飞一般的感觉！当Spark遇到Redis~

2016-03-29 炼数成金订阅号

一些内存数据结构比其他数据结构来得更高效;如果充分利用Redis，Spark运行起来速度更快。

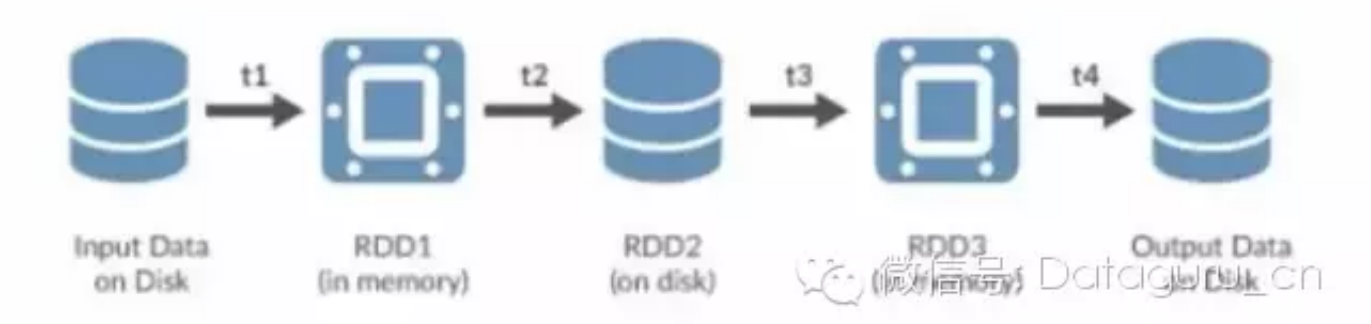
Apache Spark已逐渐俨然成为下一代大数据处理工具的典范。通过借鉴开源算法，并将处理任务分布到计算节点集群上，无论在它们在单一平台上所能执行的数据分析类型方面，还是在执行这些任务的速度方面，Spark和Hadoop这一代框架都轻松胜过传统框架。Spark利用内存来处理数据，因而速度比基于磁盘的Hadoop大幅加快(快100倍)。

但是如果得到一点帮助，Spark可以运行得还要快。如果结合Spark和Redis(流行的内存数据结构存储技术)，你可以再次大幅提升处理分析任务的性能。这归功于Redis经过优化的数据结构，以及它在执行操作时，能够尽量降低复杂性和开销。通过借助连接件访问Redis数据结构和API，Spark可以进一步加快速度。

提速幅度有多大?如果Redis和Spark结合使用，结果证明，处理数据(以便分析下面描述的时间序列数据)的速度比Spark单单使用进程内存或堆外缓存来存储数据要快45倍——不是快45%，而是快整整45倍!

为什么这很重要?许多公司日益需要分析交易的速度与业务交易本身的速度一样快。越来越多的决策变得自动化，驱动这些决策所需的分析应该实时进行。Apache Spark是一种出色的通用数据处理框架;虽然它并非百分之百实时，还是往更及时地让数据发挥用途迈出了一大步。

Spark使用弹性分布式数据集(RDD)，这些数据集可以存储在易失性内存中或HDFS之类的持久性存储系统中。RDD不会变化，分布在Spark集群的所有节点上，它们经转换化可以创建其他RDD。



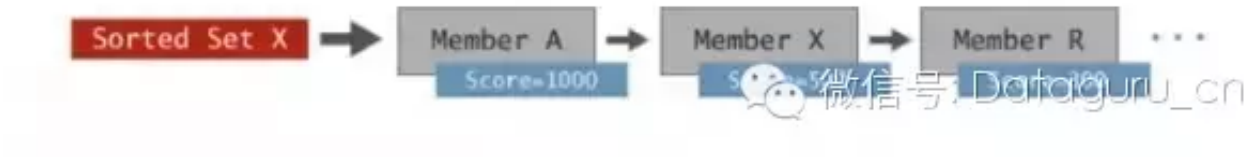
Spark RDD

RDD是Spark中的重要抽象对象。它们代表了一种高效地将数据呈现给迭代进程的容错方

法。由于处理工作在内存中进行，这表示相比使用HDFS和MapReduce，处理时间缩短了好几个数量级。

Redis是专门为高性能设计的。亚毫秒延迟得益于经过优化的数据结构，由于让操作可以在邻近数据存储的地方执行，提高了效率。这种数据结构不仅可以高效地利用内存、降低应用程序的复杂性，还降低了网络开销、带宽消耗量和处理时间。Redis数据结构包括字符串、集合、有序集合、哈希、位图、hyperloglog和地理空间索引。开发人员可以像使用乐高积木那样使用Redis数据结构——它们就是提供复杂功能的简单管道。

