

Базы данных

Лекция 00 – Berkeley DB

Преподаватель: Поденок Леонид Петрович, 505а-5

+375 17 293 8039 (505а-5)

+375 17 320 7402 (ОИПИ НАНБ)

prep@lsi.bas-net.by

ftp://student:2ok*uK2@Rwox@lsi.bas-net.by

Кафедра ЭВМ, 2024

2024.04.16

Оглавление

| | |
|---|---|
| Berkeley DB..... | 3 |
| Службы доступа к данным..... | 5 |
| Хэш-таблицы..... | 5 |
| B-деревья..... | 5 |
| Номера записей..... | 6 |
| Очереди..... | 6 |
| Услуги по управлению данными..... | 7 |
| Чем Berkeley DB не является..... | 8 |
| Berkeley DB не является реляционной базой данных..... | 9 |

Berkeley DB

1) Berkeley DB — это встроенная библиотека баз данных, которая предоставляет приложениям масштабируемые, высокопроизводительные, защищенные транзакциями сервисы управления данными.

2) Berkeley DB является «**встроенной**», поскольку она напрямую связана с приложением. Она работает в том же адресном пространстве, что и приложение. В результате для операций с базой данных не требуется никаких межпроцессных взаимодействий ни по сети, ни между процессами на одном компьютере.

3) Berkeley DB предоставляет простой API вызова функций для доступа к данным и управления ими для ряда языков программирования, включая C, C++, Java, Perl, Tcl, Python и PHP.

4) Все операции с базой данных происходят внутри библиотеки.

Несколько процессов или несколько потоков в одном процессе могут использовать базу данных одновременно, поскольку каждый из них использует библиотеку Berkeley DB.

5) Все низкоуровневое обслуживание, такое как блокировки, журналирование транзакций, управление общими буферами, управление памятью и т. д., библиотекой обрабатываются прозрачно.

6) Библиотека Berkeley DB чрезвычайно переносима. Он работает практически под всеми вариантами UNIX и Linux, Windows и многими встраиваемыми операционными системами реального времени.

7) Она работает как на 32-битных, так и на 64-битных системах.

8) Может быть развернута на высокопроизводительных интернет-серверах, настольных компьютерах, а также на карманных компьютерах, телеприставках, сетевых коммутаторах и в других местах.

9) Berkeley DB хорошо масштабируется. Сама библиотека базы данных довольно компактна (около мегабайта памяти, занимаемой исполняемым кодом на обычных архитектурах), но при этом может использовать гигабайты памяти и терабайты дискового пространства, если дисковое оборудование это позволяет.

10) Каждый из файлов базы данных Berkeley DB может содержать до **256 терабайт данных** при условии, что базовая файловая система способна поддерживать файлы такого размера.

Приложения Berkeley DB часто используют несколько файлов базы данных и объем данных, которым может управлять приложение Berkeley DB, ограничен только ограничениями, накладываемыми операционной системой, файловой системой и физическим оборудованием.

11) Berkeley DB также поддерживает **высокий уровень параллелизма**, что позволяет тысячам пользователей одновременно работать с одними и теми же файлами базы данных.

Berkeley DB превосходит реляционные и объектно-ориентированные системы баз данных во встроенных приложениях по нескольким причинам:

1) поскольку библиотека работает в том же адресном пространстве, для операций с базой данных не требуется межпроцессное взаимодействие. Стоимость связи между процессами на одной машине или между машинами в сети намного выше, чем стоимость вызова функции.

2) поскольку Berkeley DB использует простой интерфейс вызова функций для всех операций, нет языка запросов для разбора и плана выполнения.

Службы доступа к данным

Приложения Berkeley DB могут выбирать структуру хранения, которая лучше всего подходит для приложения. Berkeley DB поддерживает:

- хеш-таблицы;
- В-деревья;
- простое хранилище на основе номеров записей;
- постоянные очереди (persistent queue).

Программисты могут создавать таблицы, используя любую из этих структур хранения, и могут смешивать операции с разными типами таблиц в одном приложении.

Хэш-таблицы

Хэш-таблицы, как правило, хороши для очень больших баз данных, которым требуется предсказуемое время поиска и обновления для записей произвольного доступа.

Хэш-таблицы позволяют пользователям спрашивать: «Существует ли этот ключ?» или получить запись с известным ключом.

Хэш-таблицы не позволяют пользователям запрашивать записи с ключами, близкими к известному ключу.

В-деревья

В-деревья лучше подходят для поиска на основе диапазона, например, когда приложению необходимо найти все записи с ключами между некоторым начальным и конечным значением.

В-деревья также лучше используют *окрестность* ссылки. Если приложение может одновременно работать с ключами, расположенными рядом друг с другом, В-деревья работают очень эффективно.

Древовидная структура хранит ключи, которые находятся рядом друг с другом в памяти, поэтому для извлечения ближайших значений обычно не требуется доступ к диску.

Номера записей

Хранение на основе номеров записей естественно для приложений, которым необходимо хранить и извлекать записи, но у которых нет простого способа генерировать собственные ключи — ключом для записи является номер записи в таблице номеров записей. Berkeley DB автоматически генерирует эти номера.

Очереди

Очереди хорошо подходят для приложений, которые создают записи, а затем должны обрабатывать эти записи в порядке их создания. Хорошим примером являются системы онлайн-покупок. Заказы могут поступать в систему в любое время, но, как правило, они должны выполняться в том порядке, в котором они были размещены.

