

Я.Ю. Згуровський, Є.С. Сулема, І.О. Саяпіна

## **ПРОЄКТУВАННЯ АРХІТЕКТУРИ ДЛЯ ЗБЕРІГАННЯ ЧАСОВОЇ ЗВ'ЯЗНОЇ МОДЕЛІ ЦИФРОВОГО ДВІЙНИКА**

*Анотація. У зв'язку з розвитком інформатизації усіх сфер діяльності широкого поширення набуває використання технології цифрового двійника. Дана технологія дозволяє замінити фізичні та аналогові ресурси цифровими. Однак вона потребує зберігання та обробки великої кількості даних та інформації у режимі реального часу що ставить додаткові вимоги до пошуку відповідного сховища даних, яке їх задовольнятиме. На основі концепцій бази даних TimescaleDB було спроєктовано трирівневу архітектуру для зберігання та агрегації мультимодальних даних цифрового двійника. У роботі пропонується адаптувати часову зв'язну модель для відображення цифрового двійника використовуючи концепцію зберігання часових даних у TimescaleDB з розподіленими гіпертаблицями. Для цього у якості ключа для розподілу простору пропонується обрати унікальний ідентифікатор модальності даних. Таким чином розподіл темпоральних мультимодальних даних забезпечується власне самою базою даних TimescaleDB. Отже вибірка даних, згрупована за часом та модальністю, працюватиме швидше, оскільки навантаження на пошук розподіляється в паралелі між кількома вузлами даних. За рахунок використання та адаптації концепції розподілених гіпертаблиць TimescaleDB запропонований спосіб дозволяє досягти ефективного зберігання та відтворення часової зв'язної моделі.*

*Ключові слова: цифровий двійник, мультимодальні дані, часова модель цифрового двійника, бази даних для часових рядів, TimescaleDB, архітектура.*

**Постановка проблеми.** Останнім часом у зв'язку з розвитком інформаційних технологій широкого розповсюдження набуває технологія цифрового двійника, що дозволяє замінити фізичні та аналогові ресурси цифровими. Цифровий двійник являє собою модель фізичного об'єкта або процесу (фізичного двійника), яка якнайповніше відображає його характеристики у динаміці протягом певного періоду часу. Він може представляти продуктивність, операції, навколишнє середовище, геометрію товару та стану ресурсів на основі даних, що постійно збираються, оновлень від фізичного аналога. Цифровий двійник також є зростаючим та ефективним методом взаємодії в реальному часі. Однак, маючи величезну кількість даних та інформації в режимі реального часу

від цифрового двійника, важко представити їх користувачам та операторам інтуїтивно [1, 2].

**Аналіз останніх досліджень і публікацій.** Цифровий двійник для дослідження довільного об'єкта формально можна подати у вигляді математичних абстракцій, що були розроблені для ефективної обробки темпоральних мультимодальних даних досліджуваного об'єкта. На основі вищезгаданих абстракцій та їх застосування для формальної специфікації цифрового двійника будеться так званий мультиобраз, що фактично надає формальний опис послідовностей мультимодальних даних про досліджуваний об'єкт, які були отримані з урахуванням часу в результаті вимірювання, генерації та інших операцій, що пов'язані з отриманням даних.

Як результат застосування методу створення мультиобразу та методу синхронізації мультиобразів отримуються синхронізовані темпоральні мультимодальні дані. Вони є основою для побудови будь-якої моделі цифрового двійника досліджуваного об'єкта.

В залежності від мети дослідження фізичного двійника, отримані дані можуть розглядатися на різних рівнях абстракції.

Розглянемо відображення фізичного двійника, як об'єкту, що визначається множиною станів, що динамічно змінюються з плином часу, так звану часову зв'язну модель.

Часова зв'язна модель представляє об'єкт дослідження через набір дискретних станів, визначених у конкретний момент часу набором ознак об'єкта, які є мультимодальними даними.

В результаті дослідження відбувається формування послідовності дискретних станів досліджуваного об'єкта, кожен з яких можна характеризувати сукупністю темпоральних мультимодальних даних, що асоційовані з конкретним моментом часу, що власне і слугує часовим ключем визначеного стану об'єкта [3].

