

УДК 4.65

Застосування графових баз даних для моделювання соціальних графів

В.М. Лазурик, Є.С. Тимошенко

Харківський національний університет імені В. Н. Каразіна, майдан Свободи 4, м. Харків, 61022, Україна
e-mail: lazurik@hotmail.com

Стаття присвячена графовим системам керування базами даних. Розглянуті основні характеристики та можливості цих систем. Серед усієї множини завдань, які найкращим чином вирішуються з використанням графової моделі даних, обрані завдання, що виникають при розробці соціальних мереж. Щодо виділеного класу завдань, для дослідження обрані системи керування графовими базами даних, що найпопулярніші на теперішній час – Neo4J, OrientDB та ArangoDB. Приділено увагу таким характеристикам обраних баз даних, як проприетарне це програмне забезпечення чи вільне, чи має воно сучасну документацію, чи підтримується розробниками, чи є спільнота, де можна одержати відповіді на питання, чи багато часу треба витратити, щоб освоїти цю систему. З використанням мов Cypher, OrientDB SQL та AQL, що є мовами запитів у Neo4J, OrientDB та ArangoDB відповідно, сформовані запити, характерні щодо соціальних мереж, коли треба швидко одержувати результати з великою глибиною пошуку. Реалізоване порівняння швидкості виконання запитів. Для цього з використанням моделі Барабаші-Альберта згенеровано граф, який має 5000 вузлів та 24900 зав'язків. Сформовані тестові завдання для пошуку друзів трьох користувачів з глибиною, яка дорівнює 5. Проведене оцінювання середнього часу для кожного запиту за декілька виконань. Зроблені висновки та сформовані рекомендації щодо найкращого вибору графової бази даних при розробці програмного забезпечення, що реалізує соціальну мережу.

Ключові слова: NoSQL, граф, мультимодельні бази даних, графові бази даних, Neo4J, OrientDB, ArangoDB, мови запитів, Cypher, OrientDB SQL, AQL, час виконання запиту.

Статья посвящена графовым системам управления базами данных. Рассмотрены их основные характеристики и возможности. Среди всего множества задач, которые наилучшим образом решаются с использованием графовой модели данных, выбраны те, что которые могут возникнуть при разработке социальных сетей. Для выделенного класса задач в исследовании выбраны системы управления графовыми базами данных, наиболее популярные сегодня – Neo4J, OrientDB и ArangoDB. Уделено внимание таким характеристикам выбранных баз данных как проприетарное это программное обеспечение или свободно распространяемое, имеет ли оно современную документацию, поддерживается ли разработчиками, существует ли сообщество, где можно получить ответы на возникшие вопросы, и много ли времени необходимо затратить для того, чтобы освоить эту систему. С использованием языков запросов Cypher, OrientDB SQL и AQL, которые используются соответственно в Neo4J, OrientDB и ArangoDB, сформированы запросы, характерные для социальных сетей, когда необходимо быстро получать результаты с большой глубиной поиска. Реализовано сравнение скорости выполнения запросов в этих базах данных. Для этого с использованием модели Барабаша-Альберта сгенерирован граф, который имеет 5000 узлов и 24900 связей. Сформированы тестовые задания для поиска друзей трех пользователей с глубиной равной 5. Проведена оценка среднего времени для каждого запроса за несколько выполнений. Сделаны выводы и сформированы рекомендации относительно наилучшего выбора графовой базы данных при разработке программного обеспечения для реализации социальной сети.

Ключевые слова: NoSQL, граф, мультимодельные базы данных, графовые базы данных, Neo4J, OrientDB, ArangoDB, языки запросов, Cypher, OrientDB SQL, AQL, время выполнения запроса.

