结合大数据流特征和改进 SOM 聚类的资源动态分配算法

项丽萍[1] 杨红菊[2]

[1](晋城职业技术学院信息工程系 山西 晋城 048000)

[2](山西大学计算机与信息技术学院 山西 太原 030006)

摘 要 在大数据流中,由于数据特征的未知性,如何分配数据资源是一个难题。为了解决这个问题,提出一种大数据环境下基于数据特征预测和改进自组织映射 SOM(Self-organizing maps)的资源管理算法。根据数据的体积和速度变化,通过自回归模型对下一时间间隔到达的数据的特征进行估计,估计值用数据特征(CoD)向量表示;利用粒子群优化 PSO 算法来优化 SOM 算法的权重分布,形成改进型 SOM 算法,对 CoD 向量进行聚类,动态创建和分配云资源集群。这些集群以拓扑排序的方式创建,集群之间的联系越多,它们的排序越接近,利用这种拓扑排序来减少等待时间。实验结果表明,该算法能准确预测数据特征,有效提高了云资源的利用率。

关键词 大数据流 云计算 粒子群优化 自组织映射 数据特征 资源管理

中图分类号 TP391 **文献标识码** A DOI: 10.3969/j.issn.1000-386x.2019.05.045

DYNAMIC RESOURCE ALLOCATION ALGORITHM BASED ON BIG DATA STREAM CHARACTERISTIC AND IMPROVED SOM CLUSTERING

Xiang Liping[1] Yang Hongju[2]

[1](*Department of Information Engineering, Jincheng Institute of Technology, Jincheng 048000, Shanxi, China*)

[2](*School of Computer and Information, Shanxi University, Taiyuan 030006, Shanxi, China*)

Abstract In big data streams, it is a difficult problem to allocate data resources due to the unknown characteristics of data. To solve this problem, we proposed a resource management algorithm based on data characteristic prediction and improved self-organizing mapping (SOM) in big data environment. We estimated the data characteristics arriving at the next time interval by the autoregressive model according to the volume and velocity of the data, and the estimated value was represented by the characteristics of data (CoD) vector. Then we used the particle swarm optimization (PSO) to optimize the weight distribution of SOM algorithm, and an improved SOM algorithm was formed to cluster the CoD vectors, so as to dynamically create and allocate cloud resource clusters. These clusters were created in a topological sort. The more connections between clusters, the closer their sort was, so using this topological sort could reduce waiting time. The experimental results show that the proposed algorithm can not only accurately predict the data characteristics, but also effectively improve the utilization of cloud resources.

Keywords Big data stream Cloud computing Particle swarm optimization Self-organizing map Characteristics of data Resource management

0 引 言

现阶段,物联网和人们的生活越来越密切,物联网设备可以随机生成数据,这种随机性会导致具有未知特性的大数据流的产生[1]。这些大数据通常由多样性的图像、视频、音频和文本等数据组成。大数据流的数据特性包括 4V:体积(Volume)、速度(Velocity)、类型(Variety)和可变性(Variability)。

随着大数据流的增加,如何实时分析大数据是一

收稿日期:2018-11-03。国家自然科学基金项目(61873153)。项丽萍,副教授,主研领域:大数据,云计算。杨红菊,副教授。

个难题。使用最多的是基于云计算的大数据分析法，通过选择合适的云资源来分析数据特性已成为热门研究问题[2-3]。文献[4]强调了动态资源配置是大数据应用程序调度中的一个具有挑战性的问题。文献[5]提出了一个基于 QoS 的框架实现大数据中的最优资源分配。该框架的功能和 QoS 要求由用户提供。功能要求包括处理能力、GPU 功率、RAM 和输入数据的大小等; QoS 要求包括响应时间、输出数据质量和结果可视化。根据功能和 QoS 要求，使用朴素贝叶斯算法确定所需云群。文献[6]提出一种基于图的大数据资源处理方法，可以在不影响响应时间的情况下提高数据处理效率。文献[7]提出了一种基于优先级的多媒体数据处理方法，该方法是静态混合算法。文献[8]使用马尔可夫链和分配的节点有效地预测了大数据的大小以进行数据处理。但是该模型没有考虑大数据资源分配的高速性、多样性和可变性。文献[9]强调云数据中心的可变性会影响云资源的分配。文献[10]表明了预测大数据请求的速度对有效配置云资源至关重要。以上研究表明，流式大数据的 4V 特性是云资源分配的重要参数。

