

5-6-cn

1 本节我们以 Spark 为例讨论下内存计算模型

2 数据处理系统提供大数据计算处理能力和应用开发平台。

从计算架构的角度，将数据处理系统分为算法层、计算模型层、计算平台和引擎层等。

与大数据相关的计算算法包括机器学习算法和数据挖掘算法。

计算模型是指不同类型的大数据在不同场景下的处理方式，

包括批处理、流计算、结构化数据的大规模并发处理(MPP)模型、内存计算模型和数据流图模型。

就计算平台和引擎而言，通常具有代表性的大数据处理平台有 Hadoop、Spark、storm、Pregel 等，

本节我们讨论内存计算模型

3 内存内计算(或内存内计算)是一种完全在计算机内存(如 RAM)中运行计算的技术。

这个术语通常意味着大规模、复杂的计算，需要专门的系统软件在集群中一起工作的计算机上运行计算

作为一个集群，计算机将它们的 RAM 聚集在一起，因此计算基本上是在计算机之间运行的，并一起利用所有计算机的集体 RAM 空间。

内存计算是如何工作的？

内存计算通过消除所有缓慢的数据访问并完全依赖存储在 RAM 中的数据来工作。通过消除访问硬盘驱动器或 ssd 时常见的延迟，整体计算性能大大提高。

运行在一台或多台计算机上的软件管理计算和内存中的数据，在多台计算机的情况下，软件将计算分成更小的任务，这些任务分布到每台计算机上并行运行。

4 说到内存计算模型，我们必须谈谈 spark。

Spark 最初由加州大学伯克利分校 AMP 实验室的 Matei Zaharia 于 2009 年创建，并于 2010 年实现开源。

2013 年，捐赠给 Apache 软件基金会。最活跃的开源大数据项目之一，顶级 Apache 项目

基于内存计算模型的并行处理框架。

它可以构建在 Hadoop 平台上，使用 HDFS 文件系统存储数据，但为了支持高效的分布式内存计算，在文件系统之上构建了 RDD (Resilient Distributed dataset)体系结构。

5 2013 年 Hadoop 使用 2100 台机器在 72 分钟内完成 100TB 数据的排序任务。

2014 年 11 月，Spark 创始人 M. Zaharia 的公司 Databricks 在使用 Spark 进行大规模排序方面创造了新的世界纪录。

Spark 使用 207 台机器在 21 分钟内对 100TB 的数据进行排序，这是 Hadoop 使用的机器数量的十分之一，并将时间从 72 分钟减少到 23 分钟。

比 Hadoop 快得多，效率也更高。

Apache Spark 是一个用于大数据工作负载的开源分布式处理系统。它利用内存缓存和优化查询执行，以便对任何大小的数据进行快速查询。简单地说，Spark 是一个用于大规模数据处理的快速通用引擎。

快速的部分意味着它比以前处理大数据的方法更快，比如经典的 MapReduce。更快的秘密是 Spark 运行在内存(RAM)上，这使得处理速度比磁盘驱动器快得多。

通用部分意味着它可以用于多种事情，如运行分布式 SQL、创建数据管道、将数据输入数据库、运行机器学习算法、处理图表或数据流，等等。

6 弹性分布式数据集 RDD (Resilient Distributed Dataset)

弹性分布式数据集(RDD)，它是跨集群节点划分的元素集合，可以并行操作。

rdd 是这样创建的:从 Hadoop 文件系统(或任何其他 Hadoop 支持的文件系统)中的一个文件开始，或者

从驱动程序中的一个现有的 Scala 集合开始，然后对其进行转换。

RDD 的操作只有两种，转换和动作。在转换过程中，可以对数据进行过滤、连接映射、约简，但不进行计算，只在动作中进行计算，并生成值结果。

用户也可以要求 Spark 在内存中持久化一个 RDD，允许它在并行操作中有效重用。最后，rdd 在节点故障时自动恢复。

7 假定输入数据经过第一次迭代运算生成 RDD1，RDD1 分布式的存储在三个不同的物理内存上，然后 RDD1 经过第二次迭代运算生成 RDD2，以此类推

## 8 Spark 工作流

Transform 返回值仍然是一个 RDD。它使用链调用设计模式，在计算一个 RDD 之后，将其转换为另一个 RDD，然后可以再次转换该 RDD。这个过程是分布式的。

Action 是返回给驱动程序或存储在文件中的返回值，它是从 RDD 到结果 的转换，

Action 返回值不是 RDD。Scala 的一个普通集合：一个值，或者 空的，最终它被返回给 Driver 程序，或者 RDD 被写入文件系统。

RDD 执行应该遵循这些 transform 转换的拓扑结构和 RDD 转换的连接关系

Transformation 是从 RDD 到 RDD 的转换。

只有当动作 action 被执行时，才会计算并生成 RDD，这是 RDD 延迟执行的根源。

9 关于 RDD 的容错机制，当某个 RDD 损坏时，系统回溯到上一版本的 RDD，比如图中的 RDD2 损坏了，则系统会找到 RDD1，然后重新执行 iter.2,重新生成 RDD2 的结果，这个就是 RDD 的 recovery 恢复

## 10 Spark 架构和工作原理：

Spark 的计算程序分为 Driver（运行在 Master 节点上，也有一种模式运行在某一 Worker 节点上）和 Executor（运行在 Worker 节点上）两部分：