This article is devoted to graph database management systems. The main characteristics and capabilities of those systems have been contemplated. The problems that may occur during the social network development have been selected to be solved using a graph data model. The most popular database management systems nowadays, namely, Neo4J, OrientDB and ArangoDB have been chosen for the study. Such characteristics of the selected databases as whether the software is proprietary or freely distributed, whether databases have up-to-date documentation or not, whether they are supported by developers, whether there is a community where you can get answers to your questions, and how much time is needed to master the database have been elaborated. The typical social network queries, when you need to receive results with a large depth of search quickly, have been developed using the query languages Cypher, OrientDB SQL and AQL used in Neo4J, OrientDB and ArangoDB respectively. The comparison of query execution speed has been performed for the selected databases. For this purpose, a graph that has 5000 nodes and 24900 connections has been built by implementing the Barabashi-Albert model for generating random-scale networks. The test tasks for finding friends of three users with the depth of 5 have been generated. The average time for each request has been estimated for several executions. The conclusions have been drawn and the recommendations regarding the selection of the best graph database for social network implementation have been made. The results presented in the article can be useful for teachers in preparing lectures, as well as for students in studying databases.

Keywords: NoSQL, graph, multimodel databases, graph databases, Neo4J, OrientDB, ArangoDB, query languages, Cypher, OrientDB SQL, AQL, query execution time.

1 Вступ

У наш час кількість електронних даних зростає щосекунди. Такі цифрові гіганти, як Facebook чи Instagram щодня мають близько 500 мільйонів активних користувачів, які додають у мережу тисячі гігабайт даних. Цивілізоване людство вже не уявляє купівлю товарів без попереднього

вивчення рекомендацій чи відгуків інших користувачів Інтернету. Реляційні бази даних [1] понад двадцять років були успішною технологією, що забезпечувала персистентність, управління паралельністю і механізм інтеграції. Необхідність обробляти більші обсяги даних у великих системах обумовила появу NoSQL баз даних (БД) [2], що забезпечують перехід від вертикального масштабування до горизонтального масштабування на кластерах. Основними характеристиками NoSQL БД є відмова від використання реляційної моделі, ефективна робота на кластерах, відкритий вихідний код, врахування можливостей мережі Інтернет, відсутність структури даних [3].

Серед усіх NoSQL-моделей найбільший інтерес сьогодні представляє графова модель даних. Це пов'язано з тим, що графова модель сама по собі є найбільш природним підходом до моделювання, а також зі зростанням зв'язності даних [4]. Якщо планується розробка проекту з тих напрямків, де застосування графових систем керування базами даних (СКБД) є найбільш ефективним, на перший план виходить завдання правильного вибору серед існуючих популярних представників цих СКБД. Стаття присвячена розгляду та порівнянню на прикладі соціального графа популярних представників графових БД – Neo4J, OrientDB та ArangoDB.

2 Графові бази даних

Аналітичне агентство Gartner в 2018 році в своєму «Hype Cycle for Emerging Technologies» вказало на «Knowledge Graphs», як на висхідний тренд, і в 2019 році в «Technology Trends for 2019» [5] ці тенденції також зберігають своє місце. Графові бази даних (ГБД) [6] стоять трошки окремо від більшості NoSQL баз даних. Деякі з них забезпечують використання транзакцій – підтримують ACID, на відміну від інших представників NoSQL технології. Графові бази даних (ГБД) дозволяють зберігати суті і відносини між ними. Суті моделюються вузлами, які мають властивості. Вузол інтерпретується як екземпляр об'єкта в додатку. Відносини моделюються ребрами, які можуть мати властивості. Ребра мають напрямок; вузли організовані відповідно до відносин. Це дозволяє знаходити необхідні шаблони серед вузлів. Така організація графа дозволяє один раз записати дані, а потім інтерпретувати їх різними методами відповідно до відносин. Графові БД найкраще підходять для реалізації проектів, які передбачають природну графову структуру даних, в першу чергу соціальних мереж, а так само для створення семантичного павутиння. У подібних завданнях вони сильно випереджають реляційні БД по продуктивності, простоті внесення змін і наочності подання інформації.

