

项目背景

- 电商平台众多
- 商品鱼龙混杂
- 商家与消费者间存在信息不对称

以上三点给消费者的网购带来了困扰。

——为了减轻商家与消费者间的信息不对称，给消费者更好的用户体验。

我们构建了一个商品检索平台

产品功能

商品检索平台由以下功能组成：

商户：可以管理、创建商店，为商店添加商品。

用户：可以对商品或标签进行搜索，找到心仪的商品。同时还有猜你喜欢功能。

更多见：https://github.com/Wh1isper/ProductSearch_Backend

产品实现

- 数据结构分为用户管理和商店管理两大块
- 搜索使用倒排索引
- 使用C++编写后端 QT编写前端实现
- 使用的数据结构：
 - 线性表
 - 动态表
 - 链表
 - 堆栈
 - 队列
 - 链式哈希表
 - 红黑树(std::set)

产品结构——用户管理



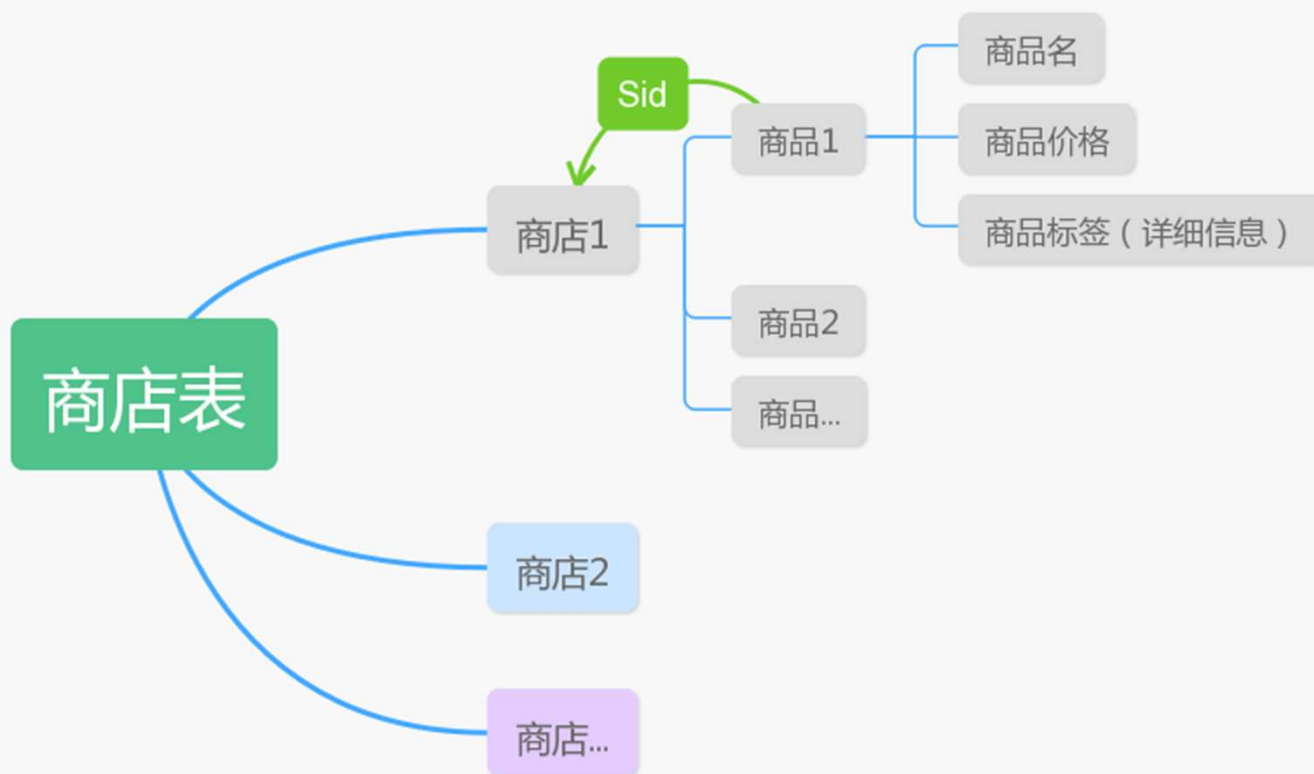
用户管理：采取不同的文件存储普通用户和商户用户

用户：使用链式哈希表查找用户，用户名是识别用户的唯一凭证

密码：采用MD5加密

用户喜好/拥有商店：采用链表存储

产品结构——商店管理

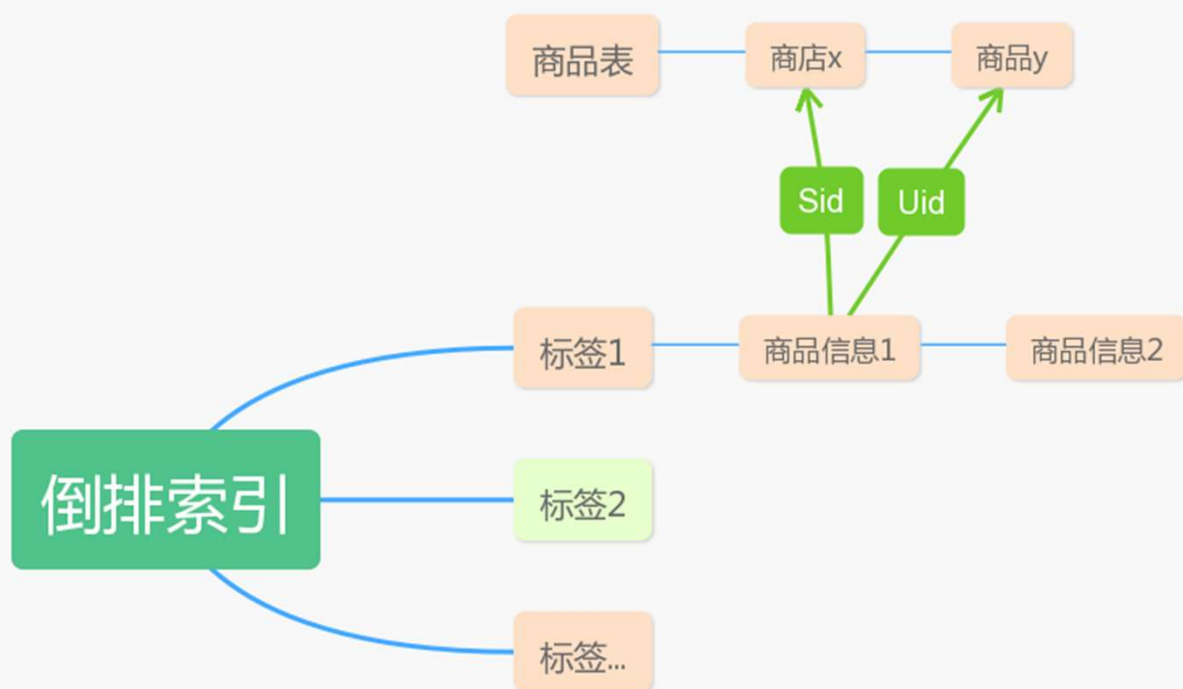


商店表采取动态表，
可以动态管理大小，
包含着所有商店。

商店中的商品也由
动态表管理。

商品标签为链表，
存储字符串。

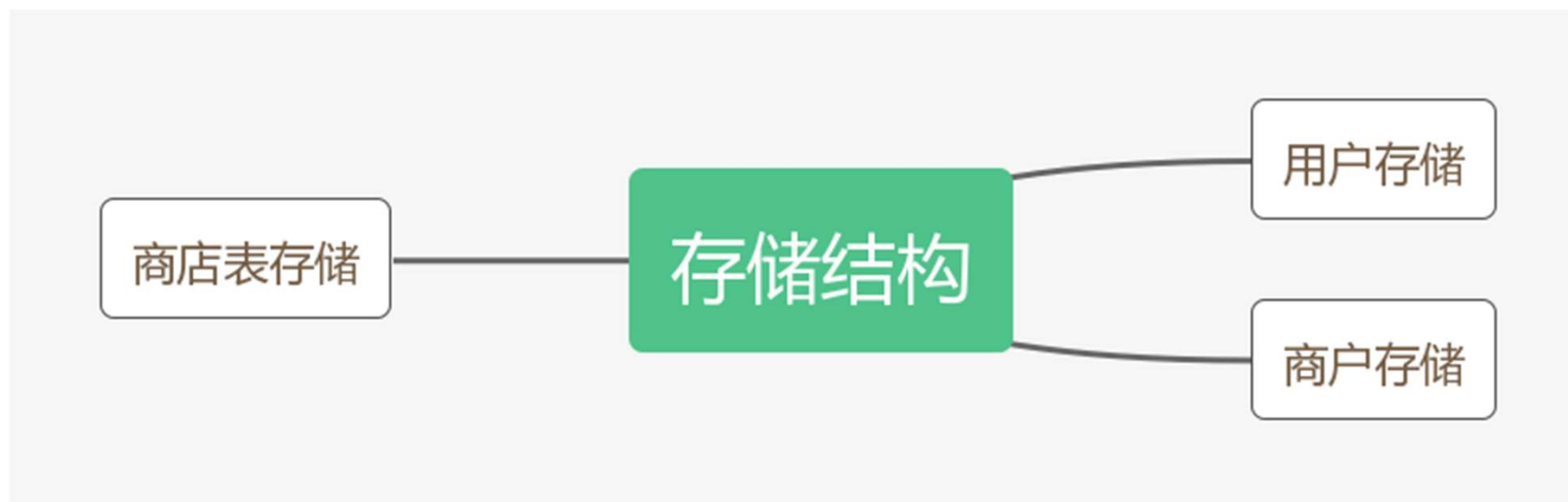
产品结构——搜索结构



遍历所有商品标签创建倒排索引

倒排索引存储内容是动态表中一一对应的位置（减少内存使用）

产品结构——存储结构