Для ефективного відтворення у подальшому часової зв'язної моделі важливим фактором є вибір бази даних, яка могла би оперативно зберігати та обробляти великі обсяги даних, що надходять від досліджуваного об'єкта. Одним з рішень для зберігання таких даних є використання нереляційних NoSQL баз даних через можливість їх масштабування, оскільки класичні реляційні бази даних віддають перевагу узгодженості та підтримці принципів ACID, жертвуючи доступністю та продуктивністю, що ускладнює їх горизонтальне масштабування [4]. Але до недоліків NoSQL баз даних можна віднести відсут-

ність підтримки SQL – потужного засобу взаємодії користувача з базою даних та необхідність додаткового навчання персоналу для обслуговування певної NoSQL системи управління базами даних (СУБД). Оптимальним рішенням може стати використання баз даних часових рядів (Time Series Databases), прикладом якої є TimescaleDB. Появі цього окремого класу баз даних посприяли особливості даних часових рядів, серед яких можна виділити індексування даних відповідними часовими мітками, постійне збільшення обсягів, агрегування та здійснення вибірки у діапазонах, вимогливість до запису [5].

За результатами тестування TimescaleDB та реляційної бази даних PostgreSQL при записі та читанні даних, що передають стан датчиків і виконавчих механізмів для системи WinCC OA SCADA, швидкість обробки TimescaleDB становить 80000 рядків за секунду для одного вузла і 150000 рядків за секунду для двох вузлів, що у 2 та 3 рази вище, ніж показники PostgreSQL [6]. Результати тестування і порівняння TimescaleDB з такими NoSQL СУБД, як Redis, MongoDB, HBase, Cassandra и Riak TS при тестуванні чотирьох сценаріїв (імпорт даних, вибірка інформації про станцію відповідно до часового поясу, індексний запит та збільшення кількості запитів при вставці даних до баз даних) при зберіганні та аналізі масивних метеорологічних даних часових рядів показали, що TimescaleDB показує кращу продуктивність, ніж Redis, MongoDB, HBase, Cassandra та Riak TS, однак завантаження центрального процесора у TimescaleDB у конкретному сценарії може бути вище, ніж у MongoDB [7].

**Мета дослідження.** Завдання дослідження полягає у проектуванні архітектури програмної системи для зберігання та подальшого відтворення часової зв'язної моделі, що використовує базу даних, призначену для зберігання даних часових рядів.

**Викладення основного матеріалу дослідження.** У якості основного двигуна для зберігання даних пропонується розглянути TimescaleDB. TimescaleDB — це реляційна база даних для даних часових рядів. Вона реалізована як розширення для PostgreSQL, це означає, що вона фактично працює на сервері PostgreSQL як частина того самого процесу, з кодом, який вводить нові можливості для керування даними часових рядів, нові функції для аналізу даних, новий планувальник запитів і оптимізацію виконання запитів та нові механізми зберігання для ефективної аналітики [8]. Будь-які операції з базою даних PostgreSQL, що включає TimescaleDB (чи то SELECT, чи INSERT, чи керування схемою, як-от створення індексів), спочатку обробляються TimescaleDB, щоб визначити, як вони мають плануватися чи виконуватися зі структурами

даних TimescaleDB. Ця модель розширення дозволяє базі даних використовувати всі переваги PostgreSQL включаючи понад 40 типів даних (цілих чисел, плаваючих чисел, рядків, часових позначок, масивів і типів даних JSON), різні типи індексів, складні схеми бази даних, розширений планувальник запитів, а також більшу екосистему розширень, яка добре працює з TimescaleDB, що включає геопросторову підтримку через PostGIS, моніторинг за допомогою `pg_stat_statements`, обгортки іноземних даних тощо.

Основні виклики для бази даних, яка повинна опрацьовувати велику кількість часових даних, полягає у її масштабованості – здатності обробляти та зберігати велику кількість даних без втрат у продуктивності. Розглянемо дві основні концепції для масштабування, що використовує TimescaleDB: абстракції даних у вигляді гіпертаблиць (hypertables) і фрагментів (chunks) (як вони зберігаються та обробляються) і як TimescaleDB можна розгорнути на одному сервері з фізичними репліками або як кластер серверів із багатьма вузлами, щоб використати концепцію розподілених гіпертаблиць (distributed hypertables).

