Выбор оптимальной базы данных для работы с графом

И. В. Гражданкин, email: grazhdankin.i.l@yandex.ru
С. В. Власов, email: svv@cs.vsu.ru
А. В. Чавдаров

Воронежский государственный университет

Аннотация. *Тестирование и анализ полученных результатов сравнения графовых баз данных Neo4j и Sparksee в контексте работы с социальным графом.*

Ключевые слова: *граф, NoSQL, графовые базы данных, Neo4j, Sparksee.*

Введение

В последние годы происходит стремительный рост интереса к системам хранения и обработки графовых данных[1].

Графовые базы данных используются для хранения взаимосвязей сущностей и навигации в них. Взаимосвязи в таких базах данных являются главной ценностью этих баз данных. Узлы используются для хранения сущностей данных, а ребра для хранения взаимосвязей между сущностями. Ребро всегда имеет исходный узел, конечный узел, тип и направление. Ребра могут описывать взаимосвязи типа “родитель-потомок”, действия, права владения. Существуют ограничения на количество взаимосвязей, которое может иметь узел.

Обход графа в графовой базе данных можно выполнять либо по определенным типам ребер, либо по всему графу. Обход соединений или взаимосвязей в графовых базах данных выполняется очень быстро, поскольку взаимосвязи между узлами не вычисляются во время выполнения запроса, а хранятся в базе данных. Графовые базы данных имеют ряд преимуществ в таких примерах использования, как социальные сети, сервисы рекомендаций и системы выявления мошенничества, маршруты перевозок, дорожные карты, когда требуется создавать взаимосвязи между данными и быстро их запрашивать[2].

[1] https://bpm-systems.ru/bpm-trends/

[2] https://veesp.com/ru/blog/sql-or-nosql/

Целью настоящей работы является тестирование и анализ скорости выполнения запросов для двух реализаций графовых баз данных Neo4j и Sparksee3.

1. Выбор базы данных для работы с графами

Для работы с графами можно использовать обычные реляционные базы данных4. Загрузить графы в виде пары ребер при достаточном количестве внешней памяти проблем это не вызовет. Но при сценарии обхода графа, получения подгрупп или выполнение любой другой аналитической операции могут возникнуть проблемы5.

Использование NoSQL6 для хранения графа и выполнения аналитических операций обладает существенными преимуществами6.

Подгруппой NoSQL хранилищ являются графовые базы данных7, которые были специально разработаны для хранения графов.

В данной работе для проведения экспериментов и анализа использовалась эмуляция социального графа8. Сегодня взаимосвязи между пользователями в социальных сетях больше несколько сотен миллиардов9. Для связей в графах используют ребра, поэтому в тестах будет увеличиваться количество ребер. Тестирование проводилось на базе данных Neo4j, так как:

- самая распространенная база данных
- имеет общирный функционал
- есть бесплатная версия

Вторая база данных для тестирования Sparksee. Основные преимущества:

- самая заявлена высокая производительность

---

[3] http://www.sparsity-technologies.com

[4] https://ru.bmstu.wiki/Реляционная_база_данных

[5] https://habr.com/ru/company/ruvds/blog/324936/

[6] https://ru.wikipedia.org/wiki/NoSQL

[7] https://ru.wikipedia.org/wiki/Графовая_база_данных

[8] https://ru.wikipedia.org/wiki/Социальный_граф

[9] https://www.web-canape.ru/business/vsya-statistika-interneta-na-2019-god-v-mire-i-v-rossii/