为了直观地表明这种数据结构如何简化应用程序的处理时间和复杂性，我们不妨以有序集合(Sorted Set)数据结构为例。有序集合基本上是一组按分数排序的成员。

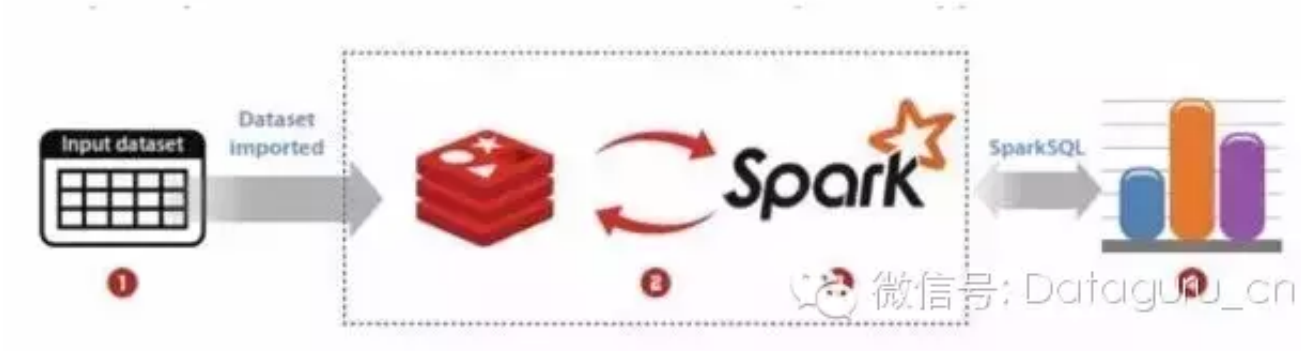


Redis有序集合

你可以将多种类型的数据存储在这里，它们自动由分数来排序。存储在有序集合中的常见数据类型包括：物品(按价格)、商品名称(按数量)、股价等时间序列数据，以及时间戳等传感器读数。

有序集合的魅力在于Redis的内置操作，让范围查询、多个有序集合交叉、按成员等级和分数检索及更多事务可以简单地执行，具有无与伦比的速度，还可以大规模执行。内置操作不仅节省了需要编写的代码，内存中执行操作还缩短了网络延迟、节省了带宽，因而能够实现亚毫秒延迟的高吞吐量。如果将有序集合用于分析时间序列数据，相比其他内存键/值存储系统或基于磁盘的数据库，通常可以将性能提升好几个数量级。

Redis团队的目标是提升Spark的分析功能，为此开发了Spark-Redis连接件。这个程序包让Spark得以使用Redis作为其数据源之一。该连接件将Redis的数据结构暴露在Spark面前，可以针对所有类型的分析大幅提升性能。



Spark Redis连接件

为了展示给Spark带来的好处，Redis团队决定在几种不同的场景下执行时间片(范围)查询，

以此横向比较Spark中的时间序列分析。这几种场景包括：Spark在堆内内存中存储所有数据，Spark使用Tachyon作为堆外缓存，Spark使用HDFS，以及结合使用Spark和Redis。

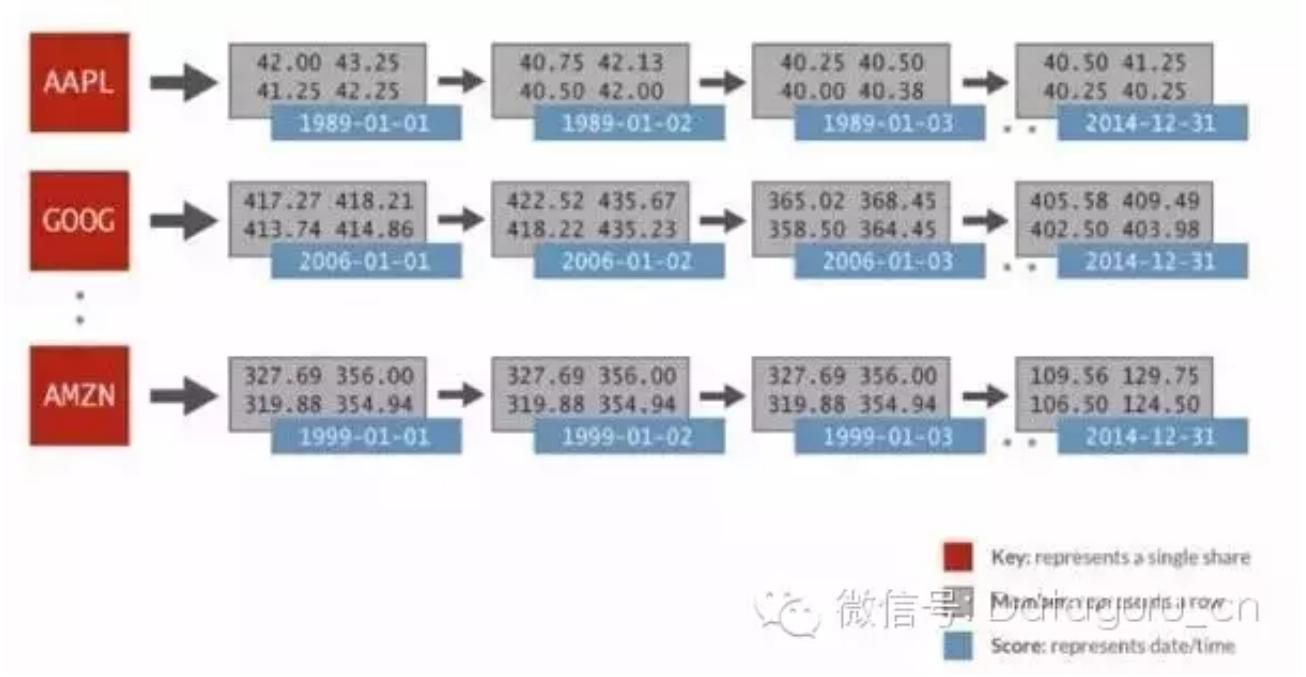
Redis团队使用Cloudera的Spark时间序列程序包，构建了一个Spark-Redis时间序列程序包，使用Redis有序集合来加快时间序列分析。除了让Spark可以访问Redis的所有数据结构外，该程序包另外做两件事：