因此，本文提出了一种大数据环境下结合数据特征预测和智能聚类的资源管理算法。其主要创新点在于: (1) 根据现有数据的体积、速度的变化情况，对下一时刻数据的特征进行估计。(2) 利用自组织映射 SOM 和粒子群优化 PSO 相结合，构建一种先进的聚类算法，根据估计的数据特征对数据进行聚类，用来分配合理资源。

1　算法流程

本文提出的算法旨在基于数据特性(4V)为大数据流分配适当的资源。为了实现该目标，算法分为两步，如图 1 所示。

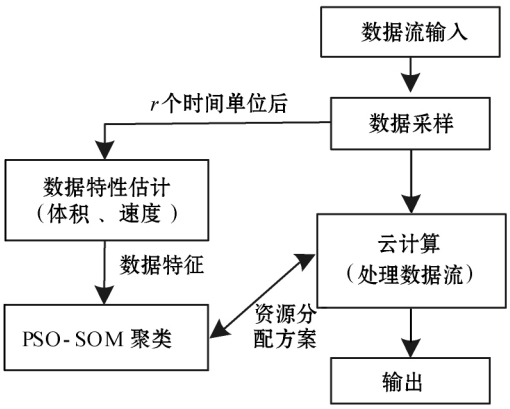

图 1　本文算法框架

1) 首先从输入流中提取小块数据，块大小是底层存储系统(例如 HDFS 的默认块大小为 64 MB)的一个整数倍，这样可减少读取和分析数据的开销。分析所选数据块，根据现有数据特征来估计下一时刻数据流的体积和速度。在估计这些值时，会考虑数据的可变性。估计值用数据特征(CoD)向量表示。

2) 利用 PSO 优化 SOM 算法的权值分布，构建一种改进型 SOM 算法，用于对 CoD 向量进行聚类，动态创建和分配空闲的云资源集群。

在 r 个时间单位之后，使用自适应采样技术[11]选择另一个块，其中，r 取决于采样率。这种机制允许算法在早期捕获新到达数据流的变化性，从而分配适当的集群。接着，算法自动调整采样率大小，这个过程中涉及到迁移开销[12]。如果迁移开销很高，r 的值会增加，如果执行时间很长，则 r 的值会降低。对采样的数据块再次进行分析，并使用上述步骤分配资源。

2　数据特性的估计

为了估计数据特征，通过四个阶段进行操作: 第一阶段，确定数据类型; 第二阶段，使用数据可变性估计数据的体积和速度; 第三阶段，计算相对体积和速度，即将这些估计值与其他到达云端的数据请求进行比较，并计算相对的体积和速度; 第四阶段，相对值的有效表示。下面给出详细的步骤。

2.1　数据类型确定

目前，数据分析主要有四种类型[13]，分别是文本分析、图像分析、音频分析和视频分析。本文使用 Bloom 过滤器[14]来确定数据的类型。Bloom 过滤器的组成包括: (1) 一个 n 位的序列，初始值均为 0; (2) k 个哈希函数的集合 $h_1, h_2, \cdots, h_k$; (3) m 个键值组成的集合 S。每个哈希函数将键值映射到 n 个块中，对应于 n 个数组。Bloom 过滤器的目的是允许 S 中的流元素通过，拒绝其他元素。

令 p 表示可接受的假阳性率，则 n 和 k 的值可由式(1)和式(2)计算得到:

$$n = \mathrm{ceil}\left(\frac{-m \times \ln(p)}{[\ln(2)]^2}\right) \tag{1}$$

$$k = \mathrm{round}\left(\ln(2) \times \frac{n}{m}\right) \tag{2}$$

Bloom 过滤器的工作流程如算法 1 所示。

算法 1　Bloom 过滤器的工作流程

输入: 数组 $BF[n]$，集合 S，哈希函数 $h_1, h_2, \cdots, h_k$

输出: 特定类型的数据 e

1. 初始化 $BF[1], \cdots, BF[n] = 0$; 2. 对于每个键值 $y_i \in S$
   计算 $h_1(y_i), \cdots, h_k(y_i)$;
   设置 $BF[h_j(y_i)] = 1, 1 \leqslant j \leqslant k$;
3. 对每个经过过滤器的流元素 e

