

KYLIN-5319 查询元数据变更导致NPE

Root Cause

通过分析Audit Log与Query Log。
其实可以基本上还原事故现场。

先说结论

这个问题是因为FilePruner中DataFlow变量的Segment元数据懒加载导致的。
因为懒加载导致dataflow中元数据不一致。

事故现场:

1. **2022-09-16 09:57:04.967** 开始Merge Segment。新建了Segment 6e30a9be-dc46-5352-441e-811668d822cf。锁定了以下Segment

```
1 aea0a7d1-fed4-88ad-1a41-7be1411f411f
2 87e44fee-a4ed-c267-bd38-77d358e96850
3 9a08f63e-9014-ea56-24f5-fa4cc9764b51
4 f6523569-070e-6dab-67f3-5f057f06b8d5
5 0c4caf8b-df5d-bcec-05cb-fd4c7af10822
6 d1f187cd-6b02-a1f4-0538-f09d258b73fa
7 8e4a29a1-3d22-1fe5-c1e0-40959f13fad9
8 79ed5a6d-80c7-47cb-ab43-78105f1c6f5a
9 4b36f5ce-ccd2-e3d0-7d48-55c619c8cb4f
```

2. 查询开始前 DataFlow 83e9c060-d16e-806d-75e9-00c60d9a136d 拥有 19 个 Segment + 1 个 Merging Segment。

```
1 ## Ready
2 d4d07fa2-8225-19a8-b119-93d99496e071
3 07c0d59f-ed4f-bc6b-7807-866d3c4effc8
4 3c9d1299-3fc7-a583-7b63-413ec8044b34
5 d61bd6ed-e3e2-a08f-8c14-88f760ba5e0a
6 5b1e47d5-87ab-fd66-bd74-6c4570126e0d
7 637de76f-cf88-c522-3d4a-3030b5356aec
8 caee2f7f-24e9-0440-af2e-ce38fa35b44a
9 470afafc-43d9-16ff-a332-8cf5ef387e39
10 29064d74-769f-d47d-d8ad-5f8b59d6e794
11 8cebe35d-193a-3f2d-6483-f931bac450d3
12
13 ## Lock
14 4b36f5ce-ccd2-e3d0-7d48-55c619c8cb4f
15 79ed5a6d-80c7-47cb-ab43-78105f1c6f5a
16 8e4a29a1-3d22-1fe5-c1e0-40959f13fad9
17 d1f187cd-6b02-a1f4-0538-f09d258b73fa
```

```
18 0c4caf8b-df5d-bcec-05cb-fd4c7af10822
19 f6523569-070e-6dab-67f3-5f057f06b8d5
20 9a08f63e-9014-ea56-24f5-fa4cc9764b51
21 87e44fee-a4ed-c267-bd38-77d358e96850
22 aea0a7d1-fed4-88ad-1a41-7be1411f411f
23
24 ## New
25 6e30a9be-dc46-5352-441e-811668d822cf
```

3. 2022-09-16 09:57:13,371 开始查询，SQL语句并没有按照时间分区过滤。
4. 2022-09-16 09:57:13,629 Segment Pruning后是可查询的19个Segment。
5. 2022-09-16 09:57:13,632 初始化FilePruner#dataflow，这个dataflow里面有10个Ready Segment + 9个Lock Segment + 1个New Segment。
 - a. 为什么博文之前的Segment校验在这里不生效，是因为这个当前时间点校验是通过的。
6. 2022-09-16 09:57:13.670 合并完成，新的Segment 上线
7. 2022-09-16 09:57:13.682 合并完成，删除索引Segment
8. 2022-09-16 09:57:13,703 当前节点同步10条元数据完成。

```
881 2022-09-16T09:57:13.700 INFO [http-nio-7070-exec-9] service.AuditLogService : Start to catchup manually
882 2022-09-16T09:57:13.700 INFO [http-nio-7070-exec-9] service.AuditLogService : End to catchup manually
883 2022-09-16T09:57:13.701 DEBUG [ReplayWorker-p-7-t-1] transaction.AuditLogReplayWorker : start restore from (587917,587927]
884 2022-09-16T09:57:13.702 DEBUG [ReplayWorker-p-7-t-1] transaction.AbstractAuditLogReplayWorker : replay 10 event for project:MARKETING
885 2022-09-16T09:57:13.702 DEBUG [ReplayWorker-p-7-t-1] transaction.UnitOfWork : UnitOfWork a9ae7fb4-48eb-52a6-3182-64329a88ed58 started on project MARKETING
886 2022-09-16T09:57:13.702 DEBUG [ReplayWorker-p-7-t-1] transaction.UnitOfWork : UnitOfWork a9ae7fb4-48eb-52a6-3182-64329a88ed58 takes 0ms to complete
887 2022-09-16T09:57:13.703 DEBUG [ReplayWorker-p-7-t-1] transaction.AuditLogReplayWorker : end restore from (587917,587927], delay queue:0
888 2022-09-16T09:57:15.852 DEBUG [MetaStoreHealthChecker-p-9-t-1] transaction.UnitOfWork : UnitOfWork 0af6574b-e814-7d02-113e-17310f88a829 started on project _health
889 2022-09-16T09:57:15.886 DEBUG [MetaStoreHealthChecker-p-9-t-1] transaction.UnitOfWork : transaction 0af6574b-e814-7d02-113e-17310f88a829 updates 2 metadata items
890 2022-09-16T09:57:15.888 DEBUG [MetaStoreHealthChecker-p-9-t-1] transaction.UnitOfWork : UnitOfWork 0af6574b-e814-7d02-113e-17310f88a829 takes 2ms to complete
```

