第 42 卷 第 6A 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

基于 Hadoop 的并行 PSO-kmeans 算法实现 Web 日志挖掘

马汉达 郝晓宇 马仁庆

(江苏大学计算机科学与通信工程学院 镇江 212013)

摘 要 互联网技术的迅速发展,使得基于单一结点的 Web 日志挖掘变得十分困难,而 Hadoop 云平台的出现,为这类问题提供了新的解决方案。但传统的 Web 日志挖掘聚类 k-means 算法对初始聚类中心的选择敏感等缺点,容易影响聚类准确率。针对这个问题,提出基于粒子群算法(PSO)的 k-means 算法,使得 k-means 算法不受初始聚类中心的影响,并且在 Hadoop 平台上实现了算法的 MapReduce 编程。实验结果证明:提出的改进算法,与传统的 k-means 算法相比,具有更高的聚类准确率;与串行单机算法相比,运行效率也有很大的提升。

关键词 Hadoop,k-means,PSO,MapReduce,Web 日志挖掘

中图法分类号 TP311 文献标识码 A

Parallel PSO-kmeans Algorithm Implementing Web Log Mining Based on Hadoop

MA Han-da HAO Xiao-yu MA Ren-qing

(School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract With the rapid development of Internet technology, Web log mining based on a single node becomes very difficult. The emergence of Hadoop cloud platform provides a new solution to this problem. However, the traditional Web log mining clustering algorithm k-means is sensitive to the initial cluster centers selection, so it will easily affect the accuracy of clustering. Thus for this problem, this paper proposed a k-means algorithm based on particle swarm optimization which makes the k-means algorithm not be affected by the initial cluster centers. And the algorithm is realized in the Hadoop MapReduce programming platform. Experimental results show that: compared with traditional k-means algorithm the proposed algorithm has the higher clustering accuracy, and compared with stand-alone serial algorithm, the operating efficiency improved greatly.

Keywords Hadoop, k-means, PSO, MapReduce, Web log mining

Web 日志文件是用来记录用户访问页面情况的 Web 服务器文件。日志文件记录最主要的是记录了访问者的 IP、浏览器类型、访问格式、访问页面协议等用户访问的一切行为信息,因此 Web 日志文件很大。通过分析 Web 日志可以分析用户行为,对优化网站结构具有十分重要的意义。分析 Web 日志文件寻找用户行为的特征信息,常用的方法就是数据挖掘。但是由于 Web 日志的数据量大,采用传统数据挖掘算法将不能很好地解决 Web 日志数据大的问题,因此,提出了将数据挖掘算法运用到 Hadoop 平台的方法,实现了高效 Web 日志挖掘。

Web 日志挖掘一般分为 3 个阶段[1]:预处理阶段、挖掘算法实施阶段、分析阶段。其中预处理阶段是 Web 日志挖掘的重要组成部分[2],预处理的结果将会很大程度影响后续挖掘算法的效果。传统的预处理实现在 TB 甚至 PB 级的大数据面前,效率十分低下。文献[3]提出使用 Hadoop 并行平台实行预处理,能极大提高预处理效率。所以本文在 Hadoop 平台下对预处理使用 MapReduce 编程实现。MapReduce 在处理时,首先要将大量的数据进行分片,如何分片也直接影响处理的效率。K-means 算法是数据挖掘算法中基于划分的经典聚类算法,实现简单,收敛速度快。但初始聚类中心的选择容易影响聚类效果[4],面对 Web 日志的数据串行化的处理效率也低。针对以上传统 k-means 算法的缺点,文献[5,6]提出以粒子群优化(Particle Swarm Optimization, PSO)算法来确定初始聚类中心,在一定程度上提高了准确度,但未能很好地提高算法的运行效率。文献[7]也提出了基于 Hadoop 的 k-means 算法,提高了数据处理效率,但并没有解决初始聚类中心的问题。所以本文将 PSO 和 k-means 算法结合起来,并且在 Hadoop 平台下实现了 PSO-kmeans 算法,消除了 k-means 算法对初始聚类中心的依赖,并且极大提高了处理效率。

