文章编号:1671-4598(2016)07-0272-04 DOI:10.16526/j.cnki.11—4762/tp.2016.07.073 中图分类号:TP311 文献标识码:A

# 基于 MapReduce 的 K-means 聚类算法的优化

孙玉强，李媛媛，陆 勇
(常州大学 信息科学与工程学院，江苏 常州 213164)

**摘要**：针对传统的聚类算法 K-means 对初始中心点的选择非常依赖，容易产生局部最优而非全局最优的聚类结果，同时难以满足人们对海量数据进行处理的需求等缺陷，提出了一种基于 MapReduce 的改进 K-means 聚类算法。该算法结合系统抽样方法得到具有代表性的样本集来代替海量数据集；采用密度法和最大最小距离法得到优化的初始聚类中心点；再利用 Canopy 算法得到粗略的聚类以降低运算的规模；最后用顺序组合 MapReduce 编程模型的思想实现了算法的并行化扩展，使之能够充分利用集群的计算和存储能力，从而适应海量数据的应用场景；文中对该改进算法和传统聚类算法进行了比较，比较结果证明其性能优于后者；这表明该改进算法降低了对初始聚类中心的依赖，提高了聚类的准确性，减少了聚类的迭代次数，降低了聚类的时间，而且在处理海量数据时表现出较大的性能优势。

**关键词**：K 均值算法；抽样；Canopy 算法；最大最小距离法

# Optimization of K-means Clustering Algorithm Based on MapReduce

Sun Yuqiang, Li Yuanyuan, Lu Yong
(School of Information Science&Engineering, ChangZhou University, Changzhou 213164, China)

**Abstract**: To deal with the problems that traditional K-means clustering algorithm is very dependent on the selection of the initial points, being prone to clustering result of local optimum rather than global optimum, and it is difficult to meet the need of dealing with massive amounts of data, an improved K-means clustering algorithm based on MapReduce is proposed. The algorithm combines systematic sampling method to get a representative sample set which is used to replace the massive data set; and uses density method and Max-Min distance method to get the optimal initial clustering centers; and adopts Canopy algorithm to get a rough clustering which can reduce the computational scale; and finally employs the idea of sequential composition of MapReduce programming model to realize the parallel extension of the algorithm, which can make full use of the computing and storage capacity of the cluster, in order to adapt to the application of massive data. The improved algorithm is compared with the traditional clustering algorithms in this paper, and the comparative results show that the performance of improved algorithm is better than the latter. The experiments show that the improved method reduces the dependence on the initial cluster centers and also reduces the number of iterations of clustering and the clustering time. Furthermore it shows greater performance advantage in dealing with massive data.

**Keywords**: K-means clustering algorithm; sampling; Canopy algorithm; Max-Min distance method

## 0 引言

随着数据规模的爆炸性增长，传统的数据挖掘工作在处理大规模数据时会出现用时过长、存储量不足等缺点。云计算的提出将这些问题迎刃而解，它是一种基于互联网的计算，是分布式计算、并行处理和网格计算的进一步发展。由 Google 提出的 MapReduce 编程框架[1]是云计算中代表性的技术，主要用于海量数据的并行计算。已有多种数据挖掘算法在 MapReduce 计算模型的基础上被实现了，而聚类是在数据挖掘中运用比较广泛的一种算法。

