Python和数据分析入门

睡不着觉的时候，一般人是数羊，Jeff Dean则是map reduce他的羊群！

七月在线 大林老师 2017/05/15
微博：http://weibo.com/u/2607195824

目录

- 多进程
- 多线程
- 进程 vs 线程
- 异步编程
- 函数式编程简介
- Hadoop简介
- Spark简介

多进程

- fork与子进程
  - 因为操作系统自动把当前进程（称为父进程）复制了一份（称为子进程），然后，分别在父进程和子进程内返回。
- 跨平台的multiprocessing库
- 进程池
  - 合理分配资源
  - 避免手工调度

多进程

- 进程间通信
  - 共享变量
  - Queue和Pipe
  - 互斥锁
- 使用subprocess创建并控制子进程

多线程

- 线程与进程的区别
- 互斥与线程局部变量
  - 互斥的成本
  - 线程局部变量使用场景与优点
- 线程池
  - 合理分配资源
  - 避免手工调度

进程 vs 线程

- GIL锁：
  - 尽管Python完全支持多线程编程， 但是解释器的C语言实现部分在完全并行执行时并不是线程安全的。 实际上，解释器被一个全局解释器锁保护着，它确保任何时候都只有一个Python线程执行。
  - 计算密集型任务的性能受到严重影响！
- 多进程模式最大的优点就是稳定性高，因为一个子进程崩溃不会影响主进程和其他子进程，缺点是创建进程的代价大。
- 多线程模式通常比多进程快一点，而且，多线程模式致命的缺点就是任何一个线程挂掉都可能直接造成整个进程崩溃，因为所有线程共享进程的内存。
- 无论是多进程还是多线程，只要数量一多，效率肯定上不去。

异步编程

- 考虑到CPU和IO之间巨大的速度差异，一个任务在执行的过程中大部分时间都在等待IO操作，单进程单线程模型会导致别的任务无法并行执行，因此，我们才需要多进程模型或者多线程模型来支持多任务并发执行。
- [补图]

异步编程

- 现代操作系统对IO操作已经做了巨大的改进，最大的特点就是支持异步IO。如果充分利用操作系统提供的异步IO支持，就可以用单进程单线程模型来执行多任务，这种全新的模型称为事件驱动模型

异步编程

- 协程与子程序的区别
  - 子程序调用总是一个入口，一次返回，调用顺序是明确的。
  - 协程看上去也是子程序，但执行过程中，在子程序内部可中断，然后转而执行别的子程序，在适当的时候再返回来接着执行。
- 为什么协程更有性能优势？
  - 子程序切换不是线程切换，而是由程序自身控制，因此，没有线程切换的开销，和多线程比，线程数量越多，协程的性能优势就越明显。
  - 不需要多线程的锁机制，因为只有一个线程，也不存在同时写变量冲突，在协程中控制共享资源不加锁，只需要判断状态就好了，所以执行效率比多线程高很多。

函数式编程简介

- 函数式编程的亮点主张：
  - 函数是第一等公民
  - 纯函数，没有副作用
- 函数式编程与分布式
  - 函数式编程不需要考虑"死锁"（deadlock），因为它不修改变量，所以根本不存在"锁"线程的问题。不必担心一个线程的数据，被另一个线程修改，所以可以很放心地把工作分摊到多个线程，部署"并发编程"（concurrency）。

```
var s1 = op1()
var s2 = op2()
var s3 = op(s1, s2)
```

