

Краткие сообщения

УДК 004.65, 004.272, 519.174.1

ПОДХОД К РАЗБИЕНИЮ СВЕРХБОЛЬШИХ ГРАФОВ С ПОМОЩЬЮ ПАРАЛЛЕЛЬНЫХ СУБД

К.С. Пан

Разбиение графов на подграфы представляет собой интересную задачу интеллектуального анализа графов, которая находит свое применение в ряде теоретических и практических задач (раскраска графа, проектирование БИС и ПЛИС, конечно-элементное моделирование и др.). Существующие последовательные и параллельные алгоритмы предполагают возможность размещения графов и промежуточных данных обработки в оперативной памяти и неприменимы для случая сверхбольших графов. Представлен подход к обработке сверхбольших графов на основе использования параллельной реляционной СУБД PostgreSQL, разработанной на базе свободной СУБД PostgreSQL.

Ключевые слова: интеллектуальный анализ, разбиение графов, параллельные СУБД.

Введение

В настоящее время одной из актуальных областей приложения технологий Data Mining [1–3] являются задачи интеллектуального анализа сверхбольших графов (имеющих сотни тысяч вершин и/или ребер), возникающие при моделировании сложных структур: химических соединений, белковых структур, биологических и социальных сетей, Web, потоков работ, XML документов и др.

Разбиение графов (graph partitioning) является одной из интересных задач интеллектуального анализа графов и определяется следующим образом. Пусть имеется граф $G = (N, E)$, где N — множество взвешенных вершин, E — множество взвешенных ребер, и целое положительное число p . Требуется найти непересекающиеся подграфы исходного графа N_1, N_2, \dots, N_p , такие, что

- $W(i) = W/p$, где $W(i)$ и W — суммы весов вершин N_i и N соответственно;
- сумма весов ребер, соединяющих подграфы, минимальна.

Далее в разделе 1 сформулирован многоуровневый подход к разбиению графов, в разделе 2 предложен способ реализации этого подхода с помощью параллельной СУБД.

1. Многоуровневое разбиение графов

Эффективное решение задачи разбиения графов имеет большое значение в ряде теоретических и практических задач. В качестве примера теоретических задач можно привести задачи раскраски графа, определения числа и состава компонент связности графа и представления графа в виде ярусно-параллельной формы. Примерами практических задач, в которых необходимо разбиение графа, являются проектирование сложных электронных схем, БИС (больших интегральных схем) и ПЛИС (программируемых логических интегральных схем), проектирование топологии локальной сети, конечно-элементное моделирование и др.

Для разбиения сверхбольших графов применяют так называемое *многоуровневое разбиение*, поскольку традиционные алгоритмы требуют недопустимо много времени и/или оперативной памяти. *Многоуровневое разбиение (multilevel partitioning)* предполагает три стадии этого процесса (рис. 1).

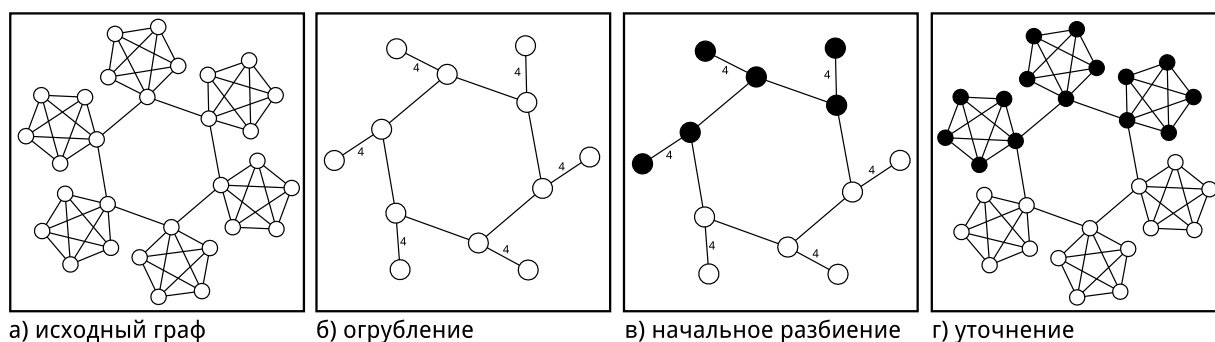


Рис. 1. Многоуровневое разбиение

На первой стадии выполняется «огрубление» (coarsening) графа, т.е. уменьшение количества вершин и ребер в нем путем «склеивания» вместе сильно связанных вершин и удаления возникающих при этом петель. На второй стадии выполняется начальное разбиение, когда огрубленный граф подвергают анализу обычными алгоритмами и получают грубое разбиение. Затем, на третьей стадии, выполняется «уточнение» (uncoarsening) грубого разбиения с помощью какой-либо эвристики.

