

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。

```

1  function map(key, value, context):
2      fileName = getFileName(context.getInputSplit())
3      tokens = splitText(value)
4      for token in tokens:
5          emitIntermediate(token, fileName, context)
6
7  function getFileName(inputSplit):
8      return inputSplit.getPath().getName()
9
10 function splitText(text):
11     return text.split("[\\s,]+")
12
13 function emitIntermediate(key, value, context):
14     context.write(key, value)
15

```

```

1  function reduce(key, values, context):
2      countMap = initializeEmptyHashMap()
3      totalCount = 0
4      for value in values:
5          fileName = value
6          incrementCount(countMap, fileName)
7          totalCount = totalCount + 1
8
9      averageFrequency = calculateAverageFrequency(totalCount, countMap.size())
10     output = generateOutputString(averageFrequency, countMap)
11     context.write(key, output)
12
13 function initializeEmptyHashMap():
14     return new HashMap()
15
16 function incrementCount(countMap, fileName):
17     if countMap.containsKey(fileName):
18         countMap.put(fileName, countMap.get(fileName) + 1)
19     else:
20         countMap.put(fileName, 1)
21
22 function calculateAverageFrequency(totalCount, numberOfFiles):
23     return totalCount / numberOfFiles
24
25 function generateOutputString(averageFrequency, countMap):
26     output = formatAverageFrequency(averageFrequency)
27     output = output + ","
28     for entry in countMap.entrySet():
29         output = output + entry.getKey() + ":" + entry.getValue() + ";"
30     return output
31
32 function formatAverageFrequency(averageFrequency):
33     return String.format("%.2f", averageFrequency)
34

```

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



Application application_1678703754602_9932

Logged in as: drwho

Cluster

About
Nodes
Node Labels
Applications

NEW
NEW SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED

Scheduler

Tools

User: 2024bbl_10

Name: Inverted Index

Application Type: MAPREDUCE

Application Tags:

Application Priority: 0 (Higher integer value indicates higher priority)

YarnApplicationState: FINISHED

Queue: 2024bpdad3ss2

FinalStatus Reported by AM: SUCCEEDED

Started: Mon May 13 14:25:44 +0800 2024

Launched: Mon May 13 14:25:44 +0800 2024

Finished: Mon May 13 14:29:21 +0800 2024

Elapsed: 3mins, 37sec

Tracking URL: History

Log Aggregation Status: TIME_OUT

Application Timeout (Remaining Time): Unlimited

Diagnosics:

Unmanaged Application: false

Application Node Label expression: <Not set>

AM container Node Label expression: <DEFAULT_PARTITION>

Total Resource Preempted: <memory 0, vCores 0>

Total Number of Non-AM Containers Preempted: 0

Total Number of AM Containers Preempted: 0

Resource Preempted from Current Attempt: <memory 0, vCores 0>

Number of Non-AM Containers Preempted from Current Attempt: 0

Aggregate Resource Allocation: 1102700 MB-seconds, 630 vcore-seconds, 0 yarn.io/gpu-seconds

Aggregate Preempted Resource Allocation: 0 MB-seconds, 0 vcore-seconds

Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
apopathmot_1678703754602_9932_000001	Mon May 13 14:25:44 +0800 2024	http://slave003.8042	Logs	0	0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last