УДК 004.6-047.44

ДЕПЕРСОНАЛИЗАЦИИ БОЛЬШИХ ДАННЫХ

А.А. Ковалевский
Разработчик, Иностранное частное производственное унитарное предприятие "ИССофт Солюшенз"

Иностранное частное производственное унитарное предприятие "ИССофт Солюшенз", Республика Беларусь
E-mail: antonkw.mail@gmail.com

Аннотация. Описываются методики и алгоритмы деперсонализации информации в контексте Больших Данных. Доклад описывает базовые подходы, которые могут использоваться для замещения персональной информации в больших объемах данных. Реализованный модуль запускается на нескольких распределенных вычислительных кластерах, показаны полученные усреднённые метрики при варьирующихся исходных условиях. Проводится демонстрация линейной масштабируемости описанных методик на реальном окружении.

Ключевые слова: персональные данные, большие данные, деперсонализация информации, распределенные вычисления.

Современное развитие технологий привело к внедрению высоких технологий во все сферы.

Переход на цифровой документооборот и возможность сохранения всевозможных транзакций привела к безостановочно увеличивающемуся количеству данных практически обо всех сферах и аспектах жизни современного человека. Естественно, информация, которая описывает образ жизни человека, включая все его привычки, траты, потребление услуг, стала крайне интересной рекламным агентствам, всевозможным аналитикам, страховым агентствам и многим другим. В таком контексте остро встает вопрос сохранения конфиденциальности данных. Как отдельные люди, так и огромные организации хотят пользоваться агрегированными данными. Данные же зачастую содержат персональную информацию различных людей, и передача в изначальном виде третьим сторонам не представляется возможной. Таким образом, возникает проблема анализа необезличенных данных. Информация, содержащая данные об отдельных людях или же просто какие-либо факты, позволяющие с помощью какого бы то ни было подхода идентифицировать отдельных людей, не может быть использована.

Соответственно, возникает задача деперсонализации информации таким образом, чтобы поставленная бизнес-задача решалась без риска раскрытия конфиденциальных данных. Например, владелец сети аптек хочет узнать какое из лекарств, продающихся исключительно по рецепту, продается активнее всего. У этой сети аптек в хранилище хранятся все транзакции: можно просмотреть все записи о покупках, в случае с лекарствами, купленными по рецепту, в записи будет отображаться полное имя покупателя. Необходимо отдать все данные о транзакциях независимым аналитикам, чтобы они смогли сформировать детализированный отчет. Но отдавать исходные данные в изначальном виде нельзя: законодательство ограничивает возможность распространения персональных данных (а таковыми являются имена покупателей). В случае же подмены оригинальных имен какими-либо случайными или сформированными по определенным правилам значениям это ограничение снимается, аналитики могут получить

наборы данных и предоставить отчет, а владелец сети аптек абсолютно законно использует данные, на которых изначально было наложено много ограничений. И если задача простого удаления конфиденциальных данных является относительно несложной, то задача замещения конфиденциальных данных другими, семантически корректными, но деперсонализированными, гораздо более многогранная и комплексная.

В данном докладе рассматриваются аспекты реализации модуля программного модуля, предназначенного для деидентификации персональных данных.

Самый простой и очевидный подход убрать персональные данные из любого набора информации – удалить непосредственно идентифицирующую информацию (имена, фамилии, адреса, номера документов). Однако такой подход делает информацию менее ценной с точки зрения дальнейшего анализа. При этом оставшаяся информация даже с учетом того, что она не является непосредственно идентифицирующей, может использоваться для повторной идентификации[1].

Маскирование данных - это метод создания структурно подобной, но недостоверной версии данных. Цель процедуры состоит в том, чтобы защитить реальные данные, когда они не требуются или же не могут быть переданы в исходном виде из соображений безопасности. При маскировании данных формат данных остается неизменным. Изменяются только значения. Данные могут быть изменены несколькими способами, включая шифрование, перетасовку символов и замену символов или слов. Независимо от того, какой метод выбран, значения должны быть изменены таким образом, чтобы сделать невозможным восстановление исходных данных. Маскировка относится к набору манипуляций непосредственно над идентифицирующей информацией, содержащейся в данных. В целом, прямые идентификаторы должны быть удалены из набора данных, также допустимо такое редактирование, которое обеспечит невозможность восстановления оригинальной информации, например, замена случайными значениями, генерируемыми по заданным правилам. При таком подходе, в зависимости от семантики, данные могут заменяться фальшивыми аналогами. Например, имена людей, как правило, могут быть заменены фальшивыми именами, выбранными случайным образом из заранее заготовленного списка имен. Числа, такие как финансовые показатели можно заменять случайно сгенерированными, но при этом допустимыми по смыслу или каким-либо правилам, значениями. Места, имена объектов также обычно могут быть отредактированы схожим образом. Такие манипуляции над данными относительно просты для выполнения над структурированными данными.

