文章编号:1672 - 058X(2012)11 - 0036 - 04

云计算环境下关联规则数据挖掘算法研究

马 洁
(重庆第二师范学院,重庆 400067)

摘 要:在海量数据的关联规则数据挖掘中,采用并行计算是非常必要的;针对当前的关联规则算法,运用并行算法的思想,结合云计算环境下的 Hadoop 架构,提出了 Hadoop 下的并行关联规则算法的设计,最后实验表明,该算法能处理节点失效,并且能实现节点负载均衡。

关键词:关联规则数据挖掘;云计算;Hadoop
中图分类号:TP372 **文献标志码**:A

挖掘关联规则(知识)就是从给定的数据集中搜索数据项(Items)之间所存在的有价值联系,其核心是通过统计数据项获得频繁项集,具体规则由支持度(Support)和信任度(Confidence)两个指标确定,现有的具有代表性的算法是 Apriori 算法,算法利用了一个层次顺序搜索的循环方法来完成频繁项集的挖掘工作,但是从计算平台上来讲,该算法基本是基于单点的算法。然而,在目前的互联网环境中,运用传统关联规则挖掘算法处理海量数据,使得处理的速度非常缓慢,大规模和多维问题所需的单纯计算工作在单处理机上的运行时间太长,超出了人们的忍受范围,从而使得在多处理机上采用并行的方式解决该类问题成为较为理想的办法。基于此,本文使用并行算法思想,结合云计算计算模型,实现了 Hadoop 架构下关联规则数据挖掘算法的设计,解决了节点失效、负载不均衡的问题。

1 云计算环境下的 Hadoop

1.1 云计算与 Hadoop

云计算是分布式计算技术的一种,最初由企业界开始发展,然后才进入学术界引起重视的,这与网格计算相反。云计算在海量数据的处理中有着重要的地位和发展空间,在数据处理上具有超大规模、虚拟化、高可靠性和高可扩展性等特点。从当前来看,云计算研究的关键技术包括虚拟机、安全管理、数据管理、云监测、能耗管理和计算模型等。

云计算的计算模型是研究如何针对某类应用特点提出效率更高的编程方式,目前云计算模型众多,而 Hadoop 就是其中之一。Hadoop 是一个开源框架,可编写和运行分布式应用处理大规模数据。与其他云计算模型相比,Hadoop 具有以下优点:

(1) 方便。Hadoop 运行在由一般商用机器构成的大型集群上,或者如亚马逊弹性计算云(EC2)等云计算服务之上。

(2) 健壮。Hadoop 致力于在一般商用硬件上运行,可以从容地处理大多数此类故障。

(3) 可扩展。Hadoop 通过增加集群节点,可以线性地扩展以处理更大的数据集。

收稿日期:2012 - 07 - 11;**修回日期**:2012 - 09 - 01.

作者简介:马洁(1979-),女,重庆市人,硕士生,讲师,从事计算机应用技术研究.

(4) 简单。Hadoop 允许用户快速编写出高效的并行代码。

1.2 Map/Reduce 模式

Map/Reduce 最早由 Google 公司提出,是一种有效的分布式并行计算框架,可用于大规模数据集的并行运算,为海量数据的计算分析提供一种可行方案。在 Map/Reduce 模型中,数据处理原语被称为 Mapper 和 Reducer。分解一个数据处理应用为 Mapper 和 Reducer 有时是繁琐的,但是一旦以 Map/Reduce 的形式写好了一个应用程序,仅需修改配置就可以将它扩展到集群中几百、几千,甚至几万台机器上运行。Map/Reduce 的执行流程主要包括 Map 和 reduce 两个阶段,具体为:

在 Map 阶段,Map/Reduce 框架将输人数据拆分为大量的数据片段,并将每一个数据片段分配给一个 Map 任务。每一个 Map 任务会将对其分配到的 Key-Value 对进行计算,生成一个中间结果,然后将中间结果中所有具有相同 Key 值的 Value 经过计算后传递给 Reduce 函数。

在 Reduce 阶段,每一个 Reduce 任务会将分配到的二元组 Key-Value 集合的片段作为输入。对于每一个这样的二元组都会调用一个用户定义的 Reduce 函数将 Value 值合并,形成一个较小的 Value 的集合,每次 Reduce 函数调用只产生 0 或 1 个 Value 值输出。

每个阶段的任务执行都是支持容错的,如果任一个或多个节点在计算过程中出现错误都会将任务自动重新分配到其他节点。同时运行多个 Map 和 Reduce 任务提供了很好的负载均衡并且保证了运行中失败的任务被重新运行的代价降到尽可能的小。

