



中山大學
SUN YAT-SEN UNIVERSITY

《数据库系统》 实验报告

学院：软件学院

班级：教五五班（9）第八组

成员：周吉昊 张依帆 钟楚璇

时间：2015 年 6 月 19 日

项目题目：Sinew 论文中的 Catalog 和 Serializer 模块的简单实现

1. 实验环境与工具

实验环境：Ubuntu 14.04

编辑器：sublime text

编辑语言：C && C++

编译器：g++

版本控制及多人协作：GitHub

2. 实验思路

首先，实现对 json 文件的解析，利用 C 语言的字符串处理函数，将 json 的键值分离，并输出到 temp_files/ 文件夹里面，同时呢实现对嵌套对象和 数组（构想了，但未实现）的处理。

第二步，根据 /temp_files/keys 中的数据生成 Catalog，并将 catalog 输出到文件或者屏幕上。同时呢，用一个结构体 struct 数组存放 Catalog 的所有数据。

第三步，根据 keys 和 values，生成序列化记录 Seria，即论文中要求的格式。生成好之后，将 Seria 推送到 8KB 的缓冲区（字符数组），若缓冲区未满，继续生成 Seria，若缓冲区已满，则用 fwrite 将缓冲区中的数据写入二进制文件中去。

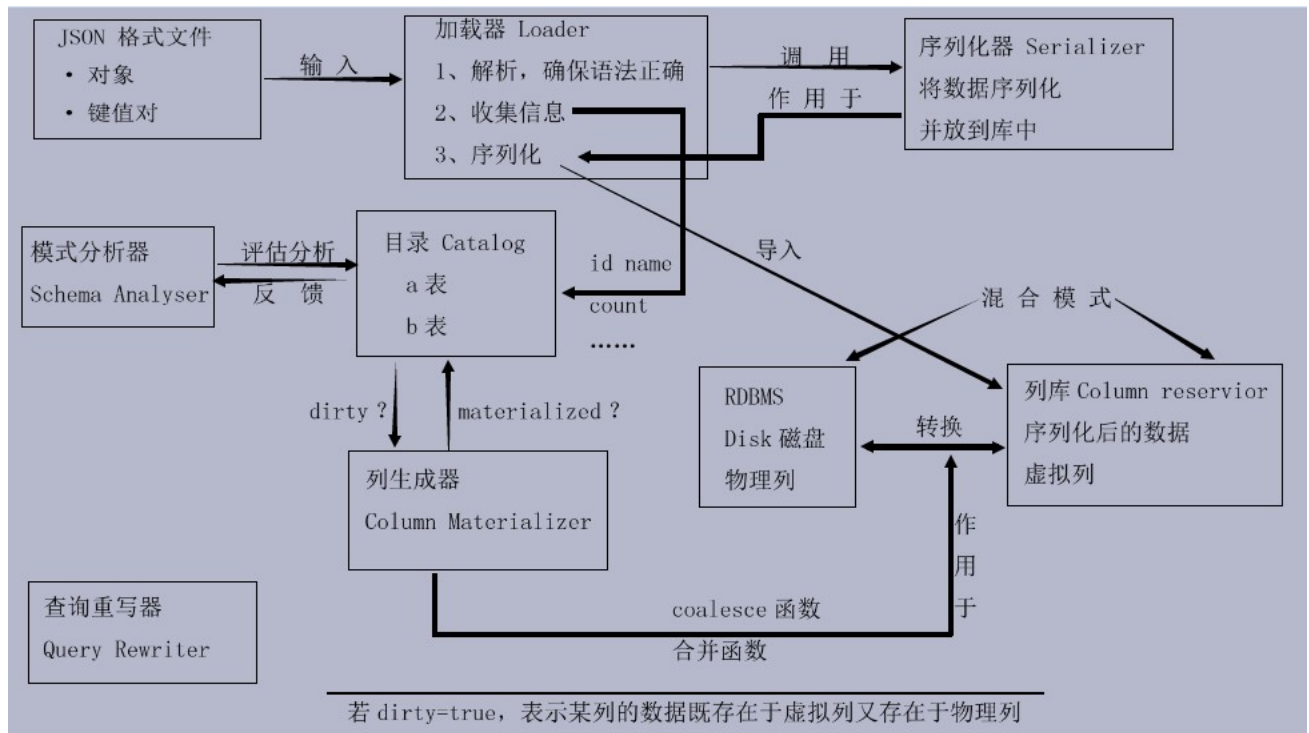
第四步，实现读文件，并将序列化的数据还原成 json 记录，方便查询，同时实现 find A = B，这其中利用到了 Catalog 和二分搜索的方法。

最后，整合所有文件，撰写主函数和 UI 界面，完成项目。

3. 实验过程设计

- 翻译 Sinew 论文
- 分析 Sinew 结构

分析 Sinew 结构的时候为了方便理解绘制出以下结构图



- 确定具体实现任务及分配工作
- 确定我们需要实现的是 Catalog 目录以及序列化器的相关内容。
- 确定 deadline 并投入工作
- 版本控制在 GitHub 中实现
- 整理工作结果并测试实验结果