Зараз багато систем керування базами даних в доповнення до основної моделі зберігання даних забезпечують своїх користувачів також іншими моделями. Такі системи називають мультимодельними. Наприклад, Oracle крім реляційної моделі дає можливість працювати з документами, застосовувати графи та RDF збереження даних. MS SQL сервер, що на початку теж підтримував тільки реляційну модель, зараз забезпечує своїх користувачів також графовими та документними можливостями. MariaDB також підтримує графові бази даних. За рейтингом DB-Engines [7], який проводить Solid IT щомісяця, на вересень 2019 р. найбільш популярними з мультимодельних СКБД (перша та третя сходинки) вважається Oracle та MS SQL сервер, MariaDB займає 14 місце, але її популярність зростає. Якщо передбачається застосування графової БД для реалізації проекту, то наведені мультимодельні СКБД можуть бути використані. Але існують СКБД, що мають підтримку графової моделі, як основної, а не допоміжної. Найбільш відома з них це Neo4j, що реалізує тільки модель орієнтованого графу, вона займає 22 місце. OrientDB та ArangoDB вважаються графовими СКБД, але дозволяють зберігати документи, та розробляти ключ-значення сховища. Вони займають 51 та 61 місце відповідно.

3 Постановка завдання

Виходячи з того, що існує досить багато СКБД, що підтримують графову модель, можна вважати важливим та актуальним завдання аналізу та порівняння деяких обраних СКБД. У цьому випадку можна зазначити досить вагомими такі критерії порівняння як час виконання однакових запитів в різних середовищах; наявність документації та підтримки розробниками; вартість, чи Open Source ліцензія; час, який треба затрати працівникам, щоб опанувати цю систему керування даними та мову розробки запитів. В роботі були обрані Neo4J, OrientDB та ArangoDB, як такі, що користуються найбільшою популярністю серед розробників ГБД. Для перевірки часу виконання запитів було обрано теж найбільш популярне завдання, а саме пошук даних у соціальному графі.

4 Neo4J

Neo4J [8] є СКБД із відкритим кодом на мові програмування JAVA. Вона використовується при розробці сервісів рекомендацій, соціальних мереж, системи менеджменту. Neo4J оптимізована для швидкого обходу графа, зменшуючи кількість вузлів, що беруть участь в обході, і знаючи вузол, де починається обхід. У Neo4J застосовується індексування для того, щоб також легко знайти вузол або відношення за певними властивостями, або щоб визначити стартовий вузол для обходу графа. Neo4J повністю підтримує ACID властивості баз даних і є транзакційною. Розробники забезпечують своїх користувачів інструментом для виконання запитів та візуалізації графа. Це Neo4J Browser, що має графічний інтерфейс на основі веб-браузера.

Спеціально для роботи із графом розробники спроектували мову запитів Cypher, яка використовує зіставлення за шаблоном як головний механізм відбору даних. Cypher – декларативна мова запитів. Neo4J також підтримує Gremlin. Ця мова імперативна, користувач описує як обходити граф.

Дослідження Neo4J на прикладі розробки рекомендаційної системи було розглянуте в [9]. Якщо в середовищі Neo4J розроблена база даних «Родичі», яка являє собою граф генеалогічного дерева, то запит на Cypher, який дозволяє знайти двоюрідних братів та сестер Тетяни виглядає, як наведено в лістингу 1. При цьому спочатку здійснюється пошук батьків Тетяни, потім пошук бабусь і дідусів, наступний крок – це пошук зв'язаних вузлів та повторення цього кроку для них.

Лістинг 1.

```
MATCH (Tanya:Woman {name:"Татьяна"})<- [r:PARENT] - (parent)
<- [:PARENT] - (grandparent) - [:PARENT] ->
(grandparentChild) - [:PARENT] -> (cousin)
RETURN cousin
```

Запит на мові Cypher, який вираховує друзів користувача у глибині пошуку, що дорівнює 3, наведений у лістингу 2.

Лістинг 2.

```
MATCH
(me:User {user_id:10})-[:FOLLOWS*3]->(myFriend:User)
RETURN myFriends
```

СКБД Neo4J легко встановлюється, підтримується і супроводжується розробниками, має безкоштовну версію та за передплатою, в співтоваристві Neo4J можна отримати відповіді на питання, що виникли. Мова Cypher проста в освоєнні, має строгий та чіткий синтаксис. СКБД має такі привабливі характеристики, як відсутність схеми БД, велику кількість драйверів для мов програмування, легкість у вивченні завдяки великій кількості ресурсів та документації.