Частным случаем задачи разбиения является бисекция графа ($p = 2$ — разбиение исходного графа на два подграфа). Разбиение на большее количество подграфов выполняется рекурсивно, т.е. каждый из найденных подграфов подвергается бисекции.

Процесс бисекции (рис. 2) начинается с присваивания вершинам произвольного «цвета», которым обозначается принадлежность вершины к тому или иному подграфу. После этого для каждой вершины производится подсчет показателя *выгоды (gain)* [4] для каждой вершины: насколько выгодно изменить цвет этой вершины на противоположный.

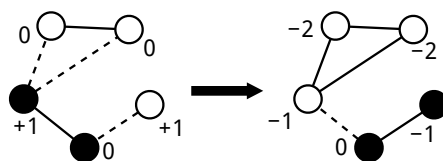


Рис. 2. Эвристика для уточнения

Выгода равна сумме весов всех ребер, соединяющих данную вершину с другими вершинами. Если ребро соединяет вершины разного цвета, то его вес увеличивает сумму, иначе — уменьшает. Затем вершины, цвет которых оказалось выгодно изменить, подвергаются «перекрашиванию». Процесс повторяется, пока такие вершины существуют.

2. Применение параллельной СУБД для разбиения графов

Существующие в настоящее время системы, использующие последовательные и параллельные алгоритмы разбиения графов [5], предполагают размещение графов и

промежуточных данных процесса обработки в оперативной памяти. Однако объем оперативной памяти современных компьютеров, хотя и велик, но не беспределен. При обработке сверхбольших графов (имеющих сотни тысяч вершин и/или ребер) упомянутые системы будут использовать жесткий диск для подкачки данных, что существенно замедлит решение задачи.

Реляционные системы управления базами данных (СУБД) на уровне программной архитектуры предполагают возможность эффективной реализации обработки данных, размер которых превышает объем доступной оперативной памяти [6, 7].

В соответствии с этим, реляционные СУБД могут быть применены для обработки графов: реализация алгоритмов выполняется в виде набора SQL запросов, а базовыми структурами хранения и обработки данных выступают реляционные таблицы и индексы. Для представления графа в виде реляционной таблицы используется список ребер.

Данный подход уже нашел свое применение в задачах поиска часто встречающихся подграфов [8] и поиска клики [9] в графах. Тем не менее, в случае обработки графов сверхбольших размеров, даже реляционная СУБД не может обеспечить надлежащую производительность.

Одним из решений данной проблемы является использование параллельной реляционной СУБД [10]. При обработке запросов к реляционной базе данных параллельная СУБД использует концепцию фрагментного параллелизма [11, 12].

Реляционные таблицы подвергаются горизонтальной фрагментации по дискам кластерной системы. На каждом узле кластера устанавливается модифицированное ядро последовательной СУБД, которое воспринимает базу данных как набор «своих» фрагментов таблиц. Запрос параллельно выполняется ядрами СУБД на всех узлах кластера, затем частичные результаты сливаются в результирующую таблицу. Модификация ядра последовательной СУБД заключается во внедрении в это ядро ряда механизмов параллельной обработки: автоматизированная фрагментация таблиц, распараллеливание запроса, координация действий ядер и обмена данными при выполнении запросов и др. Описанная идея реализована нами в параллельной СУБД PargreSQL [13], которая базируется на СУБД PostgreSQL, свободно распространяемой на уровне исходных кодов.

СУБД PargreSQL применена нами для решения задачи разбиения сверхбольших графов. Использование параллельной СУБД для решения данной задачи встречается, насколько нам известно, впервые. Схема решения задачи выглядит следующим образом (рис. 3). Сверхбольшой граф представляется в виде реляционной таблицы — списка ребер, распределяемой по узлам кластерной системы.

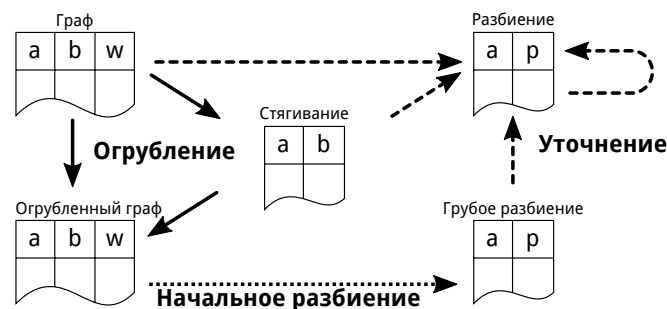


Рис. 3. Схема многоуровневого разбиения с помощью СУБД

На стадии огрубления параллельная СУБД выполняет запрос на языке SQL, результатом которого является реляционная таблица со списком несмежных ребер, имеющих наибольший суммарный вес. После этого СУБД выполняет запрос, который осуществляет стягивание найденных ребер. Получаемые в ходе стягивания петли уничтожаются, а кратные ребра преобразуются в одно ребро, имеющее вес, равный сумме весов кратных ребер. Процесс повторяется многократно, пока граф не перестанет быть сверхбольшим.