1 Hadoop

Hadoop 是一个集成了分布式文件系统 HDFS 和大规模并行计算模型 MapReduce 的开源框架[8]。Hadoop 的两大核心是 HDFS 和 MapReduce。HDFS[9]是分布式文件系统,有高容错性的特点,并且设计用来部署在低廉的硬件上;而且它提供高吞吐量来访问应用程序的数据,适合那些有着超大数据集的应用程序。HDFS 放宽了 POSIX 的要求,可以以流的形式访问文件系统中的数据。MapReduce 是一种编程模型,

马汉达(1966—),男,硕士,高级工程师,CCF 会员,主要研究方向为云计算、数据挖掘、Web 信息系统与教育信息化,E-mail:mahd@ujs.edu.cn;郝晓宇(1993—),男,硕士生,主要研究方向为云计算、数据挖掘;马仁庆(1993—),男,硕士生,主要研究方向为数据挖掘、可信计算。

用于大规模数据集(大于 1TB)的并行运算,方便编程人员将自己的程序运行在分布式系统上,分别由 Map 和 Reduce 函数实现。Map 函数用来把一组键值对映射成一组新的键值对,Reduce 函数用来保证所有映射的键值对中的每一个共享相同的键组,MapReduce 过程如图 1 所示。因此,Hadoop 是一个能够让用户轻松架构和使用的分布式计算平台,用户可以方便地在 Hadoop 上开发和运行处理海量数据的应用程序。

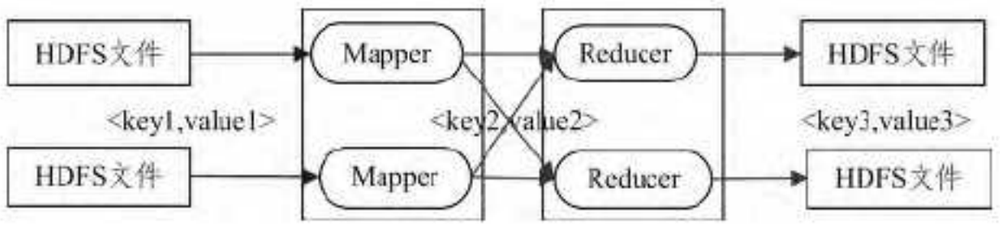

图 1 MapReduce 分析过程

2 基于 MapReduce 的 Web 日志挖掘预处理

本文实验选取的 Web 日志是 apache 服务器的 log 文件,主要格式如表 1 所列。预处理主要包括:数据清洗、用户识别、会话识别[10]等步骤。本文主要研究改进聚类算法,所以只做数据清洗、用户识别就能够达到预期结果。

表 1 apache 日志文件格式

日志内容	含义
127.0.0.1	用户 IP 地址(UserIP)
-	用户 ID(通常为空)
[28/Oct/2014:13:43:24 +0800]	访问时间(Time)
GET	请求方法(Method)
/Bootstrap/Example1/login.php	请求页面(URL)
HTTP/1.1	传输协议版本(Version)
200	返回的 Http 状态标识(Status)
482	服务器发送字节数(Byte)
/Bootstrap/Example1/index.php	用户浏览的上一页(ReferURL)
Mozilla/5.0(……)	浏览器操作系统版本(BrowserOS)

数据清洗是指根据需求,对日志文件进行处理,删除与挖掘上任务不相关的数据并合并某些记录,对用户请求页面时发生错误的记录进行适当的处理等等[10]。如把日志中文件的后缀为 gif、jpg、png、jpeg 等的记录删除;后缀名为 css、js 等的脚本文件对后面的分析处理没有任何影响,也应该删除。常见的页面请求方法包括 GET、POST 和 HEAD。由于只有 GET 方法反映了用户的访问行为,因此,本文只保留 GET 方式的记录。