K-means 算法是一种典型的基于划分的聚类算法[2]，该算法简单高效，运用于科学研究的各个领域。但是它对初始聚类中心的选择非常敏感，不能有效地对大规模数据进行聚类处理。针对 K-means 算法的特点以及不足，很多学者提出了相应的改进 K-means 算法。邓海等人[3]提出了基于最近高密度点间的垂直中心点优化初始聚类中心的 K-means 聚类算法；王玲等人[4]提出一种密度敏感的相似度度量的方法；马帅等人[5]提出了一种基于参考点和密度的快速聚类算法。这些研究通过不同的方法解决了算法对初始中心敏感的问题，但却增加了算法的复杂度。其他的例如毛典辉等人[6]提出了改进的 Canopy 和 K-means 相结合的算法；虞倩倩[7]等人提出了聚类的蚁群优化算法；马汉达等人[8]提出了基于粒子群算法(PSO)的 K-means 算法；贾瑞玉等人[9]提出了并行遗传 K-means 聚类算法，都是结合一些成熟的算法来改进 K-means 聚类算法，虽然在少量数据环境下能够显著提升 K-means 算法的性能，但却因为额外的组件增加了算法的复杂性，无法适应海量数据的处理需求。因此，本文提出了一种基于 MapReduce 的高效 K-means 并行算法，其结合了抽样方法进行预处理以减少数据量；通过密度思想对初始中心点优化提高聚类准确性；结合其他算法降低计算规模；再通过 MapReduce 分布式并行模型提高运行效率。实现的过程主要有 4 个子任务：1)抽取样本；2)计算样本初始聚类中心点；3)Canopy 划分；4)K-means 迭代。通过实验结果得出，此改进算法在处理海量数据时可以体现出更大的优势。

**收稿日期**:2016-01-19； **修回日期**:2016-02-29。
**基金项目**:国家自然科学基金项目(11271057,51176016);江苏省自然科学基金项目(BK2009535)。
**作者简介**:李媛媛(1991-),女,江苏盐城人,硕士研究生,主要从并行计算、数据挖掘等方向的研究。
孙玉强(1956-),男,河南人,教授,硕士研究生导师,主要从事并行计算、软件工程等方向的研究。

## 1　背景

### 1.1　MapReduce 简介

MapReduce 是一种分布式并行编程模型，数据被存储在分布式文件系统（distributed file system，DFS）中，以键－值对＜key，value＞的形式来表示数据。MapReduce 的具体步骤是：先将数据切割成 Split，对 Split 又分成一批键值对＜K1，V1＞传给 Map 函数；Map 函数对每个键值对进行处理后形成新的键值对＜ K2，V2＞，并把 K2 值相同的进行汇总，输出中间结果＜ K2，list＜V2＞＞；Reduce 函数对 Map 函数输出的中间结果做相应的处理后得到键值对＜K3，V3＞。MapReduce 的操作流程如图 1 所示[10]。

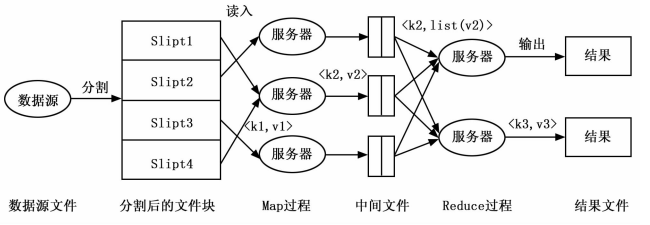


图 1　Mapreduce 的操作流程图

### 1.2　密度思想

在一个大规模的数据对象空间中，低密度区域的数据对象点一般划分着高密度区域的数据对象点，通常所说的噪声点或孤立点就是这些低密度区域的点。如果在进行聚类分析时，这些噪声点或孤立点被选为初始聚类中心，就会造成聚类不理想的结果。因此，初始聚类中心的选取应该集中到高密度区域以排除孤立点的干扰。为了有效地排除数据样本中存在的噪声点对象和孤立点对象，可以通过引入高密度点的概念来处理。

定义 1 高密度点：取数据集中的数据对象 $x_i$ 为中心点，计算此中心点在邻域半径 $\delta$ 内包含的样本个数，若样本个数不少于常数 Minds，则该中心点就被称为高密度点。其中 $\delta$ 的取值需结合输入的数据对象集，将其取为 $n$ 个数据样本间距离均方根的一半，公式如下：

$$\delta = \frac{1}{2}\sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}\| x_i - x_j \|^2} \tag{1}$$

