

Spark和Hadoop作业之间的区别

Spark目前被越来越多的企业使用，和Hadoop一样，Spark也是以作业的形式向集群提交任务，那么在内部实现Spark和Hadoop作业模型都一样吗？答案是不对的。

熟悉Hadoop的人应该都知道，用户先编写好一个程序，我们称为Mapreduce程序，一个Mapreduce程序就是一个Job，而一个Job里面可以有一个或多个Task，Task又可以区分为Map Task和Reduce Task，如下图所示：



Spark技术博客: <http://www.iteblog.com>

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：[iteblog_hadoop](#)

而在Spark中，也有Job概念，但是这里的Job和Mapreduce中的Job不一样，它不是作业的最高级别的粒度，在它只上还有Application的概念。我们先来看看Spark文档是怎么定义Application，Task，Job和Stage的：

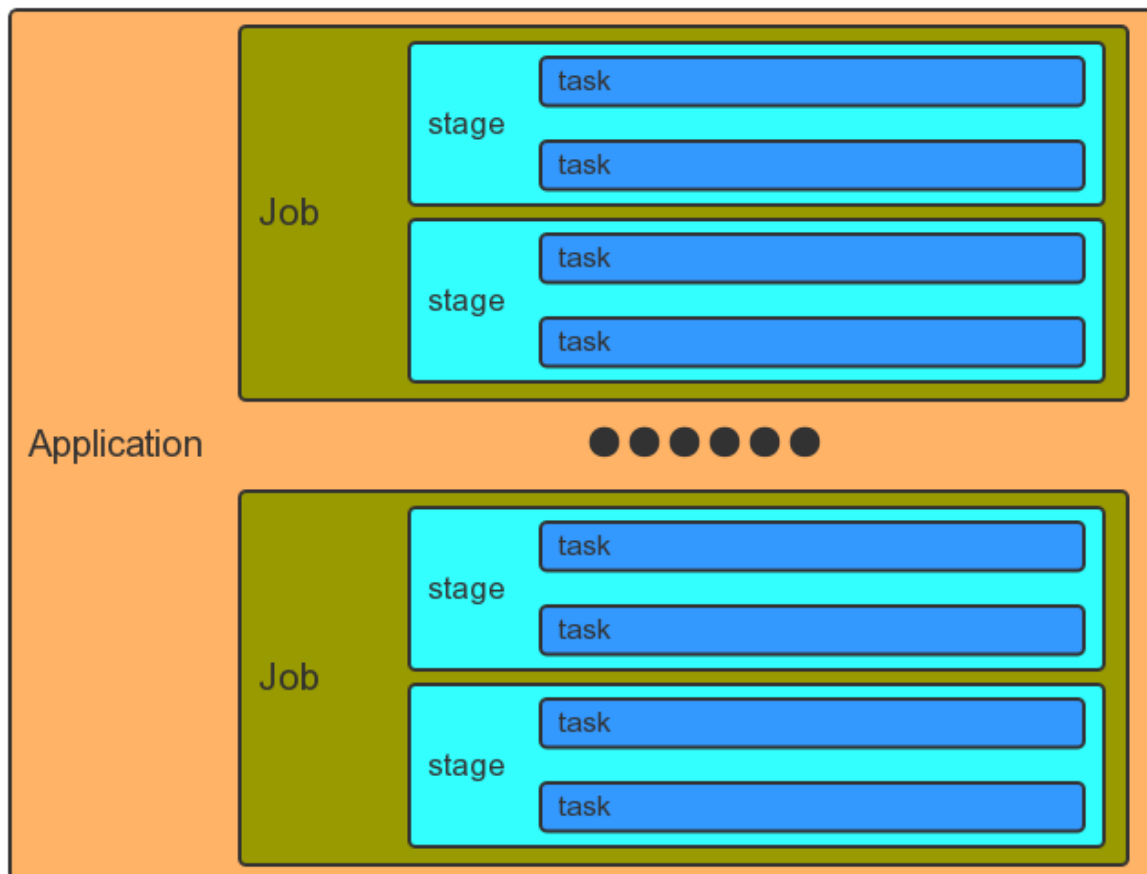
Application : User program built on Spark. Consists of a driver program and executors on the cluster.

Task : A unit of work that will be sent to one executor

Job : A parallel computation consisting of multiple tasks that gets spawned in response to a Spark action (e.g. save, collect); you'll see this term used in the driver's logs.

Stage : Each job gets divided into smaller sets of tasks called stages that depend on each other (similar to the map and reduce stages in MapReduce); you'll see this term used in the driver's logs.

一个Application和一个SparkContext相关联，每个Application中可以有一个或多个Job，可以并行或者串行运行Job。Spark中的一个Action可以触发一个Job的运行。在Job里面又包含了多个Stage，Stage是以Shuffle进行划分的。在Stage中又包含了多个Task，多个Task构成了Task Set。他们之间的关系如下图所示：



Spark技术博客: <http://www.iteblog.com>

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：[iteblog_hadoop](#)

Mapreduce中的每个Task分别在自己的进程中运行，当该Task运行完的时候，该进程也就结束了。和Mapreduce不一样的是，Spark中多个Task可以运行在一个进程里面，而且这个进程的生命周期和Application一样，即使没有Job在运行。

这个模型有什么好处呢？可以加快Spark的运行速度！Tasks可以快速地启动，并且处理内存中的数据。但是这个模型有的缺点就是粗粒度的资源管理，每个Application拥有固定数量的executor和固定数量的内存。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#) ()