This patch mainly change the getSplits() method, it used to return a list of splits, and every splits correspond to one region.

Now, I have made some changes, the current code workflow is showed below, notice that all the splits are based on uniform distribution

A list of splits, and every splits correspond to one region (the same in the original code)

N=regionSize/aveSize, then split the region into N pieces

Group those Neighbor regions into one split, so that their size similar to ave

Find average size for all the regions, and split region by average size

Continue splitting each region split into N pieces evenly

Return the list

(1) Notice that if user specify how many splits per region, this will disable autobalance

(2) And for user can specify how many splits per region feature, I wonder that if we can pass a list<regionName, mapperNumber> from user, and then the split will be better, but this will make the usage more complicated, user have to get regionName info and their size info after reading table info form somewhere

(3) And for the autoBalance feature, there is a tricky about the aveSize, sometimes if we have several big regions, say we have 5 regions there are [3G, 5G, 4G, 2G, 1G], the ave size is 3G, which might be still a little large, we can automatically set ave size to a small number say 512M, then the mappers for each region is [6,10,8,4,2]

(4) For the feature that user could specify total number of mapper for one MR job, this one is a little complicated, and I do not implement it yet.

for example, we have 100 regions, and we only set 40 mappers for this job

1. if we do not consider region size, then every 3 regions share 1 mapper, and there are more mappers left. (this one might be not hard to implement)

2. If we consider region size, it is really hard to assign mappers to regions based on their region size and mapper number (hard to implement)