Полученный огрубленный граф может быть размещен целиком в оперативной памяти, что позволяет перейти к *стадии начального разбиения*. Огрубленный граф экспортируется из базы данных и подается на вход сторонней утилиты, которая выполняет начальное разбиение с помощью одного из традиционных алгоритмов. Результат начального разбиения импортируется в базу данных в виде реляционной таблицы, содержащей список вершин графа с указанием их цвета.

На стадии уточнения параллельная СУБД выполняет SQL-запрос, который отменяет стягивание ребер графа и «окрашивает» концы восстановленных ребер в цвет соответствующей вершины огрубленного графа. После этого СУБД выполняет запрос, который находит более выгодное разбиение путем «перекрашивания» вершин согласно некоторой эвристике (см. рис. 2). Процесс уточнения повторяется столько раз, сколько итераций производилось на стадии огрубления.

Итогом разбиения является реляционная таблица из двух столбцов: номер вершины графа и номер соответствующего подграфа.

Заключение

В статье предложен подход к использованию параллельной СУБД для реализации многоуровневого метода разбиения графов. В отличие от существующих последовательных и параллельных алгоритмов, предполагающих размещение данных в оперативной памяти, данный подход позволит разбивать сверхбольшие графы, в ходе анализа которых возникает очень много промежуточных данных.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта №12-07-31217 мол_а.

Литература

1. Ordonez, C. Bayesian Classifiers Programmed in SQL / C. Ordonez, S.K. Pitchaimalai // IEEE Transactions on Knowledge and Data Engineering. – 2010. – Vol. 22, № 1. – P. 139–144.
2. Ordonez, C. Integrating K-Means Clustering with a Relational DBMS Using SQL / C. Ordonez // IEEE Transactions on Knowledge and Data Engineering. – 2006. – Vol. 18, № 2. – P. 188–201.
3. Пан, К.С. Параллельный алгоритм решения задачи анализа рыночной корзины на процессорах Cell / К.С. Пан, М.Л. Цымблер // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». – 2010. – № 16(192). Вып. 5. – С. 48–57.
4. Kernighan, B.W. An Efficient Heuristic Procedure for Partitioning Graphs / B.W. Kernighan, S. Lin // The Bell system technical journal. – 1970. – Vol. 49, № 1. – P. 291–307.

5. Aggarwal, C.C. Managing and Mining Graph Data. / C.C. Aggarwal, H. Wang // Advances in Database Systems. – Springer, 2010. – Vol. 40. – 608 p.
6. Miniakhmetov, R.M. Integrating Fuzzy c-Means Clustering with PostgreSQL / R.M. Miniakhmetov // Труды Института системного программирования РАН. – 2011. – Т. 21. – С. 263–276.
7. Миниахметов, Р.М. Интеграция алгоритма кластеризации Fuzzy c-Means в PostgreSQL / Р.М. Миниахметов, М.Л. Цымблер // Вычислительные методы и программирование: Новые вычислительные технологии (Электронный научный журнал). – 2012. – Т. 13. – С. 46–52.
8. Chakravarthy, S. DB-FSG: An SQL-Based Approach for Frequent Subgraph Mining / S. Chakravarthy, S. Pradhan // Proceedings of the 19th International Conference on Database and Expert Systems Applications DEXA 2008 (Turin, Italy, September 1–5). – Springer, 2008. – P. 684–692.
9. Srihari, S. A Framework for SQL-Based Mining of Large Graphs on Relational Databases / S. Srihari, S. Chandrashekar, S. Parthasarathy // Advances in knowledge discovery and data mining. Lecture Notes in Computer Science. – 2010. – Vol. 6119. – P. 160–167.
10. Sokolinsky, L.B. Survey of Architectures of Parallel Database Systems / L.B. Sokolinsky // Programming and Computer Software – 2004. – Vol. 30. No. 6 – P. 337–346.
11. Лепихов, А.В. Обработка запросов в СУБД для кластерных систем / А.В. Лепихов, Л.Б. Соколинский // Программирование. – 2010. – № 4. – С. 25–39.
12. Lepikhov, A.V. Query Evaluation Techniques for Cluster Database Systems / A.V. Lepikhov, L.B. Sokolinsky // Proceedings of the 14th East European Conference on Advances in Databases and Information Systems (ADBIS 2010, Novi Sad, Serbia, September 20–24). Lecture Notes in Computer Science. – Springer, 2010. – Vol. 6295. – P. 351–362.
13. Пан, К.С. Разработка параллельной СУБД на основе последовательной СУБД PostgreSQL с открытым исходным кодом / К.С. Пан, М.Л. Цымблер // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». – 2012. – № 18(277). Вып. 12. – С. 112–120.

