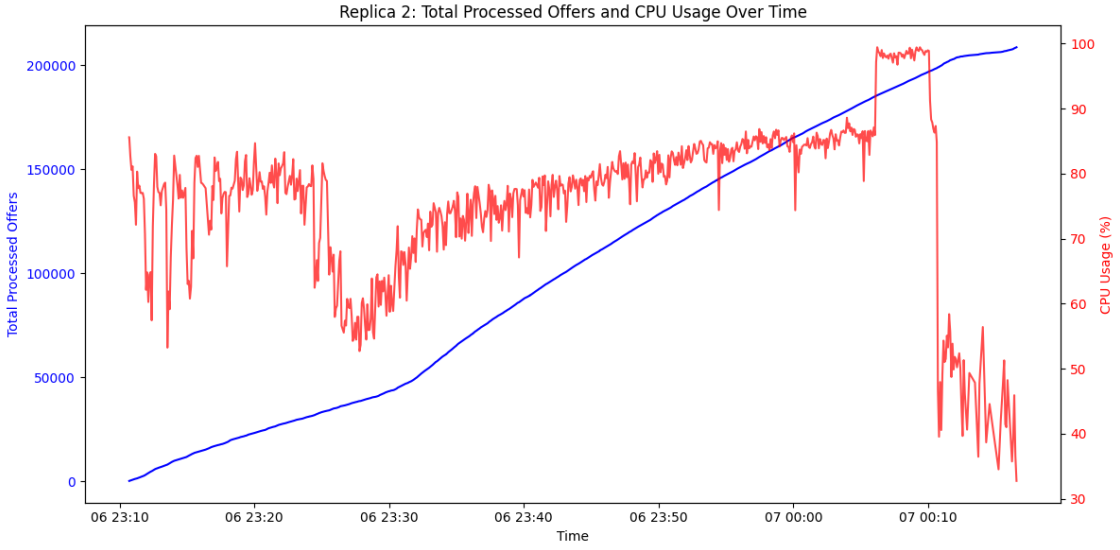

Replica 10: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

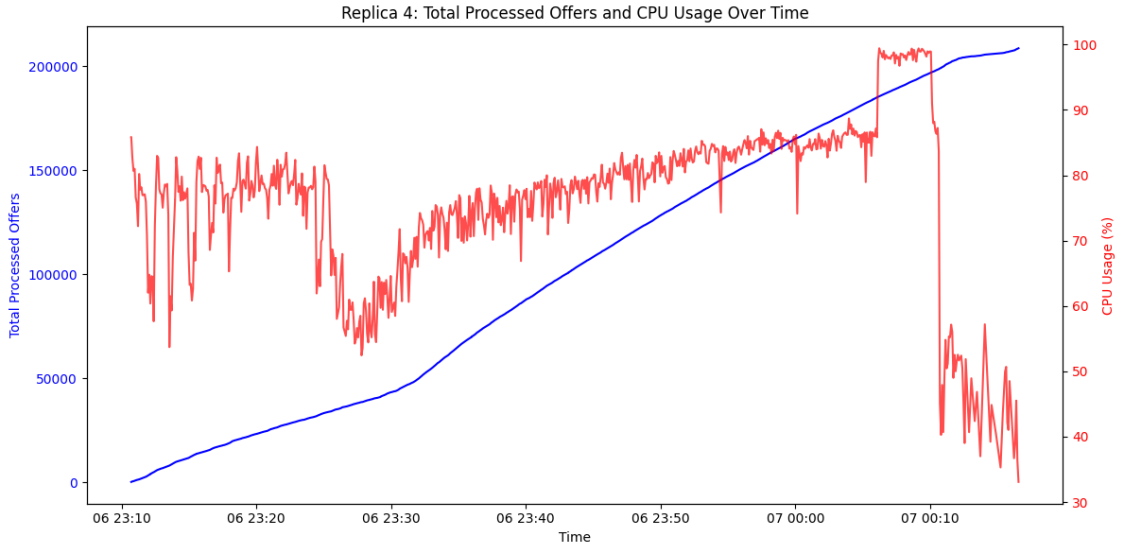

Replica 9: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

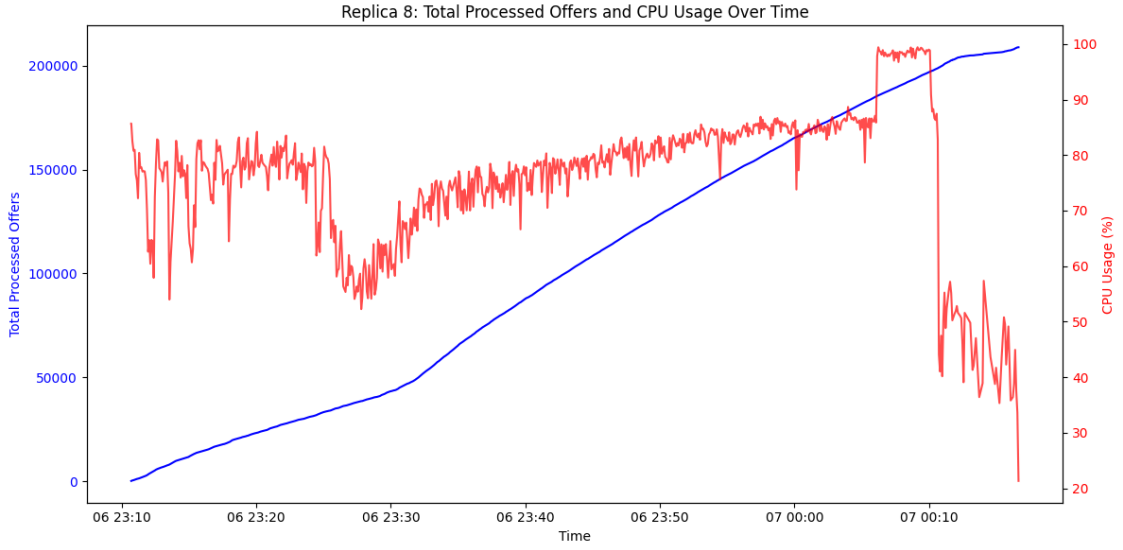

Replica 5: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

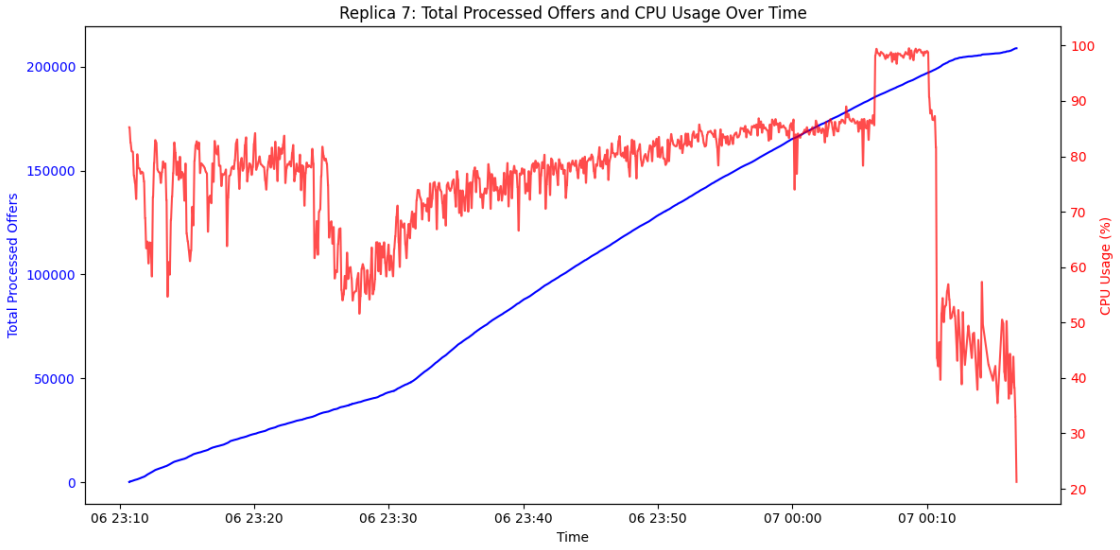

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

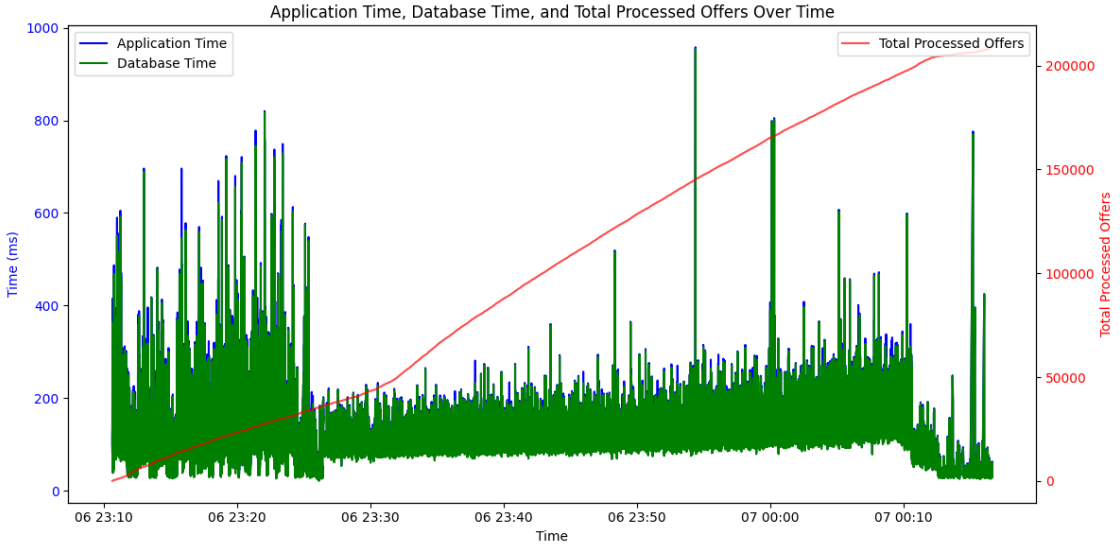

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

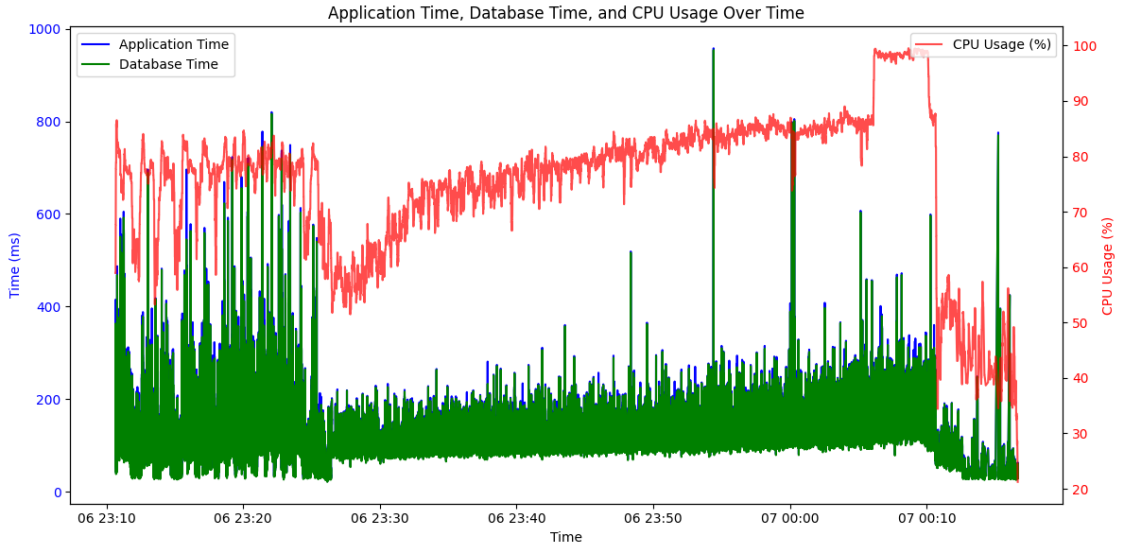

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

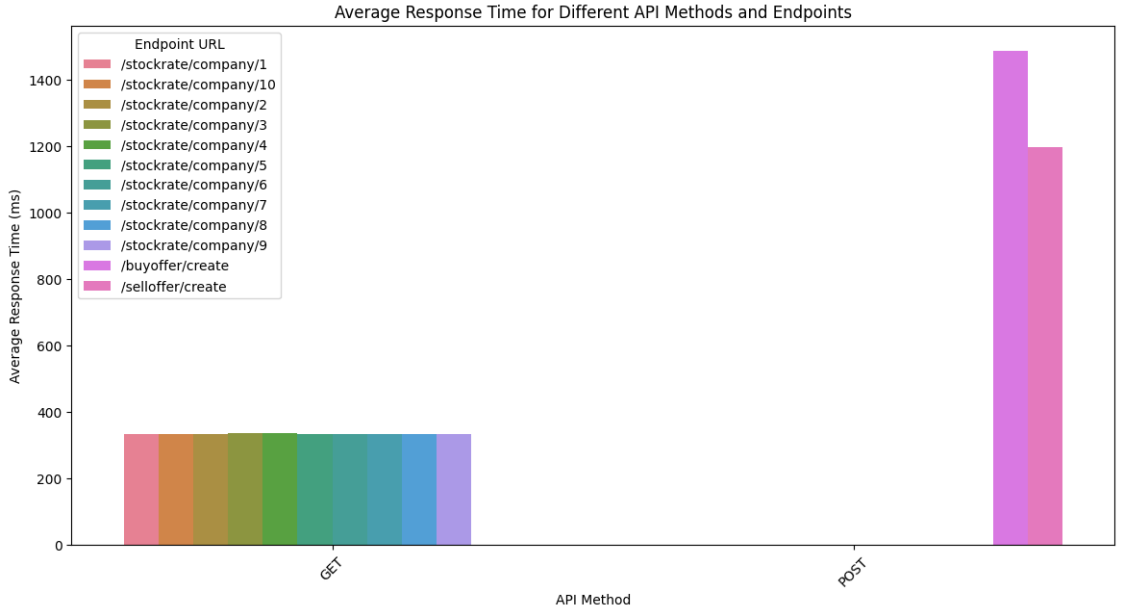

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

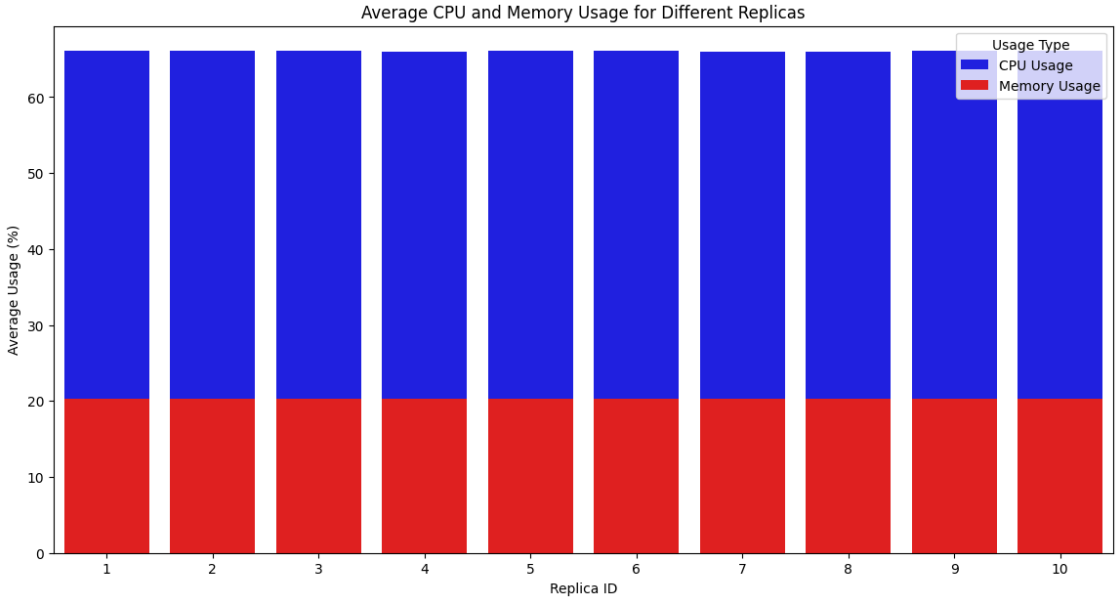

Replica 8: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

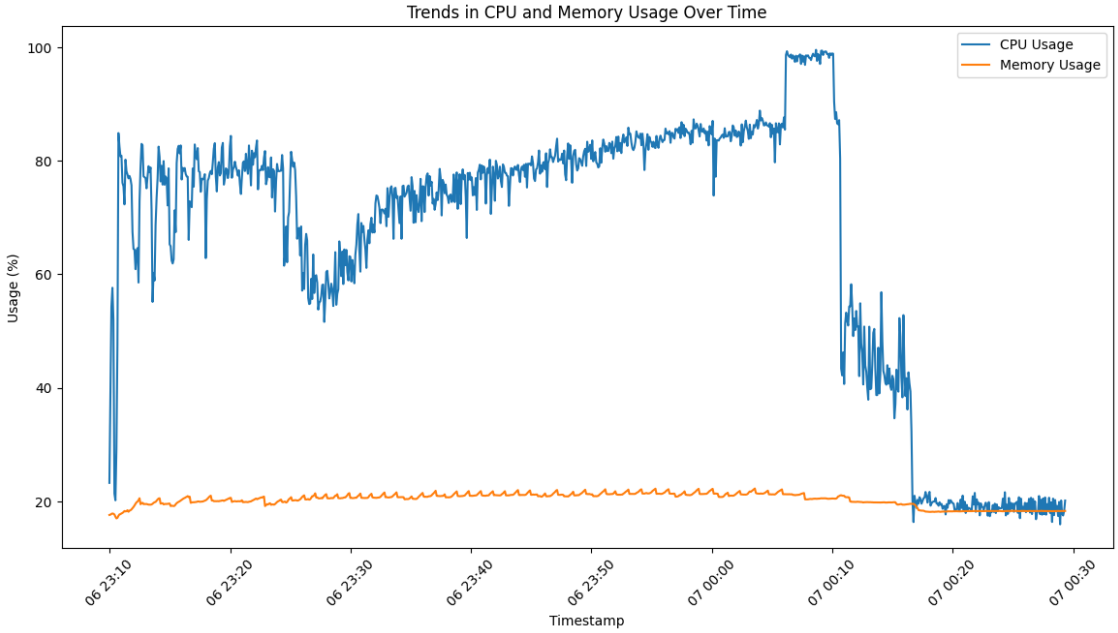

Replica 7: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

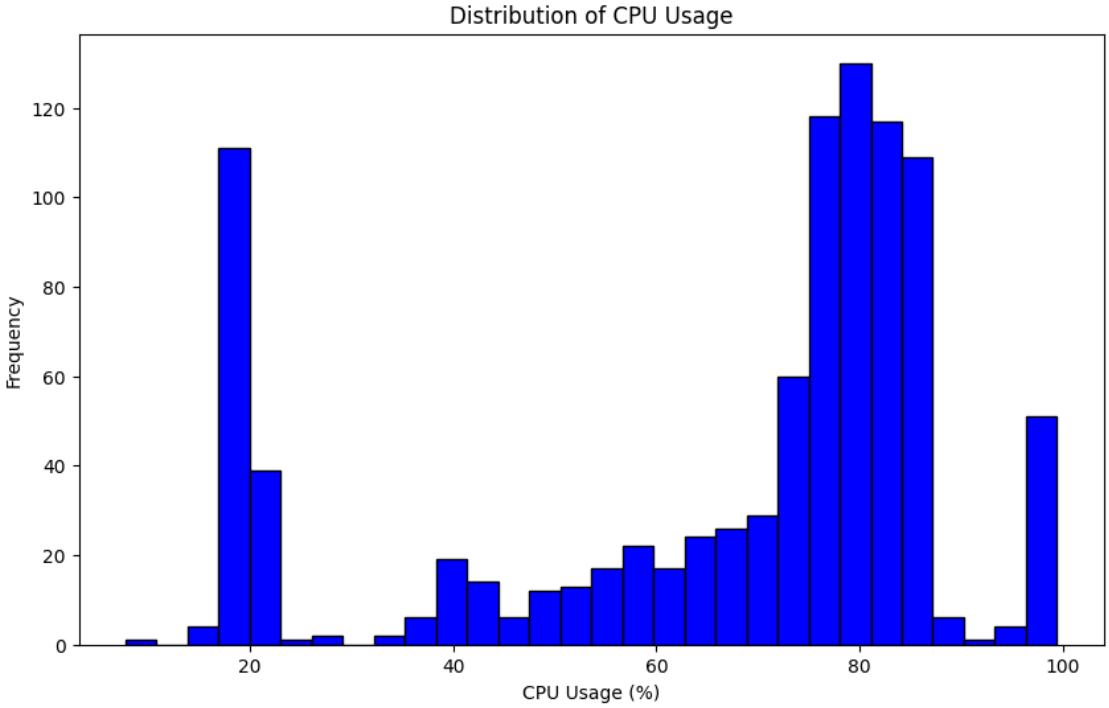

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
0
200
400
600
800
1000
0
50000
100000
150000
200000
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

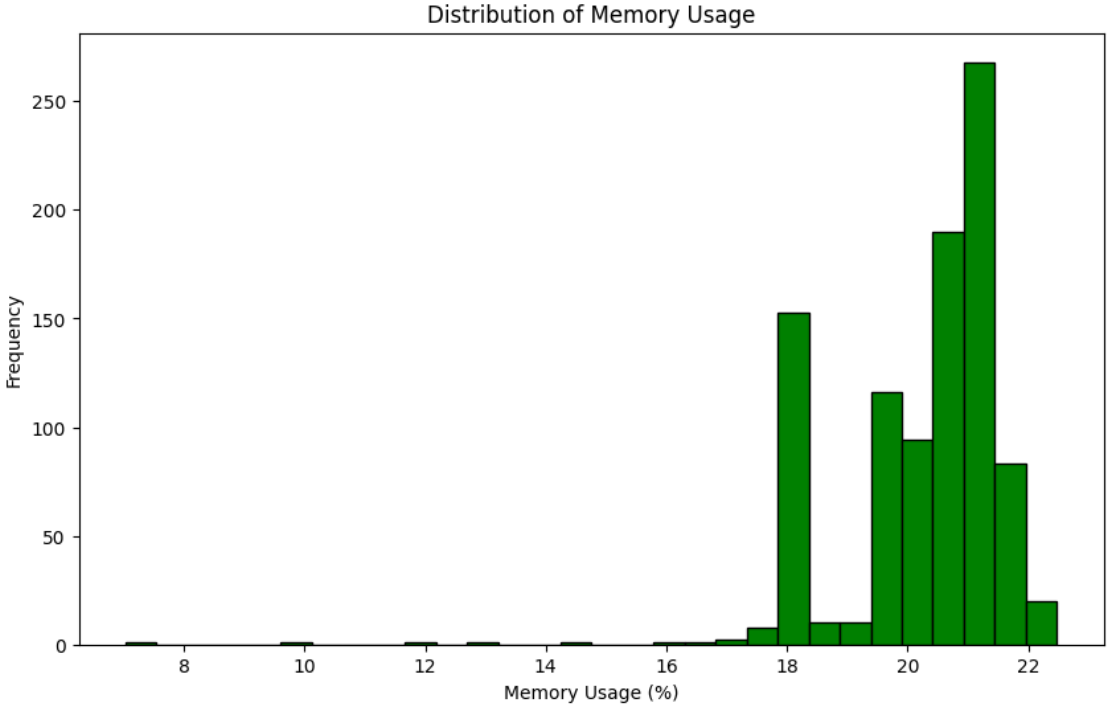

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
200
400
600
800
1000
20
30
40
50
60
70
80
90
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10

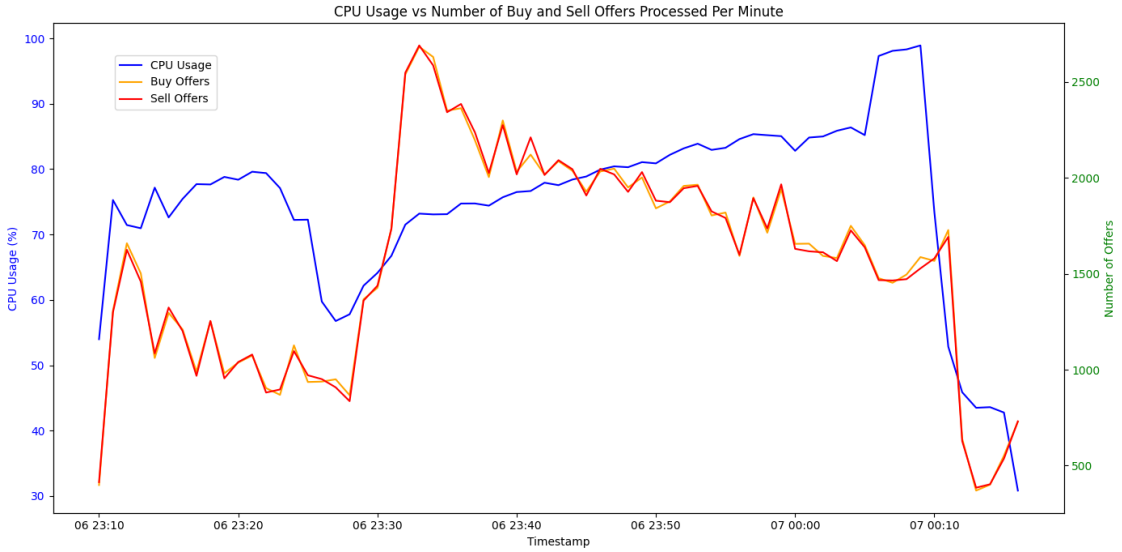

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
1400
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
Replica ID
0
10
20
30
40
50
60
1
2
3
4
5
6
7
8
9
10

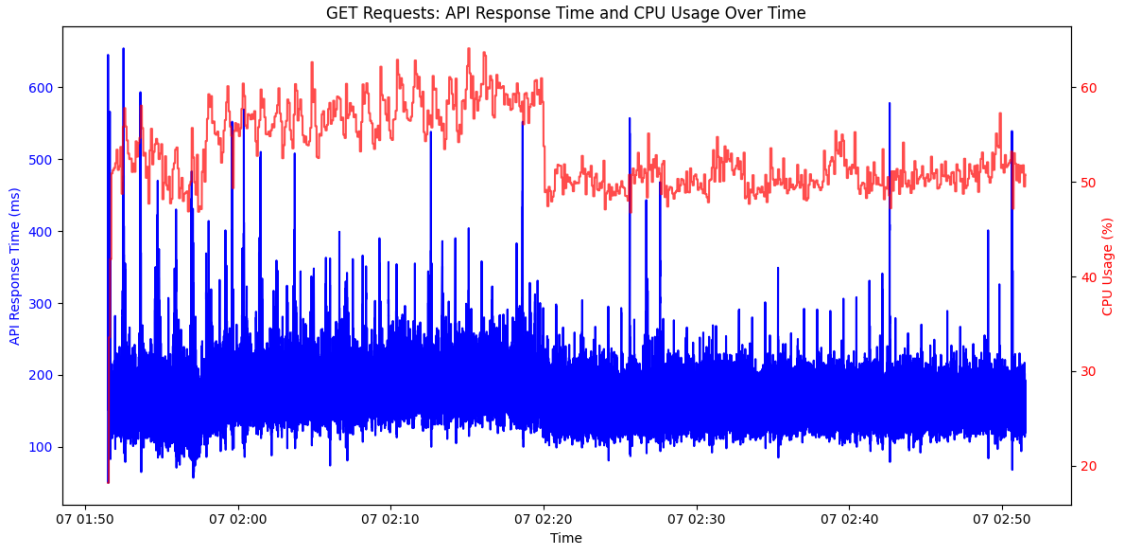

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
Timestamp
20
40
60
80
100
06 23:10
06 23:20
06 23:30
06 23:40
06 23:50
07 00:00
07 00:10
07 00:20
07 00:30

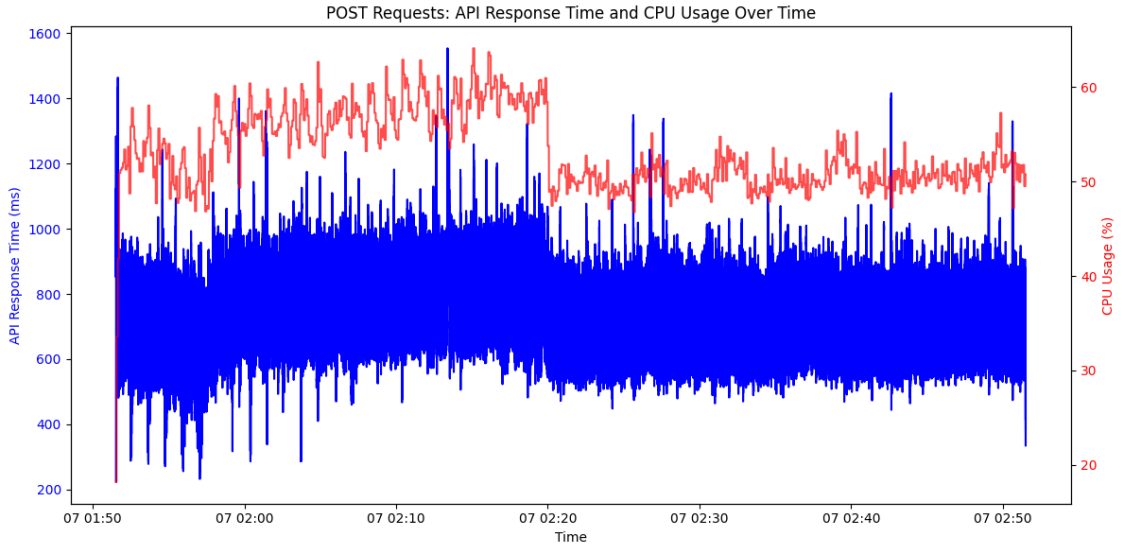

Distribution of CPU Usage
Frequency
CPU Usage (%)
0
20
40
60
80
100
120

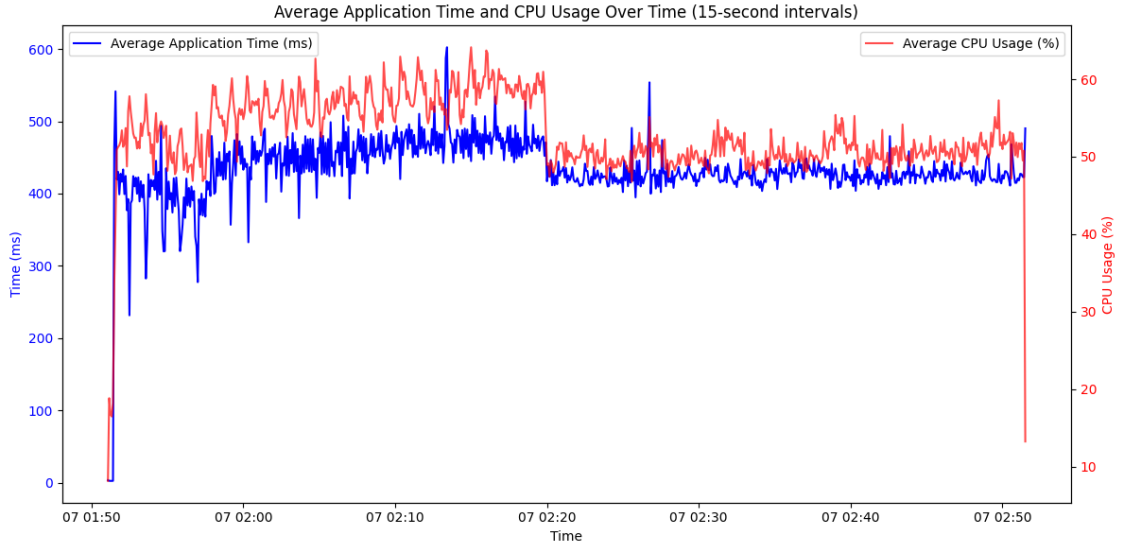

Distribution of Memory Usage
Frequency
Memory Usage (%)
0
50
100
150
200
250
8
10
12
14
16
18
20
22

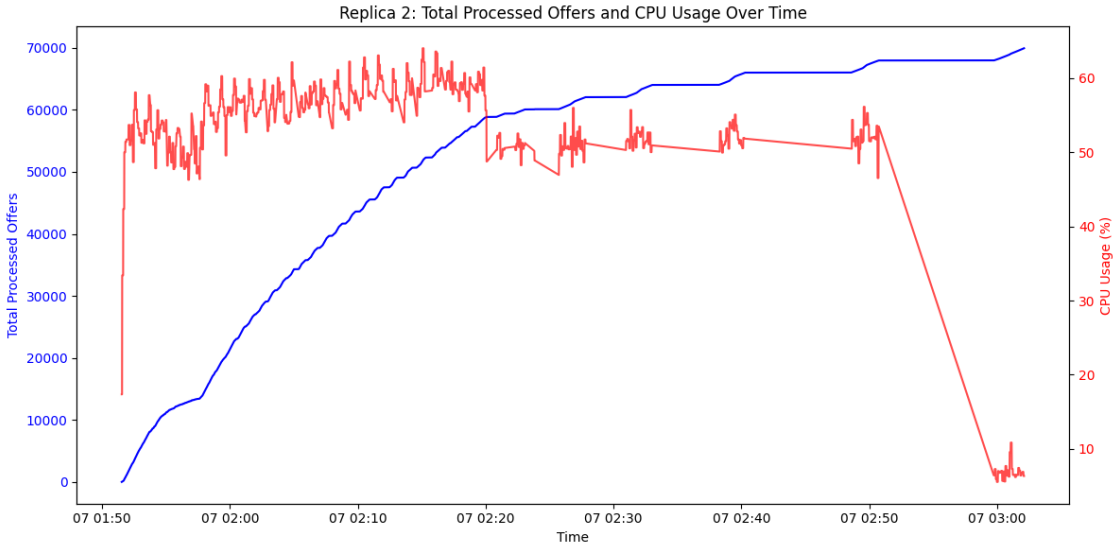


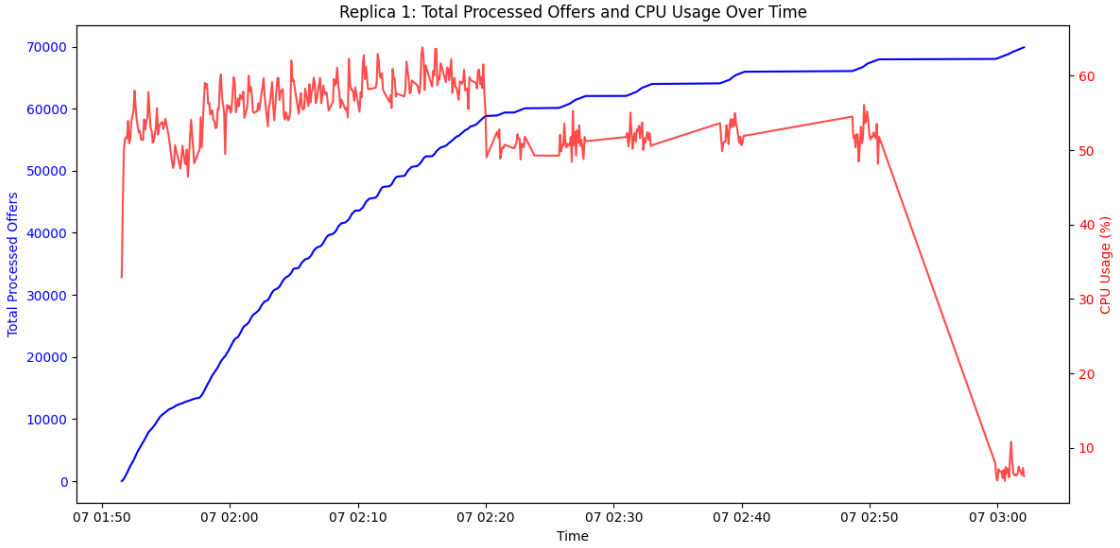

```
Loaded data from c:\Users\luki_\Desktop\logs\test1\2 5 200 500 4 500 100
successfully.
```

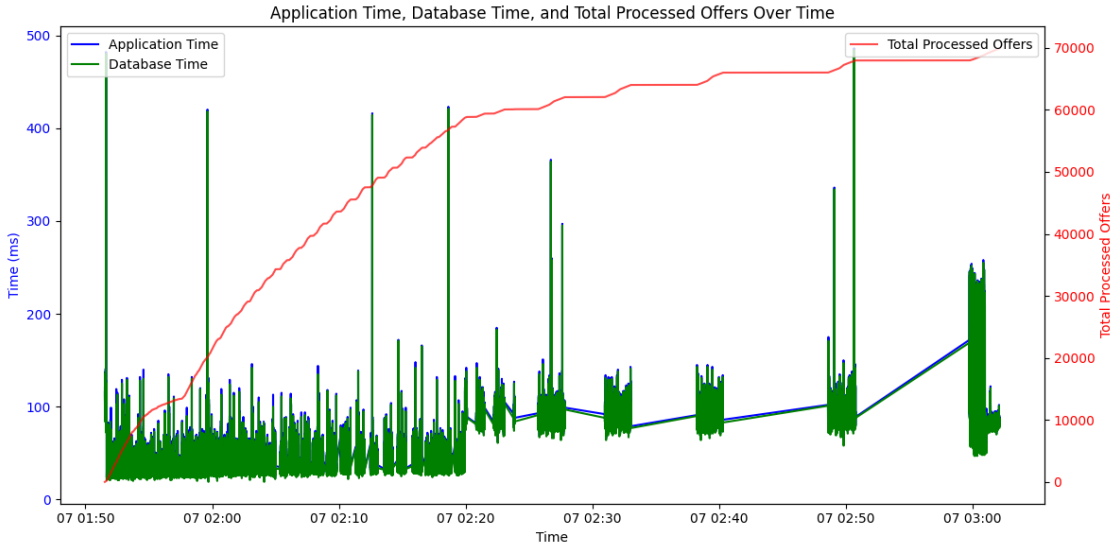


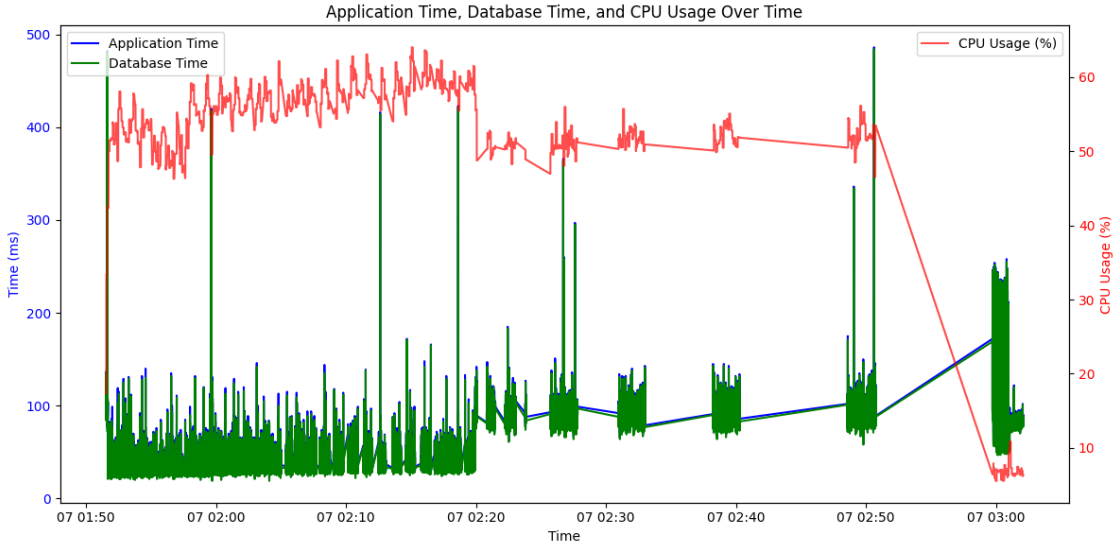

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
1600
1400
1200
1000
800
600
400
200
60
50
40
30
20
07 01:50
07 02:00
07 02:10
07 02:20
07 02:30
07 02:40
07 02:50

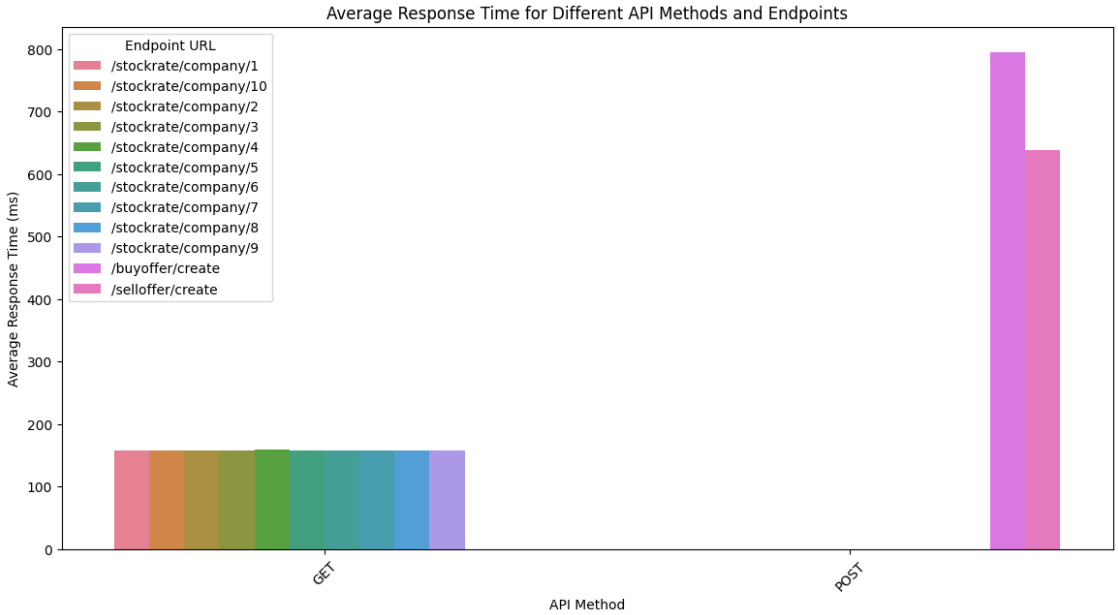

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
600
500
400
300
200
100
0
60
50
40
30
20
10
07 01:50
07 02:00
07 02:10
07 02:20
07 02:30
07 02:40
07 02:50

