УДК 004.65

Карипжанова Ардак Жумагазиевна
докторант, Евразийский национальный университет имени Л.Н.Гумилева, г.Астана
e-mail: kamilakz2001@mail.ru
Сагиндыков Каким Молдабекович
кандидат технических наук, Евразийский национальный университет имени Л.Н.Гумилева, г.Астана
e-mail: ksagin@mail.ru

СПОСОБЫ ПОВЫШЕНИЯ НАДЕЖНОСТИ ХРАНЕНИЯ ИНФОРМАЦИИ В БАЗАХ ДАННЫХ

Мақалада деректер қорында деректерді сақтаудың сенімділігін арттыру жолдары, RAID массивтері ұғымдарының анықталу, түзету кодтары, «Big Data» термині қарастырылған. Сондай-ақ, соңғы уақытта сақталған ақпараттың көлемінің күрт өсуіне байланысты туындаған ақпаратты тасымалдаушылардың сенімділік мәселесінің өзектілігі көрсетілген. Үстеме шығыстарды азайтуға мүмкіндік беретін, репликацияны пайдаланғаннан гөрі әлдеқайда аз артықтықпен деректерді сақтаудың сенімділігін жетілдіретін өшіру кодтарын пайдалану ұсынылған.

Түйін сөздер: *деректер, деректер қоры, түзету кодтары, RAID массивтері, Big Data.*

В статье рассматриваются способы повышения надежности хранения данных в базах данных, раскрываются понятия RAID массивов, корректирующих кодов, термина «Big Data». Также показана актуальность проблемы надежности носителей информации, возникшая в последнее время, с резким ростом объемов хранимой информации. Предложено применение кодов со стираниями, повышающие надежность хранения данных с гораздо меньшей избыточностью, чем при использовании репликации, что позволяет снизить накладные расходы.

Ключевые слова: *данные, база данных, корректирующие коды, RAID массивы, Big Data .*

In the article written about the ways of improving the reliability of data storage in databases, and given the concepts of RAID arrays, correction codes and the term "Big Data". Also there is shown the relevance of the problem of reliability of storage media that has arisen recently, because of sharp increase in the volume of stored information. There is suggested ways of using codes with erasures which can increase data storage reliability with much less redundancy than using replication which help to reduces overhead.

Keywords: *data, database, correction codes, RAID arrays, Big Data.*

Одним из основных требований к базам данных является надежность хранения данных во внешней памяти, т.е. база данных должна обладать способностью восстановления последнего согласованного состояния БД после любого аппаратного или программного сбоя. Возможны два вида аппаратных сбоев: «мягкие» сбои, которые приводят к внезапной остановке работы компьютера (например, аварийное выключение питания), и «жесткие» сбои, характеризуемые потерей информации на носителях внешней памяти. Программные сбои — это аварийное завершение работы системы управления базами данных или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается

незавершенной. Для восстановления БД нужно располагать некоторой дополнительной информацией, что требует избыточности хранения данных. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД-Журнал — это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. Самая простая процедура обеспечения надежности восстановления БД — откат транзакции, выполненной пользователем, для чего все записи от одной транзакции связывают обратным списком от конца к началу (аналог Undo) [1].

Надежность систем хранения данных определяется многими факторами.

Очевидный способ повышения надежности хранения путем повышения надежности жестких дисков представляется наиболее простым и приемлемым методом. Однако, это связано с уровнем развития технологии производства жестких дисков, в котором есть определенный предел, выше которого подняться в принципе невозможно. Т.е., теоретически нужно повышать надежность элементов, чтобы вероятность безотказной работы системы удовлетворяла заданным требованиям. Однако практическая реализация такой высокой надежности элементов не всегда возможна.

Особенно проблема надежности носителей информации стала актуальной в последнее время, с резким ростом объемов хранимой информации. Появился новый термин — «Big Data», характеризуя собой новые веяния — необходимость хранить и обрабатывать все большие и большие объемы информации. С увеличением темпов роста объемов информации к проблемам аппаратной надежности отдельных устройств хранения добавляются проблемы обеспечения надежности в многокомпонентных системах — эффекты, связанные с масштабированием хранилищ.