5 OrientDB

На ринку представлена OrientDB [10], яка позиціонує себе як мультимодельна СКБД, що не має ніякої успадкованої основної моделі. OrientDB – це NoSQL СКБД з відкритим кодом, що розповсюджується за ліцензією Apache 2 Open Source License, написана на JAVA, і може працювати на будь-якій платформі без настройки і установки [11]. OrientDB підтримує збереження та обробку даних в документах, графах та як об'єкти «ключ-значення». Це транзакційна СКБД, її особливістю є збереження фізичних вузлів у якості документів. На відміну від більшості продуктів NoSQL, що ще й досі використовуються щоб забезпечити масштабованість додатків, що працюють на реляційних СКБД, OrientDB може бути використана, як єдина мультимодельна СКБД з відкритим кодом, що об'єднує в собі міць графіків і гнучкість документів в одній високопродуктивній операційній базі даних, що масштабується. Вона підтримує дистрибутивність на різних вузлах серверу.

У зв'язку з тим, що більшість розробників БД мають досвід з використання мови запитів SQL, автор OrientDB Luca Garulli теж зробив SQL як мову запитів. В діалект OrientDB SQL додані деякі розширення та зміни. Розширення стосуються підтримки функціональності графа, для цього введені команди створення та модифікації вузлів та ребер. Для ефективного обходу графа в діалекті застосовується команда MATCH, що підтримує пошук за шаблоном як у Neo4J. Великою перевагою вважається велика швидкість обходу окремих частин, чи усього графу в цілому, і відсутність Join об'єднань, – замість них використовуються фізичні посилання. В

OrientDB SQL підтримуються збережені процедури. В лістингу 3 наведений приклад об'єднання трьох класів за умовою в SQL та OrientDB SQL.

Лістинг 3.

SQL	OrientDB SQL
<pre>SELECT EmployeeName FROM Employee A, City B, Country C, WHERE A.city = B.id AND B.country = C.id AND C.name = 'Ukraine';</pre>	<pre>SELECT EmployeeName FROM Employee WHERE city.country.name = 'Ukraine';</pre>

Приклад запиту, який вираховує друзів користувача у глибині пошуку, що дорівнює 3, наведений у лістингу 4.

Лістинг 4.

```
MATCH {class: User, as: me,
  where:
    (user_id='10')}.out('Follows'){as:one}.out('Follows')
{as: second}.out('Follows'){as:third}
RETURN third
```

Незважаючи на привабливі особливості OrientDB, ця СКБД має недолік, такий як відсутність інструментів імпортування даних. Існують проблеми з оновленням документації, навіть таки, щоб використати якийсь метод, треба відкривати код OrientDB, дивитися та шукати в ньому. Це робить використання СКБД незручним та призводить до марної витрати часу.

6 ArangoDB

ArangoDB – багатоцільова відкрита СКБД, що надає гнучкі моделі зберігання документів, графів і даних у форматі ключ-значення [12]. Більшість коду реалізована на JavaScript. Робота з базою здійснюється через SQL-подібну мову запитів AQL, або через спеціальні розширення мовою JavaScript. Засоби для зберігання даних відповідають вимогам ACID (атомарність, узгодженість, ізольованість, надійність), підтримують транзакції й забезпечують як горизонтальну, так і вертикальну масштабованість. Управління СКБД може проводитися через веб-інтерфейс або консольний клієнт ArangoSH. Початковий код ArangoDB поширюється під ліцензією Apache 2.

В ArangoDB відсутня схема зберігання даних (Schema-free). Цю СКБД можна використовувати як сервер для веб додатків мовою JavaScript, вона має гнучку модель зберігання даних, в якій можуть бути комбіновані різні форми зберігання даних. Мова запитів AQL забезпечує користувача можливістю поєднувати типи сховищ у одному запиті. пари ключ—значення, документи і параметри, що визначають зв'язки між записами (надаються засоби для обходу вершин графа). Вузли графа в ArangoDB – це звичайні документи, а ребра – документи спеціального виду, що мають поряд зі звичайними системними полями ще системні поля `_from` і `_to`. Документи об'єднуються в колекції. Колекції документів – це ребра, які можуть виступати також і вузлами.

В лістингу 5 наведено приклад запиту на мові AQL [13], який дозволяє повернути підмножину активних користувачів (визначається атрибутом з ім'ям `status`), відсортоване по імені в порядку зростання.

