

Б. О. ЄРОХІН

студент

Харківський національний університет радіоелектроніки
ORCID: 0009-0001-3224-550X

М. О. ВОЛК

доктор технічних наук, професор,

професор кафедри електронних обчислювальних машин
Харківський національний університет радіоелектроніки
ORCID: 0000-0003-4229-9904

ВИБІР ТЕХНОЛОГІЧНОГО СТЕКУ РОЗРОБКИ ПРИВАТНОЇ ХМАРНОЇ СИСТЕМИ ПАРАЛЕЛЬНОГО ЗБОРУ ТА ОБРОБКИ ІНФОРМАЦІЇ

Дана наукова стаття присвячена вибору технологічного стеку розробки приватної хмарної системи паралельного збору та обробки інформації. Було проаналізовано наявні системи збору та обробки інформації з погодних датчиків і виявлено, що слабкою стороною всіх централізованих систем є їхня пропускна здатність. У роботі досліджено низку технологій, що використовуються у розробці хмарних систем збору та обробки інформації, зокрема Microsoft Hyper-V, PostgreSQL, MySQL, Golang, Java, Python, C++, Node.js та розглянуто їх переваги та недоліки. Також проаналізовано основні вимоги до хмарної системи паралельного збору та обробки інформації, такі як масштабованість, невелика вартість, продуктивність, а також легкість підтримки такої системи. На основі проведеного аналізу запропоновано оптимальний технологічний стек для розробки хмарної системи, що відповідає вимогам та забезпечує ефективну роботу системи, а саме систему апаратної віртуалізації Microsoft Hyper-V, через те що вона є вбудованою в усі останні версії Windows Server, а також через підтримку всіх популярних операційних систем в ролі гостьових ОС, систему управління базами даних PostgreSQL, через її високу продуктивність та безкоштовність, що є гарним показником для розробки систем для компаній у яких не великий бюджет для розгортання таких систем, як мову програмування серверу було обрано Golang, як мову в якій є дуже потужні вбудовані інструменти паралелізму, які дозволяють легко розробляти системи, які підтримують роботу не тільки з багатьма потоками, а навіть із декількома ядрами, а також через її досить простий синтаксис, що дозволяє підтримувати таку систему, навіть спеціалістам, які мають не дуже великий досвід роботи з цією мовою програмування. Результати дослідження можуть бути корисними для розробників приватних хмарних систем і не тільки, а також для інженерів в області паралельного збору та обробки інформації.

Ключові слова: приватна хмарна система, Golang, Microsoft Hyper-V, PostgreSQL, MySQL, Java, Python, C++, Node.js збір та обробка інформації, паралелізм, СУБД, апаратна віртуалізація.

B. O. YEROKHIN

Student

Kharkiv National University of Radio Electronics
ORCID: 0009-0001-3224-550X

M. O. VOLK

Doctor of Technical Sciences, Professor,

Professor at the Department of Electronic Computers
Kharkiv National University of Radio Electronics
ORCID: 0000-0003-4229-9904

THE CHOICE OF THE TECHNOLOGICAL STACK IN THE DEVELOPMENT OF A PRIVATE CLOUD SYSTEM FOR PARALLEL COLLECTION AND PROCESSING OF INFORMATION

This scientific article is devoted to the choice of the technological stack for the development of a private cloud system for parallel collection and processing of information. The existing systems for collecting and processing information from weather sensors were analyzed and it was found that the weakness of all centralized systems is their throughput. The paper investigates a few technologies used in the development of cloud systems for collecting and processing information, in particular Microsoft Hyper-V, PostgreSQL, MySQL, Golang, Java, Python, C++, Node.js and considers their advantages and disadvantages. The basic requirements for the cloud system of parallel information collection and processing, such as scalability, low cost, performance, as well as ease of maintaining such a system. Based on the analysis, an optimal technological stack for the development of a cloud system that meets the requirements and ensures efficient operation of

