大数据处理综合实验：实验 2 —— 实验报告

小组 10

基础功能：倒排索引

一、设计思路

（1）Mapper 类（InvertedIndexMapper）：

- Key 类型：输入的键是 `Object`，这通常是 Hadoop 的默认值，但实际上它不被用到。文本内容会被忽略，因为每一行都被解析为 `Text`。
- Value 类型：输入的值是 `Text` 类型，代表了文档的内容。

思路：

- 每个 Mapper 实例负责处理一个文档。
- 在 `map` 方法中，首先获取当前处理的文档名（文件名），然后将文本内容拆分为单词。
- 对于每个单词，将其设置为输出的 Key，文档名设置为输出的 Value，然后写出。

（2）Reducer 类（InvertedIndexReducer）：

- Key 类型：输入的键是 `Text`，代表了单词。
- Value 类型：输入的值是 `Text`，代表了包含该单词的文档名。

思路：

- 每个 Reducer 实例负责处理一个唯一的单词。
- 在 `reduce` 方法中，对于每个单词，遍历其对应的文档名列表，计算每个文档名的出现次数并累加总出现次数。
- 然后将计算出的每个文档名的出现次数和总出现次数写入输出。
- 输出的 Key 仍然是单词，但 Value 是一个包含了各文档名及其出现次数的字符串。

总体来说，Map 阶段负责将文档拆分为单词并将其发送到对应的 Reducer，而 Reduce 阶段则负责计算每个单词在各文档中的出现次数，并输出倒排索引结果。

（3）伪代码

MapReduce 中 Map 和 Reduce 的伪代码。

```
function map(key, value, context):
    fileName = getFileName(context.getInputSplit())
    tokens = splitText(value)
    for token in tokens:
        emitIntermediate(token, fileName, context)

function getFileName(inputSplit):
    return inputSplit.getPath().getName()

function splitText(text):
    return text.split("[\\s,]+")

function emitIntermediate(key, value, context):
    context.write(key, value)

```

```
function reduce(key, values, context):
    countMap = initializeEmptyHashMap()
    totalCount = 0
    for value in values:
        fileName = value
        incrementCount(countMap, fileName)
        totalCount = totalCount + 1

    averageFrequency = calculateAverageFrequency(totalCount, countMap.size())
    output = generateOutputString(averageFrequency, countMap)
    context.write(key, output)

function initializeEmptyHashMap():
    return new HashMap()

function incrementCount(countMap, fileName):
    if countMap.containsKey(fileName):
        countMap.put(fileName, countMap.get(fileName) + 1)
    else:
        countMap.put(fileName, 1)

function calculateAverageFrequency(totalCount, numberOfFiles):
    return totalCount / numberOfFiles

function generateOutputString(averageFrequency, countMap):
    output = formatAverageFrequency(averageFrequency)
    output = output + ","
    for entry in countMap.entrySet():
        output = output + entry.getKey() + ":" + entry.getValue() + ";"
    return output

function formatAverageFrequency(averageFrequency):
    return String.format("%.2f", averageFrequency)

```

二、实验结果

输出文件路径：/outputSS/part-r-00000

（1）输出结果文件的部分截图

```
''All   1.00,shakespeare-lovers-62.txt:1;
''Among 1.00,shakespeare-lovers-62.txt:1;
''And   1.00,shakespeare-lovers-62.txt:1;
''But   1.00,shakespeare-lovers-62.txt:1;
''Gamut'        1.00,shakespeare-taming-2.txt:1;
''How   1.00,shakespeare-lovers-62.txt:1;
''Lo    2.00,shakespeare-lovers-62.txt:2;
''Look  1.00,shakespeare-lovers-62.txt:1;
''My    1.00,shakespeare-lovers-62.txt:1;
''Now   1.00,shakespeare-lovers-62.txt:1;
''O     2.00,shakespeare-lovers-62.txt:2;
''Od's  1.00,shakespeare-merry-15.txt:1;
''The   1.00,shakespeare-lovers-62.txt:1;
''Tis   2.00,shakespeare-comedy-7.txt:1;shakespeare-venus-60.txt:5;shakespeare-sonnets-59.txt:1;shakespeare-as-12.txt:1;
''When  1.00,shakespeare-lovers-62.txt:1;
''tis   1.00,shakespeare-alls-11.txt:1;
''twas  1.00,shakespeare-two-18.txt:1;
'--     3.29,shakespeare-two-18.txt:2;shakespeare-twelfth-20.txt:9;shakespeare-hamlet-25.txt:3;shakespeare-troilus-22.txt:1;shakespeare-
loves-8.txt:5;shakespeare-alls-11.txt:2;shakespeare-life-54.txt:1;
'--My   1.00,shakespeare-king-45.txt:1;
'--O    1.00,shakespeare-troilus-22.txt:1;
'--every        1.00,shakespeare-second-52.txt:1;
'--or   3.00,shakespeare-midsummer-16.txt:3;
'--the  1.00,shakespeare-hamlet-25.txt:1;
'--this 1.00,shakespeare-coriolanus-24.txt:1;
'--thus 1.00,shakespeare-venus-60.txt:1;
'?      1.00,shakespeare-coriolanus-24.txt:1;
'A      1.63,shakespeare-antony-23.txt:1;shakespeare-venus-60.txt:1;shakespeare-hamlet-25.txt:1;shakespeare-taming-2.txt:2;shakespeare-
third-53.txt:2;shakespeare-othello-47.txt:3;shakespeare-midsummer-16.txt:1;shakespeare-life-54.txt:2;
'ARTEMIDORUS.'  1.00,shakespeare-julius-26.txt:1;
'Above  1.00,shakespeare-twelfth-20.txt:1;
'Achilles       2.00,shakespeare-troilus-22.txt:2;
```

```
heart's-ease    1.00,shakespeare-life-54.txt:1;
heart--play     1.00,shakespeare-measure-13.txt:1;
heart--that     1.00,shakespeare-cymbeline-17.txt:1;
heart-ache      1.00,shakespeare-hamlet-25.txt:1;
heart-blood     1.67,shakespeare-tragedy-57.txt:3;shakespeare-third-53.txt:1;shakespeare-troilus-22.txt:1;
heart-break.    1.00,shakespeare-merry-15.txt:1;
heart-burned    1.00,shakespeare-much-3.txt:1;
heart-burned.   1.00,shakespeare-first-51.txt:1;
heart-burning   1.00,shakespeare-loves-8.txt:1;
heart-easing    1.00,shakespeare-rape-61.txt:1;
heart-grief     1.00,shakespeare-life-54.txt:1;
heart-heaviness 1.00,shakespeare-as-12.txt:1;
heart-inflaming 1.00,shakespeare-sonnets-59.txt:1;
heart-poor      1.00,shakespeare-rape-61.txt:1;
heart-sick.     1.00,shakespeare-cymbeline-17.txt:1;
heart-sore      2.00,shakespeare-two-18.txt:2;
heart-sorrow    1.00,shakespeare-tempest-4.txt:1;
heart-sorrowing 1.00,shakespeare-tragedy-58.txt:1;
heart-string    1.00,shakespeare-life-54.txt:1;
heart-strings   1.00,shakespeare-tragedy-58.txt:1;shakespeare-rape-61.txt:1;
heart-strings.  1.00,shakespeare-two-18.txt:1;
heart-struck    1.00,shakespeare-king-45.txt:1;
heart-whole.    1.00,shakespeare-as-12.txt:1;
```

（2）Yarn Resource Manager 的 WebUI 执行报告内容