计算 $h_1(e)$,…,$h_k(e)$;

对于所有 j,如果 $BF[h_j(e)]=1$

允许 e 通过滤波器并输出;

4. 退出

在本文所提出的算法中,使用四个 Bloom 过滤器。第一个过滤器仅允许图像类数据通过,并阻挡其他类型的数据。类似地,第二、第三和第四过滤器分别允许音频、视频和文本数据通过。

这里,一个特定 Bloom 过滤器的集合 S 由所有可能的数据格式组成。例如,在第一个过滤器中设置 S 包含所有图像格式,如 jpeg、png 等。第二个过滤器中的集合 S 由所有音频格式组成,如 mp3、wav、aiff 等。用同样的方式构造另外两个过滤器的 S 集合。因此,m 的值等于 S 中 n 和 k 的总和。

在数据加入块中时,计算块中的数据的体积和速度的绝对值。设 $\rho_t(I)$ 和 $u_t(I)$ 为 t 时刻图像数据的绝对体积和速度,这些值开始时均为零。每次将图像数据添加到其相应的存储块中时,都会使用式(3)和式(4)更新数据体积和速度值。其中,δ 表示在一个时间点通过过滤器的数据的体积量。

$$\rho_t(I)=\rho_t(I)+\delta \tag{3}$$

$$u_t(I)=u_t(I)+1 \tag{4}$$

类似地,音频数据的体积为 $\rho_t(A)$,音频数据的速度为 $u_t(A)$,视频数据的体积为 $\rho_t(V)$,视频数据的速度为 $u_t(V)$,文本数据的体积为 $\rho_t(T)$,文本数据的速度为 $u_t(T)$。这些值根据它们各自的块进行计算。在第二阶段中,将使用这些值来估计第 $(t+1)$ 时刻数据的体积和速度。

2.2 利用自回归模型预测体积和速度

数据流可能不具有周期性峰值。比如,当社交媒体上发生某些事情时,可能会导致在特定时间段内大量高速生成数据。因此,在估计下一个时间间隔内到达的数据体积和速度的过程中,可变性有着重要的影响。为了减小可变性的影响,本文采用自回归模型[15]。自回归模型使用预测变量的线性组合来预测变量的值。

设 $\rho'_{t+1}(I)$ 和 $u'_{t+1}(I)$ 分别表示 $(t+1)$ 时刻图像数据的体积和速度的预测值。使用自回归模型,这些值由式(5)和式(6)确定。

$$\rho'_{t+1}(I)=\alpha_1\rho_t(I)+\alpha_2\rho_{t-1}(I)+\cdots+\alpha_q\rho_{t-q+1}(I) \tag{5}$$

$$u'_{t+1}(I)=\beta_1 u_t(I)+\beta_2 u_t(I)+\cdots+\beta_q u_{t-q+1}(I) \tag{6}$$

其中:

$$\alpha_q=\frac{cov(\rho_t(I),\rho_{t-q}(I))}{var(\rho_t(I))} \tag{7}$$

$$\beta_q=\frac{cov(u_t(I),u_{t-q}(I))}{var(u_t(I))} \tag{8}$$

式中:$cov(\,)$ 和 $var(\,)$ 分别代表协方差和方差。同样,使用自回归模型计算音频、视频和文本数据的预测体积和速度。

2.3 计算相对体积和速度

本文并没有对大数据作一个具体的定义,因此,第二阶段的体积和速度的预测值需要与其他到达云数据中心的数据请求进行比较。比较后得到的值称为相对体积和速度。以下讨论图像数据的相对体积和速度的计算。音频、视频和文本数据的计算与图像类型的计算方式类似。