其中：$\| x_i - x_j \|$ 为公式（2）的欧式距离 $D_{(i,j)}$。

### 1.3　最大最小原则

为了避免初值选取时出现的聚类中心过于邻近的情况，最大最小原则可以用来选取尽可能远的数据对象作为初始聚类中心。“最大最小原则”[11]的算法描述如下：

1）从 $n$ 个数据集 $\{X_1, X_2, \cdots X_n\}$ 中随机选取一个样本作为第一个初始聚类中心点 $C_1$；

2）在剩余的数据对象集中找到距离 $C_1$ 最远的点，作为第二个初始聚类中心点 $C_2$；

3）分别计算剩余的数据对象集到已有的 $i$ 个聚类中心点之间的距离，取最小距离的最大者：$D_{(i+1)} = max\{min\{D_{(j,1)}, D_{(j,2)}, \cdots, D_{(j,i)}\}\}$，$j=1, 2, \cdots, n$，则第 $i+1$ 个初始聚类中心点为 $C_{i+1}=X_j$；

4）用来表示 $D_{(i+1)}$ 变化幅度的深度指标 $Depth$（$k$）可以通过边界识别的思想进行引入，其定义为：$Depth$（$k$）＝｜$D_{(k)}$－ $D_{(k-1)}$｜＋ ｜$D_{(k+1)}$－$D_{(k)}$｜，当深度值 Depth（k）取得最大值时，需求解的最优初始聚类中心点就为前 k 个记录值，同时下一轮 Canopy 的区域半径可设为 $T_1 = D_{(k)}$，避免了 Canopy 算法初始值设置的盲目性和随机性。

### 1.4　Canopy 算法

Canopy 算法[12]的思想是使用一种代价低的相似性度量方法，快速粗略地将数据划分成若干个重叠子集，每个子集可以看成是一个簇。Canopy 算法基本流程[13]如下：

1）设置阈值 $T_1$，$T_2$，其中 $T_1 > T_2$；

2）从数据集中取得一个数据对象，构建初始 Canopy，并从该数据集中删除该数据对象；

3）从剩余的数据集中取出一个数据对象，

计算其与所的有 Canopy 中心之间的距离，如果该数据对象与某个 Canopy 中心的距离在 $T_1$ 内，则将该对象加入到这个 Canopy 中，如果该对象与 Canopy 中的某个 Canopy 中心的距离小于 $T_2$，则当该数据对象与所有 Canopy 中心的距离计算完成后，将其从数据集中删除，如果该数据对象没有加入到任何 Canopy，则构建一个新的 Canopy；

4）重复步骤 3），直到数据集为空。

## 2　K-means 算法介绍和分析

### 2.1　K-means 算法思想

K-means 算法流程[14]：1）首先从 $n$ 个数据对象中随机选取 $k$ 个初始聚类中心点；2）对于剩下的其它数据对象，分别求它们与这 $k$ 个聚类中心的欧式距离（相似度），将它们分配给与其欧式距离最小（相似度最大）的聚类；3）计算新的聚类中心（聚类中所有对象的均值）；4）不断重复步骤 2）和 3）这一过程直到准则函数开始收敛为止。

1）欧式距离公式如下：

$$D_{(i,j)} = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2} \tag{2}$$

其中：$i = (x_{i1}, x_{i2}, \ldots, x_{ip})$；$j = (x_{j1}, x_{j2}, \ldots, x_{jp})$；它们分别表示一个 $p$－维数据对象。

2）均方差准则函数

$$E = \sum_{i=1}^{k}\sum_{p\in c_i} |p - m_i|^2 \tag{3}$$

其中：$E$ 为数据集中所有数据的均方差之和；$p$ 为数据对象空间中的一个点；$m_i$ 为聚类 $C_i$ 的均值（$p$ 和 $m_i$ 都是多维的）。

### 2.2　K-means 算法的不足及其改进

1）传统 K-means 算法不能有效处理大规模数据的聚类操作。而基于 MapReduce 的分布式并行计算模型，将一个大型数据切分成多个小数据模块的形式，分配给多台计算机集群进行分布式计算，与传统的单机运行平台相比，加快了总体的运行速率，减少了执行时间。