自动确保Redis节点与Spark集群一致，从而确保每个Spark节点使用本地Redis数据，因而优化延迟。

与Spark数据帧和数据源API整合起来，以便自动将Spark SQL查询转换成对Redis中的数据来说最高效的那种检索机制。

简单地说，这意味着用户不必担心Spark和Redis之间的操作一致性，可以继续使用Spark SQL来分析，同时大大提升了查询性能。

用于这番横向比较的时间序列数据包括：随机生成的金融数据，每天1024支股票，时间范围是32年。每只股票由各自的有序集合来表示，分数是日期，数据成员包括开盘价、最高价、最低价、收盘价、成交量以及调整后的收盘价。下图描述了用于Spark分析的Redis有序集合中的数据表示：



Spark Redis时间序列

在上述例子中，就有序集合AAPL而言，有表示每天(1989-01-01)的分数，还有全天中表示为一个相关行的多个值。只要在Redis中使用一个简单的ZRANGEBYSCORE命令，就可以执行这一操作：获取某个时间片的所有值，因而获得指定的日期范围内的所有股价。Redis执行这种类型的查询的速度比其他键/值存储系统快100倍。

这番横向比较证实了性能提升。结果发现，Spark使用Redis执行时间片查询的速度比Spark使用HDFS快135倍，比Spark使用堆内(进程)内存或Spark使用Tachyon作为堆外缓存快45倍。下图显示了针对不同场景所比较的平均执行时间：

Spark Redis横向比较

如果你想亲自尝试一下，不妨遵照这篇可下载的逐步指南：《Spark和Redis使用入门》(https://redislabs.com/solutions/spark-and-redis)。该指南将逐步引导你安装典型的Spark集群和Spark-Redis程序包。它还用一个简单的单词计数例子，表明了可以如何结合使用Spark和Redis。你在试用过Spark和Spark-Redis程序包后，可以进一步探究利用其他Redis数据结构的更多场景。

虽然有序集合很适合时间序列数据，但Redis的其他数据结构(比如集合、列表和地理空间索引)可以进一步丰富Spark分析。设想一下：一个Spark进程试图根据人群偏好以及邻近市中心，获取在哪个地区发布新产品效果最好的信息。现在设想一下，内置分析自带的数据结构(比如地理空间索引和集合)可以大大加快这个进程。Spark-Redis这对组合拥有无限的应用前景。

Spark支持一系列广泛的分析，包括SQL、机器学习、图形计算和Spark Streaming。使用Spark的内存处理功能只能让你达到一定的规模。然而有了Redis后，你可以更进一步：不仅可以通过利用Redis的数据结构来提升性能，还可以更轻松自如地扩展Spark，即通过充分利用Redis提供的共享分布式内存数据存储机制，处理数百万个记录，乃至数十亿个记录。

时间序列这个例子只是开了个头。将Redis数据结构用于机器学习和图形分析同样有望为这些工作负载带来执行时间大幅缩短的好处。

文章来源：云头条

欢迎加入本站公开兴趣群

软件开发技术群

兴趣范围包括：Java，C/C++，Python，PHP，Ruby，shell等各种语言开发经验交流，各种框架使用，外包项目机会，学习、培训、跳槽等交流

QQ群：26931708

Hadoop源代码研究群

兴趣范围包括：Hadoop源代码解读，改进，优化，分布式系统场景定制，与Hadoop有关的各种开源项目，总之就是玩转Hadoop

QQ群：288410967

阅读原文