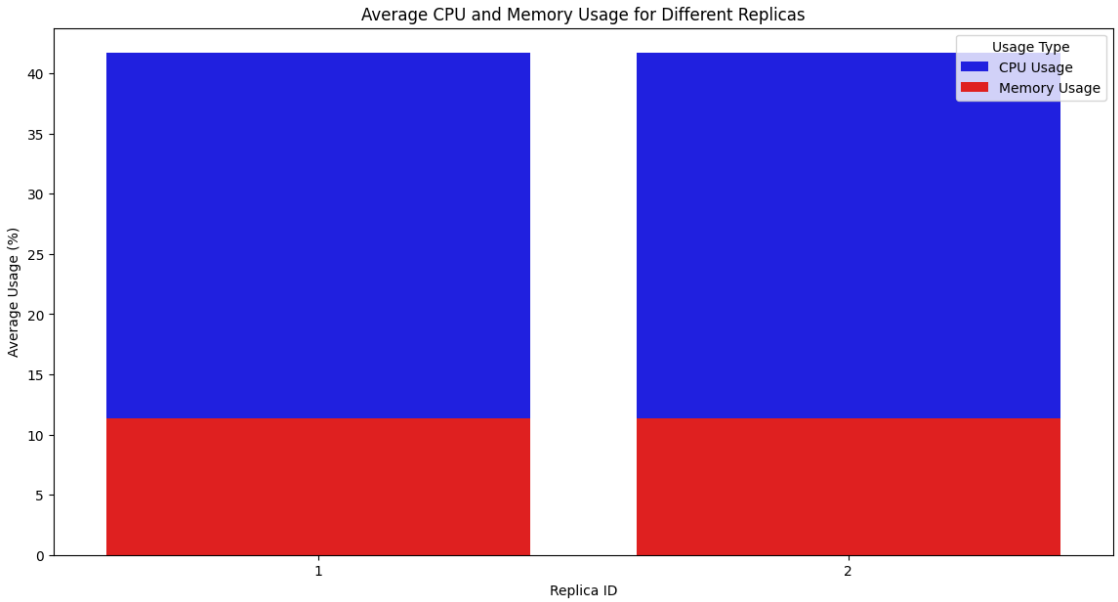

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
10000
20000
30000
40000
50000
60000
70000
10
20
30
40
50
60
07 01:50
07 02:00
07 02:10
07 02:20
07 02:30
07 02:40
07 02:50
07 03:00

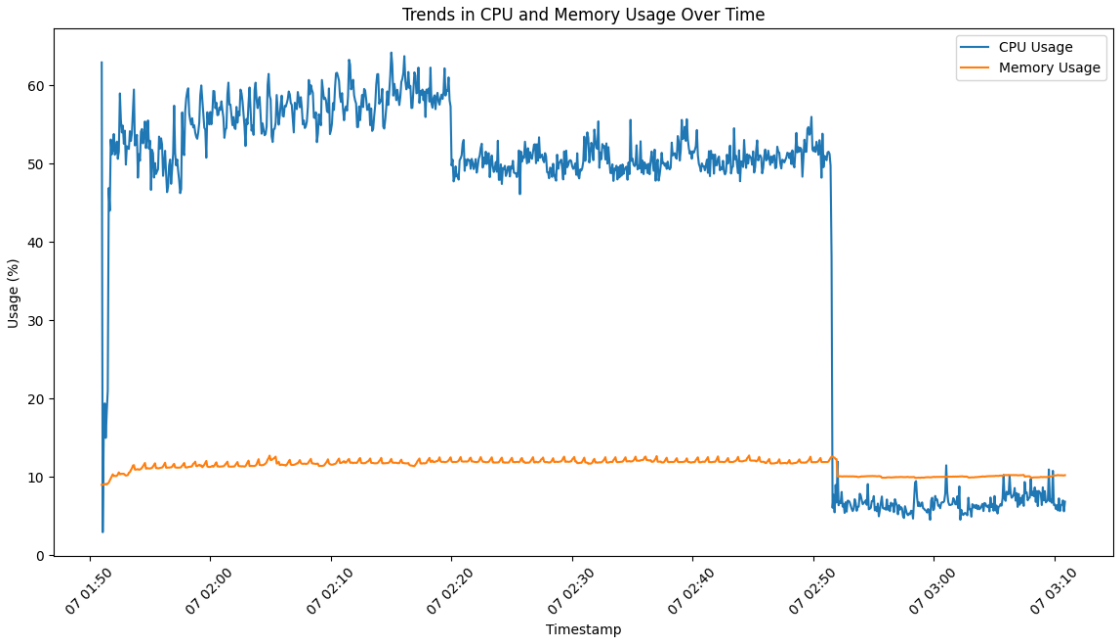

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
10000
20000
30000
40000
50000
60000
70000
10
20
30
40
50
60
07 01:50
07 02:00
07 02:10
07 02:20
07 02:30
07 02:40
07 02:50
07 03:00

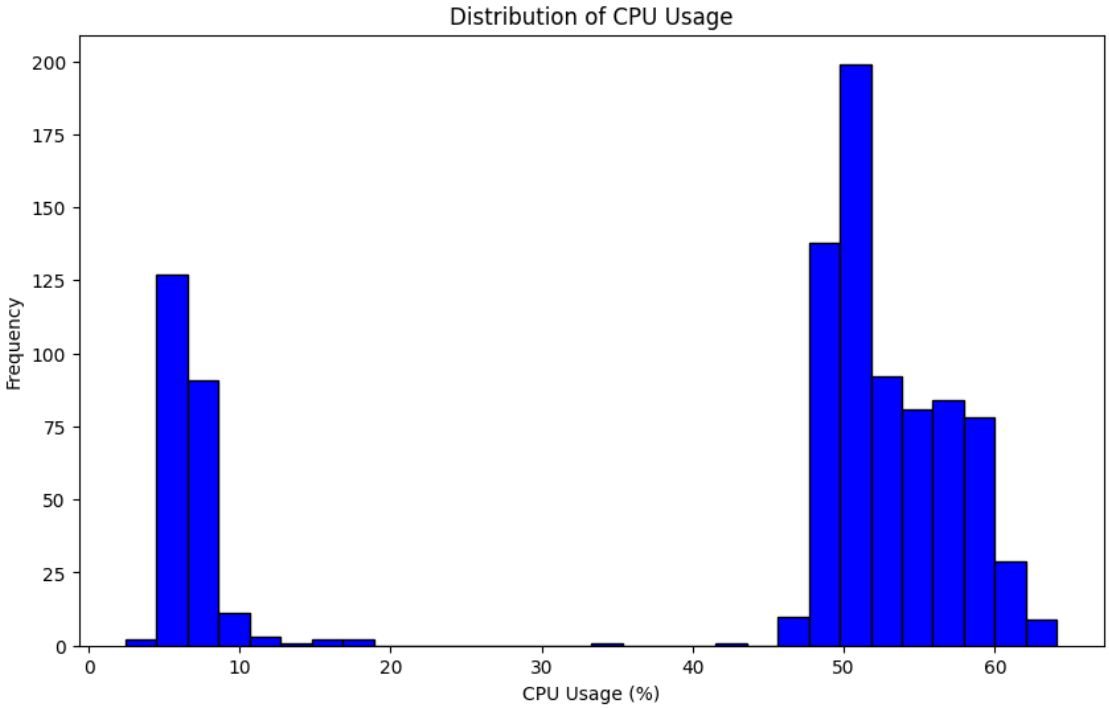

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
0
100
200
300
400
500
0
10000
20000
30000
40000
50000
60000
70000
07 01:50
07 02:00
07 02:10
07 02:20
07 02:30
07 02:40
07 02:50
07 03:00

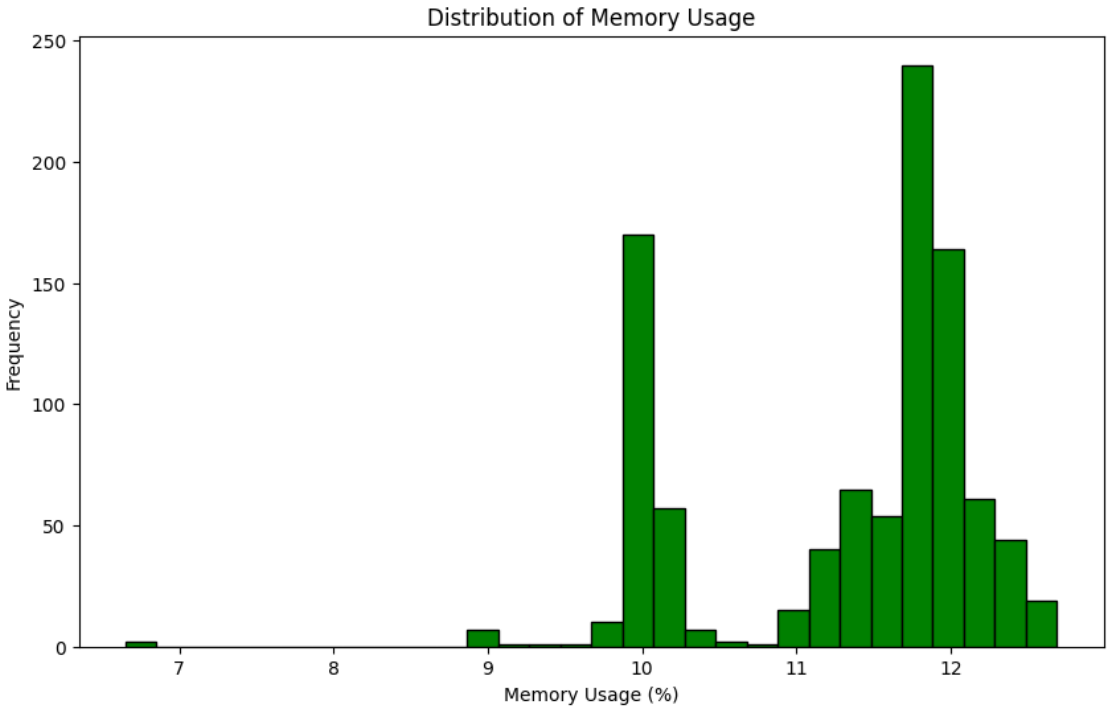

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
100
200
300
400
500
10
20
30
40
50
60
07 01:50
07 02:00
07 02:10
07 02:20
07 02:30
07 02:40
07 02:50
07 03:00

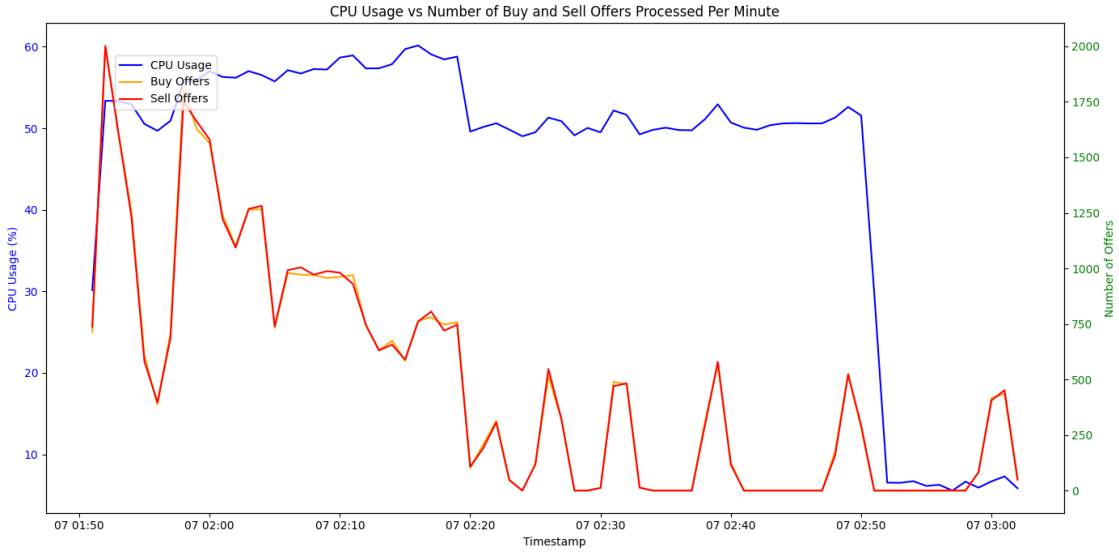

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
100
200
300
400
500
600
700
800
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
5
10
15
20
25
30
35
40
1
2
Replica ID

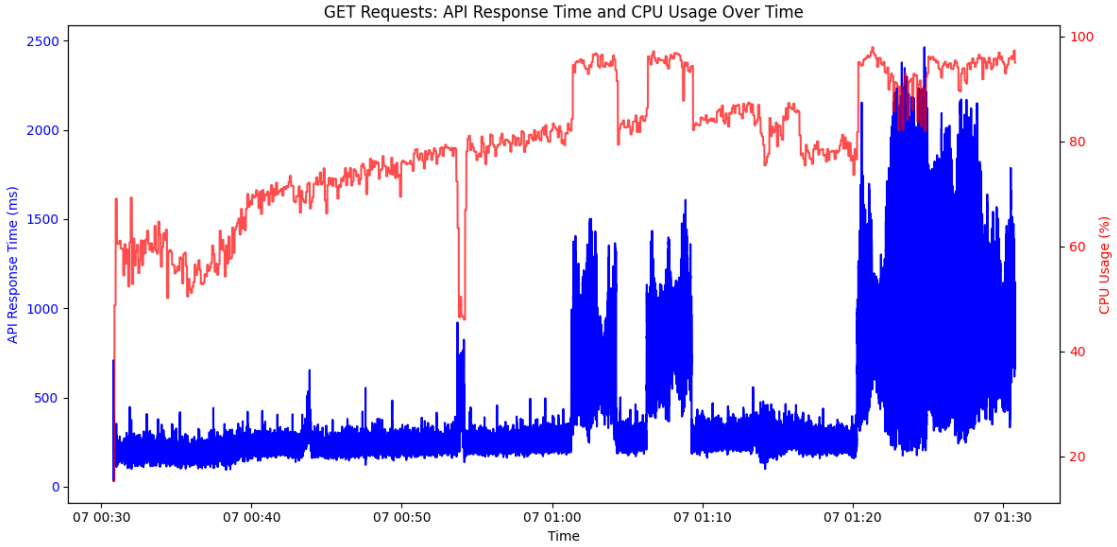

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
0
10
20
30
40
50
60
07 01:50
07 02:00
07 02:10
07 02:20
07 02:30
07 02:40
07 02:50
07 03:00
07 03:10
Timestamp

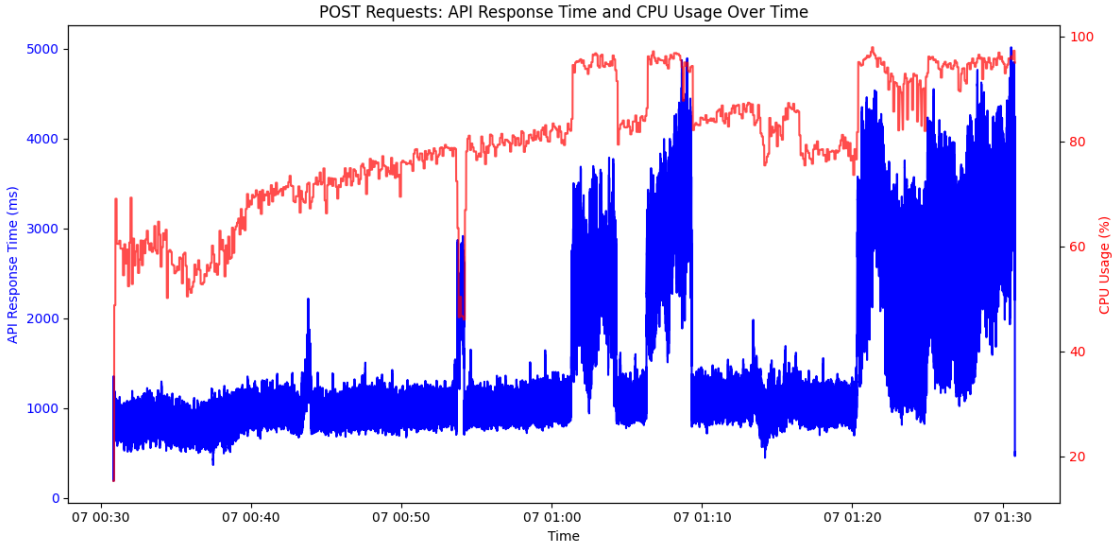

Distribution of CPU Usage
Frequency
0
25
50
75
100
125
150
175
200
0
10
20
30
40
50
60
CPU Usage (%)

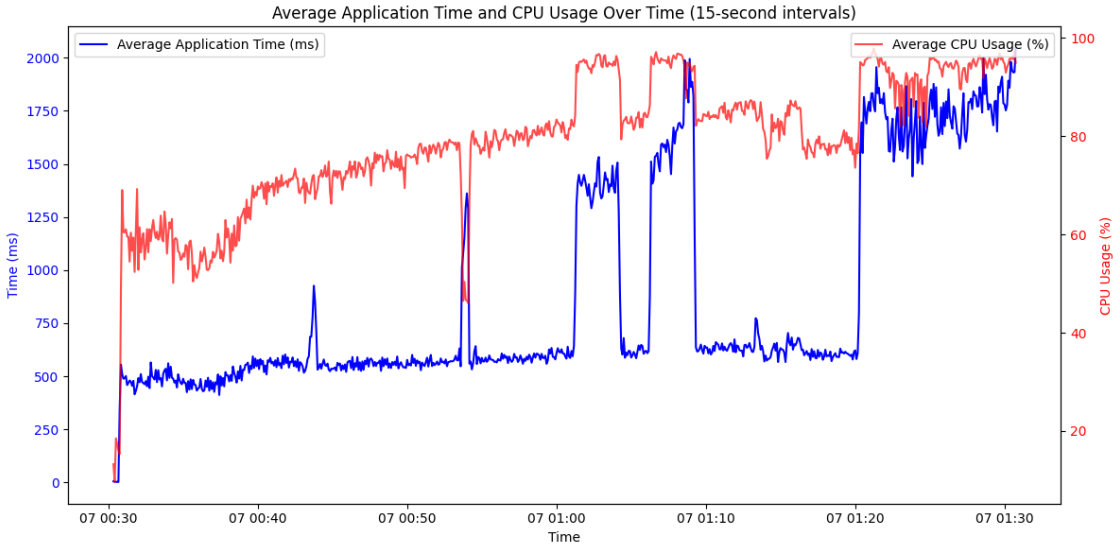


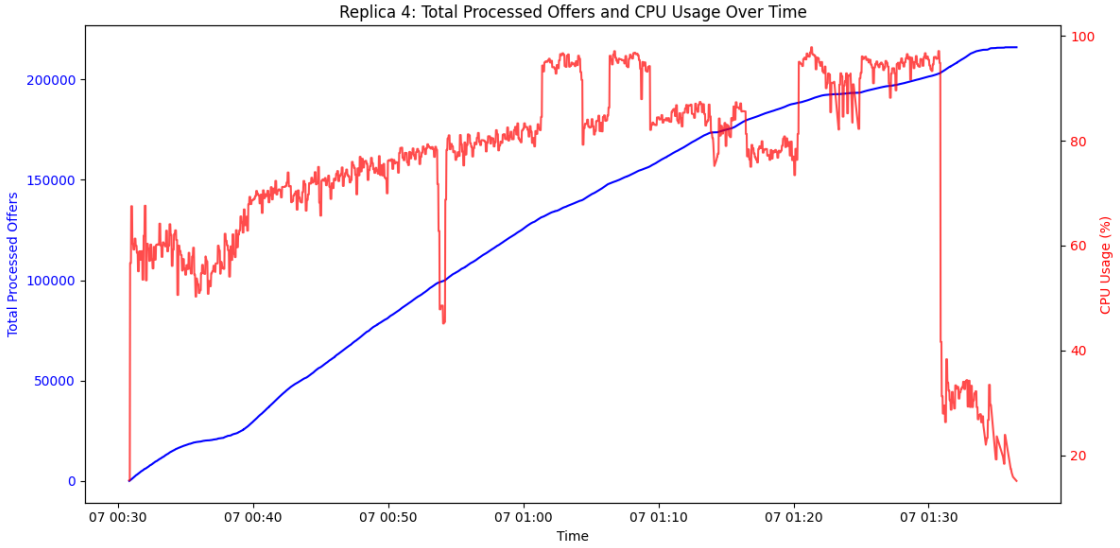


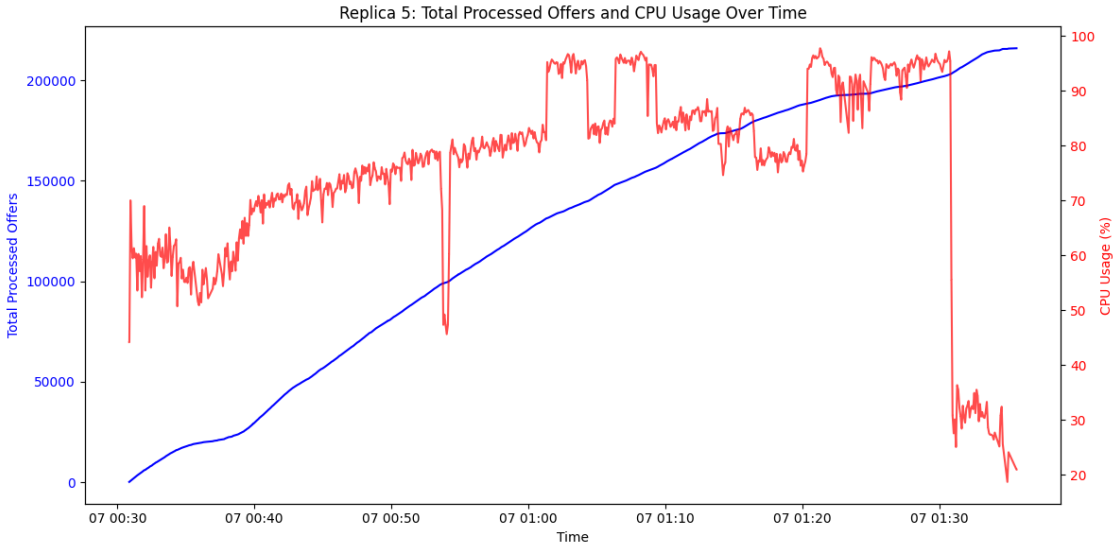

```
Loaded data from c:\Users\luki_\Desktop\logs\test1\5 2 200 500 4 500 100
successfully.
```

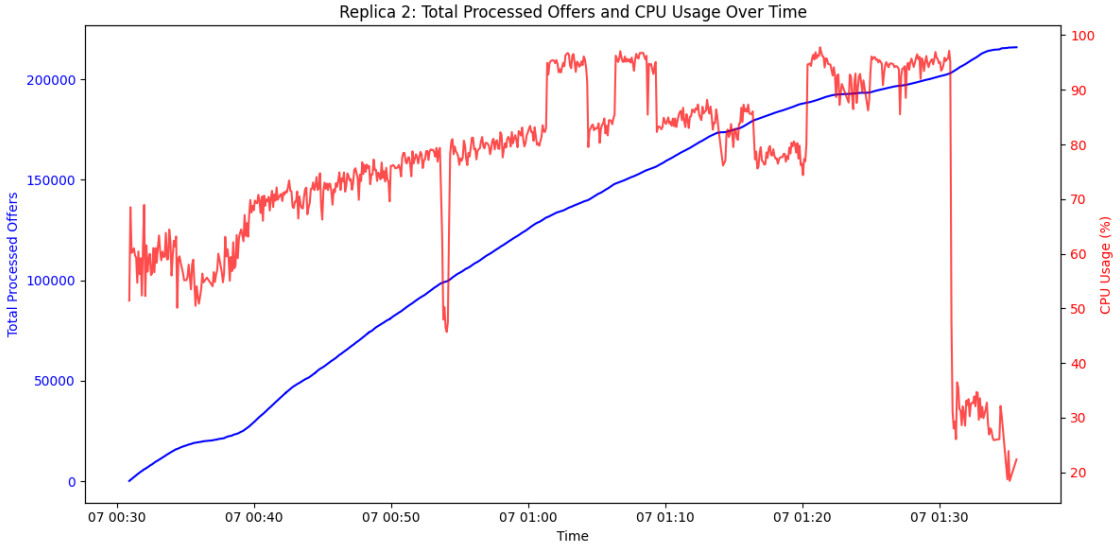

GET Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
500
1000
1500
2000
2500
20
40
60
80
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

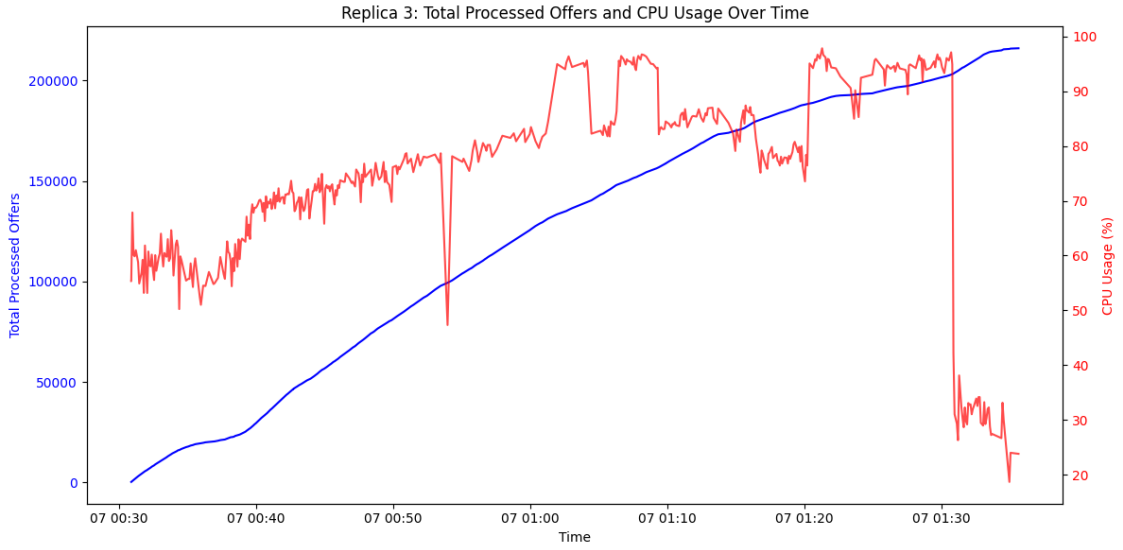

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
1000
2000
3000
4000
5000
20
40
60
80
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

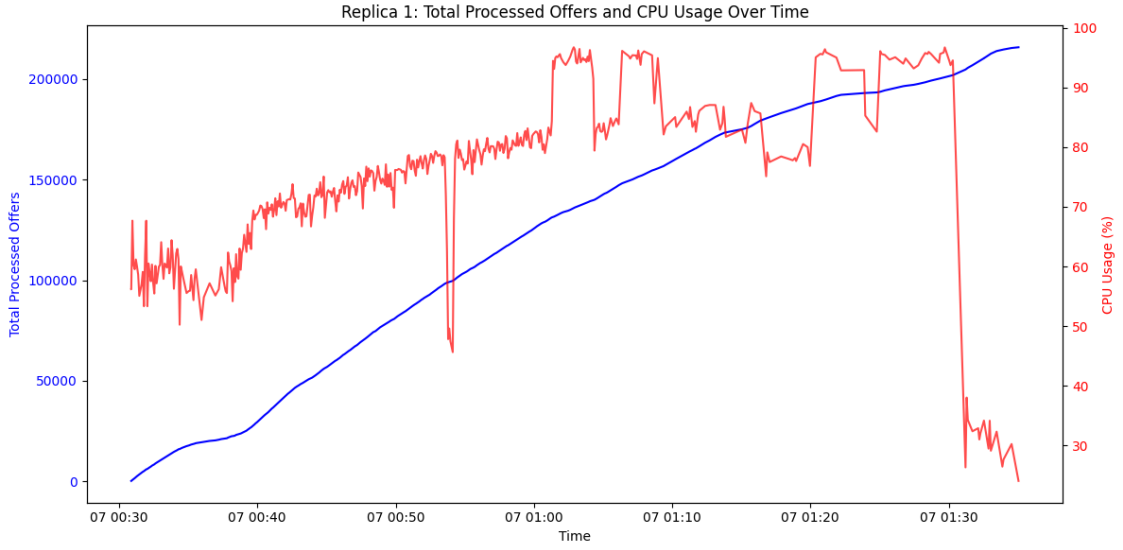

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
250
500
750
1000
1250
1500
1750
2000
20
40
60
80
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

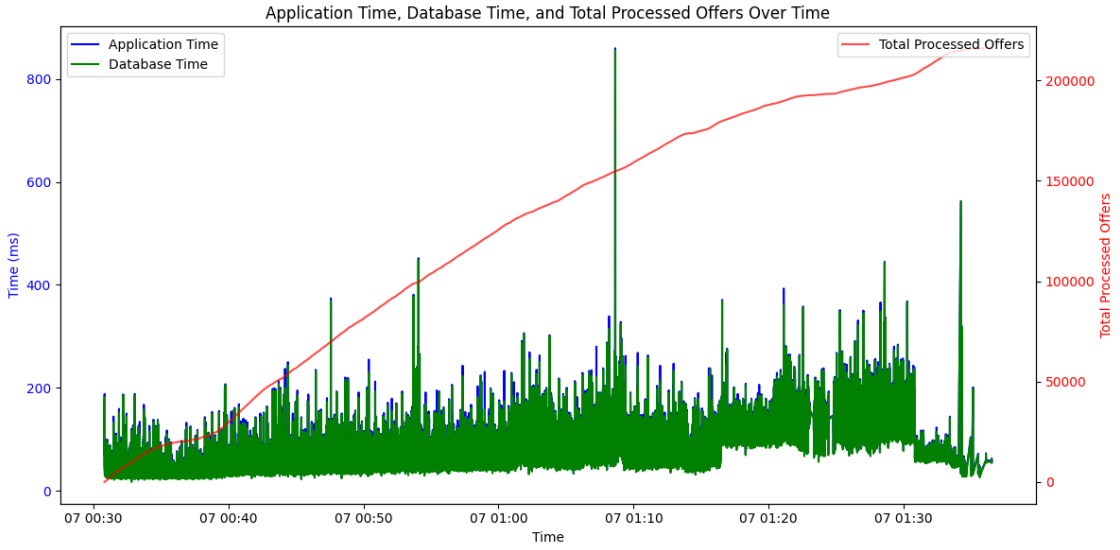

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
40
60
80
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

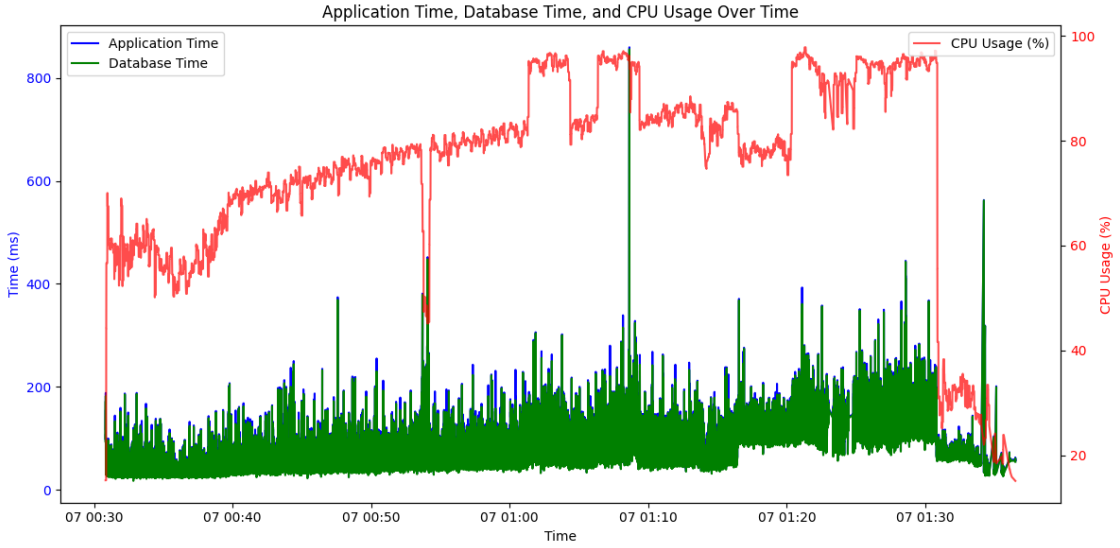

Replica 5: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
30
40
50
60
70
80
90
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

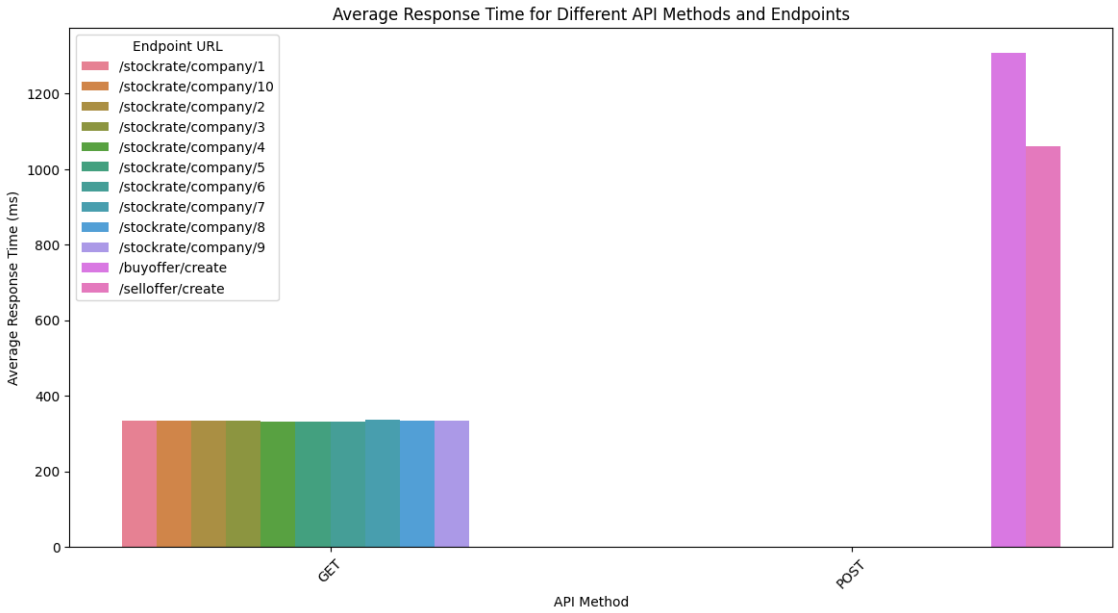

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
30
40
50
60
70
80
90
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

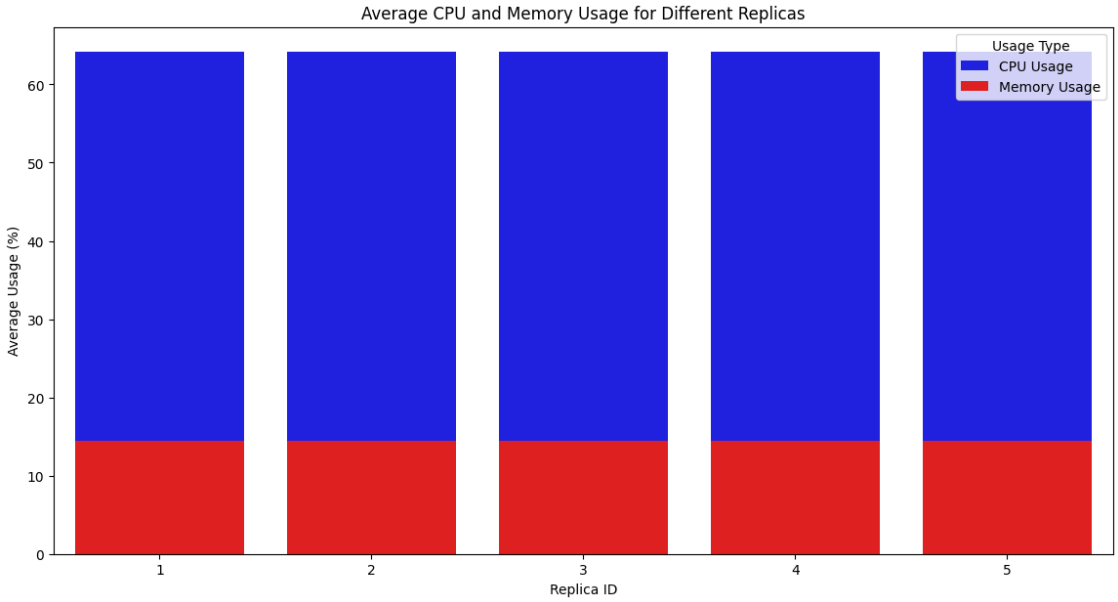

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
30
40
50
60
70
80
90
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

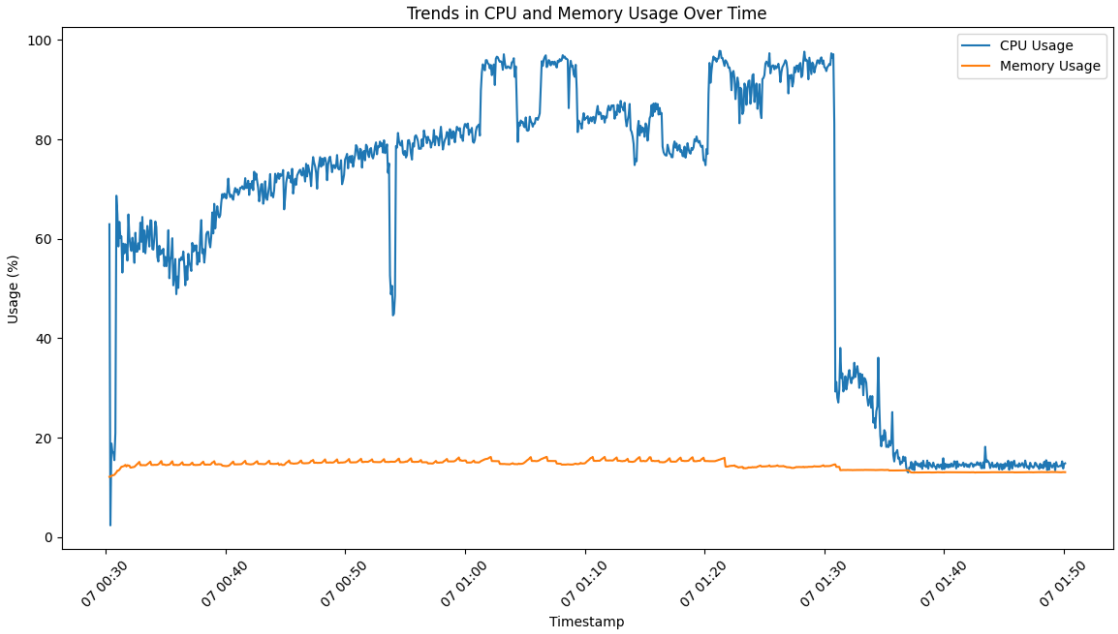

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
30
40
50
60
70
80
90
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

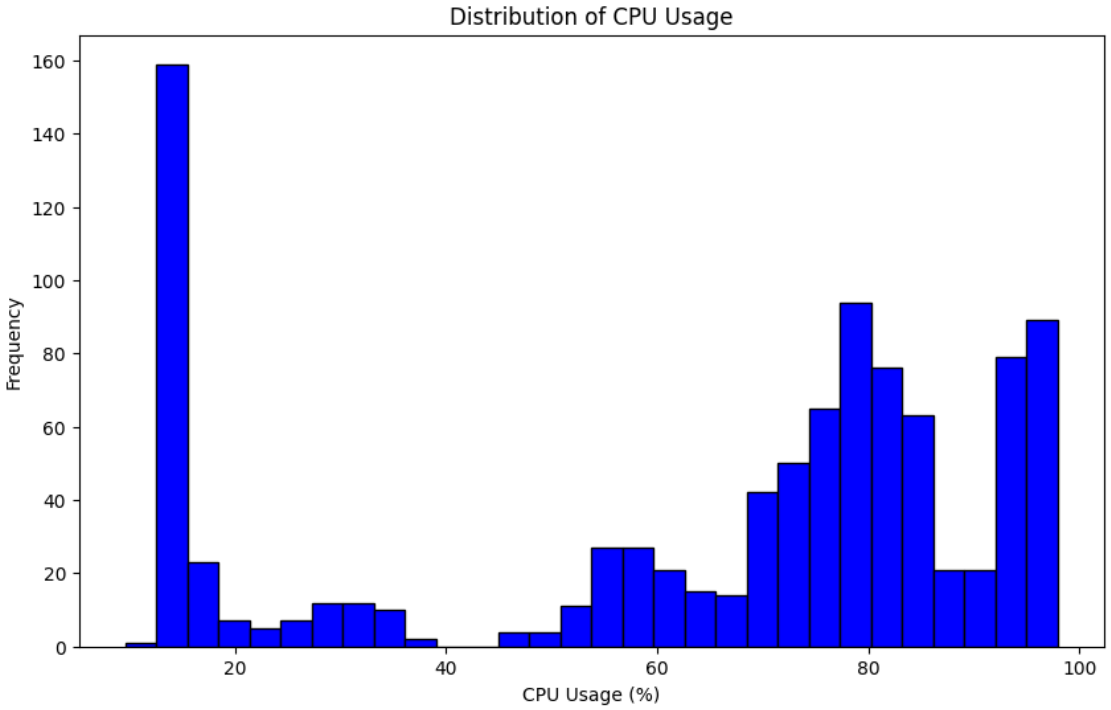

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
0
200
400
600
800
0
50000
100000
150000
200000
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

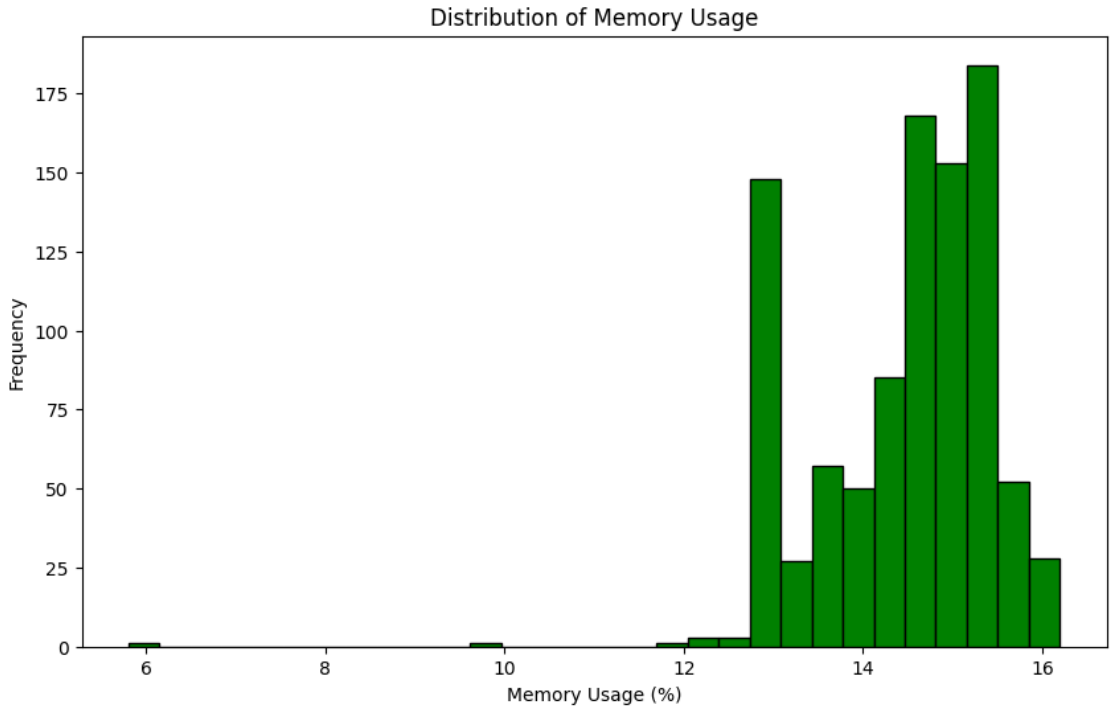

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
200
400
600
800
20
40
60
80
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30

