



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота №2
з дисципліни “Бази даних”
Варіант 7

Виконав
студент 3 курсу
групи КП-81
Каснер Максим

Київ 2021

GitHub repo - <https://github.com/kasmaks/dbLabs2/tree/main/lab2>

Умова завдання

Окремі програмні компоненти та вимоги до них

1. Redis server (RS), що виконує наступні ролі:

- 1.1. *Сховище*, що містить: дані користувачів, їхні групи (звичайний користувач та адміністратор), а також повідомлення, що пересилаються між ними.
- 1.2. *Черга повідомлень*, які підлягають перевірці на спам та відправленню адресату.
- 1.3. Інструмент *Publish/Subscribe* для ведення та розсилання журналу активності користувачів (див. *Список активностей для журналювання*).

2. Інтерфейс користувача (User Interface)

2.1. *Звичайний користувач* має змогу виконувати вхід за ім'ям (без паролю), відправляти та отримувати (переглядати) повідомлення, отримувати дані про кількість **своїх** повідомлень, згрупованих за статусом (див. *Статуси повідомлень*).

2.2. *Адміністратор* має змогу переглядати журнал подій, що відбулись (див. *Список активностей для журналювання*), переглядати список користувачів, які знаходяться online, переглядати статистику (N найбільш активних відправників повідомлень із відповідною кількістю, N найактивніших "спамерів" із відповідною кількістю).

3. Виконувач (worker) призначений для:

перегляду черги повідомлень, відбору повідомлення, перевірки його вмісту на наявність спаму (у випадку наявності спаму -- додавання запису в журнал)

Інші вимоги

1. Проаналізувавши матеріали ресурсів, наведений у пункті “Джерела”, обрати та обґрунтувати вибір структур даних Redis щодо реалізації наведених вище вимог, **обов’язково використати наступні структури даних** та інструменти Redis: List, Hash, Sorted List, Set, Pub/Sub.
2. Забезпечити роботу програмних засобів у режимі емуляції із можливістю генерації повідомлень від різних користувачів, налаштування кількості виконувачів та часу затримки обробки на спам з можливістю підключення адміністратора для перегляду подій, що відбуваються.
3. Перевірку на спам можна проемувати за допомогою затримки на псевдовипадковий час та генерацію псевдовипадкового результату (Так/Ні).

Список активностей для журналювання

Вхід/вихід користувача, наявність спаму у повідомленні.

Статуси повідомлень

“Створено”, “У черзі”, “Перевіряється на спам”, “Заблоковано через спам”, “Відправлено адресату”, “Доставлено адресату”.

Вимоги до інтерфейсу користувача

Використовувати консольний (текстовий) інтерфейс користувача.

- 1) Для збереження об'єкта користувача було використано хеш-таблицю.
- 2) Для збереження списку всіх користувачів (тобто збереження їхніх унікальних логінів) було використано SET - в цьому випадку доцільно використовувати SET, тому що ми не сортуємо всіх користувачів, а також легко доступатись за значенням.
- 3) List було використано при реалізації черги повідомлень, які потім обробляв програмний засіб (в списку зберігалися унікальні ідентифікатори повідомлень, а самі повідомлення з їхніми атрибутами, як структуру даних, було вирішено зберігати у хеш-таблиці)
- 4) Sorted List було використано при реалізації рейтингу найактивніших спамерів та найактивніших користувачів.
- 5) Вбудований шаблон Publish/Subscribe було використано для того щоб програмний засіб Worker отримував сповіщення появи нового повідомлення - був створений відповідний канал який робив publish повідомлень, а Worker робив subscribe цей канал.

