

07 | 为什么说MapReduce既是编程模型又是计算框架？

2018-11-13 李智慧

从0开始学大数据

[进入课程 >](#)



讲述：李智慧

时长 09:52 大小 4.53M



在 Hadoop 问世之前，其实已经有了分布式计算，只是那个时候的分布式计算都是专用的系统，只能专门处理某一类计算，比如进行大规模数据的排序。很显然，这样的系统无法复用到其他的大数据计算场景，每一种应用都需要开发与维护专门的系统。而 Hadoop MapReduce 的出现，使得大数据计算通用编程成为可能。我们只要遵循 MapReduce 编程模型编写业务处理逻辑代码，就可以运行在 Hadoop 分布式集群上，无需关心分布式计算是如何完成的。也就是说，我们只需要关心业务逻辑，不用关心系统调用与运行环境，这和我们目前的主流开发方式是一致的。

请你先回忆一下，在前面[专栏第 4 期](#)我们讨论过，大数据计算的核心思路是移动计算比移动数据更划算。既然计算方法跟传统计算方法不一样，移动计算而不是移动数据，那么用传统的编程模型进行大数据计算就会遇到很多困难，因此 Hadoop 大数据计算使用了一种叫作 MapReduce 的编程模型。

其实 MapReduce 编程模型并不是 Hadoop 原创，甚至也不是 Google 原创，但是 Google 和 Hadoop 创造性地将 MapReduce 编程模型用到大数据计算上，立刻产生了神奇的效果，看似复杂的各种各样的机器学习、数据挖掘、SQL 处理等大数据计算变得简单清晰起来。

今天我们就来聊聊 **Hadoop 解决大规模数据分布式计算的方案——MapReduce**。

在我看来，**MapReduce 既是一个编程模型，又是一个计算框架**。也就是说，开发人员必须基于 MapReduce 编程模型进行编程开发，然后将程序通过 MapReduce 计算框架分发到 Hadoop 集群中运行。我们先看一下作为编程模型的 MapReduce。

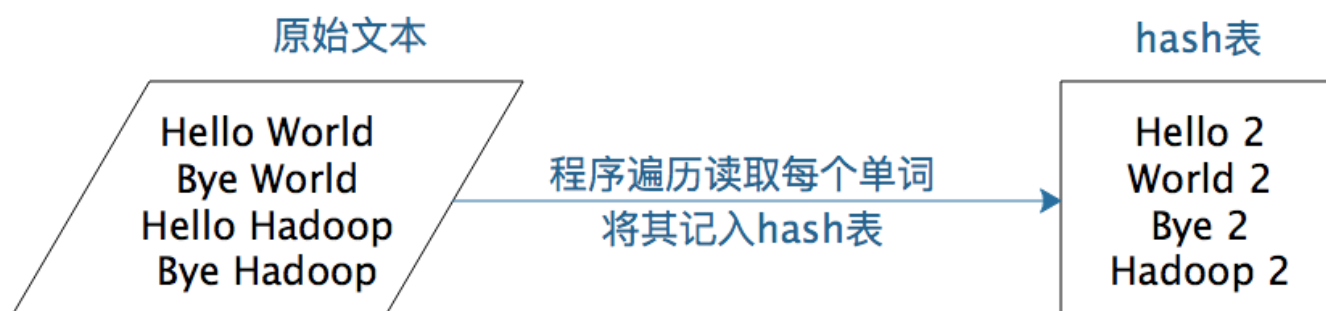
为什么说 MapReduce 是一种非常简单又非常强大的编程模型？

简单在于其编程模型只包含 Map 和 Reduce 两个过程，map 的主要输入是一对 <Key, Value> 值，经过 map 计算后输出一对 <Key, Value> 值；然后将相同 Key 合并，形成 <Key, Value 集合>；再将这个 <Key, Value 集合> 输入 reduce，经过计算输出零个或多个 <Key, Value> 对。


同时，MapReduce 又是非常强大的，不管是关系代数运算（SQL 计算），还是矩阵运算（图计算），大数据领域几乎所有的计算需求都可以通过 MapReduce 编程来实现。

下面，我以 WordCount 程序为例，一起来看下 MapReduce 的计算过程。

WordCount 主要解决的是文本处理中词频统计的问题，就是统计文本中每一个单词出现的次数。如果只是统计一篇文章的词频，几十 KB 到几 MB 的数据，只需要写一个程序，将数据读入内存，建一个 Hash 表记录每个词出现的次数就可以了。这个统计过程你可以看下面这张图。



如果用 Python 语言，单机处理 WordCount 的代码是这样的。


 复制代码

```
1 # 文本前期处理
2 str1_list = str.replace('\n', '').lower().split(' ')
3 count_dict = {}
4 # 如果字典里有该单词则加 1，否则添加入字典
5 for str in str1_list:
6     if str in count_dict.keys():
7         count_dict[str] = count_dict[str] + 1
8     else:
9         count_dict[str] = 1
```

