# Berkeley DB(BDB) 预研报告

## BDB概述

 Berkeley DB是一个开放源代码的内嵌式数据库管理系统，能够为应用程序提供高性能的数据管理服务。程序员只需要调用一些简单的API就可以完成对数据的访问和管理。

与常用的数据库管理系统（如MySQL和Oracle等）有所不同，在Berkeley DB中没有数据库服务器的概念。

应用程序不需要事先同数据库服务建立起网络连接，而是通过内嵌在程序中的Berkeley DB函数库来完成对数据的保存、查询、修改和删除等操作。

Berkeley DB可以保存任意类型的键/值对，而且可以为一个键保存多个数据。

Berkeley DB可以支持数千的并发线程同时操作数据库。

Berkeley DB可以将值（Value）中存储复合数据，只需将复合数据转化为字符串或字节数组，并将其存储为值。

## Berkeley DB特点

简单、小巧、可靠、高性能，小而精。将Berkeley DB和我们的程序共同编译为一个程序（嵌入），和我们的程序运行在相同的地址空间，既没有网络通信的开销也没有进程通信的资源耗费，而且没有复杂的SQL分析解释，使用API进行直接了当的访问。Berkeley DB本身只有几百K，但却能支持256TB的数据，还支持上千用户的并发访问。真正的事务支持——可说是其最强大的特性。支持多线程/进程处理。支持热备份。

Berkeley DB在嵌入式应用中比关系数据库和面向对象数据库要好，有以下两点原因：  
    （1）因为数据库程序库同应用程序在相同的地址空间中运行，所以数据库操作不需要进程间的通讯。在一台机器的不同进程间或在网络中不同机器间进行进程通讯所花费的开销，要远远大于函数调用的开销；  
    （2）因为Berkeley DB对所有操作都使用一组API接口，没有SQL语句接口，因此不需要对某种查询语言进行解析，也不用生成执行计划，大大提高了运行效率.

## Berkeley DB系统结构

Berkeley DB由五个主要的子系统构成.包括: 存取管理子系统、内存池管理子系统、事务子系统、锁子系统以及日志子系统。其中存取管理子系统作为Berkeley DB数据库进程包内部核心组件，而其他子系统都存在于Berkeley DB数据库进程包的外部。

## Berkeley DB存储功能概述

 Berkeley DB所管理数据的逻辑组织单位是若干个独立或有一定关系的数据库(database)，每个数据库由若干记录组成，这些记录全都被表示成(key，value)的形式。

例如：如果把一组相关的(key，value)对也看作一个表的话，那么每一个数据库只允许存放一个table，这一点不同于一般的关系数据库。实际上，在Berkeley DB中所提到的“数据库”，相当于一般关系数据库系统中的表；而“key/data”对相当于关系数据库系统中的行(rows)；Berkeley DB不提供关系数据库中列直接访问的功能，而是在“key/data”对中的data项中通过实际应用来封装字段(列)。

在物理组织上，每一个数据库在创建的时候可以由应用程序根据其数据特点来选择一种合适的存储结构。Berkeley DB 可供选择的四种文件存储结构分别是：哈希文件、B树、定长记录(队列)和变长记录(基于记录号的简单存储方式)。

Berkeley DB和关系数据库不同，它没有和关系数据库中完全对等的表的概念，而只有DBT这样的东西——就是一个Key/Data对，我们可以把key看作表的主键，而Data就是余下的n-1列。我们可以定义一个结构体，存入Data中，这样结构体的每个Field就相当于一个字段。Berkeley DB还有游标（Cursor），有了它，能够遍历多条记录，还能遍历Key相同的多记录。而Berkeley DB中最核心的数据结构就是DB了，它代表数据库，物理上就是.DB文件，我们可以设置其页面大小，存储数据的基础数据结构（平衡二叉树、哈希表、队列和记录）。另外，还有比较复杂的数据库环境DB\_ENV，提供比较高级的功能：事务、多线程、日志、备份和恢复等。