用户识别[11]是指如何识别互联网上用户的身份,常用的方法是根据 IP 地址来区分用户,如果 IP 地址相同,但是日志中用户的浏览器或操作系统不同,则认为用户不同;如果日志中两条记录的 IP 地址相同且浏览器也相同,但是没有连接关系存在于用户当前请求的 Web 页面与用户已经浏览的 Web 页面之间,就认为又出现了一个新的用户。

2.1 Map 阶段

从输入的日志文件中提取数据,使用正则表达式清洗掉不需要的日志记录,如检测 URL 字段的后缀名是否为.css、.js,过滤掉请求方式非 GET 的记录,提取记录的 UserIP 和浏览器信息(BrowserOS)拼接作为 Map 函数的 key,用户当前访问路径(path)作为 value,这样就能实现数据清洗和用户识别。

2.2 Reduce 阶段

接收 Map 阶段写入的 key 和 value,此时 key 就代表一个用户,value 中就是一系列 path 组成的该用户访问页面记录,存放在一个 Iterable 中。遍历该迭代器,统计出想要进行聚类的属性出现的次数,Reduce 函数的输出便是每个用户及其访问统计页面的次数。

3 相关算法简介

3.1 k-means 算法

目前传统的聚类方法主要是基于层次聚类和迭代的平方误差分区聚类,在诸多的聚类算法中 k-means 是应用最广泛的算法之一[12]。k-means 算法是输入聚类个数 k,以及包含 n 个数据对象的数据,输出满足方差最小标准的 k 个聚类,满足聚类内部相似度高而聚类之间相似度低的要求。具体的算法流程是:

(1)从 n 个数据对象中任意选择 k 个对象作为初始聚类中心;

(2)根据每个聚类对象的均值(中心对象),计算每个对象与这些中心对象的距离,并根据最小距离重新对相应对象进行划分;

(3)重新计算每个聚类的均值(中心对象);

(4)计算标准测度函数,当满足一定条件,如函数收敛时,则算法终止,如果条件不满足则回到步骤(2)。

3.2 PSO 算法

粒子群优化算法是由 Kennedy 和 Eberhart 提出的一种模拟鸟群觅食过程中群体行为的新型群体智能算法[13]。PSO 算法初始化为一群随机粒子(随机解),然后通过迭代找到最优解。在每一次迭代中,粒子通过跟踪两个极值来更新自己。第一个就是粒子本身所找到的最优解即个体极值,另一个就是整个种群目前所找到的最优解即全局极值。

粒子速度公式:

$$v_i = v_i(t) + c_1 \times rand() \times (p_i - x_i) + c_2 \times rand() \times (g - x_i) \tag{1}$$

其中,c_1、c_2 是学习因子,$rand()$ 为 $(0,1)$ 区间内的随机数。

PSO 适应度公式:

$$f(x_i) = \sum_{i=1}^{k} \sum_{p \in c_i} |p - \overline{m_i}|^2 \tag{2}$$

即误差平方和公式。

PSO 位置更新公式:

$$x_i = x_i + v_i \tag{3}$$

算法流程:

(1)对粒子群的随机位置和速度进行初始设定;

(2)计算每个粒子的适应值;

(3)对每个粒子,将其适应值与其所经过的最好位置 P_i 的适应值进行比较,若更好,则将其作为当前最好位置;

(4)对每个粒子,将其适应值与全局经历的最好位置 P_g 的适应值做比较,若更好,则将其作为当前的全局最好位置;

(5)根据上述位置和速度迭代公式对粒子的速度和位置进行进化;

(6)如未能达到结束条件或超出最大迭代次数,返回(2),否则执行(7);