9. 2022-09-16 09:57:13,769 FilePruner#listFiles 循环扫描19个Segment

- a. 扫描到在Merge中被删除的Segment时

```
val totalFileSize = selected.flatMap(partition => partition.files).map(_.getLen).sum
val sourceRows = selected.map(seg => { 这里不报错 这里拿到的是Null 导致NPE
    val layoutRows = dataflow.getSegment(seg.segmentID).getLayout(layout.getId).getRows Song, 2022/5/6, 16:22
    logInfo(msg = s"Source scan rows: Query Id: ${QueryContext.current().getQueryId}, Segment Id: ${seg.segmentID}
    s"Layout Id: ${layout.getId}, rows: $layoutRows.")
    layoutRows
}).sum
```

- a. dataflow.getSegment(seg.segmentID) 不会报错。因为这个dataflow是旧的，segmentID也是旧的。
- b. 问题在于 getLayout(layoutId) 拿到的Layout是个 Null 对象，导致调用 getRows 方法报错NPE

10. 我们看下面这段代码

- a. 之前有过一个元数据加载优化Design，其中将DataFlow中的SegmentDetail和LayoutMap包装进LayoutInfo内部类，改造成懒加载。
- b. 因为懒加载，所以导致用到时，去最新的元数据中加载了SegmentDetail。因为SegmentDetail已经被删了，所以导致这里拿到的是个空对象。
- c. FilePruner#dataflow初始化时，dataflow内的segment内的layoutinfo没有被加载。造成了现在的问题。

7 usages fenghuang

```
private LayoutInfo getLayoutInfo() { Fenghuang, 2022/4/20, 16:06
    if (layoutInfo == null) {
        synchronized (this) {
            if (layoutInfo == null) {
                layoutInfo = new LayoutInfo(loadDetail: true);
            }
        }
    }
    return layoutInfo;
}
```