Лістинг 5.

```
FOR doc IN users
  FILTER doc.status == "active"
  SORT doc.name
  LIMIT 10
```

В лістингу 6 наведено приклад AQL запиту, який вираховує друзів користувача у глибині пошуку, що дорівнює 3.

Лістинг 6.

```
FOR vertex IN 3..3 OUTBOUND "users/100" follows
  RETURN vertex
```

ArangoDB підтримує роботу із найпопулярнішими мовами програмування, має добру якість документацій та загальної інформації для початківців, але не користується такою популярністю, як перші два представники.

7 Порівняння швидкості виконання запитів

Для порівняння швидкості виконання запитів Neo4J, OrientDB та ArangoDB використаний приклад соціальної мережі. Для цього був згенерований граф з використанням моделі Барабаші-Альберта [14]. Принцип моделі полягає в тому, що чим більше зв'язків має вузол, тим більше ймовірність утворення нових зв'язків. Дійсно, принцип роботи моделі нагадує сучасні соціальні мережі. Згенерований граф має 5000 вузлів та 24900 зв'язків. Як можна побачити на рисунку 1 деякі вузли є більшими ніж інші, отже, ці вузли мають більше зв'язків.

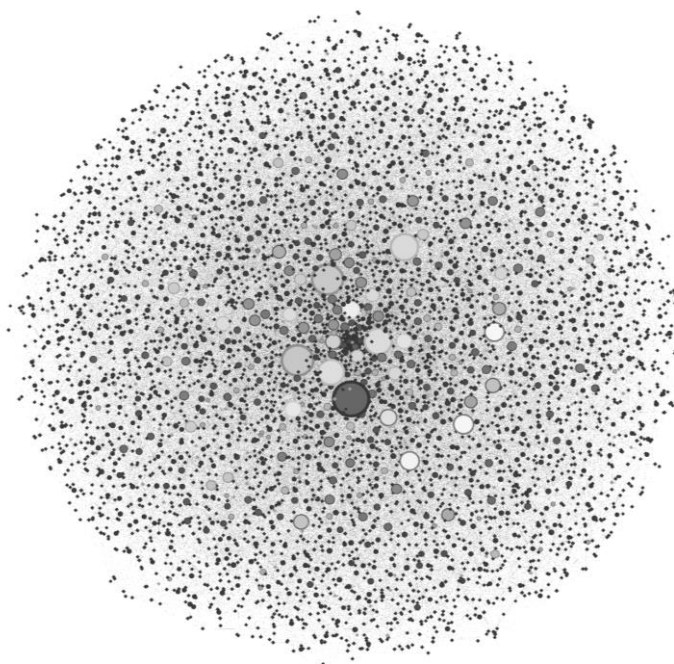


Рис1. Візуалізація графу соціальної мережі

Для порівняння часу виконання запитів на отримання зв'язків між користувачами були обрані такі стабільні версії серверів СКБД:

- Neo4J 3.5.7
- OrientDB 2.2.36
- ArangoDB 3.4.2

Тестування проводилось на комп'ютері з характеристиками:

- Операційна система Linux 16.04 64-bit
- Оперативна пам'ять 4 ГБ
- Процесор Intel Core i5-4210U
- CPU 1.70GHz

Вимір часу виконання запитів проводився виключно із використанням інтерфейсу користувачів СКБД, які поставляються із кожною базою, а саме, Neo4J Browser, ArangoDB Web Interface та OrientDB Studio. Для тестування були обрані ID трьох користувачів (10, 100, 555). Максимальна глибина пошуку складала 5. Для порівняння результатів був обраний середній час одного запиту за 6 виконань. На рисунку 2 відображено час у мілісекундах, у дужках вказана кількість вузлів на даній глибині. Тобто користувач «555» має 9 друзів, які, у свою чергу мають 29 друзів і т.д.

