第42卷 第6A期
2015年6月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

Hadoop平台下Mahout聚类算法的比较研究

牛怡晗 海 沫

(中央财经大学信息学院 北京 100081)

摘 要 聚类是数据挖掘中的一门重要技术,用于将物理或抽象对象的集合划分成由相似对象构成的多个类。如何将传统聚类算法应用于大规模数据的聚类,是当前大数据研究领域中的热点研究问题。对云计算平台Hadoop下开源机器学习软件库——Mahout中的Canopy、标准K-means、模糊K-means 3种聚类算法的原理及其MapReduce实现进行了比较,并在构建的有不同个数节点的集群上,在不同规模的数据集下对这3种聚类算法进行了实验,从加速比、可扩展性和规模增长性3个方面进行比较。实验结果表明,在并行环境下:Canopy算法运行速度最快,K-means算法次之,模糊K-means最慢;3种算法均有较好的加速比,其中Canopy算法加速比最好,模糊K-means算法在数据量和节点个数达到一定规模后加速比大幅提高;3种算法均有较好的可扩展性和规模增长性,且随着数据规模增加,可扩展性和规模增长性增强,其中Canopy算法可扩展性最好,模糊K-means算法的可扩展性和规模增长性增强幅度最大。

关键词 聚类,Hadoop,Mahout,K-means,模糊K-means,Canopy

中图法分类号 TP301.6 文献标识码 A

Comparison Research on Mahout Clustering Algorithms under Hadoop Platform

NIU Yi-han HAI Mo

(School of Information, Central University of Finance and Economics, Beijing 100081, China)

Abstract Clustering is an important technique in data mining, and it is used to divide the congregation of physical or abstract objects into multiple classes consisting of similar objects. How to apply the traditional clustering algorithm into the clustering of large scale data is the hot research issue in the current data research field. This article conducts the theory analysis and comparison on the principle of three kinds of clustering algorithms of Canopy, Standard K-means and Fuzzy K-means in open-source machine learning software library—Mahout under cloud computing platform—Hadoop and the achievement of MapReduce, and on the cluster constructed by the nodes with different number, under the data sets with different scales, conduct experiment on the three kinds of clustering algorithms, and then conduct comparison from the three aspects of speedup ratio, scalability and scale growth. The experimental results show that: in parallel environment, the running speed of Canopy algorithm is the fastest, K-means algorithm is the second and Fuzzy K-means is the slowest; the three kinds of algorithms have better speedup ratio, and among them, the speedup ratio of Canopy algorithm is the best, the speedup ratio of Fuzzy K-means algorithm substantially increases after the amount of data and the number of nodes achieving a certain scale; the three kinds of algorithms have better scalability and scale growth, and among them, the scalability of Canopy algorithm is the best, the increasing amplitude of scalability and scale growth of Fuzzy K-means algorithm is the largest.

Keywords Clustering, Hadoop, Mahout, K-means, Fuzzy K-means, Canopy

1 引言

随着云计算、物联网以及移动互联网的普及和推广,商业、科研、金融等领域中的数据正以指数级别的速度产生。如何从海量的数据中挖掘有用信息已成为一个重要的问题。聚类是解决该问题的途径之一。所谓聚类,是将一个数据单位的集合分割成几个称为簇或类别的子集,使得同一簇中的对象尽可能相似,而不同簇中的对象尽可能相异。聚类分析是一个迭代学习的过程。在实际应用中,海量的数据及重复迭代对聚类技术提出了新的问题和挑战。为了减轻计算机计算负载,提高算法效率,满足海量数据的处理需求,能有效利用多台计算机计算能力的并行聚类成为一种有效的方法。

Hadoop是用于构建云计算平台的开源项目,可实现大规模分布式计算和并行处理,主要由HDFS分布式文件系统和MapReduce编程模型两部分组成。Mahout是Apache旗下适用于Hadoop云计算平台的一个开源项目,提供一些可扩展的机器学习领域经典算法的实现,旨在帮助开发人员更加方便快捷地创建智能应用程序。当前Hadoop在数据挖掘领域得到了广泛应用,而目前基于Hadoop平台的聚类算法的研究包括以下3个方面:第一,基于Hadoop平台的聚类算法

本文受北京高等学校青年英才计划项目(YETP0988)资助。

牛怡晗(1989—),女,硕士生,主要研究方向为大数据分析、产业经济学;海 沫(1978—),女,博士,副教授,主要研究方向为分布式计算、社交网络、大数据分析。

在某具体领域的应用。包括利用 K-means 聚类算法和 Apriori 关联规则算法对医保数据进行挖掘;利用 K-means 算法对气象大数据进行挖掘,对我国温度带和干湿区进行了划分;使用 Mahout 中的 K-means 算法对图书馆中读者的借阅数据进行分析研究,得到图书使用率聚类等。第二,基于 Hadoop 中 MapReduce 编程模型的传统聚类算法的改进。主要针对具体的应用领域,根据实际数据特征,采用 MapReduce 并行编程模型,对聚类算法进行改进,使其适用于特定领域内海量数据的分析和挖掘。第三,通过实验对传统聚类算法在 Hadoop、MPI、Spark 等不同云计算平台中的运行效率和聚类质量进行比较。