Для повышения надежности применяются специальные технологии, позволяющие добиться высокой надежности хранения. В первую очередь, это технологии аппаратного резервирования и репликации данных. Аппаратные устройства, использующиеся при хранении данных, резервируются, как минимум, двукратно. Дублируются также элементы влияющие на надежность функционирования устройства хранения, например, практически всегда дублируется блок питания.

На уровне информационных данных широко распространена репликация данных. Данные, в процессе записи на носители, одновременно реплицируются (копируются) на несколько мест хранения. Репликация характеризуется коэффициентом репликации R, которая равна количеству записываемых копий.

Наиболее распространенным вариантом репликации является, например, зеркалирование — параллельная запись данных на два места одновременно, с коэффициентом репликации $R = 2.0$. В технологиях работы с большими данными «Big Data», когда используются массивы мест хранения, практически не используются коэффициенты меньше чем $R = 3.0$. Коэффициент репликации по умолчанию равен 3.0, например, в системах объектного хранения Apache Hadoop и OpenStack Swift, в распределенной файловой системе RedHat Ceph и т.д. На практике этот параметр настраивается в зависимости от требований к надежности и может достигать значений $R = 16.0 - 32.0$.

Оценим вероятность отказа систем с резервированием. Если вероятность отказа единичного диска равна P_{dev}, то вероятность отказа P_{arr} массива из n дисков равна совместной вероятности отказа n дисков, т.е. должны отказать все диски одновременно

$$P_{arr} = P_{dev,1} \cdot P_{dev,2} \cdot \cdots \cdot P_{dev,n} = \prod_{i=1}^{n} P_{dev,i}$$

Если используются одинаковые диски, то для коэффициента репликации $R = n$ получаем

$$P_{arr} = P_{dev}^{n}$$

При применении, например, дисков с AFR=1,0% с вероятность. отказа диска в течение года $P_{dev} = 0{,}01$, то зеркалирование уменьшит вероятность отказа до $P_{arr} = 10^{-4}$ (0,01%), а при коэффициенте репликации 3,0 вероятность пропажи информации в таком хранилище уменьшится до $P_{arr} = 10^{-6}$ (0,0001%).

Резервирование очень эффективный способ повышения надежности, но, в то же время, слишком расточительный. С увеличением коэффициента репликации, во столько же раз увеличивается избыточность. Если нам нужно записать D битов данных, то при записи в

хранилище с коэффициентом репликации $R = n$ фактически будет записано D_{arr} битов, что в n раз больше

$$D_{arr} = D_{dev,1} + D_{dev,2} + \cdots + D_{dev,n} = \sum_{i=1}^{n} D_{dev,i} \quad (3)$$

$$D_{arr} = nD_{dev} \quad (4)$$

При хранении и передаче данных в базах данных неизбежно возникают ошибки и/или потери информации. Для повышения надежности требуется обеспечить контроль целостности данных, исправление ошибок и восстановление потерь.

Возможны несколько стратегий при возникновении ошибок или отказов:

- обнаружение ошибок в данных и автоматический запрос повторной передачи повреждённых блоков;
- отбрасывание повреждённых блоков;
- исправление ошибок.

Корректирующие коды — это коды, служащие для обнаружения и, если возможно, исправления ошибок, возникающих при передаче и хранении информации.

В корректирующих кодах при записи данных на места хранения или во время передачи добавляется специальным образом подготовленная избыточная информация. Избыточные данные служат для обнаружения и восстановления пропавшей или поврежденной информации. Способность обнаружения ошибок, уровень допустимых потерь и повреждений, естественно, зависят от применяемого кода.

Коды, обнаруживающие ошибки, не всегда могут исправить ошибки, хотя этот процесс обнаружения и исправления ошибок тесно связаны друг с другом — установление факта наличия ошибки, не всегда означает, что ее можно исправить.

Любой код, исправляющий ошибки, может также обнаруживать ошибки, но не наоборот. Корректирующие коды способны обнаружить больше ошибок, чем способны исправить [3].