(7)输出最优值,即为 k-means 算法的最优初始聚类中心。

4 基于 Hadoop 的并行 PSO-kmeans 算法

由于 k-means 算法对初始聚类中心敏感,容易影响聚类结果,因此很多研究人员在研究聚类算法的同时采用了 PSO 算法来解决这样的问题。如文献[14]采用的基于改进的 PSO-kmeans 算法就很好地弥补了 k-means 算法的不足,提高了算法的准确度,但其串行化的程序思想仍然限制了对于大数据的处理效率。由于 PSO 算法可以在 Hadoop 并行平台上运行[15],本文提出了将 PSO 和 k-means 算法都并行化处理的思想。

4.1 基于 Hadoop 的 PSO 算法的实现

算法的描述如下:输入待聚类数据集 N、聚类数目 k、粒子群的种群规模 m;输出聚类数据集的聚类中心不再变化的 k 个聚类划分。算法的步骤:

(1)将数据集 N 读入,初始化粒子群。从 N 中选取 k 个初始聚类中心,将其作为粒子初始位置 X_i,并且初始化粒子的速度、个体最优位置及其对应的个体极值、群体最优位置及其对应的全局极值。这一过程循环进行 m 次,即可完成粒子群的初始化构造[14]。

(2)根据式(1)、式(3)更新粒子的位置和速度。

(3)依照最近邻原则划分数据集,计算每个粒子的适应度值。

(4)将粒子的 id 作为 PSO-map 函数的 key,将粒子的位置、适应度属性作为 value 输出到 PSO-Combine 函数。

(5)PSO-Combine 函数从粒子的适应度数组中找出最小适应度作为该粒子的个体极值,并记录当前位置。将粒子 id 作为 key,粒子位置和个体极值作为 value 输出到 PSO-Reduce 函数。

(6)PSO-Reduce 函数找出所有粒子的个体极值中最小的作为粒子群的全局极值。

(7)若通过群体适应度方差判断粒子群已趋向收敛或者循环达到最大迭代次数 T_{max},终止粒子群的迭代并将对应的聚类中心作为并行 k-means 算法的初值,否则转(2)继续迭代执行。

4.2 基于 Hadoop 的 k-means 算法的实现

基于 Hadoop 的 k-means 算法主要有两个阶段:Map 阶段和 Reduce 阶段,分别由 Map 函数和 Reduce 函数实现。

Map 阶段将 4.1 节中并行 PSO 算法输出的初始聚类中心作为 k-means 的初始聚类中心。

(1)计算每个记录点到聚类中心的距离,将该记录分配给距离最小的中心。

(2)将该记录所属聚类中心类别作为 key,将记录属性作为 value。

(3)写入中间结果到 Reduce 函数,context.write(key,value)。

Reduce 阶段使用 Reduce 函数首先会默认对 Map 函数写入的结果按 key 进行合并(combine),将 key 相同(即属于同一簇)的进行合并。

(1)计算每个簇的平均值。

(2)计算每个对象到各个簇中心的距离(即平均值),将它重新分配给最近的簇,更新聚类中心。

(3)计算准则函数,通常采用平方误差准则,若收敛,则转(4),否则转 Map 函数继续迭代执行。

(4)将不同的簇输出,簇编号作为 key,簇中每个对象及其属性作为 value 输出,输出结果就是聚类划分结果。

4.3 基于 Hadoop 的 PSO-kmeans 算法的实现

通过 4.1 和 4.2 节,并行的 PSO-kmeans 算法已经在 Hadoop 平台上实现了。具体思想如下:将待聚类数据集合 N 以及聚类数目 k 输入,在 PSO-Map 函数阶段:首先初始化粒子群的位置、速度,求出适应值,构建粒子群,然后将一个粒子作为一个 Map,将属性输出到 combine 函数,combine 函数将比较每个粒子的适应值,找出每个粒子的个体极值,将粒子编号和个体极值传送给 Reduce 函数;Reduce 函数通过比较找出全局极值,这个全局极值对应的粒子就是最优初始聚类中心;最后将初始聚类中心输入到并行 k-means 算法中,即可求出最佳聚类结果。整个基于 Hadoop 的 PSO-kmeans 算法的流程如图 2 所示。

