HDFS

出自开放百科 - 灰狐

Hadoop Distributed File System is designed to reliably store very large files across machines in a large cluster. It is inspired by the Google File System.

目录

Scale Requirements

- Number of nodes – 10 thousand.
- Total data size – 10 PB.

Assuming 10,000 nodes capable of storing 1TB each. This is an order of magnitude estimate.
support

- Number of files – 100 million.

o If DFS data si
has e

o
32TB to 10PB under the assumption the average file size remains the same gives us an estimate of 1

- Number of concurrent clients – 100 thousand.

If on a 1
current m

- Acceptable level of data loss – 1 hour.

Any data created or updated in DFS 1 hour ago or before is guaranteed to be recoverable in case of system failures.

- Acceptable downtime level – 2 hours.

DFS failure requires manual system recovery.
again not later than 2 hours after the recovery start.

User Interface

- Command for HDFS User:

```
hadoop dfs -mkdir /foodir
hadoop dfs -cat /foodir/myfile.txt
hadoop dfs -rm /foodir myfile.txt
```

- Command for HDFS Administrator

```
hadoop dfsadmin -report
hadoop dfsadmin -decommission datanodename
```

- Web Interface

```
http://host:port/dfshealth.jsp
```

hadoop 分布式文件系统：体系和设计

1.介绍

hadoop文件系统（HDFS）是一个运行在普通的硬件之上的分布式文件系统，它和现有的分布式文件系统有着很多的相似性，然而和其他的分布式文件系统的区别也是很明显的，HDFS是高容错性的，可以部署在低成本的硬件之上，HDFS提供高吞吐量地对应用程序数据访问，它适合大数据集的应用程序，HDFS放开一些POSIX的需求去实现流式地访问文件数据，HDFS开始是为开源的apache项目nutch的基础结构而创建，HDFS是hadoop项目的一部分，而hadoop又是lucene的一部分。

2.假定和目标

硬件故障

硬件的故障时很正常的，而不是异常。整个HDFS系统将由数百或数千个存储着文件数据片断的服务器组成。实际上它里面有非常巨大的组成部分，每一个组成部分都会频繁地出现故障，这就意味着HDFS里的一些组成部分是总是失效的，因此，故障的检测和自动快速恢复是HDFS一个很核心的结构目标。

流式的数据访问

运行在HDFS之上的应用程序必须流式地访问它们的数据集，它不是典型的运行在常规的文件系统之上的常规程序。HDFS是设计成适合批量处理的，而不是用户交互式的。重点是在数据吞吐量，而不是数据访问的反应时间，POSIX强制的很多硬性需求对很多应用不是必须的，去掉POSIX的很多关键地方的语义以获得更好的数据吞吐率。 大数据集

运行在HDFS之上的程序有很大量的数据集。这意味着典型的HDFS文件是GB到TB的大小，所以，HDFS是很好地支持大文件。它应该提供很高的聚合数据带宽，应该一个集群中支持数百个节点，还应该支持一个集群中千万的文件。

简单一致性模型

大部分的HDFS程序对文件操作需要的是一次写入，多次读取的。一个文件一旦创建、写入、关闭之后就不需要修改了。这个假定简单化了数据一致的问题和高吞吐量的数据访问。Map-Reduce程序或者网络爬虫程序都是非常完美地适合这个模型。有一个计划在将来实现文件的附加写入。

移动计算比移动数据更经济

在靠近要被计算的数据所存储的位置来进行计算是最理想的状态，尤其是在数据集特别巨大的时候。这样消除了网络的拥堵，提高了系统的整体吞吐量。这个假定就是将计算离数据更近比将文件移动到程序运行的位置更好。HDFS提供了接口，来让程序将自己移动到离数据存储的位置更近。

轻便的访问异构的软硬件平台

HDFS应该设计成这样的一种方式，就是简单轻便地从一个平台到另外一个平台，这将推动需要大数据集的应用更广泛地采用HDFS作为平台。

名字节点和数据节点

HDFS是一个主从结构的体系，一个HDFS集群是由一个名字节点，它是一个管理文件的命名空间和调节客户端访问文件的主服务器，当然还有的数据节点，一个节点一个，它来管理存储。HDFS暴露文件命名空间和允许用户数据存储成文件。

内部机制是将一个文件分割成一个或多个的块，这些块存储在一组数据节点中。名字节点操作文件

命名空间的文件或目录操作，如打开，关闭，重命名，等等。它同时确定块与数据节点的映射。数据节点来负责来自文件系统客户的读写请求。

数据节点同时还要执行块的创建，删除，和来自名字节点的块复制指示。

名字节点和数据节点都是软件运行在普通的机器之上，机器典型的都是linux，HDFS是用java来写的，任何支持java的机器都可以运行名字节点或数据节点，利用java语言的超轻便型，很容易将HDFS部署到大范围的机器上。典型的部署时将有一个专门的机器来运行名字节点软件，机群中的其他机器运行一个数据节点实例。体系结构排斥在一个机器上运行多个数据节点的实例，但是实际的部署不会有这种情况。