*Гіпертаблиці та фрагменти.* Гіпертаблиці та фрагменти дозволяють значно швидше зберігати та запитувати дані часових рядів. TimescaleDB автоматично розбиває дані часового ряду на частини або підтаблиці на основі часового ключа, або інших вказаних полів. Також можна налаштувати розмір фрагмента так, щоб останні фрагменти містилися в пам'яті для швидших запитів. Гіпертаблиця — це своєрідний шар абстракції над фрагментами, які містять власне дані часового ряду (рис. 1). Гіпертаблиці дають змогу запитувати та отримувати доступ до даних з усіх фрагментів гіпертаблиці. Таким чином дана концепція надає переваги автоматичного розподілу даних часових рядів, а також простоту роботи, оскільки з огляду на інтерфейс гіпертаблиця виглядає як стандартна єдина таблиця PostgreSQL [9].

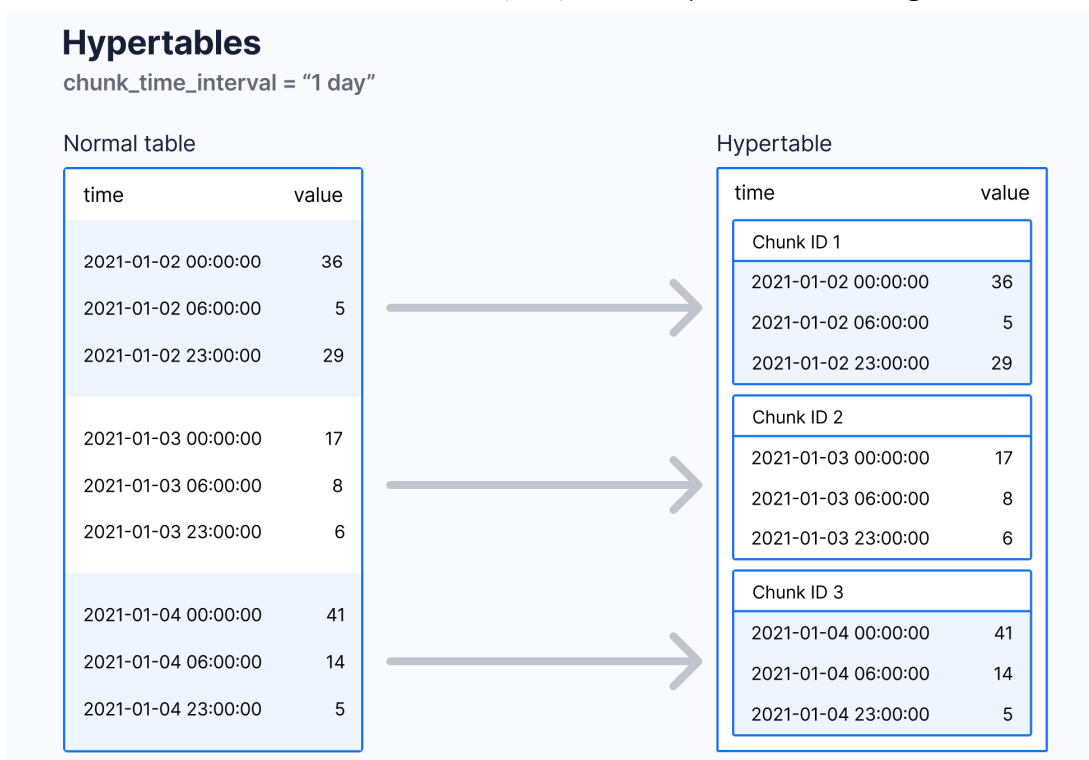


Рисунок 1 – Візуалізація гіпертаблиці та фрагментів

*Поділ на фрагменти.* Фрагменти створюються шляхом поділу даних гіпертаблиці на один (або потенційно кілька) вимірів. Усі гіпертаблиці розділені значеннями, що належать до стовпця часу, який може бути у мітці часу, даті або в різних цілочисельних формах. Наприклад, якщо інтервал розподілу часу становить один день, то рядки з мітками часу, які належать до одного дня, розміщуються в одному фрагменті, тоді як рядки, що належать до інших днів, належать до різних фрагментів.