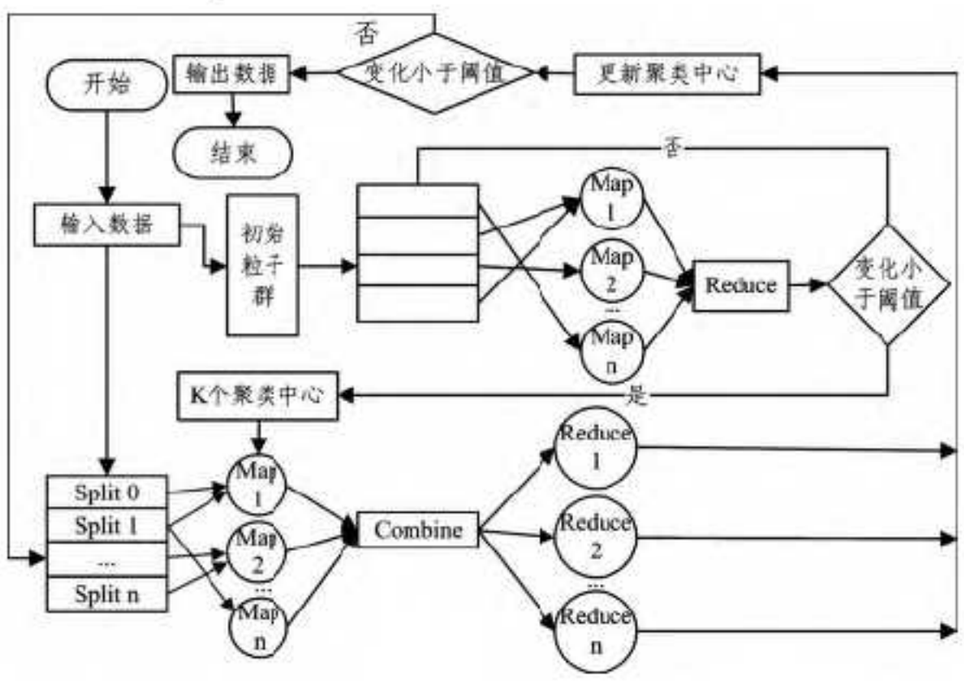

图 2 并行 PSO-kmeans 算法流程

5 实验和结果

5.1 实验环境

本实验搭配的 Hadoop 云平台是由 5 台 PC 机组成的,其中一台 PC 机作为 Master 主节点,也是 NameNode 和 JobTracker;其他 4 台作为 Slave 从节点,也是 DataNode 和 TaskTracker。5 台 PC 都是双核 2.4GHz CPU,2G 内存。每台 PC 系统均为 ubuntu 12.04 版本;hadoop-1.2.1;jdk1.7.0—67。具体集群分布如图 3 所示。

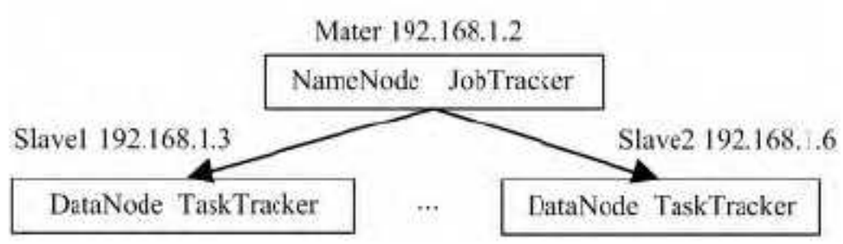

图 3 PC 集群分布

5.2 算法准确性实验

首先测试文中各种算法的聚类准确度。实验数据采用经过上文算法预处理过的 apache 服务器 Web 日志文件。将数据集分为 3 组数据,分别记作 A、B、C。其中 A 组包含 1000 条记录,属性维度为 3,数据大小为 11kB;B 组包含 5000 条记录,属性维度为 6,数据大小为 82kB;C 组包含 10000 条记录,属性维度为 10,数据大小为 234kB。其中每组数据测试 20 次,准确率取 20 次的平均值。在 PSO-kmeans 算法中设定粒子数为 30,最大迭代次数为 100。结果如表 2 所列。

