УДК 004.652

ОБРАБОТКА БОЛЬШИХ ДАННЫХ В ТЕЛЕКОММУНИКАЦИОННЫХ КОМПАНИЯХ

А. С. Соболь

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: alexander.so8ol@gmail.com

Анализируются вопросы обработки больших данных на примере задачи о сборе, хранении и анализе информации SIEM-системой посредством СУБД. Основной проблемой при решении задачи является невозможность обработки всех событий в реальном времени при использовании SIEM-системы в вычислительной территориально распределённой сети оператора. В ходе решения задачи формализуются требования к аппаратной платформе, на которой будет работать СУБД SIEM-системы. Выбирается СУБД, оптимально подходящая для решения задачи. Описываются этапы настройки и оптимизации СХД и СУБД для работы с большими данными в целом и для SIEM-системы в частности. Приводятся результаты и скорость работы готового решения в среде «частного облака». Демонстрируется, что правильный выбор и настройка СХД и СУБД могут дать увеличение производительности SIEM-системы в области анализа данных и формирования отчётности в разы, в сравнении со штатными настройками СХД и СУБД, поставляемой «из коробки».

Ключевые слова: СУБД, «облачные» вычисления, большие данные, бизнес-аналитика, СХД.

PROCESSING OF BIG DATA IN THE TELECOMMUNICATION COMPANIES

A. S. Sobol

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: a1exander.so8ol@gmail.com

The article analyses such issues as processing ”big data” by the example of the problem of collecting, storing and analyzing information in SIEM system through database. The major problem is impossibility of processing all events in real time by using a SIEM system in computational geographically distributed operator network. In the course of solving the problem, the requirements for hardware platform are formalized on the database of SIEM system. The DBMS is selected as optimally suitable for solving the problem. The steps to configure and optimize the storage and use of DBMS to work with ”big data” as a whole and SIEM system as the particular are described. The result and the performance of the working solution are provided in the ”private cloud” environment. The correct choice and configuration of the storage and use of DBMS can provide performance of SIEM system in data analysis and reporting, that gains of many times in comparison with the standard settings for storage and use of DBMS, that came out of the box.

Keywords: DBMS, Cloud computing, Big Data, BI, SAN.

Технологии больших данных успешно реализуются в различных индустриях, таких как банки, телеком, ритейл, энергетика, медицина и управление городской инфраструктурой. В секторе телекоммуникаций более 45 % компаний ведут активные проекты с участием больших данных [1]. Основными задачами в данном секторе являются бизнес-аналитика, «умные» маркетинговые кампании, выявление мошенничества, улучшения качества связи. Интересно, что при всем разнообразии задач вендорские решения в сфере больших данных пока не приобрели ярко выраженной отраслевой направленности. Рынок находится не просто на стадии активного формирования, а в самом начале этой стадии.

Под задачами выявления мошенничества в первую очередь понимаются задачи, связанные с анализом поступающей информации от различных систем (таких как DLP, IDS, антивирусов, файерволов, маршрутизаторов) для дальнейшего выявления отклонения от норм по каким-либо критериям и с последующей генерацией инцидентов.

Сама по себе данная задача не является наукоёмкой и решается при помощи SIEM-систем (Security Information and Event Management – система сбора и корреляции событий). Основная проблема заключается в том, что при расширении информационной инфраструктуры (равно как и при перенаправлении потока событий на SIEM-систему с других ИС) количество

событий, генерируемое в единицу времени, возрастает, что в итоге сказывается на увеличении времени обслуживания БД в общем и ведет к пропуску каких-либо важных инцидентов безопасности в частности, что, в свою очередь, недопустимо.

В статье ставится задача проанализировать объём, источники и типы генерируемых событий информационной инфраструктурой, сформулировать критерии к аппаратно-программной части комплекса и настроить СУБД и систему хранения данных (СХД) таким образом, чтобы SIEM-система при помощи данной СУБД могла обеспечивать обработку всех событий в режиме реального времени.