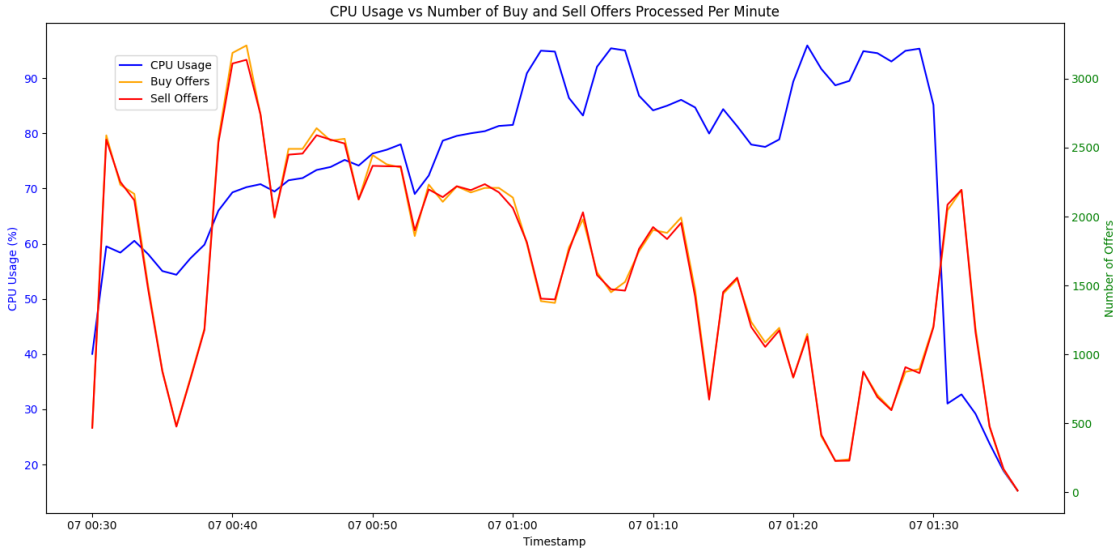

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
10
20
30
40
50
60
1
2
3
4
5
Replica ID

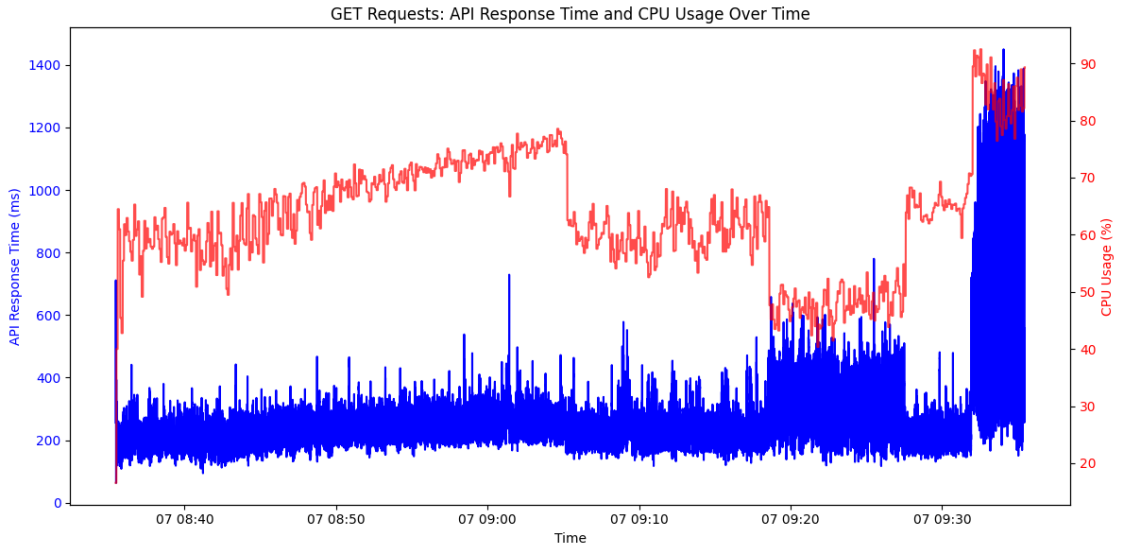

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
Timestamp
0
20
40
60
80
100
07 00:30
07 00:40
07 00:50
07 01:00
07 01:10
07 01:20
07 01:30
07 01:40
07 01:50

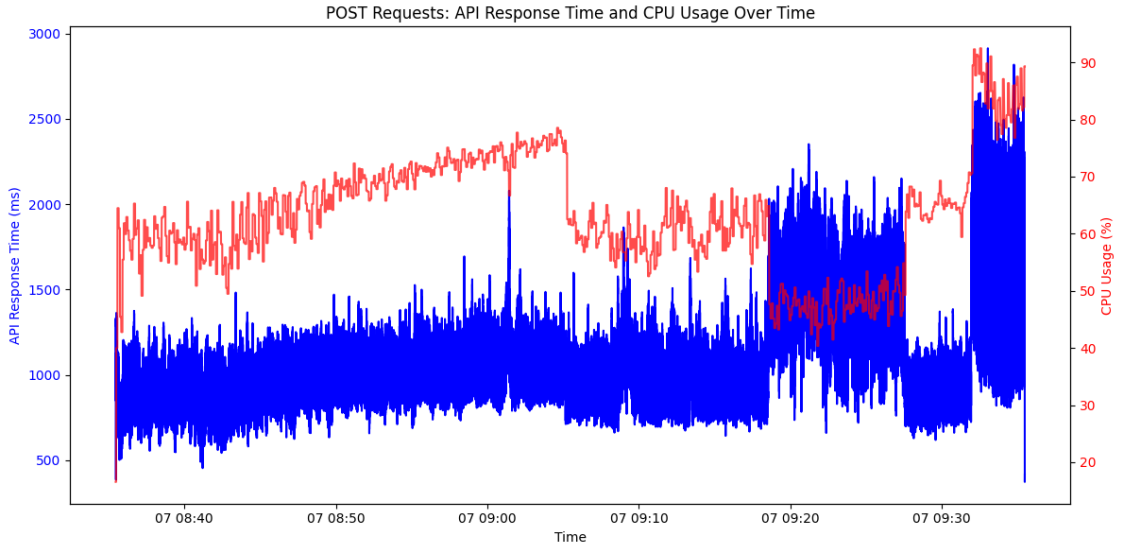

Distribution of CPU Usage
Frequency
CPU Usage (%)
0
20
40
60
80
100
120
140
160

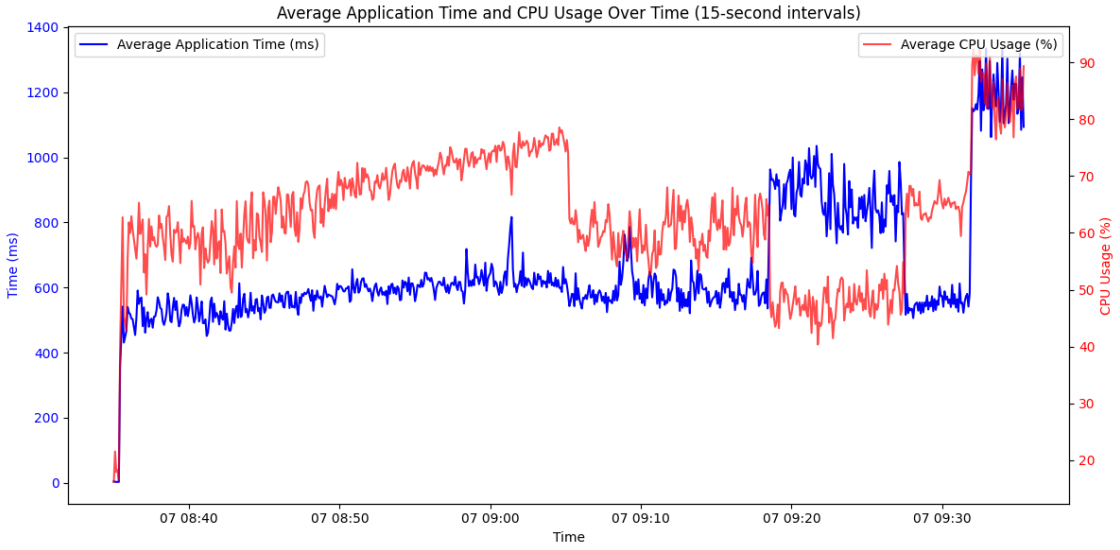


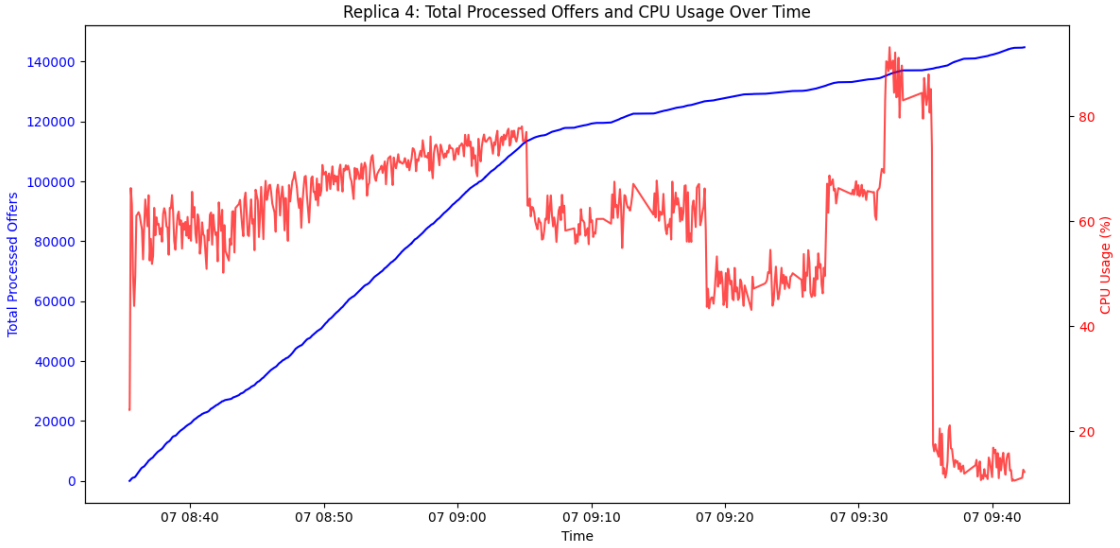


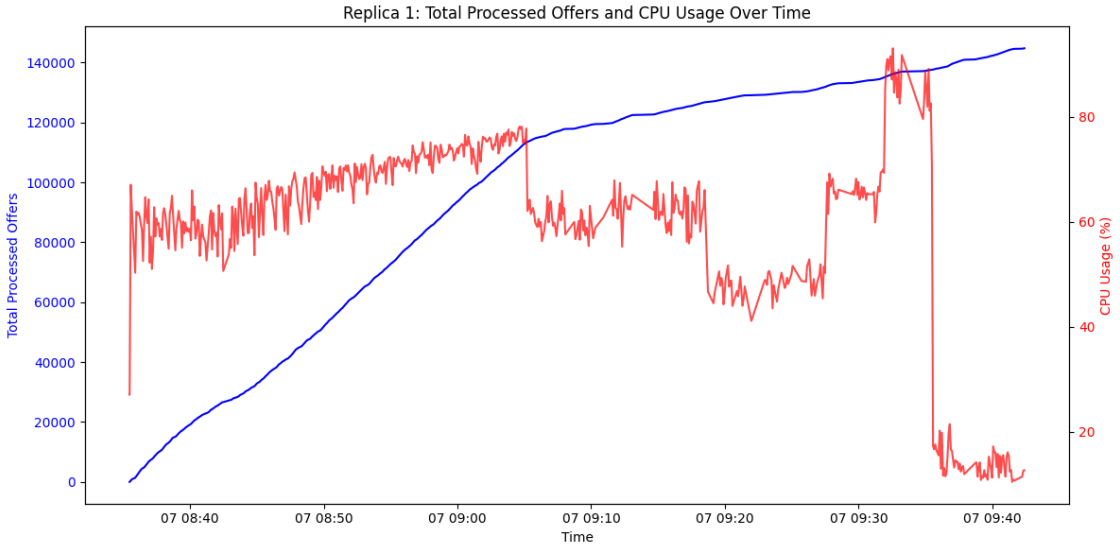

```
One or more CSV files are missing in directory:
c:\Users\luki_\Desktop\logs\test2
Loaded data from c:\Users\luki_\Desktop\logs\test2\4 3 200 500 16 500 100
successfully.
```

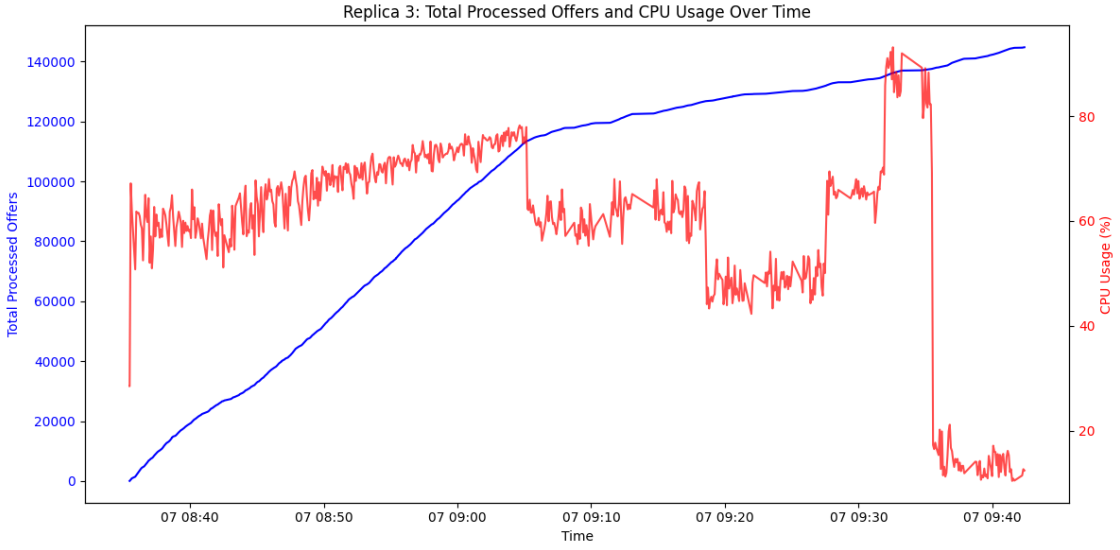

GET Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
200
400
600
800
1000
1200
1400
20
30
40
50
60
70
80
90
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30

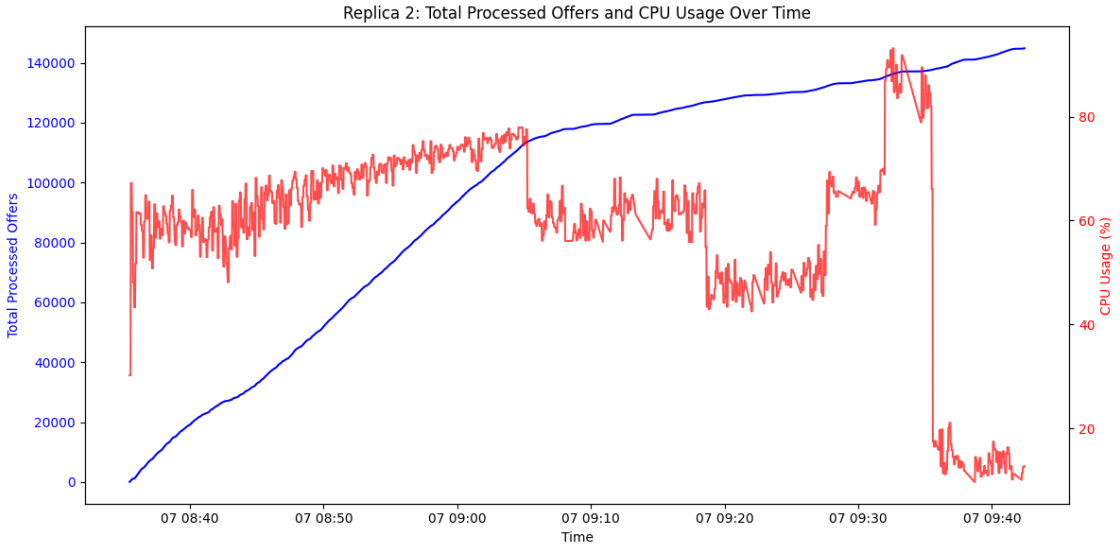

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
500
1000
1500
2000
2500
3000
20
30
40
50
60
70
80
90
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30

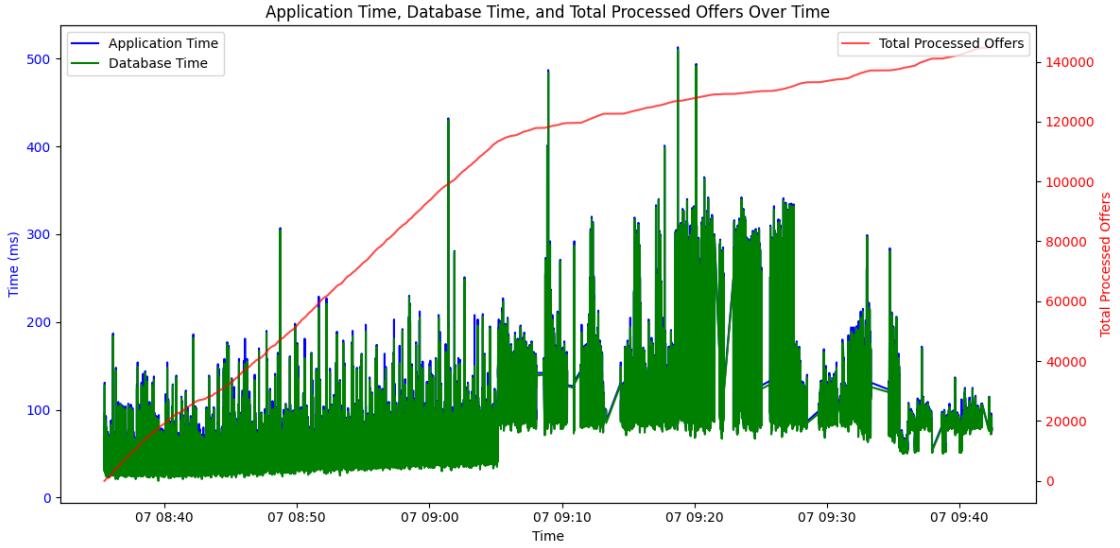

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30

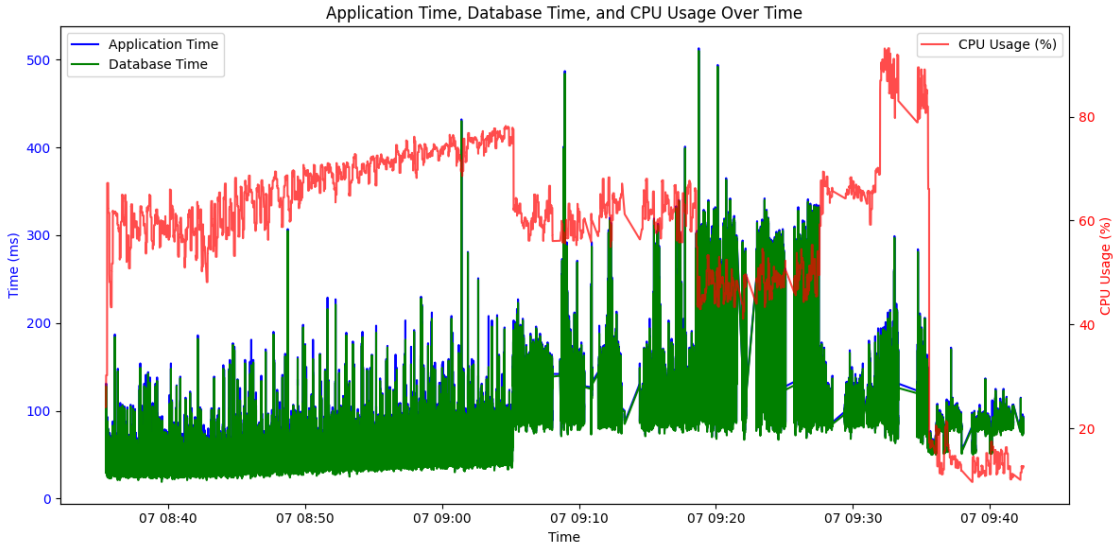

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30
07 09:40

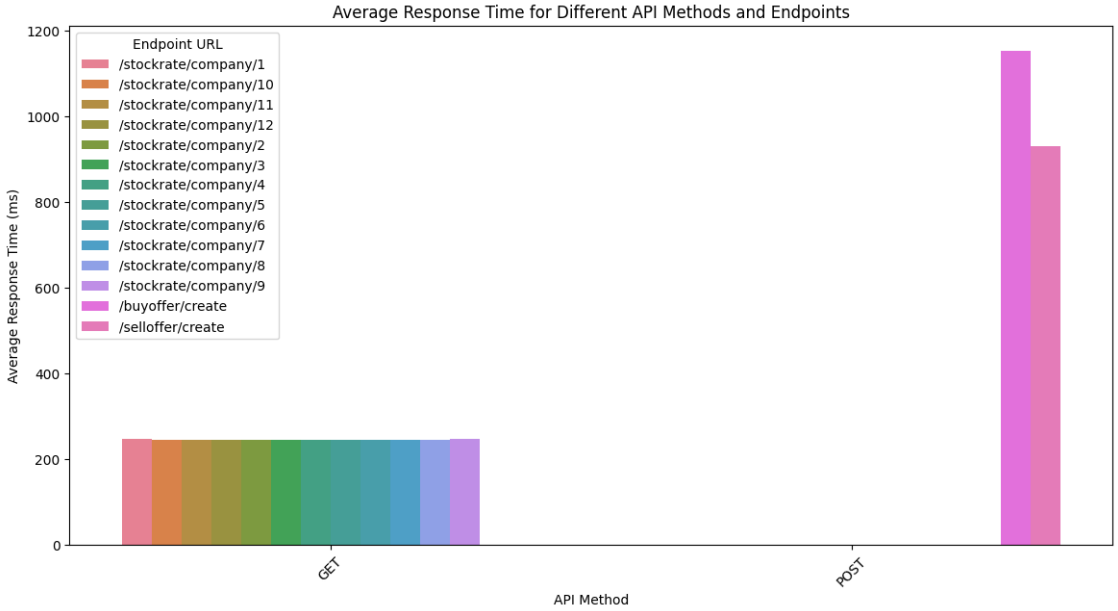

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
20
40
60
80
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30
07 09:40

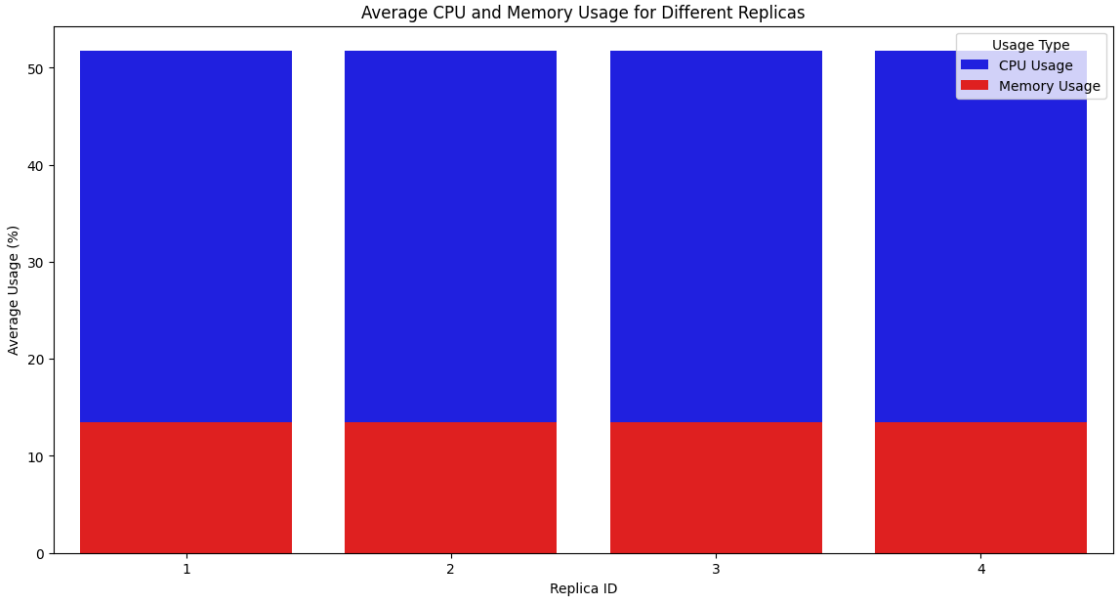

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
20
40
60
80
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30
07 09:40

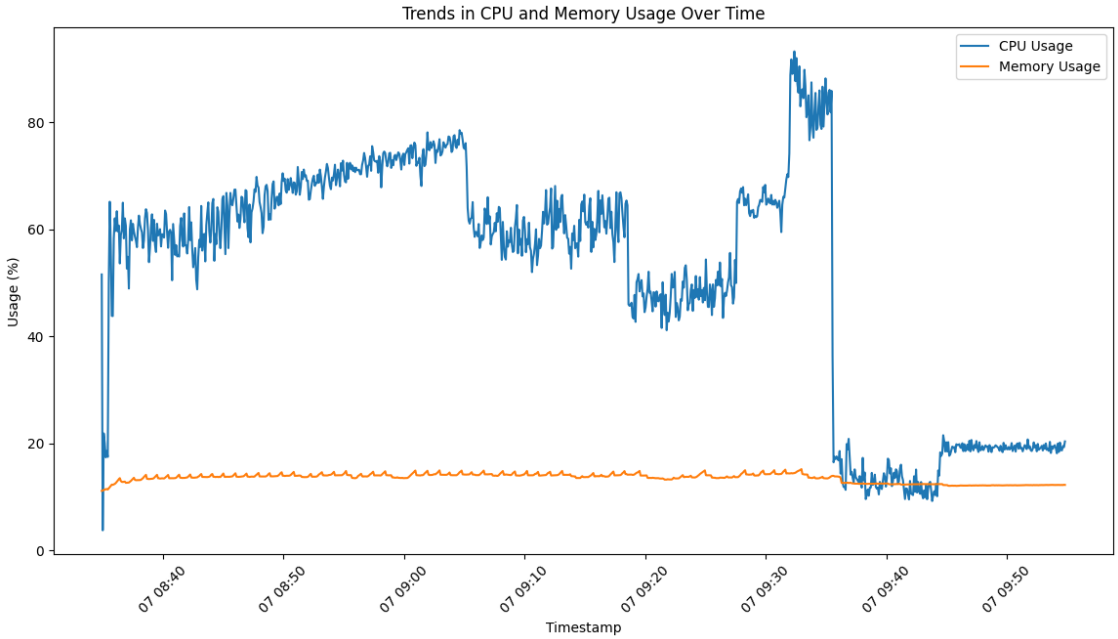

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
20
40
60
80
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30
07 09:40

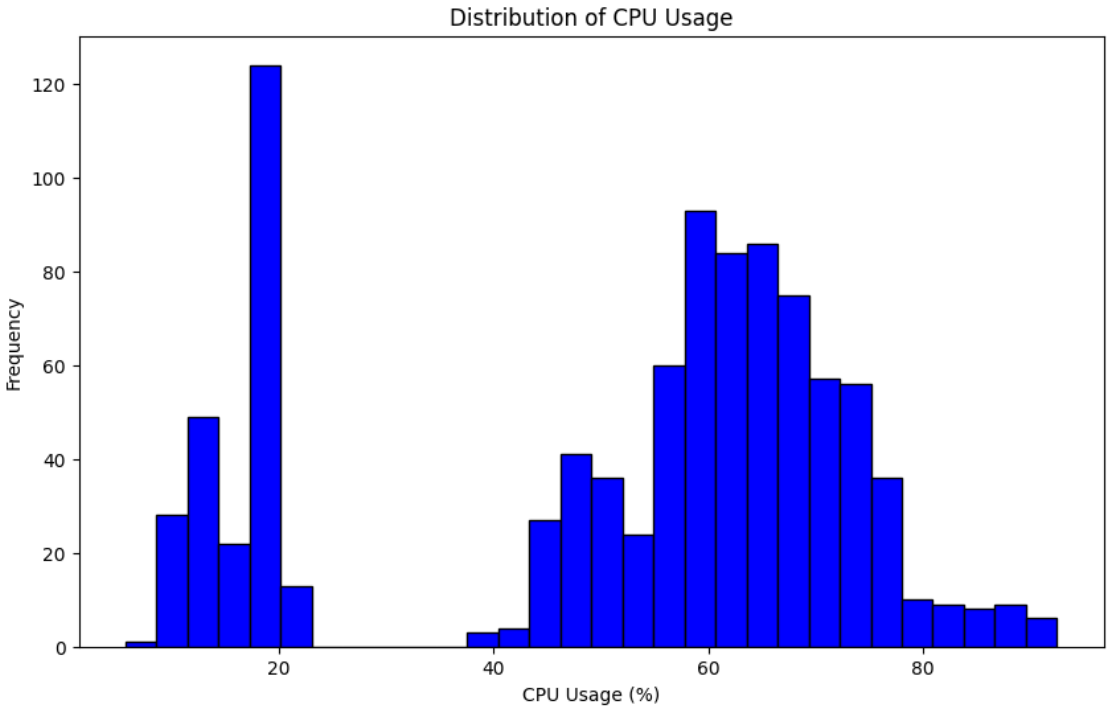

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
0
100
200
300
400
500
0
20000
40000
60000
80000
100000
120000
140000
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30
07 09:40

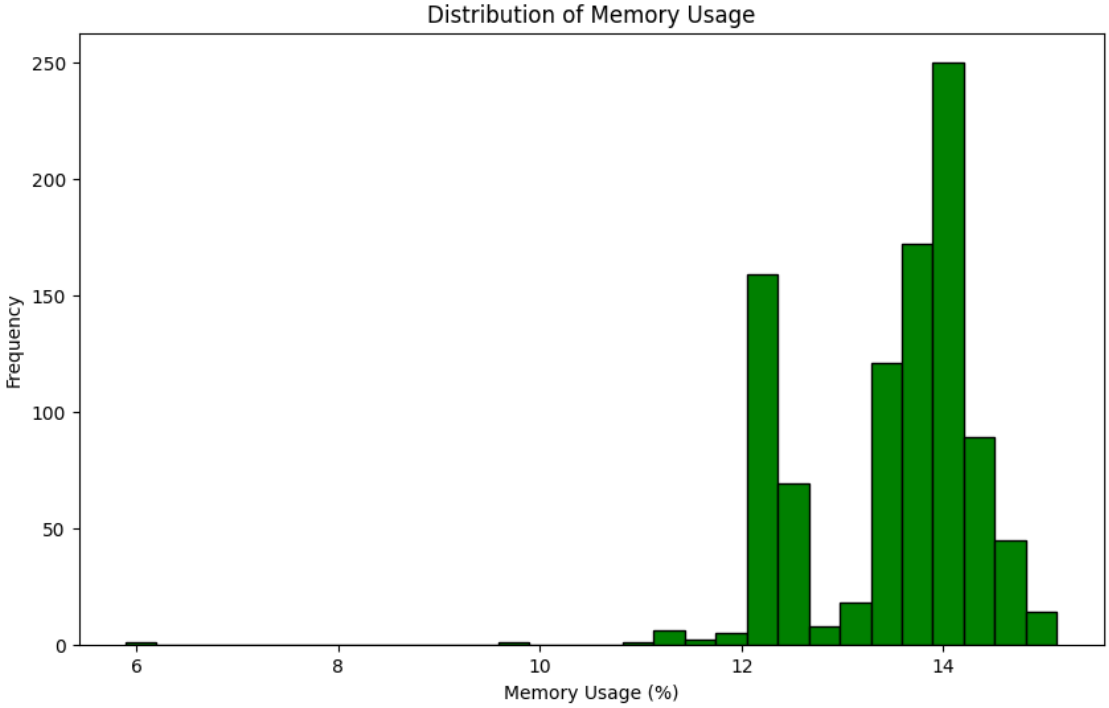

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
100
200
300
400
500
20
40
60
80
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30
07 09:40

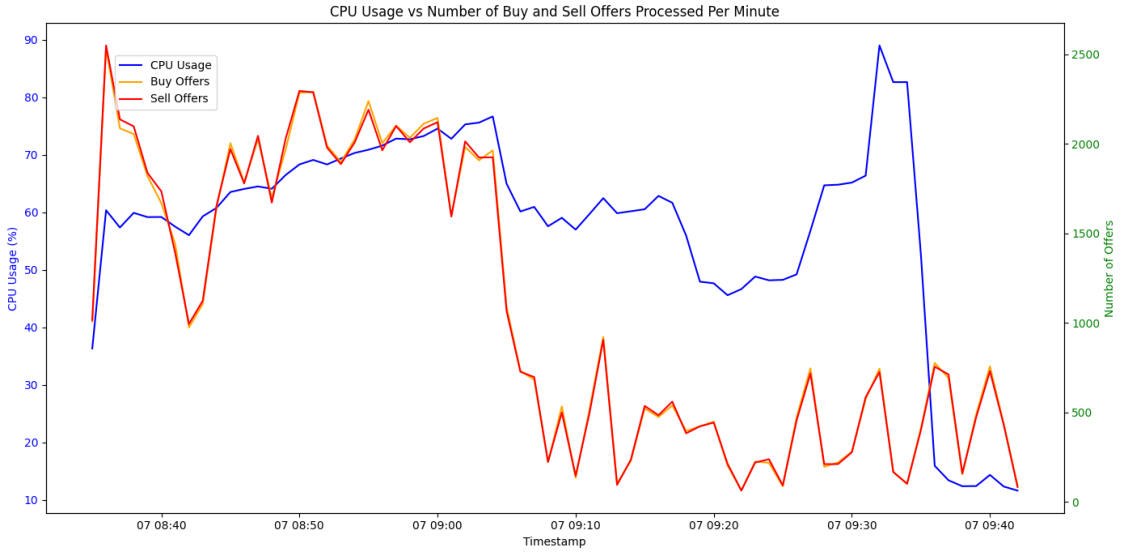

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
10
20
30
40
50
1
2
3
4
Replica ID

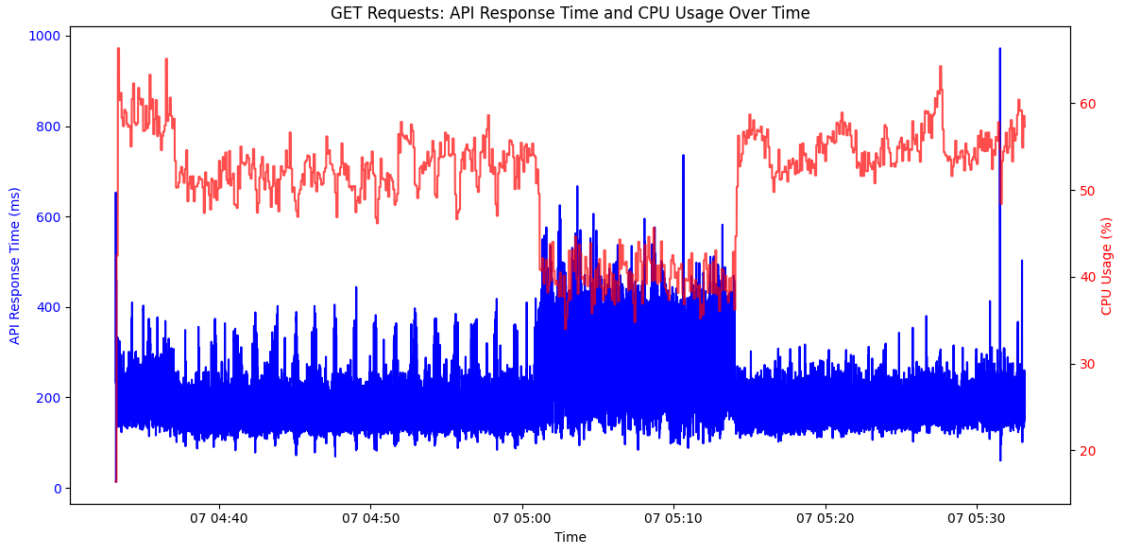

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
0
20
40
60
80
07 08:40
07 08:50
07 09:00
07 09:10
07 09:20
07 09:30
07 09:40
07 09:50
Timestamp

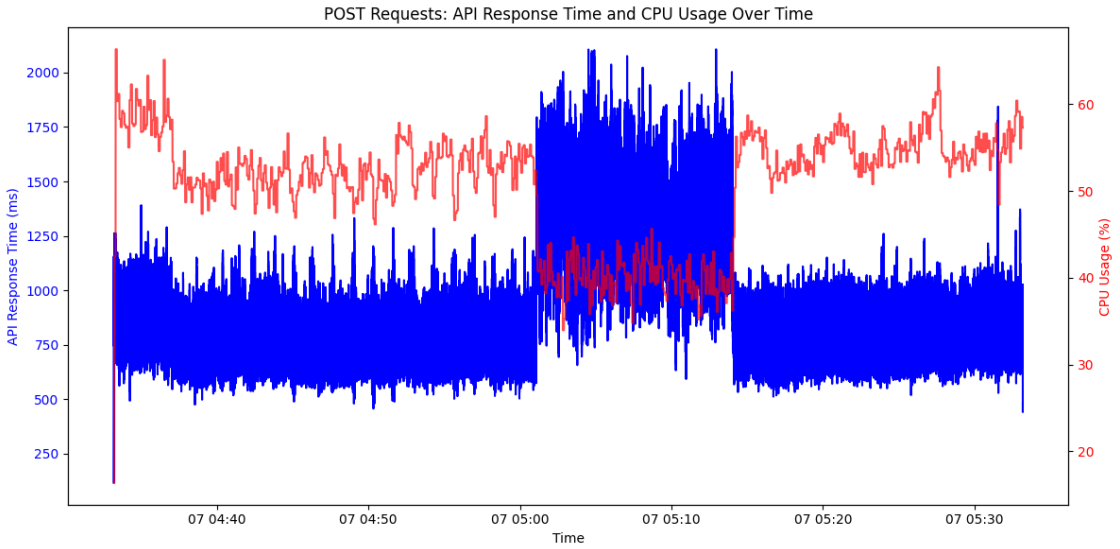

Distribution of CPU Usage
Frequency
CPU Usage (%)
0
20
40
60
80
100
120

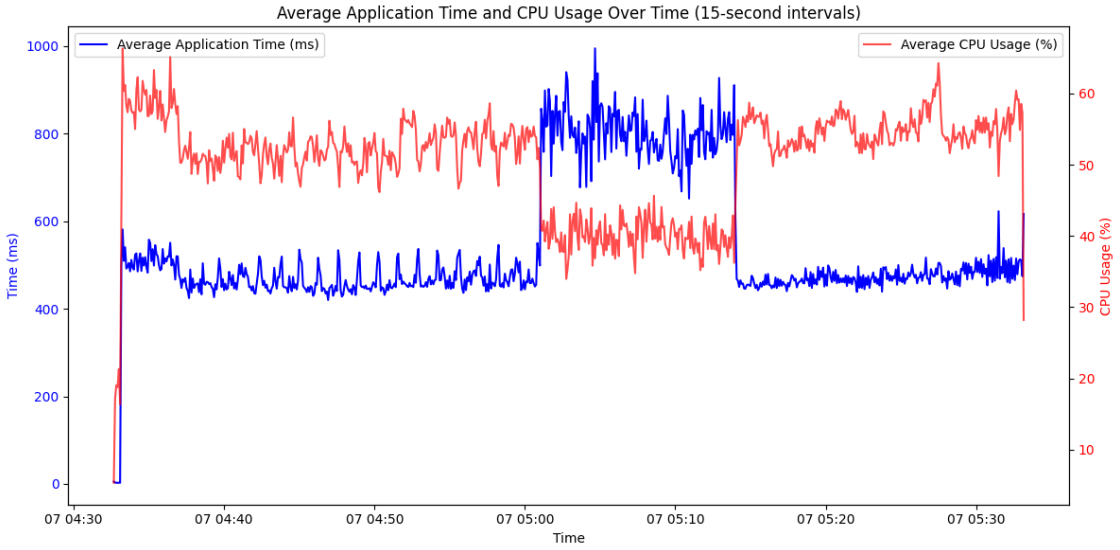

Distribution of Memory Usage
Frequency
Memory Usage (%)
0
50
100
150
200
250
6
8
10
12
14

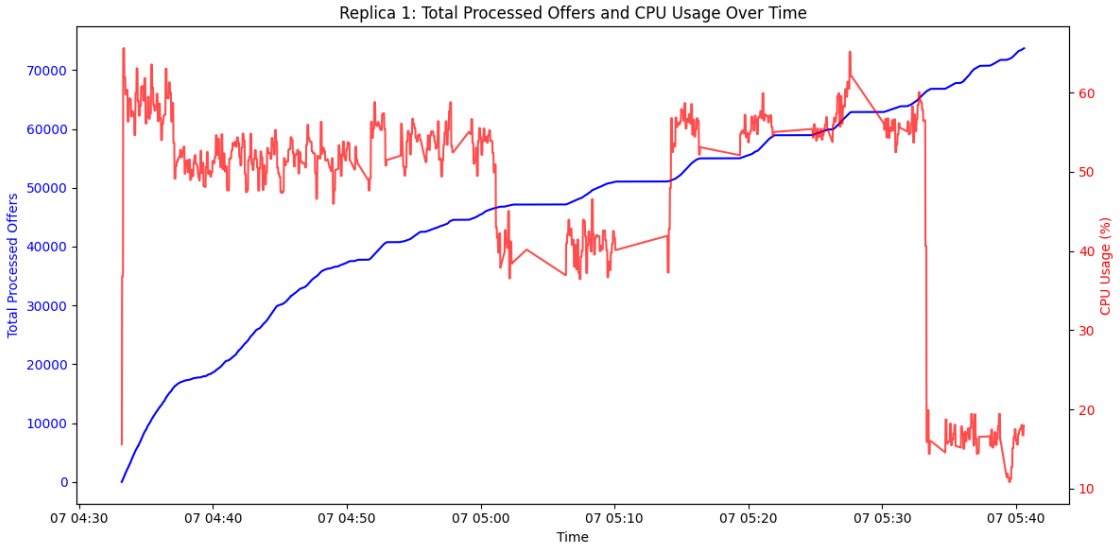


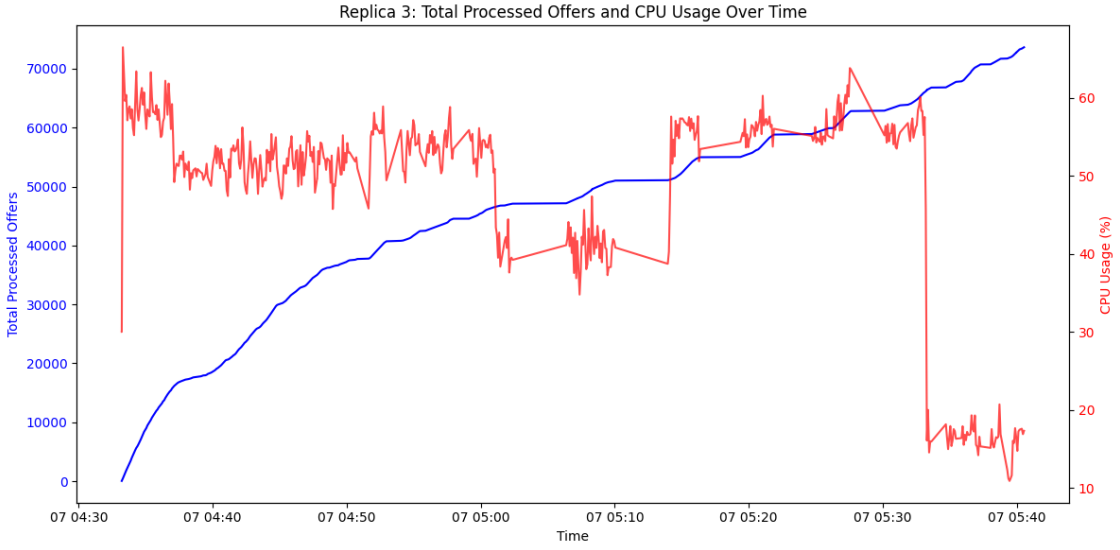

```
Loaded data from c:\Users\luki_\Desktop\logs\test2\4 3 200 500 2 500 100
successfully.
```

