

ПОРІВНЯЛЬНИЙ АНАЛІЗ ШВИДКОДІІ ДЕЯКИХ РЕЛЯЦІЙНИХ БАЗ ДАНИХ ДЛЯ ЗАДАЧІ ВІДСЛІДКОВУВАННЯ

Львівський національний університет імені Івана Франка

Анотація

Проведено якісний аналіз пропускної здатності деяких широкоживаних реляційних баз даних (MySQL, PostgreSQL, MySQL) з метою можливого використання вищезгаданих баз в якості основного сховища даних (координат та ідентифікатора ресурсу) для програмного додатку відслідковування положення об'єкту у реальному часі.

Ключові слова: бази даних, пропускна здатність, відслідковування, MySQL, PostgreSQL, MySQL.

Abstract

A qualitative analysis of throughput is performed for the MySQL, PostgreSQL and MySQL relational databases with the aim to their possible usage as a main store for the data (coordinates and resource ID) needed for a tracking application.

Keywords: databases, throughput, tracking, MySQL, PostgreSQL, MySQL.

Вступ

Відслідковування об'єкту в режимі реального часу є невід'ємною частиною функціоналу широкої кількості програмних додатків сьогодення. З урахуванням постійного зростання пропускної здатності Інтернет-мережі та кількості об'єктів моніторингу питання вибору оптимальної бази даних для збереження інформації про об'єкти моніторингу набуває значної актуальності. За наявності обмеженого вихідного ресурсу ІТ-стартапів для побудови MVP-моделей вони використовують здебільше програмні пакети, ліцензовані як OpenSource, та бази даних, встановлені з налаштуванням по замовчуванню («out of the box»)[1].

У цій роботі нашим завданням було здійснити порівняльний аналіз пропускної здатності найбільш популярних реляційних бази даних (MySQL, PostgreSQL і MySQL), встановлених по замовчуванню, які використовуються в MVP-моделях програмних додатків для імплементації завдання відслідковування об'єкту в режимі реального часу.

Результати дослідження

В якості експерименту нами було розроблено програмний додаток з використанням мови програмування C# на платформі .NET і фреймворку Entity Framework. Його завданням є опрацювання HTTP-запитів (GET – отримати усі координати об'єкту, впорядковані за часом; PUT – додати нову координату в певний момент часу). При кожному HTTP-запиті програмний додаток здійснює виклик відповідної функції фреймворку Entity Framework, який, у свою чергу, формує та виконує запит до відповідної бази даних. Ідея експерименту полягала в такому: поступово навантажуючи базу даних даними (PUT-запитами) та не створюючи ніяких спеціальних індексів у базі даних, вимірювати пропускну здатність додатку (тобто, кількість HTTP GET-запитів, успішно опрацьованих додатком за 1 с). Усі дослідження здійснювалися на тому ж персональному комп'ютері з одним і тим же ресурсом. Це дало змогу провести якісний аналіз одержаних результатів [1].

У даній роботі ми не досліджували питання масштабування та надійності бази даних. Вимірювання проводилися за допомогою програмного пакету Apache Jmeter для дослідження швидкодії програм. Усі досліджуванні бази даних було встановлено з налаштуваннями згідно замовчування для відповідної версії [1]. База даних складалася з однієї таблиці (поля: ідентифікатор

ресурсу, довгота, широта, а також час визначення координати), яку заповнювали відповідно до порядку отримання HTTP PUT-запитів програмним додатком. Під час експерименту було досліджено реляційні бази даних MsSQL, PostgreSQL і MySQL

Залежність пропускної здатності програмного додатку від кількості координат, збережених у базі даних, зображена на рис 1.

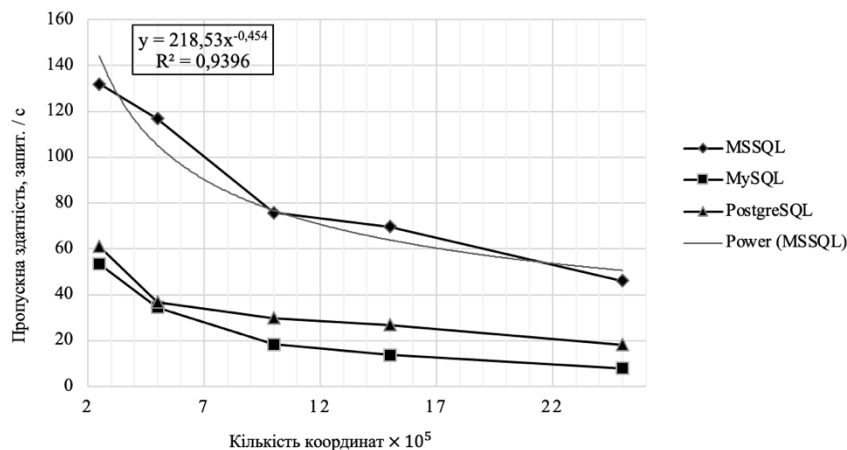


Рис. 1. Залежність пропускної здатності програмного додатку від кількості координат збережених у базі даних

Дослідження яскраво підтверджує загальновідому поведінку, притаманну всім базам даних: зниження швидкодії бази даних зі зростанням кількості даних, збережених у ній [2].

З рис. 1 випливає, що для програмних додатків, які використовують нами досліджені бази даних, спостерігаємо обернену степеневу залежність пропускної здатності від кількості координат [2]. Для випадку MsSQL коефіцієнт детермінації $R^2 = 0,939$. Зазначимо, що кількісні показники пропускної здатності для MsSQL є 2,2 більші, ніж для PostgreSQL та в 3 рази більші, порівняно з MySQL. Можна припустити, що такі високі показники для MsSQL зумовлені не тільки швидкістю бази даних а і оптимізацією, вбудованою у фреймворк Entity Framework для даної бази.

Висновки

Базуючись на отриманих вище результатах, ми встановили, що поєднання програмного додатку, розробленого з використанням мови C# та фреймворку Entity Framework, і бази даних MsSQL демонструє найвищі показники швидкодії, порівняно з іншими досліджуваними реляційними базами даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kruckenberg M. Pro MySQL / M. Kruckenberg, J. Pipes. – NY : Apress, 2005. – 583 p.
2. Zhang C. Benchmarking on intensive transaction processing / Zhang C., Li Y., Zhang R., Qian W., Zhou A. // Frontiers of Computer Science. – 2020. – Vol. 14, No. 2. – P. 145204. – Mode of access: <https://doi.org/10.1007/s11704-019-8438-0>

Білецький Олександр Русланович – студент групи ФeП-22, факультет електроніки та комп’ютерних технологій, Львівський національний університет ім. І. Франка, Львів, email: alexbiletskyi17@gmail.com

Франів Володимир Андрійович – канд. фіз-мат. наук, доцент кафедри оптоелектроніки та інформаційних технологій, Львівський національний університет імені Івана Франка, Львів, email: franivva@gmail.com

Biletskyi Oleksandr Ruslanovych – student, Electronics and Computer Technologies Department, Ivan Franko National University of Lviv, Lviv, Ukraine, email: alexbiletskyi17@gmail.com

Franiv Volodymyr Andriyovych – PhD (Eng.), Docent, Electronics and Computer Technologies Department, Ivan Franko National University of Lviv, Lviv, Ukraine, email: franivva@gmail.com