Инженерный вестник Дона, №12 (2022)
ivdon.ru/ru/magazine/archive/n12y2022/8075

Алгоритмы безопасности гетерогенных хранилищ больших данных
ОТОЗВАНА/RETRACTED 20.02.2023 г.

С.В. Сафонов

Московский технический университет связи и информатики, г. Москва
Агентство ВПС-Мониторинг, г. Москва

Аннотация: В данной статье в результате анализа было выявлено, что в гетерогенных хранилищах больших данных должен быть применен атрибутивный контроль доступа. Затем был сформулирован алгоритм переноса прав между различными моделями данных, входящими в состав гетерогенных хранилищ больших данных. Также выявлены ограничения, накладываемые на данный алгоритм. Таким образом, задача реализации способа управления правами в гетерогенных хранилищах больших данных была выполнена.
Ключевые слова: система управления базами данных, гетерогенные хранилища, большие данные, реляционные модели баз данных.

На данный момент наиболее актуальными задачами в безопасности больших данных являются: хранение, верификация и их аудит [1,2]. Для целей обеспечения защиты и обработки больших данных могут быть применены технологии распределенного реестра, а также технология BlockChain. [3]. В данной сфере уже идут работы [4-6] для решения похожих задач, но все приведенные исследования, которые рассматривают решения, связанные с обеспечением контроля доступа в гетерогенных хранилищах больших данных, обычно привязаны к конкретным решениям, например платформе Hadoop. В связи с этим, становится актуальна новая задача, которая нацелена на разработку более универсального решения, в котором все компоненты являются взаимозаменяемыми и способны варьироваться в зависимости от ситуации. Более того, такой подход позволяет проанализировать и сравнить эффективности разных технологий распределенного реестра, а также создать систему, которая способна гибко контролировать движения данных внутри гетерогенных хранилищ больших данных.

Ввиду того, что в настоящее время процесс эволюции баз данных пришел к обработке больших данных, то архитектурой для решения задач высокой нагрузки являются гетерогенные хранилища больших данных.
Главная задача в обработке информации в гетерогенных хранилищах больших данных: передача, хранение и последовательное преобразование этих фрагментов.

Гетерогенные хранилища больших данных обладают главным свойством – при выполнении последовательности или отдельных операций над массивом информации, состав узлов-обработчиков остается изменяемым, а не постоянным. Хранилища не поддерживают темпоральность данных. [7] Если входной поток, выходной поток или внутреннее хранилище всегда является предустановленным источником для каждой конкретной операции на уровне управления и обработки, то множество исполнителей или один исполнитель определенного действия над элементом конкретных данных не задан и выбирается случайно из доступного массива [1].

Гетерогенные хранилища больших данных включают в себя компоненты, основанные на различных моделях данных и актуальной задачей является согласование прав на эту информацию между компонентами, основанными на различных моделях данных.

Выделим 3 основные задачи в обработке информации в гетерогенных хранилищах больших данных: передача, хранение, а также преобразование определенных фрагментов данных, которое должно проводиться в соответствии с их типом и определенным алгоритмом. Хранение подразумевает под собой приостановку, передачу и обработку данных в процессе управления ими и ожидание наступления какого-то определенного события. Например, накопления нужного объема данных, истечения заданного временного промежутка, поступления нового запроса или других событий. Мы принимаем за аксиому, что в обычной ситуации нерационально

хранить данные, которые никогда не будут востребованы или использованы в будущем. Из этого следует, что данные, которые в настоящий момент не задействованы при выполнении процедур обработки и управления, должны быть задействованы в будущем, или уже были использованы ранее. Это базовая концепция современных СУБ (система управления данными), которая включает в себя средства обработки и хранения информации [1].

Для того, чтобы определить алгоритм согласования, необходимо проанализировать операции над данными, выполняемыми в процессе обработки. На рисунке 1. показано преобразование в процессе обработки, выполняемые разными узлами. За di обозначен сегмент данных, а за Op– преобразование, которое выполняется узлом в процессе обработки.

