ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.7, No.22, August 2011.

E-mail: kfyj@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

Hadoop 集群性能优化技术研究

辛大欣,刘飞

(西安工业大学,陕西 西安 710032)

摘要:Hadoop 技术已经在互联网领域得到广泛的应用,同时也得到了学术界的普遍关注。该文介绍了 Hadoop 作为基础数据处理平台仍然存在的问题,阐明了 Hadoop 性能优化技术研究的必然性,并介绍了当前 Hadoop 优化的三个主要思路:从应用程序角度进行优化、对 Hadoop 系统参数进行优化和对 Hadoop 作业调度算法进行优化。Hadoop 集群优化对于提高系统性能和执行效率具有重大的意义。

关键词:Hadoop 集群;性能优化;配置参数;作业调度

中图分类号:TP14 **文献标识码**:A **文章编号**:1009-3044(2011)22-5484-03

Research of Hadoop Performance Tuning Technology

XIN Da-xin, LIU Fei

(Xi'an Technological University, Xi'an 710032, China)

Abstract: Hadoop technology had been wildly used and research around the internet and academics. The article introduce the reminded problems of Hadoop data processing platform and Illustra Configuration parameters imization the hadoop performace to increase the system performace and efficiency.

Key words: Hadoop cluster; performance optimization; configuration parameters; job scheduler

hadoop 是隶属于 Apache 软件基金会(Apache Software Foundation)的开源 JAVA 项目,它是一个分布式的具有可靠性和可扩展性的存储与计算平台。历经多年发展,Hadoop 社区不断扩大,而 Hadoop 本身也已经演变成为一个拥有众多子项目的项目集合,其中最核心的部分是用于分布式存储 HDFS(Hadoop Distributed File System)文件系统和用于分布式计算的 MapReduce 计算架构,除此以外还有 HBase、Hive、Pig 和 ZooKeeper 等。

1 Hadoop 数据处理平台存在的问题

随着企业要处理的数据量越来越大,MapReduce 思想越来越受到重视。Hadoop 是 MapReduce 的一个开源实现,由于其良好的扩展性和容错性,已得到越来越广泛的应用。Hadoop 作为一个基础数据处理平台,虽然其应用价值已得到大家认可,但仍存在很多问题,主要表现在以下几个方面:

1)Namenode/jobtracker 单点故障。 Hadoop 采用的是 master/slaves 架构,该架构管理起来比较简单,但存在致命的单点故障和空间容量不足等缺点,这已经严重影响了 Hadoop 的可扩展性。

2)HDFS 小文件问题。在 HDFS 中,任何 block,文件或者目录在内存中均以对象的形式存储,每个对象约占 150byte,如果有 1000 0000 个小文件,每个文件占用一个 block,则 namenode 需要 2G 空间。如果存储 1 亿个文件,则 namenode 需要 20G 空间。这样 namenode 内存容量严重制约了集群的扩展。

3)jobtracker 同时进行监控和调度,负载过大。为了解决该问题,yahoo 已经开始着手设计下一代 Hadoop MapReduce。他们的主要思路是将监控和调度分离, 独立出一个专门的组件进行监控, 而 jobtracker 只负责总体调度, 至于局部调度, 交给作业所在的 client。

4)数据处理性能。很多实验表明,其处理性能有很大的提升空间。Hadoop 类似于数据库,可能需要专门的优化工程师根据实际的应用需要对 Hadoop 进行调优,有人称之为"Hadoop Performance Optimization" (HPO)。

由于 Hadoop 平台已经成为了大多数公司的分布式数据处理平台,随着数据规模的越来越大,对集群的压力也越来越大,集群的每个节点负担自然就会加重,而且集群内部的网络带宽有限,数据交换吞吐量也在面临考验,因此也引发了人们对大规模数据处理进行优化的思考。

2 从应用程序角度进行优化

由于 mapreduce 是迭代逐行解析数据文件的,怎样在迭代的情况下,编写高效率的应用程序,是一种优化思路。可以从以下 7 个方面来提高 MapReduce 的性能:避免不必要的 reduce 任务、外部文件引入、为 Job 添加一个 Combiner、重用 Writable 类型、使用 StringBuffer 而不是 String 和调试程序跟踪程序的瓶颈。

收稿日期:2011-06-21

作者简介:辛大欣(1966-),男,陕西西安人,西安工业大学副教授,硕士,主要研究方向为计算机体系结构;刘飞(1987-),男,湖北荆州人,西安工业大学硕士生在读,就读专业是计算机软件与理论,主要研究方向为基于云计算的分布式存储系统的研究与应用。

3 hadoop 系统参数优化研究

当前 hadoop 系统有 190 多个配置参数,怎样调整这些参数,使 hadoop 作业运行尽可能的快,也是一种优化思路。hadoop 系统参数设置优化主要包括三个方面:Linux 文件系统参数调整,Hadoop 通用参数调整和 Hadoop 作业调优参数。

