```
/**
 * 根据允许选取的主键范围，筛选出可以参与 compaction 的 L0 文件
 * @param minRowKey 主键最小值
 * @param maxRowKey 主键最大值
 * @param level0Files 所有 L0 文件
 * @return 筛选出的可进行 compaction 的 L0 文件
 */
private List<FileMeta> calcPlan(byte[] minRowKey, byte[] maxRowKey, List<FileMeta> level0Files) {
    List<FileMeta> ret = new ArrayList<>();
    if (level0Files.size() == 0 || CommonUtils.compareByteArray(minRowKey, maxRowKey) >= 0) return ret;

    long totalSize = 0L;
    byte[] lBorder = minRowKey, rBorder = maxRowKey;

    for (int ind = 0; ind < level0Files.size(); ind++) {

        long currentMajorId = level0Files.get(ind).getMajorId();

        // Process level0Files with same walId and find the segments that can be added
        List<Integer> addedInd = new LinkedList<>();
        while (ind < level0Files.size() && level0Files.get(ind).getMajorId() == currentMajorId) {
            FileMeta l0File = level0Files.get(ind);
            if (CommonUtils.contains(lBorder, rBorder,
                    l0File.getStartRecord().getKey(),
                    l0File.getEndRecord().getKey(),
                    true, true)) {
                totalSize += l0File.getFileSize();
                addedInd.add(ind);
                ret.add(l0File);

            }
            ind++;
        }
        ind--;

        // Update lBorder and rBorder
        if (addedInd.size() > 0) {
            Integer firstAdded = addedInd.get(0);
            Integer lastAdded = addedInd.get(addedInd.size() - 1);

            if (firstAdded > 0) {
                FileMeta preAdd = level0Files.get(firstAdded - 1);
                if (preAdd.getMajorId() == currentMajorId) {
```