Extendible Hashing 项目实验报告

10389048 杨帆
10389084 梁展瑞
12xxxxxx 古宣佑

May 13, 2012

1 摘要

2 实验环境与工具

1. Operating system: Linux(Debian & Arch & Ubuntu, with 3.0+ kernels)
2. Language: C
3. Compiler: GCC
4. Documentation: LaTeX, asciidoc, Markdown
5. Collaboration Platform/Project Hosting: GitHub
6. Debug utils: GDB, python, perf, gprof
7. Editor: VIM

这是在 GitHub 上的 repo，可以看到多个 branch 及历史记录。

`https://github.com/ddmbr/Extendible-Hashing/`

3 流程总览

整个流程可以划分为建立和查询两个大方面，分别叙述如下。(repo 有初期使用的草稿:draft.txt)

3.1 建立数据库

考虑到一个情况，当同时有大量的相同 key 的数据插入时，会导致一个桶无法分裂而必须拉链。但事前的统计发现，

1. key 相同的条目最多只有 7 条。
2. 依照我们的实现方式，每个桶的容量大概在 100 条。

所以我们的程序没有加入拉链的方法。以下是简要流程。

```
While the end of the raw file(i.e, lineitem.tbl) is not reached,
    Load a page from the raw file to the memory.
    Loop through the page, to:
        Read the next record in the page.
        Get its key.
        Get the hash value `hv' of the key.
        Fetch the corresponding index page,
        According to the index, fetch the corresponding
        bucket page.
        Before inserting the record, check that
        whether the bucket will be overflowed.
            If yes,
                If global depth == local depth,
                    Double the index
            Then split the bucket and redistribute.
        Write the record into the page.
```

3.2 查询

```
While not reach the end of the query,
    Read next specific key.
    Get the hash value `hv' of the key.
    Fetch the corresponding bucket
    Loop through the bucket and print the matched records.
```

4 架构

4.1 总览

程序大致划分为 4 个模块，分别为 File Manager, Buffer Manager, Parser 以及 Hash。

- File Manager 主要跟磁盘相关的操作，例如在磁盘上申请新页。
- Buffer Manager 处理跟内存管理相关的操作，例如换页等，时钟页面算法的核心在这个模块里。
- Parser 处理和源文本文件 (lineitem.tbl) 读取相关的操作。
- Hash 模块则处理和哈希算法相关的操作，含有 Extendible Hashing 的核心。

4.2 详细说明

Read the fucking source code.

5 结果与分析

6 总结