2）传统的 K-means 算法通过完全随机的策略对初始聚类中心点进行选取，不能确保聚类结果的准确性，更不能有效处理海量数据。可以通过抽样方法对初始数据进行预处理操作，得出有效样本集合。再结合密度思想和最大最小原则求得样本集的初始聚类中心。在高密度区域选择初始中心点可以排除噪声点和孤立点的干扰，而最大最小原则可以避免在高密度区域选取的中心点过于邻近。

3）传统的 K-means 算法在比较数据对象与聚类中心的距离时，计算了全局对象与中心点之间一一对应的距离，虽然在一定程度上确保了聚类的效果，但是总体上降低了算法的效率，而且不能适应大规模数据的聚类处理。对此，引入了 Canopy（华盖）的思想，每次只比较落在同一 Canopy 内的对象与相应中心点之间的距离，通过减少比较次数大大降低整个聚类的运行时间[13]。实验结果表明，该策略在处理大规模数据时，与传统方法相比，其收敛速度更快。

## 3 改进 K-means 算法的并行化

改进的 K-means 算法思想：首先采取系统抽样，得到一个具有较少数目并能代表全局数据对象的样本集合；再运用密度法和最大最小距离法相结合的方法求得样本集合的最佳初始聚类中心；最后利用 Canopy 算法对全局数据对象进行粗略划分，以改进 K-means 算法。一方面，该算法可以通过获得优化的初始聚类中心来提高聚类结果的精确度；另一方面，该算法能够智能的降低数据比较的次数。最后，将改进后的算法运用于 MapReduce 的并行计算模型中，其基本流程如图 2 所示。

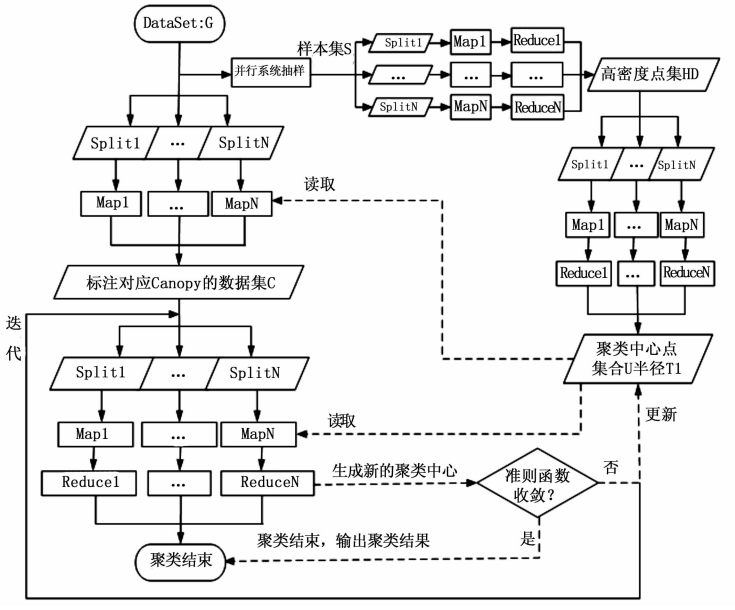


图 2 基于 Mapreduce 的改进 K-means 流程图

### 3.1 抽取样本

系统抽样主要将在 HDFS 上存储的 split 文件块进行规则设计与文件读取，设每个部分抽样概率为 $p_x$，则我们首先在 Map 阶段对每个文件块的数据量大小以及文件行数进行预估，顺序读取文件的前 $x\%$行，并指定为相同的 key 值传到同一个 Reducer 上，在 Reduce 阶段对每个块的样本数据合并，得到最终的抽样结果。

