**什么是数据倾斜**   
    简单的讲，数据倾斜就是我们在计算数据的时候，数据的分散度不够，导致大量的数据集中到了一台或者几台机器上计算，这些数据的计算速度远远低于平均计算速度，导致整个计算过程过慢。   
    相信大部分做数据的童鞋们都会遇到数据倾斜，数据倾斜会发生在数据开发的各个环节中，比如：

* 用Hive算数据的时候reduce阶段卡在99.99%
* 用SparkStreaming做实时算法时候，一直会有executor出现OOM的错误，但是其余的executor内存使用率却很低。   
      数据倾斜有一个关键因素是数据量大，可以达到千亿级。

**数据倾斜长的表现**

    以Hadoop和Spark是最常见的两个计算平台，下面就以这两个平台说明：

1、Hadoop中的数据倾斜

    Hadoop中直接贴近用户使用使用的时Mapreduce程序和Hive程序，虽说Hive最后也是用MR来执行（至少目前Hive内存计算并不普及），但是毕竟写的内容逻辑区别很大，一个是程序，一个是Sql，因此这里稍作区分。

Hadoop中的数据倾斜主要表现在ruduce阶段卡在99.99%，一直99.99%不能结束。   
这里如果详细的看日志或者和监控界面的话会发现：

* 有一个多几个reduce卡住
* 各种container报错OOM
* 读写的数据量极大，至少远远超过其它正常的reduce   
  伴随着数据倾斜，会出现任务被kill等各种诡异的表现。

**经验：** Hive的数据倾斜，一般都发生在Sql中Group和On上，而且和数据逻辑绑定比较深。

2、Spark中的数据倾斜

    Spark中的数据倾斜也很常见，这里包括Spark Streaming和Spark Sql，表现主要有下面几种：

* Executor lost，OOM，Shuffle过程出错
* Driver OOM
* 单个Executor执行时间特别久，整体任务卡在某个阶段不能结束
* 正常运行的任务突然失败

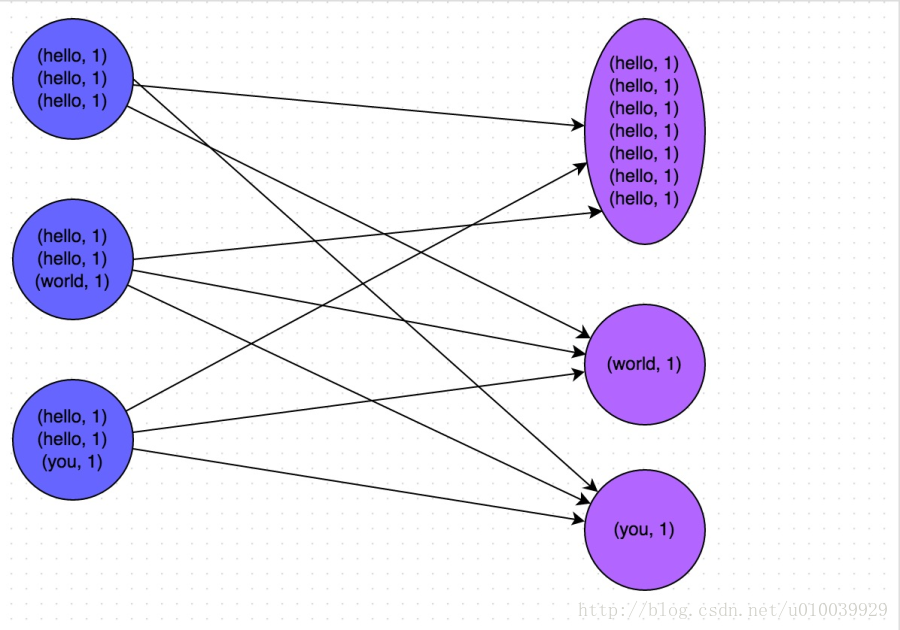
补充一下，在Spark streaming程序中，数据倾斜更容易出现，特别是在程序中包含一些类似sql的join、group这种操作的时候。 因为Spark Streaming程序在运行的时候，我们一般不会分配特别多的内存，因此一旦在这个过程中出现一些数据倾斜，就十分容易造成OOM。

