УДК 004.62;65
МРНТИ 20.23.17, 20.23.21, 20.53.19

ТЕХНОЛОГИИ ОРГАНИЗАЦИИ ХРАНЕНИЯ И ЗАПРОСА БОЛЬШИХ КОЛЛЕКЦИЙ XML-ДОКУМЕНТОВ

А.А. Мухитова[1,2], А.С. Еримбетова[2,3], В. Барахнин[4]

[1]Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
[2]Институт информационных и вычислительных технологий МОН РК,Алматы, Казахстан
[3]Satbayev University, Алматы, Казахстан
[4]Федеральный Исследовательский Центр Информационных и Вычислительных технологий, РАН, 630090, Новосибирск, Россия
[1]ORCID ID: https://orcid.org/0000-0002-4081-7694, mukhitova.aigul@gmail.com
[2]ORCID ID: https://orcid.org/0000-0002-2013-1513, aigerian@mail.ru
[4]ORCID ID: https://orcid.org/0000-0003-3299-0507

Аннотация. В настоящее время веб-приложения, использующие XML репозитории (банковское дело, бухгалтерский учет, управление персоналом, резервирование авиакомпаний, мониторинг и прогнозирование погоды, электронное управление и электронная коммерция) обрабатывают огромное количество информации временного характера и нуждаются в полной истории изменения данных и схем для эффективного и прозрачного управления информационной системой. Существующие XML СУБД и XML-инструменты не имеют поддержки данной функции, в связи с чем возникает сложность построения истории изменений, сложность возврата к предыдущему состоянию, возможность потери данных. В следствие чего возникает проблема эффективного управления хранением временных данных при работе с большим количеством небольших XML-документов и множества дополнительных элементов/атрибутов.

Ключевые слова: схема кодирования XML, отображение XML-данных, база данных NoSQL, обработка и оптимизация XML-запросов.

Введение

XML был разработан в качестве стандартного языка и формата де-факто в промышленности для хранения и обмена данными между разнородными системами. Иерархическая природа XML поставила множество исследовательских задач перед сообществом баз данных [1]. Одна фундаментальная проблема заключается в эффективном хранении и запросе таких данных с древовидной структурой.

Чтобы запросить большую коллекцию XML-документов, аналитики данных обычно создают специальные программы для поиска во всех документах определенных тегов или местоположений, что обычно приводит к значительному количеству файловых операций ввода-вывода, что не показывает своей эффективности. Преобразование XML-документов в отношения и, следовательно, преобразование XML-запросов в SQL-запросы по этим отношениям является общепризнанным механизмом, но все еще существуют несоответствия между XML-структурированными данными и данными отношений [2-3]. Кроме того, такие методы не могут обеспечить удовлетворительную производительность и обычно не масштабируются по мере роста количества документов. Другим направлением усовершенствования является адаптация собственных баз данных XML, таких как BaseX, Sedna и XDB. Хотя эти базы данных имеют значительные преимущества с точки зрения удобства использования, но они не используют сложные возможности хранения и обработки запросов.

Системы запросов к коллекциям XML-документов, рассредоточенных по P2P-сетям и децентрализованные системы интеграции данных для XML (например, Piazza [4]) могут повысить производительность обработки XML-запросов за счет метода фрагментации. К сожалению, эти методы все же имеют некоторые недостатки. Во-первых, трудно осмысленно распределить XML-данные по узлам данных, поскольку распределенные файловые системы обычно не поддерживают формат XML. Во-вторых, поскольку

обработка XML-данных не является простой задачей, разработчикам приходится либо использовать сторонние библиотеки или инструменты, такие как Avro и Mahout, либо писать свои собственные интерфейсы. В-третьих, извлечение XML-данных из этих систем способом, подобным XQuery, невозможно без дополнительного уровня управления данными поверх распределенной файловой системы.

Существует множество языков и систем обработки данных, разработанных для масштабируемых архитектур, таких как MapReduce и Dryad для облачных вычислений. В частности, Pig, Hive, Jaql и Scope можно использовать для хранения XML-документов и запросов к ним. Однако, как показал результат проведенного нами эксперимента, многие системы не оптимизированы, в результате чего возникают проблемы производительности, включая высокие требования к хранилищу, длительное время загрузки и длительное время выполнения запросов.

Способы хранения и запроса XML-документов

XML в реляционных базах данных

Предпринимаются значительные усилия по хранению и запросу XML-документов с использованием реляционных баз данных. Ключевой проблемой является сопоставление вложенных элементов с плоскими таблицами [5]. В целом существующие методы можно классифицировать как отображение нормализации и кодирование узлов.