Для определения размера СХД необходимо оценить объём занимаемого места одним событием, а также частоту поступления событий. Все поступающие события в рамках данной задачи будут разделены на две группы: неочищенные и структурированные. Большинство событий поступает по общепринятому стандарту SYSLOG. Размер записи по данному стандарту в соответствии со спецификацией RFC 3164 не может превышать 1024 байта [2]. Однако это применимо лишь к неочищенным событиям. На практике структурированные события могут превышать 5000 байт. В рамках данной задачи опытным путём при помощи таких инструментов, как WireShark, TCPdump и EtherPeek, определено, что неочищенные события в среднем занимают порядка 500 байт дискового пространства, структурированные – порядка 1600 байт.

При среднем количестве, равном 12 000 (для примера система управления интернет-рекламой в среднем генерирует порядка 50 000 событий в секунду) событий в секунду, количество генерируемых событий в день будет равняться 1 036 800 000.

Таким образом, неочищенные события в день будут занимать порядка 500 ГБ, структурированные 1,5 ТБ. Данные такого вида легко сжимаемы, поэтому конечный вариант сжатых данных при коэффициенте сжатия, равном десяти, будет составлять 50 и 150 ГБ соответственно. Наконец, определим, сколько всего необходимо пространства на СХД: для этого согласно ТЗ экстраполируем получившиеся данные на шесть месяцев и высчитаем доли, занимаемые в общем объёме неочищенными и структурированными событиями. Неочищенные события составляют приблизительно треть от общего числа генерируемых событий. Таким образом, для хранения событий в течение 180 дней необходимо хранилище, объёмом не менее 21 ТБ.

Зная общую потребность в объёме и информацию о частоте поступления событий, определим требования к аппаратной части комплекса. Ключевую роль здесь играет производительность СХД. Синтетические тесты показали, что производительности в 15 000–17 000 IOPS будет вполне достаточно для данной системы.

Традиционно, чтобы обеспечить гибкость своих ИТ-решений, компаниям приходилось постоянно обновлять свои аппаратные комплексы. Результатом таких решений на сегодняшний день являются имеющиеся в наличии множественные «малые» хранилища данных следующих форматов:

– SATA-массивы + аппаратный/программный RAID;
– SAS-массивы + аппаратный RAID;
– SAS-массивы + Flash (SSD) + аппаратный RAID.

На данном этапе важно понимать, что объёмы информации согласно закону Мура постоянно растут, а, соответственно, вычислительных мощностей со временем понадобится гораздо больше. Соответственно, при выборе СХД нужно ориентироваться на наличие у СХД такого функционала, как Tiering (перемещение данных между дисками и/или дисковыми массивами различных классов стоимости и производительности) и Thin provisioning (динамические тома). Для экономии места также желательно, чтобы СХД обладала функцией дедупликации.

Исходя из этих требований, становится очевидным, что традиционные решения «малых» хранилищ данных не подходят для решения этой задачи. Массивы начального уровня с одним контроллером также не обеспечат достаточный уровень производительности. К тому же данные решения слабо защищены от сбоев. Наиболее логичным выбором в данной ситуации является выбор СХД в сегменте Mid-Range: многоконтроллерные модульные конструкции из множества дисковых полок.

Следующий вопрос – количество, типы дисков и их соотношение в СХД. Одним из определяющих факторов является то, что частота перезаписи достаточно мала (в среднем не более 3–4 раз в год), в то время как циклов чтения, наоборот, может быть неограниченное количество (это связано с характером системы, а именно, постоянные выборки из уже имеющихся данных в рамках задачи анализа).

Учитывая наличие функционала перемещения данных между дисками и/или дисковыми массивами, наиболее рациональным решением с точки зрения цена/производительность является установка в СХД трёх типов дисков: твердотельных (маленьких по объёму, но быстрых), классических с большим RPM (средние по объёму и скорости) и классических большеобъёмных дисков (большой объём, маленькая скорость) в соотношении 1/3/2.

Хорошим вариантом Mid-Range сегмента является СХД 3PAR 7200, которая и будет использоваться для тестирования и дальнейшей настройки в данной статье.