综上所述,可见对 Hadoop 云计算平台的研究与应用已经成为大数据时代数据挖掘领域中一个非常活跃的研究课题,而从众多聚类算法中根据实际应用的具体需求挑选出适合的聚类算法并移植到 Hadoop 平台上进行分布式实现,以提高海量数据的聚类效率和质量,具有重大的意义。因此,对 Mahout 算法库中的聚类算法在 Hadoop 云计算平台下的并行化实现的比较研究成为当前的一个重要研究方向。本文在 Hadoop 云计算框架下,对 Mahout 算法库中提供的 Canopy、K-means 及模糊 K-means 3 种经典聚类算法进行比较研究,对比分析其在不同规模集群及数据量下的处理效率。结果表明,在并行环境中 3 种聚类算法对于大规模数据处理具有更高的效率。这对于构建海量数据挖掘的应用有一定的参考价值。

2 核心技术介绍

Canopy 算法是一种快速近似的聚类技术,算法使用一个快速的距离测度和两个距离阈值 T_1、T_2($T_1<T_2$),计算每个数据点到各 Canopy 中心的距离并与 T_1、T_2 比较,通过迭代数据点,将其划分为一些重叠的簇;K-means 算法是一种经典的排他性聚类算法,算法从包含 k 个中心点的初始集合开始,根据指定的距离算法计算各个数据点到各中心的距离,并根据最近距离原则将各数据点划分给最近的簇;模糊 K-means 聚类算法是 K-means 算法的扩展,可以生成有重叠的簇,其原理与 K-means 算法类似,通过对数据集迭代计算,根据关联程度将各个数据点划分到合适的簇。3 种聚类算法的具体比较如表 1 所列。

表 1　3 种聚类算法性能的比较

比较指标	Canopy	K-means	模糊 K-means
簇之间是否重叠	是	否	是
是否事先确定簇个数	否	是	是
初始中心选择	随机选取 1 个点	随机选取 k 个点	随机选取 k 个点
算法迭代次数	一次	N(N≤最大迭代次数)次	N(N≤最大迭代次数)次
算法主要参数	距离算法 dm,距离阈值 T_1、T_2	簇个数 k,距离算法 dm,最大迭代次数 x,收敛阈值 cd	簇个数 k,距离算法 dm,最大迭代次数 x,收敛阈值 cd,模糊参数 m
划分原则	与已经确定的各 Canopy 中心的距离小于距离阈值(距离小于 T_2($T_2<T_1$)时不能作为新 Canopy 中心)	距离各个簇中心最近	与各个簇中心关联程度(U)最大(关联程度与某点到簇中心的距离成正比)
中心更新原则	与已经确定的各 Canopy 中心的距离大于 T_1 则作为新 Canopy 中心	各个簇中所有点的坐标的均值	各个簇中所有点的坐标的均值
算法终止条件	数据列表为空(所有点均被划分到 Canopy 中)	达到最大迭代次数,或中心收敛于确定阈值	达到最大迭代次数,或中心收敛于确定阈值
质量影响因素	距离算法,距离阈值 T_1、T_2 选择	距离算法、初始中心位置	距离算法、初始中心位置、模糊参数(模糊参数越大,重叠部分越多)
MapReduce 实现方式	两个 map 操作,一个 reduce 操作	两个 map 操作,一个 combine 操作,一个 reduce 操作	两个 map 操作,一个 combine 操作,一个 reduce 操作
算法时间复杂度	$O(nkf^2/c)$(n 为数据对象个数,k 为聚类个数,f 为平均每个点划分到的 Canopy 个数,c 为 Canopy 总个数)	O(ndkt)(n 为数据对象个数,d 为维度,k 为聚类个数,t 为迭代次数)	O(ndkt)(n 为数据对象个数,d 为维度,k 为聚类个数,t 为迭代次数)

在 Mahout 中,并行化 Canopy 算法首先在 Datanode 用一个 map 操作对本地数据进行 Canopy 划分并输出中心值,之后在 Namenode 用一个 reduce 操作对各 Datanode 上生成的中心值进行归并,得到全局 Canopy 中心,最后使用一个 map 操作对所有数据点进行聚类。并行化 K-means 算法每次迭代在 Datanode 都用一个 map 操作对位于本地的数据集进行局部 K-means 聚类并输出新中心值,用一个 combine 操作对本地结果进行汇总,在 Namenode 用一个 reduce 操作将各 Datanode 生成的簇进行汇总,得到 k 个簇中心。迭代结束后,Namenode用一个 map 操作对所有数据点进行聚类。并行化的模糊 K-means 算法同 K-means 算法一样,每次迭代都用一个 map、一个 combine 和一个 reduce 操作得到并保存全局簇集合,迭代结束后,用一个 map 操作得到全局簇集合。