表 2　算法准确率比较

准确率 / 算法名 / 数据集	串行 k-means	并行 k-mean	串行 POS-kmeans	并行 PSO-kmeans
A	89.34	89.68	91.67	90.76
B	74.54	76.42	82.45	85.48
C	86.32	87.43	89.23	93.23

由表 2 可知，对于 3 组测试数据，k-means 算法的聚类准确率一直最低，且实验中准确率波动比较大，这是由 k-means 算法对初始聚类中心的敏感性所决定的，实验结果表明 PSO-kmeans 算法的准确率是高于 k-means 算法的；而并行 PSO-kmeans 算法的准确率也比串行 PSO-kmeans 算法的准确率要高，且当数据量越来越大，属性维度越来越多时，并行效率明显高于串行。所以实验结果证明，本文改进的基于 Hadoop 的并行 PSO-kmeans 算法准确率是显著提高的。

5.3　算法效率实验

实验内容为比较 Hadoop 中只有 1 个运算节点对串行与并行 PSO-kmeans 算法在处理各种规模数据时所花费的时间，实验结果如表 3 所列，其中 T1 代表串行时间，T2 代表并行时间，均取 5 次重复运行时间的平均值。

表 3　算法时间对比

序号	文件大小	记录数	T1(s)	T2(s)
1	2M	122880	17.2	30.1
2	35.8M	2260992	38.2	102.4
3	420M	26098325	683.4	432.7
4	1G	64526612	1743.3	1057.3
5	4G	258206428	Can't handle	2236.8

由表 3 可知，当数据量较小的时候，并行 Hadoop 平台的处理效率不如串行的效率高，但当数据量越来越大的时候，并行 Hadoop 的处理效率逐渐高于串行算法。这是因为数据量较小时，Hadoop 需要不断地读取、写入、传输数据，实际计算时间占的比例十分有限[7]，所以串行效率高；但当数据量十分庞大的时候，单台系统资源将无法承受串行算法的开销，所以耗时很长，而并行 Hadoop 的优势就体现出来了。因此，实验证明 Hadoop 在对于大数据的处理方面是十分有优势的。

5.4　集群性能实验

本实验主要测试并行 PSO-kmeans 算法随着集群中节点个数的增加，处理效率的变化。实验进行加速比的测试。加速比是单节点运行时间和增加节点运行时间的比值。本实验采用属性维度都为 3 的 4 组数据，如表 4 所列，实验结果如图 4 所示。

表 4　实验数据

数据集编号	文件大小	数据个数	数据块
A	40M	2450455	1
B	120M	7591366	2
C	1G	64526612	16
D	2G	129065225	32

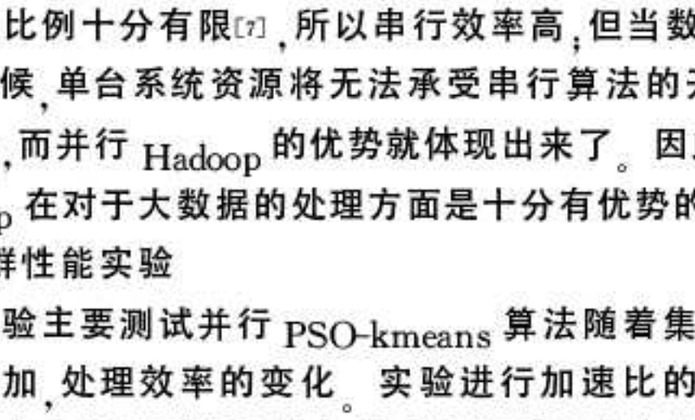

图 4　加速比

从图 4 可以看出，增加集群中的节点个数时，对于较小数据，算法的效率增加十分迅速，而对于较大数据，加速比与节点数呈正比例增长，但后期则趋向于稳定，体现了并行 Hadoop 平台时的稳定性和可扩张性。