TimescaleDB створює ці фрагменти автоматично, коли рядки вставляються в базу даних. Якщо мітка часу нововставленого рядка належить до дня, якого ще немає в базі даних, TimescaleDB створює новий фрагмент, що відповідає цьому дню, як частину процесу вставки. В іншому випадку TimescaleDB визначає існуючий блок, до якого належать нові рядки, і вставляє їх у відповідні блоки.

Інтервал поділу гіпертаблиці також може змінюватися з часом, щоб адаптуватися до змінних умов робочого навантаження. Гіпертаблиця може спочатку створювати новий фрагмент на день, а потім змінювати фрагмент наприклад для кожних 6 годин у міру збільшення робочого навантаження.

Гіпертаблиця також може бути розділена додатковими стовпцями, наприклад, ідентифікатором пристрою, ідентифікатором користувача тощо. Таке ро-

зділення на цьому додатковому стовпці зазвичай використовує хешування. Такі гіпертаблиці, розділені як за часом, так і за цим додатковим виміром, називають як розділені за «часом і простором».

*Розподілені гіпертаблиці.* Розподілена гіпертаблиця існує для розподіленої бази даних, яка складається з кількох баз даних, що зберігаються на одному або кількох екземплярах TimescaleDB. База даних, яка є частиною розподіленої бази даних, може виконувати роль або вузла доступу (access node - AN), або вузла даних (data node - DN), але не обох (рис. 2).

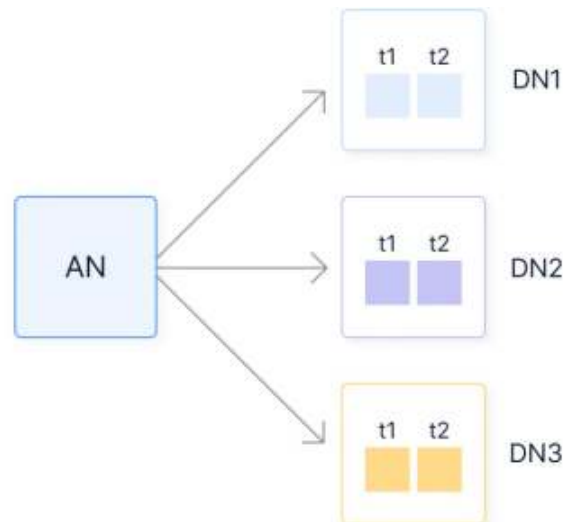


Рисунок 2 – Розподілена гіпертаблиця

Клієнт підключається до бази даних через вузол доступу. Потім вузол доступу розподіляє запити відповідним чином на вузли даних і об'єднує результати, отримані від вузлів даних. Вузли доступу зберігають всю кластерну інформацію про різні вузли даних, а також про те, як фрагменти розподіляються між цими вузлами даних. Вузли доступу також можуть зберігати нерозподілені гіпертаблиці, а також звичайні таблиці PostgreSQL [10].

Вузли даних не зберігають інформацію для всього кластера, і з точки зору моделювання виглядають так, ніби вони є окремими екземплярами TimescaleDB. Прямий доступ до гіпертаблиць або фрагментів на вузлах даних обмежено, оскільки це може призвести до неузгоджених розподілених гіпертаблиць.

Щоб забезпечити найкращу продуктивність, необхідно розділити розподілену гіпертаблицю за часом і простором. Якщо розділяти дані лише за часом, фрагмент першого вузла даних має повністю заповнитися, перш ніж вузол доступу вибере інший вузол даних для зберігання наступного фрагмента, тому протягом інтервалу часу цього фрагмента всі записи до останнього інтервалу

обробляються лише одним вузлом даних, на противагу тому коли навантаження збалансовано для всіх доступних вузлів даних. З іншого боку, якщо вказати розподіл простору, вузол доступу розподіляє фрагменти між кількома вузлами даних на основі розподілу простору, так що для даного інтервалу часу створюється кілька фрагментів, і як зчитування, так і запис у цей останній інтервал часу будуть балансування навантаження по кластеру.

*Адаптація часової моделі.* З огляду на концепцію зберігання часових даних у TimescaleDB з використанням розподілених гіпертаблиць пропонується адаптувати часову зв'язну модель для відображення цифрового двійника використовуючи вищезгадану концепцію. Для цього у якості ключа для розподілу простору пропонується обрати унікальний ідентифікатор модальності даних.

