

```

import pandas as pd
import numpy as np

data = {
    'server_id': ['server1', 'server2', 'server3'],
    'cpu_usage': [40, 70, 30],
    'memory_usage': [50, 60, 40],
    'disk_usage': [30, 45, 20]
}

df = pd.DataFrame(data)

TARGET_CPU_CAPACITY = 100
TARGET_MEMORY_CAPACITY = 100

total_cpu_usage = df['cpu_usage'].sum()
total_memory_usage = df['memory_usage'].sum()

num_servers_cpu = np.ceil(total_cpu_usage / TARGET_CPU_CAPACITY)
num_servers_memory = np.ceil(total_memory_usage / TARGET_MEMORY_CAPACITY)

consolidated_servers = max(num_servers_cpu, num_servers_memory)

print(f"Total CPU usage: {total_cpu_usage}%")
print(f"Total Memory usage: {total_memory_usage}%")
print(f"Estimated number of consolidated servers needed: {int(consolidated_servers)}")

def distribute_workloads(df, num_servers):
    workloads = []
    df = df.sort_values(by='cpu_usage', ascending=False)

    for i in range(int(num_servers)):
        workloads.append({'server_id': f'new_server{i+1}', 'cpu_usage': 0, 'memory_usage': 0})

    for idx, row in df.iterrows():

        least_loaded_server = min(workloads, key=lambda x: x['cpu_usage'])
        least_loaded_server['cpu_usage'] += row['cpu_usage']
        least_loaded_server['memory_usage'] += row['memory_usage']

    return workloads

consolidated_workloads = distribute_workloads(df, consolidated_servers)

for server in consolidated_workloads:
    print(f"Server {server['server_id']} - CPU Usage: {server['cpu_usage']}%, Memory Usage: {server['memory_usage']}%")

```

Output

Total CPU usage: 140%

Total Memory usage: 150%

Estimated number of consolidated servers needed: 2

Server new_server1 - CPU Usage: 70%, Memory Usage: 60%

Server new_server2 - CPU Usage: 70%, Memory Usage: 90%