При решении задачи стоит уделить внимание не только настройке СУБД, но и настройке СХД. Основной проблемой при настройке СХД является нахождение оптимального баланса между твердотельными (SSD) и обычными жёсткими дисками. Твердотельные накопители обладают достаточной скоростью чтения/записи, но при этом дороги и менее долговечны. Дополнительным фактором является тот факт, что общий эффективный объём массива (исходя из ТЗ, логи должны храниться шесть месяцев) с учётом 20 % запаса должен составлять не менее 25 ТБ. При этом необходимо понимать, что чтобы добиться такого эффективного объёма, номинальный объём должен быть ещё больше, в зависимости от уровня RAID (отказоустойчивого массива независимых дисков), в котором будет создан массив.

Опытным путём было установлено, что для обеспечения необходимой скорости чтения/записи при экономии места структура массива должна быть построена по типу RAID 6 – она не намного медленнее RAID 5, но гораздо более экономична.

Для реализации задачи было решено отказаться от СУБД, с которой по умолчанию работала SIEM-система, и выбрать наиболее подходящую. Среди критериев отбора были выделены следующие:

– наличие системы полнотекстового поиска;

– поддержка индексов;

– поддержка CAS (compare and set – сравнение с обменом) инструкции, необходимой для гарантии отказа в изменении объекта, если с момента последнего чтения объект был изменен другим клиентом;

– наличие инструментов массово-параллельной обработки данных.

Из списка NoSQL СУБД была выбрана MongoDB. Основным фактором выбора данной СУБД, с учётом использования в качестве серверной составляющей blade-центра, явилась поддержка вычислительной парадигмы, известной как MapReduce. Преимущество MapReduce заключается в том, что она позволяет распределённо производить операции предварительной обработки и свертки. Операции предварительной обработки работают независимо друг от друга и могут производиться параллельно (хотя на практике это ограничено источником входных данных и/или количеством используемых процессоров). MapReduce может быть применена к большим объемам данных, которые могут обрабатываться большим количеством серверов [3].

При работе с MongoDB использовался C#-драйвер – NoRM, поддерживающий LINQ (набор функций, значительно расширяющий синтаксис языка C#).

К самой СУБД были применены оптимизационные алгоритмы чтения и записи, которые в первую очередь повысили производительность, а также позволили снизить нагрузку на аппаратную часть. Среди решений по оптимизации чтения стоит отметить:

– размещение индексов в кэше, а точнее на SSD-дисках;

– использование bloom-фильтров совместно с 10-битными строками с целью уменьшения ложного поиска; примером использования может служить конструкция следующего вида: SELECT x WHERE q = = 10 AND y < 100;

– отказ от использования множества фильтров в целях уменьшения времени отклика;

– использование вторичных индексов.

Под оптимизацией записи в первую очередь понимались меры, направленные на увеличение времени жизни твердотельных накопителей. Было включено журналирование и выделено под него порядка 40 % объёма всех SSD-дисков. Для уменьшения фрагментирования размеры страниц БД были уменьшены.

Из минусов данной СУБД следует отметить невозможность работы с LSM-tree (Log-Structured Merge Trees) индексированием. Вместо него используется B-tree индексирование. Недостатками B-tree структуры построения индексов по сравнению с LSM-tree в данном случае является трудоемкость балансировки дерева при добавлении нового значения в индекс и относительно большая ресурсоемкость, так как индекс хранится в оперативной памяти.

Тестирование производительности происходило следующим образом: СУБД SIEM-системы начинала обрабатывать события. Параллельно запускался инструмент генерации отчётов об инцидентах; результатом считалось то количество обработанных событий в секунду, которое не попадало в очередь, ожидая обработки. Сравнивалась производительность СУБД SIEM-системы до оптимизации СХД и СУБД и после. Замеры производились в трёх состояниях:

1) СХД не оптимизирована, «коробочная» версия СУБД;

2) СХД оптимизирована, «коробочная» версия СУБД;

3) СХД оптимизирована, оптимизированная СУБД MongoDB.