3 实验及结果分析

3.1 实验环境、数据集和评价指标

本实验搭建的 Hadoop 集群环境由分别在 8 个物理节点上部署的 8 台 VMware 虚拟机组成,其中 1 台为 Namenode,其余 7 台为 Datanode。物理机 CPU 2.93GHz、内存 2GB、硬盘 500GB,虚拟机版本为 VMware Workstation10.0.0,CPU 2.66GHz、内存 1GB、硬盘 20GB。JDK 版本为 jdk1.6,Hadoop 版本为 Hadoop-1.1.2,Mahout 版本为 Mahout-0.7。其中 Hadoop 块大小参数设置为 32MB,单个节点最大运行 map 个数为 2,reduce 个数为 1。实验数据使用 UCI 的 Bag of Words 数值型数据集,该数据集选取 PubMed 网站上 8200000 篇文章摘要,共包含 730000000 个单词,以 docID wordID count 格式将其数值化为具有 3 个维度的数值型文本文件。为比较 3 种不同聚类算法在不同规模数据量下的性能差别,结合集群规模及节点配置的计算能力,从数据集中提取了五百万到五千万条数据进行了 4 组测试。数据集分别为:pub1:67.9MB(500 万条)、pub2:140.7MB(1000 万条)、pub3:431.9MB(3000 万条)、pub4:725.6MB(5000 万条)。

在实验中,采用加速比(speedup ratio)、可扩展性(scalability)和规模增长性(scale growth)[1] 作为评价指标。

3.2 实验结果

由于 K-means 算法和模糊 K-means 算法中有随机初始

化中心点的操作，需对每一组实验重复执行 10 次，取其平均执行时间作为最终每组实验的结果。3 种算法的参数设置如下：Canopy 算法的参数选取得到 Canopy 数为 25 的 T_1 和 T_2；K-means 和模糊 K-means 算法中，考虑到机器性能的限制，迭代次数 x 为 5、簇个数 k 为 20；距离算法 d 为欧几里得距离，收敛阈值 cd 为 0.5。

