Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование
информационных систем
Кафедра информационно-аналитических систем

Слесарев Александр Германович

Разработка анализатора индекса подсистемы хранения данных InnoDB

Курсовая работа

Научный руководитель:
ассистент кафедры Информационно-Аналитических Систем Чернышев Г. А.

Санкт-Петербург
2022

Оглавление

Введение

Во время эксплуатации СУБД могут возникать ситуации непреднамеренного отключения системы. На данный момент нет способов предсказать все нештатные ситуации при которых может произойти выход из строя сервера СУБД. Однако в основном причинами служат ошибки в программном коде СУБД или несовершенство аппаратной составляющей сервера.

Существуют встроенные способы проверки файлов базы на наличие повреждений. Например, в подсистеме хранения InnoDB [1] у каждой страницы имеется поле чексуммы (checksum), необходимое для проверки ее целостности. С его помощью можно обнаружить ошибку в случае повреждения данных на диске. Стоит заметить, что данные могут быть повреждены во время нахождения в памяти или кеше процессора, тогда чексумма не поможет.

Наиболее распространенным способом восстановления данных после нештатного отключения СУБД является журналирование изменений таблиц. В InnoDB этот механизм назван redo log[1], его функционирование осуществляется за счет добавления на каждую страницу поля lsn, позволяющего отслеживать версии страниц.

На практике возникают ситуации, при которых недостаточно стандартных механизмов восстановления базы после аварии. В таких случаях администраторам баз данных и разработчикам хранилищ требуются специальные утилиты для выяснения причин возникновения нештатных ситуаций, а также для проверки целостности файлов СУБД после аварийного отключения.

В данной работе представлен обзор существующих инструментов анализа подсистемы хранения InnoDB, а также реализована система Ibdump, предоставляющая отсутствующую в рассмотренных аналогах функциональность.

[1] `https://dev.mysql.com/doc/dev/mysql-server/latest/PAGE_INNODB_REDO_LOG.html`

1. Постановка задачи

Целью данной работы является обзор инструментов проверки целостности страниц подсистемы хранения InnoDB и реализация нового инструмента offline проверки Ibdump, имеющего недостающую у аналогов функциональность. Для достижения этой цели были выделены следующие подзадачи:

- Провести обзор инструментов проверки целостности страниц InnoDB.
- Реализовать функциональность просмотра содержимого страниц данных InnoDB.
- Реализовать функциональность валидации страниц данных InnoDB.
- Встроить тестирование Ibdump в MySQL Test Framework.

2. Принципы хранения данных в InnoDB

Для хранения данных в InnoDB используется следующая иерархия объектов в порядке вложенности: строка (row), страница (page), экстент (extent), сегмент (segment), табличное пространство (tablespace). Рассмотрим подробнее каждый из объектов.

Строка — логическая структура данных, определяемая набором колонок, которые в свою очередь задаются пользователем при определении таблицы в CREATE TABLE Statement. Набор строк составляет таблицу.

Страница — единица представления данных, необходимая для чтения данных из файла в память (buffer pool). Состоит из метаинформации и строк данных, стандартный размер – 16КБ.

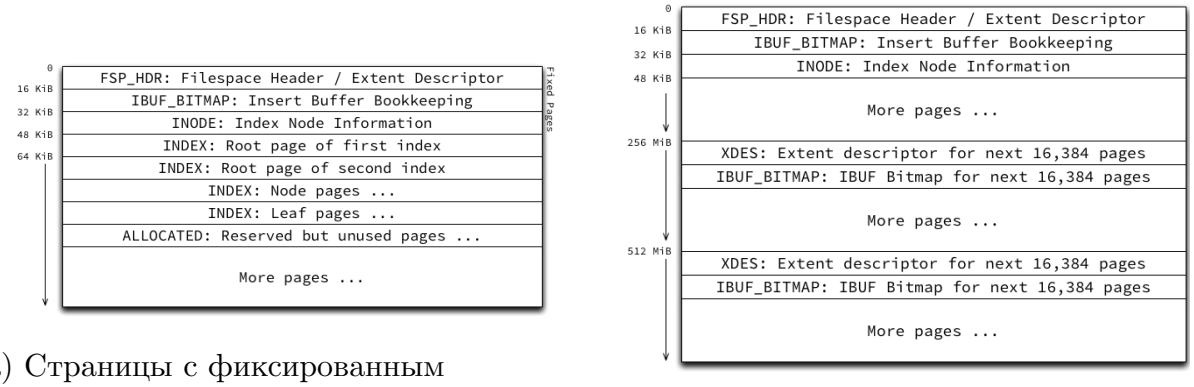
Экстент — набор страниц внутри одного табличного пространства. Используется как единица измерения количества данных в некоторых операциях, например, read-ahead или при работе определенных модулей, таких как doublewrite buffer. Стандартный размер 1МБ (64 страницы).