В среднем решение «из коробки» способно обрабатывать порядка 3500 событий, промежуточная версия с оптимизированной СХД способна справиться с 13 000 событий. Версия с изменённой СУБД смогла показать результат в 16 200 событий в секунду. Стоит отметить, что данную производительность не стоит считать максимальной, так как в данных условиях сгенерировать большее число событий не представилось возможным.

В целом результаты тестирования показали, что правильный выбор и настройка СХД и СУБД может дать увеличение производительности SIEM-системы в области обработки данных в разы, в сравнении со штатными настройками СХД и СУБД, поставляемой «из коробки». Использование парадигм «частного облака», таких как объединение ресурсов, эластичность и унифицированность, использованных в реализации задачи, гарантирует лёгкое масштабирование аппаратных ресурсов, необходимых для бесперебойной работы SIEM-системы.

Конечным результатом стала рабочая программно-аппаратная платформа с SIEM-системой, способная обрабатывать большие данные в режиме реального времени, снизив тем самым риск пропуска значимых инцидентов до нуля.

Библиографические ссылки

1. Tadviser. Государство. Бизнес. ИТ [Электронный ресурс]. URL: http://www.tadviser.ru/images/3/3d/Cnews_infa_bigdata_4.jpg (дата обращения: 15.04.2014).

2. Internet Engineering Task Force [Электронный ресурс]. URL: http://www.ietf.org/rfc/rfc3164.txt (дата обращения: 15.04.2014).

3. Tadviser. Государство. Бизнес. ИТ [Электронный ресурс]. URL: http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:MapReduce (дата обращения: 15.04.2014).

References

1. *Tadviser. Gosudarstvo. Biznes. IT* [Tadviser. Government. Business. IT]. Available at: http://www.tadviser.ru/images/3/3d/Cnews_infa_bigdata_4.jpg/. (accessed 15.04.2014).

2. Internet Engineering Task Force. Available at: http://www.ietf.org/rfc/rfc3164.txt (accessed 15.04.2014).

3. *Tadviser. Gosudarstvo. Biznes. IT* [Tadviser. Government. Business. IT]. Available at: http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:MapReduce. (accessed 15.04.2014).

УДК 658.512.001.56

СИСТЕМА ФОРМИРОВАНИЯ СОСТАВА МУЛЬТИВЕРСИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В РЕАЛЬНОМ ВРЕМЕНИ

Е. В. Соловьев

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: blackdeathangel@rambler.ru

Описывается мультиверсионное программное обеспечение и его применение для обеспечения работы системы обработки данных реального времени. Данная методология основывается на введении программной избыточности и позволяет существенно повысить уровень надежности программного обеспечения. В качестве примера реального применения программного обеспечения подобного типа была выбрана система управления системой обработки данных реального времени долговременной орбитальной станции. На основании имеющихся данных о системе управления долговременной орбитальной станции была разработана тестовая задача, представляющая собой план полета из последовательности режимов работы станции с заданными ограничениями. Проведены эксперименты с помощью стандартного алгоритма муравьиной колонии на разработанной тестовой задаче.

Ключевые слова: оптимизация, муравьиные алгоритмы, мультиверсионное программное обеспечение, системы реального времени.

THE SYSTEM OF FORMATION OF THE COMPOSITION MULTIVERSIONED VIEWS OF THE SOFTWARE IN REAL TIME

Y. V. Solovyev

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: blackdeathangel@rambler.ru

The article describes software multiversioned views and its application to ensure that the system is processing real-time data. This methodology is based on the introduction of software redundancy and allows to significantly increase the reliability of the software. As a practical example of application software of this type management system, the data processing system of real-time long-term orbital station was chosen. Based on the available data about the long-term orbital station to the task of representing the flight plan from a sequence of modes of operation of the station, with specified restrictions was developed. The experiments were conducted using standard ant colony algorithm developed on the test task.

Keywords: optimization, ant algorithms, multiversioned views software, real time systems.