

# L0 → L1 compaction 测试数据

## Legacy

### 概览

compaction 总次数=2315

现实用时=55min(其实还有约 1/4 的 shard 依然在处理大量的 L0 文件)

根据网页端观察，平稳阶段 L1 文件数约为 6.5

(下文中有两个元素的 Array 表示 L0、L1 的数值)

### 详细

1. 总消费文件数=[9202,6009] avg=[3.97,2.60]

3.97 个 L0 文件：非常贴合 pick file 配置 (4,4,8,2)；

2.60 个 L1 文件：相比 L0 消费数目，可以说是占比很大了

2. 消费文件大小=[18718,78472] avg=[8.09,33.90]

由于读 L1 算“浪费”，所以此处读扩大为 419%

3. 平均单个文件大小=[2.03,13.06]

L0 文件大小符合配置项，L1 生成的时候可能有零头，也不得不新建文件，小于 16MB 也是预期效果

4. 总共节省文件容量（压缩优化）=14207MB avg=6.14MB

5. 总用时（CPU）=22906 秒=381.8min avg=9.89 秒/次

在这 8.05 秒，平均每次做 41.99MB 的文件读，写入 35.85MB

6. 根据输入的 L1 文件的个数分别统计 compaction 数据

| 使用 L1 文件个数 | compaction 次数 | compaction 总用时 | 平均每次用时           |
|------------|---------------|----------------|------------------|
| 0          | 51            | 14413          | 282.6078431373   |
| 1          | 316           | 891891         | 2822.4398734177  |
| 2          | 713           | 4903831        | 6877.7433380084  |
| 3          | 677           | 7753863        | 11453.2688330872 |
| 4          | 554           | 9264321        | 16722.6010830325 |
| 5          | 4             | 77820          | 19455            |

对于使用相同的 L1 文件数目，Legacy 的单次用时短于 Experimental（见下），这是由于 pick 策略差，用的 L0 文件数目少。

## Experimental

### 概览

compaction 总次数=1217

现实用时=25min

根据网页端观察，平稳阶段 L1 文件数约为 6

详细

- 1. 消费文件数=[9584,2551] avg=[7.88,2.10]  
7.88 个 L0 文件：显著高于 pick file 配置 (4,4,8,2) ，应该优化算法，避免产生过大的 pick；  
2.10 个 L1 文件：本算法思想就是减少每次涉及的 L1 文件数，<2 应该是基本预期效果
- 2. 消费文件大小=[18702,31372] avg=[15.37,25.78]  
由于读 L1 算“浪费”，所以此处读扩大为 168%
- 3. 单个文件大小=[1.95,12.30]  
L0 文件大小符合配置项，L1 生成的时候可能有零头，也不得不新建文件，小于 16MB 也是预期效果
- 4. 总共节省文件容量（压缩优化）=14194MB avg=11.66MB
- 5. 总用时（CPU）=9797 秒=163.3min avg=8.05 秒/次  
在这 8.05 秒，平均每次做 40.25MB 的文件读，写入 28.59MB
- 6.

| 使用 L1 文件个数 | compaction 次数 | compaction 总用时 | 平均每次用时           |
|------------|---------------|----------------|------------------|
| 0          | 40            | 18393          | 459.825          |
| 1          | 383           | 1362706        | 3557.9791122715  |
| 2          | 365           | 2821458        | 7730.0219178082  |
| 3          | 286           | 3249375        | 11361.451048951  |
| 4          | 137           | 2311579        | 16872.8394160584 |
| 5          | 4             | 75456          | 18864            |
| 6          | 2             | 57541          | 28770.5          |

显然，用上了不多于 2 个 L1 文件的 compaction 用时短。应该控制 L1 文件使用数目。

（此处应当加上 compaction 的读写文件大小，因为使用 L1 多的话，使用 L0 应该也多，考虑这个因素才好说“效率”。

不过，从理论上分析也是应当减少涉及的 L1 文件数目比较好，或者说是提高 L0 文件与 L1 文件的比例。）

=> 下个 Metrics 版本会按照使用 viceSegments 数目统计消费的 main/viceSegment 总容量、数量，这样就能了解其“效率”。

但是不会优先重写测试报告