Предыдущие результаты показали, что отображение нормализации способно обеспечить хорошую эффективность запросов. XML-схема и нормализация данных являются двумя краеугольными камнями для обеспечения такой производительности. С другой стороны, отображение с кодировкой узлов обеспечивает более общее решение - все элементы и атрибуты XML представлены числовыми кодами местоположений узлов в XML-дереве, например Dewey Order [6], Ordpath [7]. Иерархические отношения и порядок документа неявно фиксируются структурным соединением кодировок элементов. При сопоставлении с кодировкой узлов таблицы данных всегда плотные, а элементы из одного XML-документа хранятся последовательно. Из-за отсутствия информации о схеме обработки запросов с сопоставлениями кодирования узлов, как правило, такие разработки очень затратны.

Во временной настройке данные XML могут развиваться по времени транзакции и/или допустимого времени. Таким образом, они могут иметь время транзакции, достоверное время или временный формат. Когда данные XML различных временных форматов могут сосуществовать в одном и том же хранилище XML, именуемый многопользовательским репозиторием XML.

Собственные базы данных XML

Системы баз данных Native XML (NXD) создаются с нуля для хранения XML-документов и запросов к ним: Berkeley DB XML, Apache Xindice, eXist, dbXMKL, Sedna, BaseX, xDB и OrientX. Эти системы изначально хранят XML-данные с использованием трех основных технологий хранения, а именно методов на основе текста, моделей и схем [8].

База данных	***Используемая технология хранения XML-документов***
Berkeley DB XML	база данных типа «ключ-значение», хранит XML-документы в их родном формате, основанном на текстовом подходе к хранению
Xindice, eXist, dbXML	хранит XML-документы в коллекциях как логические единицы хранения, используя подход к хранению на основе моделей
Sedna	использует блоки/наборы данных в качестве логических единиц хранения и хранит XML-документы, используя подход кластерного хранения на основе схемы

BaseX	использует табличное представление древовидных структур XML для хранения XML-документов
EMC Documentum xDB	хранит XML-документы в интегрированной объектно-ориентированной базе данных
OrientX	упорядочивает XML-документы в соответствии с их схемой и сохраняет их в наборах данных как логические единицы хранения на основе кластерного подхода к хранению с использованием этой схемы

Собственные базы данных XML имеют лучшую производительность по сравнению с подходами «XML на реляционных базах данных». Однако эти системы имеют два потенциальных ограничения. Во-первых, размер как данных, так и индекса, потребляемых собственной базой данных XML, намного больше, чем в собственных базах данных XML [9]. Во-вторых, собственные системы баз данных XML не используют сложные возможности хранения и запросов, уже предоставляемые существующими системами баз данных [10].

XML в распределенных системах хранения

Развитие одноранговой (P2P) обмена информацией проложило путь к поддержке приложений управления XML-данными в распределенной среде. В некоторых более ранних работах изучалось совместное использование данных P2P в неструктурированных сетях, включая Piazza [11] и PeerDB [12]. Обе они являются одноранговыми системами управления данными, которые позволяют совместно использовать разнородные данные распределенным и масштабируемым образом. Основная проблема этих систем заключается в том, что запросы могут передаваться всем одноранговым узлам, что значительно увеличивает сетевой трафик.

Разработанная платформа KadoP опирается на известную технологию распределенных хэш-таблиц (DHT) для поддержки сложных запросов к общим XML-данным [13]. Авторы [14] разработали методы для оценки XPath-запросов к XML-дереву, которое разделено по горизонтали и вертикали и распределено по нескольким сайтам. Однако этот подход имеет дело только с логическими запросами XPath и запросами XPath с выбором данных.

XML в облачных системах хранения

Многие системы, языки и модели данных были разработаны для обработки массивных наборов данных поверх распределенных файловых систем (например, HDFS) и в рамках MapReduce. Типичные примеры включают Jaql, Pig, Hive, Scope, HadoopDB, HadoopXML и ChuQL. Pig — это динамически типизированный язык запросов с гибкой вложенной реляционной моделью данных и удобным синтаксисом, подобным скрипту, для разработки потоков данных, которые оцениваются с помощью MapReduce. Hive использует гибкую модель данных и MapReduce с синтаксисом, основанным на SQL. Scope от Microsoft имеет те же функции, что и Hive, за исключением того, что он использует Dryad в качестве параллельной среды выполнения. HadoopDB является расширением Hive и запрашивается с помощью языка запросов, подобного SQL. Вместо хранения данных в распределенной файловой системе Hadoop он управляет несколькими экземплярами Postgres для хранения данных. Он теряет универсальность Hadoop, который поддерживает произвольные модели данных, но улучшает уровень хранения Hadoop, приближая части запроса к данным. HadoopXML — это система, которая одновременно обрабатывает множество запросов шаблонов веток для большого объема XML-данных с помощью Hadoop. ChuQL — это расширение MapReduce для XQuery для обработки XML с помощью Hadoop.

