Distributed System Practical Work 3

Map Reduce Words Count

April 1, 2020

Contents

1 Protocol Design

This Map Reduce Words Count Practical Work is written in C++, we decide to make up our own Map Reduce Implementation from C++ because idk...

2 Codes

2.1 Map

How this section works is that, we detect each word by a " " (a space bar) and it's stored in a pair of string and int.
This is the code snippet from the mapping section of the practical work:

```
11  void map(const char *ptr) {
12      //Mapping to seperate
13      int num = 0;
14      std::string word="";
15      std::pair<std::string, int> pair;
16      while (*ptr) {
17          if (*ptr == ' ') {
18              pair.first = word;
19              pair.second = 1;
20              vector.push_back(pair);
21              word = "";
22          }
23          else{
24              word.append(1,*ptr);
25          }
26          ptr++;
27      }
28  }
```

2.2 Reduce

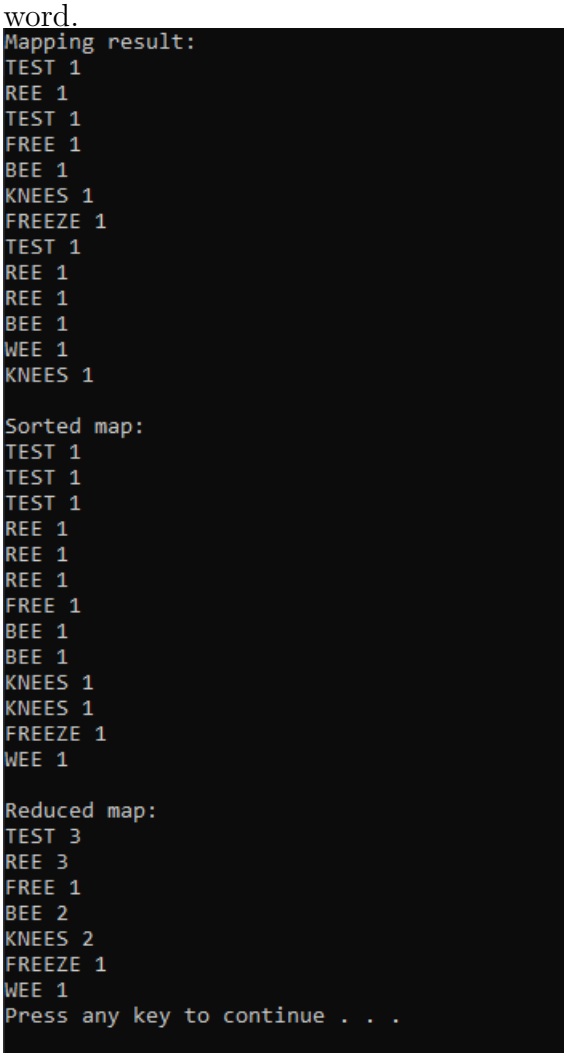
This reduce section is mainly working with vector in C++, we have 2 vectors, 1 is the first vector and the other is the result. We first look for similar words in the first vector and push it in the result vector (look for all the same words in the sentence to push those into the result bar, this is to sort the Map).
Then we check for the same words in the result vector, increment the integer value and deletes the other similar words.
This is the code snippet from the reduce section of the practical work:

```
void reduce(std::vector<std::pair<std::string, int>>) {
    std::vector<std::pair<std::string, int>>::iterator i;
    std::vector<std::pair<std::string, int>>::iterator j;
    //Sorting
    for(i = vector.begin(); i != vector.end(); i++) {
        result.push_back(std::make_pair(i->first,i->second));
        for (j = i + 1; j != vector.end(); j++) {
            if (i->first == j->first && j->second==1) {
                result.push_back(std::make_pair(j->first, j->second));
                j->first = "";
                j->second = 0;
            }
        }
        i->first = "";
        i->second = 0;
    }
    i = result.begin();
    while (i != result.end()) {
        if (i->second == 0) {
            i = result.erase(i);
        }
        else {
            ++i;
        }
    }
    //Printing out the Sorted Map
    std::cout << "Sorted map: \n";
    for (i = result.begin(); i != result.end(); i++) {
        std::string p1 = i->first;
        int p2 = i->second;
        std::cout << p1 << " " << p2 << "\n";
    }

    //Reduce
    for (i = result.begin(); i != result.end(); i++) {
        for (j = i + 1; j != result.end(); j++) {
            if (i->first == j->first) {
                i->second++;
                result.erase(j--);
            }
        }
    }
}
```

2.3 Result

The sentence can be changed in the codes, we haven't tested if it works with Uppercase and Lowercase yet, but 1 think to note is that at the end of the sentence, it needs a " " space bar or the program won't recognize the final word.



In conclusion:

The program is running smoothly and it's working as intended. The program is to our satisfaction!!