Сегмент — в контексте индексирования данных, сегменты являются составными частями индекса. Внутри одного табличного пространства, индекс имеет по одному файловому сегменту для листовых и не листовых страниц. Существуют также вспомогательные сегменты, например, для свободных страниц. Как правило для расширения файловых сегментов выделяется память, пропорциональная размеру одного экстента.

Табличное пространство — общем случае, это логическая структура, включающая в себя несколько файлов с данными. Стандартной конфигурацией InnoDB считаются single-table tablespaces. В таком случае, на каждую таблицу выделяется отдельное табличное пространство в виде отдельного .ibd файла. Изменять данную конфигурацию системы можно с помощью переменной innodb_file_per_table.

На данный момент Ibdump поддерживает single-table tablespaces, поэтому далее в работе будет подразумеваться эта конфигурация.

3. Структура индекса InnoDB



(a) Страницы с фиксированным расположением

(b) Общая структура

Рис. 1: Структура файла индекса[2]

InnoDB поддерживает несколько типов индексов. Стандартным и самым распространенным является кластеризованный индекс (clustered index), основанный на B^+-деревьях [2]. Кластеризованный индекс хранит в себе непосредственно сами данные, что позволяет экономить место и сокращает время чтения с диска. На данный момент Ibdump поддерживает обработку кластеризованного индекса, поэтому далее в работе будет подразумеваться этот тип индекса.

Файл индекса состоит из страниц метаинформации и непосредственно страниц данных (index page). Первая страница (Рис. 1a) называется FSP_HDR (file space header) и содержит в себе FSP заголовок, который нужен для хранения метаинформации обо всем файле, например, размер пространства и списки заполненных или пустых экстентов. Также FSP_HDR может хранить метаиноформацию о 256 экстентах (или 16 384 страниц, 256 МБ). Когда количество экстентов становится кратно 256, выделяется дополнительная XDES страница (Рис. 1b). Структура страниц XDES и FSP_HDR идентична, за исключением того, что структура FSP заголовка на XDES страницах пустая.

Страница IBUF_BITMAP нужна для работы Change Buffer и напрямую не влияет на структуру индекса, поэтому ее устройство не будет затронуто в данной работе.

[2] `https://blog.jcole.us/innodb`

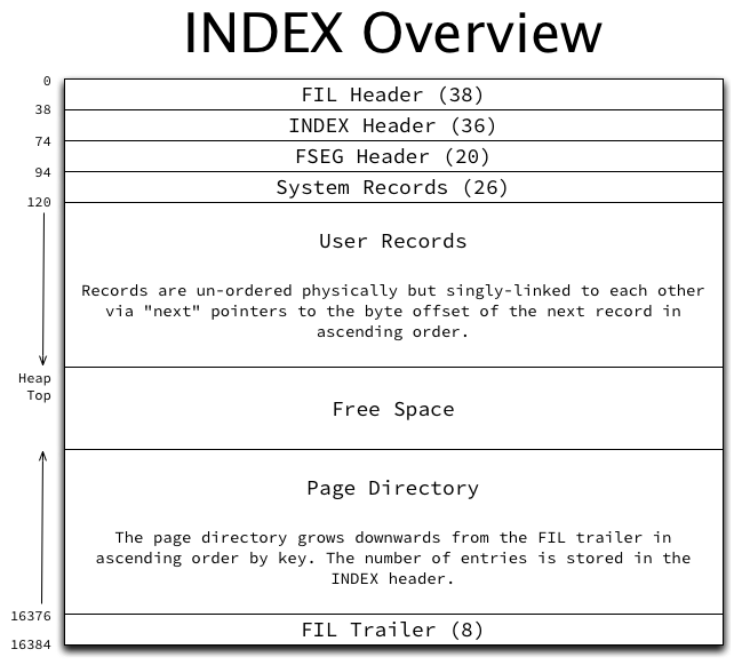


Рис. 2: Структура страницы индекса

Для понимания устройства следующего типа страниц, назывыаемых INODE, нужно ввести понятие файлового сегмента. В одной таблице может существовать несколько индексов. Файловый сегмент служит идентификатором принадлежности страницы к индексу, причем каждый индекс имеет отдельные файловые сегменты для листовых и не листовых страниц. INODE страница состоит из списка INODE записей, каждая из которых содержит метаинформацию об одном файловом сегменте.