3.2.1 3 种算法在不同节点数下运行时间的比较

3 种算法在不同节点数下运行时间的比较如图 1—图 4 所示。

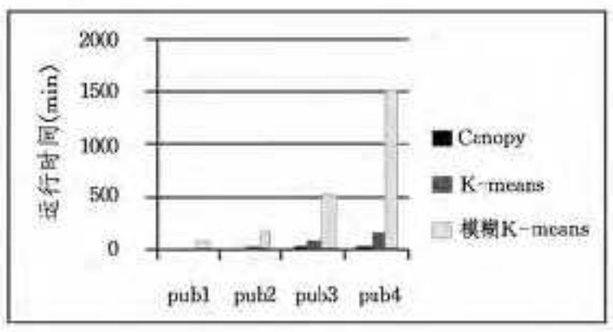

图 1 单个节点下 3 种算法运行时间对比

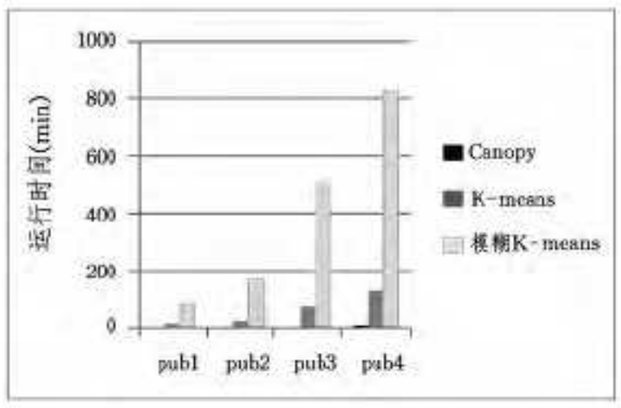

图 2 2 个节点下 3 种算法运行时间对比

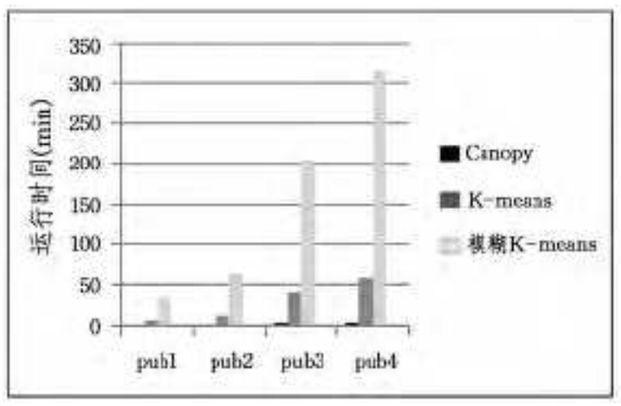

图 3 4 个节点下 3 种算法运行时间对比

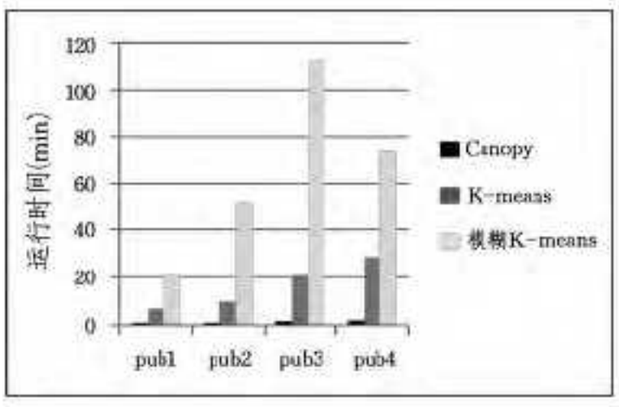

图 4 8 个节点下 3 种算法运行时间对比

从图 1—图 4 中可以看出，在不同数据量下，Canopy 算法的运行时间均明显少于 K-means 算法和模糊 K-means 算法；而 K-means 算法的运行时间少于模糊 K-means 算法。原因在于，Canopy 算法只需遍历一次数据即可得到结果，且算法仅使用 T_1 和 T_2 两个距离测度对各个点到 Canopy 中心的距离进行简单比较，从而实现对数据集的粗略划分。所以，该算法的运行时间要比 K-means 算法和模糊 K-means 的短。与 K-means 算法相比，模糊 K-means 算法由于关联程度计算公式的复杂性，在对每一个数据点进行划分时计算量更大。而且，模糊 K-means 生成可重叠的簇，故每次迭代需要处理更多的数据。所以，在设置相同的迭代次数、收敛阈值及聚类个数 k 值时，模糊 K-means 算法应比 K-means 算法效率更低。

3 种算法在节点个数分别为 1、2、4 的情况下，随着数据量增加，运行时间均增加；而在 8 个节点时，模糊 K-means 算法的运行时间在数据量由 pub3(431.9MB)增加到 pub4(725.6MB)时反而减少。因而我们可以推断，模糊 K-means 算法的性能在数据量达到某一阈值后有所提高。在节点个数和数据量到达一定阈值时，模糊 K-means 算法比标准 K-means 算法收敛得更快。并行条件下，Canopy 算法随着节点个数的增加，数据量的改变对算法运行时间的影响越小，在 2 个节点时 pub4 的运行时间大约是 pub1 的 10 倍，而在 4 个节点时 pub4 的运行时间大约是 pub1 的 5 倍，而当节点个数增加到 8 个时，pub4 的运行时间仅是 pub1 的 2 倍左右。这表明，Canopy 算法在大规模集群并行环境中能快速收敛，适用于大规模数据的预处理。

3.2.2 3 种算法加速比比较

图 5—图 7 分别为 Canopy、K-means、模糊 K-means 3 种算法在不同数据量下的加速比测试的实验结果。由图可知，3 种算法在不同规模数据集下都有大于 1 的加速比，而 Canopy 算法的加速比明显大于 K-means 算法及模糊 K-means 算法。