В технологиях надежного хранения информации применяют в основном блочные коды — способ кодирования, когда информация обрабатывается фиксированными блоками. Основной математической моделью, в рамках которой обычно рассматривается хранение информации, является модель каналов со стираниями. В данной модели основное внимание уделяется ошибкам типа стирания потому, что при хранении основной тип ошибок — это именно отказ устройств хранения, а не искажение самой информации. Применение такой модели позволяет разработать более эффективные коды, повышающие надежность хранения, так как обнаружить позицию стертого канала гораздо проще, чем вычислить позицию искаженного символа в рамках просто корректирующих кодов[4].

В технологиях хранения нашли применение, де-факто, два вида кодов со стираниями:

- циклический избыточный код (Cyclic redundancy check, CRC) — самый простой вид кодирования, не требующий сложных вычислений;

Cyclic redundancy check (CRC) — алгоритм нахождения контрольной суммы. Контрольная сумма вычисляется как четность битов в принятом блоке. По контрольной сумме можно обнаружить и исправить единичные ошибки.

- более сложный и требовательный к вычислительным ресурсам код Рида-Соломона.

Коды Рида-Соломона[5], это циклические коды, позволяющие исправлять ошибки в блоках данных. Элементами кодового вектора являются не биты, а блоки состоящие из групп битов. Наиболее распространены коды, работающие с байтами. Код Рида — Соломона является частным случаем БЧХ-кодов[6].

Применение кодов со стираниями позволяет повысить надежность хранения данных с гораздо меньшей избыточностью, чем при использовании репликации, что позволяет снизить накладные расходы. Коды со стираниями особенно актуальны в технологии Big Data и в системах архивного хранения, когда требуется хранить огромные массивы данных.

RAID массивы

Разработанная в Калифорнийском университете Беркли технология объединения массива дисков в одно устройство с отказоустойчивой к отказам дисков большей емкостью было фактическим стандартом в индустрии хранения данных. Первоначально RAID означало «Redundant array of inexpensive disks» — избыточный массив недорогих дисков. Подразумевалось, что за счет избыточности можно добиться надежного хранения на недорогих дисках, снижая стоимость хранения данных. Со временем аббревиатура стала

трактоваться как «Redundant array of independent disks» — избыточный массив независимых дисков, так как технология RAID оказалась наиболее востребована в секторе дорогих серверных решений, где определяющим преимуществом является не понижение стоимости, а повышение надежности и производительности.

Разработчиками, Д. Паттерсоном, Г. Гибсоном и Р. Катцем (David A. Patterson, Garth Gibson, and Randy H. Katz), были представлены следующие уровни спецификации RAID, которые были приняты как стандарт де-факто:

- RAID1 — зеркальный дисковый массив;
- RAID2 — зарезервирован для массивов, которые применяют код Хемминга;
- RAID3 и RAID4 — дисковые массивы с чередованием и выделенным диском чётности;
- RAID5 — дисковый массив с чередованием и отсутствием выделенного диска чётности.

Технология повышения надежности, применяемая в RAID, использует корректирующие коды CRC на основе контроля четности. Хотя для уровня RAID2 был зарезервирован немного более сложный алгоритм кодов Хэмминга [11], он так и не нашел практического применения. Коды Хэмминга, в отличие от CRC, могут обнаруживать две ошибки, хотя, как и CRC, могут исправить только одну.

Как и все многодисковые системы RAID позволяет добиться не только повышения надежности, но и увеличения производительности. Зачастую именно этот параметр имеет решающее значение при выборе RAID в серверных конфигурациях. Собственно, этим и объясняется применение именно CRC, а не более сложных кодов, корректирующих ошибки, так как любые сложные вычисления начинают сказываться на производительности.

На сегодняшний день RAID дополнен еще несколькими уровнями, появились комбинированные конфигурации, так называемый Nested RAID, многие компании разрабатывают проприетарные варианты в которых также присутствует «фирменное» название RAID, что отражает, зачастую, всего-навсего то, что это очередное многодисковое устройство.

В современных RAID-контроллерах предоставлены дополнительные уровни спецификации RAID:

- RAID0 — дисковый массив повышенной производительности с чередованием, без отказоустойчивости. Строго говоря, RAID-массивом не является, поскольку избыточность в нём отсутствует;
- RAID6 — дисковый массив с чередованием, использующий две контрольные суммы, вычисляемые двумя независимыми способами.