the system, namely the Microsoft Hyper-V hardware virtualization system, has been proposed, since it is built into all the latest versions of Windows Server, as well as support for all popular operating systems as guest OSes, the PostgreSQL database management system, due to its high performance and freeness, which is a good indicator for developing systems for companies that do not have a large budget to deploy systems such as the server programming language Golang has been chosen, as a language that has very powerful built-in parallelism tools that make it easy to develop systems that support working not only with many threads, but even with multiple cores, and also because of its fairly simple syntax that allows you to support such a system, even for specialists who do not have much experience with this programming language. The results of the research may be useful for developers of private cloud systems and not only, as well as for engineers in the field of parallel information collection and processing.

Key words: private cloud system, Golang, Microsoft Hyper-V, PostgreSQL, MySQL, Java, Python, C++, Node.js information acquisition and processing, parallelism, DBMS, hardware virtualization.

Постановка проблеми

Значення хмарних систем паралельного збору та обробки інформації в сучасному світі з кожним днем набирають популярність. Такі системи використовуються всюди, починаючи від моніторингу стану якогось обладнання на підприємстві і закінчуючи системами інформаційної підтримки та ситуаційної обізнаності на полі бою. Хмарні системи паралельного збору та обробки інформації з метеорологічних датчиків не є винятком.

Збір інформації з метеорологічних датчиків – це важлива задача, яка вимагає надійної і ефективної системи збору даних. Нижче наведено огляд кількох існуючих систем збору інформації з метеорологічних датчиків:

- Система збору даних з метеорологічних датчиків Davis Instruments [1] Система Davis Instruments є однією з найпопулярніших і надійних систем збору даних з метеорологічних датчиків. Вона включає в себе бездротові сенсори, що вимірюють температуру, вологість, швидкість вітру, напрямок вітру, опади, сонячну радіацію та інші показники, а також спеціальний приймач, що збирає дані в режимі реального часу.

- Система збору даних з метеорологічних датчиків Vaisala [2].

- Система збору даних з метеорологічних датчиків Vaisala використовується для моніторингу метеорологічних умов в різних відкритих просторах. Система включає в себе бездротові сенсори, що вимірюють температуру, вологість, швидкість вітру, напрямок вітру, тиск та опади, а також цифровий приймач, що збирає дані в режимі реального часу.

- Система збору даних з метеорологічних датчиків Campbell Scientific Система збору даних з метеорологічних датчиків Campbell scientific [3] є однією з найбільш розповсюджених систем збору даних галузі метеорології. Вона включає в себе бездротові та провідні сенсори, що вимірюють температуру, вологість, швидкість вітру, напрямок вітру, тиск та опади, а також інші параметри, такі як рівень води, рівень снігу, сонячна радіація та інші. Система включає в себе спеціальне програмне забезпечення, яке забезпечує збір даних, аналіз та візуалізацію результатів в режимі реального часу.

Кожна з цих систем має свої переваги та недоліки, і вибір конкретної системи залежить від потреб користувача. Однак, загалом, всі ці системи мають спільну мету – забезпечити точні та актуальні метеорологічні дані для користувачів. Вибір системи може залежати від таких чинників, як вартість, точність, розмір та зручність встановлення датчиків, доступність сервісів метеорологічних даних та інші. Важливо зазначити, що збір та аналіз метеорологічних даних може бути корисним для різних галузей, таких як сільське господарство, енергетика, транспорт, будівництво та інші.