集群中只有一个名字节点极大地简单化了系统的体系。名字节点是仲裁者和所有HDFS的元数据的仓库。系统设计成用户的实际数据不经过名字节点。

文件命名空间

HDFS支持传统的继承是的文件组织。一个用户或一个程序可以创建目录，存储文件到很多目录之中。文件系统的名字空间层次和其他的文件系统相似。可以创建、移动文件，将文件从一个目录移动到另外一个，或重命名。HDFS现在还没有实现用户的配额和访问控制。HDFS还不支持硬链接和软链接。然而，HDFS结构不排斥在将来实现这些功能。

名字节点维护文件的系统的命名空间，任何文件命名空间的改变和或属性都被名字节点记录。应用程序可以指定文件的复制数，文件的拷贝被称作文件的复制因子，这些信息有名字空间来负责存储。

数据复制

HDFS设计成可靠地在集群中的大量机器之间存储非常大量的文件，它以块序列的形式存储每一个文件。文件的除了最后一个块的其他块都是相同的大小。属于文件的块为了故障容错而被复制。块的大小和复制数可以为每个文件配置。HDFS中的文件都是严格地任何时候只有一个写操作。程序可以特别地为某个文件指定。复制数，文件的复制数可以在文件的创建的时候指定或者以后改变。名字节点来做所有的块复制，它周期性地接受来自集群中数据节点的心跳和块报告。一个心跳的收条表示这个数据节点是健康的，是渴望服务数据的。一个块报告包括该数据节点上的所有的块列表。

复制块的放置位置。第一个块的阶段

复制块的放置位置的选择严重影响HDFS的可靠性和性能。这个特征是HDFS和其他的分布式文件系统的区别。这个特征需要很多的调节和经验。根据机架的复制布局目的就是提高数据的可靠性，可用性和网络带宽的利用。

当前的这方面的实现方式是在这个方向上的第一步。短期的目标实现是这个方式要在生产环境下去验证，以得到它的行为和实现一个为将来的测试和研究更佳的方式的基础。

HDFS运行在跨越很多机架的集群机器之上。两个不同机架上的节点通信是通过交换机的，在大多数情况下，两个在相同机架上的节点之间的网络带宽是优于在不同的机架之上的两个机器。

在开始的时候，每一个数据节点自检它所属的机架，然后在向名字节点注册的时候告知它的机架id。HDFS提供接口以便很容易地挂载检测机架标示的模块。一个简单但不是最优的方式就是将复制跨越不同的机架，这样以保证在这个机架出现故障而不丢失数据，还能在读数据的时候充分利用不同机架的带宽。这个方式均匀地将复制分散在集群中以简单化地实现了组件实效的负载均衡，然

而，这个方式增加了写的成本，因为写的时候需要传输文件块到很多的机架。

在大多数复制数为3的普通的情况下，HDFS放置方式是将第一个放在本地节点，将第二个复制放到本地机架上的另外一个节点而将第三个复制放到不同机架上的节点。这种方式减少了机架内的写流量，提高了写的性能。机架失效的机会远小于机器实效的。这种方式没有影响数据的可靠性和可用性的保证。但是它减少了读操作的网络聚合带宽，因为文件块存在

两个不同的机架， 而不是三个。文件的复制不是均匀地分布在机架当中。1/3在同一个节点上，第二个1/3复制在同一个机架上，另外1/3是均匀地分布在其他的机架上。这种方式提高了写性能，而没有影响数据的可靠性和读性能。

上面的实现方式正在进行中。

复制的选择

HDFS尝试满足一个读操作来自离它最近的复制。假如在读节点的同一个机架上就有这个复制，就直接读这个，如果HDFS集群是跨越多个数据中心，那么本地数据中心的复制是优先于远程的复制。

安全模式

在启动的时候，名字节点进入一个特殊的状态叫做安全模式。安全模式是不发生文件块的复制的。名字节点接受来自数据节点的心跳和块报告。一个块报告包括的是数据节点向名字节点报告数据块的列表。

每一个块有一个特定的最小复制数。当名字节点检查这个块已经大于最小的复制数就被认为是安全地复制了，当达到配置的块安全复制比例时（+30s）名字节点就退出安全模式。它将检测数据块的列表，将小于特定复制数的块复制到其他的数据节点。