Таким чином розподіл темпоральних мультимодальних даних забезпечується власне самою базою даних TimescaleDB. Отже вибірка даних, згрупована за часом та модальністю працюватиме швидше, оскільки навантаження на пошук розподіляється в паралелі між кількома вузлами даних.

Для зберігання даних кожної модальності окремо існує можливість створити звичайну PostgreSQL таблицю, що матиме схему, адаптовану під кожну модальність (рис. 3).

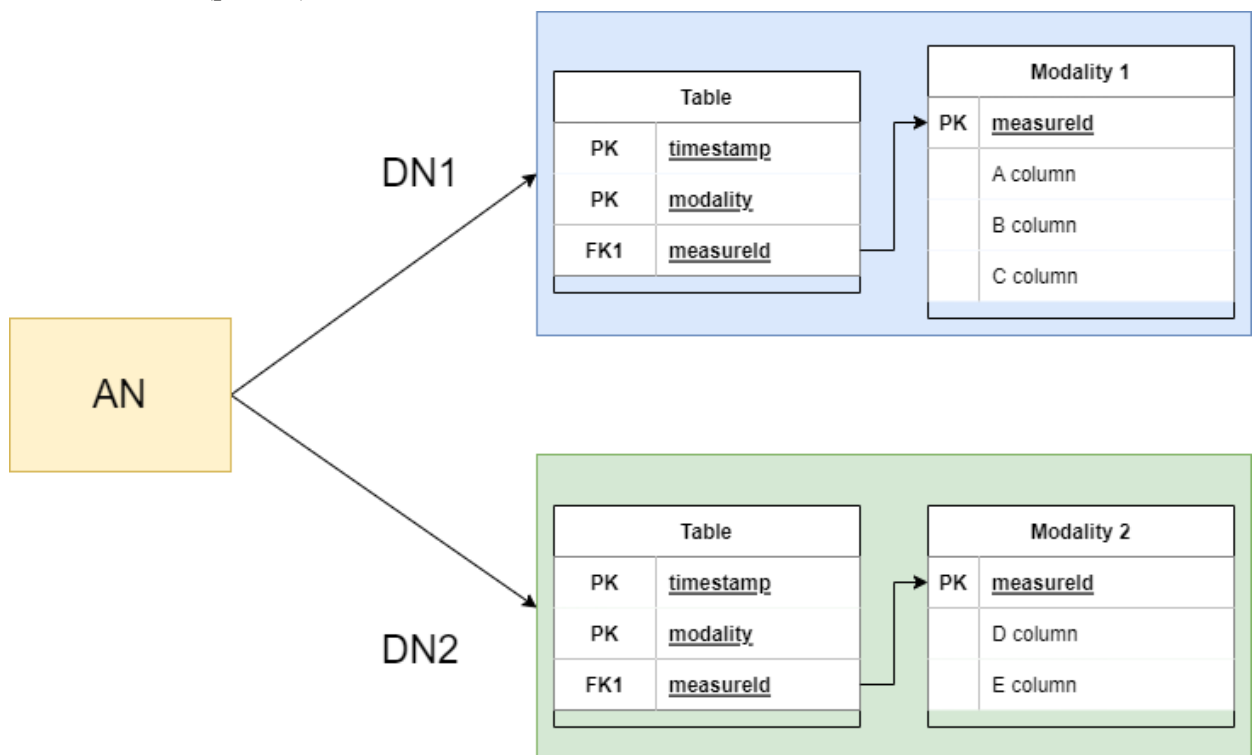


Рисунок 3 – Схема даних для розподіленої гіпертаблиці

Для програмної системи пропонується проектування трирівневої архітектури для забезпечення розподілення обов'язків у системі з наступними рівнями: рівень прийому даних, рівень зберігання та рівень обробки запитів до даних (рис. 4). Кожен рівень відповідно може незалежно масштабуватися. Така конструкція дозволить спростити кожен підсистему, розділивши відповідні обов'язки між ними, усунувши вузькі місця щодо масштабованості та знизити ймовірність виникнення пов'язаних збоїв.