$(t+1)$ 时刻图像数据的相对体积 $\rho''_{t+1}(I)$ 和速度 $u''_{t+1}(I)$ 用式(9)和式(10)计算得到,其中,$\max(\rho_t(I))$ 和 $\max(u_t(I))$ 分别表示在时间跨度 t 期间内到达的所有流中,图像数据的最大体积和速度。

$$\rho''_{t+1}(I)=round\left(\frac{\rho'_{t+1}(I)}{\max(\rho_t)},1\right) \tag{9}$$

$$u''_{t+1}(I)=round\left(\frac{u'_{t+1}(I)}{\max(u_t)},1\right) \tag{10}$$

式中:函数 $round(num,num\_digit)$ 表示将数字 num 四舍五入到具有 num_digit 位小数的数。本文中,将体积等于 $\max(\rho_t(I))$ 的大数据流的相对体积设置为 1。这意味着相对体积的最大值是 1。与之对应的,相对体积的最小值是零。因此,式(9)的取值范围为 $[0,1]$。此外,由于式(9)的结果四舍五入到小数点后一位,所以 $\{0,0.1,0.2,\cdots,0.9,1\}$ 是相对体积的可能值的集合。相对速度的计算与相对体积类似。

2.4 预测值的 CoD 表示

在云资源的分配中,有必要以一种特殊的形式表示预测的 4V 值。本文中将数据的相对体积和速度预测值进行整合,形成数据强度度量。例如,图像数据的强度 $\varphi(I)$ 可以使用式(11)计算得到。

$$\varphi(I)=\rho''_{t+1}(I)\cdot u''_{t+1}(I) \tag{11}$$

音频、视频和文本数据的强度也根据该等式进行计算。由于 $\rho''_{t+1}(\,)$ 和 $u''_{t+1}(\,)$ 的范围都在 $[0,1]$ 之间,所以强度的取值范围也是 $[0,1]$。

在获得图像、音频、视频和文本数据的强度之后,用数据特征(CoD)矢量的形式来表示它们。CoD 的定义为:$\mathrm{CoD}=(\varphi(I),\varphi(A),\varphi(V),\varphi(T))$。

3　基于改进 SOM 聚类的资源管理

3.1　PSO 改进 SOM 聚类

传统的聚类算法如 K_meas 等,往往需要事先定义聚类数目。通常基于经验知识来确定类别个数,而且一般需要多次尝试,这种方法具有很大的盲目性。

在本文资源聚类应用中,需要根据数据流特征来分配合适类型的资源,如果事先定义资源类型数量则不能适应数据流的变化。为此,本文采用了 SOM 聚类算法,利用 SOM 的可视化功能来动态确定聚类数目,避免传统聚类算法确定聚类数目的盲目性。SOM 是用于聚类分析的无监督神经网络学习技术之一[16]。初始时将每个资源都作为一个聚类,通过计算资源与数据流的距离来动态聚类,以此为每个数据流分配最佳资源。SOM 由输入层和输出层组成,其结构如图 2 所示。

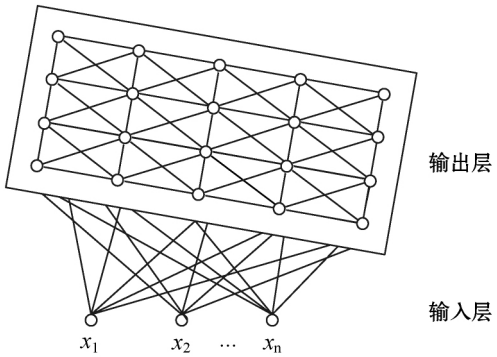

图 2　SOM 映射网络

利用 SOM 进行聚类分析可分为以下几步:

初始化参数: 输入、输出神经元个数 n、m,当前迭代次数 t,最大迭代次数 T,初始权值 $W_{ij}(1)$,学习速率因子 $\psi(1)$,邻域半径 $N(1)$。

(1) 对 SOM 中神经元的初始权值 $W_{ij}(1)$ 进行归一化,使权值的范围处在(0,1)之间。

(2) 计算训练样本和输出层神经元间的距离值,计算公式如下:

$$D_{ij}(t) = \sum_{i}^{n} (W_{ij}(t) - x_i(t))^2 \quad j = 1,2,\cdots,m \tag{12}$$