商店表采用索引-内容模式存储，由一个索引文件和多个商店文件组成。使得修改商店相关信息时不需要消耗很小。

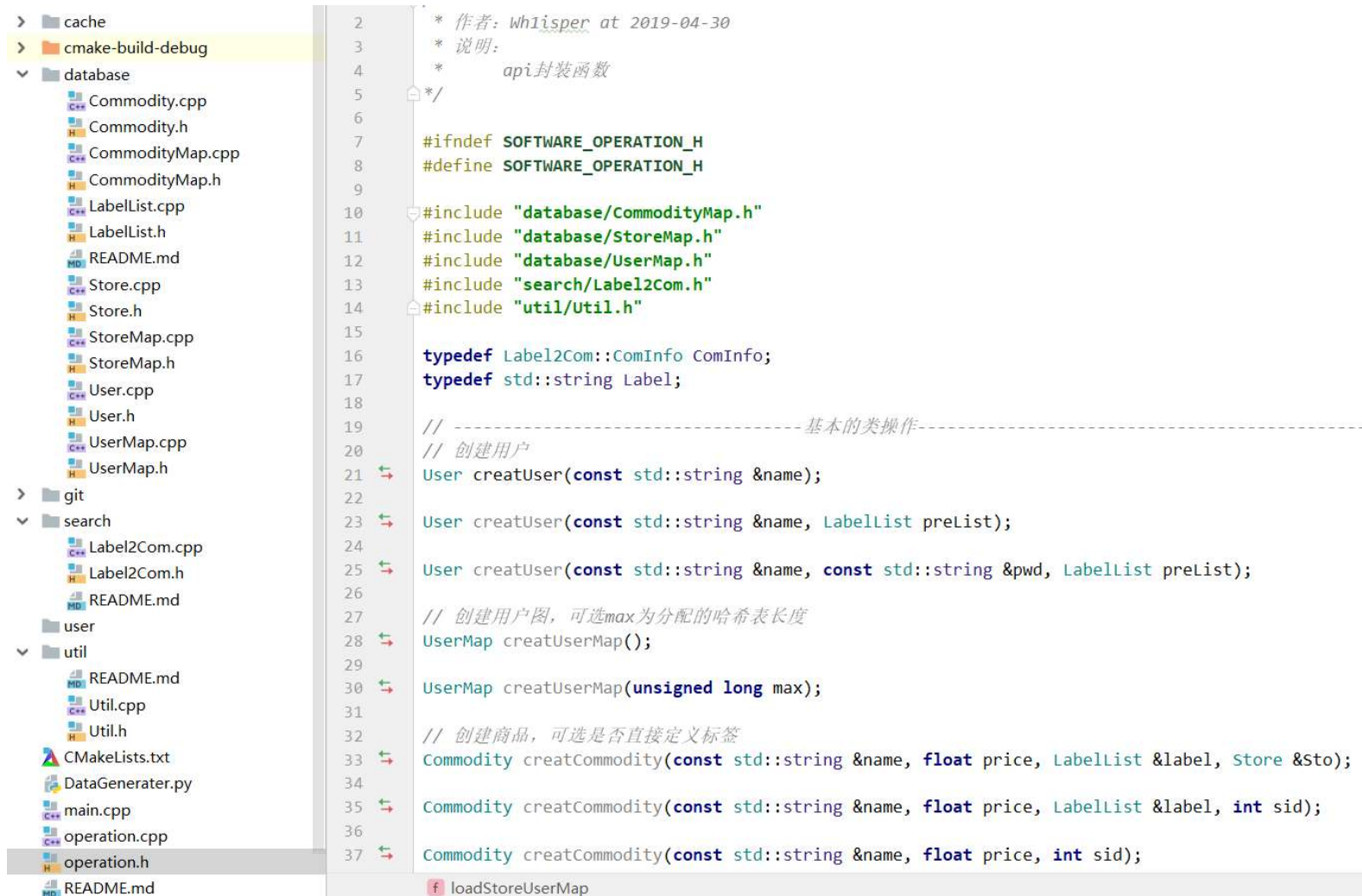
用户存储与商户存储一致，采用单一文件，密码为加密存储。

产品功能及算法实现

- 可视化的用户搜索与商户管理功能
- 各数据结构的功能（包括哈希表的扩展、动态表的扩展等）
- 猜你喜欢功能：
 - 在线算法更新用户喜好
 - 随机提供与喜好相关搜索结果
- 多标签搜索功能：
 - 深度优先搜索提取标签全排列
 - 红黑树合并集合
 - 动态表查找

产品优势

- 优秀的代码风格
- 可拓展API封装
- 完备的后端实现



The image shows a code editor with a project file structure on the left and a C++ source file on the right.

File Structure (Left):

- cache
- cmake-build-debug
- database
 - Commodity.cpp
 - Commodity.h
 - CommodityMap.cpp
 - CommodityMap.h
 - LabelList.cpp
 - LabelList.h
 - README.md
 - Store.cpp
 - Store.h
 - StoreMap.cpp
 - StoreMap.h
 - User.cpp
 - User.h
 - UserMap.cpp
 - UserMap.h
- git
- search
 - Label2Com.cpp
 - Label2Com.h
 - README.md
- user
- util
