



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

شبکه های کامپیوتری

(پاییز ۱۴۰۰)

گزارش پروژه نهایی

محمد چوپان ۹۸۳۱۱۲۵

مشکل label گذاری ها رفع شد.

در این پروژه از کتاب خانه های socket و Prometheus_client برای پایتون استفاده شده است .

در کل روند پروژه به این صورت است که ما در ابتدا یک سرور را با استفاده از سوکت بالا می آوریم که در این سرور با استفاده از ریسمان ها کاری می کنیم که به توان چند کلاینت به صورت همزمان متصل باشد. سپس متریک های ارسال شده توسط کلاینت را به پرومیتئوس وصل میکنیم. این هم به این شکل است که طبق توضیحات ویدیو فایل yml را تغییر می دهیم تا سرور ما را بشناسد در نهایت با متریک Gauge داده ها را برای سرور پرومیتئوس ارسال می کنیم.

برنامه پرومیتئوس دارای ۴ نوع متریک است که شامل Gauge,Counter,Summery,Histogram است.

: Counter

به این صورت است که مقدار آن می تواند افزایش یابد یا ریست یا همان صفر شود.

: Gauge

مقدار آن می تواند افزایش یا کاهش یابد یا به مقدار دلخواهی مقدار دهی شود با استفاده از تابع set()

: Summery

شبيه به هیستوگرام، یک خلاصه مشاهدات را نمونه می کند (معمولا چیزهایی مانند مدت زمان درخواست و اندازه پاسخ). در حالی که تعداد کل مشاهدات و مجموع تمام مقادیر مشاهده شده را نیز ارائه می دهد، چندک های قابل تنظیم را در یک پنجره زمانی کشویی محاسبه می کند.

: Histogram

از مقادیر ارسال شده نمونه برداری می کند و در سبد های منظم شمارش می کند. و مجموع تمام مقادیر مشاهده شده را نیز ارائه می دهد.

با توجه به توصیفات ارائه شده بهترین متریک برای ما Gauge است . ما از این نوع متریک برای تمامی داده های خود استفاده کردیم .

که علاوه بر متریک های سیستم تعداد کاربر های متصل را نیز نشان می دهد البته یک متریک دیگر وجود دارد تا بتوان تشخیص داد که متریک های ارسالی برای کدام کاربر است که برای جدا سازی آن ها از شماره port کلاینت استفاده میکنیم.

و در نهایت کد پروژه :

کلاینت :

```
import socket
import json
from tkinter.tix import Tree
import psutil
import time

# connect to server
def connect(ip, port):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((ip, port))
    print("Connected to server: "+ip+" : "+str(port))
    return s

#send data to server
def send(s, data):
    s.send(data.encode())

#read system data
def read_data(client):
    data = {}
    data['cpu'] = psutil.cpu_percent(interval=1)
    data['ram'] = psutil.virtual_memory().percent
    data['disk'] = psutil.disk_usage('/').percent
    data['ram_total'] = psutil.virtual_memory().total
    data["ram_available"] = psutil.virtual_memory().available
    data['ram_used']=psutil.virtual_memory().available * 100 / psutil.virtual_memory().total
    data['client_no']=client.getsockname()[1]
    return data

#convert data to json
def json_convert (data):
    return json.dumps(data)

#main
if __name__ == '__main__':
    counter =1
    while True:
        try:
            client=connect('127.0.0.1',45678)
            while True :
                data=read_data(client)
                JSON=json_convert(data)
                send(client,JSON)
                time.sleep(15)
        except:
            print("")
            print(f"can not connect to server attempt {counter}")
            counter+=1
            time.sleep(5)
```

