УДК 004.652.5

Использование графовых баз данных в целях оптимизации анализа биллинговой информации

© М.В. Бартенев, И.Э. Вишняков

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Сформулированы основные задачи обработки биллинговой информации и рассмотрена возможность оптимизации их решения с использованием графовых баз данных, поскольку они обеспечивают наиболее естественное представление и дополнительные средства эффективной реализации алгоритмов анализа связей в социальных сетях. Выполнен краткий обзор графовых систем управления базами данных Sones, Neo4J и DEX и предоставляемых ими средств, а также сравнительное тестирование их производительности и реляционной системы управления базами данных Microsoft SQL Server 2012 на поставленных задачах анализа информации. Сделан вывод о применимости той или иной базы данных в зависимости от объема обрабатываемых данных.

***Ключевые слова**: NoSQL, графовые базы данных, анализ биллинговой информации.*

Введение. С 1980-х годов реляционные системы управления базами данных (СУБД) стали занимать доминирующее положение среди средств хранения данных. Несмотря на то что реляционные хранилища обеспечивают наилучшее сочетание простоты, устойчивости, гибкости, производительности, масштабируемости и совместимости, их показатели по каждому из этих пунктов не обязательно выше, чем у аналогичных систем, ориентированных на какую-то одну особенность. Однако универсальность реляционных СУБД перевешивала какие-либо другие недостатки.

Сегодня ситуация несколько иная. Появившиеся в последние годы так называемые NoSQL (Not only SQL, не только SQL) хранилища реализуют модели данных, имеющие существенные отличия от традиционной реляционной модели. Основная их цель — расширить возможности баз данных (БД) в тех областях, где реляционная модель и SQL недостаточно гибки, и не вытеснять их там, где они справляются со своими задачами. Создатели таких БД среди множества преимуществ использования NoSQL-решений называют высокую производительность при использовании специфических моделей данных и легкость работы с ними.

Одним из наиболее популярных и актуальных подвидов нереляционных хранилищ являются графовые БД [1]. Как ясно из названия,

основная модель данных в них — классический математический граф. Проекты в области графовых БД начали появляться с конца 1980-х годов, однако в большей степени носили академический характер [2, 3]. В последнее время наблюдается бурный рост интереса к графовым БД в связи с тем, что такая система представления данных оказалась естественной и востребованной в современном мире различных социальных связей (Интернет, социальные сети и т. д.). К достоинствам графовых моделей БД по сравнению с традиционной реляционной моделью исследователи относят не только возможность естественной реализации графовых операций (поиска путей, выделения сообществ и т. п.), но и гибкую схему данных, позволяющую унифицировать хранение разнородных объектов [1, 4].

Биллинг оператора сотовой связи представляет собой информацию обо всех звонках за определенный период внутри сотовой сети. Эти данные естественным образом представляются в виде графа, где вершина — абонент, а ребро — звонок между абонентами. За последние годы наблюдается существенный рост активности абонентов сотовой связи и, соответственно, кратное увеличение объемов биллинговой информации. Следовательно, все более и более долгой и трудоемкой становится обработка с помощью реляционных хранилищ. Вариантом оптимизации такой обработки могут стать аналитические задачи, решаемые с помощью графовых БД.

В данной статье будет рассмотрено три различных реализации нереляционных графовых БД, ядро каждой из которых написано на разных языках программирования: Sones (C#) [5], Neo4J (Java) [6] и DEX (C++) [7]. Цель рассмотрения этих хранилищ — оценка их возможностей по легкой и быстрой работе с графовыми данными и увеличению скорости обработки биллинговой информации. Для решения поставленной задачи будет проведена оценка производительности операций импорта данных и двух наиболее востребованных задач анализа биллинга. Кроме того, для сравнения будут предложены результаты этих же задач, реализованных с помощью реляционной базы данных Microsoft SQL Server 2012 [8].

Задачи анализа биллинговой информации. Будем называть графом упорядоченную пару $G := (V, E)$, в которой V — непустое множество вершин, а E — множество пар вершин, представляющих ребра. В случае ориентированного графа пары в E упорядочены.

Рассмотрим операции, с помощью которых тестировались графовые БД. Первая задача, которую необходимо реализовать, — импорт графа из файла определенной структуры в БД. Данные хранятся в текстовых файлах, где каждая строка описывает ребро графа и имеет следующий вид:

{link_id; source_node; destination_node}.

Здесь *source_node* — строковое значение, идентифицирующее вершину-источник ребра; *destination_node* — строковое значение, идентифицирующее вершину-назначение ребра; *link_id* — целочисленное 64-разрядное число, дополнительный атрибут на ребре.