文件系统的元数据的持久化

HDFS的命名空间是由名字节点来存储的。名字节点用事务日志叫做EditLog来持久化每一个对文件系统的元数据的改变，例如，在HDFS中创建一个新的文件，名字节点将会插入一吊记录到EditLog来标示这个改变。类似地，改变文件的复制因子也会向EditLog中插入一条记录。名字节点在本地文件系统中用一个文件来存储这个EditLog。完整的文件系统命名空间、文件块的映射和文件系统的配置都存在一个叫FsImage的文件中，FsImage也是名字节点的本地文件系统中。

名字节点在内存中有一个完整的文件系统命名空间和文件块的映射镜像。这个元数据时设计成紧凑的，这样4G的内存的名字节点就能很轻松地处理非常大文件数和目录，当名字节点启动，它将从磁盘中读取FsImage和EditLog应用EditLog中的所有的事务到内存中的FsImage表示方法，然后将新的元数据刷新到本地磁盘的新的FsImage中这样可以截去旧的EditLog，因为事务已经被处理并已经持久化的FsImage中。这个过程叫做检查点。在现在的实现检查点在名字节点启动的时候发生。支持周期性的检查点正在进行中。

数据节点存储HDFS数据到本地的文件系统中。数据节点没有关于HDFS文件的信息。它以单独的文件存储每一个HDFS的块到本地文件系统中。数据节点不产生所有的文件到同一个目录中，而是它用启发式的检测最优的每一个目录的文件数。它在适当的时候创建子目录。在本地文件的同一个目录下创建所有的文件不是最优的，因为本地文件系统可能单个目录里有数目巨大的文件效率较差。当数据节点启动的时候，它将扫描它的本地文件系统，根据本地的文件产生一个所有HDFS数据块的列表并报告给名字节点，这个报告称作块报告。

通信协议

所有的通信协议都是在TCP/IP协议之上的。一个客户端和明确的配置端口的名字节点建立连接之后，它和名字节点的协议是ClientProtocal。数据节点和名字节点之间用DatanodeProtocal。详细的这些协议将在后面解释。

RPC抽象地包装了ClientProtocol和DataNodeProtocol。根据设计，名字节点不会发起一个RPC，它只是对数据节点和客户端发起的RPC做出反馈。

鲁莽性

HDFS的主要目标就是在存在故障的情况下可靠地存储数据。三个普通的故障是名字节点实效，数据节点实效，和网络断开

磁盘故障，心跳和重新复制

一个数据节点周期性发送一个心跳信息到名字节点。网络断开会造成一个数据节点子集和名字节点失去联系。名字节点发现这种情况是根据有没有了心跳信息。名字节点标记这些数据节点是死掉了，就不再将新的IO请求转发到这些数据节点上。而这些数据节点上的数据将对HDFS不再可用。这将导致一些块的复制因子降低到指定的值。

名字节点检查所有的需要复制的块，并开始复制他们到其他的数据节点上。重新复制会因为很多原因而必须 例如：数据节点变得比可用，被破坏了的复制，数据节点上的磁盘损坏或增加了文件的复制因子。

集群的重新均衡

HDFS体系结构是兼容数据的重新平衡方案的。在数据节点的可用空间降低到一个极限时数据可能自动的从一个数据节点移动到另外一个，而且一个突然地对一个特殊的文件发生高请求时也会引发额外的复制，将集群中的其他数据重新均衡。这种类型的重新均衡方案还没有实现。

数据正确性

从数据节点上取一个文件块有可能出现损坏的情况，这种情况可能会发生是因为存储设备，差劲的网络，软件的缺陷。HDFS客户端实现了校验去检查HDFS的文件内容。当一个客户端创建一个HDFS文件，它为每一个文件快计算一个校验码并存储校验码在同一个HDFS名字空间中的一个单独的隐藏文件中。当客户端找回这个文件内容时，它再根据这个校验码来验证从数据节点接受到的数据。如果不对，客户端可以从另外一个有该块复制的数据节点取这个块。

元数据磁盘实效

FsImage和Editlog是HDFS的中心数据结构。这些文件的损坏会导致整个集群的不工作。应为这个原因，名字节点可以配置成多个FsImage和EditLog的拷贝。任何的不管对FsImage和EditLog的更新都会同步地更新每一个拷贝。

这个同步的更新多个EditLog可能降低了名字节点的可支持名字空间的每秒交易数。但是这个降低是可接受的，因为HDFS程序都是自然地对数据要求强烈，而不是对元数据的要求强烈。名字节点重新启动时，选择最新的一致的FsImage和EditLog。

名字节点队以HDFS集群是单点实效的。假如名字节点实效，手工的干涉是必要的，当前，自动的重启和切换到另外的名字节点目前还不支持。

快照

快照支持在一个特定时间存储一个数据拷贝，快照的一个用途可以将实效的集群回滚到之前的一个正常时间点上。HDFS目前还不支持快照，但是将被将来的版本支持。

数据组织

数据块

HDFS是设计成支持大文件数的。程序也是和HDFS一样地处理大数据集。这些程序写数据仅一次，读数据一次或多次，需要一个比较好的流读取速度。HDFS支持文件的写一次读多次的。HDFS典型的块大小是64M，一个HDFS文件可以最多被切分成128MB个块，每一个块分布在不同的数据节点上。