3.1 Linux 文件系统参数调整

noatime 和 nodiratime 属性,文件挂载时设置这两个属性可以明显提高性能,默认情况下,linux et2/et3 文件系统在文件被访问,创建和修改时会记录下文件的时间戳。如果系统运行时要访问大量文件,关闭这些操作可提升文件系统的性能。Readahead buffer 参数用以调整 linux 文件系统中预读缓冲区的大小,可以明显提高顺序读文件的性能。默认 buffer 为 256sectors,可以增大为 1024 或 2408sectors(注意不是越大越好)。避免在 TaskTracker 和 DataNode 节点上执行 RADI 和 LVM 的操作,这样会降低性能。

3.2 hadoop 通用参数调整

dfs.namnode.handler.count 或 mapred.job.tracker.handler.count 是 namenode 和 jobtracker 中用于处理 RPC 的线程数, 默认是 10,对于较大集群,可适当调大比如 64。

dfs.datanode.handler.count 是 datanode 上用于处理 RPC 的线程数,默认是 3,对于较大集群可适当调大比如 8。TaskTracker.http.threads 是 HTTP server 上的线程数,运行在每个 TaskTracker 上,用于处理 map task 输出,大集群可以设置为 40~50。

dfs.block.size,HDFS 中的数据 block 大小,默认是 64M,对于较大集群,可以设置为 128 或 264M。mapred.local.dir 和 dfs.data.dir 这两个参数配置的值应当是分布在各个磁盘上的目录,这样可以充分利用 IO 读写能力。

3.3 hadoop 作业调优参数

3.3.1 map task 相关配置

io.sort.mb,当 map task 开始运算并产生中间数据时,其产生的中间结果并非直接写入磁盘,而是利用内存 buffer 来进行已经产生的部分结果的缓存。当 buffer 达到一定阈值,会启动一个后台线程对 buffer 的内容进行排序然后写入本地磁盘(一个 spill 文件)。默认是 100M 对于大集群可设置为 200M。

io.sort.spill.percent 这个值就是上述 buffer 的阈值,默认是 80%,当 buffer 中的数据达到这个阈值,后台线程会起来对 buffer 中已有的数据进行排序,然后写入磁盘。

io.sort.factor 当一 map task 执行完之后, 本地磁盘上 (mapred.local.dir) 有若干个 spill 文件,map task 最后做的一件事就是执行 merge sort,把这些 spill 文件合成一个文件(partition)。执行 merge sort 的时候,每次同时打开多少个 spill 文件由该参数决定。打开的文件越多,不一定 merge sort 就越快,所以要根据数据情况适当的调整。

mapred.compress.map.output,是否对中间结果和最终结果进行压缩,如果是,指定压缩方式 ,推荐使用 LZO 压缩。Inter 内部测试表明,相比未压缩,使用 LZO 的作业运行时间减少 60%。

3.3.2 reduce task 相关配置

mapred.reduce.parallel.copies 表示 Reduce shuffle 阶段 copier 线程数。reduce 分为三个阶段,分别是 copy->sort->reduce。copy 即 shuffle, 当 job 已完成 5%的 map tasks 数量之后开始启动 reduce, 从不同的已经完成的 map 上去下载属于自己这个 reduce 部分数据,由于 map 数量很多,对于一个 reduce 来说,可以并行的从多个 map 下载。默认值 5,对于大集群可调整为 16–25。

mapred.job.shuffle.input.buffer.percent(default 0.7),在 shuffle 阶段下载来的 map 数据,并不是立刻写入磁盘,而是先缓存在内存中,这个百分比是 shuffle 在 reduce 内存中的数据最多使用量为:0.7 × maxHeap of reduce task。

这种基于参数的调优比较“静态”,因为一套参数配置只对于一类作业是最优的。通过对这些参数的研究,可以寻找参数配置与不同作业特征之间的关联。

4 hadoop 作业调度算法优化研究

基于集群硬件信息和节点数量的 hadoop 配置能够很好的提高 hadoop 集群性能已被大量实验所验证, 但是这种方法只是静态地对集群性能做优化,在 Job 运行时无法动态地修改配置文件并使其加载生效,基于 hadoop 作业调度算法的优化能很好的解决这个问题。

在 hadoop 系统中,作业调度组件非常重要,它的作用是将系统中空闲的资源按一定策略分配给作业。在 Hadoop 中,调度器是一个可插拔的模块,用户可以根据自己的实际应用要求设计调度器。

1)默认的调度器 FIFO

最早的 Hadoop Map/Reduce 计算架构中,JobTracker 在进行作业调度时使用的 FIFO(First In First Out)算法,所有用户的作业都被提交到一个队列中,然后由 JobTracker 先按照作业的优先级高低,再按照作业提交时间的先后顺序选择将被执行的作业。其优点是调度算法简单明了,JobTracker 工作负担轻。缺点是忽略了不同作业需求差异。例如如果类似对海量数据进行统计分析的作业长期占据计算资源,那么对提交的交互型作业有可能迟迟得不到处理,从而影响用户的体验。

2)计算能力调度器(Capacity Scheduler)