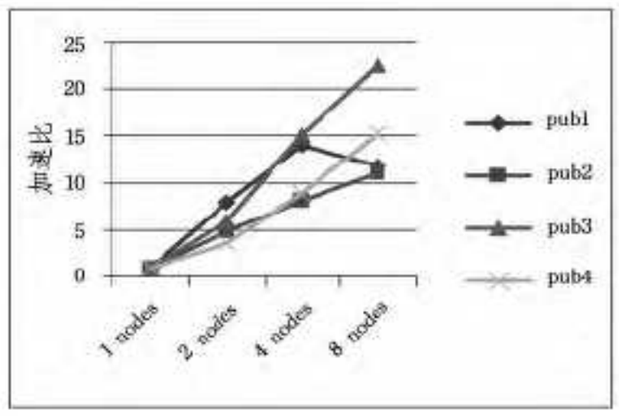

图 5 Canopy 算法加速比性能测试结果

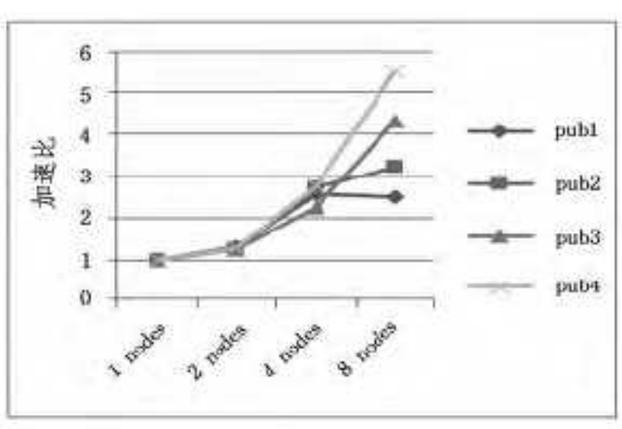

图 6 K-means 算法加速比性能测试结果

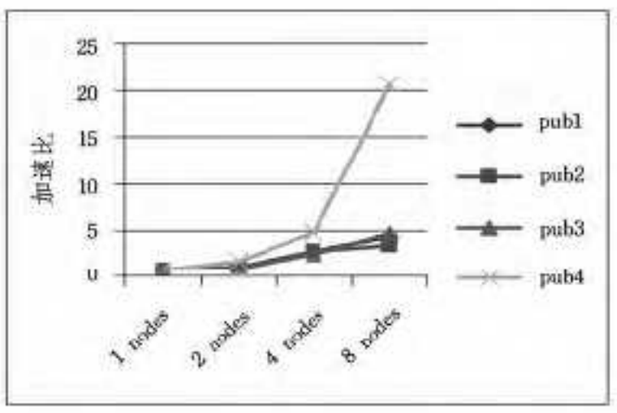

图 7 模糊 K-means 算法加速比性能测试结果

由图 5 可以看出，Canopy 算法的加速比除了在 pub1(67.9MB)数据集，均为近似线性。这表明在数据量大时，随着节点数的增加，该算法能够很好地缩短时间。但在一定数据规模下，随着数据量的增加(pub2 对比 pub1、pub4 对比 pub3)，加速比反而变差，这说明主节点和各从节点之间的通信代价比率增加，从节点之间的通信代价比率也增加，影响了算法性能。

由图 6 和图 7 可以看出，K-means 算法和模糊 K-means 算法的加速比均为近似线性；并且，在数据集规模达到 pub3(431.9MB)后，随着数据规模的扩大，加速比性能越来越好。这表明在数据量大时，随着节点数的增加，这两种算法都能很

好地缩短时间。这说明 Mahout 中这两种算法非常适合大数据的聚类。随着数据集规模增长，主节点和从节点之间的通讯代价减少比例变高，大数据集更有效地利用了每个节点的计算性能。比较这两种算法，当节点数增加到 8 个，数据集规模达到 pub4(725.6MB) 时，模糊 K-means 算法的加速比从 4.43 增加到 20.56，而 K-means 算法仅从 2.51 增加到 5.52，模糊 K-means 算法的加速比增加幅度是 K-means 算法的 2 倍。这表明数据量越大，模糊 K-means 算法相对 K-means 算法的加速比性能越好，说明模糊 K-means 算法对于大规模数据的并行化聚类效率提高得更快。

3.2.3 3 种算法可扩展性比较

由图 8 可以看出，Canopy 算法的可扩展性均大于 1，说明 Canopy 算法具有很好的可扩展性。这是因为 Canopy 算法不需要多次迭代，仅使用一次 MapReduce 操作，节点间的通信代价增加比例很小。当数据量从 pub1 增加到 pub2、pub3 增加到 pub4 时，算法可扩展性略有下降，表明 Canopy 算法在一定数据量范围下，节点个数增加对算法效率提高的影响小于节点间通信代价增加及数据量增加对算法效率的影响。而当数据量从 pub2 增加到 pub3 时，算法可扩展性增强，表明 Canopy 算法对数据量的增加引起的可扩展性的变化较迟钝，只有当数据量增加比例达到某个阈值时，可扩展性才会增强。比较 pub1、pub2 及 pub3、pub4 两组不同数据量下的可扩展性，在大数据量下，随节点个数增加，算法可扩展性下降幅度更加平缓，算法表现出更加稳定的性能。

