Goals

1. Don't use the MR.
2. Remove deleted mob cells.
3. Avoid write amplification due to rewriting files.
4. Allows for consolidation of mob files.

In the sweep tool, we use MR to scan the HBase table to find the mob files that are still referenced by the HBase whereby we could find the small or invalid mob files. In order to avoid the race condition between the major compaction and sweep tool, the Zookeeper is used to synchronize them.

For the invalid and small files, we divide them into different groups by the start key and date which is the latest cell saved in that mob file, in each group, the small/invalid files are merged, the partitioning and merging are done in the reducers.

In the mob compaction we should avoid scanning tables, instead we directly scan the mob files. We have to save the del mob files (Saving the delete markers or deleted cells will be discussed later), so in the scanning of mob files, we know which cells are deleted from the mob files.

The partition approach could be used in the mob compaction too. In the mob compaction, we should scan all the existing mob files and partition them, in each partition we do the merges. All of these could be done in HM in the first cut, in the future scanning could be done in HM and the merges could be distributed to region servers. Considering the race condition, I am afraid we have to use the Zookeeper to synchronize the major compaction and mob compaction.

The pros to do the mob compaction out of regions:

1. Only read the information of mob files once.
2. Do not care about the existing regions and region split/merge.

The cons:

1. The lock for the race condition between major compaction and mob compaction is coarser grained, we have to lock all the major compactions in all the regions when the mob compaction is running.

1 How to save the del files?

We could save deleted cells or delete markers in the del mob files.

1. The deleted cells should be the ones deleted by the delete markers. The expired cells by TTL are not saved in the del files, they should be handled by the ExpiredMobFileCleaner. In both minor and major compactions cells might be deleted, in each of them a del file is created.
2. If only the delete markers are save in the del files, we only need to create a del file after each major compaction, and the file size and count should be less than the 1st approach.

Each item decides a different way to find the invalid files, and even the way to compact mob files.

1.1 Save deleted cells in the del files

Scanning all of the del files to find the mapping between file name and deleted cells. So we could know how many cells are deleted in one particular mob file, and we could find the invalid ones.

Scanning the existing files found by fs.listStatus in one partition with the same start key and date and find the small files.

Combine the smalls, invalid files belong to the same partition, and all the scanned del files, we merge them. And move to the next partition after that.

When all the compactions are done, we try to delete these del files. Some del files might still reference mob files which are not invalid yet, we leave these files and they will be used in the next mob compaction.

Pros:

1. We could know where the deleted cells come from, so we could know which mob files are invalid, and after the mob compaction we could know which del files could be archived.

Cons:

1. We have to modify the logic of minor/major compaction to output the deleted cells into del files.

1.2 Save delete markers in the del file

Only delete markers are saved in the del files. We don't need to pre-scan the del files to find the invalid files since we only have delete markers in the del files and don't know where the deleted cells come from. We should use all the del files (all the delete markers) in the scan of the compaction. Considering the deletion is very rare in the user cases, we should only have a small number of delete markers.

In this approach, we only scan all the files and find the small files. And these small files are the candidates in the mob compaction, we open scanners to these candidates and all the del files, merge them into bigger one. The del files could not be deleted until all the mob files are selected. The "major mob compaction" is supposed to hardly happen, it means all these del files will be kept forever. But this option is simple and easy enough.

Pros:

1. Simple and easy enough to find the candidates.
2. Only output the delete markers to the del file, the size is less than the del files that have deleted cells, and less modification to the scanners.

Cons:

1. We don’t delete the del files until all the mob files are selected by the mob compaction which means it’s possible to keep all the del files forever.