三驾马车论文读后感

18301107 王磊

读完关于谷歌誉为“三驾马车”的三篇重要论文后，我对于大数据和分布式系统有了粗浅的了解。受限于技术水平，在查阅资料和询问同学后依然有着许多的问题，以下是我关于GFS 的一些理解。

GFS是一个可扩展的分布式文件系统，用于大型的、分布式的、对大量数据进行访问的应用。它运行于廉价的普通硬件上，并提供容错功能。它可以给大量的用户提供总体性能较高的服务。

一、单一Master节点

只有一个master极大的简化了设计并使得master可以根据全局情况作出合适的块的放置和复制决策。

要考虑如何避免单一的Msaster节点成为系统性能瓶颈：

1、客户端只向Master请求元数据信息，并不通过Master节点进行chunk数据读写，具体的数据读写操作由ChunkServer直接负责。

2、客户端在向Master请求某chunk元数据时,Master会一次返回包括该chunk紧接之后的几个chunk信息，有效减少客户端的请求次数。

二、Chunk尺寸

选择64MB作为chunk的大小，该尺寸远大于一般系统的Block size。这是由具体的业务特性决定的：即通常操作的是大文件。

1. 可以减少客户端与Master的通信需求，降低Master工作负载，且客户端能缓存更多数据的元数据信息。
2. 客户端在chunkserver中对一个chunk进行多个操作，延长TCP连接时间，减少网络负载。
3. 减少Master的元数据量，可将全部元数据放在内存中。

三、Metadata元数据

Master以HeartBeat信息来监控chunkserver的状态。 其中，chunk的副本位置信息，采取启动master时轮chunkserver，然后定期轮询来获取chunk的副本位置信息，而不是直接在master中持久化该信息。 这样简化了Master与chunkserver之间的的数据同步问题。

GFS 是强同步，即只有当操作日志被写入到本地磁盘以及远程机器、备用master磁盘后，才会响应客户端的操作请求。同时master会收集多个日志后，批量写入磁盘。

checkpoint: 在灾难恢复时通过回放操作日志来将系统恢复到最近状态。为了减低回放时间，使用checkpoint方式来使得回放的日志尽量短，即在恢复时只需要回放checkpoint之后的日志文件即可。之前的旧日志文件及checkpoint文件可定期删除。最新一个标记点之前的所有日志记录的操作都已经持久化保存了，而该点之后的操作保存在内存中。当内存中数据丢失时，只需回放该点之后的部分日志即可恢复。而checkpoint文件就是记录了这些点的位置等信息。

四、系统交互

Master以带超时的租约来授权主chunkserver来执行具体的操作任务，从而减小Master的管理负担。Master与chunkserver之间通过定期的HeartBeat信息来交互信息：master获取chunkserver'的状态信息，chunkserver申请延长租约时间等。

客户端写数据的流程为：

1、客户机向主服务器询问哪一个块服务器保存了当前的租约，以及其他副本的位置。如果没有一个块服务器有租约，主服务器就选择一个副本给它一个租约。

2、主服务器回复主块的标识符以及其他副本的位置。客户机为了后续的操作缓存这个数据。只有主块不可用，或者主块回复说它已经不在拥有租约的时候，客户机才需要重新跟主服务器联络。

3、客户机把数据推送到所有的副本上。客户机可以用任意的顺序推送。每个块服务器会把这些数据保存在它的内部LRU缓冲内，直到数据被使用或者过期。通过把数据流和控制流分离，我们可以基于网络负载状况对昂贵的数据流进行规划，以提高性能，而不用去管哪个块服务器是主块。

4、所有的副本都被确认已经得到数据后，客户机发送写请求到主块。这个请求标识了早前推送到所有副本的数据。主块为收到的所有操作分配连续的序列号，这些可能来自不同的客户机。它依照序列号的顺序把这些操作应用到它自己的本地状态中。

5、主块把写请求传递到所有的二级副本。每个二级副本依照主块分配的序列号的顺序应用这些操作。

6、所有二级副本回复主块说明他们已经完成操作。

7、主块回复客户机。任何副本产生的错误都会报告给客户机。错误的情况下，主块和一些二级副本可能成功的写入了数据。客户端请求被确认为失败，已经修改的区域保持不一致的状态。客户端代码通过重复失败的操作来处理这样的错误。

五、负载均衡

GFS中副本的分布策略需要考虑多种因素，如网络拓扑、机架分布、磁盘利用率等。为了提高系统的可用性，GFS会避免将同一个chunk的所有副本都存放在同一个机架的情况。

当Master创建了一个chunk，它会根据如下因素来选择chunk副本的初始位置：1）新副本所在的ChunkServer的磁盘利用率低于平均水平；2）限制每个Chunk-Server“最近”创建的数量；3）每个chunk的所有副本不能在同一个机架。第二点容易忽略但却很重要，因为创建完chunk以后通常需要马上写入数据，如果不限制“最近”创建的数量，当一台空的ChunkServer上线时，由于磁盘利用率低，可能导致大量的chunk瞬间迁移到这台机器从而将它压垮。

当chunk的副本数量小于一定的数量后，Master会尝试重新复制一个chunk副本。可能的原因包括ChunkServer宕机或者ChunkServer报告自己的副本损坏，或者ChunkServer的某个磁盘故障，或者用户动态增加了chunk的副本数，等等。每个chunk复制任务都有一个优先级，按照优先级从高到低在Master排队等待执行。例如，只有一个副本的chunk需要优先复制。另外，GFS会提高所有阻塞客户端操作的chunk复制任务的优先级，例如客户端正在往一个只有一个副本的chunk追加数据，如果限制至少需要追加成功两个副本，那么这个chunk复制任务会阻塞客户端写操作，需要提高优先级。

最后，Master会定期扫描当前副本的分布情况，如果发现磁盘使用量或者机器负载不均衡，将执行重新负载均衡操作。

在大数据的时代中，信息量呈指数增加，谷歌的三篇论文为大数据时代的方向给出了答案，分布式的处理方法将数据的处理工作分散化，既降低了成本又提高了容错率。谷歌通过这一系列的创新、改进、优化，使得大数据处理变得更加简单可靠，让大数据处理的发展进入了一个新的台阶，也使得依靠大数据处理的产业也进入了人一个新的发展阶段。