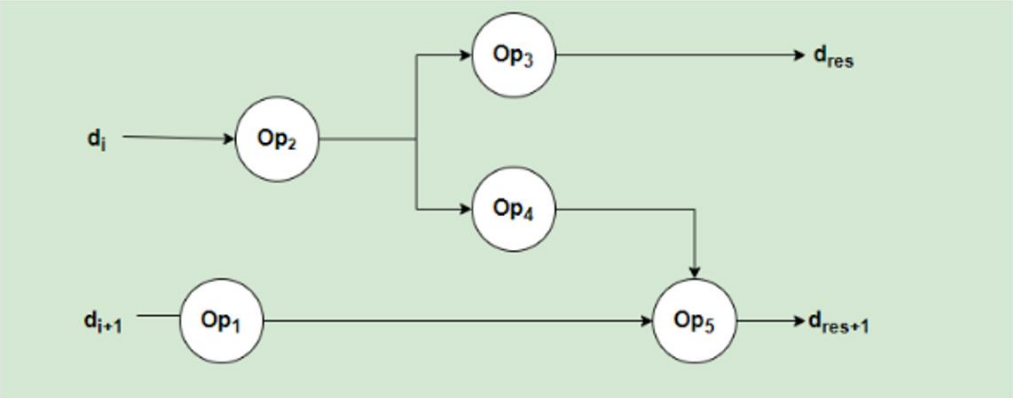

Рисунок 1. –Процесс обработки данных

Тогда из рисунка видно, что основным преобразованием над сущностями является грануляция. Данные не изменяются, а лишь делятся на меньшие блоки, либо сливаются в большие, в процессе их обработки гетерогенными хранилищами больших данных.

Теперь сформулируем основные операции, выполняемые с данными в процессе их обработки.

1. Разделение фрагментов. Получение n новых сегментов данных в результате операции над ними. (Рисунок 2)

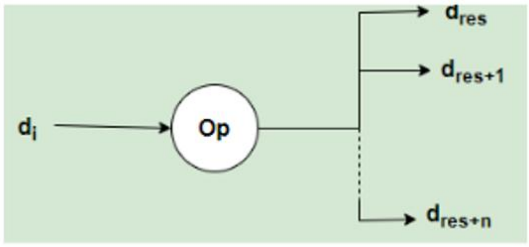

Рисунок 2 – Вариант грануляции данных

2. Агрегация. Операции слияния нескольких сегментов данных в один, как показано на рисунке 3.

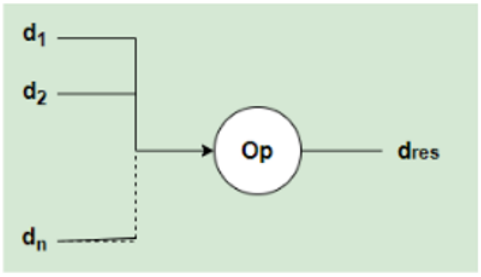

Рисунок 3 – Вариант агрегации данных

3. Трансформация. В процессе преобразования данных для изменения их грануляции предлагается использование функций Map или MapReduce. Обе функции получают на вход набор сущностей и набор отображений ($M1$, $M2$, ..., Mn), где Mi: E×...×E→E.

С помощью функции Map возможно обеспечить трансформацию данных, на выходе данной функции возвращается преобразованная сущность Iti из полученной на вход сущности, относящейся к другой модели Isi. Функционирование функции Map можно представить в виде схемы, изображенной на рисунке 4.

В свою очередь функция MapReduce позволяет обеспечить трансформацию данных, таким образом, чтобы получить сущность, преобразованную Iti из набора, полученного на вход, относящейся к $Is1$,$Is2$,...,Isn. Необходимо обратить внимание на то, что для группировки

исходных сущностей необходимо указать ключ в качестве параметра. Схема функционирования функции MapReduce представлена на рисунке 5.

