



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Лабораторна робота №4
з дисципліни
«Проектування розподілених систем»
на тему:
«Мікросервиси з використанням Messaging queue»

Виконав:
студент групи ФБ-31мп
Щур Павло
Перевірив:
Родіонов А. М.

Посилання на GitHub: <https://github.com/ShchurPavlo/distributed-systems-design-2024/tree/main/lab4>

Виконання завдань:

1) Модифікуємо Python код message service з попередніх лабораторних робіт. Додамо туди функціонал задавання порту роботи та обробки Hazelcast Distributed Queue :

```
from flask import Flask, jsonify, request
import hazelcast
import argparse
import threading

import requests
parser = argparse.ArgumentParser()
parser.add_argument("--port", type=int, required=True)
args = parser.parse_args()

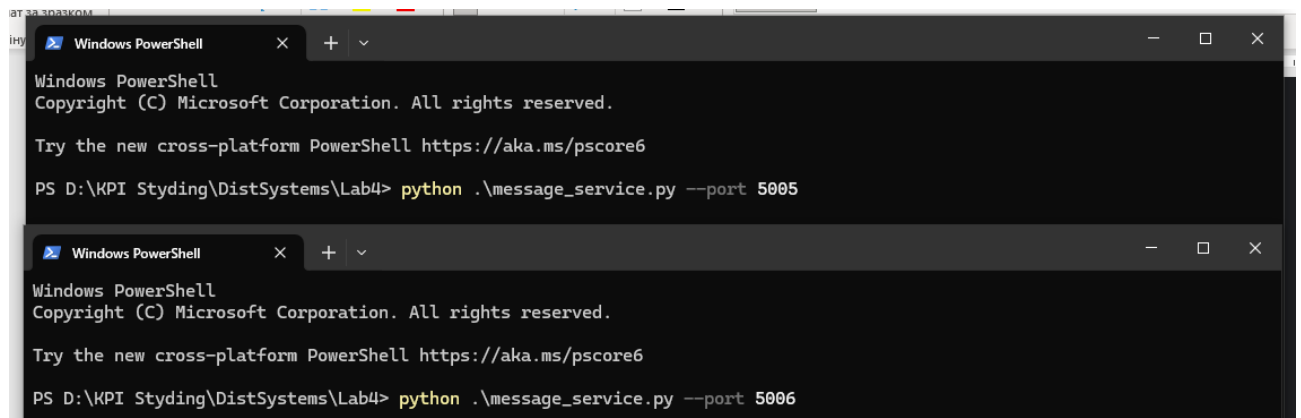
hz = hazelcast.HazelcastClient(cluster_name="ps_cluster", cluster_members=[])
messages_queue = hz.get_queue("queue").blocking()

app = Flask(__name__)
message_list=[]
def queue_event():
    while True:
        item = messages_queue.take()
        message_list.append(item)
        print("Recieved: ", str(item))

@app.route('/data', methods=['GET', 'POST'])
def data():
    if request.method == 'GET':
        return '\n'.join(message_list)
    else:
        return jsonify({'error': 'Bad request'})

if __name__ == '__main__':
    event_thread = threading.Thread(target=queue_event)
    event_thread.start()
    app.run(debug=True, port=args.port)
```

Запустимо два екземпляри message service на 5005 та 5006 порти:



Вони працюють в режимі зчитування повідомлень з черги.

Також модифікуємо facade service, додамо туди функціонал випадкового вибору з яким екземпляром messages service працювати , а також запису повідомлень в чергу:

```
from flask import Flask, request, jsonify
import random
import requests
import hazelcast

hz = hazelcast.HazelcastClient(cluster_name="ps_cluster", cluster_members=[])
messages_queue = hz.get_queue("queue").blocking()

def generate_unique_key():
    return ''.join(random.choices('123456789', k=4))

app = Flask(__name__)
@app.route('/data', methods=['GET', 'POST'])
def handle_data():
    logging_port = random.randint(5001, 5003)
    message_port = random.randint(5005, 5006)
    if request.method == 'GET':
        response_logging =
requests.get(f'http://127.0.0.1:{logging_port}/data', timeout=5)
        response_logging.raise_for_status()
        response_message =
requests.get(f'http://127.0.0.1:{message_port}/data').text
        return jsonify({'Message data': response_message, 'Log data':
response_logging.text})
    elif request.method == 'POST':
        message = request.get_data().decode('utf-8')
        messages_queue.offer(message)
        key=generate_unique_key()
        data = {"key": key, "msg": message}
        print(data)
        response = requests.post(f'http://127.0.0.1:{logging_port}/data',
data=data)
        print("Response:", response.text)
        data = response.text
        return data

if __name__ == '__main__':
    app.run(debug=True, port=5000)
```