Были разработаны некоторые новые платформы, предлагающие более общие операторы, включая Hyracks и Nephele. Hyracks предоставляет разделенную параллельную модель для выполнения вычислений с интенсивным использованием данных в кластерах без совместного использования. Nephele — это универсальная система для обработки данных в веб-масштабе. Абстракцией программирования для написания

задач являются контракты параллелизации (Parallelization Contracts - PACTs), состоящие из контрактов ввода/вывода. Камачо-Родригес и др. [15] представили подход PAXQuery для неявного распараллеливания XQuery посредством преобразования алгебраического плана XQuery в параллельный план PACT.

Другое направление — отображение XML-документов в различные базы данных NoSQL. Все хранилища NoSQL созданы на основе декларативной обработки запросов с использованием XQuery и используют стратегии отображения модели данных, таких как ключ/значение, столбцы и ориентированные на документы, в модель данных XDM.

Камачо-Родригес и др. [16] описали механизм распределенных запросов для управления большими объемами XML-документов поверх Amazon Cloud. Представлены три стратегии индексирования и реализовано подмножество XPath и XQuery. В исследовании, проведенном Wen-Syan et al. [17], облачное информационное устройство под названием Xbase предлагается специально для крупномасштабных и сложных медицинских приложений. Xbase применяет гибридный подход к распределенным репозиториям XML-документов на основе RDBMS и Hadoop HDFS, что обеспечивает более высокую производительность обработки запросов, а также снижает затраты на проектирование системы.

Шенк и др. [18] разработали постоянное хранилище под названием Xomoda, которое отображает XML-данные в хранилище с широкими столбцами. В Xomoda XML-документы разбиты на узлы, и эти узлы индексируются двумя разными индексами. Xomoda имеет некоторые общие черты с XML2HBase, но отличается двумя важными аспектами. Во-первых, Xomoda фокусируется только на запросах осей XML, и авторы не обсуждали другие операции с данными XML. Во-вторых, Xomoda хранит всю информацию (включая индекс пути, имя, тип, значение и количество одноуровневых узлов) каждого узла в XML-дереве в HBase, что в свою очередь создает огромную таблицу данных и проблему с производительностью.

Анализ и выводы

Все вышеизложенные подходы были предложены с целью хранения и извлечения разнородных данных на основе XML. Основная стратегия запросов, используемая для реляционных решений, заключается в извлечении табличных данных с использованием традиционного декларативного языка запросов: XPath или XQuery. Но их использование для обеспечения возможностей улучшения скорости обработки и масштабирования является малоэффективной.

На этом фоне большие преимущества в распределенной и параллельной обработке имеют базы данных NoSQL. Они широко используются для крупномасштабного хранения больших данных и высокопроизводительных вычислений.

Облачные системы предназначены для обеспечения достаточного объема памяти и компьютерных ресурсов для управления и совместного использования больших наборов разнородных данных. Это позволяет предотвратить неоправданную избыточность данных в сети на основе сервис-ориентированной архитектуры и соответствующих протоколов. Интеграция данных поддерживает операции с композицией и временным (виртуальным) представлением данных, хранящихся у различных владельцев. Данные остаются под контролем владельца и извлекаются по требованию клиентов для интегрированного доступа, возможно на коммерческой или свободной основе.

Заключение

В дальнейшем, в проводимом нами исследовании по внедрению адаптивного графического редактора XML-записей в распределенной инфомационной системе, предполагается добавление в систему версионирования документов, и, как следствие, добавление в приложение редактирования гетерогенных XML-документов поддержки версионирования. Также планируется добавление поддержки темпоральных (временных) документов, с разными версиями одного и того же элемента в документе, а также мульти-

схемных документов.

Список литературы

[1] Nassiri H., Machkour M., Hachimi M. Integrating xml and relational data. Proc.Comput. Sci. 2017. 110 422–427.