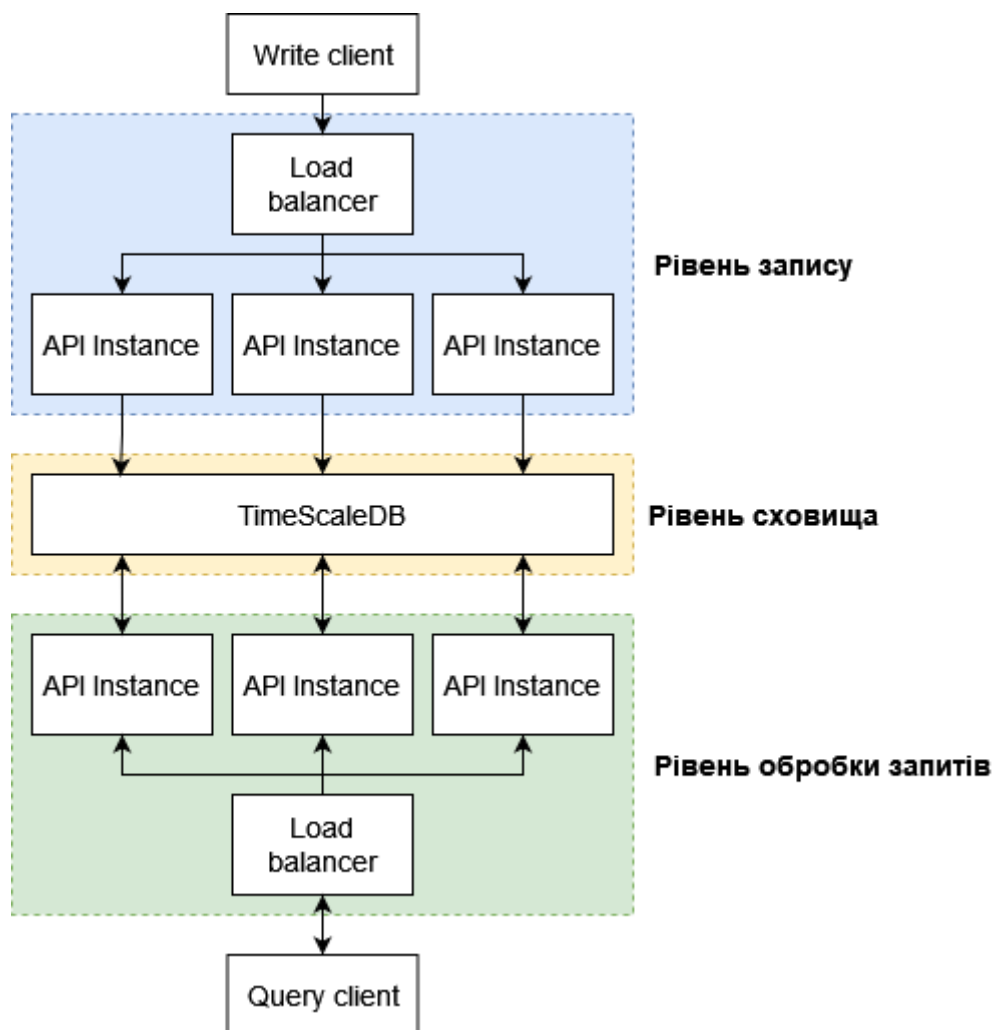


Рисунок 4 – Трирівнева архітектура

*Рівень запису.* На рівні запису пропонується розгорнути API сервіси, що будуть виконувати запити на створення рядків: попередньо валідувати та обробляти дані для подальшого запису на рівні сховища.

Під час запису даних сервіс має відповідним чином розділити запис, якщо він містить кілька модальностей, на кілька записів для кожної з таблиць модальностей та безпосередньо виконати запит на рівень сховища. Далі ці записи



будуть автоматично розподілені між вузлами даних, за індексами показника часу та значення модальності, відповідно до концепції гіпертаблиць.

*Рівень сховища.* На рівні сховища буде розгорнуто фактично TimescaleDB, яка дозволить ефективно зберігати дані та використовувати її потужності для безпосереднього запису та зчитування відповідно до концепції розподілених гіпертаблиць.