– бесплатная версия для исследований

2. Методология тестирования

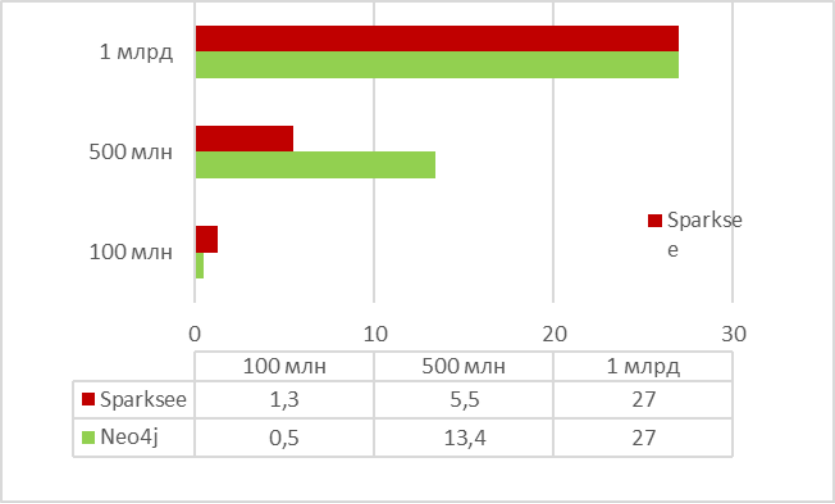
Тестовый стенд:
– Intel Xeon X6550 2.0Gz
– 80GB DDR3
– 2Tb hard drive

Настройка Neo4j:
– Version 3.5.11
– Ubuntu 18.04 LTS
– Cache size 60Gb

Настройка Sparksee:
– Version 5.2
– Ubuntu 18.04 LTS
– Cache size 60Gb

Аналитические запросы:
– Получить всех соседей вершины – простая задача
– Найти кратчайший путь – более сложная задача
– Выполнить обход графа – сложная задача

3. Анализ полученных результатов



	100 млн	500 млн	1 млрд
Sparksee	1,3	5,5	27
Neo4j	0,5	13,4	27

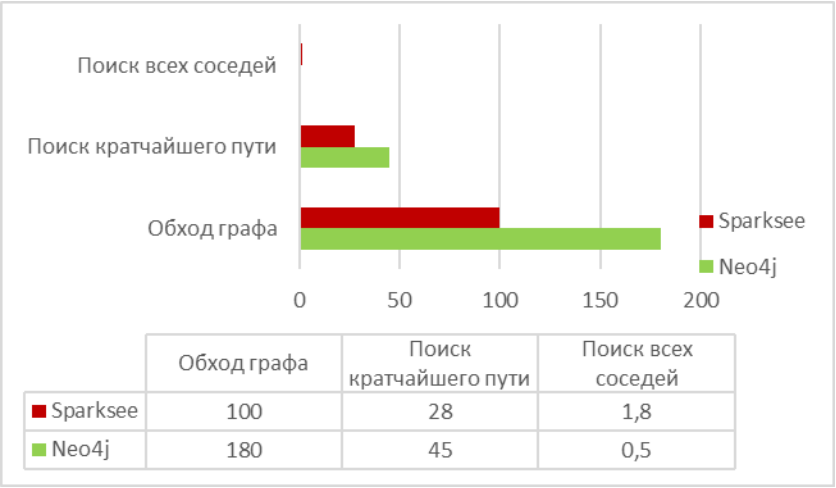
Рис. 1. Время импорта данных

Тестирование проводилось на 3 наборах ребер: 100 млн ребер, 500 млн ребер, 1 млрд ребер. В каждом следующем тесте увеличивали количество ребер, так как количество вершин не бывает так много как ребер – это естественное состояние графа. Тестирование проводилось на 3 наборах ребер: 100 млн ребер, 500 млн ребер, 1 млрд ребер. В каждом следующем тесте увеличивали количество ребер, так как количество вершин не бывает так много как ребер – это естественное состояние графа.

Как видно из графика обе базы данных с импортом 100 млн ребер справились быстро, Neo4j чуть менее часа, Sparksee более часа.

На наборе данных из 500 млн ребер появляются проблемы у Neo4j время импорта в два раза больше, чем у Sparksee.