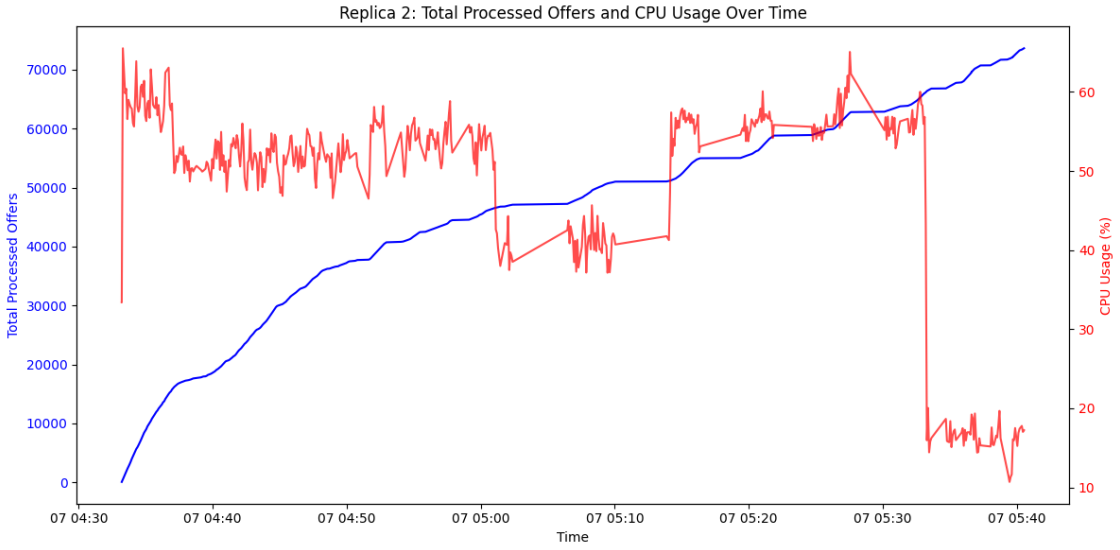


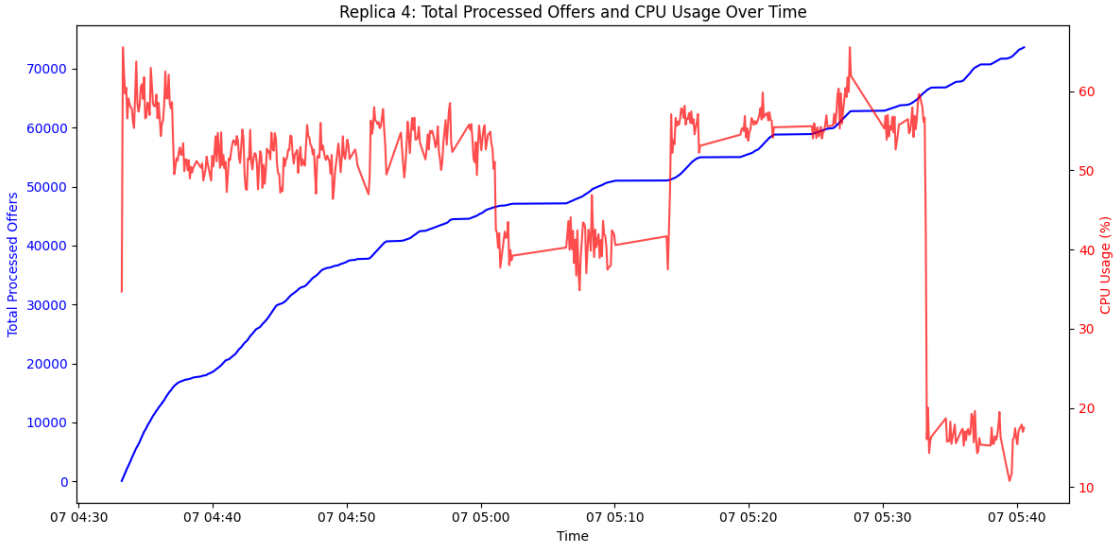

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
2000
1750
1500
1250
1000
750
500
250
60
50
40
30
20
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30

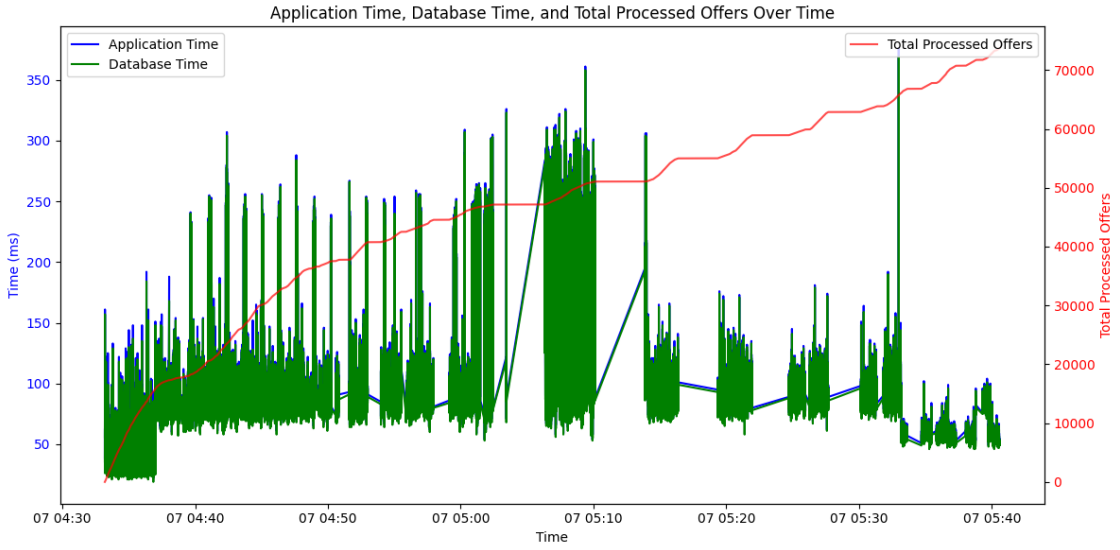

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
1000
800
600
400
200
0
60
50
40
30
20
10
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30

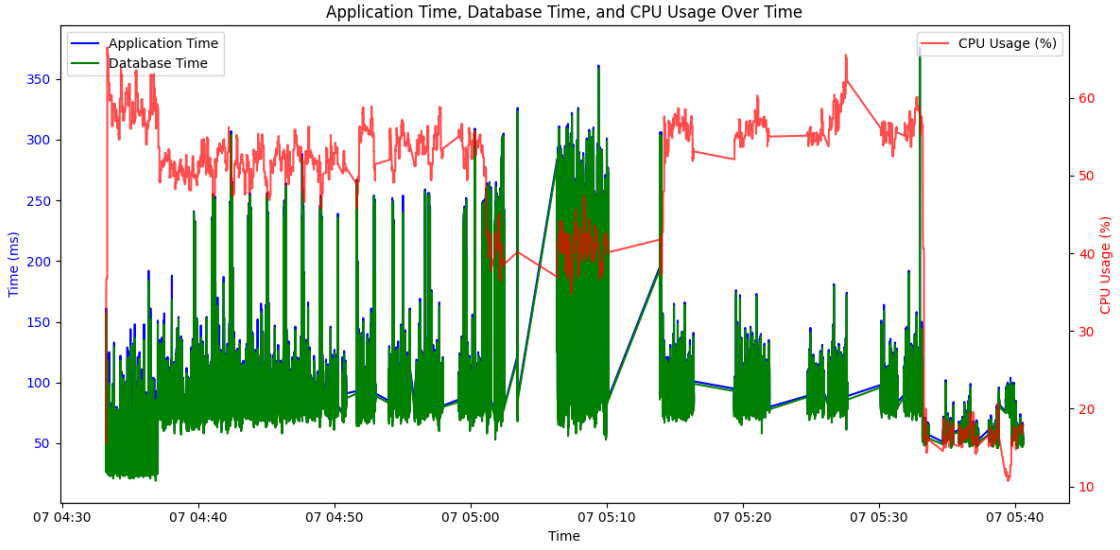

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
10000
20000
30000
40000
50000
60000
70000
10
20
30
40
50
60
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30
07 05:40

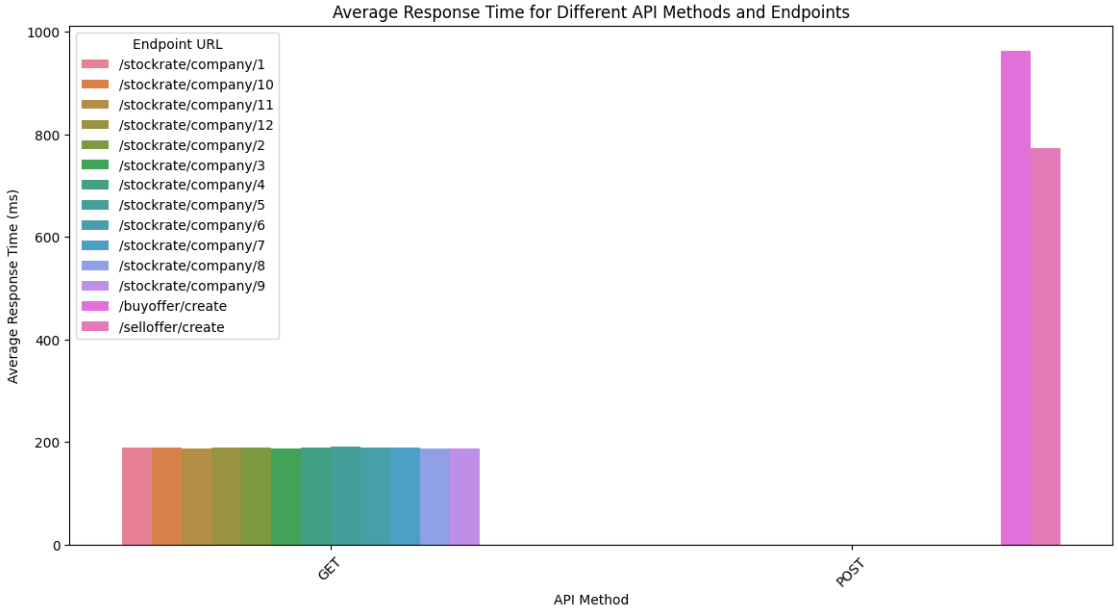

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
10000
20000
30000
40000
50000
60000
70000
10
20
30
40
50
60
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30
07 05:40

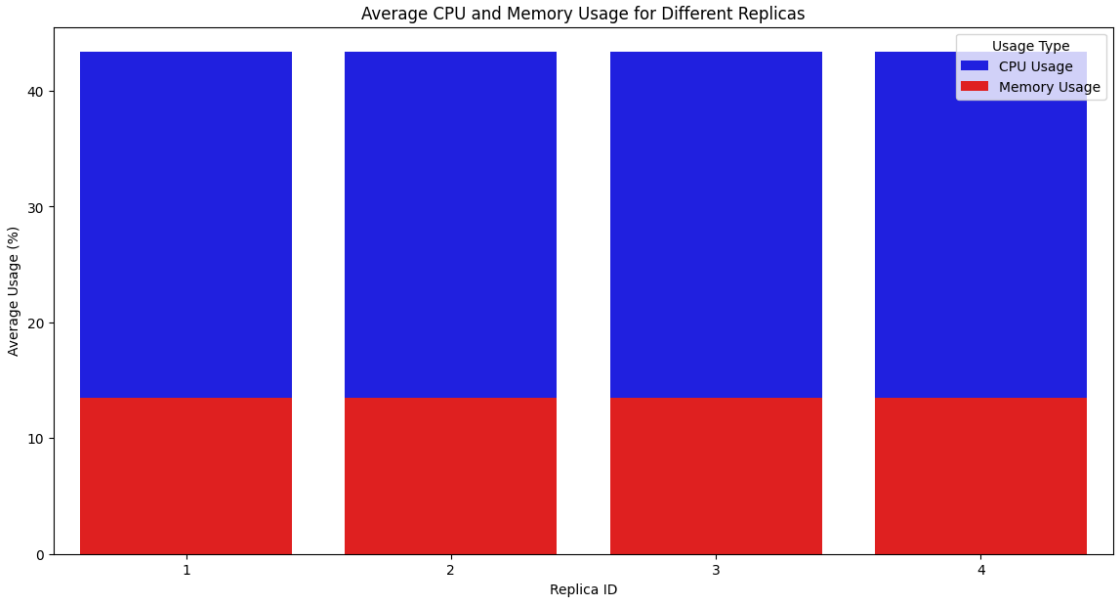

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
10000
20000
30000
40000
50000
60000
70000
10
20
30
40
50
60
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30
07 05:40

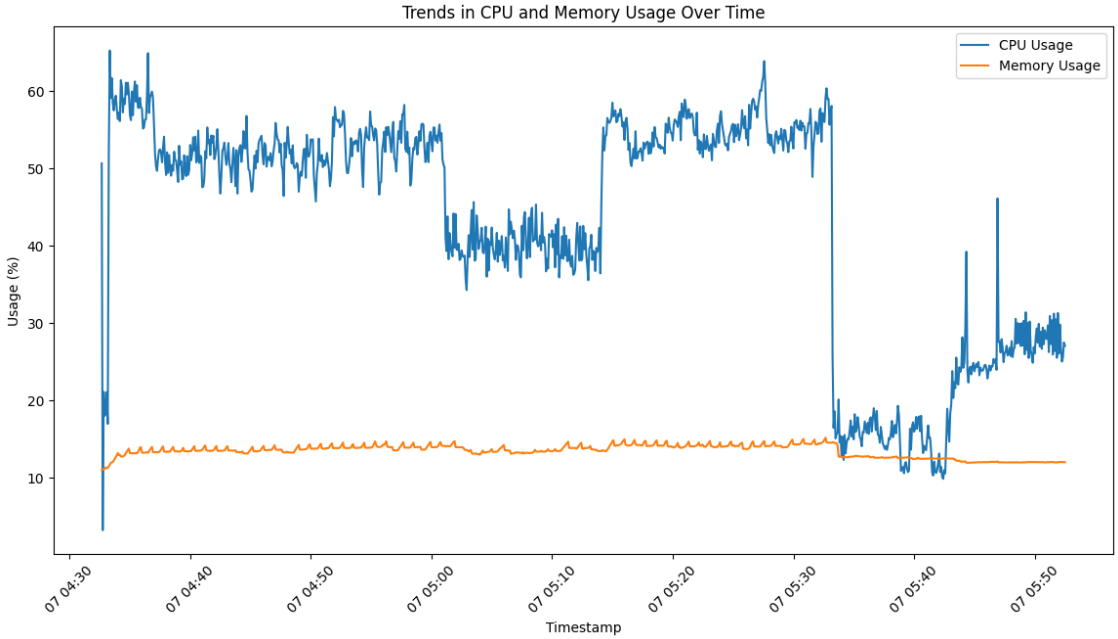

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
10000
20000
30000
40000
50000
60000
70000
10
20
30
40
50
60
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30
07 05:40

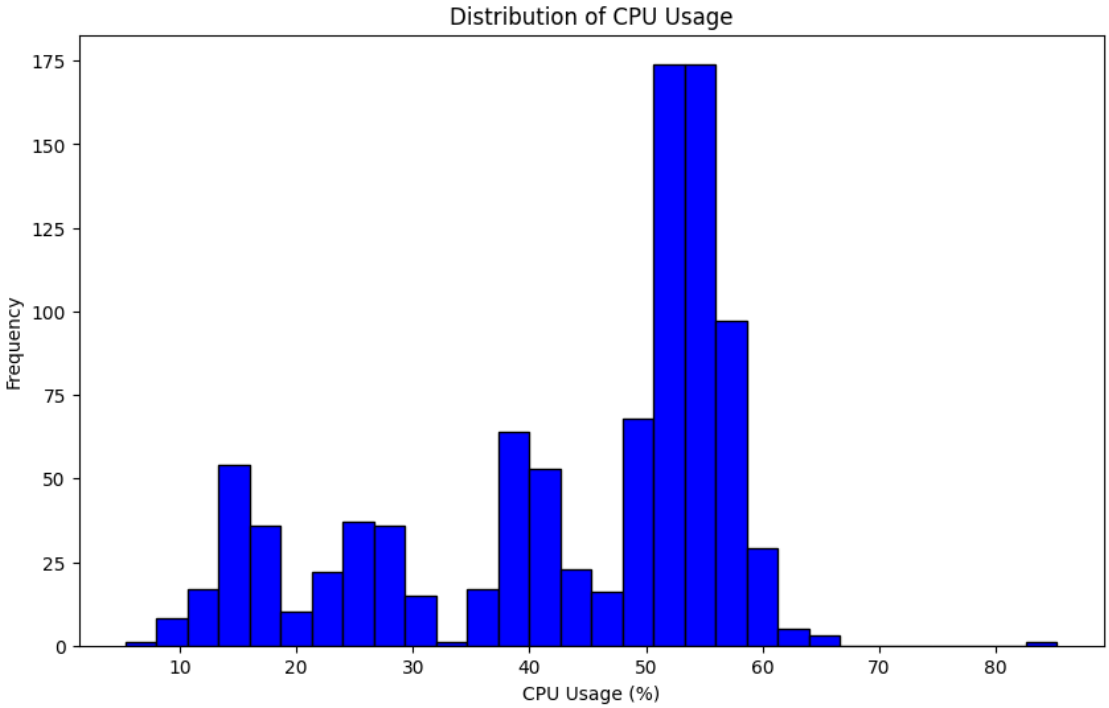

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30
07 05:40

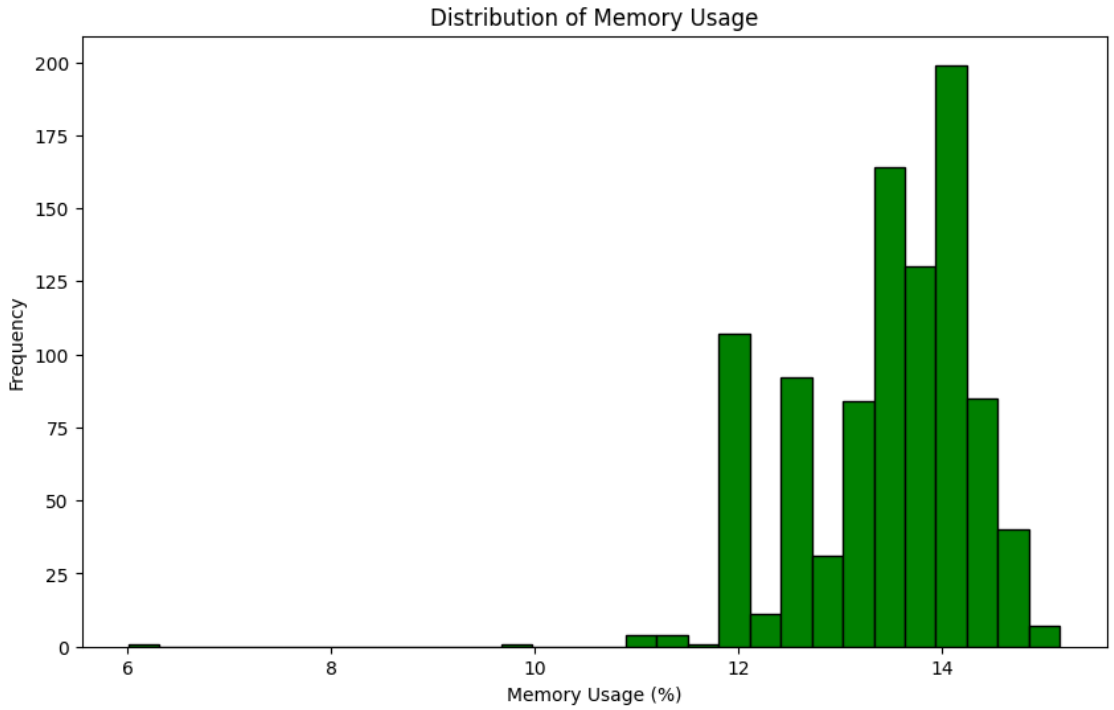

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30
07 05:40

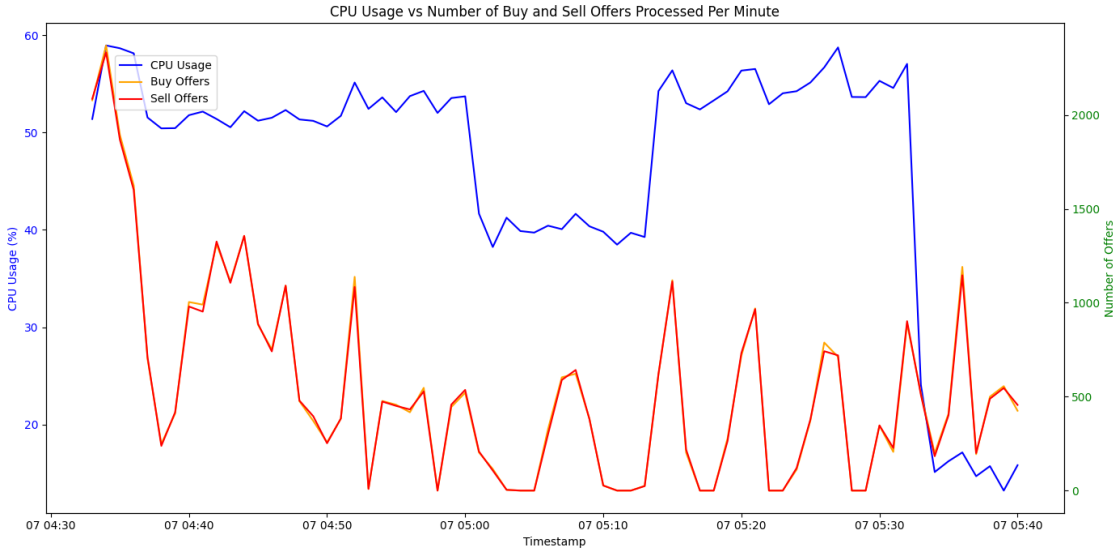

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
10
20
30
40
1
2
3
4
Replica ID

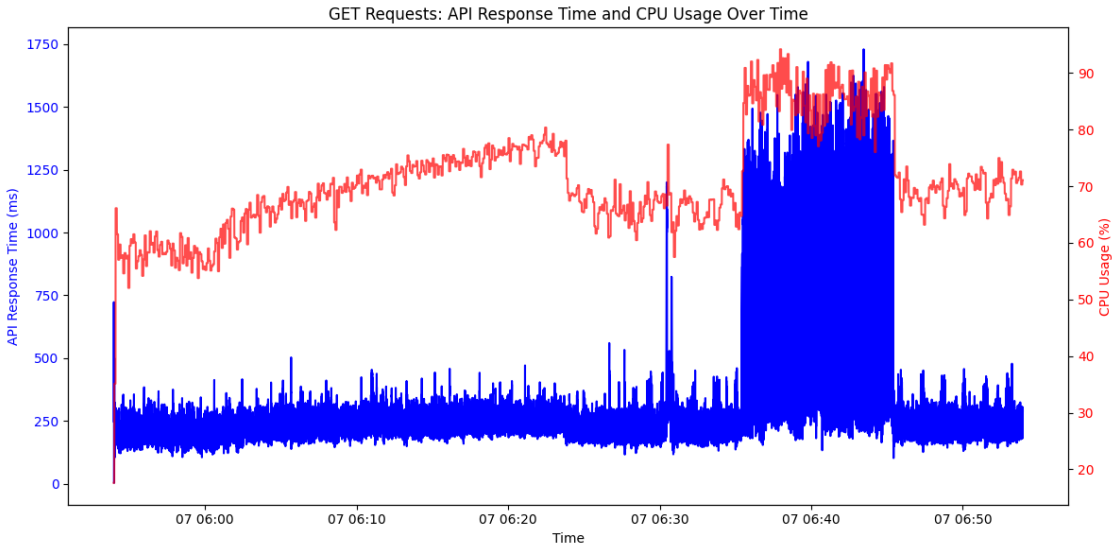

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
Timestamp
07 04:30
07 04:40
07 04:50
07 05:00
07 05:10
07 05:20
07 05:30
07 05:40
07 05:50

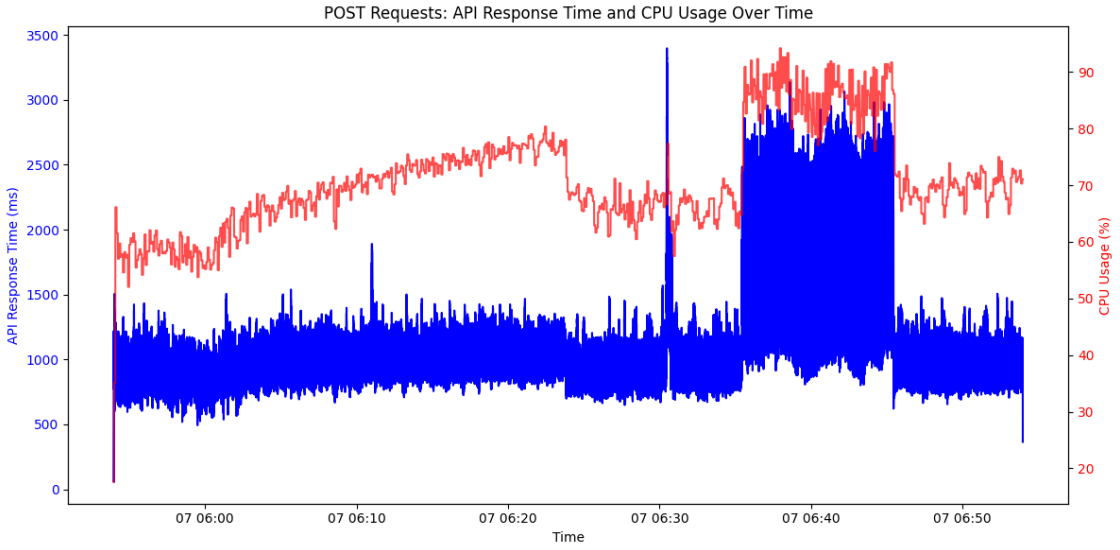

Distribution of CPU Usage
Frequency
CPU Usage (%)

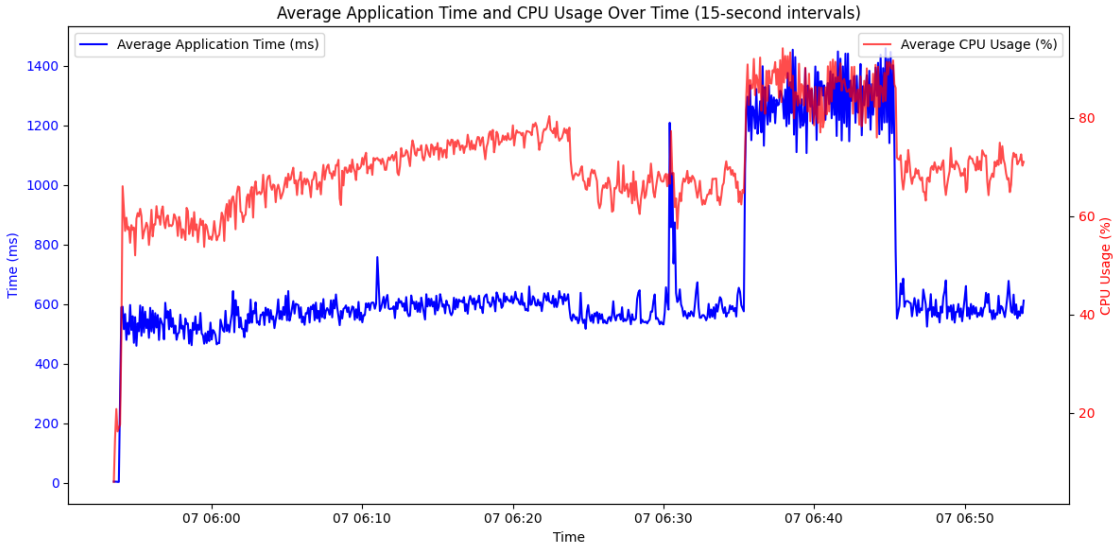


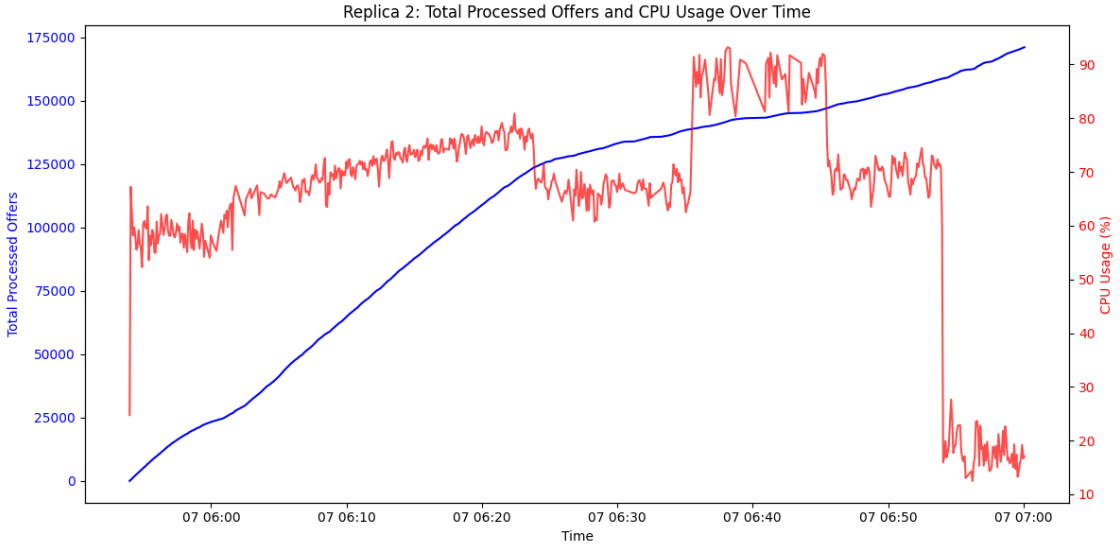


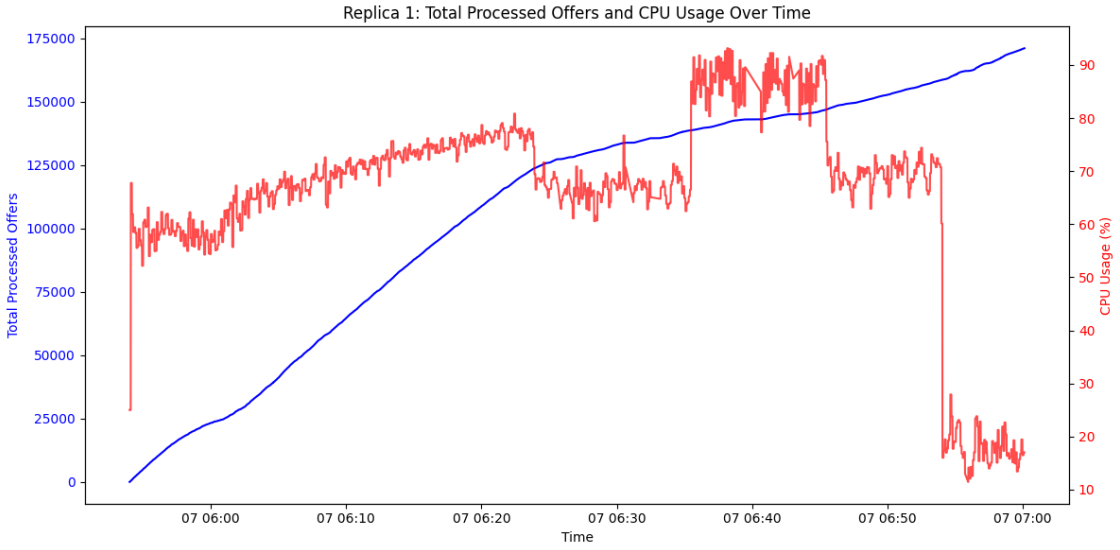

```
Loaded data from c:\Users\luki_\Desktop\logs\test2\4 3 200 500 4 500 100
successfully.
```

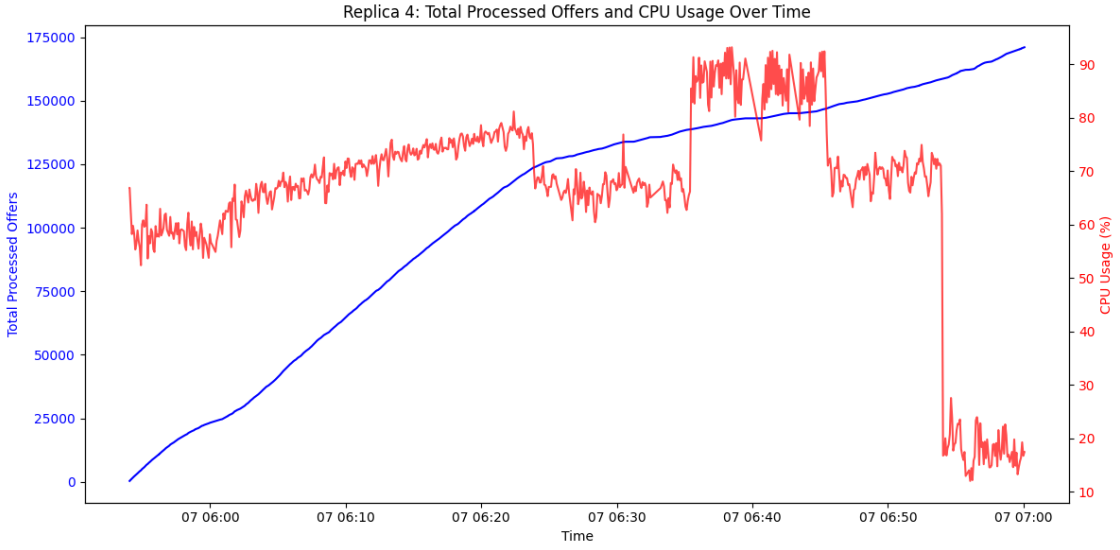

GET Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
250
500
750
1000
1250
1500
1750
20
30
40
50
60
70
80
90
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50

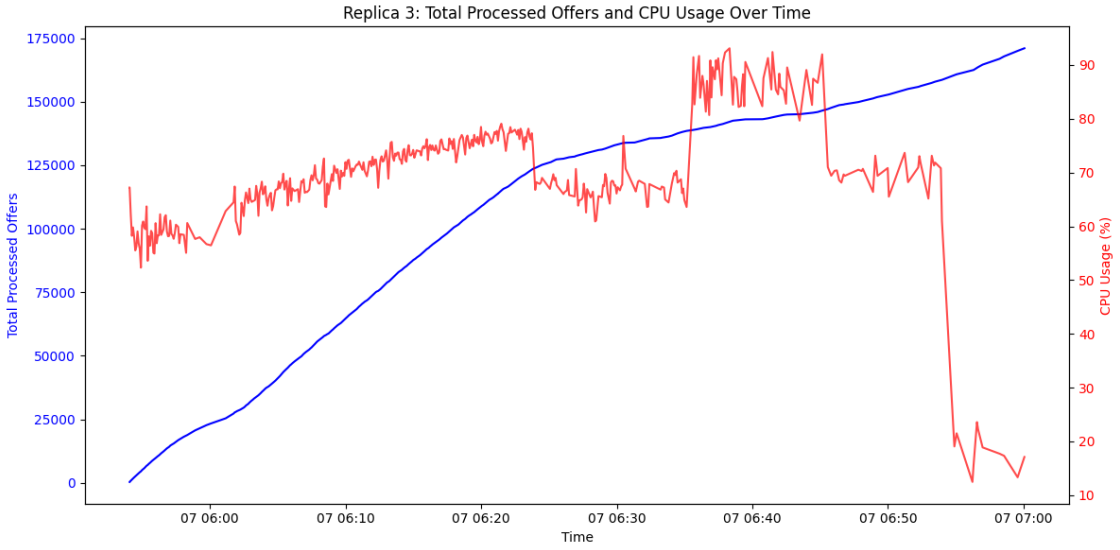

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
500
1000
1500
2000
2500
3000
3500
20
30
40
50
60
70
80
90
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50

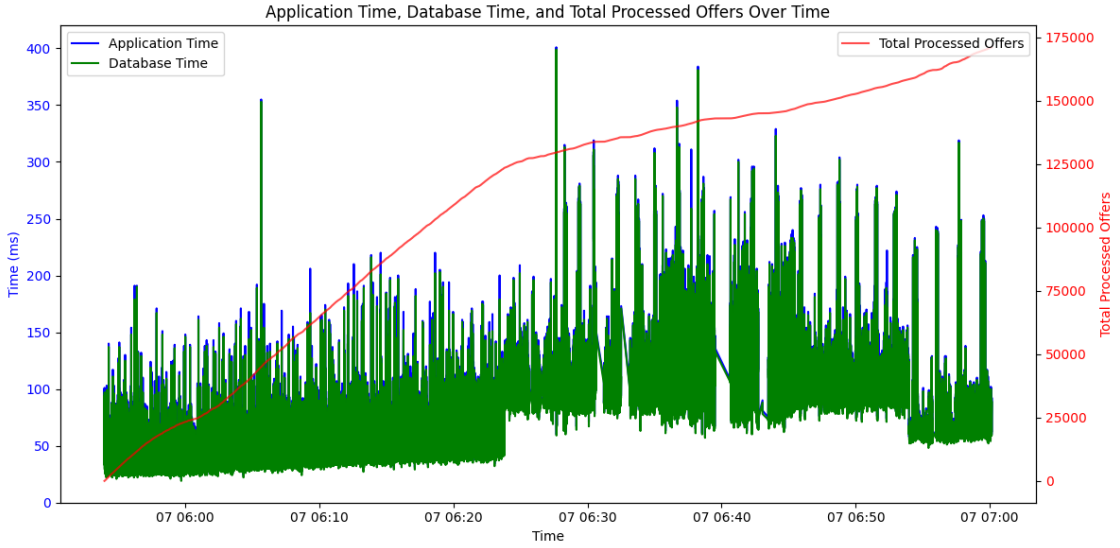

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
0
200
400
600
800
1000
1200
1400
20
40
60
80
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
Time

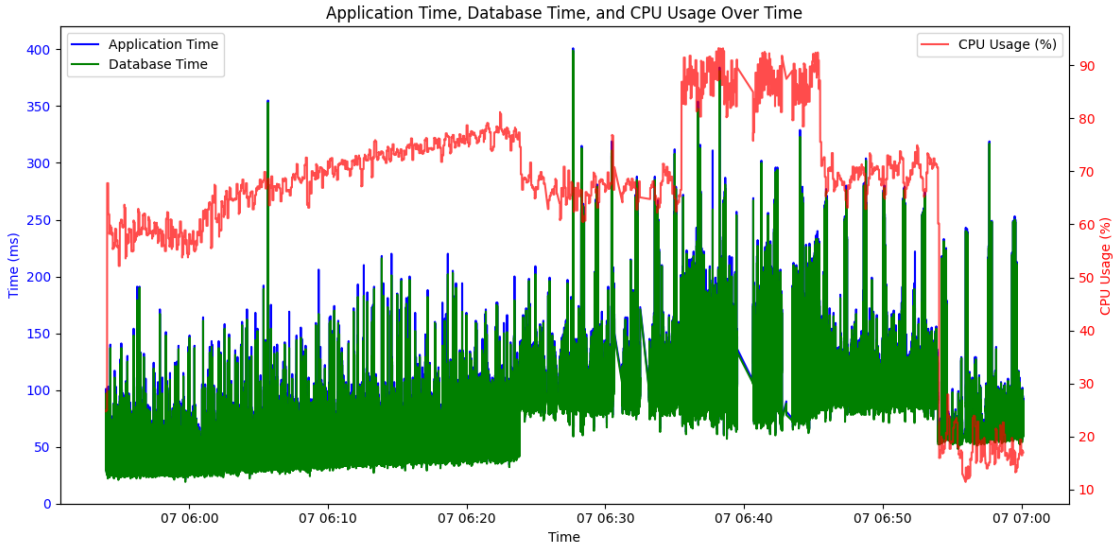

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
0
25000
50000
75000
100000
125000
150000
175000
10
20
30
40
50
60
70
80
90
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
07 07:00
Time

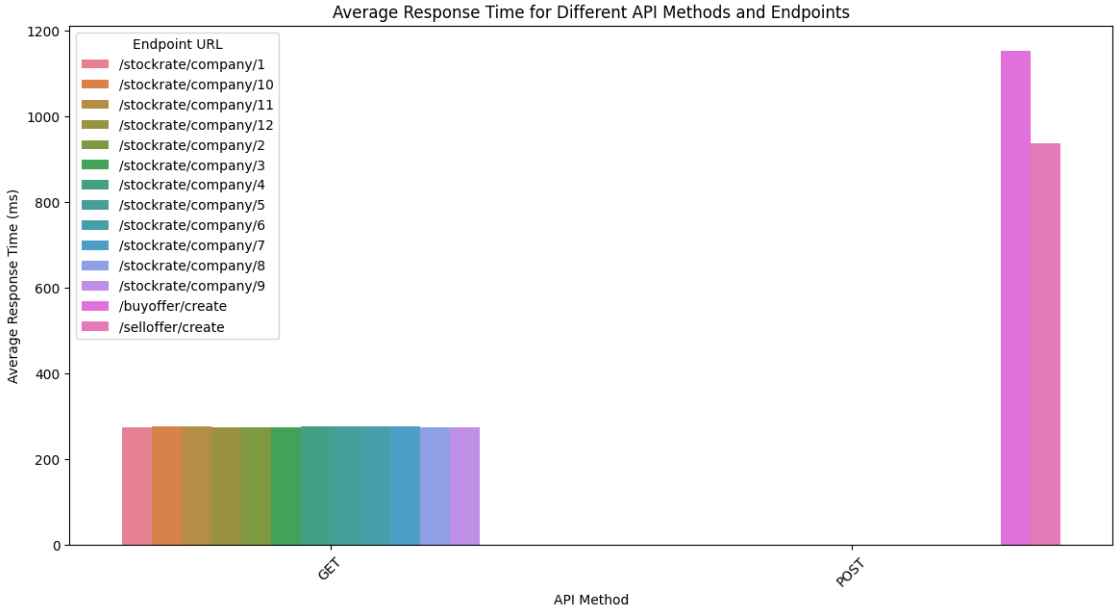

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
25000
50000
75000
100000
125000
150000
175000
10
20
30
40
50
60
70
80
90
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
07 07:00