Хмарні системи в останні десятиліття розвиваються з небаженою швидкістю, з кожним роком вони стають все більш високонавантаженими. Такі системи вже давно об'єднують в хмарні кластери задля розподілення задач, щоб збільшити їхню продуктивність. Це і було одною з причин чому компанія Google розробила мову Golang[4], адже всі існуючі на той час мови, хоча і були достатньо продуктивними (наприклад, C++), але в жодній з мов не було вбудованого інструментарію для паралельної роботи з декількома потоками чи навіть ядрами процесору. Саме через це, у 2009 році Google показала світу нову мову програмування з неймовірно продуктивною реалізацією паралелізму за допомогою горутин (goroutine), які через свої невеликі розміри (лише 4 Кб на одну горутину) дають змогу навіть на середньо продуктивних серверах працювати з декількома тисячами паралельних «потоків». Саме через таку популярність хмарних систем зрозуміло, що саме на «хмарах» треба будувати системи паралельного збору та обробки інформації.

У більшості продуктів для збору та обробки інформації такий принцип роботи: дані з датчика відправляються на сервер, який записує їх в базу даних, далі сторонні програми обробляють отриману інформацію. Проте, вагомий недолік в такій роботі – при збільшенні кількості датчиків, які має обслуговувати сервер, лінійно збільшується необхідна мінімальна потужність серверного обладнання і, відповідно, вартість обслуговування, що призводить до зменшення конкурентоспроможності.

Основними недоліками такого підходу є наявність обмеження в кількості конкурентних підключень до бази даних та необхідність у великій кількості серверів, які мають обслуговувати усі підключення.

Формулювання мети дослідження

Мета даної роботи полягає у аналізі та обґрунтуванні вибору технологій для роботи з хмарними системами паралельного збору та обробки інформації та розробки рекомендації щодо впровадження тієї чи іншої технології в сис-тему для забезпечення більшої продуктивності програмного забезпечення ви-соконавантажених систем збору та обробки інформації. Не менш важливою метою є розробка такої система, яка давала б змогу використо-вувати її не тільки компаніям з великими матеріальними ресурсами, а й тим організаціям, які тільки починають свій розвиток у сфері хмарних систем збору та обробки інформації з метеорологічних датчиків, тому однією з цілей було обрати такий технологічний стек проекту, який вимагав би найменшу кількість капіталов-кладень для її розгортання.

Виклад основного матеріалу дослідження

Хмарна системи паралельного збору та обробки інформації буде складатися з трох віртуальних машин, а саме:

- Віртуальна машина на якій буде розміщено програму емулятор передачі даних. Ця машина буде викорис-товуватися для того що генерувати показання температурних датчиків, щоб потім з цих даних було сформовано JSON запит на сервер обробки.
- Віртуальна машина на якій буде розміщено REST API сервер. Ця машина буде використовуватися збору даних з усіх температурних датчиків і подальшу обробку цих даних, наприклад, передача в потрібну базу даних організації, яка буде використовувати цю систему.
- Віртуальна машина на якій буде розгорнуто базу даних. Ця машина буде використовуватися для зберігання даних.

Всі три віртуальні машини можуть бути налаштовані для роботи в приватній хмарі або гібридна хмара.

В цілому система хмарних обчислень, що складається з трьох віртуальних машин, дозволяє ефективно управ-ляти ресурсами, підвищувати гнучкість і масштабованість системи, а також підвищувати безпеку і надійність обчислювальних процесів.

Оскільки система збору та обробки інформації повинна бути доступна всім, та забезпечувати високу про-пускну здатність було запропоновано про-аналізувати дві популярні, реляційні [4] та безкоштовні СУБД, а саме PostgreSQL та MySQL.

PostgreSQL і MySQL – це дві популярні системи управління базами да-них, які використовуються в багатьох додатках. Обидві системи мають свої

переваги та недоліки, проте PostgreSQL має кілька ключових переваг порів-няно з MySQL, це видно в таблиці 1 [5, 6]:

Таблиця 1

Порівняння СУБД PostgreSQL та MySQL за деякими чинниками

Чинник	PostgreSQL	MySQL
Архітектура	Об'єктно-реляційні; багатопроцесний	Реляційних; Єдиний процес
Підтримувані типи даних	Numeric, Date/time, Character, JSON, Boolean, Enumerated, XML, Geometric, Arrays, Ranges, Мережева адреса, HSTOREComposite	Numeric, Date/time, Character, JSON, Spatial
Підтримувані індекси	B-tree, Hash, GiST, SP-GiST, GIN, BRIN	B-tree, Hash, R-tree, Інвертовані індекси
Безпека	Добре справляється з великою кількістю читань і записів	Добре обробляє велику кількість зчитувань
Продуктивність	Керування доступом Кілька зашифрованих параметрів	Керування доступом Зашифровані параметри
Технічна підтримка	Підтримка спільноти Платна підтримка від певних постачальників	Підтримка спільноти Платна підтримка від певних постачальників

Крім цього, PostgreSQL має більш високий рівень безпеки та підтримку ACID-властивостей (Atomicity, Consistency, Isolation, Durability), що забезпе-чує стійкість та надійність бази даних. Однак, MySQL має менші вимоги до ресурсів та може працювати швидше на менших серверах.

Загалом, обидві системи мають свої переваги та недоліки, і вибір між ними залежить від конкретних потреб користувача та його проекту. Однак, для більш складних проектів з великими обсягами даних та потребою в висо-кій безпеці та надійності, PostgreSQL може бути кращим вибором.

Зважаючи на те, що кожна з мов має свої переваги та недоліки, наведемо декілька загальних переваг та недо-ліків кожної мови у вигляді таблиці для по-рівняння.

Зважаючи на те, що кожна з мов має свої переваги та недоліки, наведемо декілька загальних переваг та недо-ліків кожної мови у вигляді таблиці 2 для порівняння.

Таблиця 2

Переваги та недоліки найбільш популярних мов програмування веб-серверів

Мова програмування	Переваги	Недоліки
Go	Ефективне використання пам'яті, вбудована підтримка багатопоточності та конкурентності, простий та лаконічний синтаксис, швидкий час виконання	Обмежена кількість бібліотек порівняно з іншими мовами
Java	Багата екосистема бібліотек та фреймворків, висока переносимість, гарна підтримка об'єктно-орієнтованого програмування	Високі вимоги до ресурсів, повільний час запуску
Python	Простий та легкий синтаксис, багата екосистема бібліотек та фреймворків, висока продуктивність в області науки даних	Повільніший час виконання порівняно з іншими мовами, не ефективне використання пам'яті
C++	Висока продуктивність та ефективне використання пам'яті, багата екосистема бібліотек та фреймворків, гнучка підтримка об'єктно-орієнтованого та процедурного програмування	Складний синтаксис, більш складна розробка порівняно з іншими мовами
Node.js	Висока швидкість та продуктивність, підтримка асинхронного програмування, багата екосистема бібліотек та фреймворків	Не підходить для багато поточних та конкурентних додатків, більш складна розробка порівняно з іншими мовами

Проаналізувавши дані таблиці ще можна додати декілька переваг Go по-рівню з іншими мовами програмування:

- Ефективність пам'яті: Go має більш ефективне використання пам'яті порівняно з іншими мовами, такими як Java та Python. Це забезпечує менший розмір бінарних файлів та менші витрати на зберігання даних.
- Конкурентність: Go має вбудовану підтримку багатопоточності та конкурентності, що робить його особливо підходящим для розробки високопродуктивних додатків, які обробляють багато запитів одночасно.
- Простота: Go має простий та лаконічний синтаксис, що дозволяє швидше розробляти та тестувати код. Він також має меншу кількість ключових слів та вбудованих функцій порівняно з іншими мовами, такими як C++.
- Швидкість: Go має швидкий час виконання та високу продуктивність, порівняно з іншими мовами, такими як Python та Node.js. Це робить його особливо підходящим для додатків, що вимагають високої швидкості та продуктивності.

Отже, проаналізувавши всі переваги та недоліки [7-9] було обрано мову програмування Golang через те що ця мова має вбудовані інструменти для роботи з декількома потоками та навіть декількома ядрами процесора, ще одним із плюсів є її простота, а це значить, що навіть програмісти, які мають не дуже великий рівень знань зможуть підтримувати систему в належному стані.