(3) 找出距离最小值对应的输出层神经元,将它定义为获胜神经元 j:

$$j = \arg\min_{j} D_{ij}(t) \tag{13}$$

(4) 调整获胜神经元的权值:

$$W_{ij}(t+1) = W_{ij}(t) + \psi(t)(x_i - W_{ij}(t)) \tag{14}$$

(5) 按照式(15)更新获胜神经元的邻域以及 SOM 的学习速率。

$$\begin{cases} \psi(t) = \psi(t)(1 - t/T) \\ N(t) = N(1)(1 - t/T) \end{cases} \tag{15}$$

(6) 如果满足以下所列条件中的任意一个,则 SOM 算法结束,输出聚类结果。① $\psi(t)$ 的值下降为 0;② $t = T$。否则,$t = t + 1$,返回到步骤(2)重复执行。

虽然 SOM 算法的样本学习更加精确,并且收敛速度能够满足多数情况下的要求,但 SOM 算法的收敛效果与权值的初始分布有着密切的关系。本文利用 PSO 算法优化 SOM 算法的权值,以增强收敛效果。

在 PSO 算法中,候选解通常用粒子来表示。在一个 D 维空间中,令粒子的位置 P、速度 V 表示如下:

$$\begin{cases} P = \{p_1, p_2, \cdots, p_D\} \\ V = \{v_1, v_2, \cdots, v_D\} \end{cases} \tag{16}$$

在 PSO 算法的迭代过程中,粒子根据个体最优值和全局最优值更新自身的位置 $x_i(t+1)$ 和速度 $v_i(t+1)$:

$$\begin{cases} x_i(t+1) = x_i(t) + v_i(t+1) \\ v_i(t+1) = wv_i(t) + c_1\lambda_1(pbst_i - p_i(t)) + c_2\lambda_2(gbst_i - p_i(t)) \end{cases} \tag{17}$$

式中: w 表示惯性权重; c_1 和 c_2 表示学习因子,它们的取值范围是(1,2); λ_1 和 λ_2 是两个在(0,1)之间的常数; $pbst_i$ 和 $gbst_i$ 分别表示个体最优粒子和全局最优粒子。

基于 PSO 优化 SOM 的具体步骤如下:

(1) 粒子位置编码表示为:

$$P = \{w_{11}, w_{12}, \cdots, w_{1m} \cdots w_{n1}, \cdots, w_{nm}\} \tag{18}$$

式中: m 表示资源聚类数目,对应于 SOM 算法中的输出神经元个数; n 表示资源属性数目,对应于 SOM 算法中的输入神经元个数。

(2) 根据式(17)更新粒子的位置 $x_i(t+1)$ 和速度 $v_i(t+1)$。

(3) 计算粒子的适应度值,找出个体最优粒子 $pbst_i$ 和全局最优粒子 $gbst_i$。适应度值的计算式为:

$$fit(p) = 1 / \sum_{i=1}^{n_s} D_{ij}(t) \tag{19}$$

式中: i 表示样本, j 表示 i 相对应的获胜神经元, $D_{ij}(t)$ 表示 i 和 j 之间的距离。

(4) 当得到的全局最优粒子 $gbst_i$ 变化时,返回步骤(2)中重复执行操作,直到 $gbst_i$ 不再发生变化。

3.2　动态集群生成及资源分配

在本文提出的算法中,将特征预测步骤中得到的 CoD 向量作为输入向量,输入到改进 SOM 中进行聚类,如算法 2 所示。聚类过程从权重 W 和学习速率因子 ψ 的初始化开始,利用 PSO 初始化 SOM 的权值初始分布。SOM 随机选择一个资源向量(S_i),并从每个

CoD 向量中计算与它的距离。具有最小距离的 CoD 向量(C_J)称为获胜向量,并将资源 S_i 分配到 CoD 向量 C_J 对应的数据流。每个资源都会重复该过程,所有获胜向量为 C_J 的资源称为一个集群。