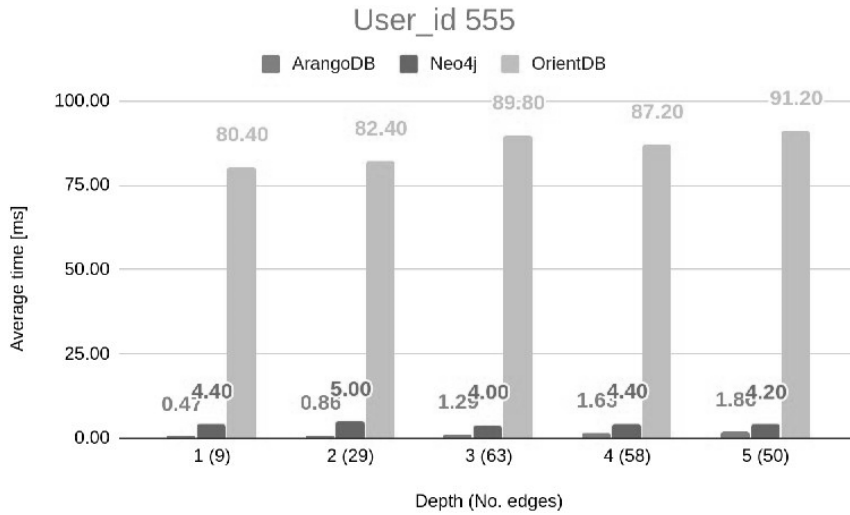


Рис 2. Графік часу виконання запиту на пошук друзів для користувача із номером 555

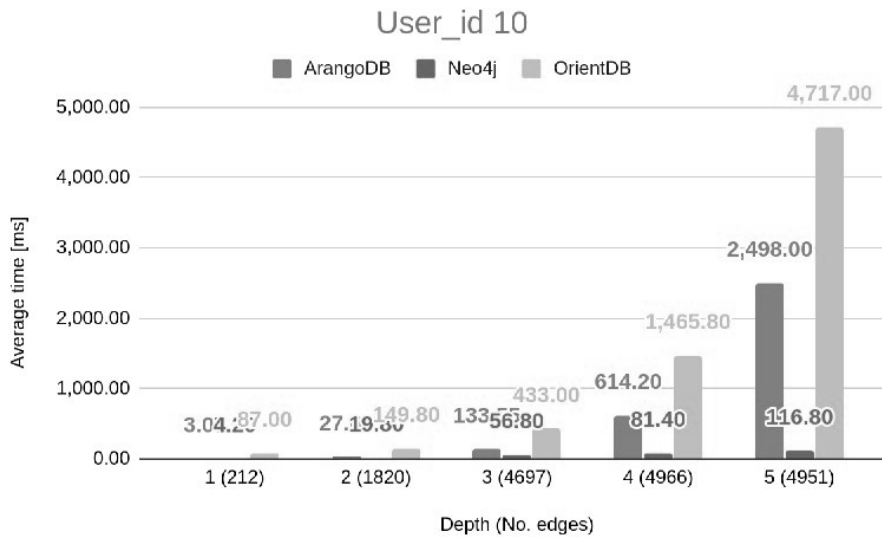


Рис 3. Графік часу виконання запиту на пошук друзів для користувача із номером 10

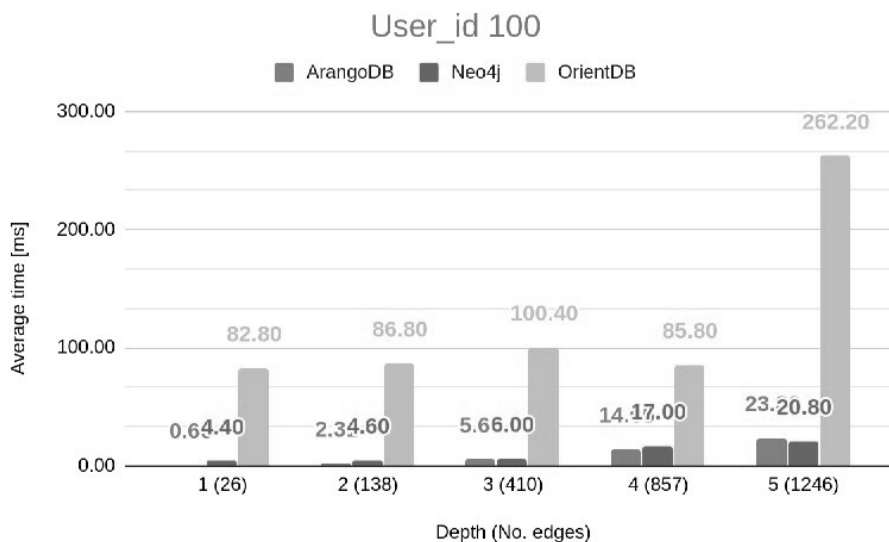


Рис 4. Графік часу виконання запиту на пошук друзів для користувача із номером 100