2 基于 Hadoop 的关联规则数据挖掘算法

2.1 算法分析

关联规则(Association Rules)挖掘就是从大量的数据中挖掘出有价值描述数据项之间相互联系的有关知识,是数据挖掘领域研究的一个重要内容。关联规则挖掘的代表算法是 Apriori 算法,在 Apriori 算法基础上,结合并行算法思想,人们提出了 CD(Count Distribution,简称 CD)算法,算法的思想是在每一个处理机上都存储全局的候选项目集和频繁项目集,每一步计算时利用 Apriori 算法计算出候选集在本地数据上的支持数,然后做一次同步,各处理机交换本地的候选项目集的支持数,使得每个处理机的候选项目集都得到全局支持数,从而得到全局频繁项目集。从其思想来看,CD 算法目标是减少通信量获得较好的任务分布性,使各处理器只对本地数据并行地进行处理,但是其缺点是内存利用有效性不足。在此依据 CD 算法的优势,结合分布式计算环境 Hadoop,提出了云计算环境下关联规则数据挖掘算法。

在 Hadoop 架构下,要求关联规则数据挖掘要具备两个方面的要求,同时也能解决 CD 算法的不足之处,即:一方面,在 Hadoop 上运行时,某一节点的计算失败,不会导致整个计算的失败;另一方面,对于一项复杂的计算任务来说,数据分块数远大于计算节点数,计算资源不会被浪费,能做到负载均衡。基于此,提出 Hadoop 下关联规则数据挖掘的算法步骤:

(1) 主进程通过 $k-1$ 项频繁项集产生 k 项候选集,并分发给每个节点;

(2) 根据 Map 进程配置数量 x,每个节点运行 x 个 Map 进程,每个 Map 进程处理一个数据分块,获取该分块的 k 项候选集的支持数。每个 Map 进程处理完一个数据分块之后,Hadoop 架构将按照需要启动下一个 Map 进程,直到处理完所有数据;

(3) Reduce 进程把 Map 进程获取的每个分块的 k 项候选集支持数合并获取全局 k 项候选集的支持数;

(4) 主进程根据 k 项候选集的支持数获取 $k+1$ 项频繁项集;

(5) 主进程决定是否进行下一步计算。跟其他并行关联规则挖掘算法相比,算法在 Hadoop 架构下不存在重复计算,能够有效节省集群中的计算资源。

2.2 算法描述

算法的基本思想是将计算任务分配到 N 个节点上,每个节点采用关联规则算法(类似 Apriori 算法),其

具体步骤:

(1) 每个计算节点利用 $k-1$ 项频繁项集产生 k 项候选集,考虑到每个节点拥有相同的 $k-1$ 项频繁项集,因此产生的 k 项候选集是相同的;

(2) 每个处理单元扫描各自的数据获得 k 项候选集的本地支持数;

(3) 每个处理单元和其他处理单元交换 k 项候选集的本地支持数获取全局的 k 项候选集支持数;

(4) 每个处理单元根据 k 项候选集的支持数获取 k 项频繁项集;

(5) 每个处理单元决定是否进行下一步计算,因为它们有相同的 k 项频繁项集,因此决定也是相同的。

在 Hadoop 分布计算环境下,进行关联规则挖掘时主要需要实现 3 个函数:main 函数、map 函数和 reduce 函数。因此基于 Hadoop 的并行关联规则挖掘算法用伪代码描述如下。

(1) 主进程 main 的伪代码描述:

```
while (k 项候选集非空){
    job = new job
    //设置处理输入数据的类,class MyInput 处理输入数据产生 key - value 对
    job. SetInputClass( class MyInput)
    //设置输出数据的关键字类型
    job. SetOutputKey( class outputkey)
    //设置输出数据的值类型
    job. SetOutputValue( class outputvalue)
    //设置进行各块数据统计的类
    job. SetMap( class MyMapClass)
    //设置合并各个分块统计的类
    job. SetReduce( class MyReduceClass)
    //运行
    job. run
    //根据 k 项频繁项集产生 k+1 项候选集
    GetNextCdt
    k = k + 1
    }
```

(2) map 的伪代码描述:

```
map ( InputKey, InputValue, OutputKey, OuputValue){
    //统计计数
    GetCount( InputKey, InputValue)
    //输出本块的候选集的计数
    OutResult( OutputKey, OutputValue)
}
```

(3) reduce 的伪代码描述:

```
reduce( InputKey, InputValue, OutputKey, OutputValue){
    //合并各个块的候选集计数
    GetAllCount( InputKey, InputValue)
    //输出候选集的全局计数
    OutResuh( OutputKey, OutputValue)
}
```

执行上述算法后,计算的可靠性是由 Hadoop 架构的可靠性提供的,Hadoop 能自动地维护数据的多份复制,并且能在任务失败后自动地重新部署计算任务,这就能保证节点失效后不会导致计算失败。