Без ограничения общности можем считать, что такой схемы данных достаточно для представления практически любого графа [1], поскольку атрибут *link_id* может являться ссылкой на строку в таблице реляционной БД, содержащей все параметры ребра. Аналогичный атрибут можно ввести и для каждой вершины, но для наших целей он не требуется.

Одна из самых актуальных задач, возникающих при анализе биллинговой информации, — поиск соседних вершин какой-либо определенной заданной вершины. Помимо непосредственно прямых соседей вершины может возникнуть потребность узнать всех соседей, находящихся на определенном удалении от заданной вершины. Для тестирования графовых БД была реализована функция, в которую передается какое-либо n, а возвратиться должны все соседние вершины, до которых минимальное расстояние не больше n. Обозначим кратчайшее расстояние от вершины s до какой-либо вершины v функцией $\sigma_s(v)$. Тогда результатом работы реализованной функции будет множество

$$V_{bfs} = \{v \mid \sigma_s(v) \le n,\ \ v \in V\}.$$

Еще одна востребованная операция обработки биллинговой информации, на которой проводилось тестирование, — пересечение найденных соседей двух вершин. Эта задача была реализована как функция, в которую передается список вершин, для каждой вершины находятся ее соседи не дальше определенного уровня, после чего для всех пар вершин проводится пересечение таких множеств. Если пересечение проводилось для найденных соседей вершин s_1 и s_2, результатом будет множество

$$V_{i\sec t} = \{v \mid \sigma_{s1}(v) \le n,\ \ v \in V\} \cap \{v \mid \sigma_{s2}(v) \le n,\ \ v \in V\}.$$

Обзор графовых баз данных. Проведем краткий обзор рассматриваемых графовых БД с целью узнать их основные возможности, с помощью которых проводилось тестирование.

Sones GraphDB. Эта графовая БД разработана компанией Sones в 2009 г. и поддерживалась до конца 2011 г., после чего компания обанкротилась. За время поддержки была выпущена версия 2.1, доступная в двух вариантах: версия Community распространяется по лицензии AGPL v3, для коммерческих же проектов необходимо приобретать версию Enterprise. Кроме того, важное отличие платной вер-

сии от бесплатной состоит в наличии возможности устойчивого хранения базы на жестком диске. Бесплатная же версия позволяет хранить данные только в оперативной памяти (in-memory).

Поскольку компания-разработчик прекратила свое существование, единственная версия БД, которую можно протестировать, — бесплатная версия Community, т. е. тестирование будет ограничено небольшими объемами данных, целиком помещающимися в оперативную память. Соответственно, работать с такими данными можно только в случае либо их маленького объема, либо постоянной очистки текущего хранилища от старых данных.

Sones не имеет встроенной поддержки алгоритмов обхода графов, в связи с чем для тестирования задачи поиска соседей был реализован поиск в ширину [9]. Кроме того, для тестирования поиска пересечений множеств соседних вершин была реализована работа с множествами найденных вершин.

Neo4J GraphDB. Эта графовая БД разработана компанией Neo Technology в 2009 г. Самая последняя стабильная на текущий момент версия — 1.9. Она доступна в трех вариантах: Community, Advanced и Enterprise. Версия Community распространяется по лицензии AGPL v3, для коммерческих же проектов необходимо приобретать версию Advanced, включающую дополнительные возможности мониторинга состояния базы. Версия Enterprise, кроме того, поддерживает резервное копирование и масштабируемость. На сегодняшний день Neo4J является наиболее популярной графовой БД, в первую очередь ввиду того, что бесплатная версия предлагает все необходимые инструменты для полноценной работы с графами.

Neo4J, в отличие от графовой БД Sones, может устойчиво хранить данные на жестком диске. Следовательно, объем хранящейся информации ограничен лишь объемами жесткого диска. Для достижения максимальной производительности в Neo4J существует два типа кэширования: файловый кэш (file buffer cache) и объектный кэш (object cache). Первый кэширует данные с жесткого диска, целью чего является увеличение скорости чтения/записи на жесткий диск. Второй кэш хранит в себе различные объекты графа: вершины, ребра и свойства в специальном оптимизированном формате для увеличения производительности обходов графа.

Neo4J обладает режимом импорта большого объема данных BatchInserter [10]. В этом режиме происходит отключение транзакций в БД, следствием чего является резкое увеличение скорости импорта. Кроме того, в Neo4J поддерживаются алгоритмы обхода графов, в том числе можно сделать обход в ширину до заданного уровня, что как раз и необходимо для решения задачи поиска соседних вершин.

Для пересечения двух множеств, как и в случае с базой данных Sones, была реализована работа с множествами.