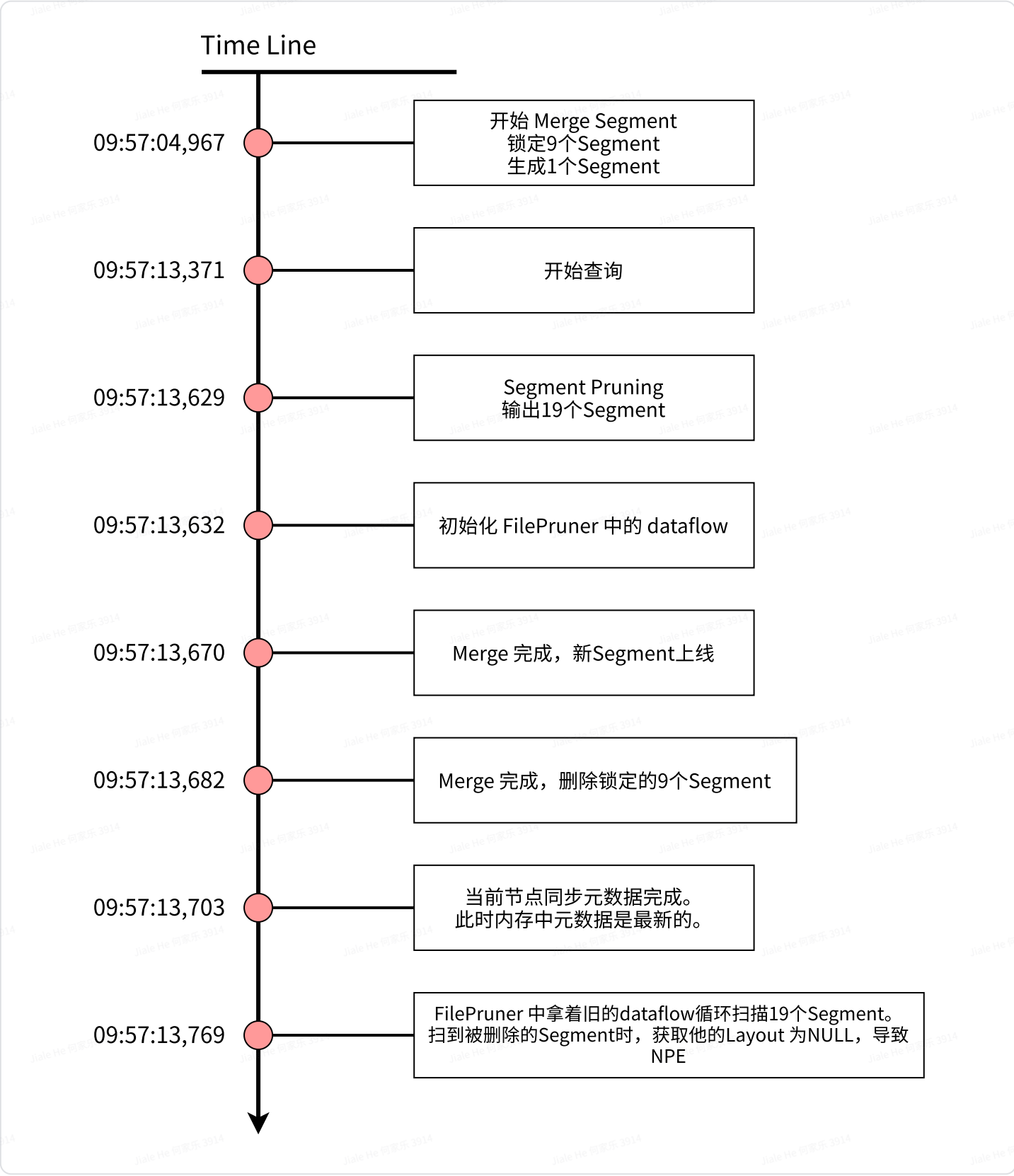
```
private LayoutInfo(boolean loadDetail) {
    if (!loadDetail) {
        return;
    }
    segDetails = NDataSegDetailsManager.getInstance(getConfig(), dataflow.getProject())
        .getForSegment(NDataSegment.this);
    if (segDetails == null) {
        segDetails = NDataSegDetails.newSegDetails(dataflow, id);
    }

    IndexPlan indexPlan = dataflow.getIndexPlan();
    if (!indexPlan.isBroken()) {
        List<NDataLayout> filteredCuboids = segDetails.getLayouts().stream()
            .filter(dataLayout -> dataLayout.getLayout() != null).collect(Collectors.toList());
        segDetails.setLayouts(filteredCuboids);
    }

    segDetails.setCachedAndShared(dataflow.isCachedAndShared());
    List<NDataLayout> cuboids = segDetails.getLayouts();
    LayoutsMap = new HashMap<>(cuboids.size());
    for (NDataLayout cuboid : cuboids) {
        layoutsMap.put(cuboid.getLayoutId(), cuboid);
        Map<Long, Long> cuboidBucketMap = Maps.newHashMap();
        cuboid.getMultiPartition().forEach(dataPartition -> cuboidBucketMap.put(dataPartition.getPartitionId(),
            dataPartition.getBucketId()));
        partitionBucketMap.put(cuboid.getLayoutId(), cuboidBucketMap);
    }
}
```

最新的元数据中，SegmentDetail 已被删除
所以这里拿到的是Null

11. 时间线分析请看下面流程图



Fix Design

在FilePruner#dataflow 初始化时，同时进行PrunedSegments的初始化。