2.3　性能评价

为了对上述算法的性能进行评价,在此选取一批数据,进行了两组实验。

第 1 组实验,选择使用 20 个文件来分析,通过修改 Hadoop 的配置变量来改变 Map 能力进行实验,经过测试发现,在 Map 能力为 2、4、6、8、10 时,计算时间依次为 302、209、189、173、160 s。该实验说明,随着 Map 能力的增加,处理时间越来越短。

第 2 组实验,设置 Hadoop 中 Map 的最大任务数为 8,选取不同个数的文件(每个文件约包含 100 万条记录)进行对比实验,经过测试发现,文件数为 10、20、39 时的计算时间依次为 150、173、232 s。该实验说明,随着任务数的增加,处理时间是线性增加的,具体处理时间 y 和文件数 x 的关系可以描述为 $y = 122 + 2.8x$,该式中常数项 122 表示每一轮计算时 Hadoop 架构需要进行一些配置花费了一定的时间,但是,随着计算任务规模的扩大,Hadoop 配置任务的时间可以忽略不计。

在实际应用中,对于一个配置好的运行 Hadoop 架构的集群,期望应用程序应能灵活地使用集群中的资源。所以,基于 Hadoop 的并行关联规则挖掘计算能有效的利用集群的计算资源,在试验的过程,发现有些 Map 计算没有成功,但主进程及时调度其他计算资源完成了任务,避免了因节点失效导致的计算失败。

从平台性能上来说,计算时间和预期有一定的距离,第 1 组实验中 Map 能力为 8 的计算时间为 173 s,不是 Map 能力为 2 时的计算时间 302 s 的 1/4。第 2 组实验得到的近似关系式 $y = 122 + 2.8x$ 包含了常数项 122,这个时间基本是 5 次 job 配置的时间,扣除掉这个时间后,第 2 组实验的计算时间和任务数是成比例的。把 122 s 的配置时间考虑到第 1 组实验中,可以得到 Map 能力为 2 时的纯计算时间是 180 s,而 Map 能力为 8 的纯计算时间是 51 s,依然不是 4 倍的关系,原因是每个 job 处理 20 个文件需要执行 20 个 Map,假设一个 Map 处理一块的时间是 t,那么 Map 能力为 2 时完成一个 job 需要的时间为 10 t,而 Map 能力为 8 时完成一个 job 的时间为 3 t(前 16 个 Map 在 2 t 时间内完成,第 3 个时间 t 只有 4 个 Map 运行),180 s 和 51 s 的比例约等于 10 t 和 3 t 的比例。由于配置时间的存在,在处理小规模的数据集时浪费太多,故在 Hadoop 架构上的关联规则算法适合处理超大规模的数据集。

3　结束语

网络是一个巨大的、分布广泛的信息服务中心,其上产生的海量数据通常是地理上分布、异构、动态的,复杂性也越来越高,如果利用传统集中式数据挖掘方法则不能满足应用的要求。针对于此,提出了将海量数据和挖掘任务分解到多台服务器上并行处理。采用 Hadoop 开源平台,以关联规则数据挖掘为例,建立一个基于关联规则并行挖掘算法来验证了该算法的高效性。最后实验表明:该算法在 Hadoop 架构下,能够处理节点失败,其运行的可靠性远远优于传统的并行关联规则数据挖掘算法。

参考文献:

[1] 戎翔,李玲娟. 基于 MapReduce 的频繁项集挖掘方法[J]. 西安邮电学院学报,2011,16(4):37-39,43

[2] 余楚礼,肖迎元,尹波. 一种基于 Hadoop 的并行关联规则算法[J]. 天津理工大学学报,2011. 2,27(1):25-28

[3] YE Y B,CHIANG C C. A Parallel Apriori Algorithm for Frequent Item sets Mining[C] //Proceedings of the Fourth International Conference on Software Engineering Research Management and Applications(SERA′06). 2006:87-94

[4] 程苗. 基于云计算的 Web 数据挖掘[J]. 计算机科学,2011,38(10A):146-148

[5] SAVASERE A,OM IECI NSKI E,NAEATHE S. An efficient algorithm for mining association rules in large database[C] //Proc of the 21st International Conference on Very Large Databases. San Francisco:Morgan Kaufmann Publishers,1995

[6] 张守玉,刘博强. 基于数据挖掘技术的装备维修经费需求研究[J]. 四川兵工学报,2012(9):139-141

[7] 范明,王秉政. 一种直接在 Trans-树中挖掘频繁模式的新算法[J]. 计算机科学. 2003(8):117-123

[8] 蔡伟杰,张晓辉,朱建秋,等. 关联规则挖掘综述[J]. 计算机工程,2001,27(5):31-33

(下转第 60 页)