МОДЕЛЬ ПРЕДВЫЧИСЛЕНИЙ В ЗАДАЧАХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ МНОГОЯДЕРНЫХ УСКОРИТЕЛЕЙ

Р.М. Миниахметов

ФГБОУ ВПО «Южно-Уральский государственный университет» (НИУ)

Введение

В настоящее время, интеллектуальный анализ данных (ИАД) широко применяется для решения большого количества задач из самых разных областей науки и производства [1]. Однако, в связи с постоянным ростом объемов данных [2], которые потенциально пригодны для ИАД, остро стоит проблема ускорения алгоритмов для анализа больших объемов данных [3]. Для решения данной проблемы ИАД достаточно успешно используются такие общие подходы, как создание параллельных и распределенных алгоритмов [4], создание масштабируемых алгоритмов [5], а также оптимизация под современные аппаратные архитектуры многоядерных ускорителей [6, 7].

В данной работе рассмотрена модель предварительных вычислений для задач ИАД, которая объединяет наиболее эффективные подходы по решению проблемы анализа больших объемов данных. Целью разработки такой модели является уменьшение времени отклика существующих систем ИАД, а также оптимизация использования вычислительных и энергетических ресурсов при анализе больших объемов данных.

1. Анализ требований модели

Рассмотрим систему анализа данных (рис. 1), состоящую из хранилища данных D (или БД) и системы ИАД S. На вход S поступает пользовательский запрос q, который может быть представлен как вызовом функции с определенными параметрами (например, как в R [8]), так и потоком работ (как в KNIME [9] или RapidMiner [10]). Множество всех запросов пользователя составляет историю запросов Q. После получения запроса, система ИАД загружает данные из хранилища и выполняет их анализ, согласно запросу пользователя. В данной системе анализа время отклика на запрос пользователя (получение результата анализа r) составляет $t_{отклика} = t_{анализа} + t_{импорта}$.

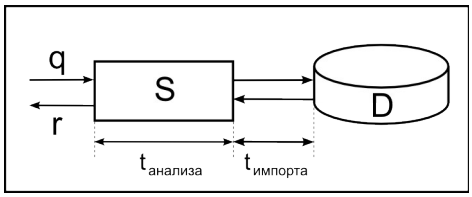

Рис. 1. Система анализа данных

Использование масштабируемых алгоритмов ИАД в данной ситуации позволит существенно снизить $t_{импорта}$, однако $t_{анализа}$ все еще может быть достаточно большим, в зависимости от алгоритма анализа и от объема анализируемых данных. При запуске масштабируемых алгоритмов ИАД на больших объемах (или потоковых) данных, для анализа будет постоянно загружаться ЦПУ и запуск нескольких таких алгоритмов с хорошим $t_{отклика}$ будет невозможным.

Параллельные и распределенные алгоритмы ИАД (GPUMiner [11], GPUMLib [12] и др.), использующие современные аппаратные платформы многоядерных ускорителей, помогают существенно снизить $t_{анализа}$, но без совместного использования с масштабируемыми алгоритмами повышают $t_{импорта}$ из-за небольшого объема основной памяти ГПУ. Кроме того, подобные алгоритмы пока еще слабо внедрены в наиболее используемые системы ИАД и не могут использоваться для ускорения уже существующих систем анализа.

Повторное использование результатов ИАД применяется в БД [13] и могут быть использованы при следующем запросе пользователя, а повторное использование запросов в системах с потоками работ [14] позволяет оптимизировать и стандартизировать запросы для предметной области пользователя. Данные подходы помогают снизить как $t_{импорта}$, так и $t_{анализа}$.

Таким образом, можно выделить следующие основные требования к модели:

1. Использование существующих систем ИАД в качестве базовых.
2. Использование результатов предвычислений в произвольный момент времени (момент получения запроса от пользователя).
3. Оптимизация предварительных вычислений с учетом предметной области пользователя.
4. Оптимизация алгоритмов анализа с учетом имеющихся аппаратных возможностей вычислительной системы.

2. Описание модели

Использование существующих систем ИАД возможно при использовании подхода предвычислений [1, 15, 16]. В рамках данного подхода необходимо разработать такие алгоритмы расчета промежуточных

(вспомогательных) значений для существующих алгоритмов ИАД, которые позволят существенно снизить $t_{анализа}$ и $t_{импорта}$.

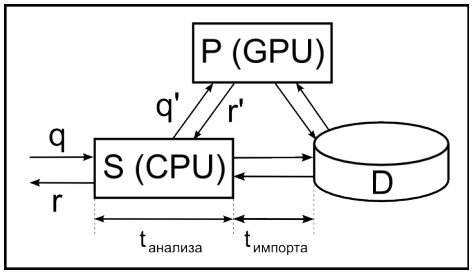

Рис. 2. Модель предвычисления ИАД

На рис. 2 представлена модель предвычислений, в который анализ данных производится следующим образом. На вход S поступает запрос пользователя q, далее система анализа сначала обращается (q') в подсистему предвычислений P в поиске вспомогательных данных для выполнения запроса q. Если такие данные получены (r'), то выполняется быстрый расчет по запросу q, иначе S выполняет запрос q «с нуля».