DEX GraphDB. Эта база разработана компанией Sparsity Technologies в 2008 г. Самая последняя стабильная на текущий момент версия — 4.8. Существует три типа лицензий, по которым предоставляется право пользования DEX GraphDB: Community, Commercial и Education. При наличии лицензии Community в базе суммарно может содержаться не более 1 млн вершин и разрешен доступ к чтению данных только одному потоку. По лицензии Commercial отсутствуют какие-либо ограничения, но годовая стоимость владения лицензией зависит от суммарного количества объектов в базе и количества потоков, имеющих доступ к чтению данных. Лицензия Education предназначена для бесплатного предоставления исследовательским и учебным некоммерческим проектам.

DEX GraphDB, как и Neo4J, имеет полноценную поддержку устойчивого хранения данных. Но в отличие от Neo4J ядро DEX написано на C++, что очень хорошо сказывается на производительности. Графовая БД DEX имеет только один объектный кэш, который держит в оперативной памяти все часто используемые объекты хранилища [11].

БД DEX имеет широкие возможности по обходу графов. Она предоставляет множество встроенных алгоритмов обхода графов, таких как поиск в глубину, поиск в ширину, Дейкстры и поиска компонентов сильной связности. Кроме того, существует встроенная поддержка работы с множествами. Таким образом, обе поставленные задачи анализа биллинговой информации можно решить, используя только встроенные средства БД.

Тестирование. При тестировании использовались самые последние версии БД, доступные на данный момент: Sones v.2.1, Neo4J v.1.9 и DEX v.4.8. Все тестовые операции реализованы на языке программирования C# (Sones, DEX) и Java (Neo4J) с помощью API, предоставляемого каждой из БД. Для конфигурирования применялись стандартные настройки, а также некоторые рекомендации из официальной технической документации: использование BatchInserter при импорте данных в Neo4J и целочисленных идентификаторов в DEX. В Microsoft SQL Server 2012 создаются две таблицы (для списка вершин и для списка ребер), а обход графа реализован с помощью хранимой процедуры [12].

Тестовые задачи выполнялись в том же порядке, в каком они представлены в таблицах: импорт, поиск соседей, поиск пересечения множеств соседей. В случае если выполнение не завершалось в течение 12 ч, операция прерывалась в связи с неактуальностью столь долгого выполнения на предложенном объеме данных. Тестовый стенд, на котором проводилось тестирование, имеет следующую конфигу-

рацию: Intel Core i5 3.1 GHz, 8 Gb DDR3, 1 Tb 7200 rpm hard drive, Windows Server 2008 x64. В БД DEX и Neo4J значение размера кэша устанавливалось равным 4 Gb. При работе с Neo4J Java Virtual Machine запускалась со следующими параметрами: -d64 -server — Xmx4096m -XX:+UseConcMarkSweepGC.

Для тестирования было предложено четыре набора данных, содержащих различное количество вершин и ребер, структура графа при этом соответствовала типичной структуре биллинговой информации. В самом маленьком количество объектов приблизительно равно 1 млн, а в самом большом — 100 млн. Результаты тестирования представлены в табл. 1–4.

Таблица 1

16 000 вершин и 600 000 ребер

Тестовая БД	Время, с			Размер БД, Мб
	импорта данных	поиска соседей вершины до максимального уровня	поиска пересечения множеств соседей для двух вершин	
Neo4J	6	3	9	21
DEX	11	0,3	0,5	30
SQL Server	9	4	8	45

Таблица 2

100 000 вершин и 1 200 000 ребер

Тестовая БД	Время, с			Размер Бб, МБ
	импорта данных	поиска соседей вершины до максимального уровня	поиска пересечения множеств соседей для двух вершин	
Sones	78	23	50	1510 (RAM)
Neo4J	13	10	31	1308
DEX	20	0,4	0,7	1809
SQL Server	19	16	42	2835

Таблица 3

4 000 000 вершин и 38 000 000 ребер

Тестовая БД	Время, с			Размер Бб, МБ
	импорта данных	поиска соседей вершины до максимального уровня	поиска пересечения множеств соседей для двух вершин	
Sones	4150	301	622	5792 (RAM)
Neo4J	2430	Н/д*	Н/д*	5197
DEX	770	23	72	7238
SQL Server	2115	215	506	10 586

* Нет данных.

Таблица 4

10 000 000 вершин и 90 000 000 ребер

Тестовая БД	Время, с			Размер Бб, МБ
	импорта данных	поиска соседей вершины до максимального уровня	поиска пересечения множеств соседей для двух вершин	
Sones	Н/д	Н/д	Н/д	Н/д
Neo4J	12 270	Н/д	Н/д	13 215
DEX	2045	73	143	18 103
SQL Server	7392	612	1398	26 209

Несмотря на то что данные в Sones полностью содержатся в оперативной памяти и никакие операции чтения/записи с жесткого диска не совершаются, импорт данных в нее все равно занял больше всего времени. При операциях импорта на небольших объемах данных (тестовые наборы 1 и 2) наилучшую производительность показала Neo4J, но она резко снизилась на больших графах. На тестовых графах 3 и 4 бесспорным лидером оказалась графовая БД DEX. Последний граф загрузить в БД Sones вообще не удалось в связи с отсутствием большого количества оперативной памяти.