سرور :

```
from multiprocessing import connection
import socket
import json
import threading
import time
import random
from _thread import *
from prometheus_client import start_http_server, Gauge
# create server
# print_lock = threading.Lock()
cpu=Gauge('cpu', 'cpu percent ',["client_no"])
ram=Gauge('ram', 'ram percent',["client_no"])
disk=Gauge('disk', 'disk percent',["client_no"])
ram_total=Gauge('ram_total', 'ram total',["client_no"])
ram_available=Gauge('ram_available', 'ram available',["client_no"])
ram_used=Gauge('ram_used', 'ram used',["client_no"])
clients_connected=Gauge('clients_connected', 'clients connected')
def create_server(ip,port):...
# send data to client
def send_data(connection,json):
    connection.send(json.encode())
# receive data from client
def rcv_data(connection):
    return connection.recv(2048).decode()

#accept client connection
def accept_client(s):
    connection, address = s.accept()
    print('Connection from: ' + str(address))

    return connection,address
#read json
def json_parser(datas):
    return json.loads(datas)
```

```

def accpet_client_data(connection):
    while True:
        try:
            data = recv_data(connection)
            if not data:
                clients_connected.dec()
                break
            data = json_parser(data)
            print(data)
            # ram.set(data['ram'])
            ram.labels(client_no=data['client_no']).set(data['ram'])
            # cpu.set(data['cpu'])
            cpu.labels(client_no=data['client_no']).set(data['cpu'])
            # disk.set(data['disk'])
            disk.labels(client_no=data['client_no']).set(data['disk'])
            # ram_total.set(data['ram_total'])
            ram.labels(client_no=data['client_no']).set(data['ram_total'])
            # ram_available.set(data['ram_available'])
            ram_available.labels(client_no=data['client_no']).set(data['ram_available'])
            # ram_used.set(data['ram_used'])
            ram_used.labels(client_no=data['client_no']).set(data['ram_used'])
            # client_no.set(data['client_no'])

            time.sleep(random.randint(1,10))
        except:
            connection.close()
            clients_connected.dec()
            break

#main
if __name__ == '__main__':
    print("server started at port 45678")
    server=create_server('0.0.0.0',45678)
    start_http_server(1234)
    while True:
        connection,address=accept_client(server)
        clients_connected.inc()

        # print_lock.acquire()
        start_new_thread(accpet_client_data, (connection,))
        connection.close()

```

خروجی حاصل :

زمانی که کلاینت به تنهایی متصل باشد و سرور فعال نباشد :

```
(venv) PS S:\uni\6\CN\Project\CN project\AUT-Computer-Network> python client.py
```

```
can not connect to server attempt 1
```

```
can not connect to server attempt 2
```

به صورت ۵ ثانیه یک بار اتصال را امتحان میکند.

زمانی که سرور متصل شود :

```
(venv) PS S:\uni\6\CN\Project\CN project\AUT-Computer-Network> python server.py  
server started at port 45678  
Connection from: ('127.0.0.1', 63552)  
{'cpu': 6.0, 'ram': 63.6, 'disk': 60.8, 'ram_total': 16558358528, 'ram_available': 6022291456, 'ram_used': 36.37009940216219, 'client_no': 63552}
```

```
can not connect to server attempt 7  
Connected to server: 127.0.0.1 : 45678
```

خروجی پرومیتئوس :

```

# HELP python_gc_objects_collected_total Objects collected during gc
# TYPE python_gc_objects_collected_total counter
python_gc_objects_collected_total{generation="0"} 112.0
python_gc_objects_collected_total{generation="1"} 273.0
python_gc_objects_collected_total{generation="2"} 0.0
# HELP python_gc_objects_uncollectable_total Uncollectable object found during GC
# TYPE python_gc_objects_uncollectable_total counter
python_gc_objects_uncollectable_total{generation="0"} 0.0
python_gc_objects_uncollectable_total{generation="1"} 0.0
python_gc_objects_uncollectable_total{generation="2"} 0.0
# HELP python_gc_collections_total Number of times this generation was collected
# TYPE python_gc_collections_total counter
python_gc_collections_total{generation="0"} 41.0
python_gc_collections_total{generation="1"} 3.0
python_gc_collections_total{generation="2"} 0.0
# HELP python_info Python platform information
# TYPE python_info gauge
python_info{implementation="CPython",major="3",minor="9",patchlevel="2",version="3.9.2"} 1.0
# HELP cpu_cpu percent
# TYPE cpu_gauge
cpu{client_no="63552"} 18.1
# HELP ram_ram percent
# TYPE ram_gauge
ram{client_no="63552"} 1.6558358528e+010
# HELP disk_disk percent
# TYPE disk_gauge
disk{client_no="63552"} 60.8
# HELP ram_total ram total
# TYPE ram_total_gauge
# HELP ram_available ram available
# TYPE ram_available_gauge
ram_available{client_no="63552"} 5.709529088e+09
# HELP ram_used ram used
# TYPE ram_used_gauge
ram_used{client_no="63552"} 34.4811268480827
# HELP clients_connected clients connected
# TYPE clients_connected_gauge
clients_connected 1.0