Для реализации были выбрано алгоритмы, позволяющие обрабатывать следующие категории данных, которые могут быть заменены на фиктивные:

–сочетания имени и фамилии человека;

–сочетания почтового индекса, штата и города;

–адресные строки, включающие в себя город, улицу и дом;

–даты;

–адреса электронной почты;

–случайные строки;

–номера юридических лиц;

–номера страховых полисов;

–инициалы людей;

–телефонные номера.

Для формирования конечного набора сочетания имен и фамилий людей использовались списки имен и фамилий людей, участвовавших в переписи населения США 2000-го года. Для получения максимально большого набора данных было произведено декартово произведение списка имен на список фамилий. Как результат были получены всевозможные сочетания имен

и фамилий. При деидентификации конкретного имени и фамилии выбирается случайное сочетание из подготовленного словаря. Если имя и фамилия деидентифицировались ранее, будут использоваться те же самые значения.

Для получения данных о существующих штатах, индексах и городов использовался официальный сайт правительства Америки. Реализованный алгоритм на базе имеющейся имеющегося адреса генерирует новое местоположение. Из оригинальной записи выбирается только штат. В рамках этого штата выбирается случайным образом город. Индекс выбирается случайным образом, но при этом индекс должен соответствовать городу, а город - индексу.

Адресные строки генерируются на базе имеющегося словаря названий улиц. Название улицы выбирается случайным образом. Номера дома и квартиры генерируются случайным образом.

Даты генерируются исходя из того, что деидентифицируемая дата - это дата рождения человека. Дата выбирается случайным образом из диапазона, который включает в себя даты, которые не изменят возраст человека на момент запуска модуля, если их использовать в качестве дня рождения.

Оставшиеся категории данных деидентифицируются без помощи подготовленных заранее словарей. Для каждой категории подготовлены и описаны правила в виде регулярных выражений. Генерация реализована с помощью технологии Generex[2], свободно распространяемой библиотекой с открытым исходным кодом.

С помощью данной технологии реализованы генераторы для следующих категорий:

–адреса электронной почты;

–случайные строки;

–номера юридических лиц;

–номера страховых полисов;

–инициалы людей;

–телефонные номера.

Первый модуль определяет три главные стратегии:

–генерация нового значения алгоритмом с дальнейшим сохранением соответствия оригинального значения деидентифицированному;

–генерация нового значения алгоритмом без дальнейшего сохранения связей;

–использование случайно выбранных значений из предварительно подготовленного списка с дальнейшим сохранением соответствия оригинального значения деидентифицированному.

В рамках распределенных вычислений сохранение и использование уникальных соответствий потенциально может стать причиной медленной обработки. Каждый из узлов кластера вынужден проверять наличие соответствия деидентифируемому значению уже подготовленного значения. При его отсутствии выбирается новое значение. При этом любой другой узел вычислительного кластера может проделывать те же операции, поэтому необходимо делать несколько дополнительных атомарных запросов вида «проверить наличие значения, и, если его нет в хранилище – добавить».

Были выбраны следующие технологии для реализации модуля:

–Apache Hadoop[3];

–Apache Spark[4];

–Apache HBase[5].

Высокоуровневая архитектура разделяет работу модуля на следующие этапы:

–загрузка файла в формате CSV из локального окружения в распределенную файловую систему (HDFS), для этого может использоваться протокол SCP;

–загрузка данных из HDFS в память машин кластера с помощью SparkSQL;

–работа распределенных алгоритмов, использующих Spark и HBase;

–выгрузка обработанных данных в HDFS;

–выгрузка данных из HDFS в локальное окружение.

Для возможности снятия метрик в течение нагрузочного тестирования был реализован генератор тестовых данных. Было сгенерировано два файла различных размеров: 1 и 10 гигабайт. Для произведения замеров был арендован кластер Amazon AWS с предустановленным дистрибутивом Hortonworks HDP. Кластер состоял из восьми вычислительных серверов. Такого количество работающих параллельно машин достаточно для того, чтобы определить, является ли линейной зависимость времени обработки данных от количества имеющихся ресурсов.

Было произведено десять запусков для каждого сценария использования. Для построения графика зависимости времени обработки от количества ресурсов использовались медианные значения результатов измерения. Зависимость времени обработки 1 гигабайта данных от количества серверов при первичном запуске отображено на рисунке 1.