支持多个队列,每个队列可配置一定的资源量,每个队列采用 FIFO 调度策略,为了防止同一个用户的作业独占队列中的资源,该调度器会对同一用户提交的作业所占资源量进行限定。调度时,首先按以下策略选择一个合适队列:计算每个队列中正在运行的任务数与其应该分得的计算资源之间的比值,选择一个该比值最小的队列;然后按以下策略选择该队列中一个作业:按照作业优先级和提交时间顺序选择,同时考虑用户资源量限制和内存限制。

Capacity Scheduler 能有效的对 hadoop 集群的内存资源进行管理,以支持内存密集型应用。作业对内存资源需求高时,调度算法将把该作业的相关任务分配到内存资源充足的 task tracker 上。在作业选择过程中,Capacity Scheduler 会检查空闲 task tracker 上的

内存资源是否满足作业要求。Task tracker 上的空闲资源数量值可以通过 task tracker 的内存资源总量减去当前已经使用的内存数量得到,而后者包含在 task tracker 向 job tracker 发送的周期性心跳信息中。目前,基于内存的调度只能在 linux 平台下起作用,关于内存调度的相关参数可以通过配置文件来设置。

3)公平份额调度算法(Fair Scheduler)

Fair Scheduler 是由 Facebook 公司提出的,为了解决 Facebook 要处理生产型作业(数据分析、HIVE)、大型批处理作业(数据挖掘、机器学习)、小型交互型作业(HIVE 查询)的问题。同时满足不同用户提交的作业在计算时间、存储空间、数据流量和响应时间上都有不同需求的情况下,使 hadoop mapreduce 框架能够应对多种类型作业并行执行,使得用户具有良好的体验,所以 Facebook 提出了该算法。

Fair Scheduler 的设计思想是,尽可能保证所有的作业都能够获得等量的资源份额。系统中只有一个作业执行时,它将独占集群所有资源。有其他作业被提交时就会有 TaskTracker 被释放并分配给新提交的作业,以保证所有的作业都能够获得大体相同的计算资源。

这三种调度算法存在一定的缺陷,目前 hadoop 集群作业调度算法已近成为是研究的重点之一,例如适用于异构集群的调度器 LATE 和适用于实时作业的调度器 DeadLine Scheduler 和 Constraint-based Scheduler 都提出了新的处理机制。

5 结论

总体来说,对于 Hadoop 平台,现在主要有三种优化思路,分别为:从应用程序角度角度进行优化,从参数配置角度进行优化,从作业调度算法角度进行优化。对于第一种思路,需要根据具体应用需求而定,同时也需要在长期实践中积累和总结;对于第二种思路,大部分采用的方法是根据自己集群硬件和具体应用调整参数,找到一个最优的。对于第三种思路,难度较大,但效果往往非常明显。

参考文献:

[1] Zaharia M,Borthakur D,Sarma J S,et,al.Job scheduling for multi-user mapreduce clusters [C].EECS Department,University of California, Berkeley,Tech.Rep,Apr 2009.

[2] Tian C,Zhou H,He Y.A dynamic mapreduce scheduler for heterogeneous workloads [C]//Proceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing,ser.GCC'09. Washington, DC, USA:IEEE Computer Society,2009:218-224.

[3] http://developer.yahoo.com/blogs/hadoop/posts/2011/02/mapreduce-nextgen/.

[4] Xuhui Liu,Jizhong Han.Implementing WebGIS on Hadoop:A case study of improving small file I/O performance on HDFS[Z].CLUSTER, 2009:1-8.

(上接第 5483 页)

```
            controlModel.Commands.Add(new WheelZoom());
        }
```

4)地图视频监视事件的实现

将设备信息写入到地图文件中,自定义信息查询组件,当鼠标单击信息工具时获取图元的各种信息并显示。信息查询组件的开发技术原理与鼠标中键缩放功能的组件开发技术原理大体一致,这里不再赘述。

5 总结

本文介绍了利用 MapXtreme 技术和.NET 平台开发信息化 WebGIS 系统的流程,实现了地图鹰眼、鼠标中键缩放、实时监控、设备信息监控等功能,详细阐述了实现过程中的关键技术,所开发的系统为管理和决策者提供了实时信息,方便其快速准确地做出决策。

参考文献:

[1] 王桥,张宏,李旭文,等.水资源地理信息系统[M].北京:科学出版社,2004.

[2] 孟令奎,史文中,张鹏林,等.网络地理信息系统原理与技术[M].北京:科学出版社,2005.

[3] 张建新,赵黎民.基于 3S 技术的县级土地利用数据库建设[J].国土资源科技管理,2008,25(4):67-70.

[4] StephenWalther.ASP.NET 2.0 揭秘[M].北京:人民邮电出版社,2007.

[5] Karli.C# 入门经典[M].3 版.北京:清华大学出版社,2006.

[6] 哈特.ASP.NET 2.0 经典教程[M].北京:人民邮电出版社,2007.

[7] Christian Nagel,Bill Evjen,Jay Glynn.C# 高级编程[M].7 版.李铭,译.北京:清华大学出版社,2010.