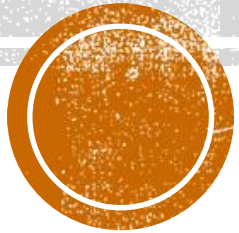
GeekBand 极客班

C++项目简介

C++ Project: Simple In-memory KeyValue DB



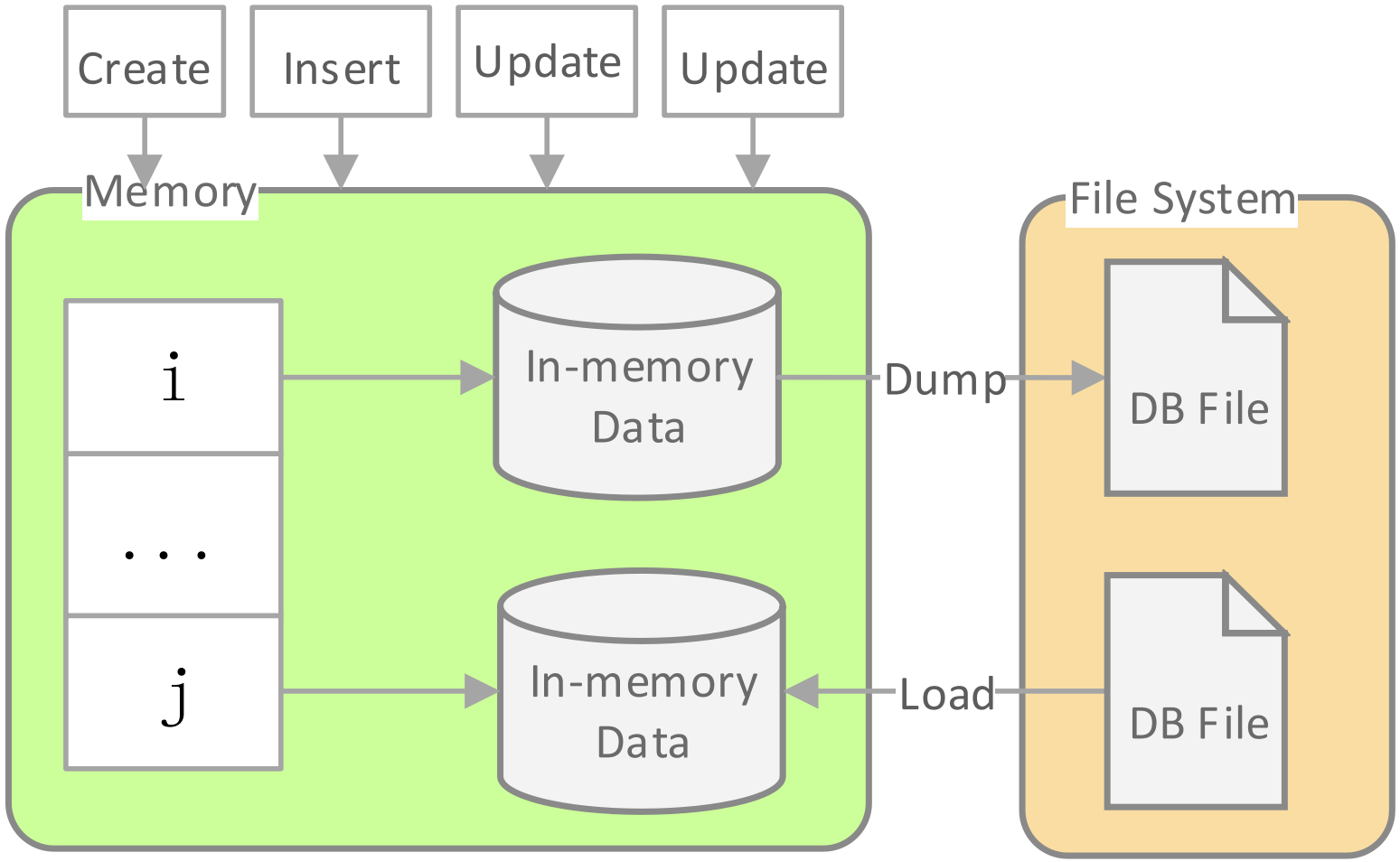
Project目标

开发一款程序库，在内存中启动一个或多个Key-Value Pair组成的数据库。

- 提供数据库基本的增、删、改操作；
- 提供数据库从内存Dump到文件系统功能；
- 提供数据库从文件系统Load到内存的功能
- *提供数据库的遍历和搜索功能
- *在数据库中添加索引
- *Compound file structure (复合文档结构)的文件结构实现

*为高级功能需求

整体实现架构


Create
Insert
Update
Update
Memory
i
...
j
In-memory Data
In-memory Data
Dump
Load
File System
DB File
DB File

接口

- CreateDatabase
  - 在内存中创建一个新的数据库
  - 以何种数据结构创建？ Vector? List? Map? Hashtable? …
- InsertKeyValue
  - 在数据库中插入Key/Value
  - 插入时是否允许键值重复？
- UpdateKeyValue
  - 更新数据库中的Key/Value
  - 考虑搜寻指定的Key所需要的复杂度

接口(续)

