Technical Report for Fourth Project in Parallel Computing

Yasheng Sun
117020910076
sunyasheng123@gmail.com

Abstract

This report concludes the fourth project in parallel computing. This projcet requires us to compute the highest and lowest temperature in some specified cities, which involves a classical mapreduce pattern – Summarization Pattern. We generate 10000000 items in 5 cities and obtain the statistics of highest and lowest temperature through MapReduce technique. All my code is publicly available on my github https://github.com/sunyasheng/Parallel-Computing.

1. Data Preparation

Downloading the history temperature of a certain city is a non-trivial thing, requring usage of API to some website. Unfortunately, some project has been out of date such as weatherData project[1]. Therefore, we generate the data randomly to simulate this scenario without loss of generality. The statistics of our generated data is shown in Fig. 1. We generate our data according to Gaussian distribution. The mean is set to 0 and the variance is set to 100. Temperature in 5 cities is generated and each city contains 10000000 weather items.

2. Methodology

In this section, the whole workflow of MapReduce is illustrated in Fig. 2. The task of Mapper is to select every item which contains a city name and its corresponding temperature. Then the output stream will be shuffled, sorted and passed to Reducer. Reducer will go through all the items and maintain the highest and lowest temperature for each city.

It is worthy to mention that the number of weather data transportation from mapper to reducer is huge. The internode transportation is usually expensive. To lower the amount of transportation, Combiner is introduced here to collect the highest and lowest temperature in one single data node. The Combiner is essentially a lcoal reducer. With this pre-reduction, the transportation amount between different nodes can be decreased significantly.

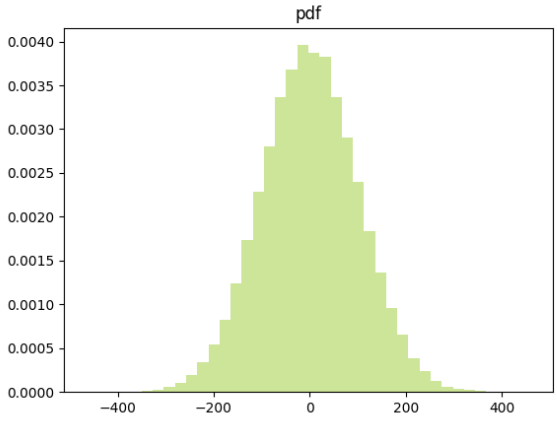

Figure 1. Statistics of the generated data. The mean is set to 0 and the variance is set to 100.

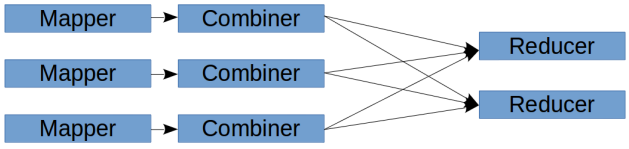

Figure 2. The MapReduce workflow.

3. Result

The output of this mapreduce task is shown in Fig. 3. The highest and lowest temperature for each city is given here.

From Fig. 4 we could know that the mapper task is split to five nodes and the output stream will be passed to one reduce node to finally summarize the result after map operation.

Fig. 5 shows that the Combiner technique significantly reduces the amount of reduce input records and reduce shuffle bytes compared with the traditional routine without Combiner as is shwon in Fig. 6. This improvement will

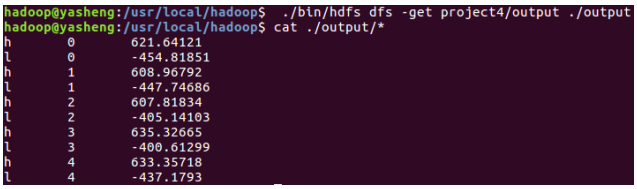

Figure 3. System output for the highest and lowest temperature in those 5 cities.

Task Type	Total
Map	5
Reduce	1

Figure 4. System information in MapReduce Task.

Reduce input groups	0	5
Reduce input records	0	18
Reduce output records	0	10
Reduce shuffle bytes	0	290

Figure 5. The reduce input records and shuffle bytes information with Combiner.

Reduce input groups	0	5
Reduce input records	0	50,000,000
Reduce output records	0	10
Reduce shuffle bytes	0	731,655,457

Figure 6. The reduce input records and shuffle bytes information without Combiner.

play a more important role when it comes to a larger scale dataset.

4. Conclusion

We show the classical MapReduce paradigme – Summarization Pattern by a minimum/maximax search problem in this project. A practical technique, Combiner, is introduced here to lower the transportation amount between different node. Future work will involve realtime data grabbing and reducing task under distributed computation framework.

References

[1] http://ram-n.github.io/weatherData