简单说来，就是建一个 Hash 表，然后将字符串里的每个词放到这个 Hash 表里。如果这个词第一次放到 Hash 表，就新建一个 Key、Value 对，Key 是这个词，Value 是 1。如果 Hash 表里已经有这个词了，那么就给这个词的 Value + 1。

小数据量用单机统计词频很简单，但是如果统计全世界互联网所有网页（数万亿计）的词频数（而这正是 Google 这样的搜索引擎的典型需求），不可能写一个程序把全世界的网页都读入内存，这时候就需要用 MapReduce 编程来解决。

WordCount 的 MapReduce 程序如下。

 复制代码

```
1 public class WordCount {
2
3     public static class TokenizerMapper
4         extends Mapper<Object, Text, Text, IntWritable>{
5
6         private final static IntWritable one = new IntWritable(1);
7         private Text word = new Text();
8
9         public void map(Object key, Text value, Context context
10             ) throws IOException, InterruptedException {
11             StringTokenizer itr = new StringTokenizer(value.toString());
12             while (itr.hasMoreTokens()) {
13                 word.set(itr.nextToken());
14                 context.write(word, one);
15             }
16         }
17     }
18 }
```

```

19 public static class IntSumReducer
20     extends Reducer<Text,IntWritable,Text,IntWritable> {
21     private IntWritable result = new IntWritable();
22
23     public void reduce(Text key, Iterable<IntWritable> values,
24                         Context context
25                         ) throws IOException, InterruptedException {
26         int sum = 0;
27         for (IntWritable val : values) {
28             sum += val.get();
29         }
30         result.set(sum);
31         context.write(key, result);
32     }
33 }
34 }

```

你可以从这段代码中看到，MapReduce 版本 WordCount 程序的核心是一个 map 函数和一个 reduce 函数。

map 函数的输入主要是一个 <Key, Value> 对，在这个例子里，Value 是要统计的所有文本中的一行数据，Key 在一般计算中都不会用到。

 复制代码


```

1 public void map(Object key, Text value, Context context
2                 )

```

map 函数的计算过程是，将这行文本中的单词提取出来，针对每个单词输出一个 <word, 1> 这样的 <Key, Value> 对。

MapReduce 计算框架会将这些 <word, 1> 收集起来，将相同的 word 放在一起，形成 <word, <1,1,1,1,1,1,1...>> 这样的 <Key, Value 集合> 数据，然后将其输入给 reduce 函数。

 复制代码

```

1 public void reduce(Text key, Iterable<IntWritable> values,
2                    Context context
3                    )

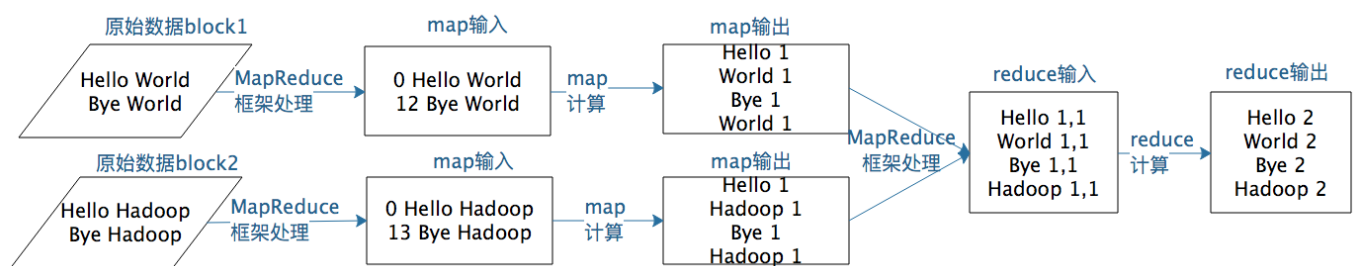
```

这里 reduce 的输入参数 Values 就是由很多个 1 组成的集合，而 Key 就是具体的单词 word。

reduce 函数的计算过程是，将这个集合里的 1 求和，再将单词（word）和这个和（sum）组成一个 <Key, Value>，也就是 <word, sum> 输出。每一个输出就是一个单词和它的词频统计总和。

一个 map 函数可以针对一部分数据进行运算，这样就可以将一个大数据切分成很多块（这也正是 HDFS 所做的），MapReduce 计算框架为每个数据块分配一个 map 函数去计算，从而实现大数据的分布式计算。

假设有两个数据块的文本数据需要进行词频统计，MapReduce 计算过程如下图所示。



以上就是 MapReduce 编程模型的主要计算过程和原理，但是这样一个 MapReduce 程序要想在分布式环境中执行，并处理海量的大规模数据，还需要一个计算框架，能够调度执行这个 MapReduce 程序，使它在分布式的集群中并行运行，而这个计算框架也叫 MapReduce。