- DeleteKeyValue
  - 删除数据库中的Key/Value
  - 考虑搜寻指定的Key所需要的复杂度

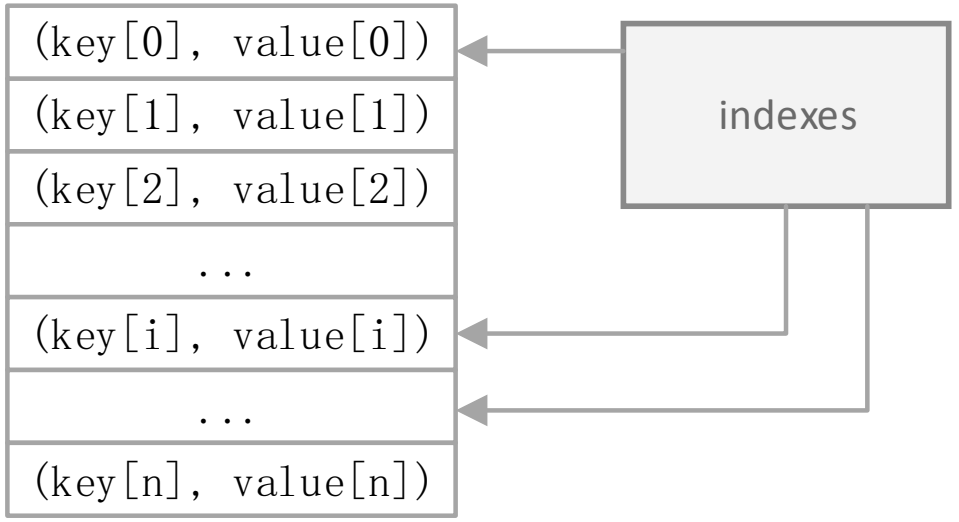
- QueryKeyValue
  - 在数据库中查找Key/Value
  - 查询效率取决于内存结构
  - 如果允许重复的Key，Query的结果应该是一组key/value
  - 可以利用索引提供查询效率

接口(续)

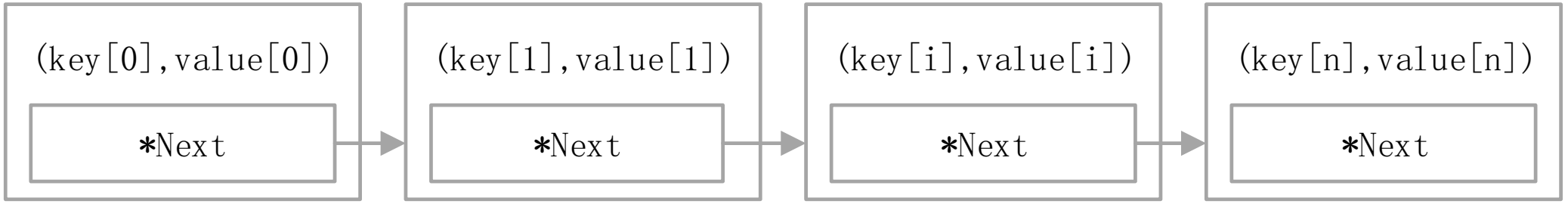
- DumpDatabase
  - 将内存数据库写入文件系统
  - 文件格式的定义？
  - 如果有多个Key/Value的内存库，如何存放？
- LoadDatabase
  - 将文件系统数据库加载至内存

内存的几种结构

- array of <key/value pair>
  - 线性分布
  - 可添加索引提高查询和搜索效率



- list<key/value pair>
  - 链表结构
  - 对于删除和查询有效率问题
  - 可添加索引提高搜索效率

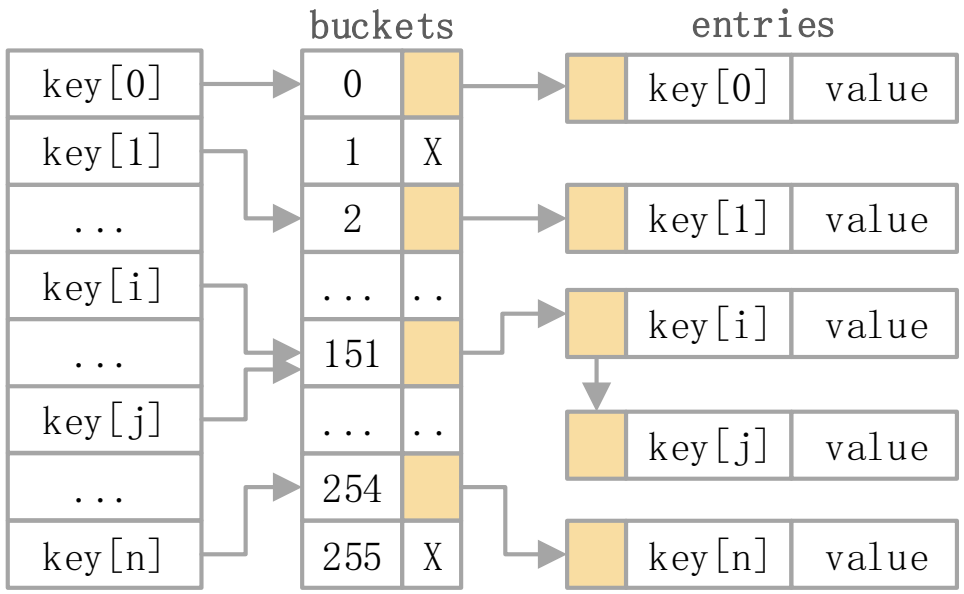


内存的几种结构(续)

- map<key, value>

key[0]	value[0]
key[1]	value[1]
...	
key[i]	value[i]
...	
key[n]	value[n]

- hashtable
  - hash函数的选择
  - 碰撞的解决方法
    - Separate chaining using linked list
    - Separate chaining with list head cells
    - ...
  - 查询具有较高效率



文件结构

- Flat file structure (平面文件结构)