# 1.1 Hadoop背景概述

1.能与Hadoop协同工作的处理模式：

1. 交互式SQL。利用MapReduce进行分发并使用一个分布式查询引擎，使得在Hadoop上获得SQL查询低延迟响应的同时还能保持对大数据集规模的可扩展性。这个引擎使用指定的“总是开启”守护进程或容器重用(Tez上的Hive)
2. 迭代处理。例如机器学习算法，其本身带有迭代性，因此和那种每次迭代都从硬盘加载的方式相比，这种在内存中保存每次中间结果集的方式更加高效。MapReduce的结构不允许这样，但如果Spark就会比较直接，它在数据集方面展现了一种高度探究的风格。
3. 流处理。例如Spark Streaming或Samaza使得在无边界数据流上运行实时、分布式的计算，并向Hadoop存储系统或为外部系统发布结果称为可能。
4. 搜索。Solr搜索平台能够在Hadoop集群上运行，当文档加入HDFS后就可对其进行索引，且根据HDFS中存储的索引为搜索查询提供服务。

2.为什么不能用配有大量硬盘的数据库来进行大规模数据分析？为什么需要Hadoop？

1. 计算机硬盘的发展趋势是，**寻址时间的提升远远不敌于传输速率的提升**。寻址就是将磁头移动到特定硬盘位置进行读/写操作的过程。它是导致硬盘操作延迟的主要原因，而传输速率取决于硬盘的带宽。
2. 如果**数据访问模式中包含大量的硬盘寻址**，那么读**取大量数据集就必然花更长的时间**（相较于流数据的读取形式，流读取主要取决于传输速率）。另一方面，如果数据库只更新一部分记录，那么传统的B树（受限于寻址的速率）更有优势。但数据库系统如果有大量数据更新时，B树的效率就明显落后于MapReduce，因为需要使用"排序/合并"来重建数据库。
3. MapReduce可以视为关系型数据库管理系统的补充。MapReduce比较适合批处理分析整个数据集的问题，尤其是一些特定目的的分析。RDBMS适用于索引后数据集的点查询和更新，建立索引的数据库系统嫩巩固提供对小规模数据的低延迟数据检索和快速更新。

3.**MapReduce适合一次写入、多次读取数据的应用，关系型数据库则更适合持续更新的数据集**。下表是关系型数据库和MapReduce的比较。

|  |  |  |
| --- | --- | --- |
|  | 传统的关系型数据库 | MapReduce |
| 数据大小 | GB | PB |
| 数据存取 | 交互式和批处理 | 批处理 |
| 更新 | 多次读/写 | 一次写入，多次读取 |
| 事务 | ACID | 无 |
| 结构 | 写时模式 | 读时模式 |
| 完整性 | 高 | 低 |
| 横向扩展 | 非线性的 | 线性的 |

3.Hadoop和关系型数据库的另一个区别在于它们所操作的数据集的**结构化程度**。**结构化数据是具有既定格式的实体化数据**，如**XML文档**或满足特定预处理格式的数据库表。显然RDBMS能够胜任这类任务。

4.**半结构化数据**比较松散，虽然可能有格式，但经常被忽略，所以它只能为对数据结构的一般性指导。例如电子表格，它在结构上是由单元格组成的网格，但是每个单元格内可以保存任何形式的数据。

5.**非结构化数据**没有什么特别的**内部结构**，例如**纯文本或图像数据**。Hadoop对非结构化数据或半结构化数据非常有效，因为它是在处理数据时才对数据进行解释（即所谓的”读时模式”）。这种模式在提供灵活性的同时避免了RDBMS数据加载阶段带来的高开销，因为在Hadoop中仅仅是一个文件拷贝操作。

6.关系型数据往往是**规范的，以保持其数据完整性且不含冗余**。规范反而给Hadoop处理带来问题，因为它记录读取称为非本地操作，**而Hadoop的核心假设之一偏偏就是可以进行（高速的）流读/写操作**。

7.Web服务器日志时典型的**非规范化数据记录**（例如，每次都需要记录客户端主机全名，这会导致同一客户端的全名被多次出现），**这也是Hadoop非常适用于分析各种日志文件的原因之一**。注意，Hadoop也能够做连接(join)操作，只不过这种操作没有在关系型数据库中用的多。

8.**MapReduce以及Hadoop中其他的处理模型是可以随着数据规模线性伸缩的**。对数据分区后，**函数原语（如Map和Reduce）能够在各个分区上并行工作**。这意味着，如果**输入的数据量是原来的两倍，那么作业的运行时间也需要两倍**。但如果**集群规模扩展为原来对的两倍，那么作业的运行速度却仍然与原来一样快**。SQL查询一般不具备该特性。

9.**Hadoop尽量在计算节点上存储数据**，以**实现数据的本地快速访问**。**数据本地化**特性是**Hadoop数据处理的核心**，并因此而获得**良好的性能**。

10**.Hadoop通过显式网络拓扑结构来保留网络带宽**。这种排列方式并没有降低Hadoop对计算密集型数据进行分析的能力。

11.**MapReduce采用无共享框架**，MapReduce才能够呈现出这种特性，**这意味着各个任务之间是彼此独立的，执行顺序无关紧要**。

12.MapReduce有三大设计目标：

1. 为只需要短短几分钟或几个小时就可以完成的作业提供服务
2. 运行于同一个内部有高速网络连接的数据中心内
3. 数据中心内的计算机都是可靠的、专门的硬件。