Услуги по управлению данными

Berkeley DB предлагает важные услуги по управлению данными, включая:

- параллелизм;
- транзакции;
- восстановление.

Все эти услуги работают на всех структурах хранения.

Несколько пользователей могут работать с одной и той же базой данных одновременно.

Berkeley DB прозрачно обрабатывает блокировку, гарантируя, что два пользователя, работающие с одной и той же записью, не будут мешать друг другу.

По умолчанию библиотека обеспечивает строгую семантику транзакций ACID:

- атомарность (Atomicity);
- непротиворечивость (Consistency);
- изолированность (isolation);
- долговечность (durability).

Тем не менее, приложения могут ослаблять изоляцию, которую гарантирует система баз данных.

Несколько операций могут быть сгруппированы в одну транзакцию и могут быть зафиксированы или отменены атомарно.

Приложение при запуске может попросить Berkeley DB запустить восстановление.

Восстановление восстанавливает базу данных до согласованного состояния со всеми зафиксированными изменениями даже после сбоя. База данных гарантированно непротиворечива, и все зафиксированные изменения гарантированно будут присутствовать после завершения восстановления.

Чем Berkeley DB не является

Berkeley DB не является реляционной базой данных.

Berkeley DB не является объектно-ориентированной базой данных.

Berkeley DB не является сетевой базой данных.

Berkeley DB не является сервером базы данных.

В отличие от большинства других систем баз данных, Berkeley DB предоставляет относительно простые службы доступа к данным.

Записи в Berkeley DB представляют собой пары (ключ, значение). Berkeley DB поддерживает только несколько логических операций с записями:

- вставить запись в таблицу;
- удалить запись из таблицы;
- найти запись в таблице, поискав ее ключ;
- обновить уже найденную запись.

Berkeley DB никогда не работает со значением записи.

Значения — это просто полезная нагрузка, которая хранится вместе с ключами и надежно доставляется обратно в приложение по запросу.

И ключи, и значения могут быть произвольными строками байтов фиксированной или переменной длины.

В результате программисты могут помещать структуры данных на их любимом языке программирования в базу данных, не преобразовывая их сначала в формат чужой записи.

Хранение и извлечение очень просты, но **приложению необходимо заранее знать структуру ключа и структуру значения** — приложение не может запросить эту информацию у Berkeley DB, потому что Berkeley DB ее не знает.

Ключи, и значения могут иметь длину до четырех гигабайт, поэтому в одной записи могут храниться изображения, аудиопотоки или другие большие значения данных.

Berkeley DB не является реляционной базой данных.

Доступ к данным, хранящимся в Berkeley DB, осуществляется с помощью традиционных API-интерфейсов Berkeley DB.

Традиционные API-интерфейсы Berkeley DB — это способ, которым большинство пользователей Berkeley DB будут использовать Berkeley DB.

Хотя интерфейсы довольно просты, они нестандартны в том смысле, что не поддерживают операторы SQL.

Поддержка SQL — это палка о двух концах.

Одним из больших преимуществ реляционных баз данных является то, что они позволяют пользователям писать *простые декларативные запросы на языке высокого уровня*. Система базы данных знает все о данных и может выполнить команду. Соответственно, не требуется программирование, а также легко искать данные новыми способами и задавать новые вопросы базе данных.

С другой стороны, если программист может заранее предсказать, как приложение будет обращаться к данным, то написание низкоуровневой программы для получения и хранения записей может быть быстрее. Это устраняет накладные расходы на синтаксический анализ, оптимизацию и выполнение запросов.

Программист должен понимать представление данных и должен написать код для выполнения работы, но как только это будет сделано, приложение может работать очень быстро.

В Berkeley DB нет понятия схемы и типов данных, как в реляционных системах. Схема — это структура записей в таблицах и отношения между таблицами в базе данных. Например, в реляционной системе программист может создать запись из фиксированного набора типов данных. Поскольку типы записей объявляются в системе, реляционная машина может проникать внутрь записей и проверять в них отдельные значения. Кроме того, программисты могут использовать SQL для объявления связей между таблицами и для создания индексов для таблиц. Реляционные механизмы обычно поддерживают эти связи и индексы автоматически.

В Berkeley DB ключ и значение в записи непрозрачны для Berkeley DB.

Они могут иметь сложную внутреннюю структуру, но библиотека об этом не знает. В результате Berkeley DB не может разложить часть значения записи на составные части и не может использовать эти части для поиска интересующих значений. Это может сделать только приложение, которое знает структуру данных.

Berkeley DB поддерживает индексы для таблиц и автоматически поддерживает эти индексы по мере изменения связанных с ними таблиц.

Berkeley DB не является реляционной системой. Системы реляционных баз данных семантически богаты и предлагают высокоуровневый доступ к базе данных. По сравнению с такими системами Berkeley DB представляет собой высокопроизводительную транзакционную библиотеку для хранения записей.

Поверх Berkeley DB можно построить реляционную систему.

Фактически, популярная реляционная система MySQL использует Berkeley DB для управления таблицами с защитой транзакций и берет на себя весь анализ и выполнение SQL. Он использует Berkeley DB для уровня хранения и предоставляет инструменты семантики и доступа.