

# РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ КОПІЮВАННЯ ДАНИХ МІЖ ХМАРНИМИ СХОВИЩАМИ

Вінницький національний технічний університет

## *Анотація*

*В роботі проведено дослідження переваг використання бази даних Redis для розробки програмного забезпечення автоматизованої системи копіювання даних на прикладі сервісу для створення резервних копій файлів та копіювання між хмарними сховищами. Було показано, що дана база даних завдяки наявності механізму реалізації черги повідомлень суттєво скорочує використання ресурсів та забезпечує просту інтеграцію.*

**Ключові слова:** база даних, NoSQL, Redis, Amazon SQS, Amazon DynamoDB, MySQL, SQL, Windows Service, черга повідомлень.

## *Abstract*

*In this scientific work, we reviewed the advantages of using a database Redis for developing automated data backup system service which creates backup copies of files. This DB due to the ability to create a message queues which will reduce the use of the resources. Redis is the most simple way to integrate message queues to any kind of application and programming languages.*

**Keywords:** Data Base, NoSQL, Redis, Amazon SQS, Amazon DynamoDB, MySQL, SQL, Windows Service, message queue.

## **Вступ**

Сьогодні великі компанії для зберігання масштабних об'ємів інформації все частіше використовують хмарні сховища, які забезпечують можливість зберігання значної кількості даних та високу швидкість доступу до них [1]. Метою цієї роботи є розробка системи, яка забезпечить можливість автоматизованого резервного копіювання файлів між такими хмарними сховищами.

## **Результати досліджень**

Для виконання роботи по копіюванню файлів був розроблений Windows сервіс, який мав на вході дані, за допомогою яких можна було отримати файл з одного хмарного сховища і координати цільового для завантаження резервної копії.

Інформацію про нову задачу копіювання з повним переліком файлів було вирішено зберігати у базі даних, а тому були досліджені такі параметри як: швидкість, ціна, ресурсозатратність, наявність черги повідомлень у наступних БД – Redis, MySQL, Amazon DynamoDb, Amazon SQS.

Windows-сервіс, у свою чергу, має перевіряти БД на наявність нових задач та записувати результати процесу виконання так само в базу даних.

При розгляді MySQL бази даних нас не влаштувала відсутність можливості реалізації будь-яких механізмів черги повідомлень, а тому сервіс буде вимушений перевіряти циклічно записи у БД для виявлення нових задач [2]. Така циклічна перевірка буде забирати велику кількість ресурсів, тому від цієї моделі було вирішено відмовитись.

Amazon DynamoDB позиціонує себе як NoSQL база даних з високою швидкістю [3]. Проте як і MySQL вона не має механізмів для реалізації черги повідомлень. Для цього у парі з цією БД найдоречніше було б використовувати сервіс для реалізації черг повідомлень Amazon SQS.

Остання розглянута NoSQL база даних Redis включає в себе механізм реалізації черг, який в контексті цієї БД називається «Redis-pubsub» [4]. Redis надає зручний інтерфейс для команд publish і subscribe, які дозволяють прослуховувати повідомлення на заданому «каналі». Ми можемо публікувати повідомлення в цей канал з будь-якого додатку і навіть використовуючи будь-яку мову програмування, що забезпечує простий зв'язок між тим хто відсилає нову задачу копіювання і сервісом, що приймає її і виконує [5]. Загальний вигляд роботи такої системи наведено на рисунку 1.

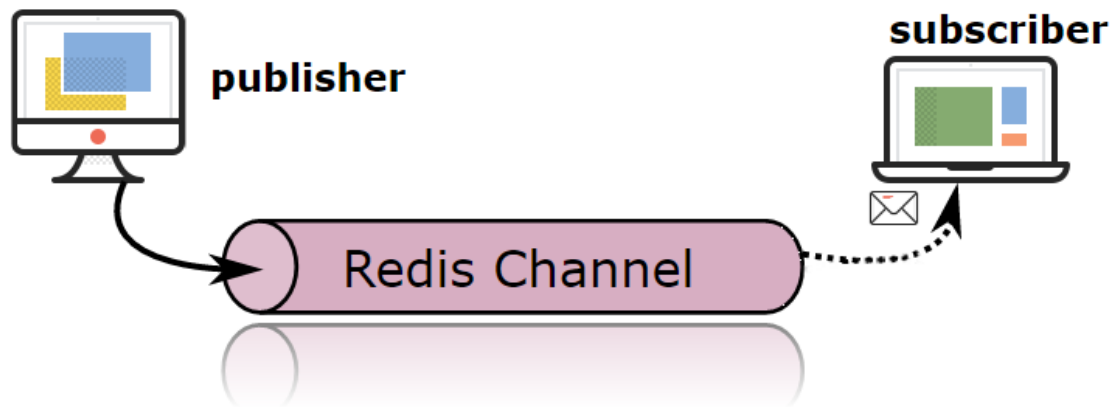


Рис. 1. Загальний вигляд роботи «Redis-pubsub»

На відміну від SQL баз даних, використовуючи «Redis-pubsub» нам не потрібно ініціалізувати безкінечний цикл перевірки, а всього лише «підписатись» на потрібний канал і очікувати повідомлення про нову задачу. Водночас ми можемо використовувати Redis для зберігання теперішнього стану виконання копіювання.

Таким чином, на відміну від моделі з використанням Amazon сервісів нам не потрібно використовувати два сервіси, що зекономить гроші і ресурси.

### Висновки

Використання бази даних Redis є найбільш доречним при реалізації сервісу, який має підтримувати роботу з чергою повідомлень та одночасного запису інформації про теперешній стан задачі.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Paul DuBois. MySQL Cookbook: Solutions for Database Developers and Administrators 3rd Edition / Paul DuBois – 2016 – С. 1-20.
2. Seyed M.M. (Saied) Tahaghoghi. Learning MySQL: Get a Handle on Your Data / Seyed M.M. (Saied) Tahaghoghi, Hugh Williams – 2016 – С. 1–110.
3. Andreas Wittig. Amazon Web Services in Action 1st Edition / Andreas Wittig – 2016 – С. 5-22
4. Josiah L. Carlson. Redis in Action / Josiah L. Carlson – 2016 – С. 40–52
5. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems 1st Edition / Martin Kleppmann – 2017 – С. 1-200

**Костенецький Кирило Валерійович** – студент групи 2AB-136, факультет комп’ютерних систем і автоматики, Вінницький національний технічний університет, Вінниця, e-mail: kirill.kosteneckiy@mail.ru

**Гришук Тетяна Вікторівна** – к. т. н, доцент, доцент кафедри комп’ютерних систем управління, Вінницький національний технічний університет, м. Вінниця

**Kostnetskiy Kyrylo V.** — Department of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, email: kirill.kosteneckiy@mail.ru

**Tetiana Gryshchuk** – PhD, associate professor, lecturer of the Chair Computer Control Systems, Vinnytsia National Technical University, Vinnytsia