 - README.md
 - Util.cpp
 - Util.h
- CMakeLists.txt
- DataGenerator.py
- main.cpp
- operation.cpp
- operation.h
- README.md

Code (Right):

```
2  * 作者: Wh1isper at 2019-04-30
3  * 说明:
4  *      api封装函数
5  */
6
7  #ifndef SOFTWARE_OPERATION_H
8  #define SOFTWARE_OPERATION_H
9
10 #include "database/CommodityMap.h"
11 #include "database/StoreMap.h"
12 #include "database/UserMap.h"
13 #include "search/Label2Com.h"
14 #include "util/Util.h"
15
16 typedef Label2Com::ComInfo ComInfo;
17 typedef std::string Label;
18
19 // -----基本的类操作-----
20 // 创建用户
21 User createUser(const std::string &name);
22
23 User createUser(const std::string &name, LabelList preList);
24
25 User createUser(const std::string &name, const std::string &pwd, LabelList preList);
26
27 // 创建用户图, 可选max为分配的哈希表长度
28 UserMap createUserMap();
29
30 UserMap createUserMap(unsigned long max);
31
32 // 创建商品, 可选是否直接定义标签
33 Commodity createCommodity(const std::string &name, float price, LabelList &label, Store &sto);
34
35 Commodity createCommodity(const std::string &name, float price, LabelList &label, int sid);
36
37 Commodity createCommodity(const std::string &name, float price, int sid);
38
39 loadStoreUserMap
```

- https://github.com/Wh1isper/ProductSearch_Backend

产品不足

- 简陋的前端界面
- 未优化的性能
- 悬而未决的服务器交互功能

后端开发日记

- https://github.com/Wh1isper/ProductSearch_Backend
- 从4月23日开始，至5月27日
- 共11次更新
- 有完备的更新日志，每个模块下均有README描述
- 总代码量：2115行，注释超过200行，标明了每一个方法/api的使用
 - 商户管理、用户管理：966行
 - 搜索操作：368行
 - 文件操作：258行
 - api封装：398行
 - 实例函数：69行
 - Python模拟数据生成脚本：56行