## Berkeley DB本地读写速度测试

测试方案

环境： Windows本地；

存储数据类型：JSON语句；

数据库存储结构：BTree、DB\_HASH、DB\_RECNO、DB\_QUEUE、DB\_HEAP

参数：num\_tables(通道数量), 线程池ThreadPool，num\_records（插入数据数量），线程池线程数量ThreadNum, frequency（每一事务提交执行语句的数量，也即采样频率）；

测试任务：开启事务，使用批量插入，同时向1024通道插入数据1000条数据，即1024个通道都要同时插入1000条数据，采样频率1KHZ；

测试指标：每一秒能存多少数据点，耗时，数据库文件大小。

注释：这个测试方案的表为key，value为JSON语句

测试结果：

写入速度测试：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指标  存储结构 | 写入总共用时 | 每秒能存储数据 | 每个通道每秒存储数据 | 生成数据文件大小 |
| BTree | 1.223s | 837285 | 818 | 74.5MB |
| DB\_HASH | 1.600s | 640000 | 625 | 75.3MB |
| DB\_RECNO | - | - | - | - |
| DB\_QUEUE | - | - | - | - |
| DB\_HEAP | - | - | - | - |

存储结构“DB\_RECNO”、“DB\_QUEUE”的指标为“-”的原因是：类似顺序文件，按记录号存储键值对。适用于按照记录顺序访问数据的应用场景,这里方案中采用多线程写入，不能保证顺序写入和顺序访问的数据场景，出现报错“BDB1540 configured environment flags incompatible with existing environment”。

使用存储结构“DB\_HEAP”的指标为“-”的原因是：会出现访问堆栈缓冲区发生越界问题，原因是：代码中访问堆栈缓冲区越界造成，数据项没有特定的顺序，多线程同时写入Heap数据库时，由于没有明确的数据排序，可能会导致数据项之间的竞争条件，引发冲突和数据损坏，容易造成访问越界的问题。

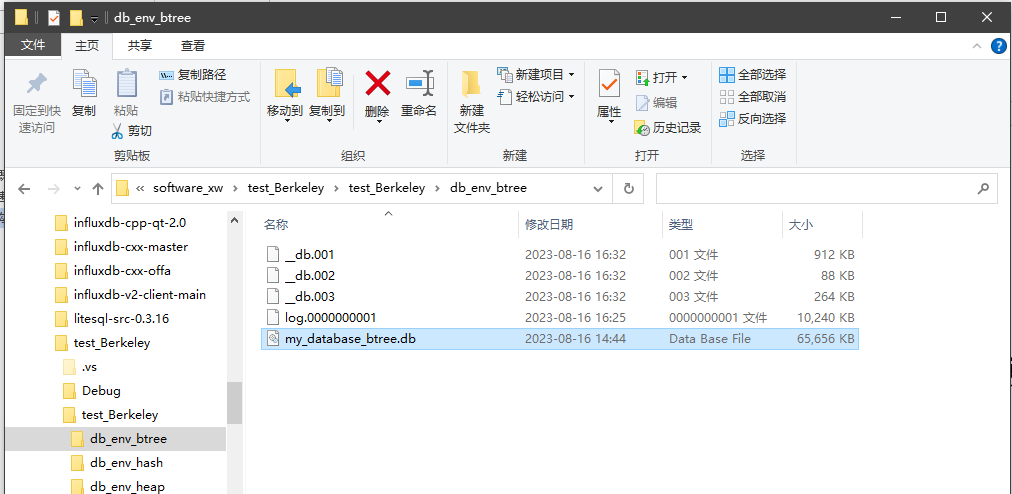
当每个通道写入的数据量分别达到10000，100000,1000000时，指标为：

|  |  |  |
| --- | --- | --- |
| 指标  存储结构 | 写入总共用时 | 生成数据文件大小 |
| BTree | 10.913s/112.182s/1232.65s | 619MB/6.11GB/62.1G |
| DB\_HASH | 11.048s/112.952s/1235.04s | 619MB/6.11GB/62.1G |
| DB\_RECNO | - |  |
| DB\_QUEUE | - |  |
| DB\_HEAP | - |  |