Результати роботи програми

```
Choose an option:
  0. Admin 0
> 1. Simple User 1 <
  2. Worker 2
  3. Exit 3
Please enter your username:
kasner
```

Рис.1 Вхід у якості звичайного користувача

```
Choose an option:
  0. kasner 0
> 1. kasner maksym 1 <
kasner maksym
kasner maksym
send_a_message
Me, myself and I|
```

Рис.2 Відправка повідомлення користувачем

```
It is a spam _from_kasner maksym_to_kasner maksym_2021-06-04 22:02:23.577984
Approved Message, _from_kasner maksym_to_kasner_2021-06-04 22:02:44.346618
```

Рис.3 Відображення роботи Worker (повідомлення може бути підтверджене або заблоковане через псевдоспам)

```
from: kasner maksym
to: kasner
date of sending: 2021-06-04 22:01:37.422114
message: Hello, kasner, how are you?
from: kasner maksym
to: kasner
date of sending: 2021-06-04 22:01:55.846328
message: I miss you!
from: kasner maksym
to: kasner
date of sending: 2021-06-04 22:02:44.346618
message: Send something to me, please!
```

Рис.4 Користувач має змогу читати повідомлення від інших користувачів

```
messages_are_blocked_due_to_spam
from: kasner maksym
to: kasner
date of sending: 2021-06-04 22:02:11.754525
message: Hope you're doing great
from: kasner maksym
to: kasner maksym
date of sending: 2021-06-04 22:02:23.577984
message: Really looking forward to meet ya
from: kasner maksym
to: kasner maksym
date of sending: 2021-06-04 22:07:50.766117
message: Me, myself and I!
```

Рис.5 Повідомлення, що були заблоковані через спам

```
sent_but_not_read_messages
from: kasner
to: kasner
date of sending: 2021-06-04 22:14:46.607787
message: aaaaa
from: kasner maksym
to: kasner
date of sending: 2021-06-04 22:17:03.491990
message: hiiii
from: kasner maksym
to: kasner
date of sending: 2021-06-04 22:17:15.409855
message: simply forgotten
```

Рис.6 Повідомлення, що ще не були прочитані

```
kasner 2021-06-04 22:00:11.541790 user logged in
kasner 2021-06-04 22:00:28.507276 user logged out
kasner maksym 2021-06-04 22:01:03.811521 user logged in
kasner maksym 2021-06-04 22:03:17.521898 user logged in
kasner 2021-06-04 22:05:27.506605 user logged in
kasner maksym 2021-06-04 22:06:43.827421 user logged in
kasner 2021-06-04 22:11:06.547699 user logged in
kasner maksym 2021-06-04 22:15:44.908011 user logged in
kasner 2021-06-04 22:17:40.528603 user logged in
```

Рис.7 Список логів

```
rating_of_active_users
('kasner maksym', 8.0)
('kasner', 7.0)
```

Рис.8 Рейтинг активних юзерів

```
rating_of_active_spammers  
( 'kasner', 4.0)  
( 'kasner maksym', 3.0)
```

Рис.9 Рейтинг спамерів

Загалом програма може працює в режимі декількох вікон, за допомогою чого можна ввімкнути Worker, який буде перевіряти повідомлення або ж адміністратора, який буде відслідковувати кількість користувачів онлайн й іншу статистику. Спам було реалізовано за допомогою генерації псевдовипадкового числа, тобто те чи є повідомлення спамом навіть не залежить від самого повідомлення)

Контрольні запитання

1. *Визначити сфери застосування основних структури даних redis (List, Hash, Sorted List, Set, Pub/Sub).*

Hash ми використовуємо, коли хочемо зберегти об'єкт-орієнтовані дані, тобто ті, які містять якісь атрибути, Set ми використовуємо, коли нам важливий пошук за значенням і не має сенсу порядок розташування елементів - це може бути для збереження логінів всіх користувачів, Sorted List - доцільний при побудові користувацький рейтингів, коли до значення прив'язується числа, і залежно від величини числа відбувається сортування. Pub/Sub доцільно використовувати у швидких чатах або швидких розсилках, коли трафік дуже великий але водночас він і не дуже цінний - роботу потрібно робити швидко і безпека та цілісність даних не є пріоритетом - підписник отримує повідомлення з каналу зв'язку, на який було опубліковано повідомлення.

2. *Визначити основні переваги та недоліки redis.*

Оскільки це key-value database, то вона забезпечує дуже швидку роботу з даними і доступ до них. Також за допомогою цієї бд можна створювати різні real-time застосунки для обміну повідомленнями. Найбільше при роботі мені сподобався зручний доступ до даних, при роботі було використано бібліотеку redis мови Python.