

# 刘涛华

博客园 首页 新随笔 联系 订阅 管理

## 公告

昵称: Liutaohua  
园龄: 6个月  
粉丝: 1  
关注: 0  
[+加关注](#)

|    |         |    |    |    |    |    |   |
|----|---------|----|----|----|----|----|---|
| <  | 2020年6月 |    |    |    |    |    | > |
| 日  | 一       | 二  | 三  | 四  | 五  | 六  |   |
| 31 | 1       | 2  | 3  | 4  | 5  | 6  |   |
| 7  | 8       | 9  | 10 | 11 | 12 | 13 |   |
| 14 | 15      | 16 | 17 | 18 | 19 | 20 |   |
| 21 | 22      | 23 | 24 | 25 | 26 | 27 |   |
| 28 | 29      | 30 | 1  | 2  | 3  | 4  |   |
| 5  | 6       | 7  | 8  | 9  | 10 | 11 |   |

## 搜索

## 常用链接

[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)

## 我的标签

[IoTDB\(5\)](#)  
[数据库\(5\)](#)  
[时序数据\(4\)](#)  
[物联网\(4\)](#)  
[行式数据库\(3\)](#)  
[TsFile\(3\)](#)  
[车联网\(3\)](#)  
[列式数据库\(3\)](#)  
[InfluxDB\(2\)](#)  
[数据库性能\(2\)](#)

## 随笔档案

[2020年2月\(5\)](#)

## 最新评论

随笔 - 5 文章 - 0 评论 - 4

## 时序数据库 Apache-IoTDB 源码解析之文件数据块（四）

上一章聊到行式存储、列式存储的基本概念，并介绍了 TsFile 是如何存储数据以及基本概念。详情请见：

### 时序数据库 Apache-IoTDB 源码解析之文件格式简介（三）

打一波广告，欢迎大家访问[IoTDB 仓库](#)，求一波 Star 。欢迎关注头条号：列炮缓开局，欢迎关注[OSCHINA](#)博客

这一章主要想聊一聊：

1. TsFile的文件概览
2. TsFile的数据块

## TsFile文件概览



一个完整的 TsFile 是由图中的几大块组成，图中的数据块与索引块之间使用 1 个字节的分隔符 2 来进行分隔，这个分隔符的意义是当 TsFile 损坏的时候，顺序扫描 TsFile 时，依然可以判断下一个是 Metadata 是什么东西。

### 1. 识别符（Magic）

现在各种软件五花八门，很多软件都拥有自己的文件格式用来存储数据内容，但当硬盘上文件非常多的时候如何有效的识别是否为自己的文件，确认可以打开呢？经常用 windows 系统的朋友可能会想到用扩展名，但假如文件名丢失了，那我们如何知道这个文件是不是能被程序正确访问呢？

这时候通常会使用一个独有的字符填充在文件开头和结尾，这样程序只要访问 1 个固定长度的字符就知道这个文件是不是自己能正常访问的文件了，当然，TsFile 作为一个数据库文件，肯定需要在这个识别符上精心打造一番，它看起来是这样：

```
(decimal) 84 115 70 105 108 101
(hex)      54 73 46 69 6c 65
(ASCII)    T s F i l e
```

非常 cool 。

## 1. Re:时序数据库 Apache-IoTDB 源码解析之文件数据块（四）

楼主您好，我刚接触这个数据库一天 安装的时候有个问题卡住了 能加个微信吗。指点一下

--Smile\_灰太狼

## 2. Re:时序数据库 Apache-IoTDB 源码解析之文件数据块（四）

@daconglee 目前没有...

--Liutaohua

## 3. Re:时序数据库 Apache-IoTDB 源码解析之文件数据块（四）

有C#的读写代码吗？

--daconglee

## 4. Re:时序数据库 Apache-IoTDB 源码解析之前言（一）

有C#的读写代码吗？

--daconglee

### 阅读排行榜

1. 时序数据库 Apache-IoTDB 源码解析之文件索引块（五）（312）
2. 时序数据库 Apache-IoTDB 源码解析之系统架构（二）（227）
3. 时序数据库 Apache-IoTDB 源码解析之前言（一）（200）
4. 时序数据库 Apache-IoTDB 源码解析之文件数据块（四）（152）
5. 时序数据库 Apache-IoTDB 源码解析之文件格式简介（三）（125）