*Рівень обробки запитів.* На рівні обробки запитів буде розгорнуто API сервіси, які мають валідувати, агрегувати та виконувати відповідні отримані запити використовуючи рівень сховища.

**Висновки.** Запропонований спосіб для побудови часової зв'язної моделі цифрового двійника на основі TimescaleDB з використанням концепції гіпертаблиць дозволить досягти ефективного використання ресурсів за рахунок паралелізації навантаження під час пошуку даних, а трирівнева архітектура, спроектована для розподілення обов'язків програмної системи, здатна значно спростити імплементацію та подальшу обробку часової моделі цифрового двійника.

#### ЛІТЕРАТУРА

1. Cimino C. Review of digital twin applications in manufacturing / C. Cimino, E. Negri, L. Fumagalli // Computers in Industry. – 2019. – Т. 113. – С. 103-130.
2. Liu M. et al. Review of digital twin about concepts, technologies, and industrial applications/ M. Liu // Journal of Manufacturing Systems. – 2020.
3. Сулема, Є. С. Методи, моделі та засоби обробки мультимодальних даних цифрових двійників досліджуваних об'єктів : дис. ... д-ра техн. наук : 01.05.03 – математичне та програмне забезпечення обчислювальних машин і систем / Сулема Євгенія Станіславівна. – Київ, 2020. – 343 с.
4. B. Shah. Performance Study of Time Series Databases / B. Shah, P. M. Jat, K. Sashidhar // Computer Science and Information Technology Trends. – 2022. Crossref. DOI: <https://doi.org/10.5121/csit.2022.121406>.
5. J. Mostafa. SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things / J. Mostafa, S. Wehbi, S. Chilingaryan, and A. Kopmann // Proceedings of the 34th International Conference on Scientific and Statistical Database Management (SSDBM '22). Association for Computing Machinery. – New York, 2022. – Стаття 12. – С. 1-11. DOI: <https://doi.org/10.1145/3538712.3538723>.

6. M. Kammakomati. Performance Evaluation of TimescaleDB for Storage of Historical Data from WinCC OA SCADA Systems / M. Kammakomati // Zenodo. – 2021. DOI: <https://doi.org/10.5281/zenodo.5541641>
7. Shen, L., Lou, Y., Chen, Y., Lu, M., Ye, F. Meteorological Sensor Data Storage Mechanism Based on TimescaleDB and Kafka / L. Shen, Y. Lou, Y. Chen, M. Lu, F. Ye // Data Science. ICPCSEE 2019. Communications in Computer and Information Science. – Singapore, Springer, 2019. – Вип. 1058 – С. 137-147. DOI: [https://doi.org/10.1007/978-981-15-0118-0\\_11](https://doi.org/10.1007/978-981-15-0118-0_11)
8. Freedman M. TimescaleDB vs. InfluxDB: Purpose Built Differently for Time-Series Data [Електронний ресурс] / M. Freedman, A. Sewrathan. – 2020. – Режим доступу до ресурсу: <https://www.timescale.com/blog/timescaledb-vs-influxdb-for-time-series-data-timescale-influx-sql-nosql-36489299877/>.
9. Hypertables and chunks [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://docs.timescale.com/timescaledb/latest/overview/core-concepts/hypertables-and-chunks/#hypertables>.
10. About distributed hypertables [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://docs.timescale.com/timescaledb/latest/how-to-guides/distributed-hypertables/about-distributed-hypertables/#about-distributed-hypertables>.

## **REFERENCES**

1. Cimino C., Negri E., Fumagalli L. Review of digital twin applications in manufacturing //Computers in Industry. – 2019. – V. 113. – P. 103130.
2. Liu M. et al. Review of digital twin about concepts, technologies, and industrial applications //Journal of Manufacturing Systems. – 2020.
3. Sulema Ye. S. Methods, models, and means of researched objects digital twins multimodal data processing: The thesis for the degree of Doctor of technical sciences, specialty 01.05.03 – Mathematical Support and Software of Computers and Systems / Sulema Yevgeniya Stanislavivna. – Kyiv, 2020. – 343 p.
4. B. Shah, P. M. Jat, K. Sashidhar. Performance Study of Time Series Databases // Computer Science and Information Technology Trends. – 2022. Crossref. DOI: <https://doi.org/10.5121/csit.2022.121406>.
5. J. Mostafa, S. Wehbi, S. Chilingaryan, and A. Kopmann. SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things // Proceedings of the 34th International Conference on Scientific and Statistical Data-

base Management (SSDBM '22). Association for Computing Machinery. – New York, 2022. – Article 12. – P. 1–11. DOI: <https://doi.org/10.1145/3538712.3538723>.