指定参数 $0<\varepsilon<1$ 来控制样本大小，设 $n$ 为数据集大小，则通过概率 $p_x=1/(n\varepsilon^2)$ 对整个数据集进行系统抽样，可以得到均匀的样本。算法 1 描述了利用 MapReduce 模型进行系统抽样（RS）的过程。

算法 1：系统抽样算法。

输入：初始数据集 $G$；

输出：样本集 $S$。

Map 函数：

1）每个 Map 读取数据 $G$ 和抽样概率 $p_x$，利用 $p_x$ 进行抽样；

2）将所有数据作为 value，为它们指定相同的 key，形成中间键值对；

Reduce 函数：

1）将中间键值对 shuffle 到同一 Reducer 上；

2）输出样本数据集 $S$；

### 3.2 计算样本初始聚类中心点

3.2.1 计算高密度数据集

算法 2：高密度点生成的并行化。

输入：样本集 $S$。

输出：高密度点集 $HD$。

Map 函数：

1）每个 Map 读取样本集 $S$，计算每个样本点 $xi$ 与其余样本点之间的距离 $D_{(i,j)}$；

Reduce 函数：

（1）根据公式（3）计算邻域半径 $\delta$，并统计到每个样本点 $x_i$ 距离小于 $\delta$ 的样本个数；

2）读取密度阈值 Minds，若个数大于 Minds，则把该点定义为高密度点，得到高密度点集合 $HD$。

3.2.2 计算优化初始中心点

算法 3：样本中心点生成算法。

Map 函数：

输入：高密度点集 $HD$；

输出：中心点的初始集合 $Q$。

（1）设置初始中心点的集合，$Q=null$；

（2）设置迭代次数，While $k<$sqrt（$N$）（当前节点数据规模 $N$）；

（3）$if$ 集合 $Q$ 为空，求数据集 $V$（当前节点数据集 $V$）中密度最大点，并保存该点至集合 $Q$；$else$ 求数据集 $V$ 中数据点与集合 $Q$ 中数据点的距离最小值中最大者，并保存该点至集合 $Q$；