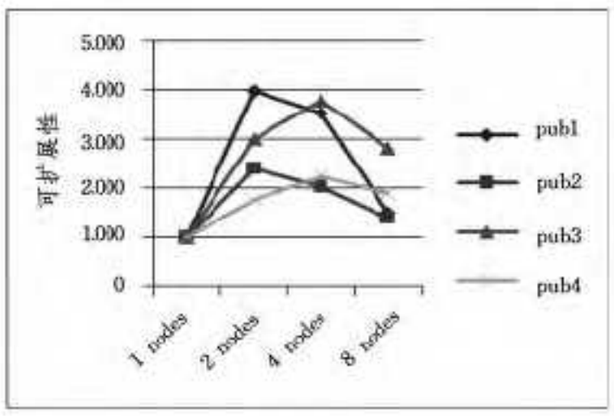

图 8 Canopy 算法可扩展性性能测试结果

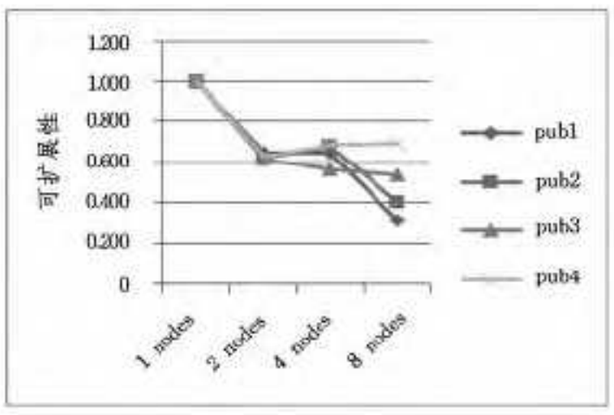

图 9 K-means 算法可扩展性性能测试结果

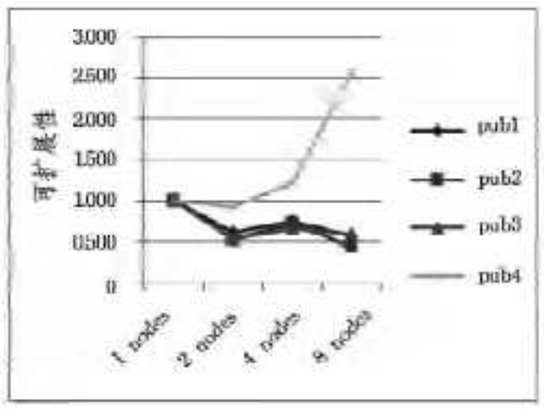

图 10 模糊 K-means 算法可扩展性性能测试结果

由图 9、图 10 可以看出，K-means 算法及模糊 K-means 的可扩展性均小于 1。这主要是由于节点数增加，节点之间的通信时间增加，算法完成计算的时间加长，加速比减小。但随着数据规模增加，两种算法的扩展性都在增强，且随着节点个数的增加，可扩展性下降得更平缓。这主要由于在相同数量的节点下，数据集越大，节点之间的通信时间比率就越小，加速比就越大。

3.2.4 3 种算法规模增长性比较

图 11—图 13 分别为 Canopy、K-means、模糊 K-means 3 种算法在不同节点个数的平台下的规模增长性性能测试的实验结果。从图中可以看出，当节点数为 1 时，3 种算法的运行时间随着数据集的增大而迅速增加；随着节点个数不断增加，3 种算法的运行时间随着数据集的增大增长得越缓慢。由此可见，当数据集比较大，节点数比较多时，3 种算法的执行效率都有所提高。

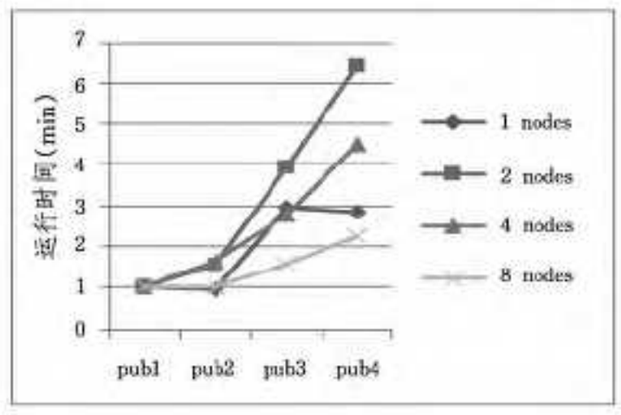

图 11 Canopy 算法规模增长性性能测试结果

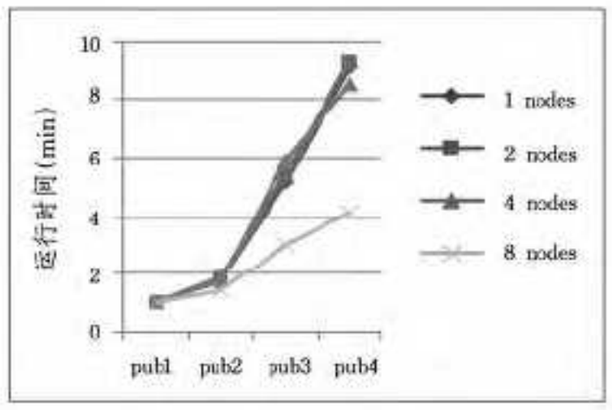

图 12 K-means 算法规模增长性性能测试结果

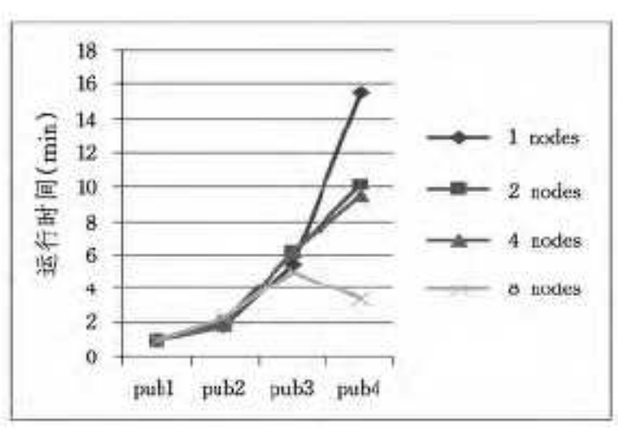

图 13 模糊 K-means 算法规模增长性性能测试结果

比较 K-means 和模糊 K-means 算法，模糊 K-means 算法运行时间在不同节点数及不同数据集规模下都要明显多于 K-means 算法的。但随着节点数的增加，模糊 K-means 算法的规模增长性减少幅度明显大于 K-means 算法，说明模糊 K-means 算法能更好地适应大规模并行计算平台。