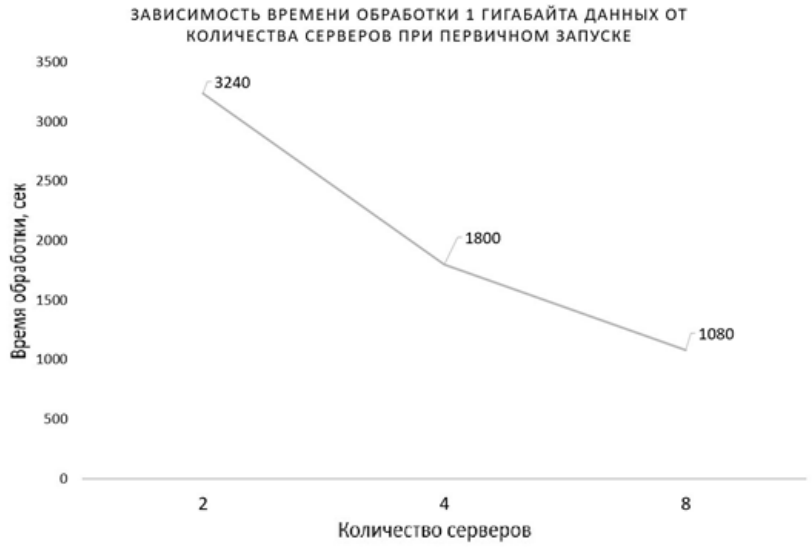

Рисунок 1. Обработка при первичном запуске

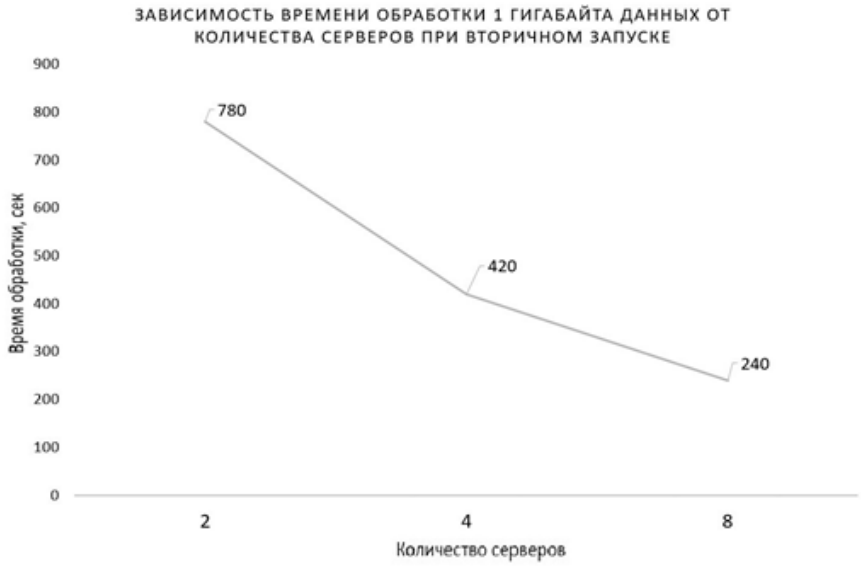

Рисунок 2. Обработка при вторичном запуске

При вторичном запуске обработка должна производится быстрее, так как вместо вычисления нового значению алгоритм просто извлекается уже имеющееся соответствие. График зависимости времени обработки 1 гигабайта данных от количества серверов при вторичном запуске отображен на рисунке 2.

Можно утверждать, что масштабируемость реализованных распределенных алгоритмов близка к линейной. Так, обработка 10 гигабайт данных при первичном запуске на двух машинах заняла 34920 секунд. На восьми машинах обработка этих же данных заняла 12600 секунд. То есть увеличение вычислительных мощностей в 4 раза привела к увеличению скорости обработки в 3 раза. В случае распределенной обработки на отдельно стоящих серверах наращивание мощности всегда ведет к увеличению затрат на коммуникацию между вычислительным машинами, поэтому вышеуказанные результаты позволяют утверждать, что реализованные алгоритмы имеют масштабируемость, близкую к линейной, имея в виду, что алгоритмы эффективно распределяют вычисления при добавлении дополнительных вычислительных мощностей в виде дополнительных серверов.

Список литературы

[1]. Simson L. Garfinkel // De-Identification of Personal Information // National Institute of Standards and Technology (NIST) Internal Report 8053 (Gaithersburg, MD: NIST, October 2015)

[2]. Generex [Электронный ресурс]. — Электронные данные. — Режим доступа: https://github.com/mifmif/Generex. — Дата доступа: 14.02.2018.

[3]. Apache Hadoop [Электронный ресурс]. — Электронные данные.— Режим доступа: http://hadoop.apache.org. — Дата доступа: 01.03.2018.

[4]. Apache Spark [Электронный ресурс]. — Электронные данные.— Режим доступа: http://spark.apache.org. — Дата доступа: 01.03.2018.

[5]. Apache HBase [Электронный ресурс]. — Электронные данные.— Режим доступа: http://hbase.apache.org/. — Дата доступа: 01.03.2018.

DEPERSONALISATION OF BIG DATA

A.A. KOVALEVSKY
Developer, ISsoft Solutions

ISsoft Solutions, Republic of Belarus
E-mail: antonkw.mail@gmail.com

Abstract. I will describe a methodology and algorithms for information deidentification is scope of Big Data. The report describes basic approaches which could be used for replacing personal information in big amount of data. Implemented module runs on various clusters, so I can demonstrate linear scalability of described approaches on real environment.

Key words: personal information, big data, data depersonalization, distributed calculations.