Reduce 函数：

输入：各节点中心点的初始集合 $Q=\{Q_1, Q_2, \cdots, Q_n\}$；

输出：样本中心点的最终集合 $U$ 以及半径 $T_1$；

（1）求取集合 $Q$ 的数据总量，$N=$Count（$Q$）；

（2）设置迭代次数，While k< sqrt（N）；

（3）求数据集中数据点之间距离最小值中最大者，并保存该点至集合 $Q'$；

（4）求取集合 $Q'$的数据量 $m=$Count（$Q'$）；

（5）while $j<m$，求取集合 $Q'$中 Depth（$k$）最大值并输出 $T_1$，同时将集合 $Q'$前 $k$ 个点赋值至集合 $U$。

### 3.3 Canopy 划分

算法 4：Canopy 划分算法。

Map 函数：

输入：初始数据集 $G$；

输出：标注对应 Canopy 的数据集 $C$。

（1）读取样本中心点的最终集合 $U$ 以及半径 $T_1$，计算初始数据集与样本中心点之间的距离 $D$；

（2）当 $D<=T_1$，则标注该数据点属于对应的 Canopy，并将结果输出。

### 3.4 K-means 迭代

该阶段的主要任务是进行精确的聚类计算，其算法流程为：

算法 5：K-Means 算法。

输入：标注对应 Canopy 的数据集 $C$；

输出：$K$ 中心点集合 $U'$。

Map 函数：

1）读取样本中心点的最终集合 $U$，将样本中心点的最终集合 $U$ 作为初始 $K$ 中心点集合，$U'=U$；

2）通过 Map 函数比较已标注的输入数据点与对应中心点的距离，输出当前数据点及对应最近距离的中心点编号；

Reduce 函数：

1）通过 Reduce 函数将同一中心点下的数据点归并，计算均值并将结果输出作为新的中心点；

2）判断准则函数是否收敛，若收敛则终止程序，否则进行下一轮 MapReduce 的迭代。

## 4 实验结果

本实验在 Hadoop 分布式集群平台上对多组测试数据进行了仿真测试[15]，传统的 K-means 算法 $K$：任意选取 $k$ 个聚类中心点后再用 K-means 算法进行聚类；改进的 Canopy-Kmeans 算法 CK：首先结合最大最小原则计算出初始 k 个 Canopy 中心点，利用各个 Canopy 中心点对数据集进行粗略地 Canopy 聚类，再结合 K-means 算法对每个 Canopy 类进行精确地聚类；高效的 DM Canopy-Kmeans（Canopy-Kmeans Clustering Algorithm based on Density Method and Max-min Distance Algorithm）算法 DMCK：1）结合密度法和最大最小距离法计算已有的抽样数据对象的 $k$ 个初聚类中心 $C$；2）利用样本聚类中心 $C$ 作为全局数据对象的初始聚类中心，结合 Canopy 算法进行 Canopy 划分；3）再利用 K-means 算法计算出最终的聚类结果。数据集的记录数为 $n$，聚类个数为 $k$，具体算法为 $M$，迭代次数为 $t$，算法执行时间为 $T$。

实验一：验证改进的 K-means 算法的可行性。

实验说明：采用 50 条测试数据进行测试，运行结果见表 1。

表 1　不同聚类算法和不同聚类个数的聚类结果

| $k$ | $M$ | $t$ | $T$ |
|---|---|---|---|
| 5 | K | 3.5 | 29.11 |
| | CK | 2.4 | 22.62 |
| | DMCK | 2.5 | 22.89 |
| 6 | K | 3.0 | 22.21 |
| | CK | 2.4 | 21.92 |
| | DMCK | 2.9 | 22.23 |
| 7 | K | 2.4 | 22.03 |
| | CK | 2.5 | 21.78 |
| | DMCK | 2.5 | 22.19 |

从表 1 可以得出以下结论，改进的 Canopy-Kmeans 算法运行的聚类结果与高效的 DM Canopy-Kmeans 算法的聚类结果相比，其迭代次数和运行时间取得了相似的结果，而且保持了稳定性；且改进的 Canopy-Kmeans 算法与传统的随机选择聚类中心的 K-means 算法相比，表现出了更好的性能优势。但在处理海量数据时，若用改进 Canopy-Kmeans 算法，聚类迭代次数会增加，甚至会造成内存不足。所以提出了这种折中的用样本数据初始聚类中心代替全局数据初始聚类中心的聚类算法。

实验二：验证改进的 K-means 算法的有效性。

实验说明：用随机产生的记录数来验证方法的有效性，记录数 $n$（单位：万）分别是 1、10、100，环境：单机伪分布条件下，聚类方法同上，聚类个数为 100 时结果见表 2。

表 2　单机下的聚类结果

| $n$ | $M$ | $t$ | $T$ |
|---|---|---|---|
| 1 | K | 6.3 | 42.81 |
| | CK | 4.6 | 83.62 |
| | DMCK | 3.5 | 40.35 |
| 10 | K | 8.2 | 106.79 |
| | CK | 3.0 | 428.94 |
| | DMCK | 2.6 | 36.57 |
| 100 | K | 7.2 | 1 132.51 |
| | CK | 时间过长 | 时间过长 |
| | DMCK | 2.5 | 581.18 |

通过表 2 得出以下结论：改进的 Canopy-Kmeans 聚类算法在用最大最小原则计算初始聚类中心点时，随着处理的数据规模的扩大，迭代时间也会随之增长，会导致算法效率变低。当处理的数据量不断增加时，这种改进的聚类算法将不再适用于初始中心点的计算。而提出的高效 DM Canopy-Kmeans 算法在面对海量数据量时，大大减少了执行的时间，提高了算法的运行效率和聚类的精度。