结束语 本文对 Mahout 下 Canopy、K-means 及模糊 K-means 3 种聚类算法做了深入的分析和比较。首先，分析比较了 Mahout 下 3 种聚类算法的原理及其 MapReduce 并行化实现；接着对不同规模数据集在不同节点个数集群中进行了实验。根据实验结果可得如下结论：1. 在数据量和节点个数相同的情况下，Canopy 算法的运行速度最快，K-means 算法次之，模糊 K-means 算法最慢。2. 在 Hadoop 环境中，3 种算法均有较好的加速比，Canopy 算法加速比明显大于其余两种算法。模糊 K-means 算法相较于 K-means 算法收敛得更慢，但随着节点个数和数据量的增大时，模糊 K-means 算法的加速比增加幅度是 K-means 算法的 2 倍，说明模糊 K-means 算法对于大规模数据的并行化聚类效率提高得更快。3. 在不同

节点个数下，3 种算法都具有较好的扩展性，其中 Canopy 算法具有最好的可扩展性。当数据量增大时，Canopy 算法相较 K-means 算法和模糊 K-means 算法速度优势更明显，在 8 个节点数据量为 pub4 时，Canopy 算法时间约为 K-means 算法的 1/15，为模糊 K-means 算法的 1/25，表明 Canopy 非常适合用于海量数据聚类的预处理。4. 随着节点个数不断增加，3 种算法的运行时间随着数据集的增加增长得越缓慢，但随着节点数的增加，模糊 K-means 算法的规模增长性减少幅度明显大于 K-means 算法，说明模糊 K-means 算法能更好地适应大规模并行计算平台。5. 由于 3 种算法都具有较好的加速比、可扩展性和可伸缩性，说明 Mahout 下的 3 种算法能很好地运行于 Hadoop 平台，可以有效地应用于海量数据的聚类。

参 考 文 献

[1] 赵卫中，马慧芳，傅燕翔，等. 基于云计算平台 Hadoop 的并行 k-means 聚类算法设计研究[J]. 计算机科学，2011(10)：166-168，176

[2] Owen S，Anil R，Dunning T，et al. Mahout in action[M]. USA：Manning Publications，2010

[3] 胡俊. 集群环境下聚类算法的并行化研究与实现[D]. 上海：华东师范大学，2010

[4] Ericson C，Pallickara S. On the performance of high dimensional data clustering and classification algorithms[J]. Future Generation Computer Systems，2013(29)：1024-1034

[5] 潘昊斌. 基于云计算的并行 K-means 气象数据挖掘研究与应用[D]. 南京：南京信息工程大学，2013

[6] 怀特. Hadoop 权威指南[M]. 北京：清华大学出版社，2010

[7] 王彦明，奉国和，薛云. 近年来 Hadoop 国外研究综述[J]. 计算机系统应用，2013，22(6)：1-5，28

[8] Apache Hadoop[OL]. http://Hadoop.apache.org

[9] Apache Mahout[OL]. http://Mahout.apache.org

[10] 张明辉. 基于 Hadoop 的数据挖掘算法的分析与研究[D]. 昆明：昆明理工大学，2012

(上接第 446 页)

计算(按协议正常计算即可)，而 s_1 为用户 θ 自己所随机选择的，故可如上设置 $K'_{\theta,\sigma}$。

2)如果 $t=R$，此时应有 $\theta=A$，设置 $C'=g^s$(取自判定 q-parallel BDHE 问题实例 $\vec{y}$)。选择一个 LSSS 访问结构(M,ρ)，直觉上，$\vec{v}$ 应该为：

$$\vec{v}=(s,sa+y_2',sa^2+y_3',\cdots,sa^{n-1}+y'_{n^*})\in Z_p^{n^*}$$

其中，$y_2',\cdots,y'_{n^*}\in Z_p$ 是随机选择的值。选择 $r_1',\cdots,r_l'\in Z_p$，对于 $i=1,\cdots,n^*$，定义 R_i 表示对于 $k\neq i$，有 $\rho^*(i)=\rho^*(k)$ 的一个 k 集合，如下设置 C_i,D_i：

$$D_i=g^{-r_i'}g^{-sb_i},$$

$$C_i=h_{\rho^*(i)}^{r_i'}\Big(\prod_{j=2,\cdots,n^*}(g^a)^{M^*_{i,j}y_j'}\Big)(g^{b_i s})^{-z_{\rho^*(i)}}\cdot$$

$$\Big(\prod_{k\in R_i}\prod_{j=1,\cdots,n^*}(g^{a^j s(b_i/b_k)})^{M^*_{k,j}}\Big)$$