Для розгортання системи апаратної віртуалізації для x64-систем на основі гіпервізора було обрано Microsoft Hyper-V. Цю систему віртуалізації було обрано через те що вона є вбудованою системою віртуалізації в усіх останніх версіях будь-якої ОС на базі Windows. Hyper-V підтримує гостьові операційні системи Windows і Linux і надає такі функції, як жива міграція, висока доступність і віртуалізація мережі. Він також підтримує інтеграційні служби, які забезпечують зв'язок між хостом і гостьовими операційними системами. Саме через те що Hyper-V вбудовано в Windows Server 2022, через його гарну інтеграцію між хостом та гостьовою системою, цей гіпервізор було обрано для розгортання віртуальної хмари для системи паралельного збору та обробки даних з метеорологічних датчиків.

Тестування системи проводилось на локальному сервері, який було розгорнуто за допомогою Microsoft Hyper-V у віртуальному середовищі. Для тес-тового стенду було згенеровано 10 сенсорів, 365 днів та 24 години в одному дні отже загалом було. Сервіс генерації точок був налаштований на генерацію нових точок для кожного окремого замовлення з частотою 1000 генерацій в секунду. В тестовому стенді використовувались комп'ютер з системою на кри-сталі Intel Core I5-3230M та 9 Гб оперативної пам'яті. Навантаження на сис-тему було розподілене між віртуальною машиною, на якій знаходився сервіс генерації та відправки на сервер даних на метеорологічних датчиках (рис. 1), яка паралельно збирала показники з кожного метеорологічного датчика(рис. 2) та PostgreSQL, яка зберігала всі показники датчиків в момент їхньої генерації (рис. 3).

На рис. 1–3 зображені два основні показники навантаження на систему: навантаження на процесор та оперативну пам'ять. Найбільше навантаження на диск має сервіс PostgreSQL через зберігання даних на диску. Вказане навантаження на процесор є трохи завищеним через те, що ці дані відповідають дійсності тільки для контейнеру, котрий оперує лише ізольованими ресурсами. Загалом навантаження на ресурси комп'ютера всією системою не перебільшує-вало 2 Гб оперативної пам'яті та 51% навантаження на процесор. Три основні сервіси сумарно використовували не більше 450 Мб пам'яті.

Таким чином, в ході роботи було запропоновано обрати для аналізу на-ступний технологічний стек:

- безкоштовна об'єктно-реляційна система управління базами даних
- компільована, багатопотокова мова програмування Go;
- апаратна система віртуалізації для систем на базі x64-процесорів на базі гіпервізора Microsoft Hyper-V.