[2] Brahmia, Z., Hamrouni, H., Bouaziz, R. XML data manipulation in conventional and temporal XML databases: A survey. Computer Science Review Volume 36, 1 May 2020, №100231.

[3] A.V. Lyamin, E.N. Cherepovskaya. “Xml-relational mapping using production rule system.” Intelligent Systems Conference, IntelliSys 2017, IEEE, 2017, 422–429.

[4] A.Y. Halevy, Z.G. Ives, J. Madhavan, P. Mork, D. Suciu, I. Tatarinov. “The Piazza peer data management system.” IEEE Trans. Knowl. Data Eng. 16 (7) (2004), 787–798.

[5] L.J. Chen, P.A. Bernstein, P. Carlin, D. Filipovic, M. Rys, N. Shamgunov, J.F. Terwilliger, M. Todic, S. Tomasevic, D. Tomic. “Mapping xml to a wide sparse table.” IEEE Trans. Knowl. Data Eng. 26 (6) (2014),1400–1414.

[6] I. Tatarinov, S.D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, C. Zhang. “Storing and querying ordered xml using a relational database system.” Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, ACM, 2002, 204–215.

[7] P. O’Neil, E. O’Neil, S. Pal, I. Cseri, G. Schaller, N. Westbury. “Ordpaths: insert-friendly xml node labels.” Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, ACM, 2004, 903–908.

[8] M. Marjani, F. Nasaruddin, A. Gani, S. Shamshirband. “Measuring transaction performance based on storage approaches of native xml database.” Measurement 114 (2018), 91–101.

[9] S. Balamurugan, A. Ayyasamy. “Performance evaluation of native xml database and xml enabled database.” Int. J. Adv. Res. Comput. Sci. Softw. Eng. 7 (5) (2017).

[10] A.M. Saba, E. Shahab, H. Abdolrahimpour, M. Hakimi, A. Moazzam. “A comparative analysis of xml documents, xml enabled databases and native xml databases.” Preprint, arXiv:1707.08259, 2017.

[11] A.Y. Halevy, Z.G. Ives, J. Madhavan, P. Mork, D. Suciu, I. Tatarinov. “The Piazza peer data management system.” IEEE Trans. Knowl. Data Eng. 16 (7) (2004), 787–798.

[12] W.S. Ng, B.C. Ooi, K.-L. Tan, A. Zhou. “Peerdb: a p2p-based system for distributed data sharing.” 19th International Conference on Data Engineering, 2003, Proceedings, IEEE, 2003, 633–644.

[13] S. Abiteboul, I. Manolescu, N. Polyzotis, N. Preda, C. Sun. “Xml processing in dht networks.” 2008 IEEE 24th International Conference on Data Engineering, IEEE, 2008, 606–615.

[14] G. Cong, W. Fan, A. Kementsietsidis, J. Li, X. Liu. “Partial evaluation for distributed xpath query processing and beyond.” ACM Trans. Database Syst. 37 (4) (2012), 32-43.

[15] J. Camacho-Rodríguez, D. Colazzo, I. Manolescu Paxquery. “Efficient parallel processing of complex xquery.” IEEE Trans. Knowl. Data Eng. 27 (7) (2015), 1977–1991.

[16] J. Camacho-Rodríguez, D. Colazzo, I. Manolescu. “Building large xml stores in the Amazon cloud.” 2012 IEEE 28th International Conference on Data Engineering Workshops, ICDEW, IEEE, 2012, 151–158.

[17] W.-S. Li, J. Yan, Y. Yan, J. Zhang. “Xbase: cloud-enabled information appliance for healthcare.” Proceedings of the 13th International Conference on Extending Database Technology, EDBT, ACM, 2010, 675–680.

[18] A. Šenk, M. Valenta, W. Benn. “Distributed evaluation of xpath axes queries over large xml documents stored in mapreduce clusters.” 2014 25th International Workshop on Database and Expert Systems Applications, IEEE, 2014, pp. 253–257.

