Spark性能优化—开山篇

– 学习Spark的性能优化以及学习Spark性能优化的意义

首先，学习了Spark，而没有掌握关于Spark性能优化的相关知识，就好像你学习了一门技术，而根本就没掌握如何利用这门技术去为你的项目服务。总所周知，Spark之所以高效，是有很多原因的，例如基于内存、迭代式计算、容错等等。既然Spark有如此多的特性，是不是就说明Spark已经非常完美了呢？答案是否定的。高性能必然会带来一些实际操控的问题，如内存溢出、任务丢失、效率提升等等。这些实际问题也就引出了我们的主题 – Spark性能优化。大家都知道，目前Spark不稳定的主要原因就是：基于内存。在实际的生产环境中，数据量是远远高于你的测试环境的，就算你的Spark程序在测试环镜中圆满的达到预期，但在实际生产环境中，还是有很大的可能出现像OOM这样的问题。那么，如何解决这样的问题，也就涉及到了你的Spark功力有多强，而你的Spark功力的强弱也恰恰是体现在了，你对Spark应用的性能调优。其实，从另一角度讲，并不是Spark不稳定，而是你无法驾驭Spark，无法令如此优秀的大数据框架发挥出它的潜能。在实际的工作中，这样的场景比比皆是，只有学好并熟练掌握了Spark性能优化的相关知识，你才能查准问题，从而最高效的解决问题。

其次，学习Spark性能优化之前，我们应该具有什么样的知识储备呢？Scala语言，这是最基本的，因为Spark就是用Scala写的，不会Scala还搞什么Spark；除此之外，最重要的就是Spark的源码和内核。对于任何一个有过实际开发经验的开发人员来讲，无论你学习哪一个框架技术，若不了解这个框架的内核的话，你是根本无法驾驭这个框架的，也是根本不可能成为这个框架应用者中的高手，而只是一个搬砖工而已，也就更无从谈起进入国内外大型公司，如百度、阿里、腾讯、Intel、IBM等等，因为越顶级的大公司越需要Spark顶级的高手。而衡量一个Spark高手的基本标准就是：你对Spark内核的运行原理的了解。面试中，诸如此类的问题也是贯穿其中。也只有在了解并彻底掌握了Spark的内核，并阅读了一定遍数的Spark源码之后，你也才可能在实际生产环境中解决一切线上问题。举个简单的例子，假如你的Spark程序再生产中奔溃了，接下来你要怎么做？哭，恨老天为什么为难你；跑，免得被老板骂。当然，这些都不是一个Spark高手接下来该做的事情。作为一个Spark高手，一旦遇到这样的问题，第一反应就是看日志，因为日志中记载着程序运行的轨迹、堆栈调用的信息，而这些信息又全部都是源于内核的。只有看懂了这些信息，你才能知道问题究竟处在哪里，你才能在最短时间内解决问题。所以，要想解决一切实际生产中遇到的问题，都是要从内核出发，寻找问题的根源来解决问题。我们暂且先不把问题讲的如此严重，假设你写的Spark程序在线上运行正常，但是，老板和客户就是不满意你的程序，因为它跑的太慢了，还不如用Hadoop来跑呢。这个时候你又该怎么办？听之任之，告诉你的老板和客户，反正你写的东西就这个样子，等等就是了？这说明你根本不了解Spark，而你的老板也开始准备磨刀了。**作为一个真正了解Spark的高手，当遇到性能调优的问题时，一定会从Spark的内核入手，对症下药，来提升运行程序的执行速度**。最后，你要明白，**在精通了Spark内核源码之后，你能做到：1. 当故障出现的时候，你能知道故障是什么，故障为什么会出现，以及快速的解决故障；2. 精通Spark的性能优化；3. 这点也是最高境界，就是改造Spark框架的实现。**

综上所诉，大家便不难理解，学习Spark不学习内核，就好像你学武功，不练内功，而只学些招式吓唬人。**只有在学会了Spark内核的基础上，熟练的掌握了Spark性能调优的相关知识，你才能成为真正的“武林高手**”；才能在任何的场景下，一招毙命你的敌人；才能让大公司对你刮目相看；最后，才能真正实现你年薪百万的梦想！