При выполнении аналитических операций на маленьких графах все испытуемые БД показали адекватные результаты, но на больших объемах данных производительность Neo4J, как отмечалось, очень заметно снизилась, и дождаться выполнения операции не удалось. DEX GraphDB показывает достойные результаты во всех тестах (см. также [1]).

Заключение. В статье проведено исследование трех графовых БД с целью определения возможности их использования для оптимизации анализа биллинговой информации. Из полученных результатов можно сделать вывод о том, что все они вполне жизнеспособны и показывают неплохие результаты, особенно на небольших объемах данных. Однако графовой БД Sones вряд ли найдется применение в реальном мире, поскольку она хранится только в оперативной памяти и при этом проигрывает по производительности всем остальным БД. Neo4J может быть использована в качестве хранилища для обработки небольших подграфов, объемом до 1,5 млн объектов, поскольку с такими графами она работает очень быстро. Кроме того, Neo4J, являясь наиболее популярной графовой БД, обладает огромным набором различных API для разных языков программирования и множеством дополнительных функций, что делает ее наиболее простой в работе.

БД DEX является единственным из рассмотренных продуктов, способным действительно быстро обрабатывать большие графы. Время выполнения в ней линейно зависит от количества загружен-

ных объектов. Этот продукт способен стать решением для обработки биллинговых данных.

ЛИТЕРАТУРА

[1] Dominguez-Sal D., Urbon-Bayes P., Gimenez-Vano A., Gomez-Villamor S., Martınez-Bazan N., Larriba-Pey J.L. Survey of graph database performance on the HPC scalable graph analysis benchmark. *Proceedings of the 2010 int. conf. on Web-age information management* (WAIM'10). Berlin, Heidelberg, Springer-Verlag, 2010, pp. 37–48.

[2] Angles R., Gutierrez C. Survey of Graph Database Models. *ACM Computing Surveys*, 2008, vol. 40 (1), pp. 1:1–1:39.

[3] Angles R. A comparison of current graph database models. *Proceedings of the 2012 IEEE 28th Int. Conf. on Data Engineering Workshops*, (ICDEW'12). Wash., IEEE Computer Society, 2012, pp. 171–177.

[4] Vicknair C., Macias M., Zhao Z., Nan X., Chen Y., Wilkins D. A comparison of a graph database and a relational database: a data provenance perspective. *Proceedings of the 48th Annual South-East Regional Conf.* (ACM SE'10). N.Y., ACM, 2010, pp. 42:1–42:6.

[5] Sones Graph Database. URL: http://www.sones.de/web/sones/home

[6] Neo4J Graph Database. URL: http://www.neo4j.org

[7] DEX Graph Database. URL: http://www.sparsity-technologies.com/dex.php

[8] Microsoft SQL Server 2012 Developer Resource. URL: http://msdn.microsoft.com/en-us/sqlserver/aa336270.aspx

[9] Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. *Алгоритмы. Построение и анализ*. Москва, Вильямс, 2012, 1296 с.

[10] Neo4J BatchInserter. URL: http://docs.neo4j.org/chunked/milestone/batchinsert.html.

[11] DEX. Cache configuration. URL: http://www.sparsity-technologies.com/dex_tutorials5?name=Configuration

[12] Graphs and Graph Algorithms in T-SQL. URL: http://hansolav.net/sql/graphs.html

Статья поступила в редакцию 26.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Бартенев М.В., Вишняков И.Э. Использование графовых баз данных в целях оптимизации анализа биллинговой информации. *Инженерный журнал: наука и инновации,* 2013, вып. 11. URL: http://engjournal.ru/catalog/it/hidden/1058.html

Бартенев Максим Владимирович родился в 1993 г., студент 5-го курса кафедры «Теоретическая информатика и компьютерные технологии» МГТУ им. Н.Э. Баумана. Специализируется в области графовых баз данных. e-mail: max.bartenev@yandex.ru

Вишняков Игорь Эдуардович родился в 1980 г., окончил кафедру «Программного обеспечения ЭВМ и компьютерных технологий» МГТУ им. Н.Э. Баумана в 2003 г. Старший преподаватель кафедры «Теоретическая информатика и компьютерные технологии» МГТУ им. Н.Э. Баумана. Автор семи публикаций. Специализируется в области хранения, обработки и визуализации больших массивов данных. e-mail: vscie@yandex.ru