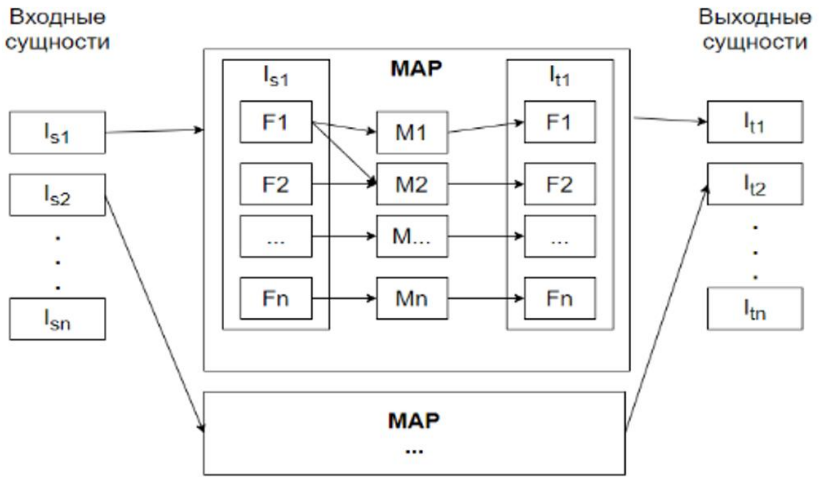

Рисунок 4–Пример преобразования с помощью Map

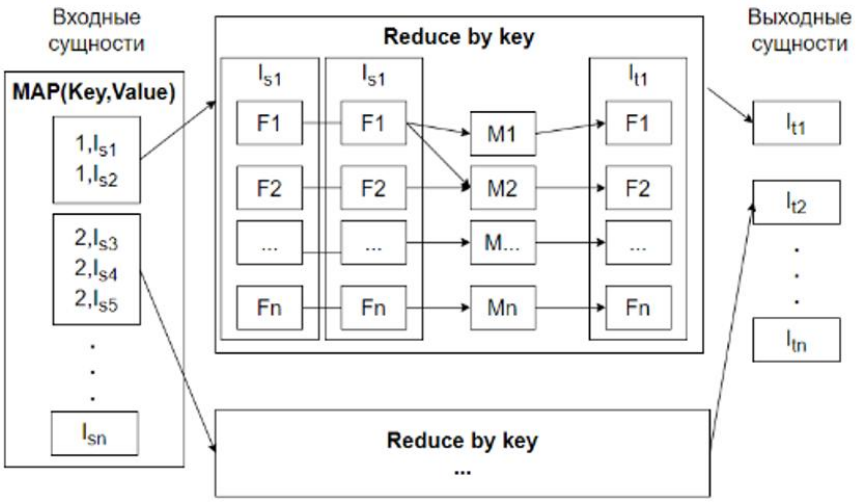

Рисунок 5. –Пример преобразования с помощью MapReduce

Основываясь на работах, связанных с переносом информации между различными моделями данных [8-10], можно сделать вывод о том, что преобразования сводятся к выполнению грануляции таким образом, чтобы представить данные в соответствии с необходимой моделью. Исходя из этого, видна схожесть операций, происходящих при переносе информации и в процессе их обработки гетерогенными хранилищами больших данных.

Таким образом, в рамках данной работы необходимо реализовать алгоритм согласования прав между компонентами, основанными на различных моделях данных. Данный алгоритм необходим для того, чтобы все операции с гетерогенным хранилищем больших данных были выполнены пользователями, имеющими на это необходимые права.

В ходе анализа решения проблем переноса прав была выбрана модель безопасности, основанная на атрибутивном контроле доступа. В рамках гетерогенных хранилищ данных можно представить всю систему в виде совокупности активных сущностей, то есть вычислительных узлов сети – субъектов (S), а также в виде пассивных сущностей (сегментов данных) – объектов (O), также в виде множества атрибутов субъектов (SA), и объектов (OA), и в виде некоторого набора правил (P) по которому проверяется, имеет ли субъект доступ к объекту. Однако необходимо учитывать специфику выполняемой операции над данными для определения прав на доступ к новым объектам, полученным на основе существующих в системе. Необходимо ввести отображение ϕ такое, что:

ϕ: $OAoi1\times OAoi2\times...\times OAoin\rightarrow OAoN+1\times OAoN+2\times...\times OAoN+z$,

где $OAOk-$атрибуты объекта $ok\in O$, $N-$общее количество объектов, $n-$количество входных объектов, $z-$количество выходных объектов.

Пусть в системе используются два вида прав:

−право на чтение (R);

−право на запись (W).

В качестве отображение ϕ предлагается использовать минимальное возможный набор атрибутов по некоторым правилам:

$$\phi(aOi1,aOi2,,...,aOin)=\min(aOi1,aOi2,,...,aOin),$$

Где $aoij$ − набор атрибутов объекта oij.

Данный метод является наиболее подходящим ввиду того, что на каждом последующем шаге не может быть получена привилегия выше, чем на

предыдущем, однако в то же время не может быть получена привилегия ниже минимальной существующей. Также необходимо разработать правила, которые будут обеспечивать выдачу необходимых для дальнейшего функционирования системы атрибутов.

В качестве алгоритма распространения событий, реализован алгоритм Gossip, позволяющий быстро распространять событие между узлами. Из-за того, что в тестируемой схеме всего 3 узла, все узлы будут принимать участие в принятии решения об истинности события. В стандартном случае наоборот, чаще задействуются не все узлы в сети. Суть алгоритма Gossip состоит в том, что созданное событие распространяется от одной вершины к случайно выбранной другой, таким образом распространение события имеет экспоненциальную скорость.