Комбинированные уровни RAID:

- RAID10 — массив RAID0, построенный из массивов RAID1;
- RAID01 — массив RAID1, построенный из массивов RAID0 (имеет низкую отказоустойчивость);
- RAID1E (зеркало из трёх устройств);
- RAID50 — массив RAID0 из массивов RAID5;
- RAID05 — RAID5 из RAID0;
- RAID 60 — RAID0 из RAID6.

Аппаратный RAID-контроллер может иметь дополнительные функции и одновременно поддерживать несколько RAID-массивов различных уровней. RAID-контроллеры стали встраивать даже в BIOS дорогих персональных компьютеров предназначенных для высокопроизводительных вычислений. При этом контроллер, встроенный в материнскую плату, в настройках BIOS имеет всего два состояния включён или отключён, поэтому новый жёсткий диск, подключённый в незадействованный разъём контроллера при активированном режиме RAID, может игнорироваться системой, пока он не будет ассоциирован как ещё один RAID-массив типа JBOD, состоящий из одного диска.

Рассмотрим технологию RAID в плане обеспечения надежности хранения больших массивов информации на примере наиболее популярного RAID5.

До определенного момента все работает хорошо, появляющиеся сбойные секторы к потере данных не приводят, так как они тут же компенсируются за счет данных четности. При выходе из строя одного из дисков, RAID теряет возможность восстанавливать ошибки и необходимо заменить диск на исправный. Чтобы восстановить потерянные данные с отказавшего диска необходимо запустить процедуру Rebuild.

Если вероятность отказа единичного диска равна P_{dev}, то вероятность отказа одного из дисков в массива из n дисков равна сумме всех вероятностей

$$P = P_{dev,1} + P_{dev,2} + \cdots + P_{dev,n} \quad (5)$$

Вероятность отказа диска в массиве выше вероятности отказа одиночного диска в n раз. Т.е. в RAID вероятность единичного отказа диска тем выше, чем больше в нем дисков.

$$P = nP_{dev} \quad (6)$$

В обычных условиях, когда 1ТБ данных считалось достаточно большим объемом, в RAID использовались диски малой емкости. В серверных системах, где главное это производительность, использовались высокоскоростные диски со скоростью вращения 10-15 тыс. оборотов в минуту и емкостью максимум 11ГБ. На сегодня требуется обработка и хранение огромных объемов информации, поэтому емкость дисков достигает терабайтов. В таких условиях начинают сказываться аппаратные факторы надежности, которыми уже нельзя пренебрегать. Рассмотрим влияние на надежность такого параметра, как UER.

Пусть имеется RAID5 состоящий из 6 дисков по 600ГБ с UER=10^{-15}. После отказа одного из дисков запущен процесс восстановления пропавших данных.

Общий объем считываемых данных при Rebuild: 5 дисков по 600ГБ = $0{,}24 \cdot 10^{14}$ бит.

Вероятность появления невосстановимой ошибки:

$$P_{UER} = (0{,}24 \cdot 10^{14}) \cdot 10^{-15} = 0{,}024 \quad (7)$$

С пяти дисков нужно считать страйпы, рассчитать контрольные суммы и записать их на шестой диск, который был заменен новым исправным. В процессе Rebuild есть вероятность возникновения ошибок в виде «испорченного» бита (англ. «гнилой бит» – bit rot), который невозможно скорректировать.

Для рассматриваемой конфигурации, (8) массиве из 6 дисков по 600ГБ класса Enterprise, вероятность невозможности восстановления данных 2,4%. Это означает, что, если использовать RAID5 с дисками большого объема, мы можем получить недопустимо высокую вероятность потери данных.

Чем больше размер диска и больше количество дисков в массиве вероятность потери данных возрастает. Например, для массива из 16 дисков по 2ТБ

Общий объем считываемых данных при Rebuild: 15 дисков по 2ТБ = $2{,}40 \cdot 10^{14}$ бит.

Вероятность появления невосстановимой ошибки:

$$P_{UER} = (2{,}40 \cdot 10^{14}) \cdot 10^{-15} = 0{,}24$$

вероятность ошибки восстановления уже достигает 24%.