4. 代码结构

```

|----- src
|
|----- Project_UI.cpp (main 函数)
|
|----- File.h File.cpp
|----- Catalog.h Catalog.cpp
|----- Serializer.h Serializer.cpp
|----- Makefile
|
|----- temp_files/
|----- keys
|----- values
|----- catalog
|----- serializer
|----- nestedarr

```

5. 实验结果与测试计划

开始执行

```
> Hello. Welcome to our DB Project.
> And the first thing you should do is to insert a file.
> And if you want to insert a new file, please make clean first.
> You can type command 'i' to start inserting a file. eg: nobench_data.json.
> After that you can check the catalog or find A = B.
> And if you find some bugs during using this program, just forgive us.
> We feel sorry for our mistakes.
> Hope you enjoy using it! :)
>
```

帮助界面（可用命令）

```
~~~~~Help Screen~~~~~
Available Commands:
'I'or'i':          Insert Filename.
'C'or'c':          Check Catalog.
'F'or'f':          Find A = B.
'H'or'h':          This Help Screen
'Q'or'q':          Quit The Program.
~~~~~
```

插入文件的过程，同时对文件进行处理

```
Now Please input your command.(one character)
> i
Command OK
Please input the file name.(relative path)
nobench_data.json
    ---> Succeed in opening the inFile: nobench_data.json
    ---> Insert File Completed!
    ---> Extract Keys To File Completed!
    ---> Extract Values To File Completed!
    ---> Generate Catalog Datas Completed!
    ---> Output Catalog Completed!
    ---> Serializer Gets The Catalog Completed!
    ---> Generate Serialized Data Completed!
Now you can check the catalog or find A = B.
```

Check Catalog 命令

```
> c
Command OK
You can open "./temp_files/catalog" to check the Catalog.
Also you can see it in the terminal.
Show On Screen? Y/N?
█
```

Catalog

	_id	Key_name	Key_type	count
1				
2				
3	0	nested_obj.num	Int	100000
4	1	nested_obj.str	String	100000
5	2	dyn2	Dynamic	100000
6	3	dyn1	Dynamic	100000
7	4	nested_arr	Nested_arr	100000
8	5	str2	String	100000
9	6	str1	String	100000
10	7	thousandth	Int	100000
11	8	sparse_087	String	1000
12	9	sparse_086	String	1000
13	10	sparse_085	String	1000
14	11	num	Int	100000
15	12	bool	Bool	100000
16	13	sparse_082	String	1000
17	14	sparse_081	String	1000
18	15	sparse_080	String	1000
19	16	sparse_079	String	1000

查询过程 find A = B

```
> f
Command OK

Notice: If you want to query nested object, just using the pointer separator.
For example: nested_obj.num, nested_obj.str
Now please input the keyname A: nested_obj.num
Notice: If the key is string, please include it in a pair of quotation marks.
For example: find nested_obj.str = "GBRDCMBQGA====="
Then input the value B: 72320
{"nested_obj": {"num": 72320, "str": "GBRDCMBQGAYTCMBRGAYTAMBQGAIDAMA="}, "sparse_209": "GBRDC==", "dyn2": false, "dyn1": "GBRDCMBRGAYTCMJQG
BQGAYA====", "nested_arr": [], "str2": "GBRDCMBQGAYTCMBRGAYTAMBQGAIDAMA=", "str1": "GBRDCMBRGAYTCMJQGAYTCMBQGAIDAMA====", "thousandth": 320, "nu
2320, "bool": false, "sparse_208": "GBRDC==", "sparse_205": "GBRDC==", "sparse_204": "GBRDC==", "sparse_207": "GBRDC==", "sparse_206": "G
==", "sparse_201": "GBRDC==", "sparse_200": "GBRDC==", "sparse_203": "GBRDC==", "sparse_202": "GBRDC=="},
> █
```

程序正常退出：

```
Now Please input your command.(one character)
> q
Program exits normally.
zhoujihao@DELL:~/Github/DatabaseProject/src$
```

6. 实验心得体会

通过这次实验，熟悉了 C++ 的面向对象特性，C 语言的字符数组处理特点，同时也理解了数据库查询的算法原理，对数据库中 Catalog 的概念也有了进一步的理解。

这次试验的难点在于缓冲区和序列化数据的处理，少有差错就可能全盘皆崩，刚开始设计的时候，由于思路没想好，结果造成了错误的结果，后来整理了思路，采用了结构更清晰的算法，问题才得以完成。所以，在以后的编码过程中，一定要明确自己的思路。

收获还是很多的，虽然过程略微艰辛，但有付出总有收获。

7. 小组信息和分工

教务五班 9，10 班联合组队，归属于 9 班第八小组。

周吉昊 13331371

- > 关于文件处理的一些接口。
- > Serializer 模块的实现。
- > 管理代码版本。

张依帆 13331345 钟楚璇 13331366

- > Catalog 模块的实现
- > project UI
- > 报告 pdf