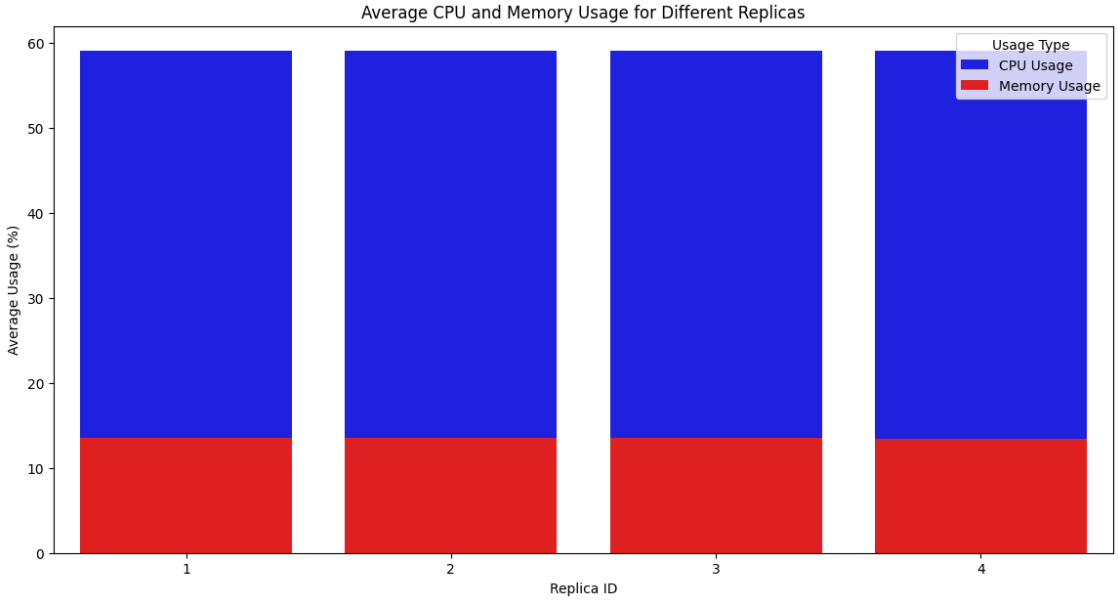

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
25000
50000
75000
100000
125000
150000
175000
10
20
30
40
50
60
70
80
90
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
07 07:00

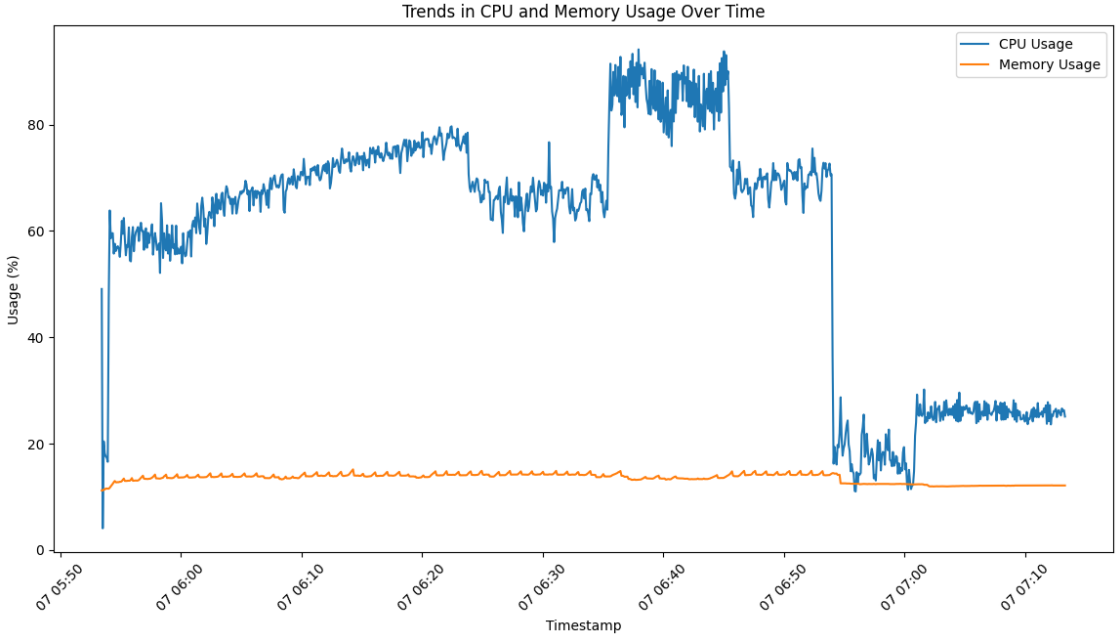

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
175000
150000
125000
100000
75000
50000
25000
0
90
80
70
60
50
40
30
20
10
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
07 07:00

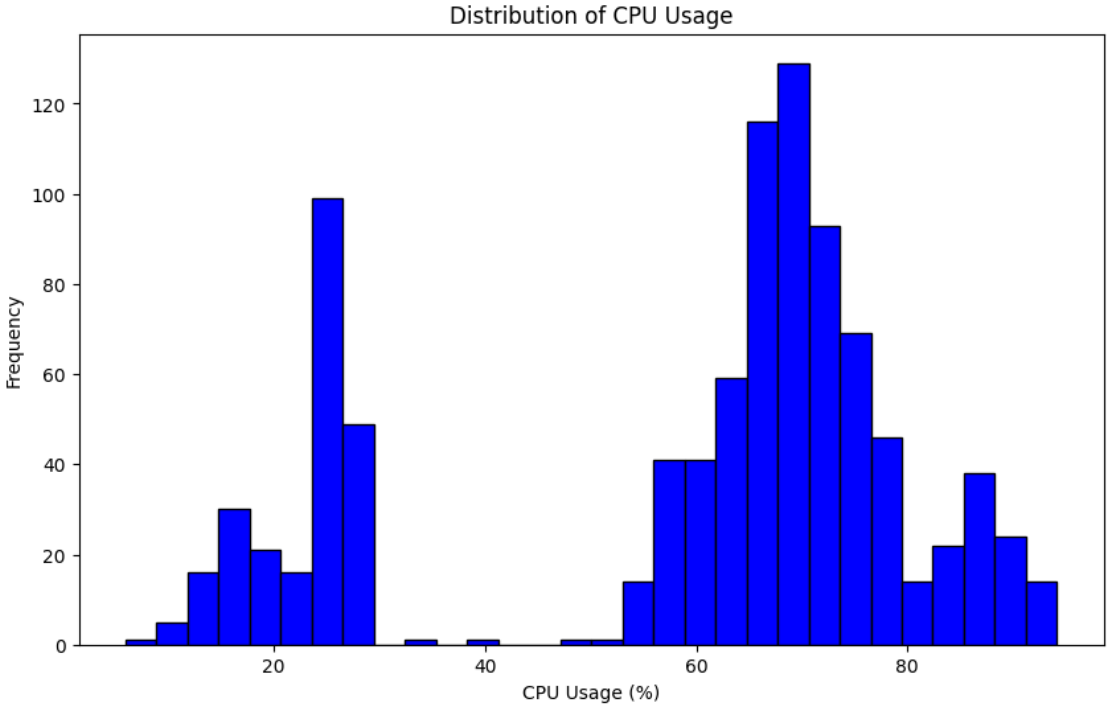

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
400
350
300
250
200
150
100
50
0
175000
150000
125000
100000
75000
50000
25000
0
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
07 07:00

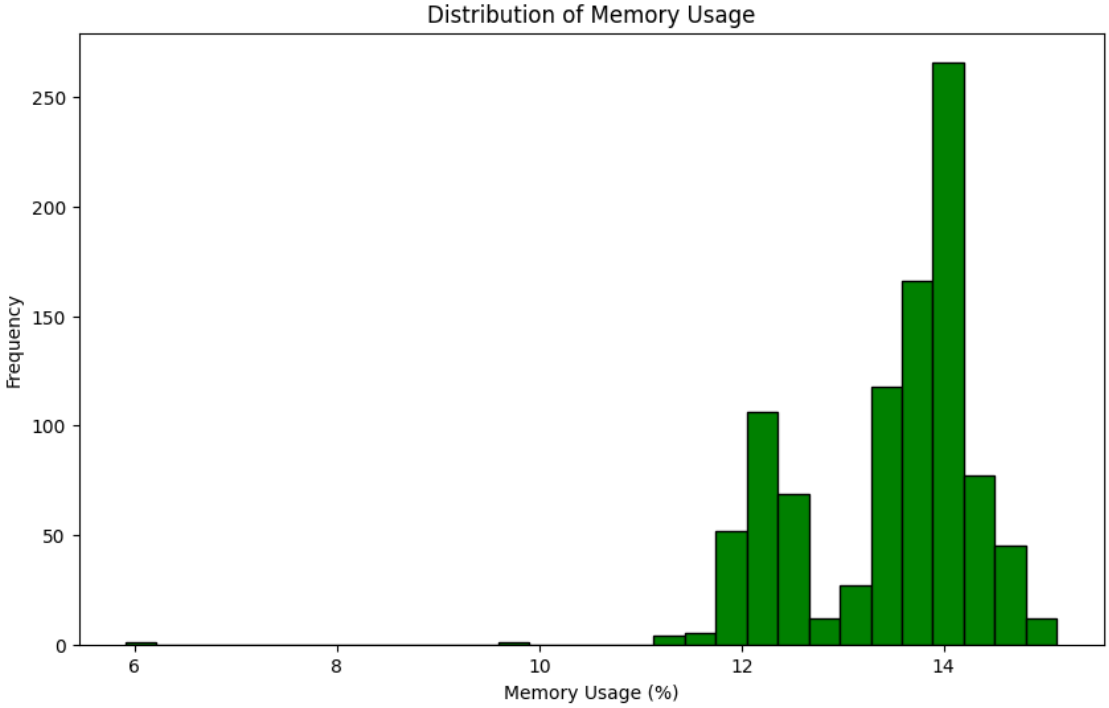

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
50
100
150
200
250
300
350
400
10
20
30
40
50
60
70
80
90
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
07 07:00

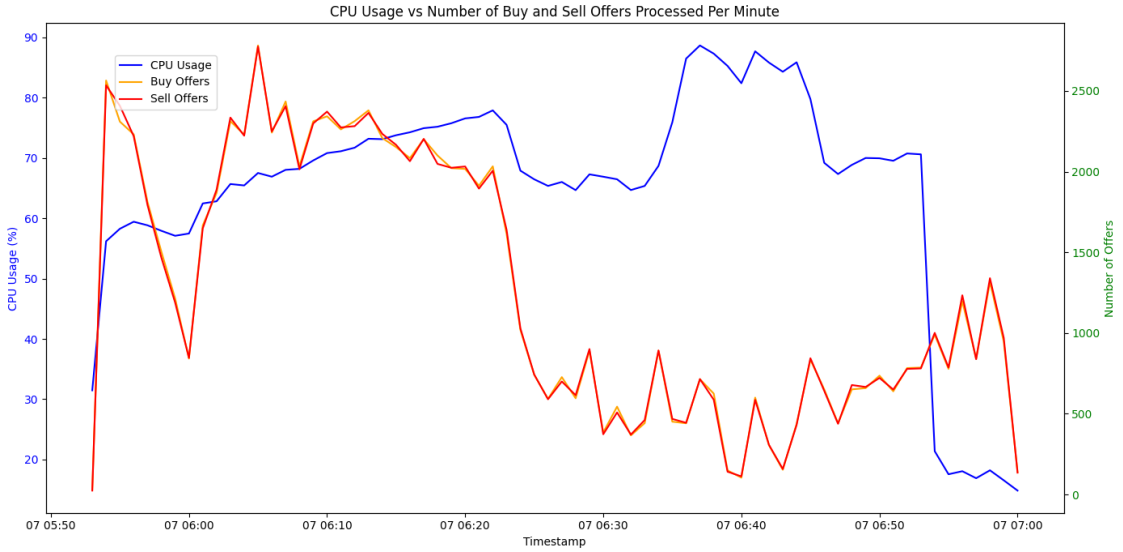

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
Replica ID
0
10
20
30
40
50
60
1
2
3
4

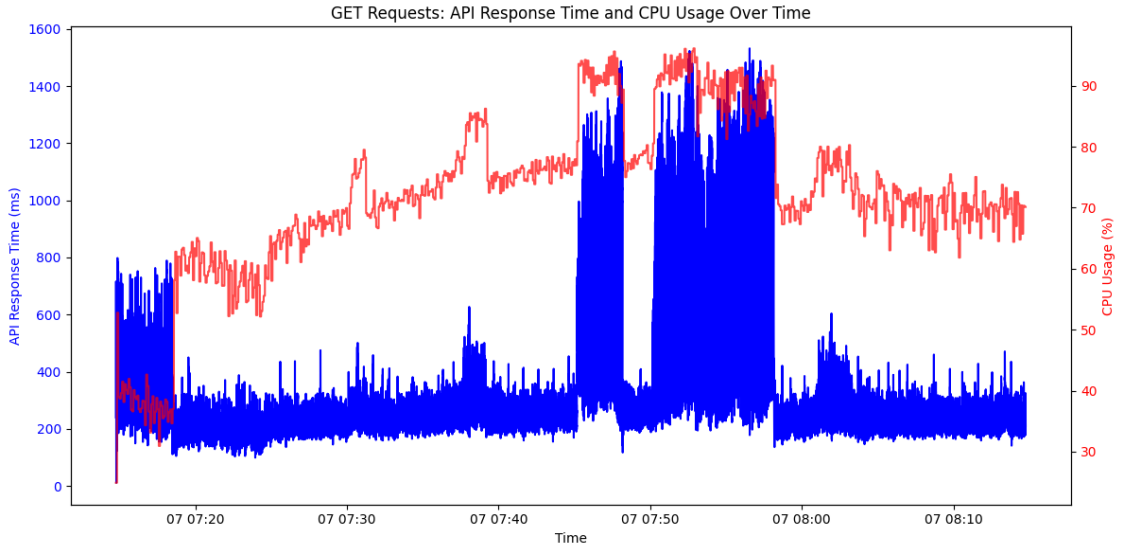

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
Timestamp
0
20
40
60
80
07 05:50
07 06:00
07 06:10
07 06:20
07 06:30
07 06:40
07 06:50
07 07:00
07 07:10

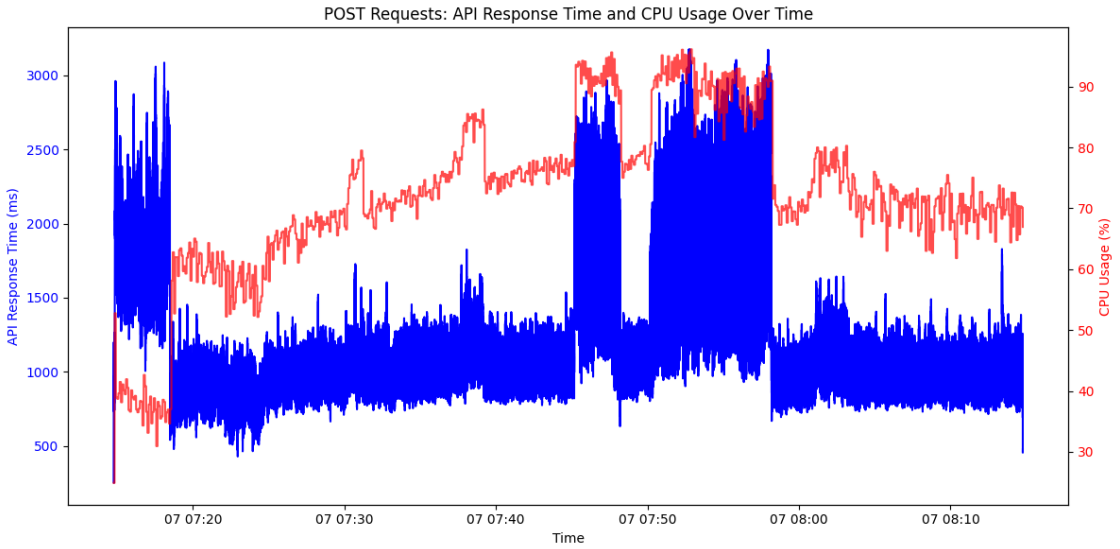

Distribution of CPU Usage
Frequency
CPU Usage (%)
0
20
40
60
80
100
120

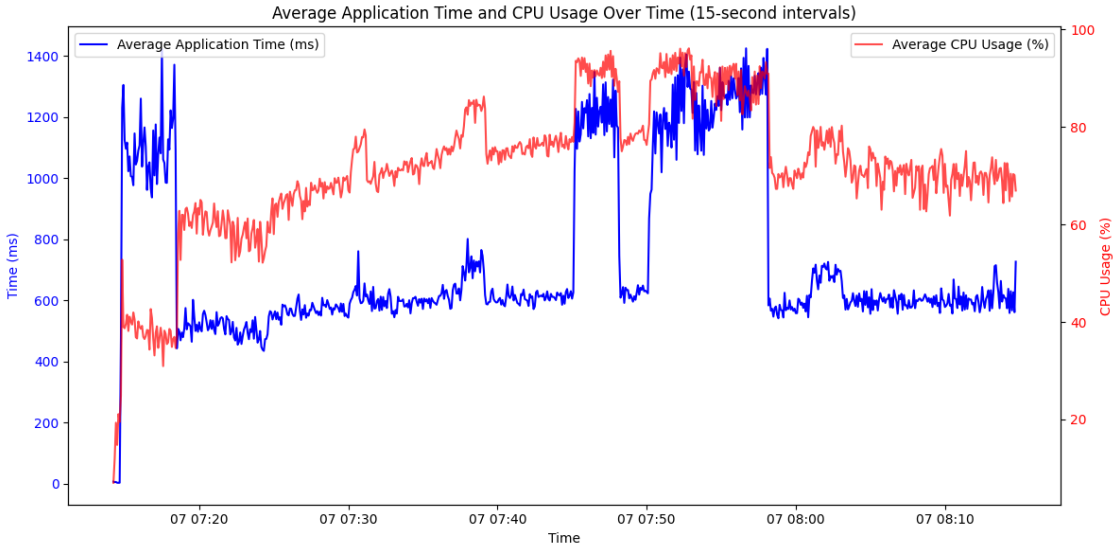

Distribution of Memory Usage
Frequency
Memory Usage (%)
0
50
100
150
200
250
6
8
10
12
14

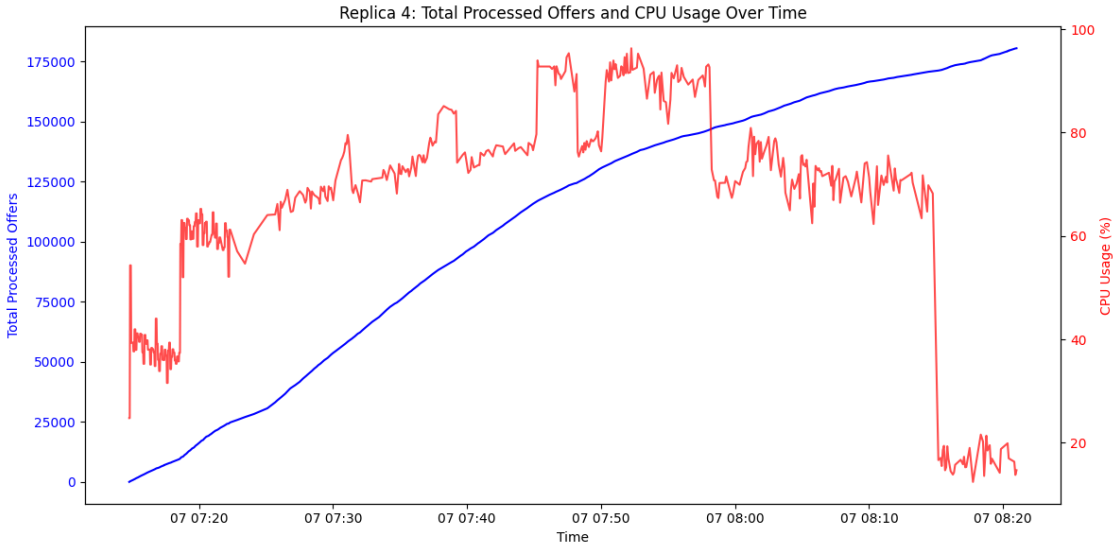


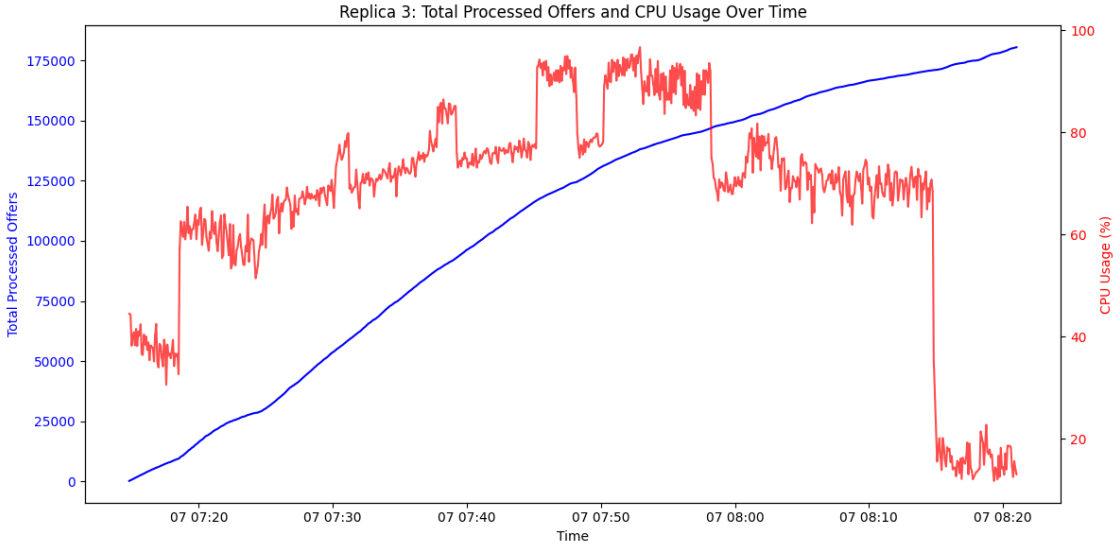

```
Loaded data from c:\Users\luki_\Desktop\logs\test2\4 3 200 500 8 500 100
successfully.
```

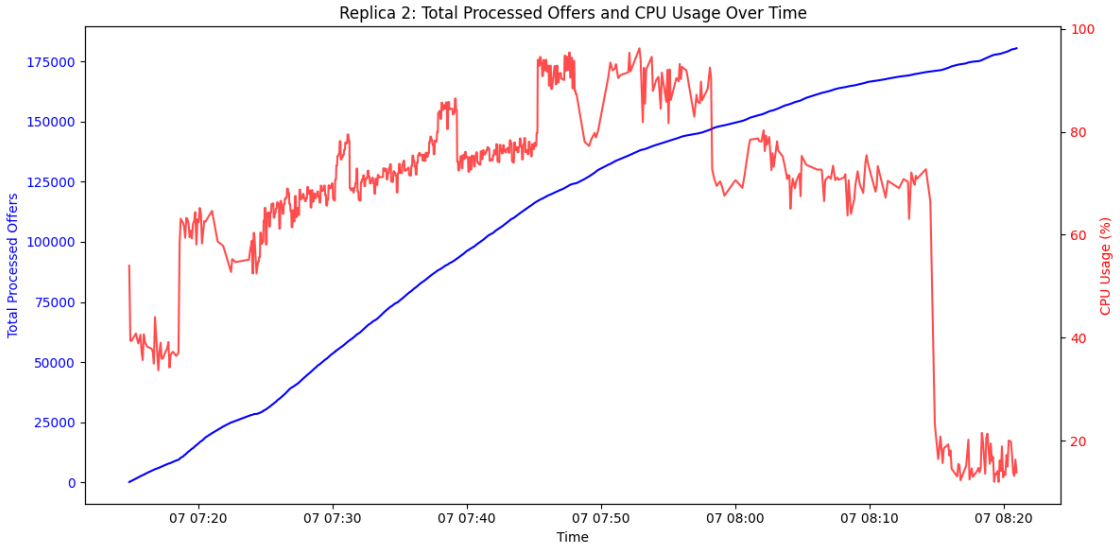


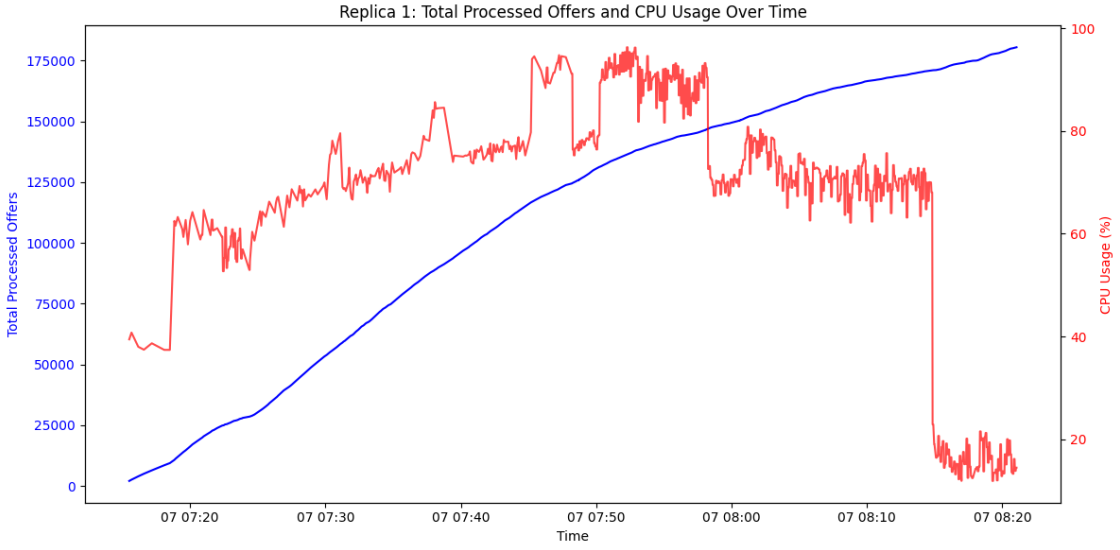

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
3000
2500
2000
1500
1000
500
90
80
70
60
50
40
30
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10

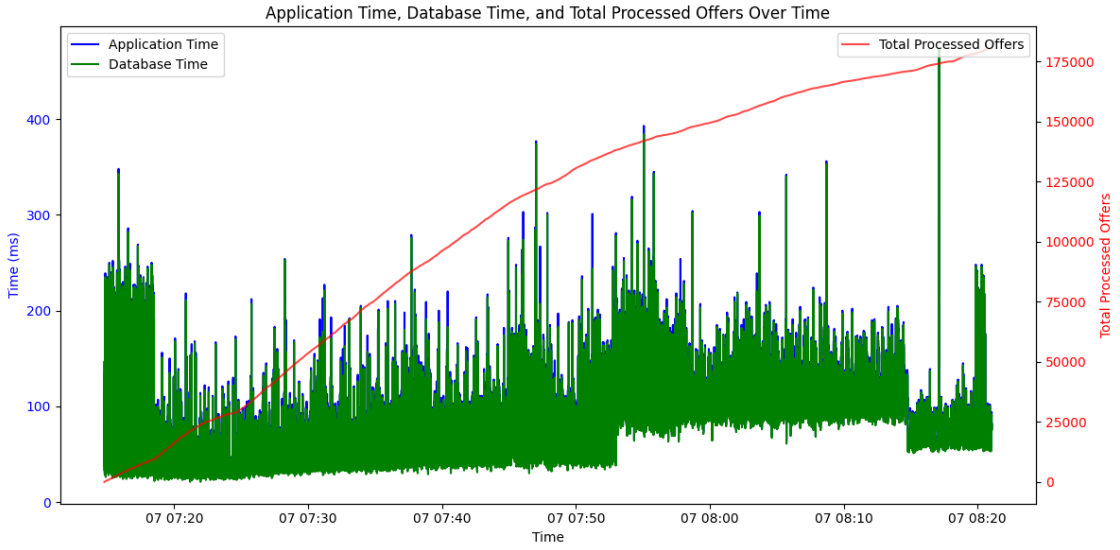

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
1400
1200
1000
800
600
400
200
0
100
80
60
40
20
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10

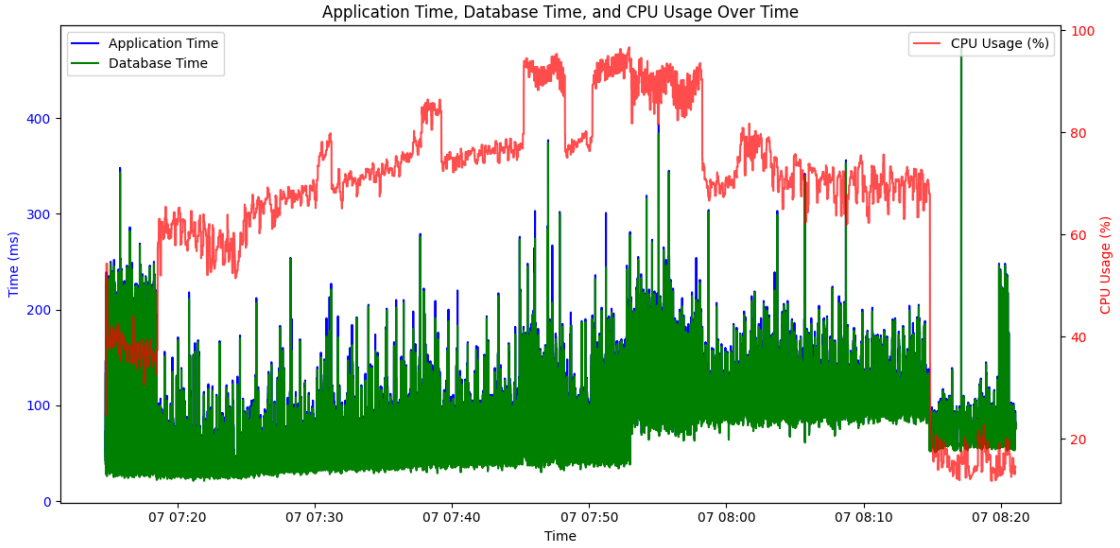

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
25000
50000
75000
100000
125000
150000
175000
20
40
60
80
100
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10
07 08:20

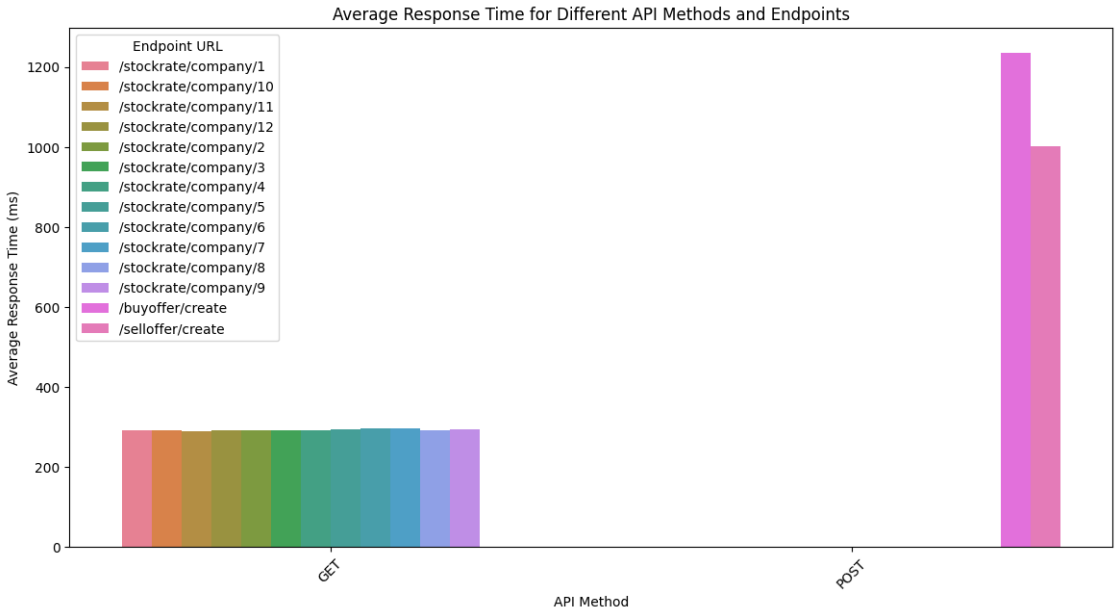

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
25000
50000
75000
100000
125000
150000
175000
20
40
60
80
100
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10
07 08:20

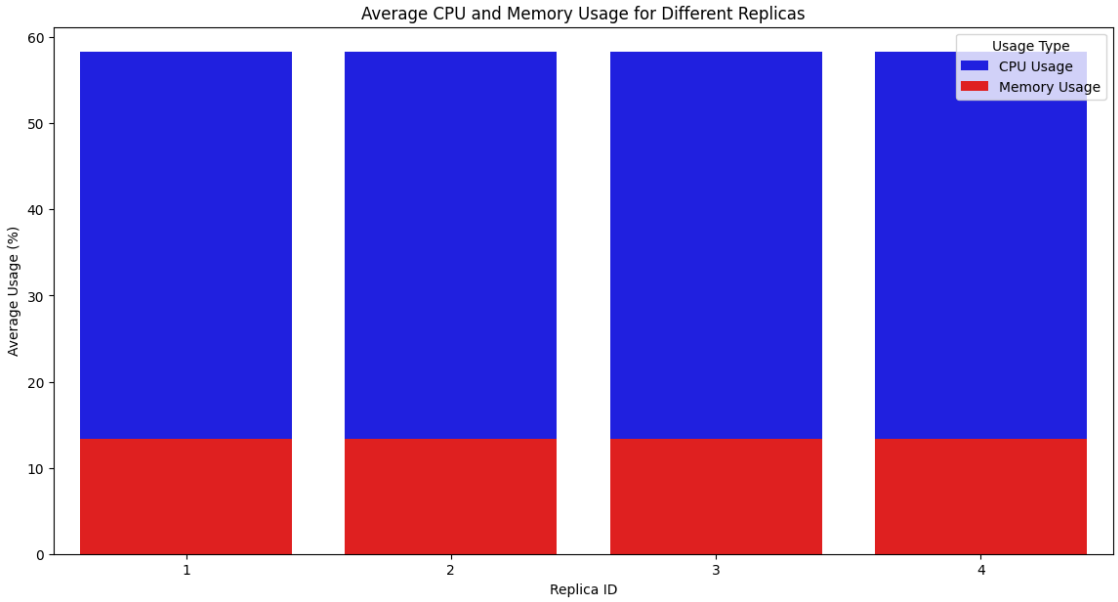

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
25000
50000
75000
100000
125000
150000
175000
20
40
60
80
100
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10
07 08:20

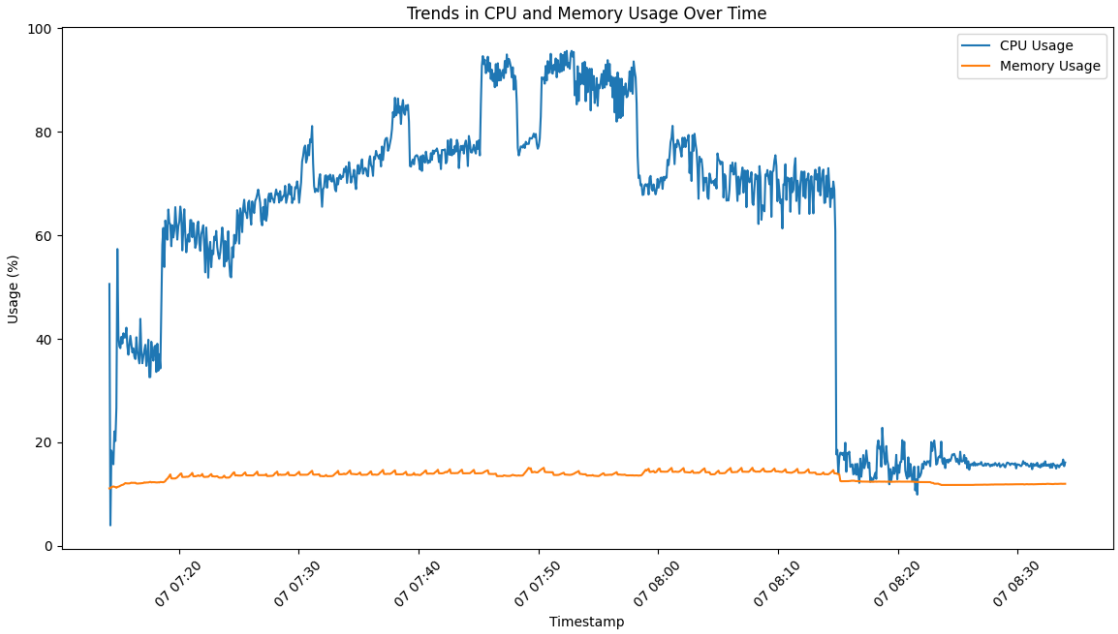

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
25000
50000
75000
100000
125000
150000
175000
20
40
60
80
100
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10
07 08:20

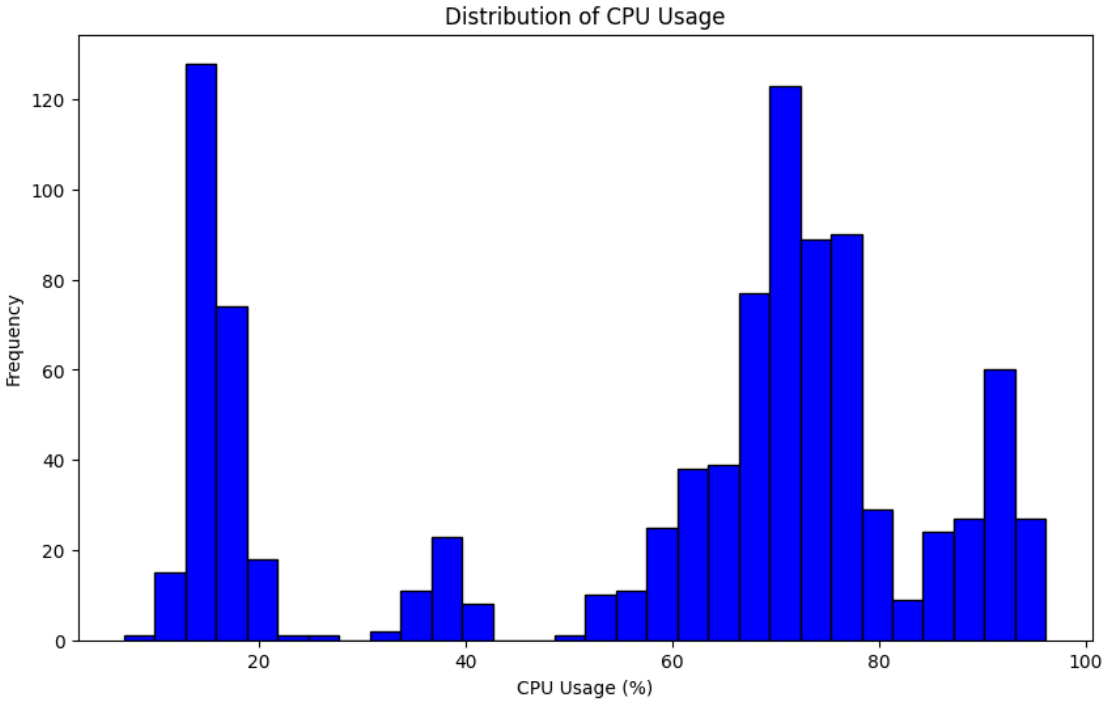

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
0
100
200
300
400
0
25000
50000
75000
100000
125000
150000
175000
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10
07 08:20

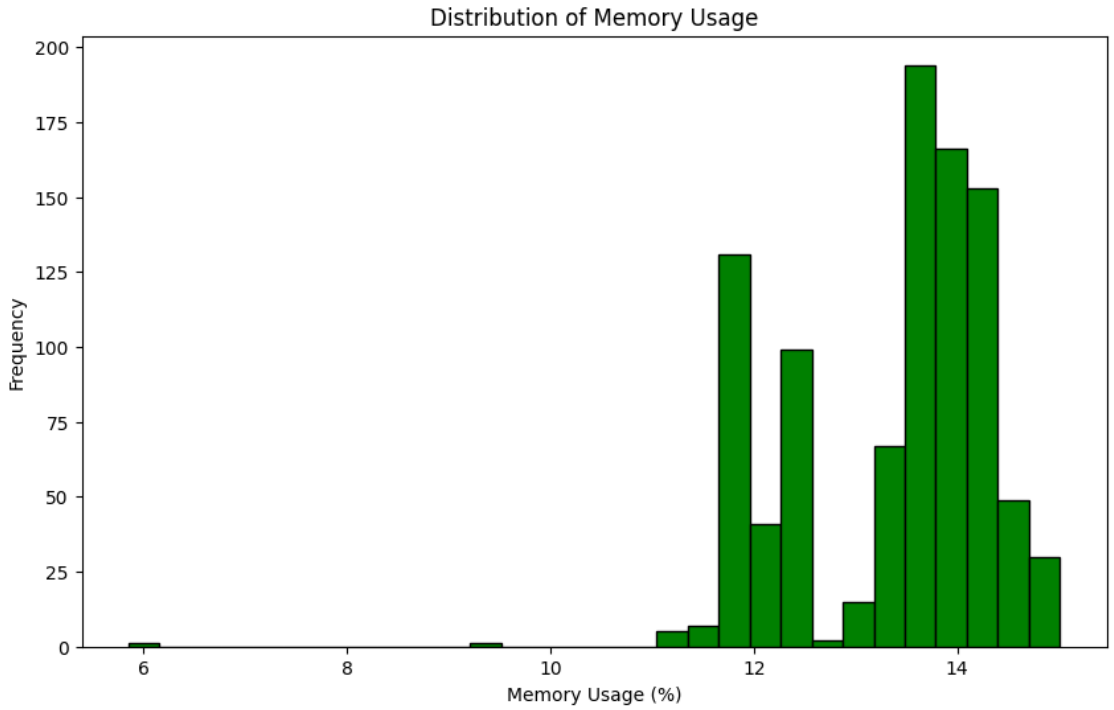

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
100
200
300
400
20
40
60
80
100
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10
07 08:20

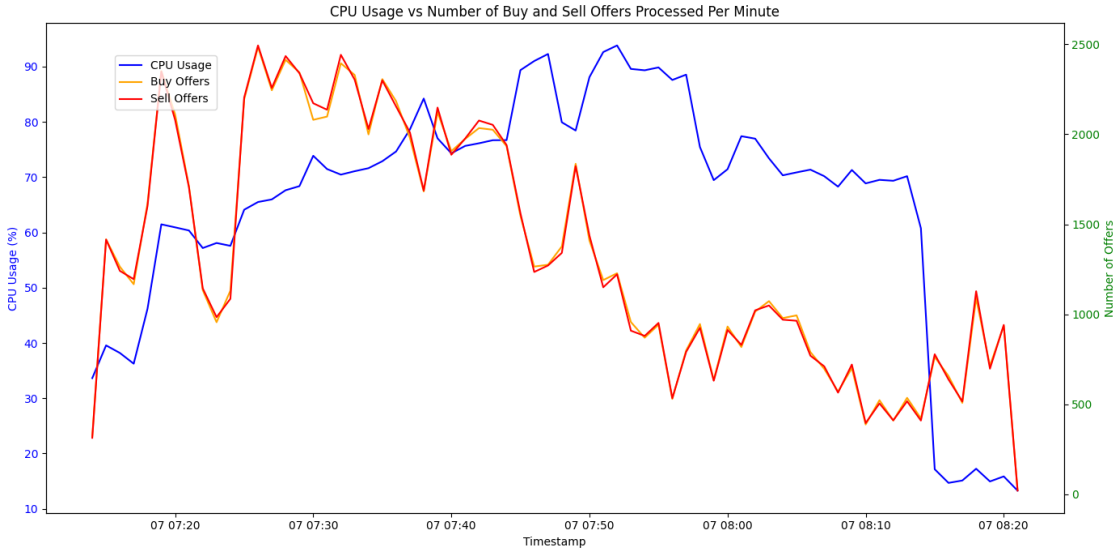

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
10
20
30
40
50
60
1
2
3
4
Replica ID