Driver 负责把应用程序的计算任务转化成有向非循环图（DAG）

Executor 则负责完成 worker 节点上的计算和数据存储

在每个 worker 上， Executor 针对一个个分发给它的数据 partition 再生成一个个 Task 线程，完成并行计算

Spark 的运行过程如下:Spark 架构在分布式计算中采用 Master-Slave 模式。Master 是对应集群中包含 Master 进程的节点，Slave 是集群中包含 Worker 进程的节点。Master 是整个集群的控制器，负责整个集群的正常运行;Worker 相当于一个计算节点，从主节点接收命令并报告状态;Executor 负责执行任务;客户端作为用户的客户端负责提交应用程序，驱动程序负责控制一个应用程序的执行。Spark 集群部署完成后，需要在主节点和从节点上分别启动 Master 和 Worker 进程，以控制整个集群。在 Spark 应用程序的执行过程中，Driver 和 Worker 是两个重要的角色。Driver 程序是应用程序逻辑执行的起点，负责作业调度，即 Task 任务的分配，使用多个 Workers 管理计算节点，创建 Executor 并行处理任务。在执行阶段，驱动程序序列化 Task 和 Task 所依赖的文件和 jar，并将它们传输到相应的 Worker 机器。同时，Executor 处理相应数据分区的任务。Executor /Task 每个程序都有自己的，不同的程序是相互隔离的，任务是多线程并行的，集群对 Spark 是透明的，只要 Spark 能够获取相关的节点和进程，driver 就会与 Executor 保持通信并协同处理

## 11 Spark 的特征

### 1) 内存计算

PySpark 提供内存内计算。计算结果存储在分布式内存(RAM)中，而不是稳定存储(磁盘)中。它提供了非常快的计算

### 2) 懒惰的进化

PySpark rdd 中的转换是懒惰的。它不会立即计算结果，这意味着直到触发操作才开始执行。当我们调用 RDD 中的某个操作进行转换时，它不会立即执行。惰性进化在节省计算开销方面发挥着重要作用。它

通过减少查询的数量来提供优化。

### 3) 容错

rdd 跟踪数据沿袭信息以自动重建丢失的数据。如果 rdd 的任何分区发生故障，则可以从原始的容错输入数据集重新计算该分区，以创建它。

### 4) 不可改变性

创建的数据可以随时检索，但其值不能更改。rdd 只能通过确定性操作创建。

### 5) 分区

rdd 是各种数据项的集合，它们的大小非常大。由于其大小，它们不能装入单个节点，必须跨多个节点进行分区。

### 6) 持久性

它是一种可以保存 RDD 评估结果的优化技术。它存储中间结果，以便在需要时进一步使用它。它降低了计算复杂度。

### 7) 粗粒度的操作

粗粒度操作意味着我们可以转换整个数据集，但不能转换数据集上的单个元素。另一方面，细粒度意味着我们可以转换数据集中的单个元素。

## 12 Spark 组件

Apache Spark Core—Spark Core 是 Spark 平台的底层通用执行引擎，所有其他功能都建立在此基础上。它提供内存计算和引用外部存储系统中的数据集。

Spark SQL - Spark SQL 是 Apache Spark 用于处理结构化数据的模块。Spark SQL 提供的接口为 Spark 提供了更多关于数据结构和正在执行的计算的信息。

Spark Streaming——该组件允许 Spark 处理实时流数据。数据可以从许多来源摄取，如 Kafka、Flume 和 HDFS (Hadoop 分布式文件系统)。然后，可以使用复杂的算法处理数据，并将其推送到文件系统、数据库和实时仪表板。

MLlib(机器学习库)——Apache Spark 配备了一个称为 MLlib 的丰富库。这个库包含了大量的机器学习算法——分类、回归、聚类 and 协同过滤。它还包括用于构建、评估和调优 ML 管道的其他工具。所有这些功能都有助于 Spark 在集群中扩展。

GraphX - Spark 还附带了一个库，用于操作图形数据库和执行称为 GraphX 的计算。GraphX 在单个系统中统一了 ETL(提取、转换和加载)过程、探索性分析和迭代图计算。

## 13 Spark 的优势

快速处理—Apache Spark 最重要的特性 速度。大数据具有海量、多样性、速度、准确性等特点，需要以更高的速度进行处理。Spark 包含弹性分布式数据集(Resilient Distributed Dataset, RDD)，可以节省读写操作的时间，使其运行速度比 Hadoop 快近 10 到 100 倍。

灵活性——支持多种语言，允许开发人员用 Java、Scala、R 或 Python 编写应用程序。

内存计算——Spark 将数据存储在服务器的 RAM 中，这允许快速访问，进而加快分析的速度。

实时处理——Spark 能够处理实时流数据。与 MapReduce 只处理存储数据不同，Spark 能够处理实时数据，因此能够产生即时结果。

更好的分析——与 MapReduce 包含 Map 和 Reduce 功能相比，Spark 包含的功能要多得多。Apache Spark 由一组丰富的 SQL 查询、机器学习算法、复杂分析等组成。有了所有这些功能，分析可以在 Spark 的帮助下以更好的方式执行。

## 14 Spark 生态系统

Spark 是目前最流行的用于大数据分析的开源分布式计算引擎。

Spark 是用于大数据处理和机器学习的行业标准分析引擎，在全球数千个组织中被数据工程师和数据科学家使用。

Spark 使您能够以闪电般的速度处理批处理和流工作负载的数据。Spark 可以运行在 Hadoop、

mongoDB、Cassandra Kubernetes、YARN 或独立平台上——它可以处理广泛的数据输入和输出。

15 本节课我们以 Spark 计算平台为例讲解了内存计算模型，今天的内容就到这里，谢谢大家