TECHNOLOGIES FOR STORING AND QUERYING LARGE COLLECTIONS OF XML DOCUMENTS

A.Mukhitova[1,2], A. Yerimbetova[2,3] , V. Barakhnin[4]

[1]al-Farabi Kazakh National University, Almaty, Kazakhstan

[2]Institute of Information and Computational Technologies, Almaty, Kazakhstan

[3]Satbayev University, Almaty, Kazakhstan

[4]Federal Research Center for Information and Computational Technologies, Ac., 630090 Novosibirsk, Russia

[1]ORCID ID: https://orcid.org/0000-0002-4081-7694, mukhitova.aigul@gmail.com

[2]ORCID ID: https://orcid.org/0000-0002-2013-1513, aigerian@mail.ru

[4]ORCID ID: https://orcid.org/0000-0003-3299-0507

Abstract. Currently, web applications using XML repositories (banking, accounting, personnel

management, airline reservations, weather monitoring and forecasting, electronic control and e-commerce) process a huge amount of temporary information and need a complete history of data and schema changes for effective and transparent management of the information system. Existing XML DBMS and XML tools do not support this function, which makes it difficult to build a history of changes, difficult to return to the previous state, and the possibility of data loss. As a result, there is a problem of effective management of temporary data storage when working with a large number of small XML documents and many additional elements /attributes.

Keywords: XML encoding scheme, XML data mapping, NoSQL database, XML query processing and optimization

XML ҚҰЖАТТАРДЫҢ ҮЛКЕН КОЛЛЕКЦИЯСЫНА СҰРАНЫСТАР МЕН САҚТАУДЫ ҰЙЫМДАСТЫРУ ТЕХНОЛОГИЯЛАРЫ

А.Мухитова[1,2], А.Еримбетова[2,3], В. Барахнин[4]
[1]Ақпараттық және есептеуіш технологиялар институты ҒК БҒМ, Алматы, Қазақстан
[2]әл-Фараби атындағы Қазақ Ұлттық Университеті, Алматы, Қазақстан
[3]Satbayev University, Алматы, Қазақстан
[4]Ақпараттық және Есептеу Технологиялары бойынша Федералды Зерттеу Орталығы, РҒА, 630090, Новосибирск, Ресей

Аңдатпа. Қазіргі уақытта XML репозиторийлерін қолданатын веб-қосымшалар (банк ісі, бухгалтерлік есеп, адам ресурстарын басқару, авиакомпанияларды брондау, ауа райын бақылау және болжау, электрондық үкімет және электрондық коммерция) үлкен көлемдегі уақытша ақпаратты өңдейді және ақпараттық жүйені тиімді және ашық басқару мақсатында деректер мен схема өзгерістерінің толық тарихын қажет етеді. Қолданыстағы XML ДҚБЖ және XML құралдары бұл функцияны қолдамайды, бұл өзгерістер тарихын құруды қиындатады, алдыңғы күйге оралуды қиындатады және деректердің жоғалу мүмкіндігін тудырады. Нәтижесінде, уақытша мәліметтерді сақтауды тиімді басқару мәселесі көптеген шағын XML құжаттарымен және көптеген қосымша элементтермен/атрибуттармен жұмыс істегенде туындайды.

Кілттік сөздер: XML кодтау сызбасы, XML деректерін салыстыру, NoSQL деректер қоры, XML сұраныстарын өңдеу және оңтайландыру.

Сведения об авторах

Қаз: Мухитова Айгуль - Әл-Фараби атындағы Қазақ ұлттық университетінің докторанты, mukhitova.aigul@gmail.com

Рус: Мухитова Айгуль - докторант Казахского национального университета им. аль-Фараби, mukhitova.aigul@gmail.com

Англ: Mukhitova Aigul – a doctoral student at Al-Farabi Kazakh National University, mukhitova.aigul@gmail.com

Қаз: Еримбетова Айгерим Сембековна – Ақпараттық және есептеуіш технологиялар институтының аға қызметкері, Satbayev University профессоры, PhD, т.ғ.к., қауымд. профессор, aigerian@mail.ru

Рус: Еримбетова Айгерим Сембековна – старший научный сотрудник Института информационных и вычислительных технологий, профессор Satbayev University, PhD, к.т.н., ассоц. профессор, aigerian@mail.ru

Англ: Yerimbetova Aigerim Sembekovna – Senior Researcher of Institute of Information and Computing Technologies, Professor of Satbayev University, PhD, Candidate of Technical Sciences, Assoc. Professor, aigerian@mail.ru

Қаз: Владимир Барахнин - Ақпараттық және Есептеу Технологиялары бойынша Федералды Зерттеу Орталығының профессоры, т. ғ. д. РҒА, 630090, Новосибирск, Ресей

Рус: Владимир Барахнин – д.т.н., профессор Федерального Исследовательского Центра Информационных и Вычислительных технологий, РАН, 630090, Новосибирск, Россия

Англ: Vladimir Barakhnin - Professor, Doctor of Engineering Sciences of Federal Research Center for Information and Computational Technologies, Ac., 630090 Novosibirsk, Russia.