CHAPTER 2. MAPREDUCE AND THE NEW SOFTWARE STACK

Raymond

November 21, 2022

1 Exercise 2.2.1

1.1 a

Yes, I expect there to be more skew in the times taken by the various reducers to process their value list. Assume that there are m documents to be considered. The reason is that, for the word that appears more time, such as the stop word, if we don't use the combiner in the Map tasks, the reduce process should calculate significantly more times of sum, however if we perform combiner, the total summing time will be no more than m times. In other words, the combiner in the Map tasks help to reduce the pressure of the reduce tasks, so that among the reduce task, the workload should be smaller than without the combiner.

1.2 b

We can expect more skew if we combine 10 reduce tasks than 10,000 reduce tasks. Assume the time of the reduce task follows a normal distribution with $N\ (\mu, \sigma^2)$, and there are in total m calculation. If we divide them into 10 tasks, for each task the total time should follow $N\ (\frac{m}{10} * \mu, \frac{m}{10} * \sigma^2)$. If we divide them into 10,000 tasks, should follow $N\ (\frac{m}{10{,}000} * \mu, \frac{m}{10{,}000} * \sigma^2)$, where we can see that the latter variance is lower than the former one.

2 Exercise 2.3.1

2.1 a) The Largest Integer

Map function:

```
    Map(file): maxLocal = MIN-INTEGER
    for i in file:
    if i >= maxLocal:
    maxLocal = i
    return ['max',maxLocal]
    Reduce function([key,value]):
    maxGlobal = MIN-INTEGER
    for i in value:
    if i >= maxGlobal:
    maxGlobal = i
    return maxGlobal
```

3 Exercise 2.4.1

$\mathbb{E}$(#Times Failure per Task) = tp, since it follows a Binomial Distribution. So, $\mathbb{E}$(Total Time)= $nt + tp * 10t * n = nt(1 + 10pt)$

4 Exercise 2.4.2

Suppose executing a superstep costs t, there are m total supersteps, and we take a checkpoint every n steps. $\mathbb{E}$(Total Time)$= p * m * \frac{(n-1)}{2} * t + t * m + \frac{m}{n} * c * t = tm(p * \frac{n-1}{2} + 1 + \frac{1}{n} * c)$. Take derivative of n, we have $\frac{p}{2} - \frac{c}{n^2}$. If $\frac{p}{2} < \frac{c}{n^2}$, $\mathbb{E}$(Total Time) is monotone decreasing, and we can have the min value when $n = (\frac{2c}{p})^{\frac{1}{2}}$