结束语　针对 k-means 算法对初始聚类中心的依赖造成算法的不稳定性问题，以及串行算法的效率低下的现实问题，本文提出了基于并行 Hadoop 下的 PSO-kmeans 算法，并在 Hadoop 平台上用 MapReduce 编程框架加以实现。实验证明本文算法有效地提高了传统 k-means 算法聚类的准确性，并且相对于串行平台下，效率也有了很大的提高。对于如何从海量数据中快速准确挖掘到有价值的信息具有重要的现实意义。然而由于 Hadoop 本身不支持迭代计算，而 PSO-kmeans 算法需要多次迭代计算，文中算法只能把 Reduce 的结果写进 HDFS 文件，同时启动新的 job 时从 HDFS 中读文件，造成 I/O 压力，一定程度上影响效率。基于需求，Hadoop 应运而生，一个 job 中可以有多个 Map-Reduce 对，而且提供了判断接口，这样不需要再启动额外的任务来判断迭代是否结束。所以笔者下一阶段将着重研究基于 Hadoop 的 PSO-kmeans 算法，进一步提高其效率。

参 考 文 献

[1] 杨怡玲，管旭东，陆丽娜. 一个简单的 Web 日志挖掘系统[J]. 上海交通大学学报，2000，34(7)：35-37

[2] 孙玲芳，夏聪. Web 使用挖掘在用户行为分析中的应用[J]. 江苏科技大学学报：自然科学版，2011，25(3)：258-261

[3] 毛严奇，彭沛夫. 基于 MapReduce 的 Web 日志挖掘预处理[J]. 计算机与现代化，2013(9)：35-36

[4] Wang J，Su X. An improved K-Means clustering algorithm[C]// 2011 IEEE 3rd International Conference on Communication Software and Networks(ICCSN). IEEE，2011：44-46

[5] 吕奕清，林锦贤. 基于 MPI 的并行 PSO 混合 K 均值聚类算法[J]. 计算机应用，2011，31(2)：428-431

[6] 傅涛，孙亚民. 基于 PSO 的 K-means 算法及其在网络入侵检测中的应用[J]. 计算机科学，2011，38(5)：54-55

[7] 周婷，张君瑛，罗成. 基于 Hadoop 的 K-means 聚类算法的实现[J]. 计算机技术与发展，2013，23(7)：18-20

[8] 周诗慧，殷建. Hadoop 平台下的并行 Web 日志挖掘算法[J]. 计算机工程，2013，39(6)：43-46

[9] Shvachko K，Kuang H，Radia S，et al. The hadoop distributed file system[C]// 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies(MSST). IEEE，2010：1-10

[10] 宋莹，沈奇威，王晶. 基于 Hadoop 的 Web 日志预处理的设计与实现[J]. 电信工程技术与标准化，2011，24(11)：85-86

[11] 张晓强. MapReduce 在 Web 日志挖掘中的应用[D]. 成都：电子科技大学，2011

[12] 彭长生. 基于 Fisher 判别的分布式 K-Means 聚类算法[J]. 江苏大学学报：自然科学版，2014，4(35)：422-423

[13] Kennedy J，Eberhart R C. Particle swarm optimization [C]// Proceedings of IEEE international conference on neural networks. Perth：[s. n.]，1995：1942-1948

[14] 谢秀华，李陶深. 一种基于改进 PSO 的 K-means 优化聚类算法[J]. 计算机技术与发展，2014，24(2)：35-37

[15] McNabb A W，Monson C K，Seppi K D. Parallel pso using mapreduce[C] // IEEE Congress on Evolutionary Computation，2007(CEC 2007). IEEE，2007：7-14