Пан Константин Сергеевич, аспирант, кафедра системного программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), kvaren@gmail.com.

APPLYING PARALLEL DBMS FOR VERY LARGE GRAPH MINING

C.S. Pan, South Ural State University (Chelyabinsk, Russian Federation)

Graph partitioning is an interesting topic in graph mining, that comes into use for some theoretical and practical problems (graph coloring, integrated circuit desing, finite element modeling, etc.). The existing serial and parallel algorithms suppose that the graph being analyzed can fit into main memory along with all the intermediate data, so they cannot be

applied for very large graphs. We introduce a new way of partitioning - using the parallel relational DBMS PargreSQL that is based on open-source PostgreSQL DBMS.

Keywords: data mining, graph partitioning, parallel DBMS.

References

1. Ordonez C., Pitchaimalai S.K. Bayesian Classifiers Programmed in SQL. *IEEE Transactions on Knowledge and Data Engineering*. 2010, Vol. 22, No. 1, P. 139–144.
2. Ordonez C. Integrating K-Means Clustering with a Relational DBMS Using SQL. *IEEE Transactions on Knowledge and Data Engineering*. 2006, Vol. 18, No. 2, P. 188–201.
3. Pan C.S. A Parallel Algorithm for Market Basket Analysis on Cell Processor. *SUSU Bulletin "Mathematical Modeling and Programming"*. 2010, No. 5, P. 48–57.
4. Kernighan B.W., Lin S. An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell system technical journal*. 1970, Vol. 49, № 1, P. 291–307.
5. Aggarwal C.C., Wang H. Managing and Mining Graph Data. *Advances in Database Systems*. Springer, 2010, Vol. 40, 608 p.
6. Miniakhmetov R.M. Integrating Fuzzy c-Means Clustering with PostgreSQL. *Proceedings of the Institute for System Programming of RAS*. 2011, Vol. 21, P. 263–276.
7. Miniakhmetov R.M., Zymbler M.L. Integracija algoritma klasterizacii Fuzzy c-Means v PostgreSQL [Embedding Fuzzy c-Means into PostgreSQL]. *Vychislitel'nye metody i programmirovaniye: Novye vychislitel'nye tehnologii (Jelektronnyj nauchnyj zhurnal)* [Computational Methods and Programming: New Computational Technologies (Digital Scientific Journal)]. 2012, Vol. 13, P. 46–52.
8. Chakravarthy S., Pradhan S. DB-FSG: An SQL-Based Approach for Frequent Subgraph Mining. *Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA 2008, Turin, Italy, September 1–5)*. Springer, 2008, P. 684–692.
9. Srihari S., Chandrashekar S., Parthasarathy S. A Framework for SQL-Based Mining of Large Graphs on Relational Databases. *Advances in knowledge discovery and data mining. Lecture Notes in Computer Science*. 2010, Vol. 6119, P. 160–167.
10. Sokolinsky L.B. Survey of Architectures of Parallel Database Systems. *Programming and Computer Software*. 2004, Vol. 30, No. 6, P. 337–346.
11. Lepikhov A.V., Sokolinsky L.B. Query Processing in a DBMS for Cluster Systems. *Programming and Computer Software*. 2010, Vol. 36, No. 4, P. 205–215.
12. Lepikhov A.V., Sokolinsky L.B. Query Evaluation Techniques for Cluster Database Systems. *Proceedings of the 14th East European Conference on Advances in Databases and Information Systems (ADBIS 2010, Novi Sad, Serbia, September 20–24)*. *Lecture Notes in Computer Science*. Springer, 2010, Vol. 6295, P. 351–362.
13. Pan C.S., Zymbler M.L. Development of a Parallel Database Management System on the Basis of Open-Source PostgreSQL DBMS. *SUSU Bulletin "Mathematical Modeling and Programming"*. 2012, Vol. 18, No. 12, P. 112–120.

Поступила в редакцию 16 октября 2012 г.