Далее рассмотрим устройство INDEX страницы (Рис. 2). FIL заголовок и FIL трейлер — типичные для всех типов страниц структуры. Содержат чексумму, lsn, метаинформацию и указатели на предыдущую и следующую страницы. В случае индекса указатели на страницы нужны для поддержания связного списка одного уровня индекса и отсортированы по возрастанию ключа.

Затем расположен INDEX заголовок, необходимый для менеджмента записей на странице. Этот заголовок содержит в себе ID индекса, к которому принадлежит страница, число записей, ID транзакции последней модификации, номер уровня страницы, информацию о слотах каталога (directory slots) и некоторые другие вспомогательные данные.

FSEG заголовок — заполняется только у корневых страниц, хранит указатели на корневую и листовую INODE страницы.

Большую часть страницы индекса занимают записи, они выделяются по возрастанию отступа от начала страницы (offset). В конце страницы находятся слоты каталога, это список из офсетов, указывающих на группы по 4-8 записей на странице. С помощью слотов каталога можно производить бинарный поиск по записям. Список слотов каталога растет по убыванию офсета.

4. Обзор утилит проверки целостности базы данных

В этом разделе приведен обзор существующих консольных инструментов с открытым исходным кодом, используемых для анализа и восстановления файлов базы данных.

UnDrop for InnoDB[3] — инструмент создан на языке C и C++, основным применением является восстановление данных при отсутствии резервной копии после следующих сбоев: таблица или база данных удалены, табличное пространство InnoDB повреждено, сбой жесткого диска, повреждение файловой системы, записи были удалены из таблицы, таблица была удалена и создана пустая.

Ibd2sdi[4] — подсистема в InnoDB для извлечения SDI метаинформации, которая может быть полезна при анализе стуктуры табличного пространства. В MySQL 8.0 введены SDI страницы, дублирующие данные из системных файлов, поэтому SDI может быть полезна при восстановлении после сбоя.

Innodb_ruby[5] — парсер файлов InnoDB на языке Ruby. Печатает информацию со страниц или из отдельных записей в собственном формате. Помимо вывода информации со страниц позволяет печатать статистики по страницам, например, их заполненность данными или времена последних обновлений lsn.

MySQL InnoDB Java Reader[6] — утилита анализа файлов InnoDB на Java. Во многом схожа с innodb_ruby, но помимо уже перечисленных возможностей способна печатать тепловые карты обновлений lsn и выполнять поиск записи по первичному и вторичному ключу.

Inno_space[7] — инструмент на C и C++ для анализа и восстановления файлов InnoDB. Система позволяет удалять поврежденные страницы и обновлять чексумму страницы после внесения изменений.

[3] https://github.com/twindb/undrop-for-innodb

[4] https://dev.mysql.com/doc/refman/8.0/en/ibd2sdi.html

[5] https://github.com/jeremycole/innodb_ruby

[6] https://github.com/alibaba/innodb-java-reader

[7] https://github.com/baotiao/inno_space

5. Обзор системы Ibdump

Работа выполнялась по заказу компании Huawei, поэтому исходный код системы на данный момент не общедоступен. По требованию заказчика разработка велась на языке C++ 14 для MySQL 8.0.

Ibdump выполнен в виде консольной утилиты для ОС на основе Linux и тестировался на серверах с архитектурой процессора x86 и ARM. Система имеет два основных режима: дамп и валидация.

Режим дампа позволяет выводить весь ibd файл целиком или только указанный диапазон страниц. Поддерживаются следующие форматы вывода: бинарный, 16 разрядный, JSON, XML, "человекочитаемый". Также для сокращения количества выводимой информации доступна функция фильтрации по типу страниц, можно фильтровать системные, листовые индексные и нелистовые индексные страницы. Помимо фильтрации реализован подробный режим (verbose), который определяет количество выводимой информации о страницах.

Проверки из режима валидации разбиты на несколько групп: валидация структуры системный страниц, валидация структуры страниц индекса, проверка чексумм, общие проверки.

Валидация структуры системных страниц ibd файла содержит в себе проверки:

1. Поле state у XDES записей имеет допустимое значение (от 0 до 5).
2. Оффсеты XDES записей удовлетворяют структуре списка.
3. Узел (List node) в XDES записи не пустой.
4. Узел (List node) в XDES записи содержит действительный адрес.
5. У каждой страницы с данными действительная XDES запись.
6. Записи FULL_FRAG листа имеют соответствующее значение поля state.
7. Записи FREE_FRAG листа имеют соответствующее значение поля state.

