**MapReduce产生背景**

Log日志：10万个文件，10T

工作：统计里面 某个单词出现的总次数

**上面的想法虽好，但是实际做起来却很困难，比如如何解决一下问题**

这个设计相对上一个设计最大的不同：

* 上一个设计：数据移动到运算，客户端在一个节点，数据存储在其它的多个节点，客户端通过网络在其它的节点一点一点的读取数据。
* 这个设计：运算移动到数据，数据在哪里，我们就在那里运算。这么做其实运算也变成了一个分布式的了，这么做的好处：
* 那么程序在读取数据的时候就会节省很多时间，不用通过网络传输

**将程序分发到集群的每一台DN上去统计**

现在的情况：10T数据，用多个节点存储，任何一个节点存储的都是某个文件的某些块

我们的做法：写一个客户端，我们从HDFS上读取数据，读一点就计算一点

新的问题：

* 不知道你有没有发现你的程序还是一个单机版，如果有10T数据，那么你要慢慢读完这10T数据，而且因为数据存储在HDFS上，所以数据的读取是通过网络的(这样数据更慢)，请问你准备都多久？
* 你要去处理数据，那你避免不掉的要保存一定的数据到内存，如果是单机，你确定你电脑的内存够用吗？

**解决海量数据的存储问题，HDFS**

首先：处理逻辑简单，读取内容，判断，+1

问题：如果一份日志有10T，请问单机如何存储

**单机**

* 代码怎么拷贝到机器：用U盘吗？
* 怎么启动每一个节点的程序：手动的一个一个的启动吗？当你启动到最后一个的时候第一个可能已经运行完了
* 如果有100个节点，但是我们只用了30个节点来存储数据，因此我们只需要将程序拷贝到这30个节点。可是你又如何准确的知道是哪几个节点？
* 假如以上的工作你都完成了，那么如果有一个机器宕机了呢？
* 还有我们现在只是计算了每一个节点的数据，请问这些数据如何汇总呢？

**要解决上面的所有问题，你就得花费很多的时间去设计，去实现**

MapReduce帮我们解决了所有的问题，我们只需要将重点放在业务逻辑上