实验三：验证改进算法可以并行执行。

实验说明：4 台均是装有 Centos5 操作系统的虚拟机，其中一台是 master，剩余三台是 slave，内存 512 M，硬盘 100G，2.5Ghz 双核 CPU。数据：使用实验 2 中数据，在集群平台下聚类为 100 时的运行结果见表 3。

表 3　集群下运行结果

| $n$ | $M$ | $t$ | $T$ |
|---|---|---|---|
| 1 | K | 6.2 | 49.18 |
| | CK | 2.4 | 50.82 |
| | DMCK | 2.0 | 17.53 |
| 10 | K | 8.2 | 84.18 |
| | CK | 2.5 | 352.83 |
| | DMCK | 2.1 | 31.15 |
| 100 | K | 6.8 | 985.54 |
| | CK | 4.3 | 4 126.72 |
| | DMCK | 3.2 | 536.62 |

表 3 与表 2 对比得出以下结论，在同样的条件下，并行化与串行相比，其操作时间明显是降低的，且提高了算法的运行效率，进一步体现了海量数据在集群环境下处理的优势。

## 5 结束语

在处理海量数据时，为减少 k-means 算法对初值的依赖性，

（下转第 279 页）

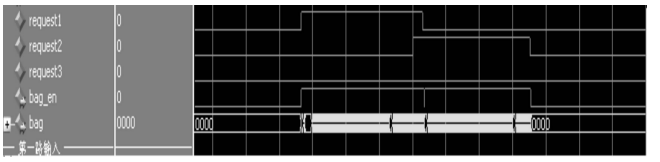


图 7　高优先级抢占低优先级传输

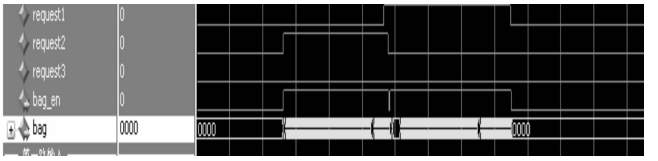


图 8　低优先级抢占高优先级

进行数据包的发送，而在不同通道包发送过程中，程序不会去响应其他通道发送请求，能够达到设想的效果。

最终的发送请求和传输细节可由图 9 看出。

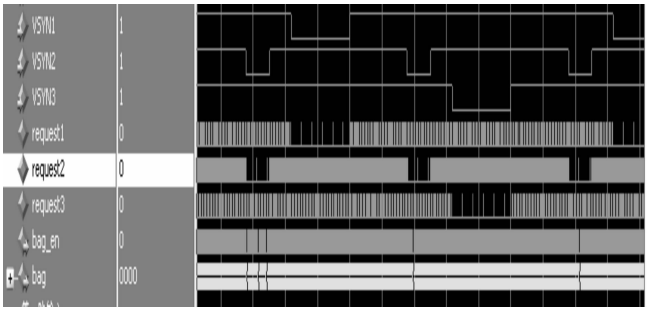


图 9　信道发送综合效果图

在图中，当视频信号处于消隐期的时候，模块依然可以通过内部定时产生等间隔的超时发送请求，满足实时传输的要求。而且由请求信号和 bag _ en 信号的疏密可知，3 路视频已经完整的复用在了最终的数据流 bag 中，达到了实时复用的目的。

由仿真结果可以看出，数据编码、时分复用打包模块达到了设计的效果，整个多数据单信道复用的传输方法算法有效可行，达到了预期的目的。

## 4　结论

本文所设计的多路数据单信道传输时分复用模块，很好地解决了多路视频同时传输的问题。尤其是发送请求模块，结合时间轮片和基于优先级的调度算法，在满足各通道实时传输的条件下，为尽可能多的时分复用信道提供一种可靠方法，理论上可以复用接近一个串行通道的传输极限，对于 2.5Gbps 的信道，最多可以复用接近 2Gbps 的传输数据，是一种经济、实用的图像数据传输方式，可以应用于多路视频同时传输系统。

**参考文献：**

[1] 吴　贺. 多通道数字光电转换系统的设计与实现 [D]. 西安：西安电子科技大学，2013.

