4-2-cn

1

在本节中我们讨论数据存储系统中的 分布式文件系统

2

大数据计算系统可归纳为三部分:

数据存储系统，数据处理系统，数据应用系统

数据存储架构是大数据计算的基础。

在数据存储系统中，有 4 个部分来完成不同的任务，

即数据收集与建模、分布式文件系统、分布式数据库/数据仓库和统一数据访问接口。

我们学习了数据收集和建模，现在我们关注分布式文件系统。

实际上，文件系统可以是集中式文件系统，也可以是分布式文件系统，但在大数据场景中，为了实现规模和效率，大多采用分布式文件系统。

3

分布式文件系统为数据提供了物理存储架构。数据存储在服务器上的文件系统。

在分布式文件系统中，数据被访问和处理，就好像它存储在本地客户端机器上一样。

以可控的和授权的方式在网络上的用户之间共享信息和文件很方便，这对用户是透明的。

它达到了单个服务器的成倍存储。

目前，大数据计算架构中的文件系统主要有两种，分别是开源社区的架构 HDFS 和 Google 的 GFS（Google 文件系统）后来演变成 Colossus。

4

在 HDFS 中，每个存储文件首先被划分为固定长度 128MB 的多个数据块，然后这些数据块按照一定的规则分布存储到不同的 Datanode 上，这些数据块被复制成 3 个副本，并按照一定的规则存储在不同的 Data 节点上。容错 ： 当一个数据节点崩溃时，我们仍然可以从另外 2 个数据节点的另外 2 个副本中

检索相同的数据块。数据一旦写入就无法更改，因此 HDFS 中的数据是不可变的。 所以 HDFS 只支持批量读写操作，不支持更新操作。

HDFS 采用主从结构，一个 HDFS 集群包括一个名称节点 name node 也即主节点，以及若干个数据节点 DataNode，也即从节点名称节点作为中心服务节点，负责管理文件系统命名空间、数据文件到数据块到 Datanode 的映射关系，以及客户端对文件的访问调度。

HDFS 还有一个次名称节点 secondary name node， 他定期与主名称节点连接，讲系统目录的即时映像存储在本地磁盘上，当主名称节点实效或者崩溃时，次名称节点可以提供名称节点的回滚恢复和重启功能。

这意味着一个 DataNode 可以存储来自**不同文件的数据块**。

每个数据节点都运行一个节点程序或者进程，负责处理文件系统客户端的读写请求，在名称节点的统一调度下进行数据块的创建、删除和复制等操作。主节点 namenode 与从节点 Datanode 各自执行任务

Namenode

管理文件系统命名空间

保存文件到数据块到数据节点的映射关系

调度客户端对文件的访问

元数据存储在内存中，便于快速访问

Datanode

存储文件数据块

实现数据块到数据节点本地文件系统的映射

数据块存储在本地磁盘上

5

现在我们来了解一下 HDFS 中写入数据的过程，该图总结了 Hadoop 中的文件写入操作。

1) 客户端通过调用 DistributedFileSystem 上的 create() 方法来创建文件。

2) DistributedFileSystem 对 namenode 进行 RPC 调用，以在文件系统的命名空间中创建一个新文件，其中没有与之关联的块。

3) namenode 执行各种检查以确保文件不存在并且客户端具有创建文件的正确权限。如果所有这些检查都通过，namenode 会记录新文件；否则，文件创建失败并且客户端被抛出一个 IOException。分布式文件系统返回一个 FSDataOutputStream 供客户端开始向数据节点写入数据。 FSDataOutputStream 包装了一个 DFSOutputStream，它处理与数据节点和名称节点的通信。

4) 当客户端写入数据时，DFSOutputStream 将其拆分为数据包，并将其写入内部队列，称为数据队列。数据队列由 DataStreamer 使用，它负责通过选择合适的数据节点列表来请求名称节点分配新块以存储副本。

datanodes 列表组成一个 pipeline，默认 replication level 是 3，所以 pipeline 中有 3 个 node。DataStreamer 将数据包流式传输到管道中的第一个数据节点，该数据节点存储数据包并将其转发到管道中的第二个数据节点。

5) 类似地，第二个数据节点存储数据包并将其转发到管道中的第三个（也是最后一个）数据节点。

6) DFSOutputStream 还维护一个内部数据包队列，等待数据节点确认，称为确认队列。只有当管道中的所有数据节点都确认了一个数据包时，它才会从 ack 队列中删除。

7) 当客户端完成写入数据时，它会在流上调用 close()。它将所有剩余的数据包刷新到数据节点管道并等待确认，然后再联系名称节点以发出文件已完成的信号。名称节点已经知道文件是由哪些数据块组成的，所以它只需要等待块被最小化复制就可以成功返回。

6

HDFS 2.0 中读取数据的过程是这样的，

1、客户端通过调用 DistributedFileSystem 上的 open() 方法打开文件。

2、DistributedFileSystem 对 namenode 进行 RPC 调用，以确定文件以块形式存储的数据节点的位置。对于每个块，namenode 返回具有块副本的数据节点的地址（块和数据节点的元数据）。数据节点根据邻近度（取决于网络拓扑信息）进行排序。 DistributedFileSystem 向客户端返回一个 FSDataInputStream（支持文件搜索的输入流）以供其读取数据。 FSDataInputStream 然后包装了一个 DFSInputStream，它管理 datanode 和 namenode I/O。

3、然后客户端在流上调用 read()。存储了文件中前几个块的数据节点地址的 DFSInputStream ，然后连接到文件中第一个块的第一个（最近的）数据节点。

4、数据从数据节点（以数据包的形式）流回客户端，并且客户端在流上重复调用 read()。

5、到达块的末尾时，DFSInputStream 将关闭与数据节点的连接，然后为下一个数据块找到最佳数据节点 datanode

6、当客户端完成读取后，它会在 FSDataInputStream 上调用 close()

此外，在读取过程中，如果 DFSInputStream 在与数据节点通信时遇到错误，它将尝试该块的下一个最接近的数据节点。它还将记住已失败的数据节点，以便在以后的块中不必要地重试它们。

DFSInputStream 还验证从数据节点传输到它的数据的校验和。如果发现损坏的块， DFSInputStream 会尝试从另一个数据节点读取该块的副本；它还将损坏的块报告给名称节点。

7

本节我们以 HDFS 为例学习了大数据分布式文件系统机制，即大数据的物理存储。

我们了解了 HDFS 的架构、名称节点、数据节点以及它们的职责，容错机制。

我们还学习了 HDFS 的数据写入和数据读取过程。

8

今天的课就学习到这里，谢谢大家