В современных системах хранения данных применяются диски с емкостями до 10ТБ и, очевидно, процесс наращивания объемов будет продолжаться. Применение в системах хранения данных стандартных RAID контроллеров недопустимо. В связи с этим разрабатываются новые технологии многодисковых хранилищ с использованием более сложных алгоритмов коррекции ошибок.

Список использованной литературы

1 G. Schulz, Resilient Storage Networks: Designing Flexible Scalable Data Infrastructures., Oxford: Elsevier Science, 2004.

2 Тикунов В. С. Геоинформатика. Обеспечение надежности хранения данных в БД. Поддержка языков управления БД. Типовая организация СУБД.

3 Wiki, «Обнаружение и исправление ошибок,» Википедия, [В Интернете]. Available: https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B8_%D0%B8%D1%81%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BE%D1%88%D0%B8%D0%B1%D0%BE%D0%BA.

4 P. B. G. Y. W. M. O. W. a. K. R. Alexandros G. Dimakis, «Network Coding for Distributed Storage Systems,» EEE Transactions on Information Theory, т. 56, № 9, p. 4539–4551, September 2010.

5 Википедия, «Код Рида — Соломона,» [В Интернете]. Available: https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A0%D0%B8%D0%B4%D0%B0_%E2%80%94_%D0%A1%D0%BE%D0%BB%D0%BE%D0%BC%D0%BE%D0%BD%D0%B0.

6 Википедия, «Код Боуза — Чоудхури — Хоквингема,» [В Интернете]. Available: https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%91%D0%BE%D1%83%D0%B7%D0%B0_%E2%

80%94_%D0%A7%D0%BE%D1%83%D0%B4%D1%85%D1%83%D1%80%D0%B8_%E2%80%94_%D0%A5%D0%BE%D0%BA%D0%B2%D0%B8%D0%BD%D0%B3%D0%B5%D0%BC%D0%B0.
7 У. Э. Питерсон У., Коды, исправляющие ошибки: Пер. с англ., Москва: Мир, 1976.

Карипжанова Ардак Жумагазиевна
Лауазымы: Л.Н.Гумилев атындағы ЕҰУ «Ақпараттық жүйелер» кафедрасының докторанты
Пошталық мекен-жайы: F18C3H6 индексі, Семей қаласы, Красин көшесі, 223
Ұялы тел.: +77779845361
Сагиндыков Каким Молдабекович
Лауазымы: техника ғылымдарының кандидаты, доцент, заведующий кафедры «Информатика және ақпараттық қауіпсіздік» кафедрасының меңгерушісі, Л.Н.Гумилев атындағы ЕҰУ
Пошталық мекен-жайы: 010008, Қазақстан Республикасы, Астана қ., Пушкин к.,11
Ұялы тел.: +7 777 984 53 61
Деректер қорында ақпараттарды сақтау сенімділігін жетілдіру жолдары

Карипжанова Ардак Жумагазиевна
Должность: докторант кафедры «Информационные системы» ЕНУ имени Л.Н.Гумилева
Почтовый адрес: F18C3H6 , Республика Казахстан, г. Семей, ул Красина, 223
Сот. тел.: +7 777 984 53 61
Сагиндыков Каким Молдабекович
Должность: кандидат технических наук, доцент, заведующий кафедры «Информатики и информационной безопасности» Евразийского национального университета имени Л.Н.Гумилева
Почтовый адрес: 010008, Республика Казахстан, г. Астана, ул. Пушкина,11
Сот. тел.: +7 777 984 53 61
Способы повышения надежности хранения информации в базах данных

Karipzhanova Ardak Zhumagazievna
Position: doctoral student of the department "Information Systems", ENU named after after L.N.Gumilyov
Mailing address: F18C3H6, Republic of Kazakhstan, Semey, Krasina str., 223
cells. ph: +77779845361
Sagindykov Kakim Moldabekovich
Position: Candidate of Technical Sciences, Associate Professor, Head of the Department of "Informatics and Information Security", ENU named after after L.N.Gumilyov
Mailing address: 010008, Republic of Kazakhstan, Astana, Pushkin str.,11
cells. ph: +77779845361
Methods to improving the reliability of storage of information in databases