8. Записи FREE листа имеют соответствующее значение поля state.
9. Записи FSEG листа имеют соответствующее значение поля state.
10. Количество FULL_FRAG записей совпадает с полем счетчика в FSP заголовке.
11. Количество FREE_FRAG записей совпадает с полем счетчика в FSP заголовке.
12. Количество FREE записей совпадает с полем счетчика в FSP заголовке.
13. Количество FSEG записей совпадает с полем счетчика в FSP заголовке.
14. Указатели на соседние элементы в двусвязных списках корректны.

Валидация структуры страниц индекса состоит из следующих проверок:

1. Указатель на предыдущий элемент самого левого узла горизонтально прошитого B^+-дерева равен нулю.
2. Указатель на следующий элемент самого правого узла горизонтально прошитого B^+-дерева равен нулю.
3. Информация об id дочерних узлов текущего узла совпадает с действительными id дочерних узлов.
4. Страницы одного уровня в B^+-дереве должны иметь одинаковое поле space id.
5. Нелистовые страницы не должны быть листовыми.

Проверка чексумм включает в себя:

1. Чексумма заголовка совпадает с чексуммой трейлера.

2. Поле чексуммы совпадает с вычисленной чексуммой страницы.

Общие проверки:

1. Поле с типом страницы корректно заполнено.

2. Поле с информацией о сжатии страницы корректно.

В Таблице 1 приведено сравнение функциональности Ibdump с существующими аналогами. В строке "источник информации об индексе" указывается способ получения метаинформации о количестве и составе индексов в текущем ibd файле. Рассмотрено 3 способа:

- Пользовательский — передача CREATE TABLE STATEMENT в качестве одного из аргументов утилиты.
- SDI — чтение информации из SDI страниц внутри ibd файла. Недоступно в версии 5.7.
- Data dictionary — чтение информации из системных файлов.

Таблица 1: Сравнительные характеристики систем анализа файлов InnoDB

утилита	UnDrop	Ibd2sdi	Innodb ruby	InnoDB Java Reader	Inno space	Ibdump
поддерживаемые типы страниц	индекс	SDI	все	большинство	индекс	большинство
валидация чексуммы	есть	нет	есть	нет	есть	есть
валидация оффсетов структур	есть	нет	есть	нет	есть	есть
валидация ограничений значений	есть	нет	нет	нет	нет	есть
валидация дерева индекса	нет	нет	нет	нет	нет	есть
источник информации об индексе	пользователь	SDI	data dictionary	пользователь	нет	data dictionary
бинарный дамп страницы	есть	нет	нет	нет	нет	есть
структурированный дамп страницы	нет	нет	есть	есть	есть	есть
фильтрация дампа	нет	нет	есть	есть	есть	есть
сбор статистики	нет	нет	нет	нет	нет	есть
устранение поломок	есть	нет	нет	нет	нет	нет

6. Тестирование

Тестирование реализовано в виде mtr тестов внутри MySQL Test Suite. Также настроен Gitlab CI для сборки и тестирования Ibdump на процессорах ARM и x86. Всего создано порядка 30 тестовых случаев, разбитых по категориям тестирования дампа, фильтрации, диапазона и валидации. Тестирование опций Ibdump производится на файле базы, приложенном к тестам в виде zip архива. MySQL Test Suite поддерживает использование в тестах Perl скриптов, эта функциональность была задействована при тестировании валидации, поскольку для этого нужно вносить поломки в определенные места ibd файла.

7. Заключение

В рамках данной работы были получены следующие результаты:

- Проведен обзор инструментов проверки целостности страниц InnoDB.
- Реализована функциональность просмотра содержимого страниц данных InnoDB.
- Реализована функциональность валидации страниц данных InnoDB.
- Встроено тестирование Ibdump в MySQL Test Framework.

Список литературы

[1] Fruehwirt Peter, Donko-Huber Markus, Mulazzani Martin, and Weippl Edgar. InnoDB Database Forensics. — 2010. — 01. — P. 1028–1036.

[2] Zhang Donghui, Baclawski Kenneth Paul, and J. Tsotras Vassilis. B+-Tree // Encyclopedia of Database Systems / ed. by LIU LING and ÖZSU M. TAMER. — Boston, MA : Springer US, 2009. — P. 197–200. — ISBN: 978-0-387-39940-9. — Access mode: https://doi.org/10.1007/978-0-387-39940-9_739.