Та запустимо його:

```
PS D:\KPI Styding\DistSystems\Lab3> python .\facade.py
* Serving Flask app 'facade'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 318-908-639
```

2) Запустимо 3 екземпляри logging-service та client.py:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\KPI Styding\DistSystems\Lab3> python .\logging_service.py --hzport 5703 --logport 5003
* Serving Flask app 'logging_service'
* Debug mode: off

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\KPI Styding\DistSystems\Lab3> python .\logging_service.py --hzport 5701 --logport 5001
* Serving Flask app 'logging_service'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\KPI Styding\DistSystems\Lab3> python .\logging_service.py --hzport 5702 --logport 5002
* Serving Flask app 'logging_service'
* Debug mode: off

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

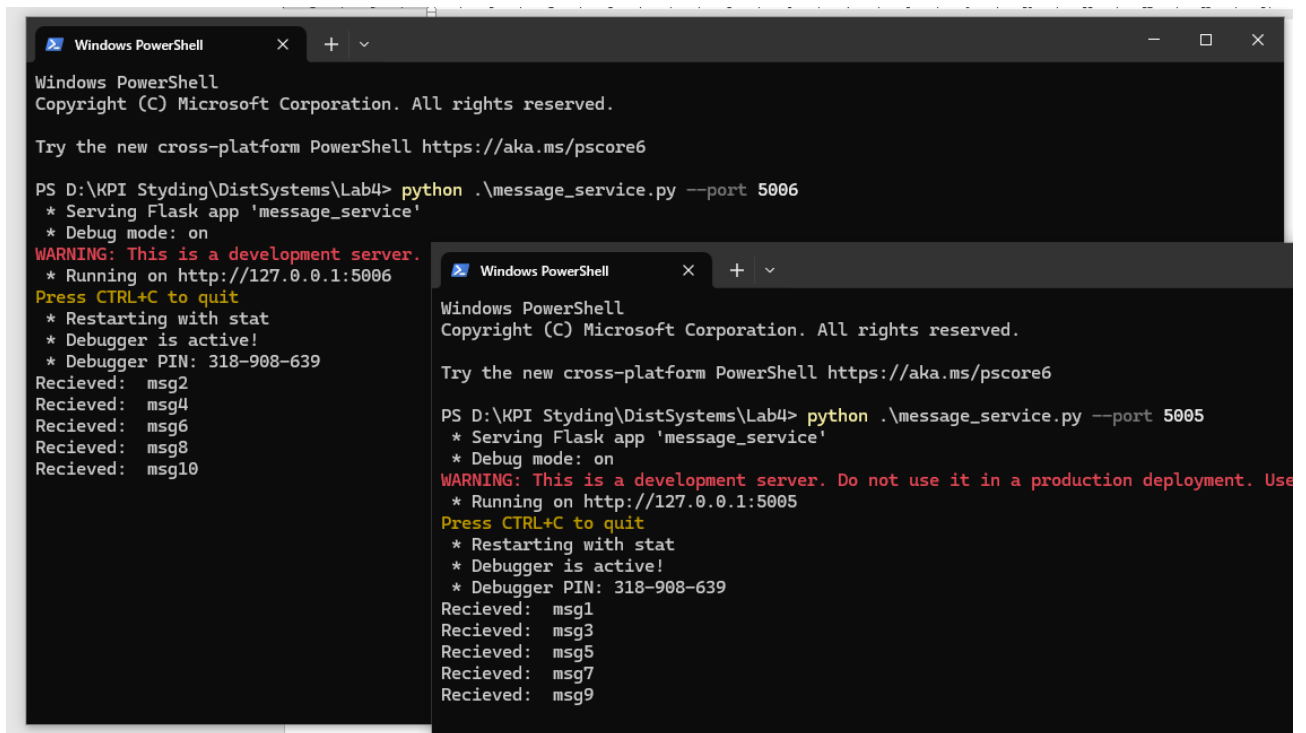
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\KPI Styding\DistSystems\Lab3> python .\client.py
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:
```

3) Запишемо 10 повідомлень msg1-msg10 через facade-service:

```
PS D:\KPI Styding\DistSystems\Lab4> python .\client.py
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:1
Message:msg1
POST відповідь: Success!
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:1
Message:msg2
POST відповідь: Success!
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:1
Message:msg3
POST відповідь: Success!
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:1
Message:msg4
POST відповідь: Success!
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:1
Message:msg5
POST відповідь: Success!
```

4) Перевіримо розподіл повідомлень на кожному з message service:



The image shows two side-by-side Windows PowerShell windows. The left window shows the execution of `python .\message_service.py --port 5006`. It displays the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\KPI Styding\DistSystems\Lab4> python .\message_service.py --port 5006
* Serving Flask app 'message_service'
* Debug mode: on
WARNING: This is a development server.
* Running on http://127.0.0.1:5006
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 318-908-639
Recieved: msg2
Recieved: msg4
Recieved: msg6
Recieved: msg8
Recieved: msg10
```

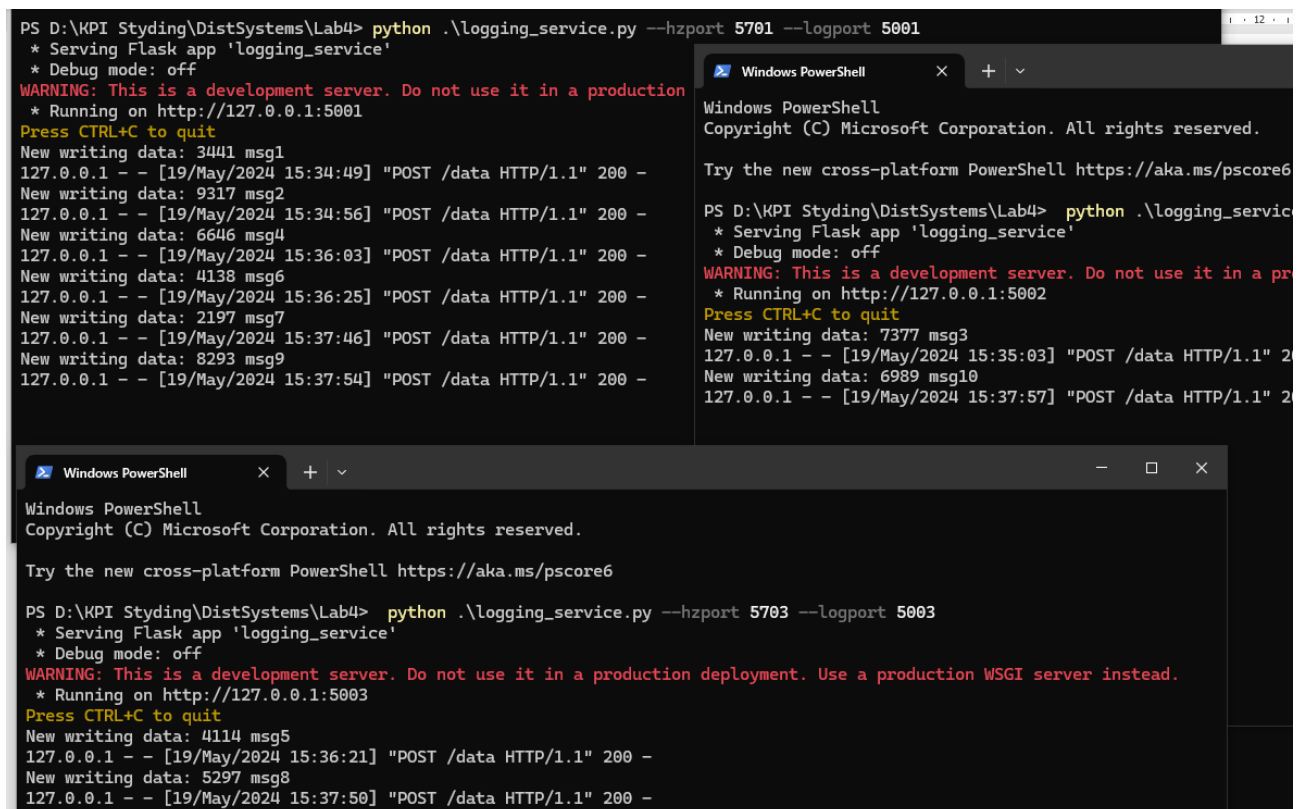
The right window shows the execution of `python .\message_service.py --port 5005`. It displays the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\KPI Styding\DistSystems\Lab4> python .\message_service.py --port 5005
* Serving Flask app 'message_service'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use
* Running on http://127.0.0.1:5005
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 318-908-639
Recieved: msg1
Recieved: msg3
Recieved: msg5
Recieved: msg7
Recieved: msg9
```

5) Перевіримо розподіл повідомлень на кожному з logging service:



The image shows three Windows PowerShell windows. The top window shows the execution of `python .\logging_service.py --hzport 5701 --logport 5001`. It displays the following output:

```
PS D:\KPI Styding\DistSystems\Lab4> python .\logging_service.py --hzport 5701 --logport 5001
* Serving Flask app 'logging_service'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production
* Running on http://127.0.0.1:5001
Press CTRL+C to quit
New writing data: 3441 msg1
127.0.0.1 - - [19/May/2024 15:34:49] "POST /data HTTP/1.1" 200 -
New writing data: 9317 msg2
127.0.0.1 - - [19/May/2024 15:34:56] "POST /data HTTP/1.1" 200 -
New writing data: 6646 msg4
127.0.0.1 - - [19/May/2024 15:36:03] "POST /data HTTP/1.1" 200 -
New writing data: 4138 msg6
127.0.0.1 - - [19/May/2024 15:36:25] "POST /data HTTP/1.1" 200 -
New writing data: 2197 msg7
127.0.0.1 - - [19/May/2024 15:37:46] "POST /data HTTP/1.1" 200 -
New writing data: 8293 msg9
127.0.0.1 - - [19/May/2024 15:37:54] "POST /data HTTP/1.1" 200 -
```

The middle window shows the execution of `python .\logging_service.py --hzport 5703 --logport 5003`. It displays the following output:

```
PS D:\KPI Styding\DistSystems\Lab4> python .\logging_service.py --hzport 5703 --logport 5003
* Serving Flask app 'logging_service'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5003
Press CTRL+C to quit
New writing data: 4114 msg5
127.0.0.1 - - [19/May/2024 15:36:21] "POST /data HTTP/1.1" 200 -
New writing data: 5297 msg8
127.0.0.1 - - [19/May/2024 15:37:50] "POST /data HTTP/1.1" 200 -
```

The bottom window shows the execution of `python .\logging_service.py --hzport 5701 --logport 5001`. It displays the following output:

```
PS D:\KPI Styding\DistSystems\Lab4> python .\logging_service.py --hzport 5701 --logport 5001
* Serving Flask app 'logging_service'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5001
Press CTRL+C to quit
New writing data: 7377 msg3
127.0.0.1 - - [19/May/2024 15:35:03] "POST /data HTTP/1.1" 200 -
New writing data: 6989 msg10
127.0.0.1 - - [19/May/2024 15:37:57] "POST /data HTTP/1.1" 200 -
```

6) Виконаємо декілька разів запит Get та пересвідчимось у відмінності отримання даних з message service:

```
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:2
GET відповідь: {'Log data': 'msg6\nmsg3\nmsg9\nmsg4\nmsg10\nmsg8\nmsg5\nmsg2\nmsg1\nmsg7', 'Message data': 'msg4\nmsg8'}
1)Sent POST-request
2)Sent GET-request
3)Exit
Your choose:2
GET відповідь: {'Log data': 'msg6\nmsg3\nmsg9\nmsg4\nmsg10\nmsg8\nmsg5\nmsg2\nmsg1\nmsg7', 'Message data': 'msg3\nmsg7'}
1)Sent POST-request
2)Sent GET-request
3)Exit
```