Рассмотрим алгоритм работы технологии HashGraph на примере. Пусть в системе имеется N узлов. Узел 1 хочет распространить событие 1. 1 посылает его случайно выбранному узлу и тот рассылает событие 1 другому случайно выбранному узлу, и это продолжается до тех пор, пока все узлы не будут осведомлены о событии 1. Далее другие два узла хотят послать новые события 2 и 3, и информация распространяется по системе таким же образом, как и в случае с событием 1. Но так как события 1 и 2 были отправлены одновременно, то вычисляется среднее значение временной метки всех событий системы, и порядок событий 2 и 3 считается из этих значений. Чтобы получить информацию об истории рассылки событий дополнительно пересылается информация о событиях, полученных каждым узлом, и, в конечном итоге, у каждого узла имеется граф пересылки событий от узла к узлу.

Пример графа представлен на рисунке 6 и при получении информации по графу можно определять, является ли сообщение истинным или ложным. Пример работы программы представлен на рисунке 7.

Права доступа к данным могут быть заданы нестандартным образом для разных моделей данных, поэтому мы будет рассматривать только основные права на чтение и запись.

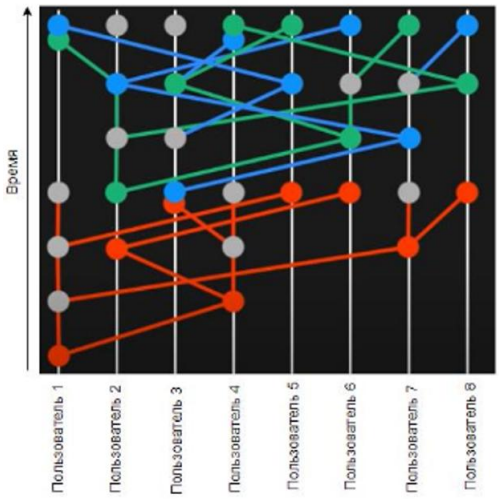

Рисунок 6. – Пример графа, хранимого на каждом узле

```
New_event
Transactions:[{'creator_id': '0.0.0.1', 'receiver_id': '0.0.0.1', 'trans_id': 1, 'segment_id': 2, 'source_sigments': [1],
working node: 1, event number: 0
working node: 1, event number: 0
New_event
Transactions:[{'creator_id': '0.0.0.1', 'receiver_id': '0.0.0.2', 'trans_id': 2, 'segment_id': 2, 'source_sigments': [2],
working node: 1, event number: 1
working node: 1, event number: 1
```

Рисунок 7. – Пример работы программы, основанной на технологии HashGraph

Во всех моделях данных можно представить права доступа к базе данных с помощью универсального математического объекта матрицы доступа. Тогда алгоритм для миграции прав в гетерогенных системах включает в себя следующие этапы:

1. Получение данных о правах пользователей СУБД.
2. Создание матрицы доступа.
3. Перенос данных для различных моделей данных:

3.1 Для переноса в документо-ориентированную модель данных [7,8].

3.2 Для переноса в модель данных семейство столбцов [8].

3.3 Для переноса в модель данных “ключ-значение” [10].

4. Преобразование строк матрицы доступа в правила для:

4.1 Создание пользователей.

4.2 Создание атрибутов.

4.3 Добавление прав на объекты с определенными атрибутами.

4.4 Выстраивание взаимосвязей между атрибутами и пользователями.

Данный алгоритм можно представить в виде схемы, изображенной на рисунке 8.

Рисунок 8. –Процесс миграции прав

Введем ограничения, накладываемые на данный алгоритм:

1. Различные СУБД, входящие в состав гетерогенных хранилищ данных, построены на разных моделях данных и права в таких СУБД выдаются на разные объекты, поэтому необходимо понимать грануляцию данных, чтобы правильно выдавать необходимые права.
2. В различных СУБД возможна более гибкая настройка прав доступа, а в некоторых таких возможностей не предоставляется. Например, в большинстве реляционных СУБД можно обеспечить контроль доступа к атрибутам, а в реализациях не реляционных моделей регулируется доступ только к коллекциям (в документно-

ориентированных СУБД, СУБД “семейства столбцов” и СУБД “ключ -значение”).

Поэтому необходимо выявлять и определять грануляцию различных объектов СУБД, входящих в состав гетерогенных хранилищ. В результате анализа были выявлены следующие преимущества и недостатки метода, основанного на работах [8-10].

Преимущества:

− возможность функционирования гетерогенных хранилищ в составе единой системы;

− корректная работа для основных прав, которые существуют во всех реализациях.

Недостатки:

− перенос прав между объектами, связанными типом связи многие ко многим, требует специального применения и могут возникать ситуации неправильной трансформации и потери данных, а, значит, и прав при переносе.