函数式编程简介

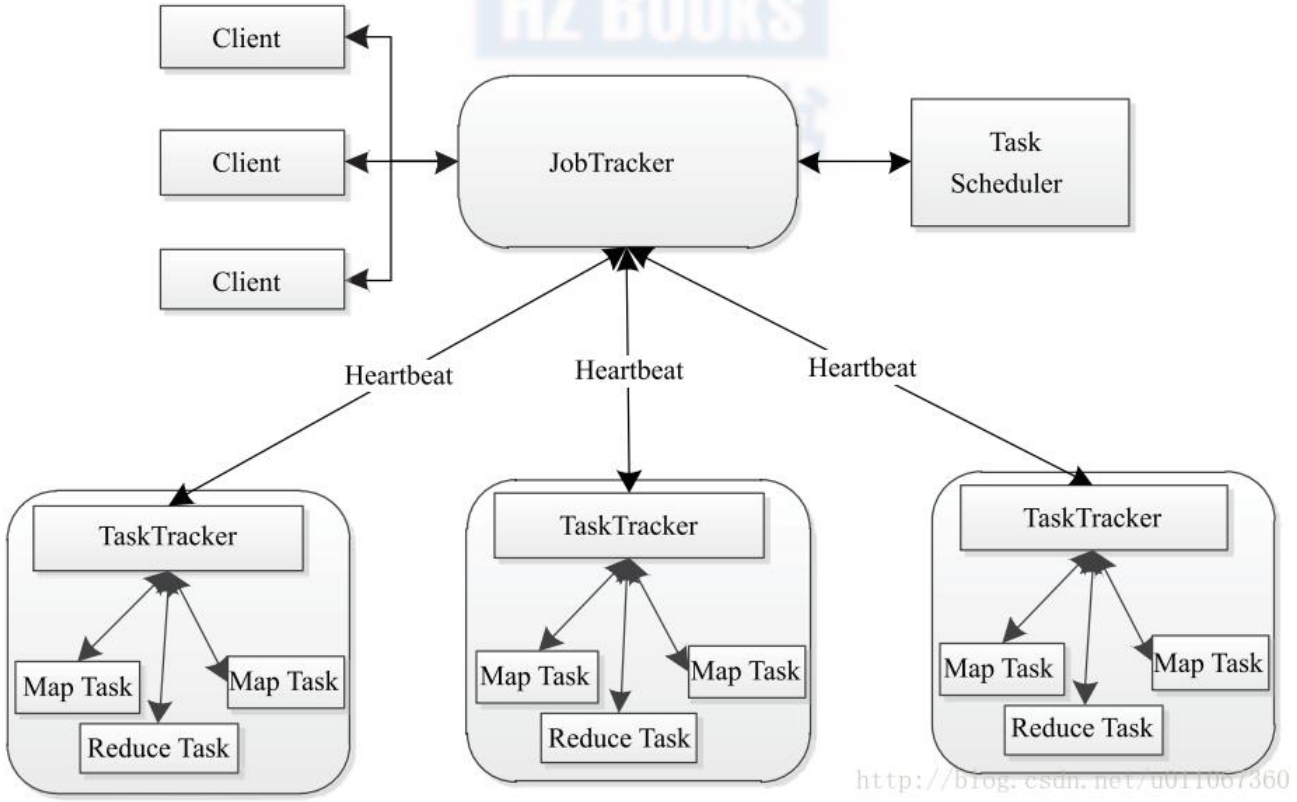
- Python的高阶函数
  - map/reduce
  - filter
  - Sorted
- 闭包与返回函数
- 偏函数
- 匿名函数

Hadoop简介

- 什么是Hadoop
- Hadoop解决的主要问题
  - 海量数据存储 HDFS
  - 海量数据分析 MapReduce
  - 资源管理调度 YARN

Hadoop简介

- Map/Reduce 架构示意图



Hadoop简介

- 通过Hadoop Streaming支持Python
  - mapper和reducer会从标准输入中读取用户数据，一行一行处理后发送给标准输出。Streaming工具会创建MapReduce作业，发送给各个tasktracker，同时监控整个作业的执行过程。
  - 如果一个文件（可执行或者脚本）作为mapper，mapper初始化时，每一个mapper任务会把该文件作为一个单独进程启动，mapper任务运行时，它把输入切分成行并把每一行提供给可执行文件进程的标准输入。 同时，mapper收集可执行文件进程标准输出的内容，并把收到的每一行内容转化成key/value对，作为mapper的输出。

Hadoop简介

- Python实战
  - 部署单机版Hadoop安装与配置
  - 实现map/reducer
  - 用命令行管道检查map/reducer正确工作
  - 上传数据文件
  - 用streaming启动任务
  - 读取结果

Spark简介

- Spark是基于内存计算的大数据并行计算框架.Spark基于内存计算，提高了在大数据环境下数据处理的实时性,同时保证了高容错性和高可伸缩性,允许用户将Spark部署在大量的廉价硬件之上,形成集群
- 为什么说Spark比Hadoop快
  - 中间结果写入缓存而不是磁盘
  - DAG算模型（有向无环图）

Spark简介

- 单机版提交任务：./bin/spark-submit --class org.apache.spark.examples.SparkPi ./examples/jars/spark-examples_2.11-2.1.1.jar 100
- Scala速成
  - 函数
  - 类
  - 特质（trait）
  - 类型
  - 单例对象

作业

- 部署Hadoop和Spark单机学习环境，实现最简单word count例子。
- 用线程池实现矩阵乘法运算
- 自己实现异步网页抓取的例子