**数据倾斜的原理**

1、数据倾斜产生的原因   
        我们以Spark和Hive的使用场景为例。他们在做数据运算的时候会设计到，countdistinct、group by、join等操作，这些都会触发Shuffle动作，一旦触发，所有相同key的值就会拉到一个或几个节点上，就容易发生单点问题。

2、万恶的shuffle   
        Shuffle是一个能产生奇迹的地方，不管是在Spark还是Hadoop中，它们的作用都是至关重要的。那么在Shuffle如何产生了数据倾斜？

Hadoop和Spark在Shuffle过程中产生数据倾斜的原理基本类似。如下图。



        大部分数据倾斜的原理就类似于下图，很明了，因为数据分布不均匀，导致大量的数据分配到了一个节点。

3、从业务计角度来理解数据倾斜

        数据往往和业务是强相关的，业务的场景直接影响到了数据的分布。再举一个例子，比如就说订单场景吧，我们在某一天在北京和上海两个城市多了强力的推广，结果可能是这两个城市的订单量增长了10000%，其余城市的数据量不变。然后我们要统计不同城市的订单情况，这样，一做group操作，可能直接就数据倾斜了。

**如何解决**

        数据倾斜的产生是有一些讨论的，解决它们也是有一些讨论的，本章会先给出几个解决数据倾斜的思路，然后对Hadoop和Spark分别给出一些解决数据倾斜的方案。   
一、几个思路   
    解决数据倾斜有这几个思路：   
        1.业务逻辑，我们从业务逻辑的层面上来优化数据倾斜，比如上面的例子，我们单独对这两个城市来做count，最后和其它城市做整合。   
        2.程序层面，比如说在Hive中，经常遇到count（distinct）操作，这样会导致最终只有一个reduce，我们可以先group 再在外面包一层count，就可以了。   
        3.调参方面，Hadoop和Spark都自带了很多的参数和机制来调节数据倾斜，合理利用它们就能解决大部分问题。

二、从业务和数据上解决数据倾斜

        很多数据倾斜都是在数据的使用上造成的。我们举几个场景，并分别给出它们的解决方案。   
数据分布不均匀：   
前面提到的“从数据角度来理解数据倾斜”和“从业务计角度来理解数据倾斜”中的例子，其实都是数据分布不均匀的类型，这种情况和计算平台无关，我们能通过设计的角度尝试解决它。

* 有损的方法：   
              找到异常数据，比如ip为0的数据，过滤掉
* 无损的方法：   
              对分布不均匀的数据，单独计算   
              先对key做一层hash，先将数据打散让它的并行度变大，再汇集   
  •数据预处理

三、Hadoop平台的优化方法

    列出来一些方法和思路，具体的参数和用法在官网看就行了。

        1.mapjoin方式   
        2.count distinct的操作，先转成group，再count   
        3.hive.groupby.skewindata=true   
        4.left semi jioin的使用   
        5.设置map端输出、中间结果压缩。（不完全是解决数据倾斜的问题，但是减少了IO读写和网络传输，能提高很多效率）

四、Spark平台的优化方法   
    列出来一些方法和思路，具体的参数和用法在官网看就行了。   
        1.mapjoin方式   
        2.设置rdd压缩   
        3.合理设置driver的内存   
        4.Spark Sql中的优化和Hive类似，可以参考Hive

总结

数据倾斜的坑还是很大的，如何处理数据倾斜是一个长期的过程，希望本文的一些思路能提供帮助。文中一些内容没有细讲，比如Hive Sql的优化，数据清洗中的各种坑，这些留待后面单独的分享，会有很多的内容。另外千亿级别的数据还会有更多的难点，不仅仅是数据倾斜的问题，这一点在后面也会有专门的分享。