### 评论排行榜

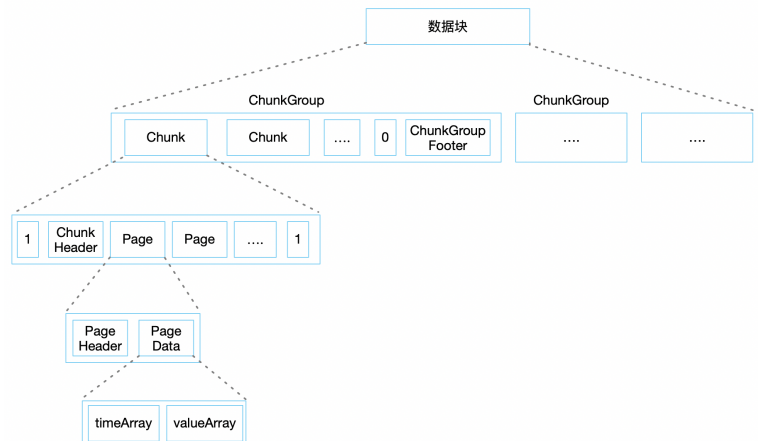
1. 时序数据库 Apache-IoTDB 源码解析之文件数据块（四）（3）
2. 时序数据库 Apache-IoTDB 源码解析之前言（一）（1）

## 2.文件版本（Version）

再精妙的设计也难免产生一些问题，那么就需要升级，那么文件内容也一样，有时候当你的改动特别大了，就会出现完全不兼容的两个版本，这个很好理解不过多解释。TsFile 中采用了 6 个字节来保存文件版本信息，当前 0.9.x 版本看起来就是这样：

```
(decimal)  48  48  48  48  48  50
(hex)      30  30  30  30  30  32
(ASCII)    0   0   0   0   0   2
```

## 3.数据块



### 3.1 ChunkGroup

文件的数据块中包含了多个 ChunkGroup，其中 ChunkGroup 的概念已经在上一章聊过，它代表了设备(逻辑概念上的一个集合)一段时间内的数据，在 IoTDB 中称为 Device。

在实际的文件中，ChunkGroup是由多个 Chunk 和一个 ChunkGroupFooter 组成。其中最后一个 Chunk 的结尾和 ChunkGroupFooter 之间使用 1 个字节的分隔符 0 来做区分，ChunkGroupFooter 没有什么具体作用，不做详细解释。

### 3.2 Chunk

一个 ChunkGroup 中包含了多个 Chunk，它代表了测点数据(逻辑概念上的某一类数据的集合,如体温数据)，在 IoTDB 中称为 Measurement。

在实际文件中 Chunk 是由 ChunkHeader 和多个 Page 组成，并被 1 个字节的分隔符 1 包裹。ChunkHeader中主要保存了当前 Chunk 的数据类型、压缩方式、编码方式、包含的 Pages 占用的字节数等信息。

### 3.3 Page

一个 Chunk 中包含多个 Page，它是一个数据组织方式，数据大小被限制在 64K 左右。

在实际文件中由 PageHeader 和 PageData 组成。其中 PageHeader 里主要保存了，当前 page 里的一些预聚合信息，包含了最大值、最小值、开始时间、结束时间等。他的存在是非常有意义的，因为当某些特定场景的读时候，不必要解开 page 的数据就能够得到结果，比如说 `selece 体温 from 王五 where time > 1580950800`，当读到 PageHeader 的时候，找到 `startTime` 和 `endTime` 就能判断是否可以使用当前 page。这个聚合信息的结构同样出现在索引块中，下一章再具体聊这个聚合结构。

### 3.4 PageData

一个 Page 中包含了一个 PageData，里面有两个数组：时间数组和值数组，且这两个数组的下标是对齐的，也就是时间数组中的第一个对应值数组中的第一个。举个例子：

```
timeArray: [1,2,3,4]
valueArray: ['a', 'b', 'c', 'd']
```

在page中就是这样保存的数据，其中 1 代表了时间 1970-01-01 08:00:00 后的 1 毫秒，对应的值就是 'a'。

### 数据块展示

我们继续使用上一章聊到的示例数据来展示真正的TsFile中是如何保存的。

| 时间戳        | 人名 | 体温   | 心率  |
|------------|----|------|-----|
| 1580950800 | 王五 | 36.7 | 100 |
| 1580950911 | 王五 | 36.6 | 90  |

当数据被写入 TsFile 中，大概就是下面一个展示的情况，这里省略了索引部分。

```

POSITION|  CONTENT
-----|  -----
        0|  [magic head] TsFile
        6|  [version number] 000002
        // 因为 6个字节的magic + 6个字节的
version 所以 chunkGroup 从 12 开始
||||| [Chunk Group] of wangwu be
gins at pos 12, ends at pos 253, version:0, num of
Chunks:2

        // 这里展示的是 ChunkHeader 中保存
的信息

        12|  [Chunk] of xinlv, numOfPoi
nts:1, time range:[1580950800,1580950800], tsDataT
```

```

type:INT32,
                                [minValue:100,maxValue:100
,firstValue:100,lastValue:100,sumValue:100.0]
                                | [marker] 1 //
chunk 的真正开始是从这个分隔符 1 开始的
                                | [ChunkHeader] /
/ header 的数据在上面展示了
                                | 1 pages //
这里保存的具体数据
                                | time:1580950800; v
alue:100
                                // 下一个 chunk
                                121| [Chunk] of tiwen, numOfPoi
nts:1, time range:[1580950800,1580950800], tsDataT
ype:FLOAT,
                                [minValue:36.7,maxValue:36
.7,firstValue:36.7,lastValue:36.7,sumValue:36.7000
0076293945]
                                | [marker] 1
                                | [ChunkHeader]
                                | 1 pages
                                | time:1580950800; v
alue:36.7
                                230| [Chunk Group Footer]
                                | [marker] 0 // chun
kFooter 和 chunk 使用 0 作为分隔
                                | [deviceId] wangwu
                                | [dataSize] 218
                                | [num of chunks] 2
|||||||||||||||| [Chunk Group] of wangwu en
ds