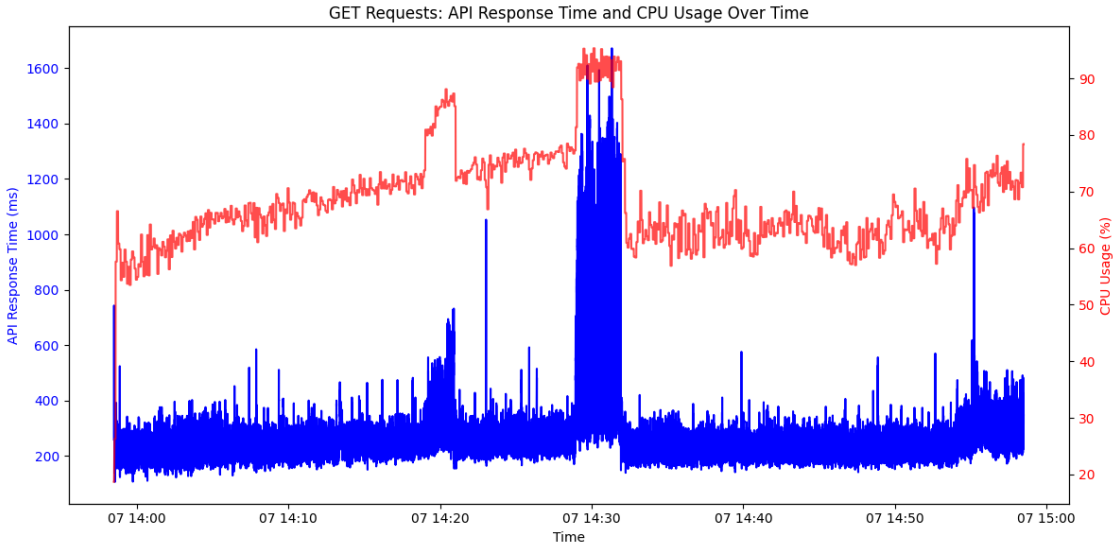

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
Timestamp
0
20
40
60
80
100
07 07:20
07 07:30
07 07:40
07 07:50
07 08:00
07 08:10
07 08:20
07 08:30

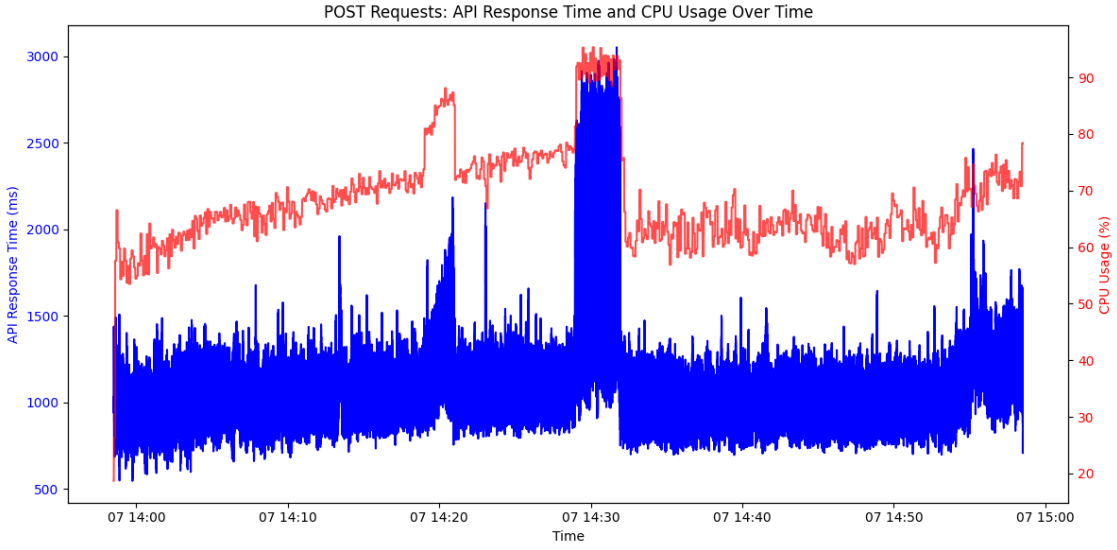

Distribution of CPU Usage
Frequency
CPU Usage (%)
0
20
40
60
80
100
120

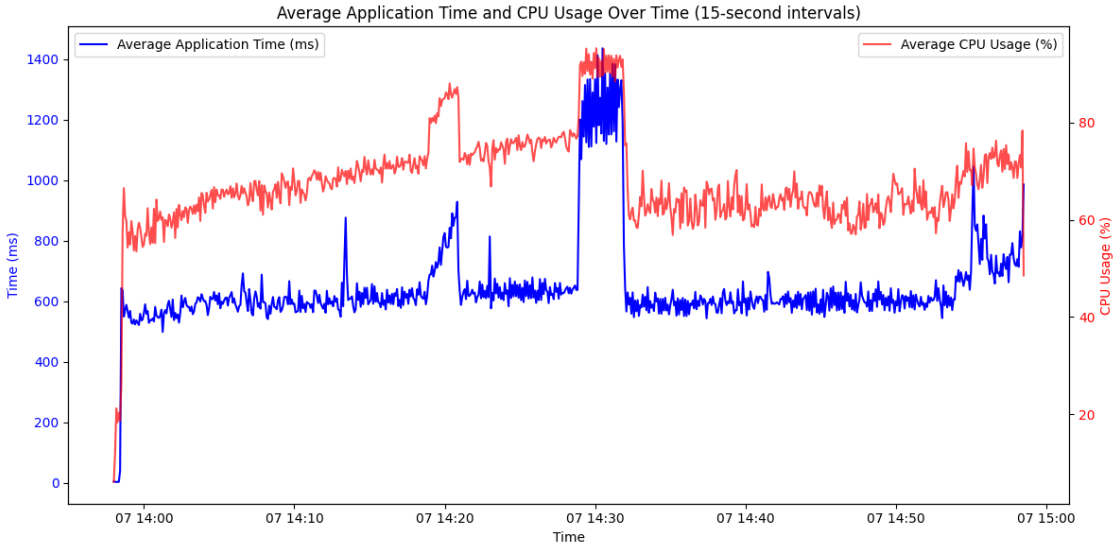


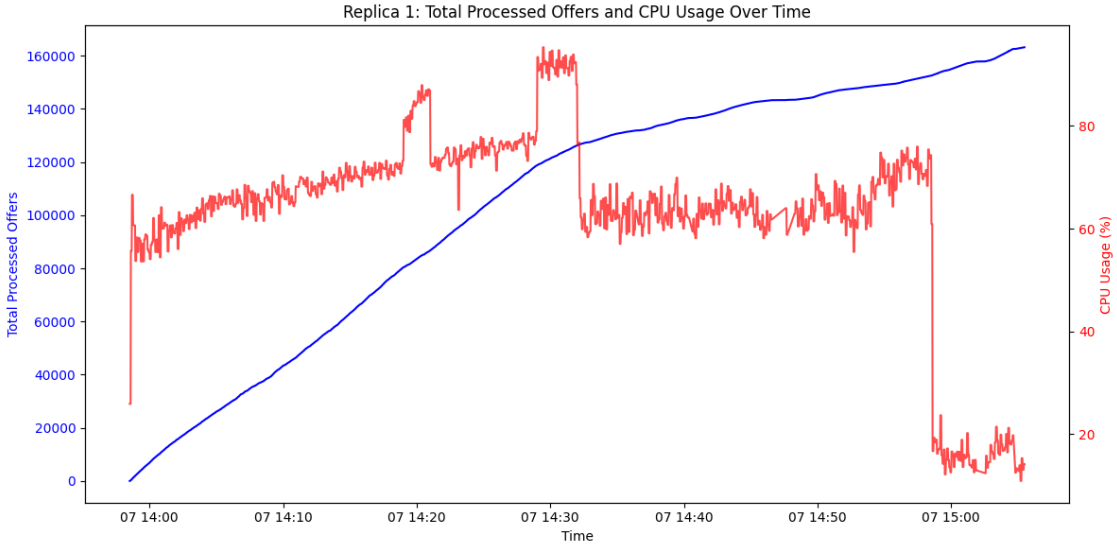


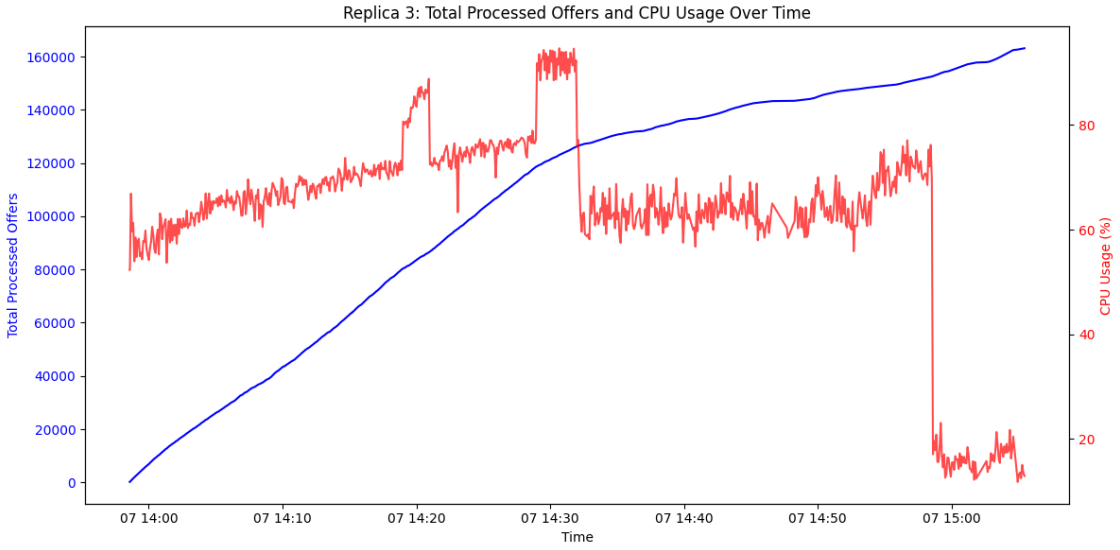

```
One or more CSV files are missing in directory:
c:\Users\luki_\Desktop\logs\test3
Loaded data from c:\Users\luki_\Desktop\logs\test3\4 3 200 1000 4 500 100
successfully.
```

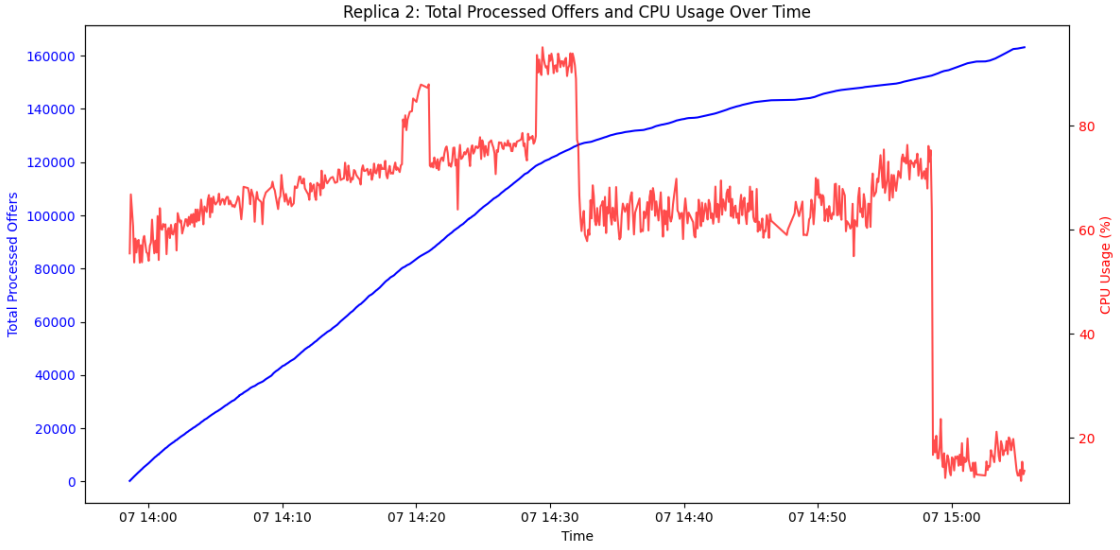

GET Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00

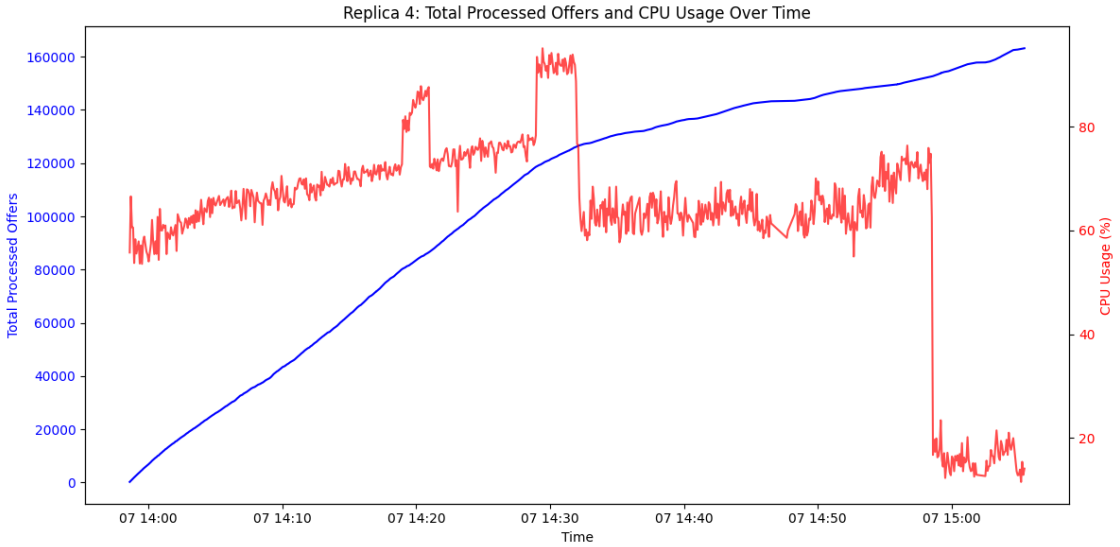

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00

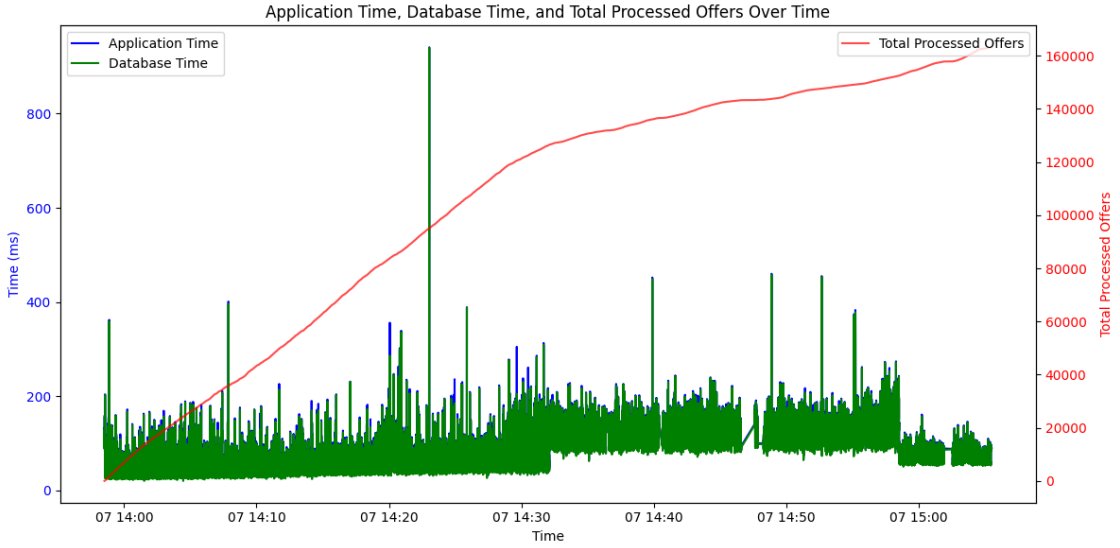

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
0
200
400
600
800
1000
1200
1400
20
40
60
80
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00
Time

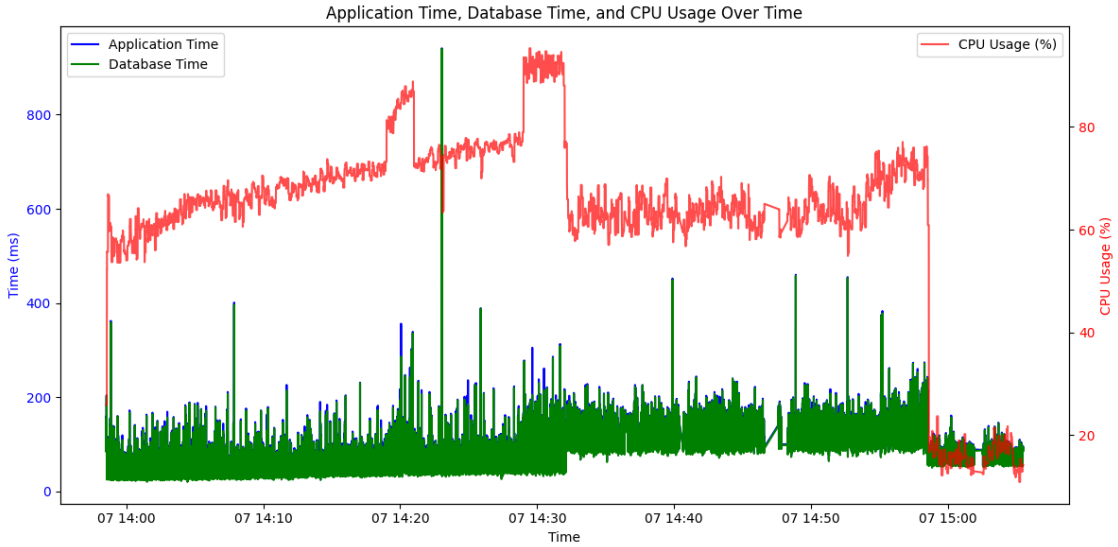

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00
Time

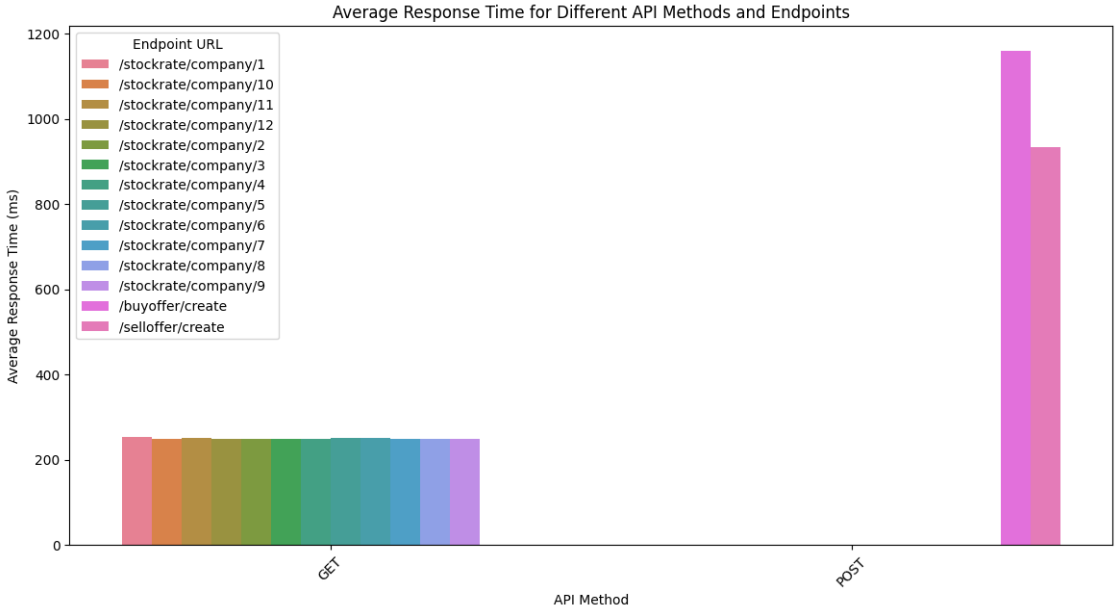

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00

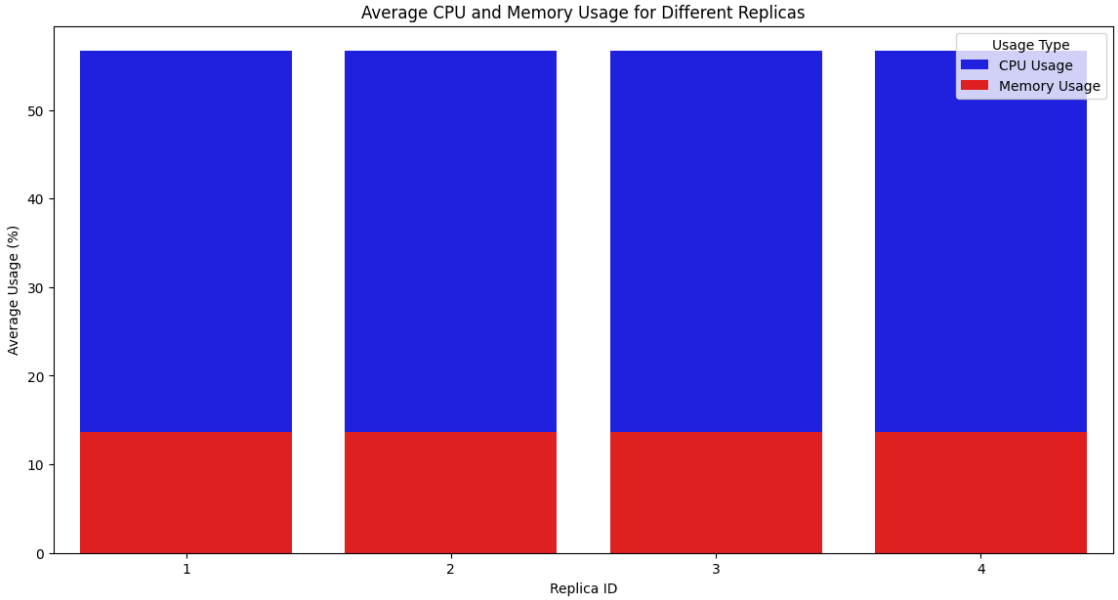

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00

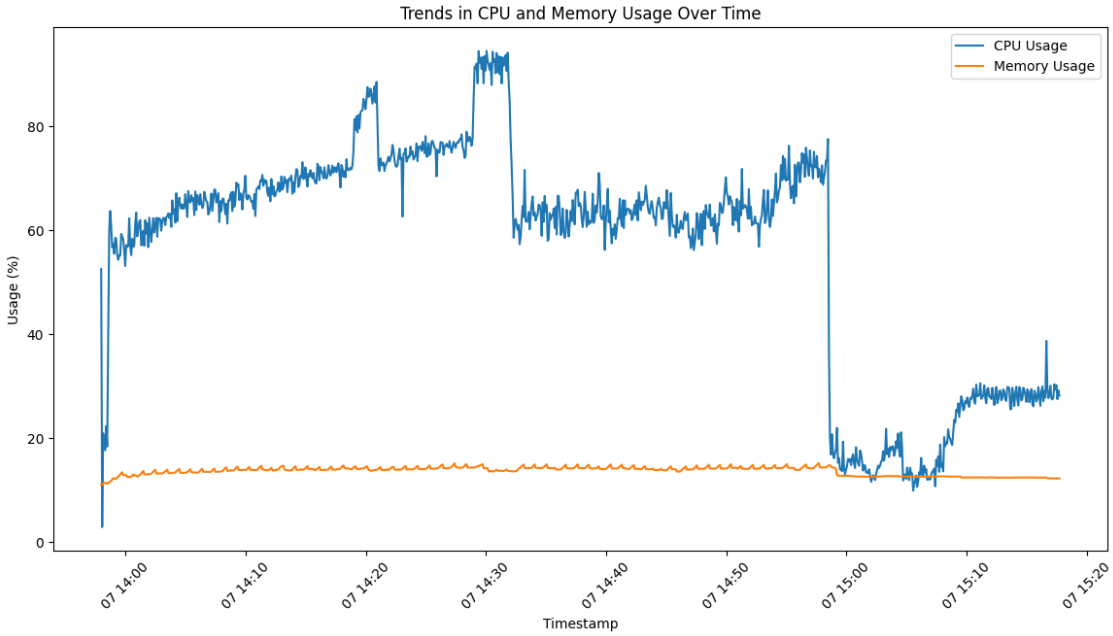

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00

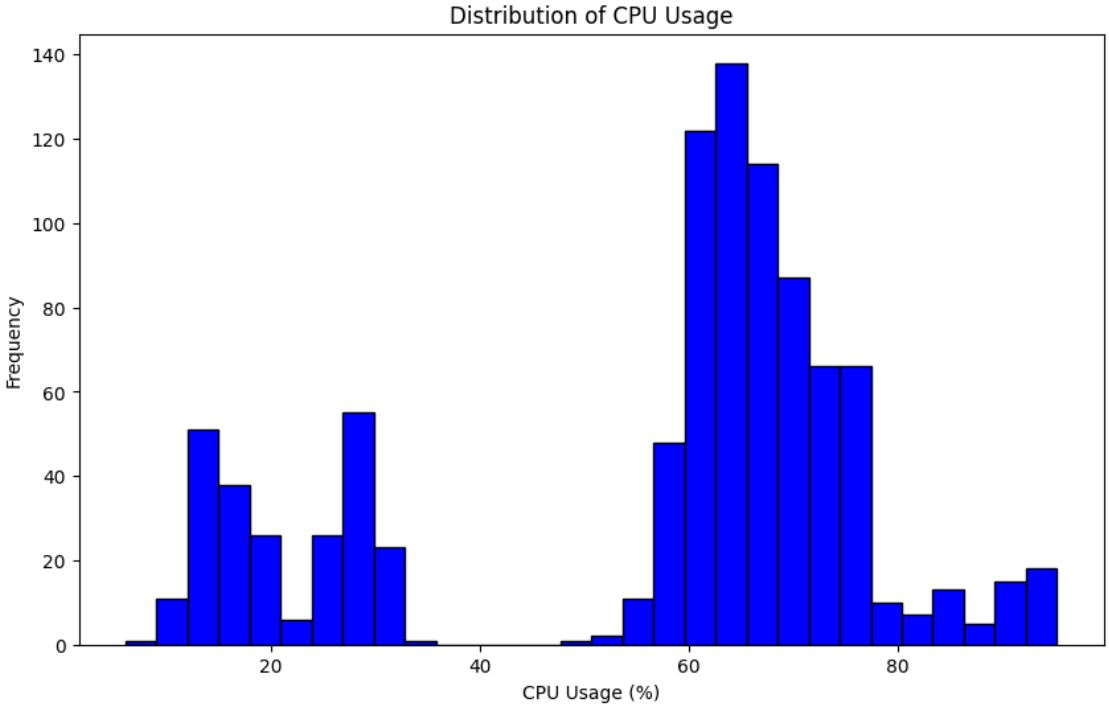

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
0
200
400
600
800
0
20000
40000
60000
80000
100000
120000
140000
160000
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00

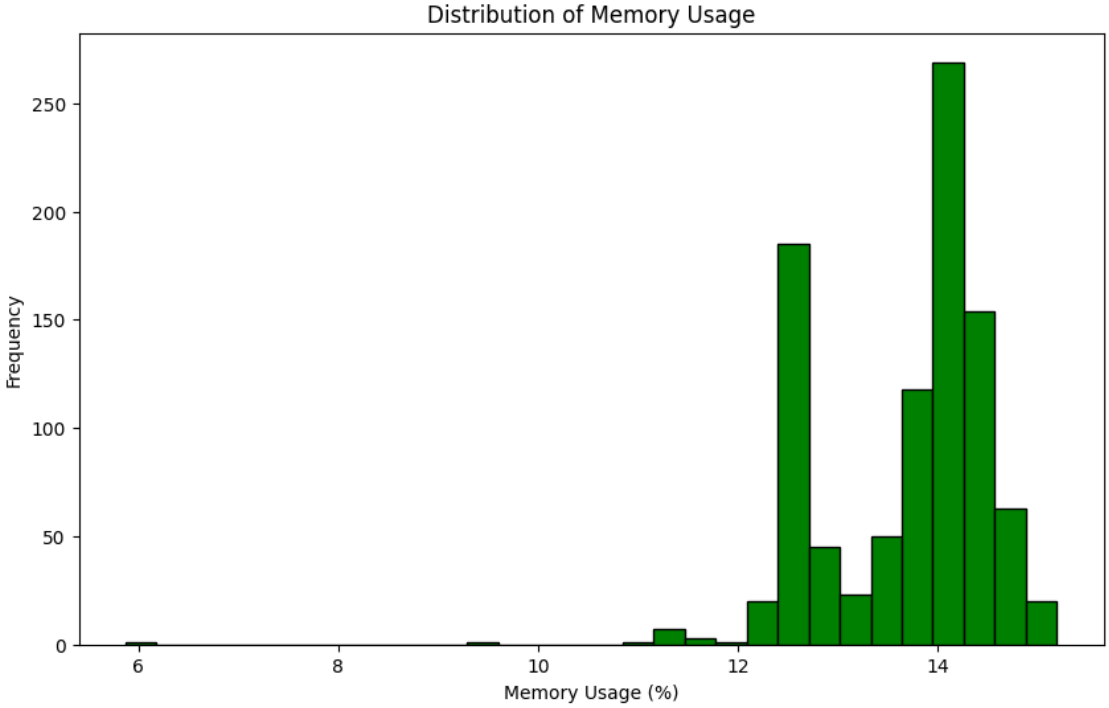

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
0
200
400
600
800
20
40
60
80
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00
Time

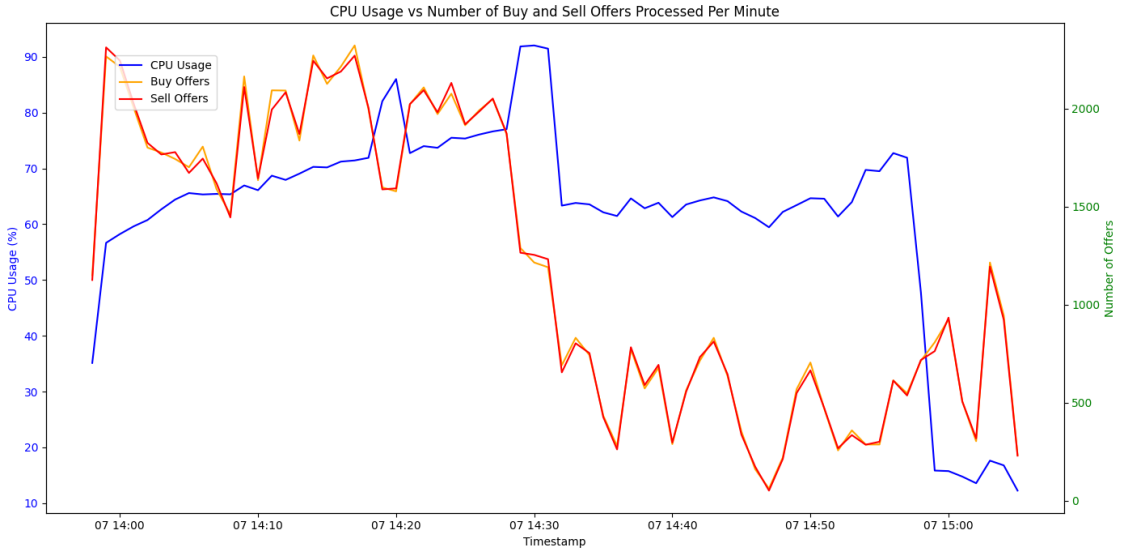

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
10
20
30
40
50
1
2
3
4
Replica ID

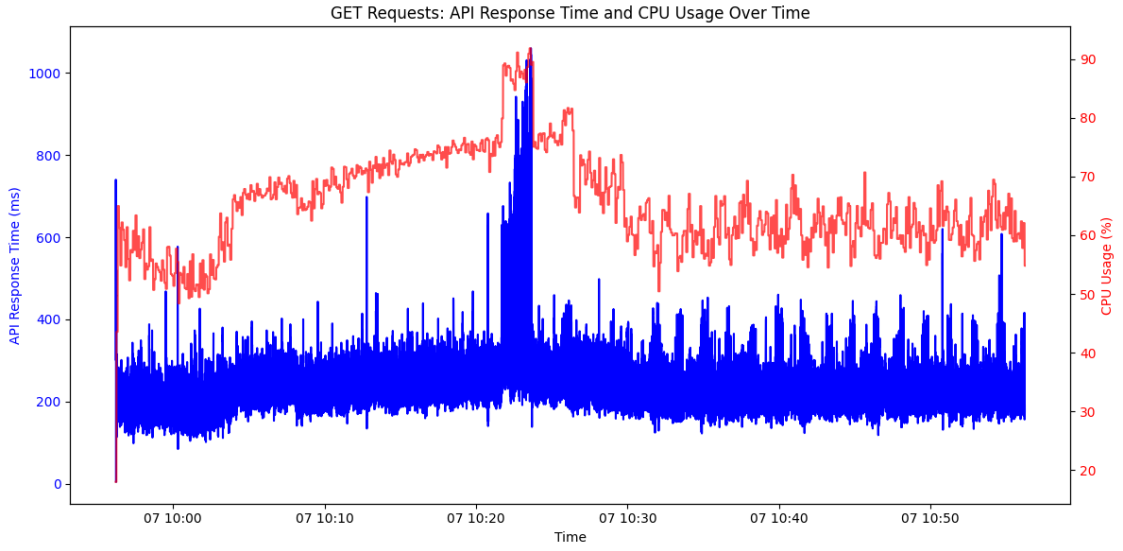

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
0
20
40
60
80
07 14:00
07 14:10
07 14:20
07 14:30
07 14:40
07 14:50
07 15:00
07 15:10
07 15:20
Timestamp

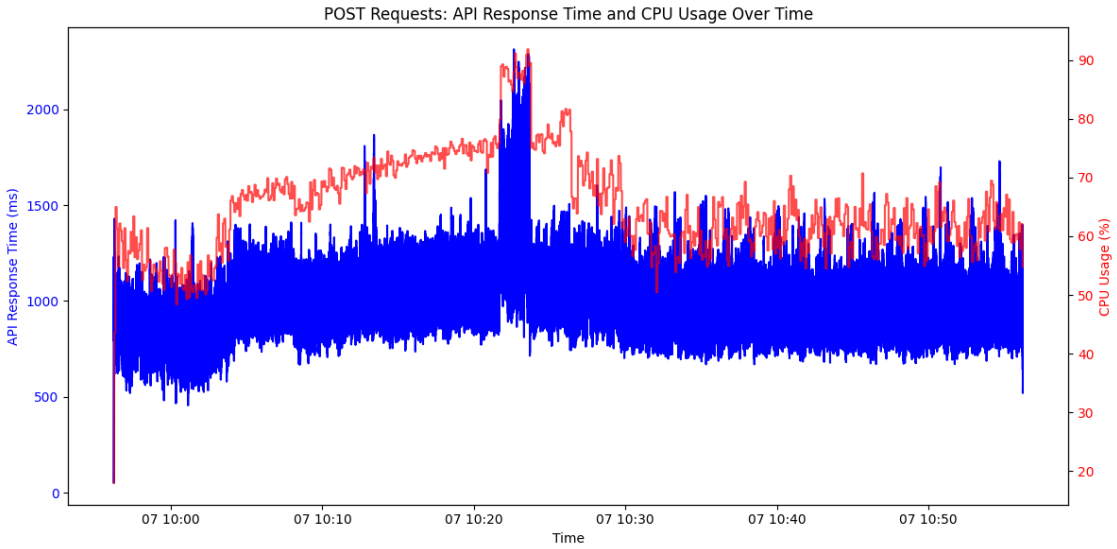

Distribution of CPU Usage
Frequency
CPU Usage (%)
0
20
40
60
80
100
120
140

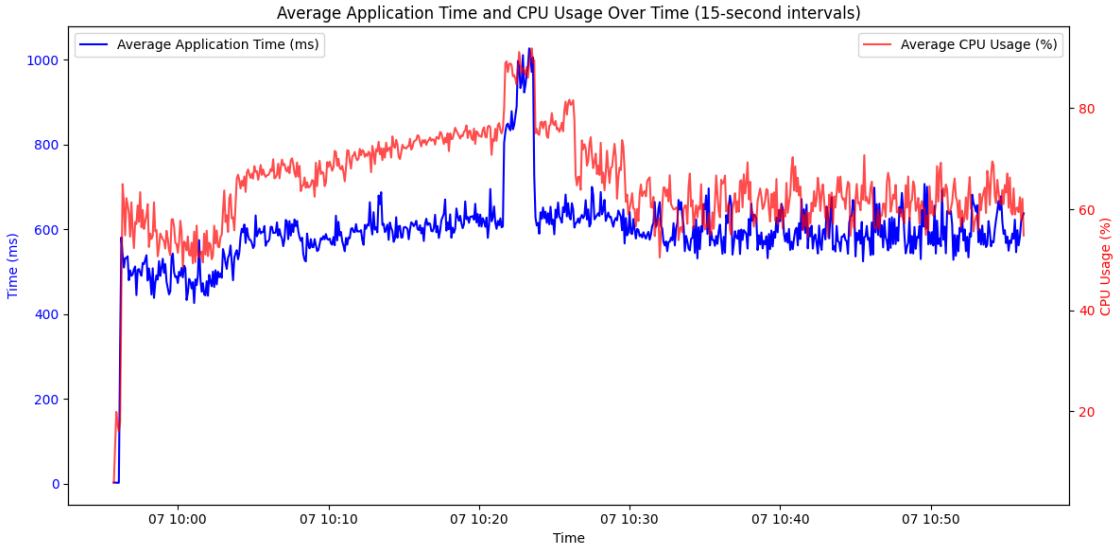

Distribution of Memory Usage
Frequency
Memory Usage (%)
0
50
100
150
200
250
6
8
10
12
14

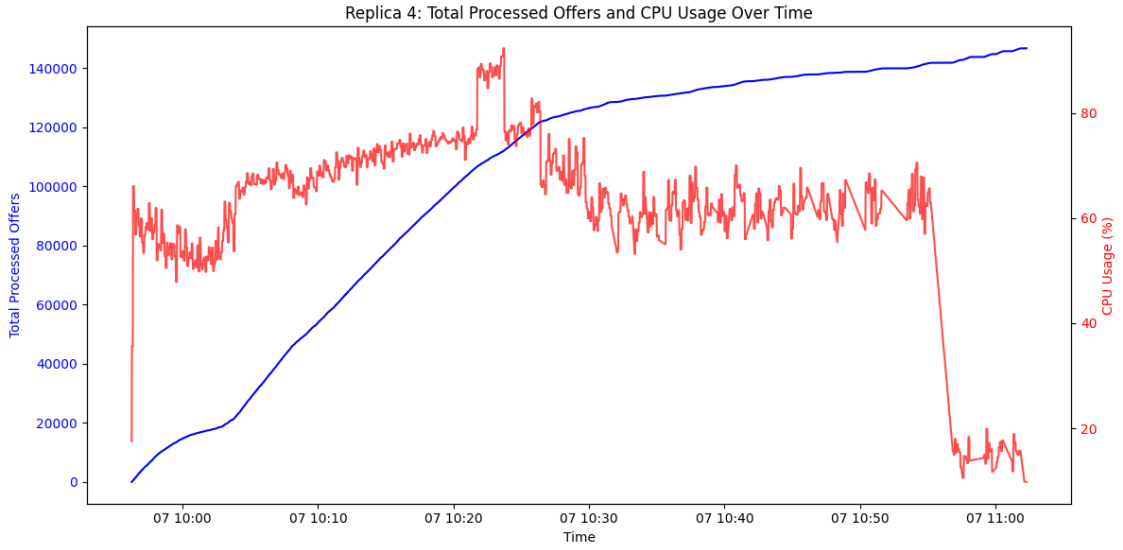


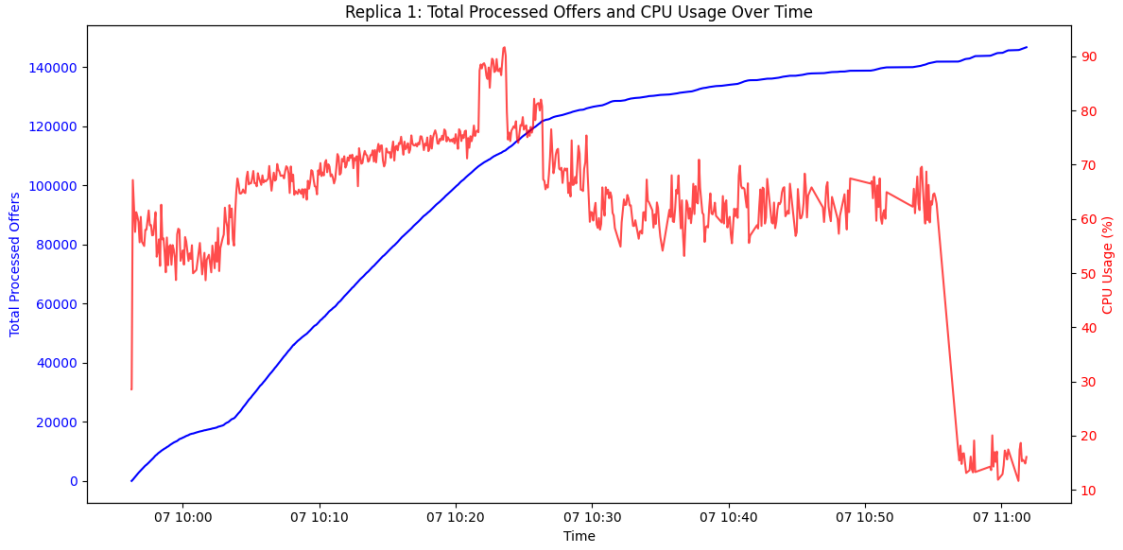

```
Loaded data from c:\Users\luki_\Desktop\logs\test3\4 3 200 250 4 500 100
successfully.
```