分段运输

当一个客户端请求创建一个文件的时候，并不是立即请求名字节点，事实是，HDFS客户端在本地的文件中缓存文件数据，应用程序的写操作明显地转移到这个临时的本地文件。当本地文件堆积到大于HDFS块大小的时候，客户端联系名字节点。名字节点插入文件名到文件系统层次当中，然后构造一个数据块。名字节点回应客户端的请求包括数据节点（可能多个）的标识和目标数据块，客户端再将本地的临时文件刷新指定的数据节点数据块中。

当文件关闭，还有一些没有刷新的本地临时文件被传递到数据节点。客户端就通知名字节点，这个文件已经关闭。这个时间和，名字节点提交文件的创建操作到持久化存储。假如名字节点在文件关闭之前死掉，文件就丢掉了。

上面的方式在仔细地考虑运行在HDFS之上的目标程序之后被采用。应用程序需要流式地写文件。如果客户端直接写到远程文件系统，而没有本地的缓冲对网速和网络吞吐量产生相当的影响。这种方式也不是没有前科，早期的分布是文件系统，例如AFS也用客户端的缓冲来提高性能，POSIX需求也不拘束高性能的数据上传的实现。

流水线操作

当客户端写数据到HDFS文件中，像上面所讲数据首先写道本地文件中，假设HDFS的复制因子是3，当本地文件堆积到一块大小的数据，客户端从名字节点获得一个数据节点的列表。这个列表描述一些数据节点将接管块的复制。客户端刷新数据块到第一个数据节点。第一个数据节点开始接收数据到一个很小的位置（4kb），写每一个部分到本地的库中，而且传输每一个部分到列表中的第二个数据节点，这样就轮到第二个数据节点，第二个数据节点如同第一个数据节点给第三个数据节点，第三个数据节点直接写到本地的库中。一个数据节点可以接受来自前一个的节点的数据，同时还可以将数据流水式传递给下一个节点，所以，数据时流水式地从一个数据节点传递到下一个。

可访问

HDFS可以由应用程序多种方式存取，自然地，HDFS提供为程序提供java api，为c语言包装的java api也是可以的，还有一个HTTP浏览HDFS中的文件，通过WebDAV协议访问HDFS内容库正在进行。

DFSS

HDFS允许用户数据由文件和文件夹式的管理，它提供一个接口叫DFSShell，让用户和HDFS中的数据交互

命令集的语法跟其他的shells（bash,csh）相似

创建目录foodir : hadoop dfs -mkdir /foodir

查看文件 /foodir/myfile.txt : hadoop dfs -cat /foodir/myfile.txt

删除文件/foodir/myfile.txt : hadoop dfs -rm /foodir myfile.txt

DFS

DFSAdmin命令集是用于管理dfs集群的，这些命令只由HDFS管理员使用

将集群设置成安全模式 : bin/hadoop dfsadmin -safemode enter

产生一个数据节点的列表 : bin/hadoop dfsadmin -report

去掉一个数据节点: bin/hadoop dfsadmin -decommission datanodename

浏览接口

典型的HDFS安装配置了一个web 服务去暴露HDFS的命名空间，允许web浏览器去浏览HDFS的命名空间和查看

HDFS文件的内容

空间回收

文件删除和恢复删除

当一个文件被用户或程序删除，它并不是立即从HDFS中删除，而是HDFS将它重新命名到/trash目录下的文件，这个文件只要还在/trash目录下保留就可以重新快速恢复。当这个文件在/trach里呆够配置的时间，名字节点就将它从名字空间中删除，这个删除将导致这个文件的文件块都被释放。这个时间间隔可以被感知，从用户删除文件到HDFS的空闲空间的增加。

用户可以在删除一个文件之后，它还在/trash目录下的情况下，恢复删除一个文件，如果一个用户希望恢复删除他已经删除的文件，可以浏览/trash目录，重新获得这个文件。/trash目录之保存最新版本的删除文件。/trash目录也像其他目录一样，只有一个特殊的功能，就是HDFS应用一个特定的规则，自动地删除这个目录里的文件，当前默认的规则是删除在此目录呆够6小时的文件，将来这个规则将由一个接口来配置。

减少复制因子

当文件的复制因子减少了，名字节点选择过度的复制去删除掉，下一次的心跳的时候传递这个信息给数据节点。数据节点移除相应的块，相应的空闲空间将显示在集群中，这一点要注意的就是这个可能会有段时间过程在完成setReplication和显示集群的空闲空间。