8 Висновки

За отриманими результатами тестування можна зробити наступні висновки. OrientDB є найповільнішою з трьох розглянутих СКБД. Причиною може бути використання неоптимальних запитів, але в зв'язку з браком документації оптимізувати їх важко. Отже, використання цієї СКБД для побудови соціального графу, на основі даного дослідження, не рекомендується. Щодо ArangoDB та Neo4J, вони конкурують між собою у деяких випадках. А саме, з рисунку 2 можна побачити, що Neo4J має приблизно однаковий час виконання для всіх 5-ти випадків, навіть, враховуючи різну кількість вузлів на кожному з них. Час ArangoDB зростає із кожною глибиною. Можна зробити висновок, що для отримання малої кількості вузлів, приблизно до 1000, ArangoDB випереджає Neo4J. Але, виходячи з аналізу результатів, відображених на рисунках 3 і 4, можна стверджувати, що Neo4J значно випереджає ArangoDB на великих об'ємах. Використання обох з них підходить для соціальних мереж, і завдяки швидкості виконання запитів, і за простотою написання самих запитів. Але, треба заважити, завдяки тому що Neo4J прекрасно справляється з великими об'ємами даних, ця СКБД може бути використана для Big Data.

Висновки, зроблені на підставі проведених досліджень, підтверджуються результатами ранжування систем управління базами даних відповідно до їх популярності, що проводяться Solid IT щомісячно [7]. Це стосується популярності Neo4J, але треба визначили, що ArangoDB, згідно нашому дослідженню, трохи випереджає OrientDB, хоча до Neo4J їм обом далеко.

ЛІТЕРАТУРА

1. Codd E.F. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*. 13 (6): 377–387. doi:10.1145/362384.362685 (June 1970).
2. NoSQL: Матеріал з Вікіпедії – вільної енциклопедії. [Електронний ресурс] Режим доступу: <https://ru.wikipedia.org/wiki/NoSQL>
3. Мартин Фаулер, Прамодкумар, Дж. Садаладж. NoSQL. Новая методология разработки нереляционных баз данных.: пер. с англ. М.: ООО "И.Д. Вильямс", 2013. 192 с.
4. NitrosData: Применение графовых баз данных. [Электронный ресурс] Режим доступа: <http://nitrosdata.ru/2019/02/20/primenenie-grafovyh-baz-dannyh/>
5. Gartner Identifies Top 10 Data and Analytics Technology Trends for 2019. [Електронний ресурс] Режим доступу: <https://www.gartner.com/en/newsroom/press-releases/2019-02-18-gartner-identifies-top-10-data-and-analytics-technolo>
6. Ian Robinson Graph Databases / Ian Robinson, Jim Webber, Emil Eifrem. O'Reilly Media, 2013. 178 p.
7. DB-Engines Ranking provided by solid IT, September 2019. [Електронний ресурс] Режим доступу: <https://db-engines.com/en/ranking>
8. Документація Neo4j. [Електронний ресурс] Режим доступу <https://neo4j.com/docs/>
9. Тимошенко Е.С., Лазурик В.М. Реляционные или графовые базы данных: Труды международной науч.-техн. конференции. *Компьютерное моделирование в наукоемких технологиях*. Харьков: ХНУ им. В.Н.Каразина, 2018 . С. 289-292.
10. OrientDB: Матеріал з Вікіпедії – вільної енциклопедії. [Електронний ресурс] Режим доступу: <https://ru.wikipedia.org/wiki/OrientDB>
11. OrientDB Community. [Електронний ресурс] Режим доступу <https://orientdb.org/docs/>
12. ArangoDB: Матеріал з Вікіпедії – вільної енциклопедії. [Електронний ресурс] Режим доступу: <https://ru.wikipedia.org/wiki/ArangoDB>
13. ArangoDB v3.5.0 Documentation. [Електронний ресурс] Режим доступу: <https://www.arangodb.com/docs/stable/>
14. Модель Барабаши — Альберт. [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/Модель_Барабаши_—_Альберт.