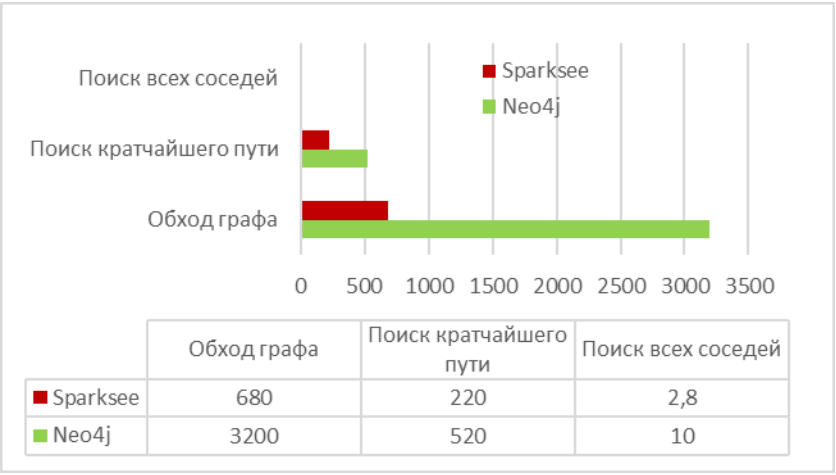
Импорт набор данных из 1 млрд ребер занимает более суток у двух баз данных, поэтому в следующих тестах этот набор ребер не участвовал.



	Обход графа	Поиск кратчайшего пути	Поиск всех соседей
Sparksee	100	28	1,8
Neo4j	180	45	0,5

Рис. 2. Время обработки аналитических запросов 100 млн

Как видно из графика обе базы данных аналитические запросы выполняют относительно быстро. Самая долгая обработка запроса занимает 180 секунд у Neo4j обход графа. Но из графика видно, что Sparksee тратит меньше времени для “Поиск кратчайшего пути” и “Обход графа”. Более простой запрос “Поиск всех соседей” Neo4j выполняет быстрее.



	Обход графа	Поиск кратчайшего пути	Поиск всех соседей
Sparksee	680	220	2,8
Neo4j	3200	520	10

Рис. 3. Время обработки аналитических запросов 500 млн

Как видно из рисунка-3 у Neo4j возникают сложности с обходом графа. Время обхода у Sparksee в 6 раз меньше. “Поиск кратчайшего пути” у Neo4j занимает адекватное время, но с этим же запросом Sparksee справляется в разы быстрее. “Поиск всех соседей” у Neo4j занимает 10 секунд, у Sparksee менее 3 секунд. С легким запросом Sparksee справляется тоже в разы быстрее, чем Neo4j.

4. Выводы по результатам исследований

На основе результатов исследований видно, что разумное время импорта данных получаем если, количество ребер не превышает одного миллиарда.

Sparksee более предпочтительнее так как он более производительнее, чем Neo4j. В основном Sparksee показывает себя более производительнее. Очень хорошо это заметно при выполнении сложных аналитических запросов, и при использовании большой части графа.

Главной характеристикой для высокой производительности графовых баз данных является оперативная память. Поэтому для более быстрой работы с графом, у которого количество ребер превышает один миллиард лучше использовать кластерные решения.

Заключение

В ходе работы были изучены графовые базы данных. Изучены особенности Neo4j и Sparksee. Проведено тестирование этих двух баз данных с различным количеством ребер. После анализа полученных результатов в ходе тестирования можно утверждать, что для работы с большими графами, а социальный граф считается большим графом, лучше использовать Sparksee.

Список литературы

1. Графовые базы данных: новые возможности для работы со связанными данными / Ян Робинсон, Джим Вебер, Эмиль Эифрем. – 2016. – 258 с.
2. Graph Databases in Action / Dave Bechberger, Josh Perryman. – 2020. – 366 с.
3. Neo4j Documentation [Электронный ресурс]. – Режим доступа: https://neo4j.com/docs/
4. Sparksee User Manual [Электронный ресурс]. – Режим доступа: http://sparsity-technologies.com/UserManual/Index.html