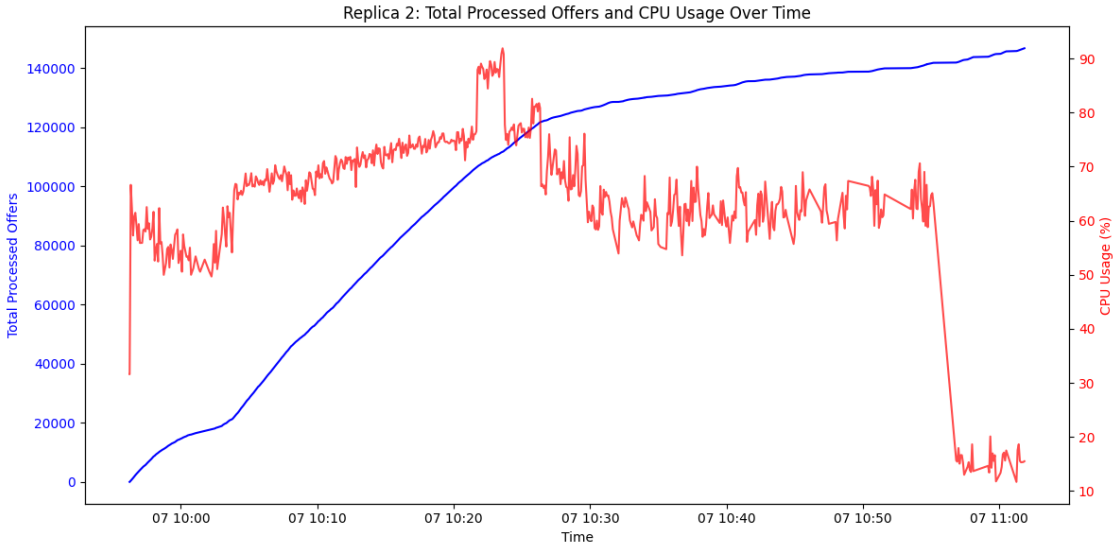


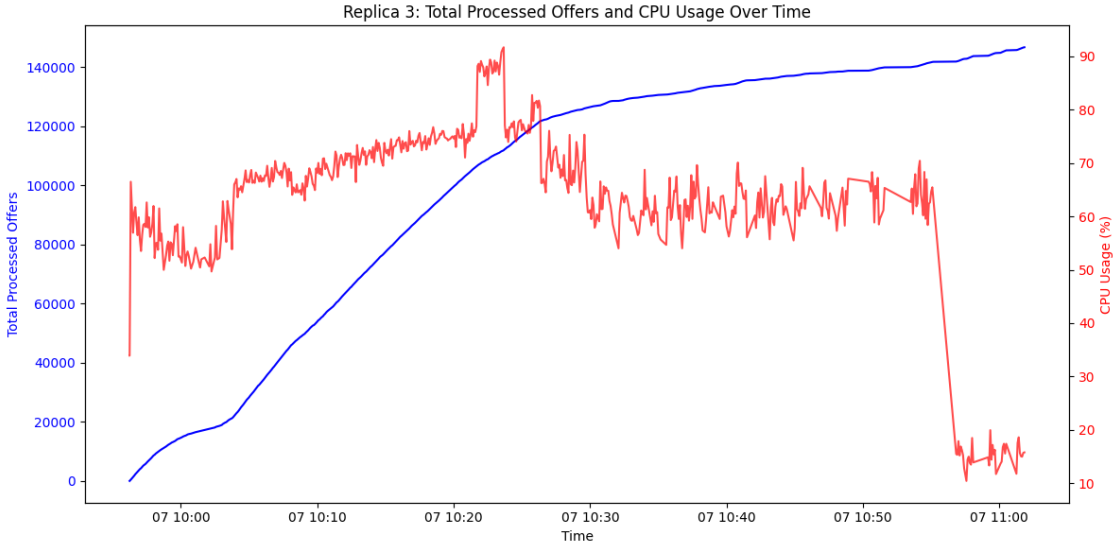

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
500
1000
1500
2000
20
30
40
50
60
70
80
90
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50

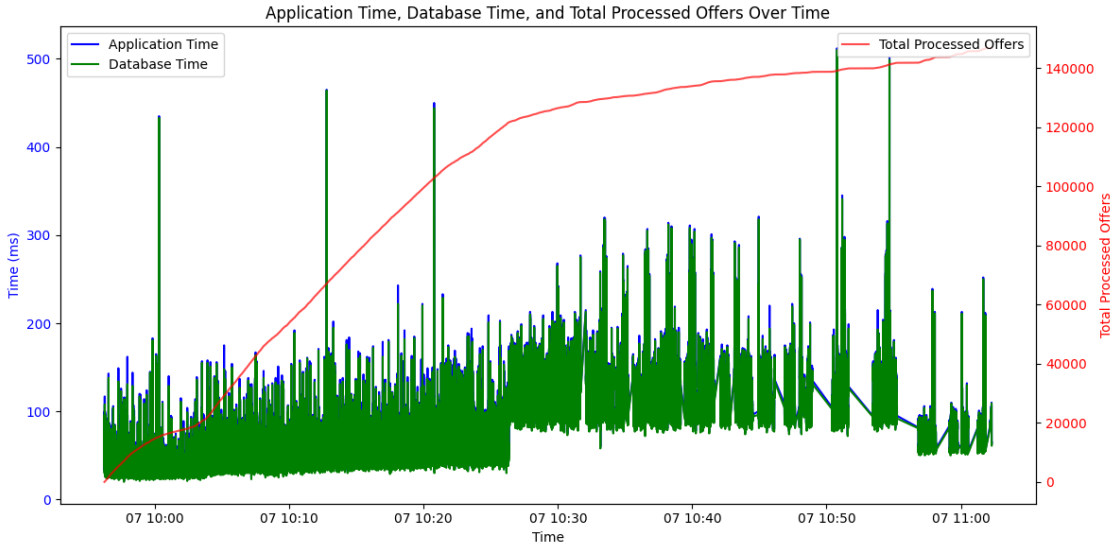

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
200
400
600
800
1000
20
40
60
80
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50

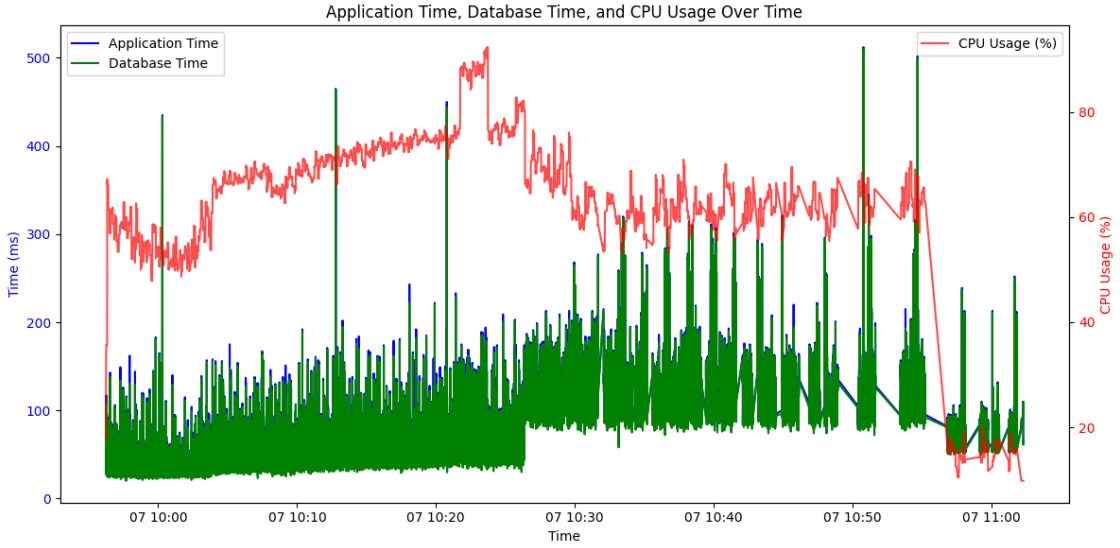

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
20
40
60
80
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50
07 11:00

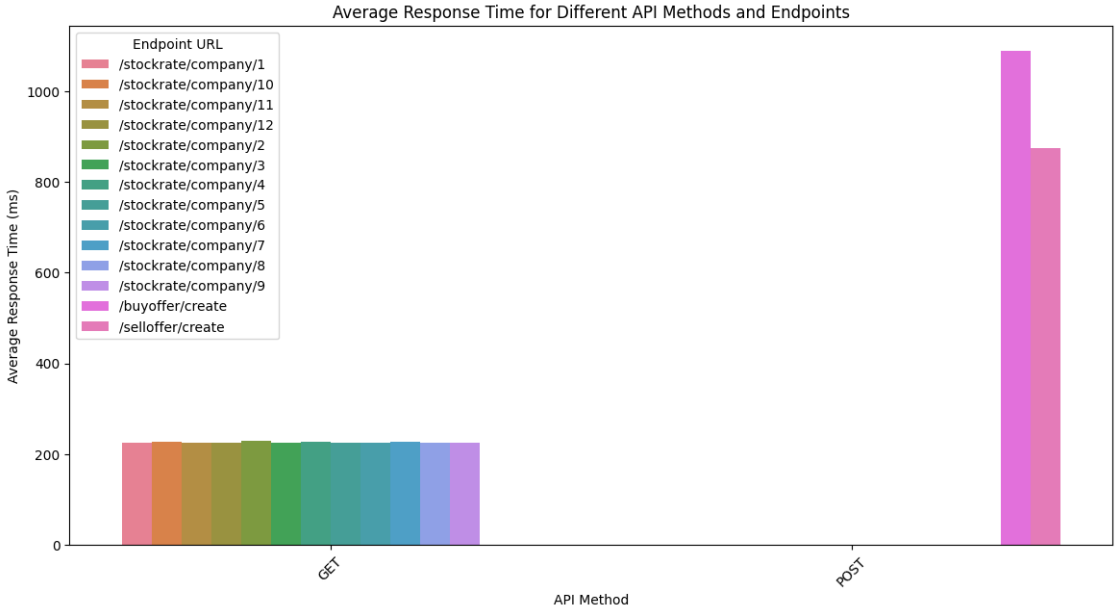

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
10
20
30
40
50
60
70
80
90
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50
07 11:00

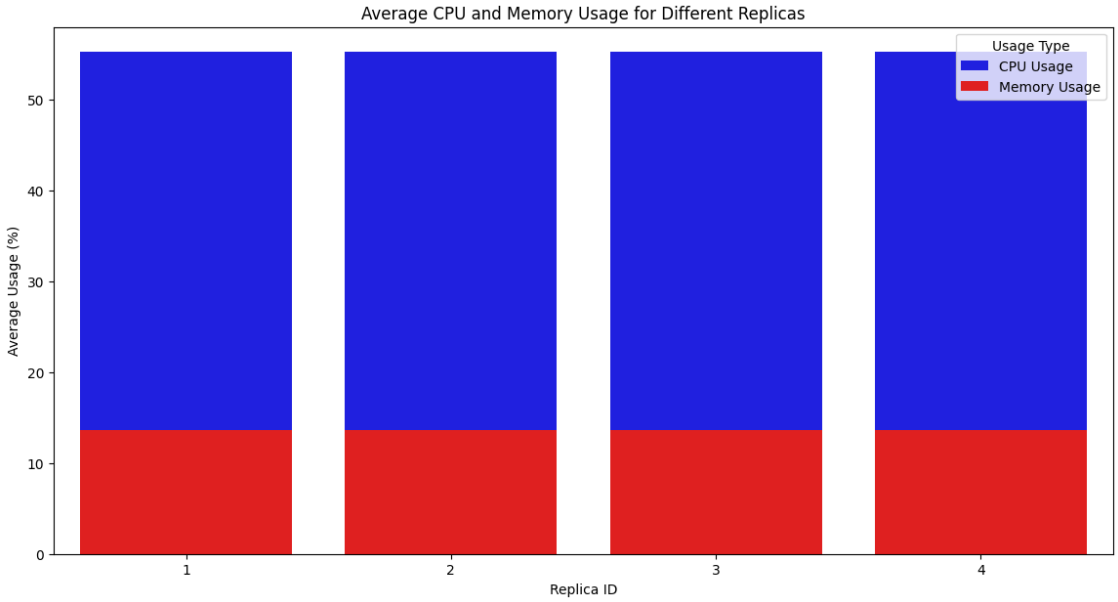

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
10
20
30
40
50
60
70
80
90
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50
07 11:00

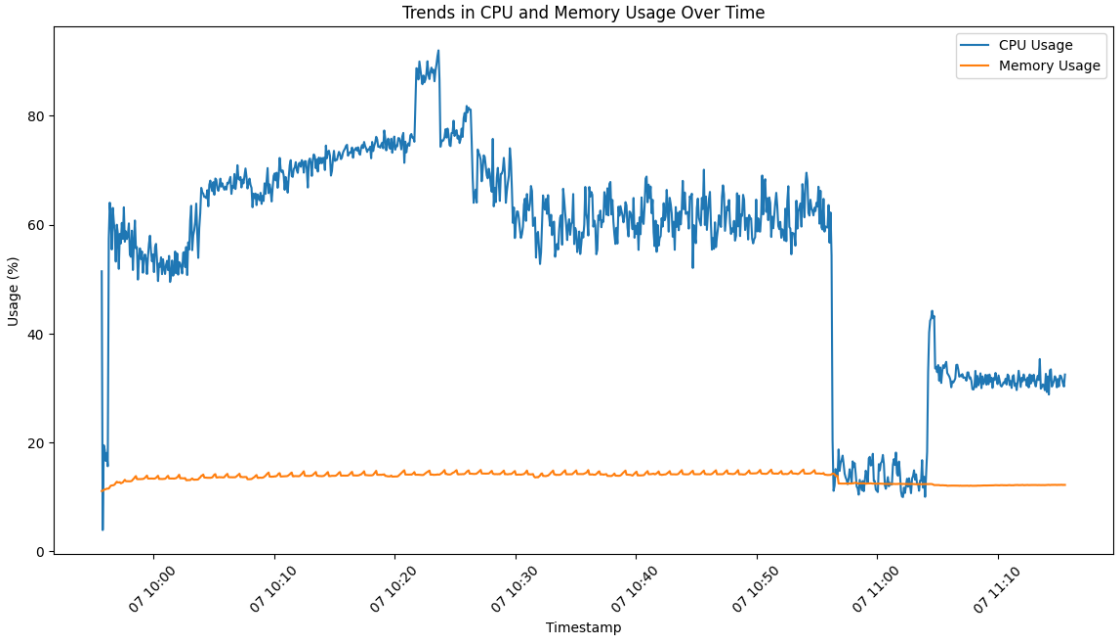

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
10
20
30
40
50
60
70
80
90
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50
07 11:00

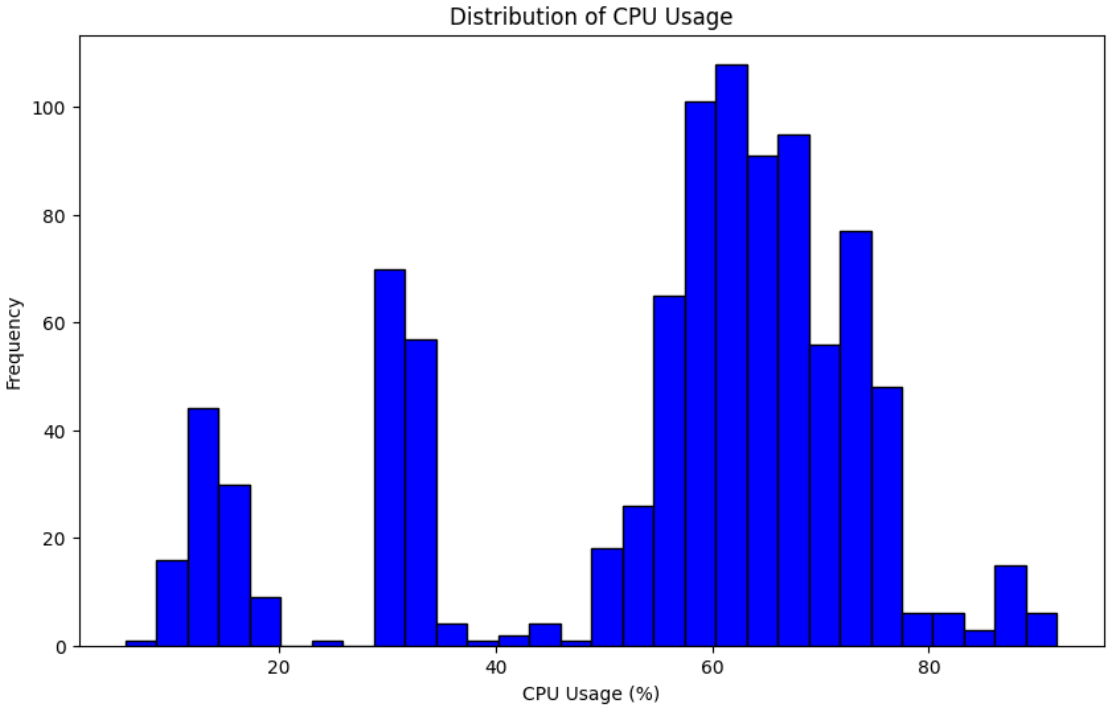

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50
07 11:00

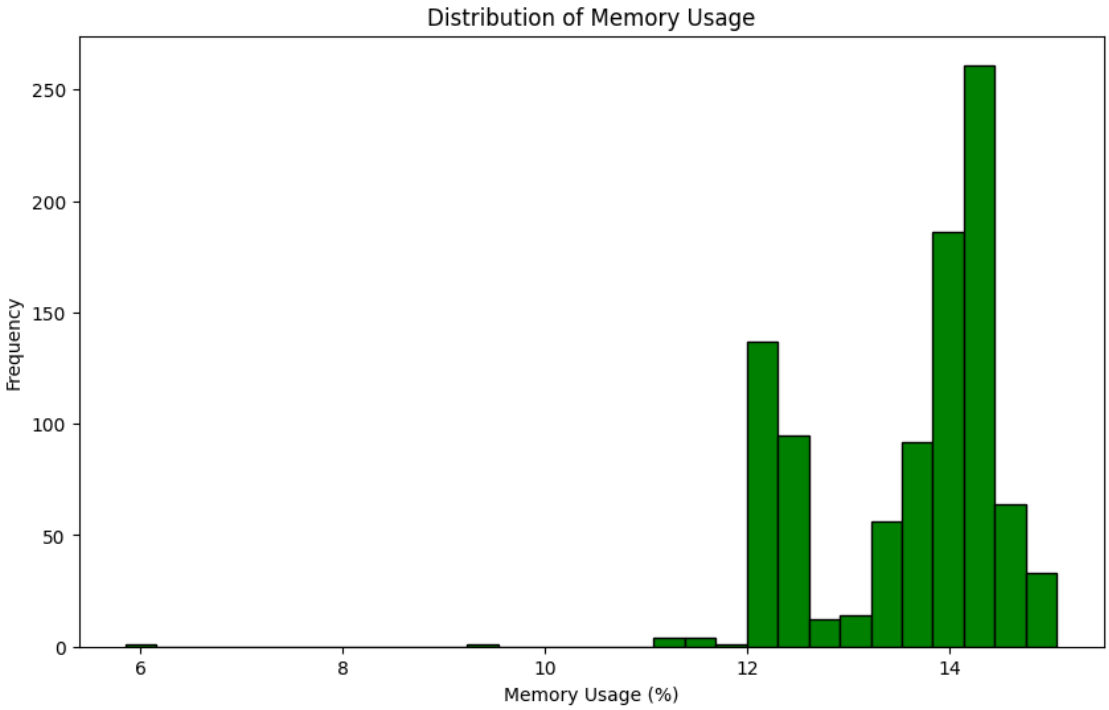

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50
07 11:00

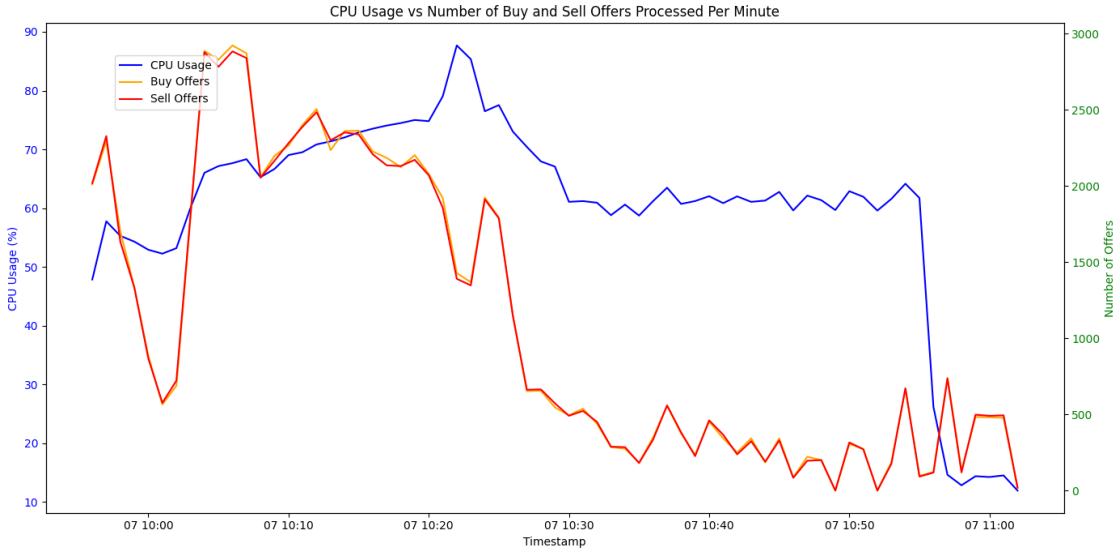

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
10
20
30
40
50
1
2
3
4
Replica ID

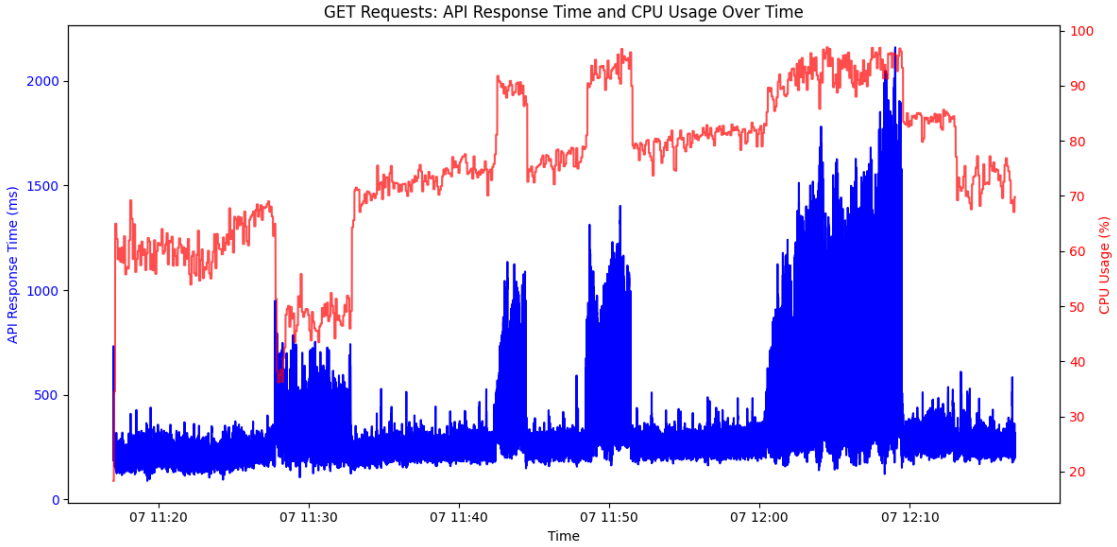

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
0
20
40
60
80
07 10:00
07 10:10
07 10:20
07 10:30
07 10:40
07 10:50
07 11:00
07 11:10
Timestamp

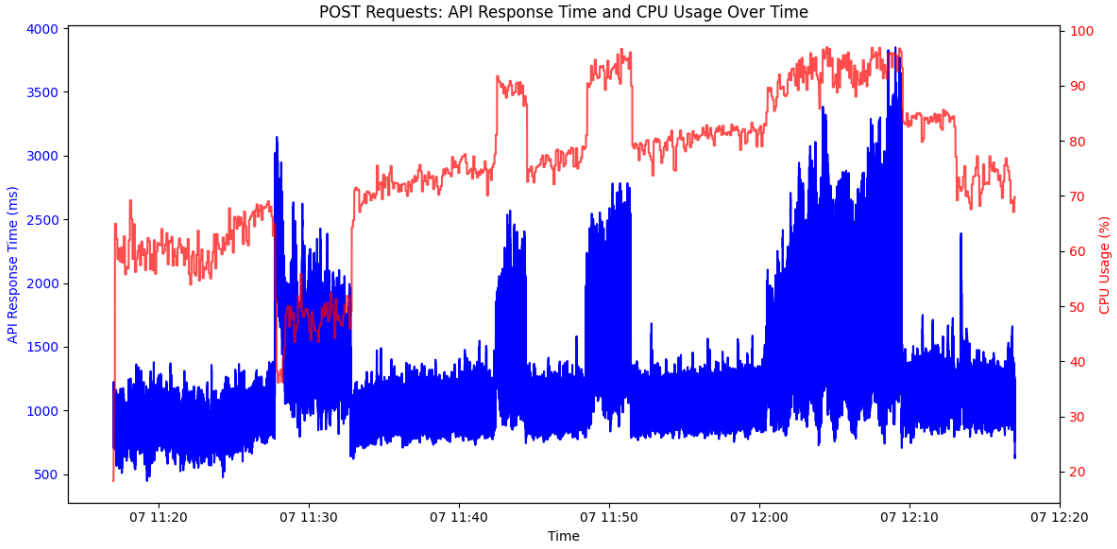

Distribution of CPU Usage
Frequency
0
20
40
60
80
100
20
40
60
80
CPU Usage (%)

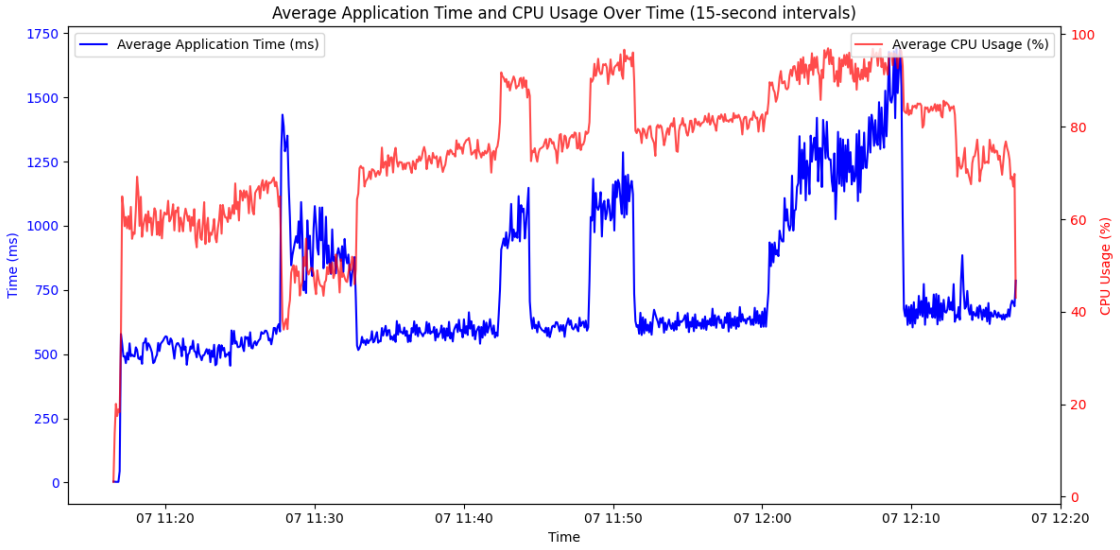


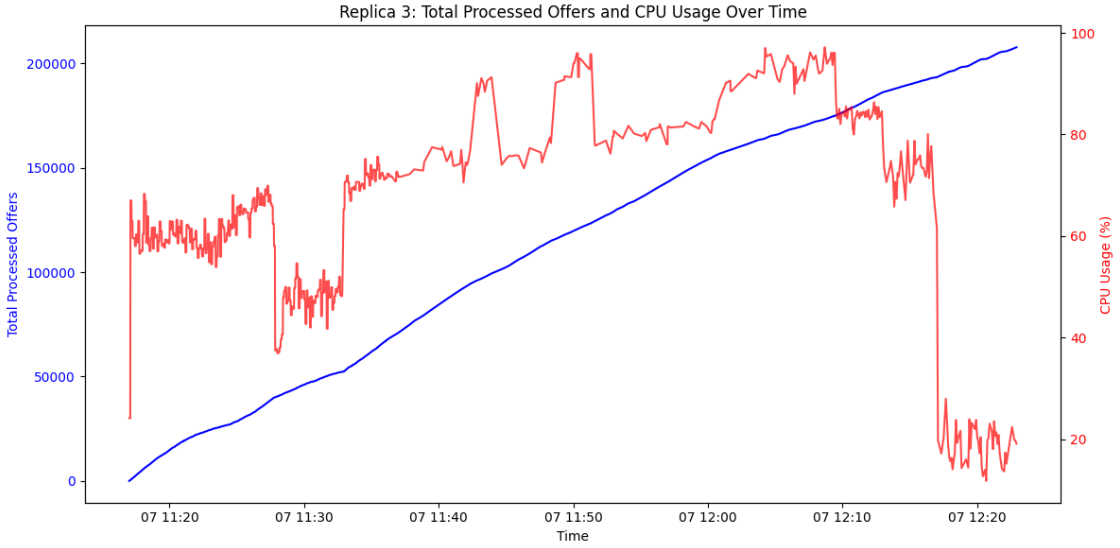


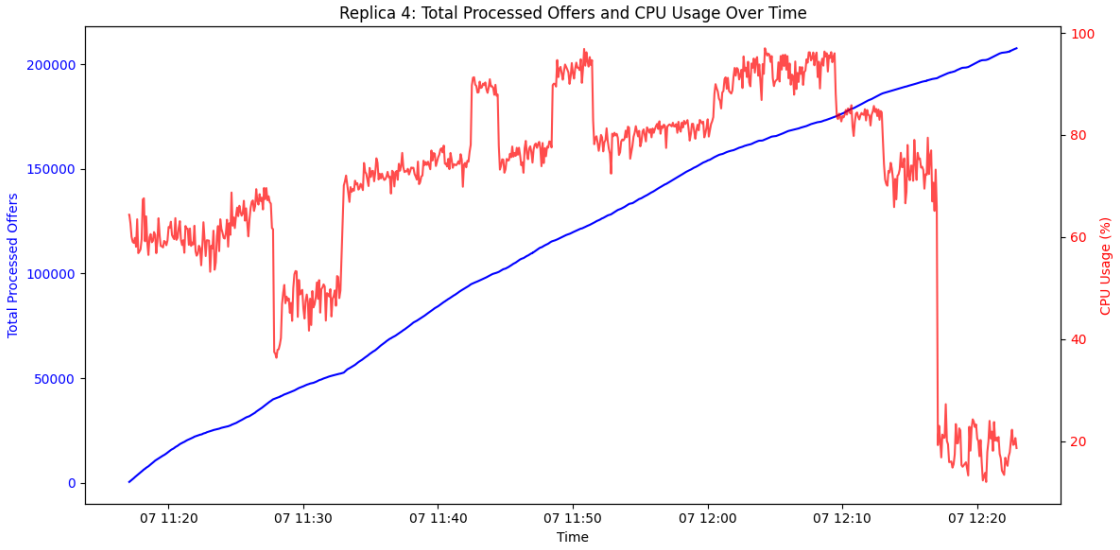

```
Loaded data from c:\Users\luki_\Desktop\logs\test3\4 3 200 500 4 500 100
successfully.
```

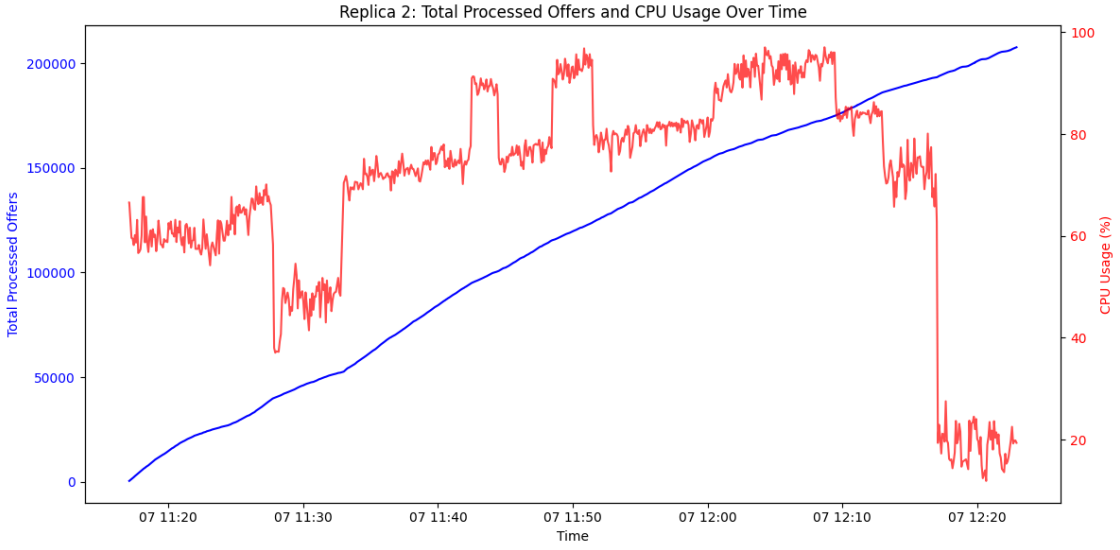

GET Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
500
1000
1500
2000
20
30
40
50
60
70
80
90
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10

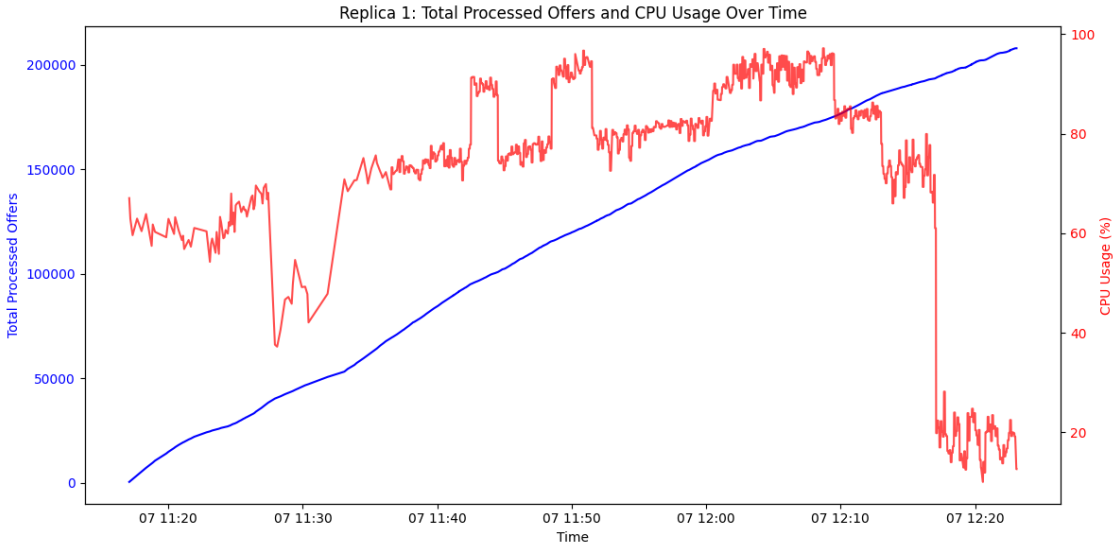

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
500
1000
1500
2000
2500
3000
3500
4000
20
30
40
50
60
70
80
90
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

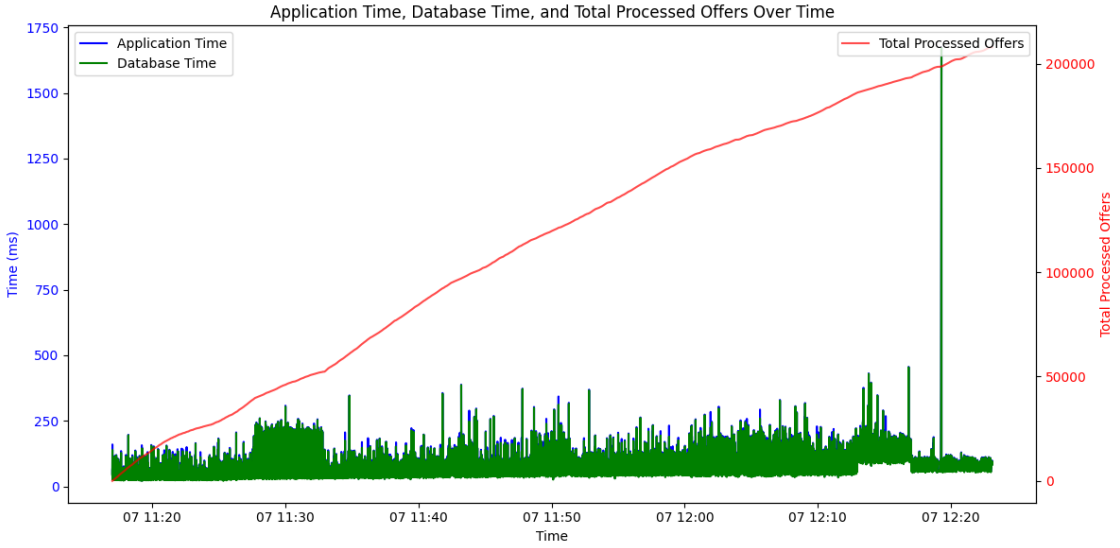

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
250
500
750
1000
1250
1500
1750
0
20
40
60
80
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

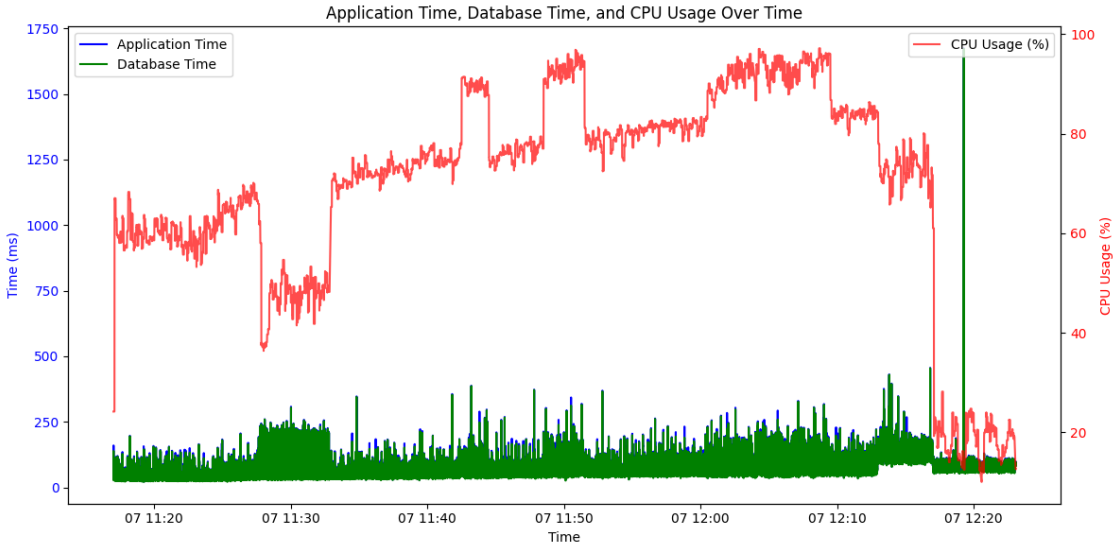

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
40
60
80
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

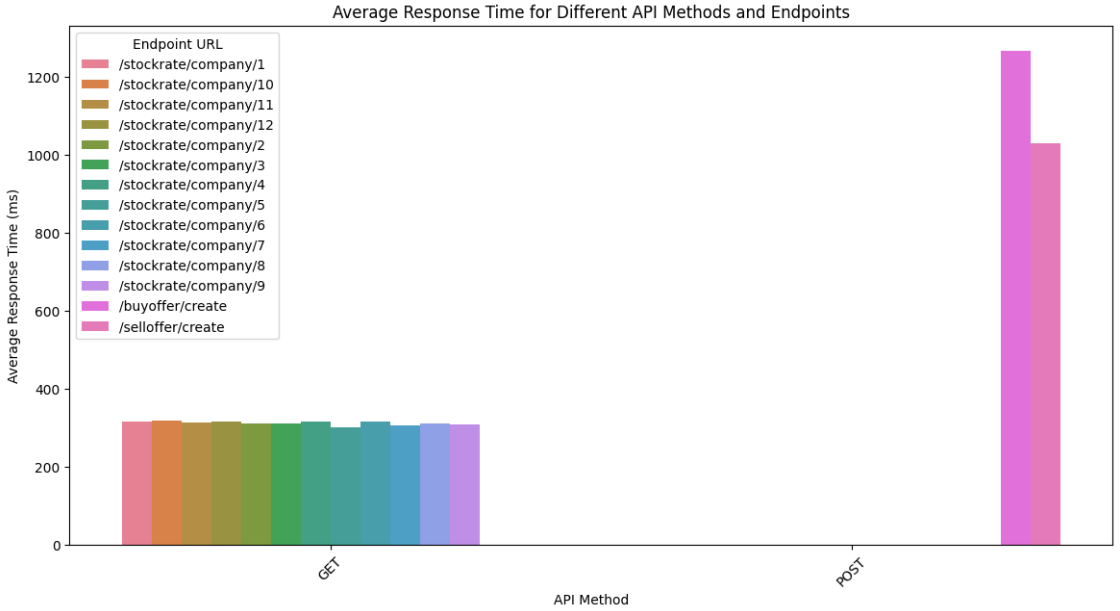

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
40
60
80
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

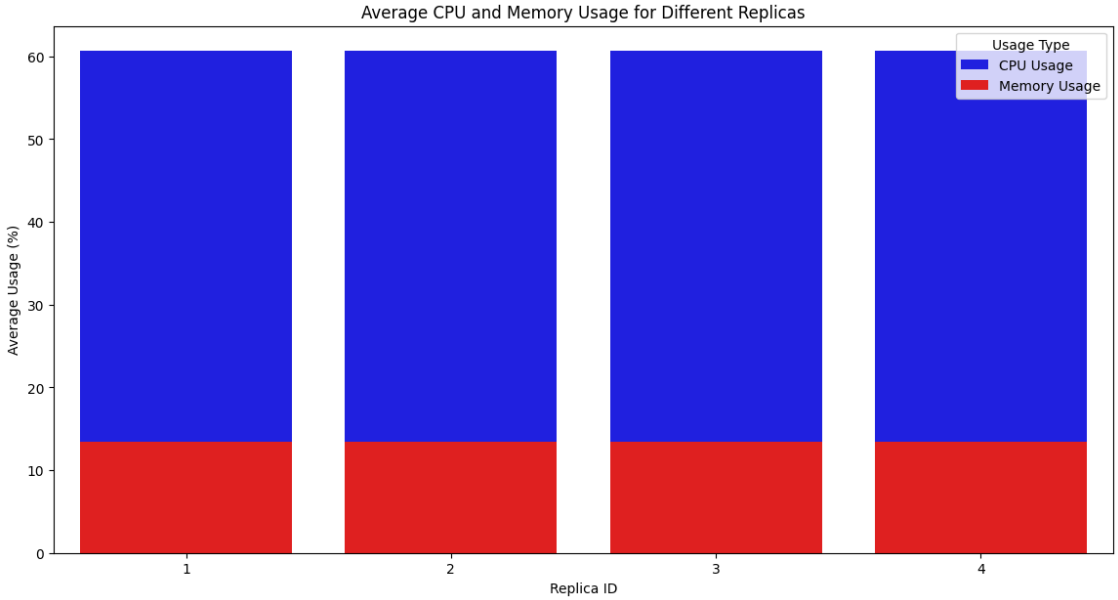

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
40
60
80
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