```

حال زمانی که یک کلاینت دیگر متصل شود :

```

server started at port 45678
Connection from: ('127.0.0.1', 63552)
{'cpu': 6.0, 'ram': 63.6, 'disk': 60.8, 'ram_total': 16558358528, 'ram_available': 6022291456, 'ram_used': 36.37009940216219, 'client_no': 63552}
{'cpu': 2.3, 'ram': 63.9, 'disk': 60.8, 'ram_total': 16558358528, 'ram_available': 5974257664, 'ram_used': 36.080011517431494, 'client_no': 63552}
{'cpu': 18.1, 'ram': 65.5, 'disk': 60.8, 'ram_total': 16558358528, 'ram_available': 5709529088, 'ram_used': 34.4811268480827, 'client_no': 63552}
{'cpu': 12.2, 'ram': 67.6, 'disk': 60.8, 'ram_total': 16558358528, 'ram_available': 5362659328, 'ram_used': 32.38641873185559, 'client_no': 63552}
{'cpu': 6.1, 'ram': 66.8, 'disk': 60.8, 'ram_total': 16558358528, 'ram_available': 5503295488, 'ram_used': 33.2357550942866, 'client_no': 63552}
Connection from: ('127.0.0.1', 63708)
{'cpu': 4.0, 'ram': 68.2, 'disk': 60.8, 'ram_total': 16558358528, 'ram_available': 5267980288, 'ram_used': 31.81462872114953, 'client_no': 63708}

```

```

# HELP python_gc_objects_collected_total Objects collected during gc
# TYPE python_gc_objects_collected_total counter
python_gc_objects_collected_total{generation="0"} 112.0
python_gc_objects_collected_total{generation="1"} 273.0
python_gc_objects_collected_total{generation="2"} 0.0
# HELP python_gc_objects_uncollectable_total Uncollectable object found during GC
# TYPE python_gc_objects_uncollectable_total counter
python_gc_objects_uncollectable_total{generation="0"} 0.0
python_gc_objects_uncollectable_total{generation="1"} 0.0
python_gc_objects_uncollectable_total{generation="2"} 0.0
# HELP python_gc_collections_total Number of times this generation was collected
# TYPE python_gc_collections_total counter
python_gc_collections_total{generation="0"} 41.0
python_gc_collections_total{generation="1"} 3.0
python_gc_collections_total{generation="2"} 0.0
# HELP python_info Python platform information
# TYPE python_info gauge
python_info{implementation="CPython",major="3",minor="9",patchlevel="2",version="3.9.2"} 1.0
# HELP cpu_cpu_percent
# TYPE cpu_gauge
cpu{client_no="63552"} 5.8
cpu{client_no="63708"} 13.7
# HELP ram_ram_percent
# TYPE ram_gauge
ram{client_no="63552"} 1.6558358528e+010
ram{client_no="63708"} 1.6558358528e+010
# HELP disk_disk_percent
# TYPE disk_gauge
disk{client_no="63552"} 60.8
disk{client_no="63708"} 60.8
# HELP ram_total_ram_total
# TYPE ram_total_gauge
# HELP ram_available_ram_available
# TYPE ram_available_gauge
ram_available{client_no="63552"} 5.385039872e+09
ram_available{client_no="63708"} 4.862861312e+09
# HELP ram_used_ram_used
# TYPE ram_used_gauge
ram_used{client_no="63552"} 32.52158034200043
ram_used{client_no="63708"} 29.368015578216617
# HELP clients_connected_clients_connected
# TYPE clients_connected_gauge
clients_connected 2.0

```

نمودار یک متریک :



```
ram_used(client_no="63552", instance="localhost:1234", job="my_other_job")  
ram_used(client_no="63708", instance="localhost:1234", job="my_other_job")
```

و برای تعداد بالاتر :

```

python_info{implementation="CPython",major="3",minor="8"}
# HELP cpu cpu percent
# TYPE cpu gauge
cpu{client_no="63552"} 4.8
cpu{client_no="63708"} 2.1
cpu{client_no="64015"} 4.8
# HELP ram ram percent
# TYPE ram gauge
ram{client_no="63552"} 1.6558358528e+010
ram{client_no="63708"} 1.6558358528e+010
ram{client_no="64015"} 1.6558358528e+010
# HELP disk disk percent
# TYPE disk gauge
disk{client_no="63552"} 60.8
disk{client_no="63708"} 60.8
disk{client_no="64015"} 60.8
# HELP ram_total ram total
# TYPE ram_total gauge
# HELP ram_available ram available
# TYPE ram_available gauge
ram_available{client_no="63552"} 5.208993792e+09
ram_available{client_no="63708"} 5.21904128e+09
ram_available{client_no="64015"} 5.210574848e+09
# HELP ram_used ram used
# TYPE ram_used gauge
ram_used{client_no="63552"} 31.45839476293287
ram_used{client_no="63708"} 31.5190740143394
ram_used{client_no="64015"} 31.467943149008253
# HELP clients_connected clients connected
# TYPE clients_connected gauge
clients_connected 3.0

```

ram_used{client_no="63552", instance="localhost:1234", job="my_other_job"}	30.806705044912047
ram_used{client_no="63708", instance="localhost:1234", job="my_other_job"}	30.656924014636243
ram_used{client_no="64015", instance="localhost:1234", job="my_other_job"}	30.812666602021288