6. M. Kammakomati. Performance Evaluation of TimescaleDB for Storage of Historical Data from WinCC OA SCADA Systems // Zenodo. – 2021.

DOI: <https://doi.org/10.5281/zenodo.5541641>

7. Shen, L., Lou, Y., Chen, Y., Lu, M., Ye, F.. Meteorological Sensor Data Storage Mechanism Based on TimescaleDB and Kafka // Data Science. ICPCSEE 2019. Communications in Computer and Information Science. – Singapore, Springer, 2019. – Vol. 1058 – P. 137-147. DOI: [https://doi.org/10.1007/978-981-15-0118-0\\_11](https://doi.org/10.1007/978-981-15-0118-0_11)

8. Freedman M. TimescaleDB vs. InfluxDB: Purpose Built Differently for Time-Series Data [Електронний ресурс] / M. Freedman, A. Sewrathan. – 2020. – Режим доступу до ресурсу: <https://www.timescale.com/blog/timescaledb-vs-influxdb-for-time-series-data-timescale-influx-sql-nosql-36489299877/>.

9. Hypertables and chunks [Електронний ресурс]. – 2021. – Режим доступу до ресурсу:

<https://docs.timescale.com/timescaledb/latest/overview/core-concepts/hypertables-and-chunks/#hypertables>.

10. About distributed hypertables [Електронний ресурс]. – 2021. – Режим доступу до ресурсу:

<https://docs.timescale.com/timescaledb/latest/how-to-guides/distributed-hypertables/about-distributed-hypertables/#about-distributed-hypertables>.

Received 03.05.2022.

Accepted 06.05.2022.

### ***Design of an architecture for storing a time connected model of a digital twin***

*Recently, in connection with the development of information technology, digital twin technology has become widespread, which allows replacing physical and analog resources with digital ones. The digital twin is also a relevant and effective method of real-time interaction. However, with a wealth of real-time data and information from the digital twin, it is difficult to present them intuitively to users and operators.*

*A digital twin for the study of an arbitrary object can formally be represented as mathematical abstractions designed for efficient processing of temporal multimodal data of the object under study. Based on the above abstractions and their application for the formal specification of the digital twin, the so-called multiimage is built, which actually provides a formal description of the sequences of multimodal data about the object under study, obtained taking into account time as a result of measurement, generation and other related data acquisition operations. As a result of the study, a sequence of discrete*

*states of the object under study is formed, each of which can be characterized by a set of temporal multimodal data associated with a specific moment in time, which actually serves as a temporal key for a certain state of the object.*

*The purpose of the study is to design the architecture of a software system for storing and then reproducing a temporal connected model using a database designed to store time series data. TimescaleDB, a relational database for time series data, is proposed as the main data storage engine. The main challenges for a database that must process a large amount of temporary data is its scalability - to process and store a large amount of data without performance degradation.*

*The proposed method for building a temporal connected model of a digital twin based on TimescaleDB using the concept of hypertables will achieve efficient use of resources by parallelizing the load when searching for data, and a three-level architecture designed to distribute the responsibilities of a software system can greatly simplify the implementation and subsequent processing of a temporal model of a digital twin.*

**Згуровський Ярослав Юрійович** – магістрант кафедри програмного забезпечення комп’ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

**Сулема Євгенія Станіславівна** – д.т.н., доцент, завідувач кафедри програмного забезпечення комп’ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

**Саяпіна Інна Олександрівна** – к.т.н, доцент, доцент кафедри програмного забезпечення комп’ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

**Zgurovsky Yaroslav Yuriyovych** - Master's student of the Computer Systems Software Department, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

**Sulema Yevgeniya Stanislavivna** – Doctor of Technical Sciences, Associate Professor, Head of the Computer Systems Software Department, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

**Saiapina Inna Oleksandrivna** - Ph.D., Associate Professor, Associate Professor of the Computer Systems Software Department, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.