```

回想我们的查询语句 `select 体温 from 王五`，当经历过索引之后会得到 `offset` 的值等于 121，这时候我们只需要调用 `reader.seek(121)`，从这里开始就是所有体温数据的开始点，从这里一直读到 230 的 `ChunkGroupFooter` 结构的时候，就可以返回给用户数据了。

有兴趣自己实验的朋友可以，引入 `TsFile` 的包，自行实验，下面给出测试代码：

```

<dependency>
    <groupId>org.apache.iotdb</groupId>
    <artifactId>tsfile</artifactId>
    <version>0.9.1</version>

```

```
</dependency>

public static void main(String[] args) throws IOException, WriteProcessException {
    MeasurementSchema chunk1 = new MeasurementSchema(
        "tiwen", TSDataType.FLOAT, TSEncoding.PLAIN);
    MeasurementSchema chunk2 = new MeasurementSchema(
        "xinlv", TSDataType.INT32, TSEncoding.PLAIN);

    Schema chunks = new Schema();
    chunks.registerMeasurement(chunk1);
    chunks.registerMeasurement(chunk2);

    TsFileWriter writer = new TsFileWriter(new File(
        "test"), chunks);

    RowBatch chunkGroup = chunks.createRowBatch("wangwu");

    long[] timestamps = chunkGroup.timestamps;
    Object[] values = chunkGroup.values;

    timestamps[0] = 1580950800;
    float[] tiwen = (float[]) values[0];
    int[] xinlv = (int[]) values[1];

    // 写入王五的体温
    tiwen[0] = 36.7f;
    //写入王五的心率
    xinlv[0] = 100;
    chunkGroup.batchSize++;

    timestamps[1] = 1580950800;
    // 写入第二条王五的体温
    tiwen[1] = 36.6f;
    //写入第二条王五的心率
    xinlv[1] = 90;
    chunkGroup.batchSize++;

    writer.write(chunkGroup);
    writer.close();
}
```


执行完成之后你可以使用 IoTDB 中的 TsFileSketchTool 来查看文件结构，得到文中示例的展示结果；或者使用 od 等工具查看，祝玩

儿的开心。[IoTDB 0.9.1 版本下载](#)

这一章聊到了 TsFile 分为了 数据块 和 索引块，并且介绍了数据块的具体组成部分和查询逻辑。那么索引块是什么结构，怎样完成了在大量混杂的数据中搜索到的想要的的数据，请持续关注。

标签: [TsFile](#), [行式数据库](#), [时序数据](#), [IoTDB](#), [列式数据库](#), [车联网](#), [物联网](#), [数据库](#)



 [Liutaohua](#)  
 关注 - 0  
 粉丝 - 1  
[+加关注](#)

0 0

« 上一篇: [时序数据库 Apache-IoTDB 源码解析之文件格式简介（三）](#)

» 下一篇: [时序数据库 Apache-IoTDB 源码解析之文件索引块（五）](#)

posted @ 2020-02-12 12:06 [Liutaohua](#) 阅读(152) 评论(3) 编辑 收藏

## 评论列表

#1楼 2020-03-08 10:34 [daconglee](#)

有C#的读写代码吗？

支持(0) 反对(0)

#2楼 [楼主] 2020-03-09 00:37 [Liutaohua](#)

@daconglee  
 目前没有

支持(0) 反对(0)

#3楼 2020-05-14 10:37 [Smile\\_灰太狼](#)

楼主您好，我刚接触这个数据库一天 安装的时候有个问题卡住了 能加个微信吗。指点一下

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)， [访问](#) 网站首页。

**【推荐】** [超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库](#)

**【推荐】** [20本必看的阿里精品免费电子书](#)

#### 相关博文：

- [时序数据库Apache-IoTDB源码解析之文件索引块（五）](#)
- [时序数据库Apache-IoTDB源码解析之文件索引块（五）](#)
- [时序数据库Apache-IoTDB源码解析之文件格式简介（三）](#)
- [松果时序数据库压缩文件格式](#)
- [时序数据管理引擎ApacheIoTDB](#)
- » [更多推荐...](#)

#### 最新 IT 新闻：

- [GitHub 开源 Super Linter，用自动化解决开发者的需求](#)
- [量子计算机领域内第一种高级编程语言 Silq 诞生](#)
- [微软正式推出 gRPC-Web for .NET](#)
- [Visual Studio Code 6 月 Python 扩展更新](#)
- [再见 Python，你好 Julia!](#)
- » [更多新闻...](#)