REFERENCES

1. E. F. Codd, "A Relational Model of Data for Large Shared Data Banks." *Communications of the ACM*, vol. 13, pp. 377–387, 1970.

2. "NoSQL" *En.wikipedia.org*. Internet: <https://eu.wikipedia.org/wiki/NoSQL>, Aug 17, 2019 [Oct 25, 2019].
3. P. Sadalage and M. Fowler, *NoSQL distilled*. Upper Saddle River, NJ: Addison-Wesley, 2015.
4. NitrosData, "Usage of graph databases". Internet: <http://nitrosdata.ru/2019/02/20/primenenie-grafovyh-baz-dannyh/> [Oct 20, 2019].
5. "Gartner Identifies Top 10 Data and Analytics Technology Trends for 2019." Internet: <https://www.gartner.com/en/newsroom/press-releases/2019-02-18-gartner-identifies-top-10-data-and-analytics-technolo> Feb, 2019 [Sep 25, 2019].
6. I. Robinson, and J. Webber, E. Eifrem, *Graph Databases*. O'Reilly Media, 2013.
7. DB-Engines Ranking provided by solid IT. Internet: <https://db-engines.com/en/ranking>, Sep, 2019. [Sep 30, 2019]
8. Documentation Neo4j. Internet: <https://neo4j.com/docs/> [Oct 27, 2019]
9. Y. Tymoshenko and V. Lazurik "Rational vs graph databases" *Proceedings of the international scientific and technical conference "Computer modeling in high technology"*, Kharkiv: V. N. Karazin Kharkiv National University, pp. 289-292, 2018. [in Russian]
10. "OrientDB" *En.wikipedia.org*. Internet: <https://ru.wikipedia.org/wiki/OrientDB>, Oct 11, 2019 [Oct 27, 2019].
11. OrientDB Community. Internet: <https://orientdb.org/docs/> [Oct 27, 2019]
12. "ArangoDB" *En.wikipedia.org*. Internet: <https://en.wikipedia.org/wiki/ArangoDB> [Oct 27, 2019]
13. ArangoDB v3.5.0 Documentation. Internet : <https://www.arangodb.com/docs/stable/> , [Oct 27, 2019].
14. "Barabasi-Albert model" *En.wikipedia.org*. Internet: https://en.wikipedia.org/wiki/Barabasi-Albert_model, Oct 21, 2019 [Oct 27, 2019].

Лазурик Валентина Михайлівна – старший викладач кафедри штучного інтелекту та програмного забезпечення, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022;

e-mail: lazurik@hotmail.com; ORCID: 0000-0002-3340-9780.

Lazurik Valentina M. Senior lecturer of Department of Artificial Intelligence and Software V.N. Karazin Kharkiv National University, Svobody Sq 4, 61022, Kharkiv, Ukraine,

e-mail: lazurik@hotmail.com; ORCID: 0000-0002-3340-9780.

Лазурик Валентина Михайловна – старший научный сотрудник кафедры программного обеспечения и программного обеспечения, Харьковский национальный университет имени В.Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: lazurik@hotmail.com; ORCID: 0000-0002-3340-9780.

Тимошенко Єлизавета Станіславівна – магістрант факультету комп'ютерних наук, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: liza.timoshenko0603@gmail.com; ORCID: 0000-0001-6885-289X.

Tymoshenko Yelyzaveta S. Undergraduate of the Faculty of Computer Science V. N. Karazin Kharkiv National University, Svobody Sq 4, 61022, Kharkiv, Ukraine ;

e-mail: liza.timoshenko0603@gmail.com; ORCID: 0000-0001-6885-289X.

Тимошенко Єлизавета Станиславовна - магистрант факультета компьютерных наук, Харьковский национальный университет им. В.И. Н. Каразина, площадь Свободы, 4, Харьков-22, Украина, 61022; e-mail: liza.timoshenko0603@gmail.com; ORCID: 0000-0001-6885-289X.