读取速度测试：

|  |  |  |  |
| --- | --- | --- | --- |
| 指标  存储结构 | 读取总共用时 | 每秒能读取数据 | 每个通道每秒读取数据 |
| BTree | 0.076s | 13473684 | 13157 |
| DB\_HASH | 0.077s | 13298701 | 12987 |
| DB\_RECNO | - | - | - |
| DB\_QUEUE | - | - | - |
| DB\_HEAP | - | - | - |

## Berkeley DB数据库结构损坏恢复测试



通过删除\_db.001、\_db.002、\_db.003任意一个文件，使用恢复语句命令

db\_recover -v -h /path/to/database\_environment\_directory

即可恢复删除文件，具体过程为：删除任意一个文件—\_db.00x文件，执行恢复语句后，该文件下的\_db.00x所有被删除，并生成日志log.0000000001文件，且可以使用打印日志语句进行查看数据库系统记录的执行操作。

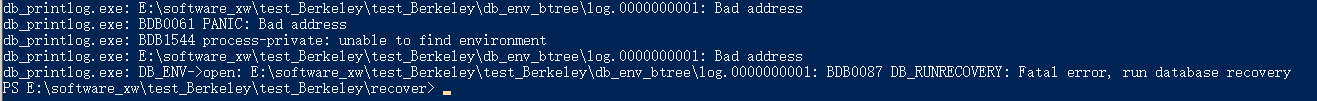
# 查看数据库环境中所有日志文件的内容

db\_printlog -h /path/to/database\_environment

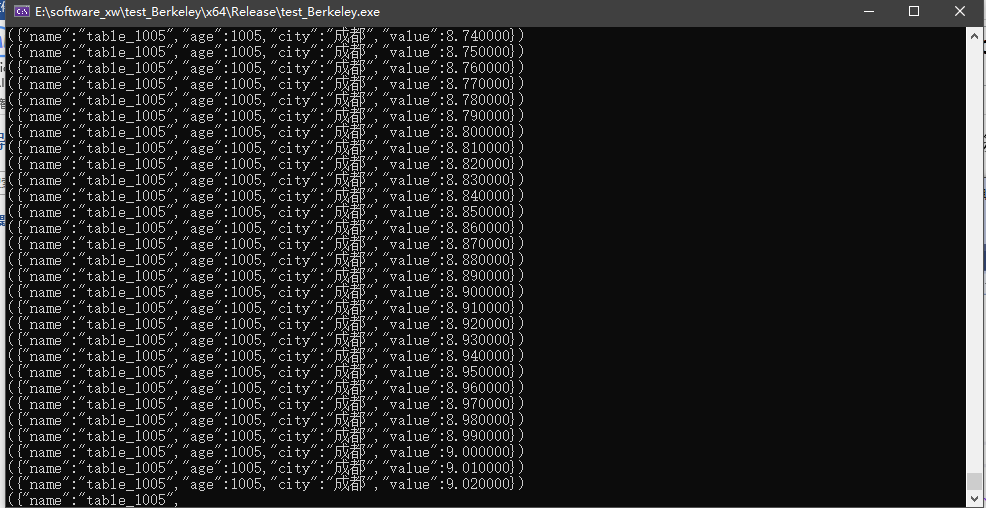
# 查看特定日志文件的内容

db\_printlog -h /path/to/database\_environment log.0000000001 log.0000000002

日志执行记录为：



经恢复\_db.00x文件后，可以使用api语句查询到数据库内容为：



总结：1、损坏的日志文件（Log Files）： db\_recover 可以尝试恢复损坏或不完整的日志文件，以确保事务的一致性和完整性。它可以检查和修复日志文件中的记录，以及执行事务的回滚和重放操作。

2、损坏的数据文件（Data Files）： 如果数据库数据文件损坏，db\_recover 可以尝试恢复数据文件的结构，确保数据能够正确访问。

3、索引文件（Index Files）： 如果数据库的索引文件损坏，db\_recover 可以尝试恢复索引的结构，以确保数据可以通过索引正确地访问和查询。