[2] 肖儿良，韦荣明，颜文超. 基于 Aurora 及 CameraLink 的高速数字图像传输 [J]. 信息技术，2015 (4)：94 - 97.

[3] 姚七栋，张春玉. CRC 校验及其软件实现 [J]. 现代电子技术，2006 (13)：67 - 68.

[4] 陈忠平，高金定，高见芳. 基于 Quartus II 的 FPGA/CPLD 设计与实践 [M]. 北京：电子工业出版社，2010.

[5] 王　诚，蔡海宁，吴继华. Altera FPGA/CPLD 设计（基础篇）[M]. 北京：人民邮电出版社，2011.

(上接第 275 页)

我们详细探讨了初始聚类中心的优化选择问题，并提出全新的 DM Canopy-Kmeans 算法。为提高算法的运行效率，在 MapReduce 的平台下对改进聚类算法进行实现。并通过实验对其进行对比分析，验证了基于并行模型下的改进算法更优于传统的算法，提高了算法的执行效率。下一步的主要工作是进一步改进抽样数据的样本质量，可以用来更精确地代表全局数据对象。

**参考文献：**

[1] Dean J，Ghemawat S. MapReduce：simplified data processing on large clusters [J]. Communications of the ACM，2008，51 (1)：107 - 113.

[2] Han J，Kamber M. Data Mining：Concepts and Techniques [M]. Fan Ming，Meng Xiaofeng，translation. 2 edition. Beijing：Mechanical Industry Press，2007.

[3] 邓　海，覃　华，孙　欣. 一种优化初始中心的 K-means 聚类算法 [J]. 计算机技术与发展，2013，23 (11)：42 - 45.

[4] 王　玲，薄列峰，焦李成. 密度敏感的普聚类 [J]. 电子学报，2007，35 (8)：1577 - 1581.

[5] 马　帅，王腾蛟，唐世渭，等. 一种基于参考点和密度的快速聚类算法 [J]. 软件学报，2003，14 (6)：1089 - 1095.

[6] 毛典辉. 基于 MapReduce 的 Canopy-Kmeans 改进算法 [J]. 计算机工程与应用，2012，48 (27)：22 - 26.

[7] 虞倩倩，戴月明，李晶晶. 基于 MapReduce 的 ACO-K-means 并行聚类算法 [J]. 计算机工程与应用，2013，49 (16)：117 - 120.

[8] 马汉达，郝晓宇，马仁庆. 基于 Hadoop 的并行 PSO-kmeans 算法实现 Web 日志挖掘 [J]. 计算机科学，2015，42 (6A)：470 - 473.

[9] 贾瑞玉，管玉勇，李亚龙. 基于 MapReduce 模型的 K-means 并行遗传聚类算法 [J]. 计算机工程与设计，2014，35 (2)：657 - 660.

[10] Isard M，Budiu M，Yu Y，et al. Dryad：Distributed data-parallel programs from sequential building blocks [A]. Proc. of the 2nd European Conf. on Computer Systems (EuroSys) [C]. 2007：59 - 72.

[11] 刘远超，王晓龙，刘秉权. 一种改进的 K-means 文档聚类初值选择算法 [J]. 高技术通讯，2006 (1)：11 - 15.

[12] 孙吉贵，刘　杰，赵连宇. 聚类算法研究 [J]. 软件学报，2008，19 (1)：48 - 61.

[13] 李应安. 基于 MapReduce 的聚类算法的并行化研究 [D]. 广州：中山大学，2010.

[14] Han Jiawei. Kamber. Data mining：Concepts and techniques [M]. Beijing：Mechanical Industry Press. 2008：288 - 375 (in Chinese).

[15] Apache. Hadoop [EB/OL]. [2012 - 10 - 10]. http：//hadoop.apache. org.