

Интеллектуальные информационные системы

Решение задач MapReduce

Кафедра управления и интеллектуальных технологий НИУ «МЭИ»
2023 г.

Word Count

```
Method Map(docid id, doc d):  
    for each term t in doc d:  
        Emit (term t, count 1)
```

```
Method Reduce(term t, counts [c1,c2,c3]):  
    int sum = 0;  
    for each term c in counts [c1,c2,c3]:  
        sum += c  
    Emit (term t, count sum)
```

Много данных от mapper-ов, можем после каждого mapper-а считать промежуточную сумму на combiner-е. Тогда на reducer-ы будет передаваться меньше данных.

Word Count

```
Method Map(docid id, doc d):  
    for each term t in doc d:  
        Emit (term t, count 1)
```

```
Method Combine(term t, counts [c1,c2,c3]):  
    int sum = 0;  
    for each term c in counts [c1,c2,c3]:  
        sum += c  
    Emit (term t, count sum)
```

```
Method Reduce(term t, counts [c1,c2,c3]):  
    int sum = 0;  
    for each term c in counts [c1,c2,c3]:  
        sum += c  
    Emit (term t, count sum)
```

Тот же код, что и на reducer.

Combiner на той же машине, что и mapper – нет большой передачи данных по сети.

Но по-прежнему много операций чтения и записи, пусть и на одной и той же машине, т.к. результат mapper записывает в файл, а combiner считывает из файла.

Word Count. In-mapper combiner

```
Method Map(docid id, doc d):  
  H: new AssociativeArray  
  for each term t in doc d:  
    H{t} = H{t} + 1  
  for each term t in H:  
    Emit (term t, count H{t})
```

Теперь данные храним в ассоциативном массиве **H** и передаем в reducer после каждого документа. Учтеть, чтобы памяти хватило под размер документа. Объем данных по сети падает, при этом нет записи в файл для combiner-а как в прошлом примере.

Word Count. In-mapper combiner - 2

```
Method Init() :
```

```
    H: new AssociativeArray
```

```
Method Map(docid id, doc d) :
```

```
    for each term t in doc d:
```

```
        H{t} = H{t} + 1
```

```
Method Close() :
```

```
    for each term t in H:
```

```
        Emit (term t, count H{t})
```

Передаем данные в reducer по завершению работы mapper-а.

Объем данных по сети падает еще больше, но острее стоит вопрос памяти для хранения ассоциативного массива.

Контроль памяти, и передача данных в reducer в случае достижения заданного размера массива.

Среднее значение. Mapper

yandex.ru 10
vk.com 120
mail.ru 45
yandex.ru 25
yandex.ru 30
vk.com 350
vk.com 100
mail.ru 35

Время посещения каждого сайта. Найти, сколько в среднем пользователи проводили на каждом сайте.

```
Method Map(string site, integer t):  
    Emit (string site, integer t)
```

Среднее значение. Reducer

```
Method Reduce (string site, integers [t1, t2, t3]) :  
    sum = 0;  
    cnt = 0;  
    for each integer t in integers [t1, t2, t3] :  
        sum += t  
        cnt += 1  
    float avg = sum / cnt  
    Emit (string site, float avg)
```

Reducer получает много данных, можем оптимизировать с помощью combiner

Среднее значение. Reducer

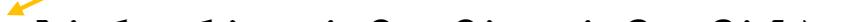
```
Method Map(string site, integer t):  
    Emit (string site, integer t)
```

Reducer может не выполняться,
Поэтому нужно предусмотреть это в
mapper и reducer!



```
Method Combine(string site, integer t):  
    sum = cnt = 0;  
    for each integer t in integers [t1,t2,t3] :  
        sum += t1  
        cnt += 1  
    Emit (string site, pair (sum, cnt))
```

```
Method Reduce(string site, pairs [(s1,c1), (s2,c2), (s3,c3)]):  
    sum = cnt = 0;  
    for each pair p in pairs [(s1,c1), (s2,c2), (s3,c3)]:  
        sum += p.s  
        cnt += p.c  
    float avg = sum / cnt  
    Emit (string site, float avg)
```



Среднее значение. Reducer

```
Method Map(string site, integer t):  
    Emit (string site, pair (t, 1))
```

Reducer может не выполняться,
Поэтому нужно предусмотреть это в
mapper и reducer!

```
Method Combine(string site, pair (t, 1)):  
    sum = cnt = 0;  
    for each integer t in integers [t1,t2,t3] :  
        sum += p.s  
        cnt += p.c  
    Emit (string site, pair (sum, cnt))
```

```
Method Reduce(string site, pairs [(s1,c1), (s2,c2), (s3,c3)]):  
    sum = cnt = 0;  
    for each pair p in pairs [(s1,c1), (s2,c2), (s3,c3)]:  
        sum += p.s  
        cnt += p.c  
    float avg = sum / cnt  
    Emit (string site, float avg)
```