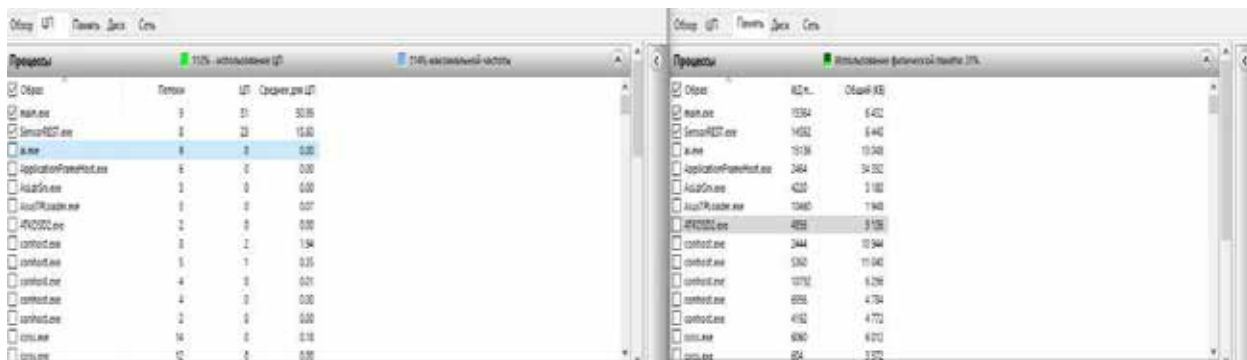


Рис. 1. Навантаження на програму генерації та відправлення даних

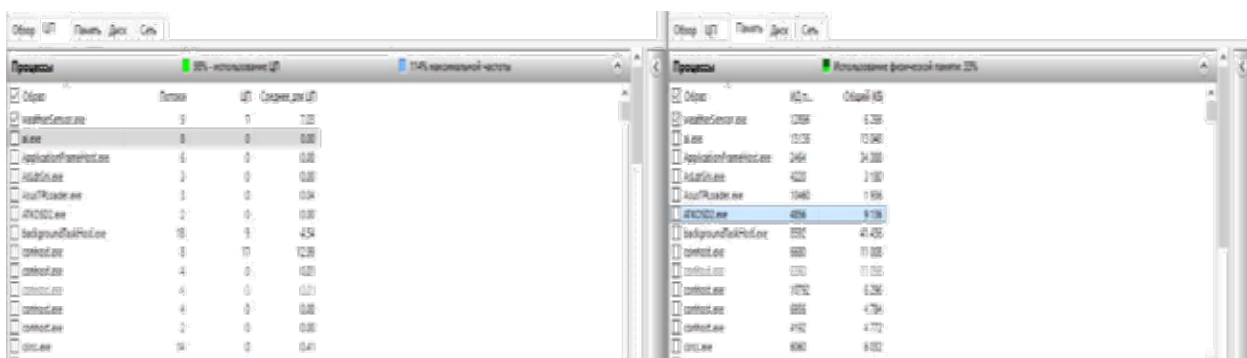


Рис. 2. Навантаження на процесор та ОЗУ REST API серверу

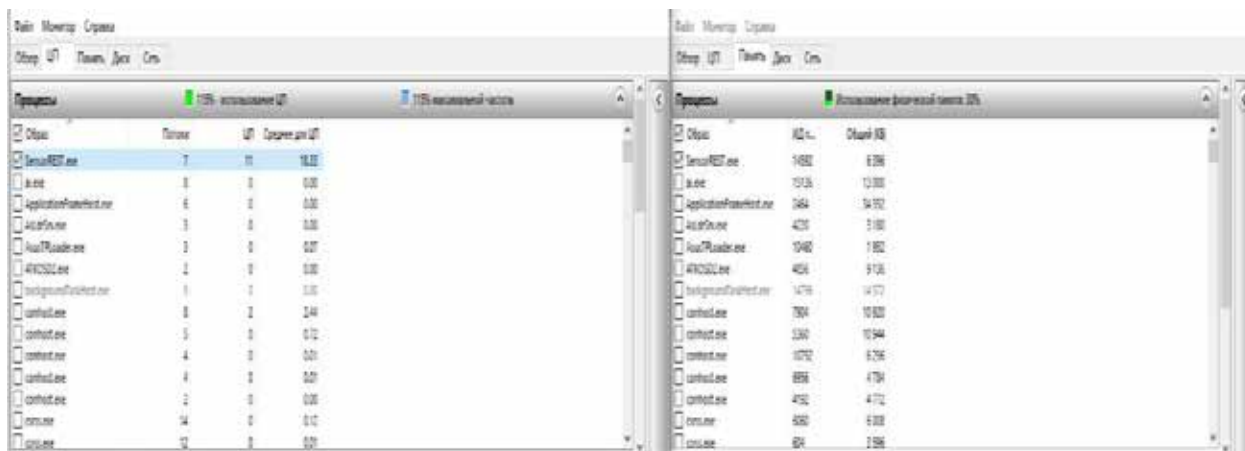


Рис. 3. Навантаження на сервер збору інформації та запису в PostgreSQL

Висновки

В результаті дослідження можна зробити висновок, що вибір технологічного стеку для розробки приватної хмарної системи паралельного збору та обробки інформації є важливим етапом в процесі розробки. Для досягнення ефективної та безперебійної роботи системи, необхідно враховувати різні аспекти, такі як масштабованість, продуктивність, безпеку та інші.