Настройка необходимых предварительных рассчетов может быть выполнена либо пользователем вручную, либо с использованием моделирования поведения пользователя на основе истории Q [17] или интеллектуального помощника [18]. Данная настройка позволит адаптировать работу системы анализа для данных конкретной предметной области.

Подсистема предвычислений может работать в двух режимах: повышения производительности — предварительные данные расчитываются постоянно при поступлении новых данных в хранилище, оптимизации простоя — предварительные данные рассчитывается только при отсутствии загрузки узла.

Для эффективного использования платформы многоядерных ускорителей в подсистеме предвычислений должны применяться такие структуры данных, как битовые карты [11], kd-деревья [19], и др.

ЛИТЕРАТУРА:

1. J. Han, M. Kamber, J. Pei Data Mining: Concepts and Techniques. Morgan Kaufmann, 2006. 743 p.
2. Press Release EMC2 // URL: http://www.emc.com/about/news/press/2012/20121211-01.htm (дата обращения: 30.05.2014).
3. W. Fan, A. Bifet Mining Big Data: Current Status, and Forecast to the Future // ACM SIGKDD Explorations Newsletter. Vol. 14, Iss. 2. P. 1-5.
4. S. Paul New Fundamental Technologies in Data Mining: Parallel and Distributed Data Mining. InTech, 2011. Ch. 3. P. 43-54.
5. A. Bifet Mining Big Data in Real Time // Informatica (Slovenia). Vol. 37, No. 1. P. 15-20.
6. C. Böhm, R. Noll, C. Plant, B. Wackersreuther, A. Zherdin Data Mining Using Graphics Processing Units // Lecture Notes in Computer Science. Vol. 5740. P. 63-90.
7. A. Gainaru, E. Slusanschi, S. Trausan-Matu Mapping Data Mining Algorithms on a GPU Architecture: A Study Lecture Notes in Computer Science. Vol. 6804. P. 102-112.
8. R Development Core Team (2013) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2013.
9. M.R. Berthold, N. Cebron, F. Dill, et al. KNIME: The Konstanz Information Miner // Proceedings of the 31st Annual Conference of the Gessellschaft fur Klassifikation. Springer, 2008. P. 319-326.
10. I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, T. Euler Yale: Rapid prototyping for complex data mining tasks // Proceedings of the 12th ACM SIGKDD International Conference on Knowledge discovery and data mining. ACM, 2006. P. 935-940.
11. W. Fang, K.K. Lau, M. Lu, X. Xiao, C.K. Lam, P.Y. Yang, B. He, Q. Luo, P.V. Sander, K. Yang Parallel Data Mining on Graphics Processors // Technical Report HKUST-CS08-07. Hong Kong Univ. of Science and Technology (HKUST), 2008.
12. N. Lopes, B. Ribeiro, R. Quintas GPUMLib: A New Library to Combine Machine Learning Algorithms with Graphics Processing Units // Proceedings of the 10th International Conference on Hybrid Intelligent Systems (HIS). Dynamic Publishers Inc., 2010. P. 355-362.
13. M. Zakrzewicz, M. Morzy, M. Wojciechowski A Study on Answering a Data Mining Query Using a Materialized View // Lecture Notes in Computer Science. Vol. 3280. P. 493-502.
14. S. Rüping, D. Wegener, P. Bremer Re-using Data Mining Workflows // Proceedings of the ECML PKDD 2010 Workshop on Third-Generation Data Mining: Towards Service-Oriented Knowledge Discovery (SoKD '10). P. 25-30.
15. N.U. Rehman, M.H. Scholl Enabling Decision Tree Classification in Database Systems through Pre-computation // Proceedings of the 27th British National Conference on Data Security and Security Data. Springer, 2012. P. 118-121.
16. G.K. Gupta Introduction to Data Mining with Case Studies. Prentice-Hall of India Pvt., 2011. 508 p.

17. M.-A. Aufaure, N. Kuchmann-Beauger, P. Marcel, S Rizzi, Y. Vanrompay Predicting Your Next OLAP Query Based on Recent Analytical Sessions // Proceedings of the Data Warehousing and Knowledge Discovery - 15th International Conference, DaWaK 2013. Springer, 2013. P. 134-145.
18. F. Serban, J. Vanschoren, J.-U. Kietz, A. Bernstein A Survey of Intelligent Assistants for Data Analysis // ACM Computing Surveys. Vol. 45, No. 3. P. 1-35.
19. M. Goldfarb, Y. Jo, M. Kulkarni General transformations for GPU execution of tree traversals // Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '13). ACM, 2013. P. 1-12.