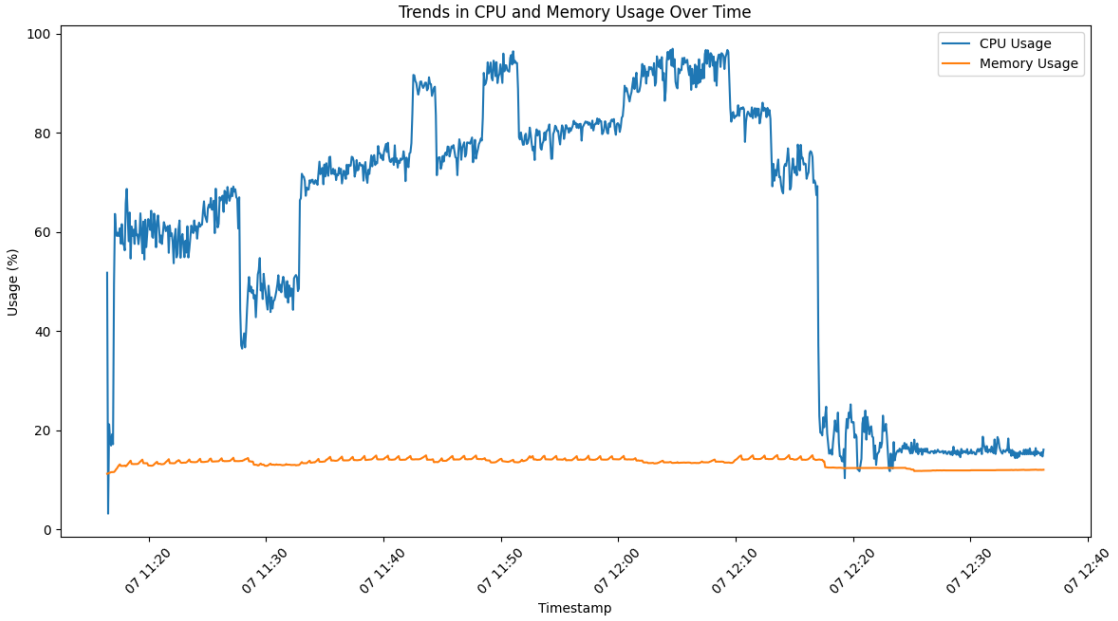

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
50000
100000
150000
200000
20
40
60
80
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

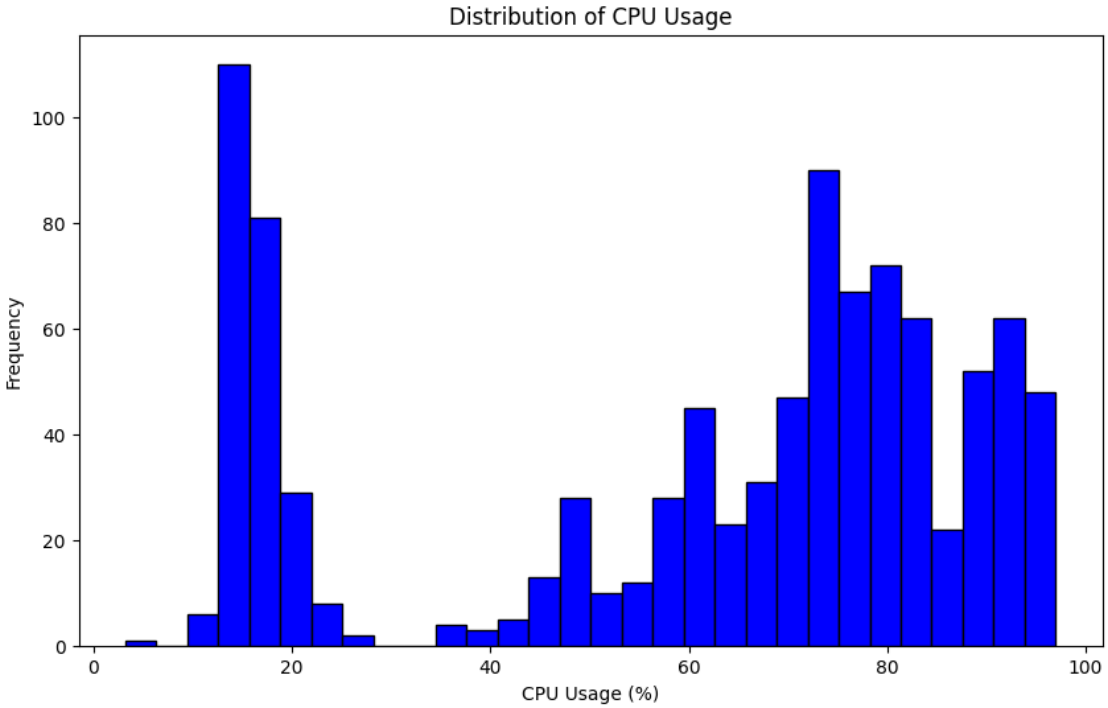

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
0
250
500
750
1000
1250
1500
1750
50000
100000
150000
200000
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

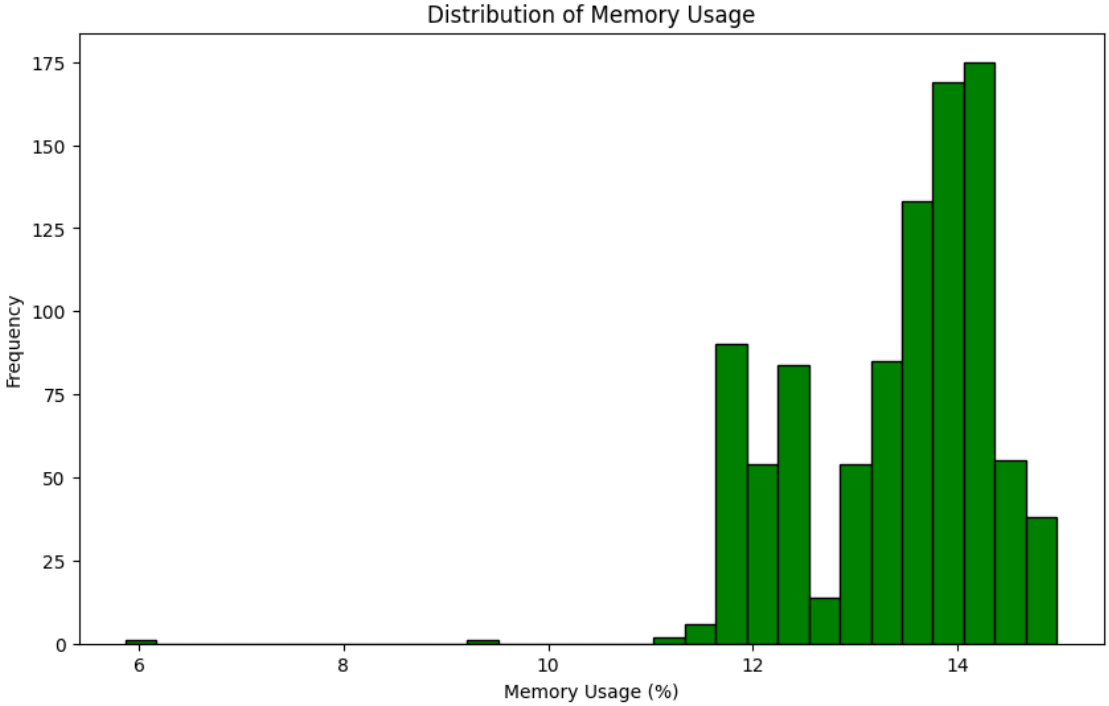

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
250
500
750
1000
1250
1500
1750
20
40
60
80
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20

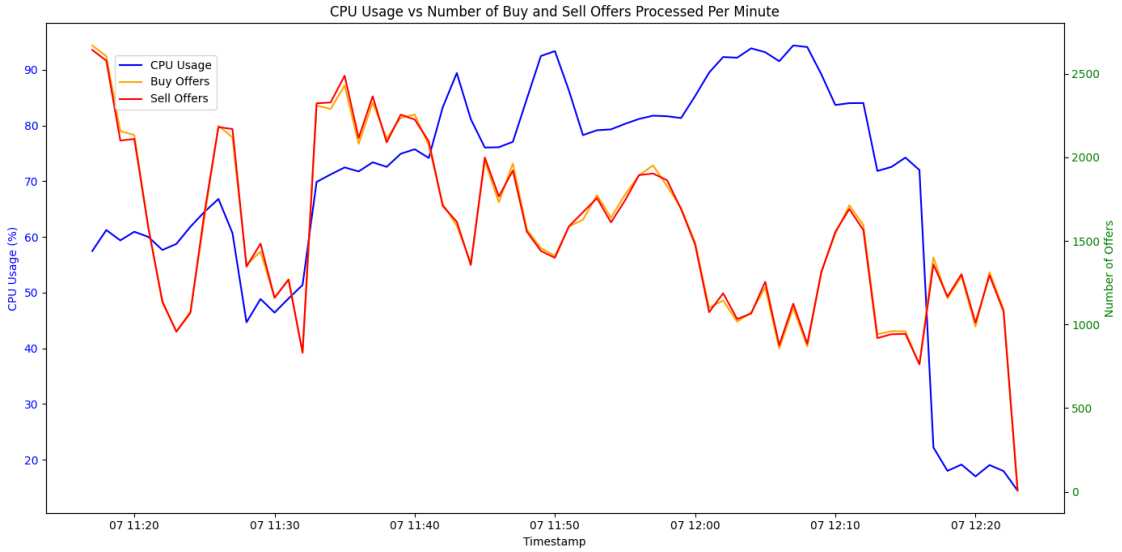

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
Replica ID
0
10
20
30
40
50
60
1
2
3
4

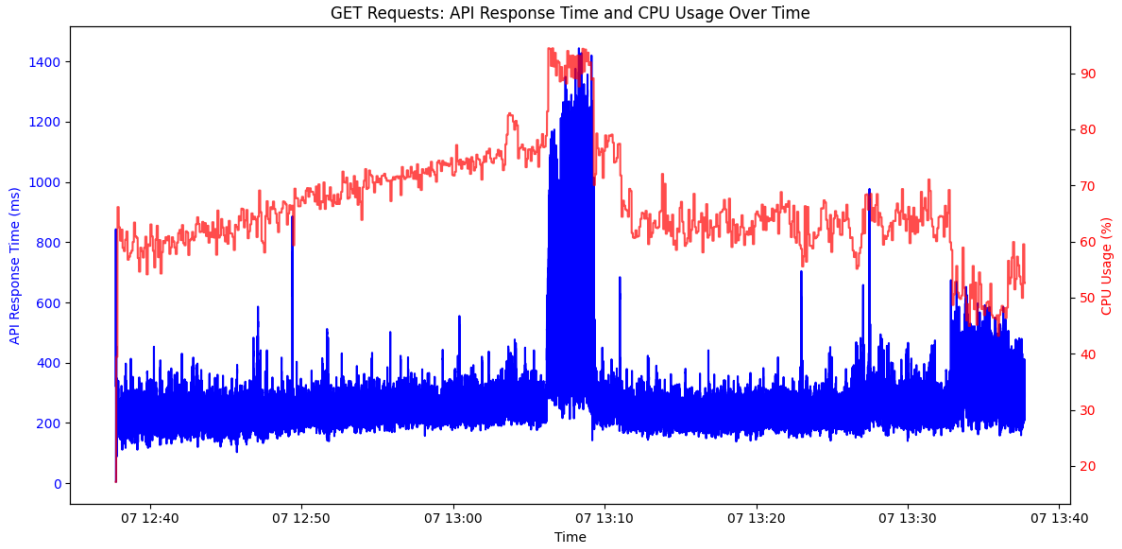

Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
Timestamp
0
20
40
60
80
100
07 11:20
07 11:30
07 11:40
07 11:50
07 12:00
07 12:10
07 12:20
07 12:30
07 12:40

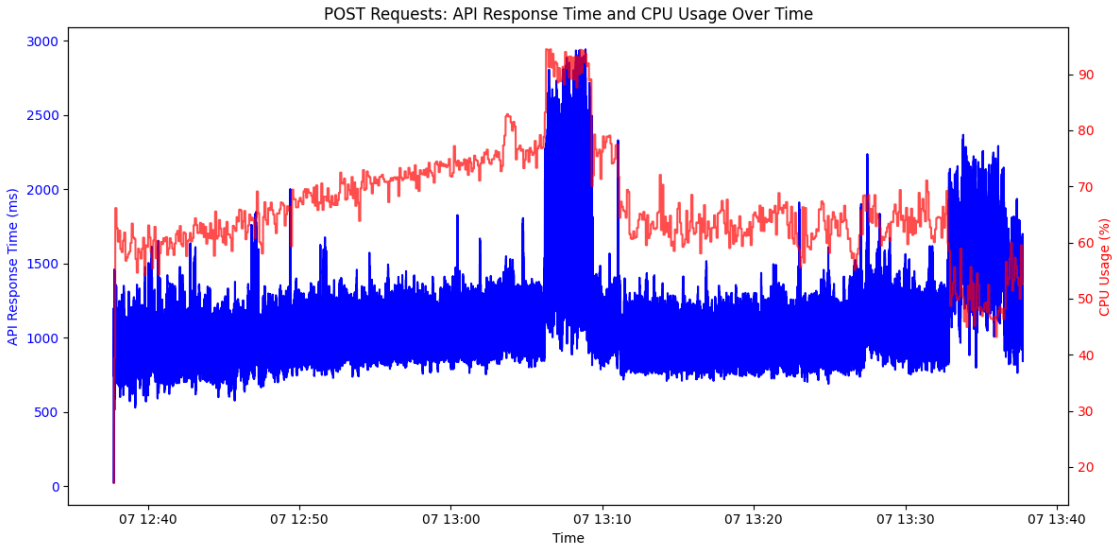

Distribution of CPU Usage
Frequency
CPU Usage (%)
0
20
40
60
80
100
0
20
40
60
80
100

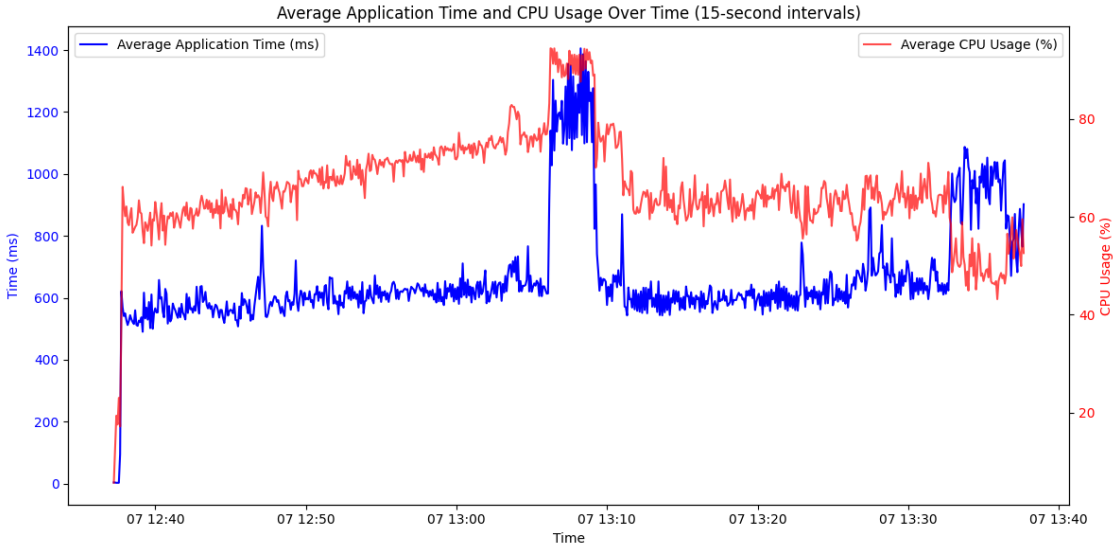

Distribution of Memory Usage
Frequency
Memory Usage (%)
0
25
50
75
100
125
150
175
6
8
10
12
14

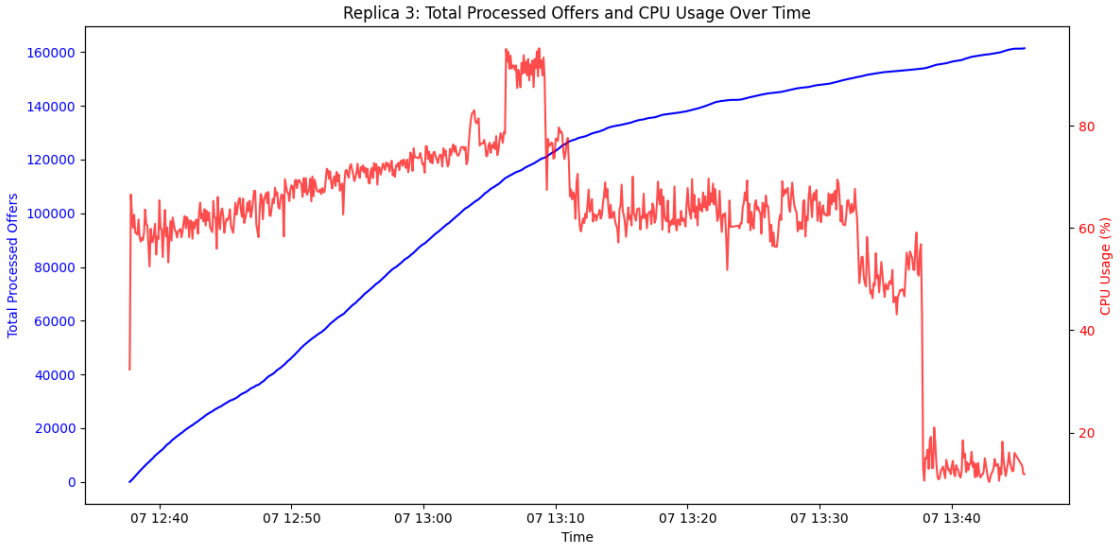


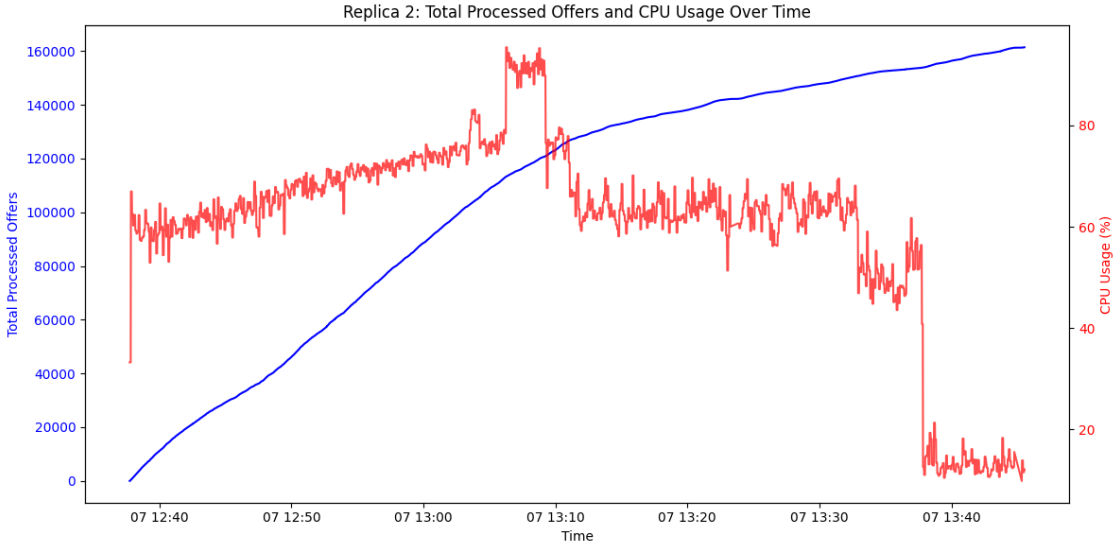

```
Loaded data from c:\Users\luki_\Desktop\logs\test3\4 3 200 750 4 500 100
successfully.
```

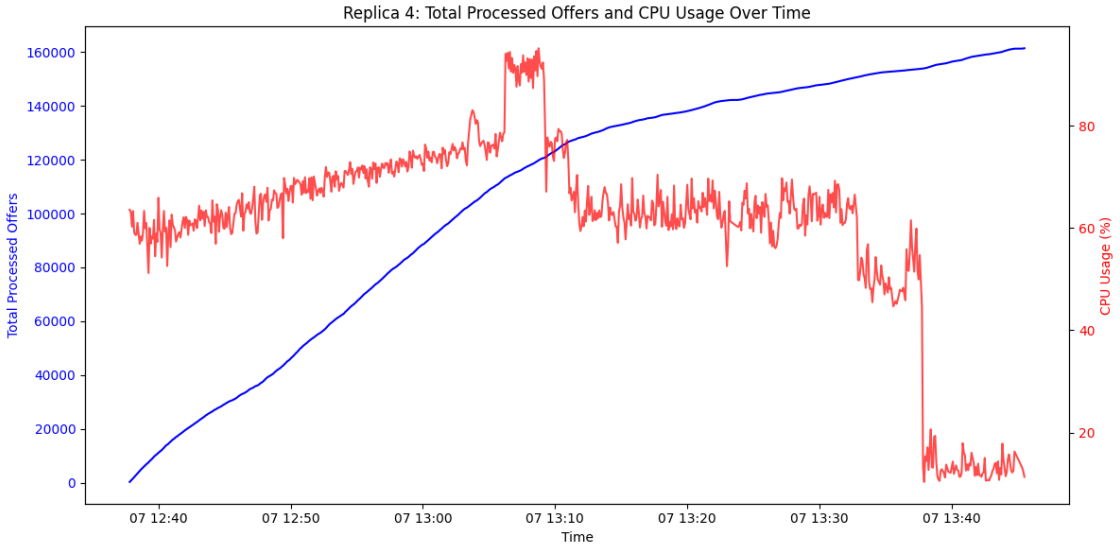


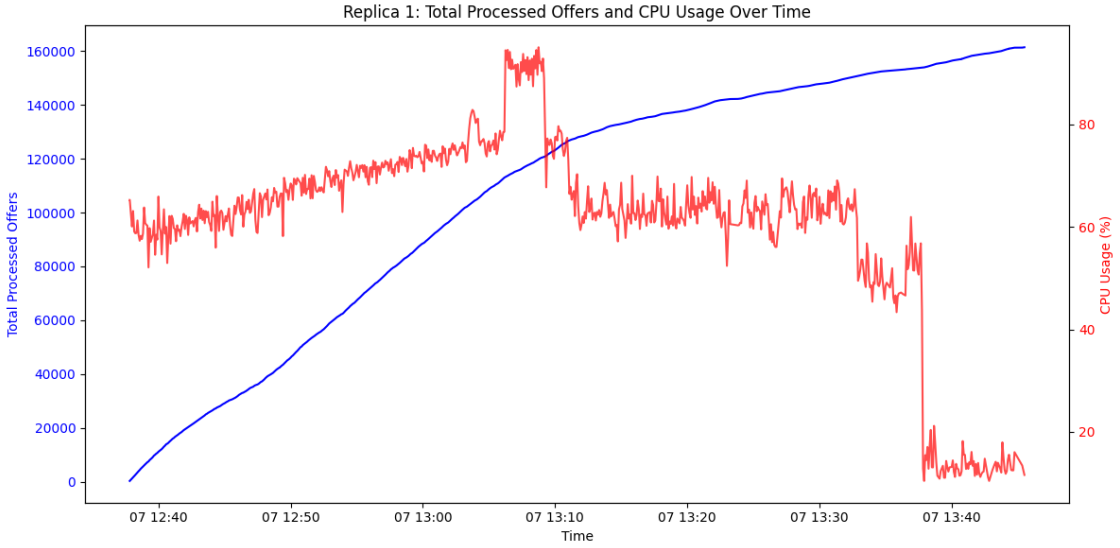

POST Requests: API Response Time and CPU Usage Over Time
API Response Time (ms)
CPU Usage (%)
Time
0
500
1000
1500
2000
2500
3000
20
30
40
50
60
70
80
90
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

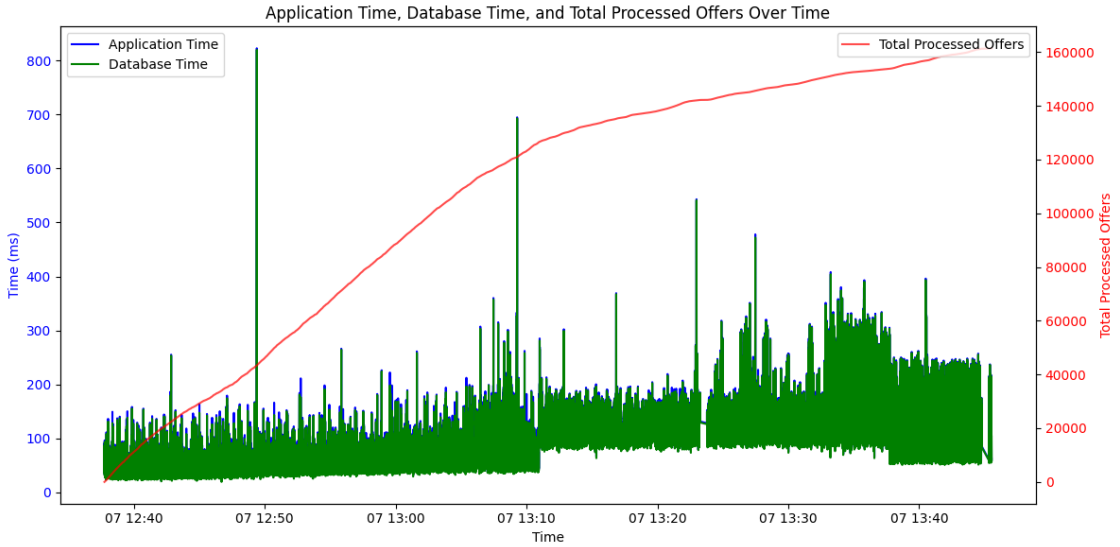

Average Application Time and CPU Usage Over Time (15-second intervals)
Average Application Time (ms)
Average CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
0
200
400
600
800
1000
1200
1400
20
40
60
80
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

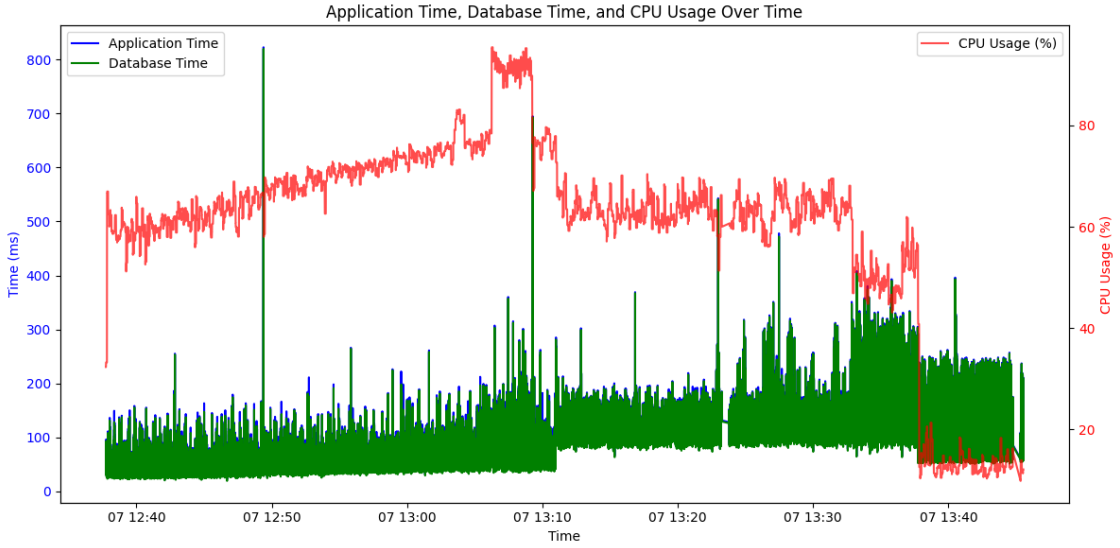

Replica 3: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

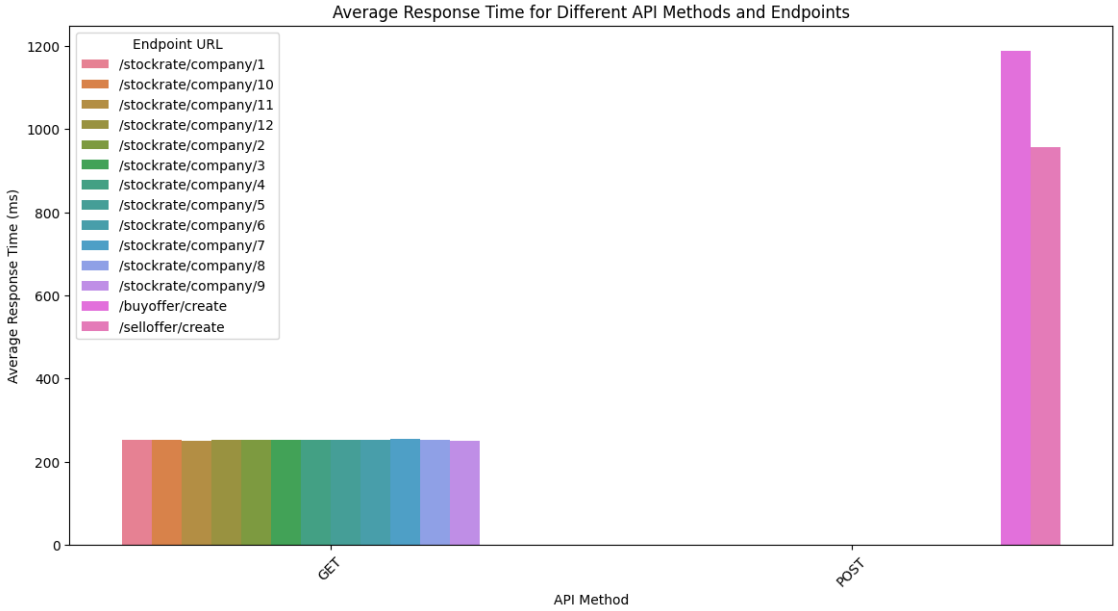

Replica 2: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

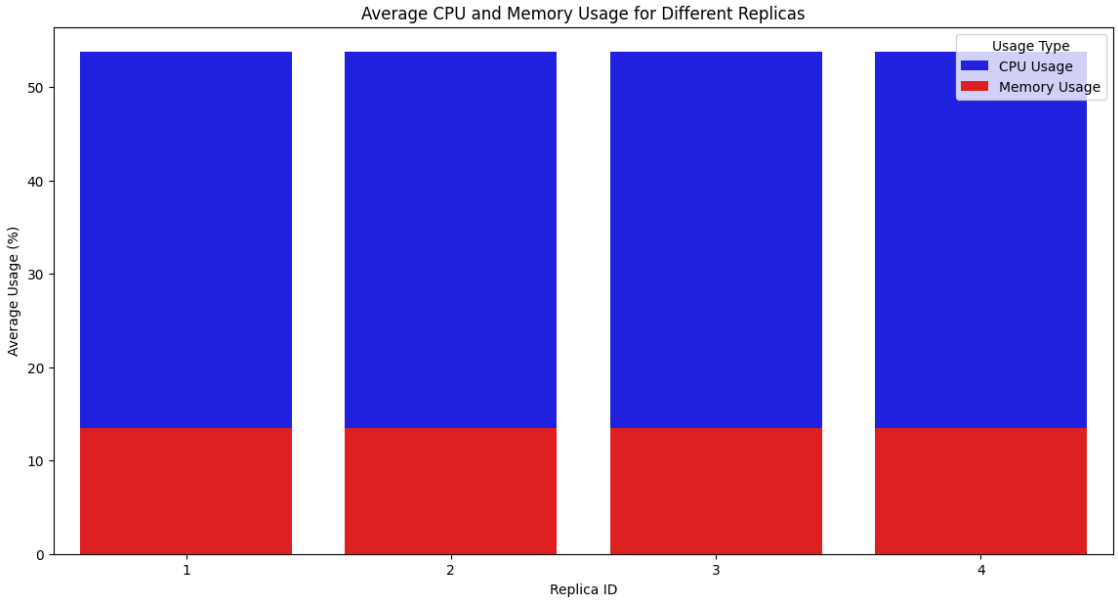

Replica 4: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

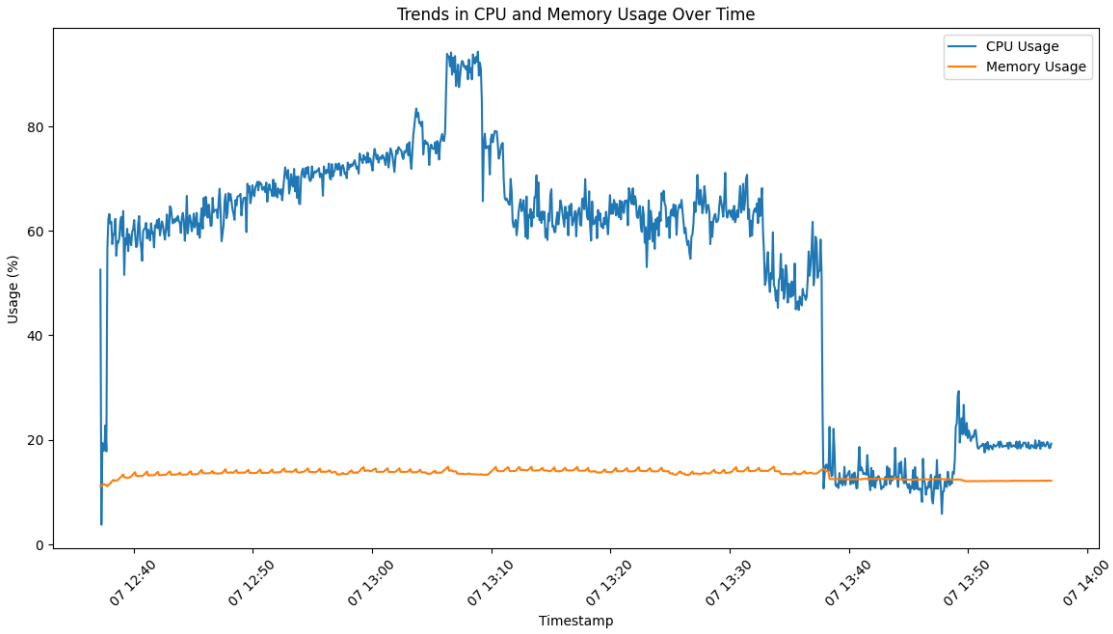

Replica 1: Total Processed Offers and CPU Usage Over Time
Total Processed Offers
CPU Usage (%)
Time
0
20000
40000
60000
80000
100000
120000
140000
160000
20
40
60
80
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

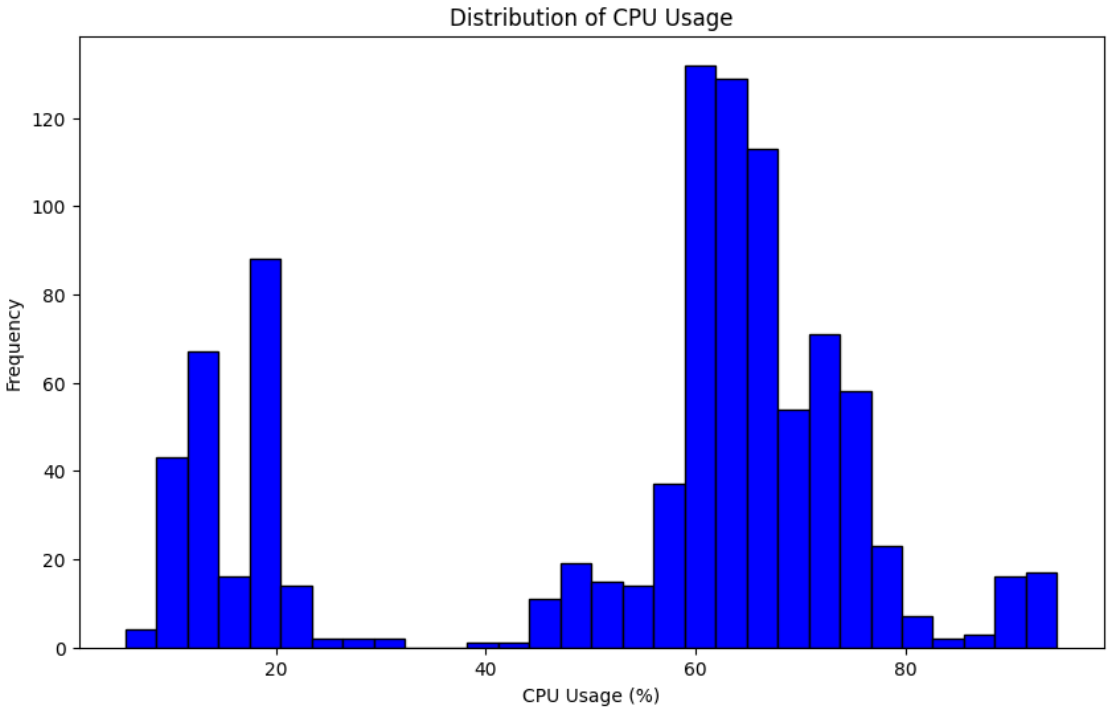

Application Time, Database Time, and Total Processed Offers Over Time
Application Time
Database Time
Total Processed Offers
Time (ms)
Total Processed Offers
Time
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

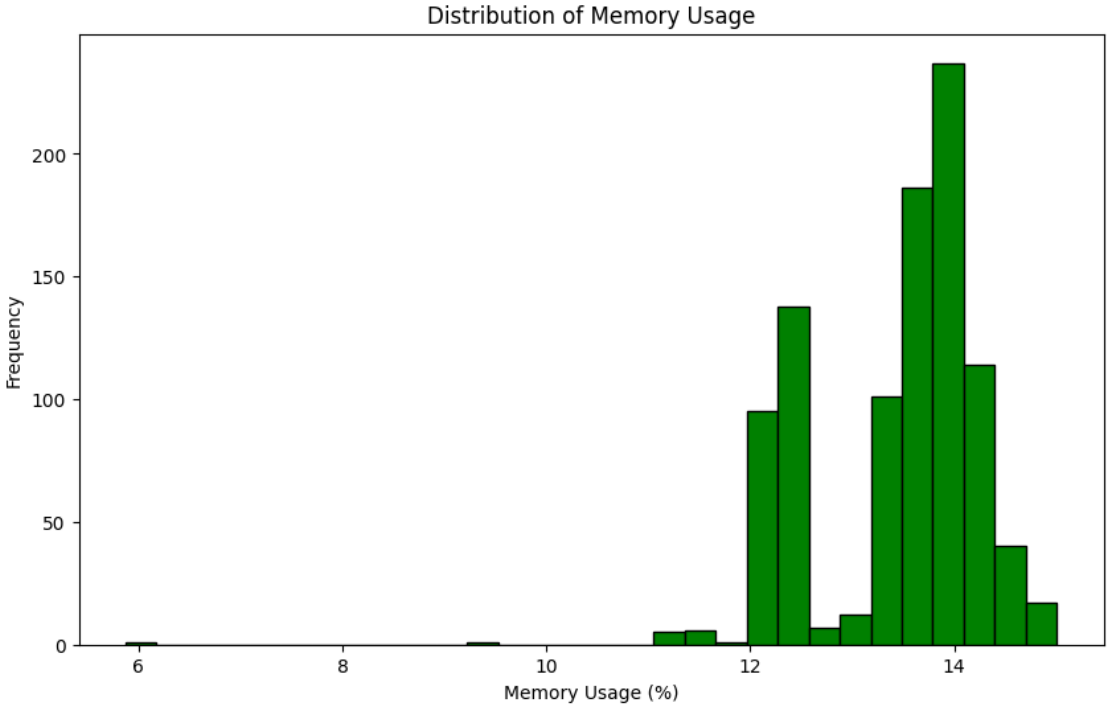

Application Time, Database Time, and CPU Usage Over Time
Application Time
Database Time
CPU Usage (%)
Time (ms)
CPU Usage (%)
Time
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40

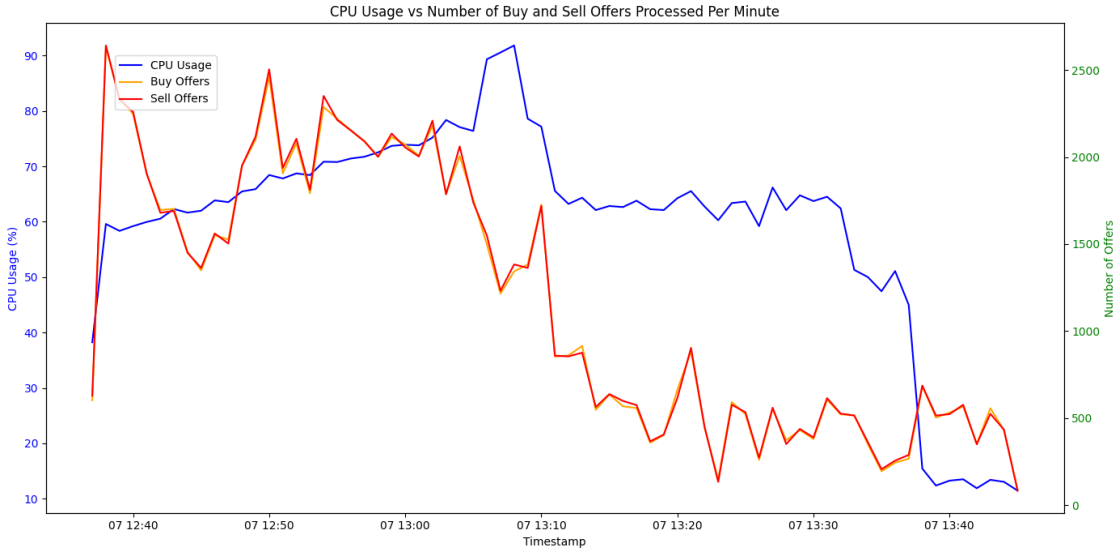

Average Response Time for Different API Methods and Endpoints
Endpoint URL
/stockrate/company/1
/stockrate/company/10
/stockrate/company/11
/stockrate/company/12
/stockrate/company/2
/stockrate/company/3
/stockrate/company/4
/stockrate/company/5
/stockrate/company/6
/stockrate/company/7
/stockrate/company/8
/stockrate/company/9
/buyoffer/create
/selloffer/create
Average Response Time (ms)
0
200
400
600
800
1000
1200
GET
POST
API Method


Average CPU and Memory Usage for Different Replicas
Usage Type
CPU Usage
Memory Usage
Average Usage (%)
0
10
20
30
40
50
1
2
3
4
Replica ID


Trends in CPU and Memory Usage Over Time
CPU Usage
Memory Usage
Usage (%)
0
20
40
60
80
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40
07 13:50
07 14:00
Timestamp


Distribution of CPU Usage
Frequency
0
20
40
60
80
100
120
20
40
60
80
CPU Usage (%)


Distribution of Memory Usage
Frequency
Memory Usage (%)
0
50
100
150
200
6
8
10
12
14


CPU Usage vs Number of Buy and Sell Offers Processed Per Minute
CPU Usage
Buy Offers
Sell Offers
CPU Usage (%)
Number of Offers
Timestamp
07 12:40
07 12:50
07 13:00
07 13:10
07 13:20
07 13:30
07 13:40