Проаналізувавши найбільш популярні системи управління реляційними базами даних з відкритим початковим кодом, а саме PostgreSQL та MySQL було визначено що для розробки приватної хмарної системи паралельного збору та обробки інформації найбільше підходить PostgreSQL.

Після того як було обрано СУБД, було проаналізовано мови програмування веб серверів, і спираючись на такі чинники як продуктивність, можливість працювати з багатьма потоками чи навіть ядрами процесора, споживання невеликих об'ємів оперативної пам'яті, а також простоти мови, було GoLang, для розробки веб серверу.

Отже, вибір технологічного стеку є важливим етапом в розробці приватної хмарної системи паралельного збору та обробки інформації, прорівнивши аналіз наявних мов СУБД, мов програмування та систем апаратної

віртуалізації було виявлено, що найбільш оптимальним технологічним стеком для розробки приватної хмарної системи паралельного збору та обробки інформації, являється наступний стек:

- PostgreSQL[12], як СУБД з відкритим кодом та гарними показниками продуктивності;
- Golang, як продуктивна та проста мова програмування веб серверів;
- Microsoft Hyper-V, як система апаратної віртуалізації на основі гіпервізора

Список використаної літератури

1. Офіційна сторінка Davis Instruments. URL: <https://www.davisinstruments.com/>
2. Офіційна сторінка Vaisala. URL: <https://www.vaisala.com>
3. Офіційна сторінка Campbell Scientific. URL: <https://www.campbellsci.com/>
4. Офіційна сторінка Golang. URL: <https://go.dev>
5. Офіційна сторінка PostgreSQL. URL: <https://www.postgresql.org/>
6. Офіційна сторінка MySQL URL: <https://www.mysql.com/>
7. What is a Relational Database? URL: https://aws.amazon.com/relational-database/?nc1=h_ls
8. PostgreSQL vs MySQL: The Critical Differences. URL: <https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/>
9. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. URL: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
10. Nodejs vs. Go – Which One Is the Right Choice for You? URL: <https://medium.com/javarevisited/nodejs-vs-go-which-one-is-the-right-choice-for-you-7e085a720f7e>
11. Go vs. Python in 2023: Which Language Should You Choose. URL: <https://uvik.net/blog/go-vs-python/>
12. 14 Best Programming Languages to Learn in 2023. URL: <https://hackr.io/blog/best-programming-languages-to-learn>

References

1. The official homepage of Davis Instruments. URL: <https://www.davisinstruments.com/>
2. The official homepage of Vaisala. URL: <https://www.vaisala.com>
3. The official homepage of Campbell Scientific. URL: <https://www.campbellsci.com/>
4. The official homepage of Golang. URL: <https://go.dev>
5. The official homepage of PostgreSQL. URL: <https://www.postgresql.org/>
6. The official homepage of MySQL URL: <https://www.mysql.com/>
7. What is a Relational Database? URL: https://aws.amazon.com/relational-database/?nc1=h_ls
8. PostgreSQL vs MySQL: The Critical Differences. URL: <https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/>
9. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. URL: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
10. Nodejs vs. Go – Which One Is the Right Choice for You? URL: <https://medium.com/javarevisited/nodejs-vs-go-which-one-is-the-right-choice-for-you-7e085a720f7e>
11. Go vs. Python in 2023: Which Language Should You Choose. URL: <https://uvik.net/blog/go-vs-python/>
12. 14 Best Programming Languages to Learn in 2023. URL: <https://hackr.io/blog/best-programming-languages-to-learn>