В ходе работы выявлено, что решение применимо в таких гетерогенных системах, где компоненты и модули определены заранее, так как новые могут иметь иную структуру прав доступа.

Также для миграции подходят только те права, которые поддерживаются всеми компонентами системы, иначе они будут потеряны.

Для того, чтобы защитить систему от проблемы “потери прав”, предлагается использовать только основные права на чтение и запись, так как эти права реализованы во всех СУБД, где реализован контроль над правами. Однако, если необходимы иные права доступа, то необходимо учитывать это на этапе проектирования гетерогенного хранилища больших данных и обеспечивать поддержку этих прав каждым из компонентов, с которым требуется согласование.

Также необходимо избегать наличия связи «многие ко многим» и разбивать его на промежуточную сущность, с которой будет организована связь «один ко многим», так как алгоритмы из работ [8-10] трудно справляются с данной задачей.

Литература:

1. Полтавцева М.А., Калинин М.О. Моделирование системы управления Большими данными в информационной безопасности. Проблемы информационной безопасности. 2019. №1. с. 69-78.

2. Zhang Conghui, Li Yi, Sun Wenwen, Guan Shaopeng. Blockchain Based Big Data Security Protection Scheme. 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). 2020. pp. 574-578.

3. Aujla G. S., Chaudhary R., Kumar N., Das A. K. and C. Rodrigues J. J. P. SecSVA: Secure Storage, Verification, and Auditing of Big Data in the Cloud Environment. IEEE Communications Magazine. 2018. V. 56, No. 1, pp. 78-85.

4. Li Jiaxing, Wu Jigang, Jiang Guiyuan, Srikanthan Thambipillai. Blockchain-based public auditing for big data in cloud storage. Information Processing & Management. 2020. V. 57. No. 6. p. 102.

5. Пучков Е.В., Пономарева Е.И. Разработка информационно-аналитической системы на основе многомерного хранилища данных. Инженерный вестник Дона. 2012. № 4 (часть 1). URL: ivdon.ru/ru/magazine/archive/n4p1y2012/1123.

6. Zhao Yanqi, Yu Yong, Li Yannan, Han Gang, Du Xiaojiang. Machine learning based privacy-preserving fair data trading in big data market. Information Sciences. 2019. V. 478, 449-460.

7. Сафонов С.В. Реализация временных моделей для темпоральных надстроек системы управления баз данных. Инженерный вестник Дона. 2022. № 11. URL: ivdon.ru/ru/magazine/archive/n11y2022/8009.

8. Kuszera E. M., Peres L. M., Fabro M. D. Toward RDB to NoSQL: transforming data with metamorfose framework. Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. 2019. pp. 456-463.

9. Ramzan S., Bajwa I. S., Kazmi R. An intelligent approach for handling complexity by migrating from conventional databases to big data. Symmetry. 2018. V. 10. No. 12. p. 698.

10. Ghotiya S., Mandal J., Kandasamy S. Migration from relational to NoSQL database. IOP Conference Series: Materials Science and Engineering. IOP Publishing. 2017. V. 263. p. 4.

References:

1. Poltavceva M.A. Kalinin M.O. Problemy informacionnoj bezopasnosti. 2019. №1. pp. 69-78.

2. Zhang Conghui, Li Yi, Sun Wenwen, Guan Shaopeng. 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). 2020. pp. 574-578.

3. Aujla G. S., Chaudhary R., Kumar N., Das A. K. and C. Rodrigues J. J. P. IEEE Communications Magazine. 2018. V. 56, No. 1, pp. 78-85.

4. Jiaxing Li, Jigang Wu, Guiyuan Jiang, Thambipillai Srikanthan. Information Processing & Management. 2020. V. 57. No. 6. p. 102.

5. Puchkov E.V., Ponomareva E.I. Inzhenernyj vestnik Dona. 2012. № 4. URL: ivdon.ru/ru/magazine/archive/n4p1y2012/1123.

6. Yanqi Zhao, Yong Yu, Yannan Li, Gang Han, Xiaojiang Du. Information Sciences. 2019. V. 478, 449-460.

7. Safonov S.V. Inzhenernyj vestnik Dona. 2022. № 11. URL: ivdon.ru/ru/magazine/archive/n11y2022/8009.

8. Kuszera E. M., Peres L. M., Fabro M. D. Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. 2019. pp. 456-463.

9. Ramzan S., Bajwa I. S., Kazmi R. Symmetry. 2018. V. 10. No. 12. p. 698.

10. Ghotiya S., Mandal J., Kandasamy S. IOP Conference Series: Materials Science and Engineering. IOP Publishing. 2017. V. 263. p. 4.