返回 $M_\theta=\{(M,\rho),C',(C_1,D_1),\cdots,(C_{n^*},D_{n^*})\}$。模拟器可以设置 $K'_{\theta,\sigma}=K_\sigma\cdot T\cdot e(g^s,g)^{\alpha'}$，其中 K_σ 可根据用户 θ 的私钥(可通过调用 $Corrupt(\theta)$ 获得)和自敌手收到的 M 来计算(按协议正常计算即可)，而 g^s 来自判定 BDHE 问题实例，α' 为模拟器自己随机选择的值，故可如上设置 $K'_{\theta,\sigma}$。

$Reveal(\Pi^t_{\theta,\sigma})$：如果询问的预言机是 $\Pi^R_{A,B}$，或者是它的匹配预言机 $\Pi^{R'}_{B,A}$，则模拟器中止。否则，返回该预言机持有的会话密钥给敌手。

$Test(\Pi^t_{\theta,\sigma})$：敌手最后对预言机询问一次 $Test$。如果敌手询问的不是 $\Pi^R_{A,B}$，模拟器中止。否则，需要预言机 $\Pi^R_{A,B}$ 处于接受状态，并且 $\Pi^R_{A,B}$，或者是它的匹配预言机 $\Pi^{R'}_{B,A}$ 没有回答过 $Reveal$ 询问，以及 A,B 没有回答过 $Corrupt$ 询问。模拟器抛一枚硬币 μ，如果 $\mu=0$，返回 $\Pi^R_{A,B}$ 持有的会话密钥，否则随机返回会话密钥空间里的一个值。

如果敌手输出 $\mu'=\mu$，模拟器输出 0，否则输出 1。根据先前的设置，挑战者有 1/2 的概率设置 $T=e(g,g)^{\alpha^{q+1}s}$，在这种情况下如果模拟器正常模拟且假设没有中止，那么模拟器有不可忽略的优势 $\varepsilon/2q_o$ 判定 q-parallel BDHE 问题。这同判定 q-parallel BDHE 假定成立相矛盾。因此得出结论：不存在概率多项式时间敌手 A，能以不可忽略的优势 ε 成功攻击协议，故文中所设计的协议在标准模型下是一个安全的认证密钥协商协议。

结束语　本文基于 Waters 的密文策略属性基加密方案中的密钥提取形式设计了一个两方的消息策略的属性基密钥协商协议，协议具有较高的效率、标准模型下的可证明安全以及访问结构表达能力强等优点。特别地，选择安全模型下的证明可进一步扩展到支持完全安全模型下的证明。此外，同时满足上述特性的属性基群密钥协商协议是本文的后续研究工作。

参 考 文 献

[1] Sahai A，Waters B. Fuzzy identity based encryption：Advances in Cryptology-Eurocrypt 2005[C]// LNCS Berlin：Springer-Verlag，2005，3494：457-473

[2] Beimel A. Secure schemes for secret sharing and key distribution[D]. Haifa：Israel Institute of Technology，1996

[3] Wang Hao，Xu Qiu-liang，Fu Xiu. Two-Party attribute-based key agreement protocol in the standard model[C]// the 2009 International Symposium on Information Processing(ISIP'09). Finland：Academy Publisher，2009：325-328

[4] Wang Hao，Xu Qiu-liang，Fu Xiu. Revocable attribute-based key agreement protocol without random oracles[J]. Journal of Networks，2009，4(8)：787-794

[5] 王永涛，宋璟，贺强，等. 一个基于属性的密钥协商协议[J]. 计算机工程，2014，40(2)：134-139

[6] 魏江宏，刘文芬，胡学先. 全安全的属性基认证密钥协商协议[J]. 计算机应用，2012，32(1)：. 38-41

[7] 王永涛，封维端，刘孝男，等. 一个消息策略基于属性的密钥协商协议[J]. 计算机科学，2013，40(9)：106-110

[8] Waters B. Ciphertext-policy Attribute-based Encryption：An Expressive，Efficient，and Provably Secure Realization[C]// PKC，LNCS，vol. 6571，Berlin：Springer-Verlag，2011：53-70

[9] Boneh D，Franklin M. Identity based encryption from the Weil pairing：Advances in Cryptology-Crypto 2001[C]// LNCS，vol. 2139，Berlin：Springer-Verlag，2001：231-229

[10] Beimel A. Secure Schemes for Secret Sharing and Key Distribution[D]. Israel Institute of Technology，Technion，Haifa，Israel，1996

[11] Chen Li-qun，Cheng Zhao-hui，Smart N P. Identity-based Key Agreement Protocols From Pairings. Cryptology ePrint Archive[OL]. http://eprint.iacr.org/2006/199

[12] Bellare M，Rogaway P. Entity authentication and key distribution：Advances in Cryptology-CRYPTO 1993[C]// LNCS，vol. 773，Berlin：Springer-Verlag，1994：232-249