所以，当我们说 MapReduce 的时候，可能指编程模型，也就是一个 MapReduce 程序；也可能是指计算框架，调度执行大数据的分布式计算。关于 MapReduce 计算框架，我们下期再详细聊。

小结

总结一下，今天我们学习了 MapReduce 编程模型。这个模型既简单又强大，简单是因为它只包含 Map 和 Reduce 两个过程，强大之处又在于它可以实现大数据领域几乎所有的计算需求。这也正是 MapReduce 这个模型令人着迷的地方。

说起模型，我想跟你聊聊我的体会。

模型是人们对一类事物的概括与抽象，可以帮助我们更好地理解事物的本质，更方便地解决问题。比如，数学公式是我们对物理与数学规律的抽象，地图和沙盘是我们对地理空间的抽象，软件架构图是软件工程师对软件系统的抽象。

通过抽象，我们更容易把握事物的内在规律，而不是被纷繁复杂的事物表象所迷惑，更进一步深刻地认识这个世界。通过抽象，伽利略发现力是改变物体运动的原因，而不是使物体运动的原因，为全人类打开了现代科学的大门。

这些年，我自己认识了很多优秀的人，他们各有所长、各有特点，但是无一例外都有个共同的特征，就是**对事物的洞察力**。他们能够穿透事物的层层迷雾，直指问题的核心和要害，不会犹豫和迷茫，轻松出手就搞定了其他人看起来无比艰难的事情。有时候光是看他们做事就能感受到一种美感，让人意醉神迷。

这种洞察力就是来源于他们对事物的抽象能力，虽然我不知道这种能力缘何而来，但是见识了这种能力以后，我也非常渴望拥有对事物的抽象能力。所以在遇到问题的时候，我就会停下来思考：这个问题为什么会出现，它揭示出来背后的规律是什么，我应该如何做。甚至有时候会把这些优秀的人带入进思考：如果是戴老师、如果是潘大侠，他会如何看待、如何解决这个问题。通过这种不断地训练，虽然和那些最优秀的人相比还是有巨大的差距，但是仍然能够感受到自己的进步，这些小小的进步也会让自己产生大大的快乐，一种不荒废光阴、没有虚度此生的感觉。


我希望你也能够不断训练自己，遇到问题的时候，停下来思考一下：这些现象背后的规律是什么。有时候并不需要多么艰深的思考，仅仅就是停一下，就会让你察觉到以前不曾注意到的一些情况，进而发现事物的深层规律。这就是洞察力。

思考题

对于这样一张数据表


pv_users	
pageid	age
1	25
2	25
1	32
2	25

如果存储在 HDFS 中，每一行记录在 HDFS 对应一行文本，文本格式是

 复制代码

```
1 1,25
2 2,25
3 1,32
4 2,25
```

根据上面 WordCount 的示例，请你写一个 MapReduce 程序，得到下面这条 SQL 的计算结果。

 复制代码

```
1 SELECT pageid, age, count(1) FROM pv_users GROUP BY pageid, age;
```

TIPS：如何用 MapReduce 实现 SQL 计算，我们在后面还会进一步讨论。

欢迎你写下自己的思考或疑问，与我和其他同学一起讨论。

从 0 开始学大数据

智能时代你的大数据第一课

李智慧

同程艺龙交通首席架构师
前 Intel 大数据架构师



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 06 | 新技术层出不穷，HDFS依然是存储的王者

下一篇 08 | MapReduce如何让数据完成一次旅行？

精选留言 (42)

写留言



落叶飞逝的...

2018-11-13

46

老师，我是个大数据的初学者，因为这个专栏是从零入门的，但是目前的我还不知道如何在自己机器上安装哪些软件？如何操作？因为这些问题没解决，所以没办法真切的体会到文中的处理单词统计大数据的魅力。所以希望老师能讲下必备软件的安装的内容，及操作环节。谢谢



intuition

2018-11-13

19

李老师的文章已经不仅仅局限于技术本身 更多的是对人生的思考 如何去成为一个思考者才是我们所追求的目标



无形

2018-11-13

👍 16

把pageID和age当做key计算出现的次数并做汇总，然后对key排序，输出排序后的key和其对应的总次数



西贝木土

2018-11-13

👍 12

package com.company.sparkcore

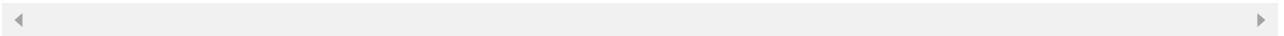
```
import org.apache.log4j.{Level, Logger}
```

```
import org.apache.spark.{SparkConf, SparkContext}
```

...

