

A Solution for Data Skew in HBase- MapReduce Jobs

yeweichen@qiyi.com

Motivation

- In production environment, data skew is a very common case. A HBase table may contains a lot of small regions and several large regions.
- Small regions waste a lot of computing resources. If we use a job to scan a table with 3000 small regions, we need a job with 3000 mappers.
- Large regions always block the job. If in a 100-region table, one region is far large then the other 99 regions. When we run a job with the table as input, 99 mappers will be completed very quickly, and then we need to wait for the last mapper for a long time.

Configuration

- Add two new configuration

hbase.mapreduce.input.autobalance = true means enabling the “auto balance” in HBase-MapReduce jobs. The default value is false.

hbase.mapreduce.input.ratio = 3 (default is 3). If a region size is larger than 3x average region size, treat the region as “proportionately too large”

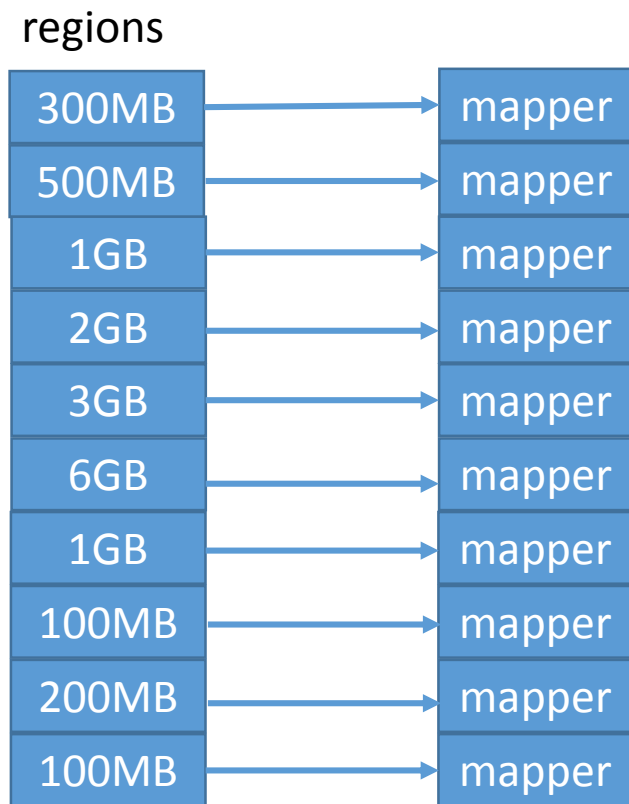
If (**region size** \geq average size * ratio) : cut the region into two MR input splits

If (average size \leq **region size** $<$ average size * ratio) : one region as one MR input split

If (**sum of several continuous regions size** $<$ average size): combine these regions into one MR input split.

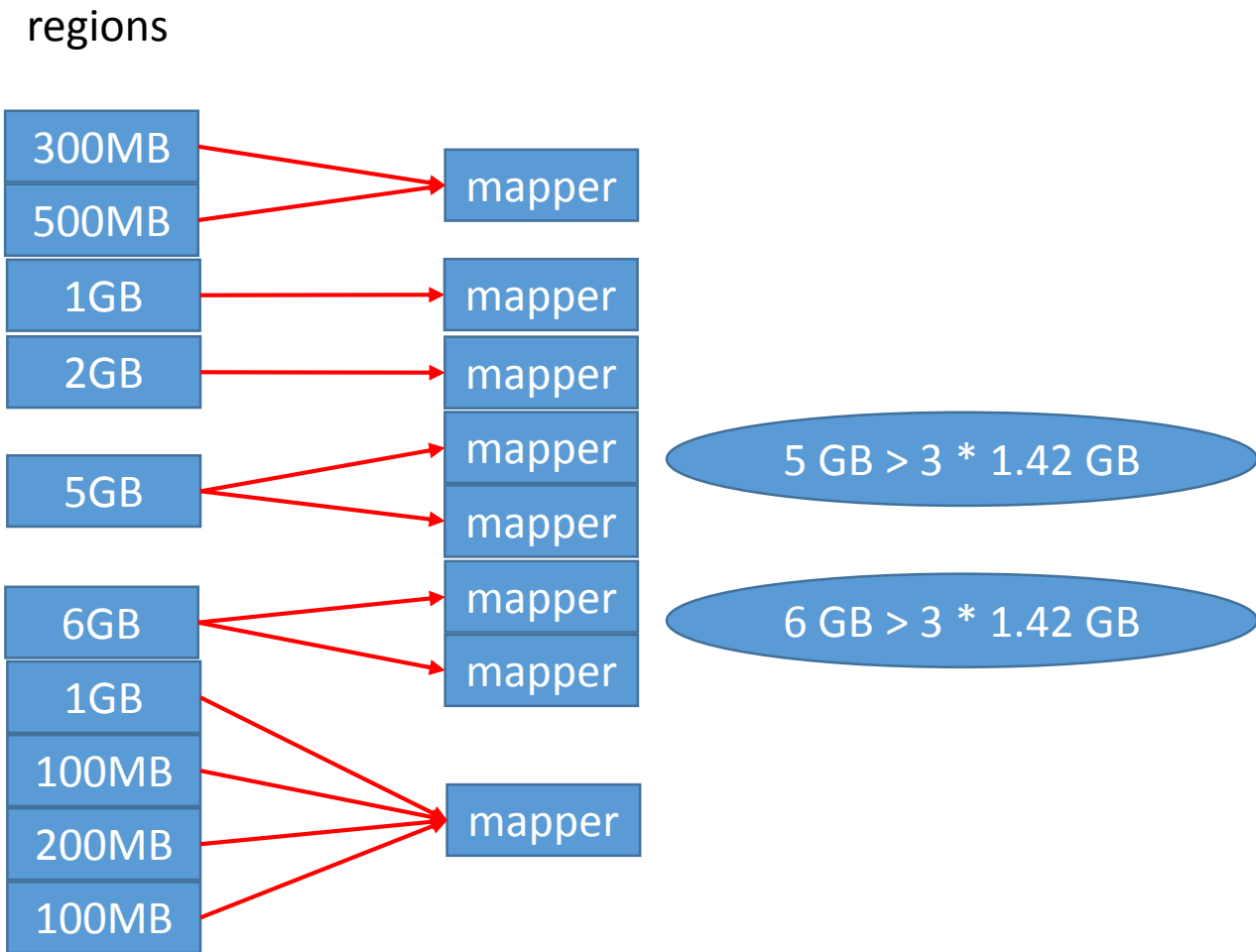
Example

- For example, if we have a table with 10 regions, the average region size is 1.42 GB.



Traditional HBase-Mapreduce Job

Average Size=1.42GB



HBase-Mapreduce Job with “Auto Balance”

```
* hbase.mapreduce.input.ratio = 3
```

Notice

- [HBase-2302](#) is a feature to exclude some specific regions from the MR job.
- When we set ***hbase.mapreduce.input.autobalance = true***, the feature in HBASE-2302 would be ineffective. Because in a TableSplit object there are only “startRow ” and “endRow”. So all regions between the start row and end row are included in the TableSplit. The mapper deal with all continuous regions between the start and end rows, no one can be excluded.