展开 ▾

作者回复: 👍



朱国伟

2018-11-17

👍 9

单机安装伪hadoop集群

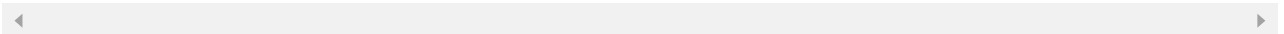
见: <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html>

注: 在Mac中安装遇到了一些问题 但是google一下就能解决 恕不一一道来

...

展开 ▾

作者回复: 👍



一箭中的

2018-11-13

👍 7

将pageid和 age拼接成字符串当做一个key，然后通过Map和Reduce计算即可得出对应的count



Ahikaka

👍 6

2018-11-13

老师能不能推荐些学大数据的书籍，博客，网站。

展开 ▾



喜笑延开

2018-11-23

👍 5

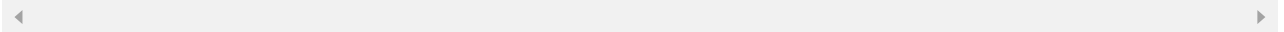
不能光想，必须动手实践：

Mapper

```
public class PageMapper extends Mapper<LongWritable,Text,Text,IntWritable> {  
    @Override  
    protected void map(LongWritable key, Text value, Context context) throws...
```

展开 ▾

作者回复: 👍



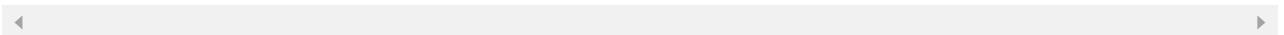
有铭

2018-11-15

👍 5

我想问一下那个计算过程的示意图里map输入部分，上面的是0，12，下面是0，13，是啥意思？

作者回复: map函数输入的key，表示这行数据在文件中的偏移量，通常忽略掉



呆猫

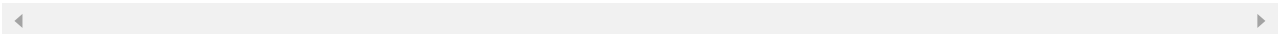
2018-11-15

👍 4

文章真的是看的赏心悦目，尤其是那段描述抽象的文字 😊

展开 ▾

作者回复: 谢谢



Lambda

2018-11-13

👍 4

好像中间拉下了 shuffle

展开 ▾



三木子

2018-11-13

👍 4

看到这个问题，我在想我在怎么想？

展开 ▾



小成

2019-01-01

👍 2

老师，我是大数据初学者，除了编程语言本身的，可以推荐一些书籍或者资料辅助这个专栏的学习吗，像hadoop相关类的，这期的代码看不懂了。



无处不在

2018-11-21

👍 2

这个如果在复杂或者高级一点，就需要用mapreduce的序列化对象作为key的功能去实现了，最近也在学习大数据，学的时候感觉找到了sql的本质，记得公司前年的项目就是手写了一堆js函数，实现了mongodb的类似sql的分组聚合操作。

后续可以开设视频的专栏就更好了

展开 ▾



糊糊

2018-11-15

👍 2

mapreduce核心思想就跟传统的SQL中的group by一样

展开 ▾



ward-wolf

2018-11-14

👍 2

我的思路和前面几个同学的类似，就是把文本直接当做key，value使用数字统计，最后就是通过reduce统计出现次数了

作者回复: 是的



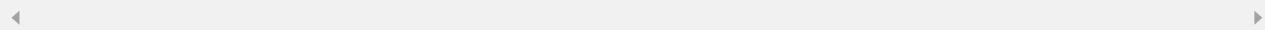
明天更美好

2018-11-13

👍 2

对于大数据来说是盲区，如果应用直接往hbase中写可以吗？2.5万的并发。hbase可以满足我们的查询需求吗？还有日志分析

作者回复: 你可能需要一个完整的技术架构方案，而不只是HBASE能不能搞定的问题，建议你看下我写另一本书《大型网站技术架构：核心原理与案例分析》，专栏后面也会对大数据架构有各个角度的探讨，欢迎持续关注



老男孩

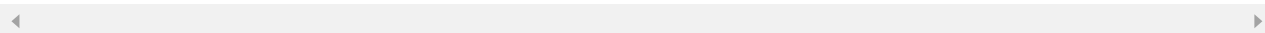
2018-11-13

👍 2

老师关于抽象是洞察事物本质的总结很精辟。关于思考题，我的思路是把pageid+age作为map函数计算key值，value分别是1。然后reduce再根据key对value的集合进行sum。就可以得出sql的结果。

展开 ▾

作者回复: 是的



落叶飞逝的...

2019-03-04

👍 1

回过头来继续看文章，目前使用的是这个搭建的阿里云伪集群
<http://archive.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.7.0/>



blacksmith...

2018-11-21

👍 1

后面一段话，一看就是好人，好老师。

展开 ▾

作者回复: 谢谢，共勉。