算法 2 利用改进 SOM 生成动态集群

1. 利用 PSO 初始化 SOM 的权值初始分布;
2. 将学习速率因子 ψ 定义为略小于 1 的值;
3. 重复 3 - 9,直至计算边界不满足条件;
4. 对于每个输入资源向量 S_i,重复步骤 6 - 8; 5. 对于每个输出神经元 j,根据下式计算 S 的欧氏距离的平方:

$$D(j) = \sum_{k=1}^{q} (S_{ik} - W_{jk})^2;$$

6. 选择 $D(j)$ 最小的 j;
7. 设置聚类特征 $CoC(S_i) = CoD(C_j)$;
8. 根据下式更新 j 的所有拓扑邻居的权值:

$$W_{jk}(t+1) = (1-\psi) W_{jk}(t) + \psi(t) S_k;$$

9. ψ 单调递减;
10. 基于各自的 CoC 输出虚拟聚类;

集群形成后,使用具有 $CoC = CoD$ 属性的集群来为数据流分配资源。如果选择的集群有足够的资源来处理数据流,则分配该集群。否则,搜索并分配具有足够资源,且最近的拓扑有序集群,这么做可以减小数据流的等待时间。

4 实验分析

实验中分为两个步骤,即数据流的特征估计和资源分配。其中,在 PC 机上通过 Java 编程实现自回归模型,对下一时间间隔到达的数据的特征进行估计。在第二个步骤中,使用 AWS SDK for Java 工具,在 Amazon Web Services 云平台上开发和部署 Java 应用程序,连接 Amazon 弹性计算云 EC2 中的计算资源,实现资源分配。实验分为两个部分,即预测分析和算法对比分析。

4.1 实验数据集

生成四个容量较大的数据集。第一个数据集是由 86 280 个图像组成的图像数据集,第二个数据集是一个音频集合,包含 1 568 首音乐曲目,第三个数据集是一个包含 32 小时视频的监控视频数据集。第四个数据集是一个由 1 000 万个单词组成的文本数据集。使用这四个数据集创建一个数据库,该数据库满足以下特征:

(1) 它由 12% 的图像数据,21% 的音频数据,24% 的视频数据和 43% 的文本数据组成。

(2) 图像数据在数据集前部时的体积较小,当它越靠近数据集尾部时,体积逐渐增大。

(3) 音频数据在数据集前部中的体积很大,而在数据集的末端有所下降。

(4) 视频数据的体积先下降然后增加。

(5) 文本数据的体积在整个数据集中保持均匀,几乎不变。

将数据库中的数据以流的形式输入到算法中,时间为 1 小时。其中图像、音频、视频和文本数据的速度遵循以下模式:

(1) 图像数据的速度先下降然后随时间增加,使其形成流的整体速度的 19%。

(2) 音频数据的速度随时间减小,使其形成流的整体速度的 23%。

(3) 视频数据的速度随时间增加,使其形成流的整体速度的 22%。

(4) 文本数据的速度几乎保持不变,使其形成流的整体速度的 36%。

4.2 数据流体积和速度的预测

将数据流的实际值与本文算法的预测值进行比较,结果如表 1 和表 2 所示。本次实验中,以 0.5 个样本/min 的恒定采样频率进行采样。

表 1 数据流体积的实际值与本文算法的预测值比较

GB

数据类型		10 min	20 min	30 min	40 min	50 min	60 min
图像数据	实际值	5.06	10.19	15.31	18.75	24.62	28.33
	预测值	5.01	9.63	15.99	19.02	24.57	28.25
音频数据	实际值	37.65	33.56	21.94	18.36	15.21	10.34
	预测值	37.42	33.61	22.32	17.95	15.05	10.07
视频数据	实际值	25.15	23.26	18.64	27.44	31.56	35.65
	预测值	25.96	23.71	17.96	27.56	31.51	35.99
文本数据	实际值	39.75	42.51	41.17	42.72	42.05	41.29
	预测值	40.23	43.42	41.33	41.89	42.52	41.57

表 2 数据流速度的实际值与本文算法的预测值比较

数据流量数量/min

数据类型		10 min	20 min	30 min	40 min	50 min	60 min
图像数据	实际值	116.34	93.52	78.59	85.76	112.48	139.56
	预测值	119.50	95.34	77.69	89.38	108.53	138.89
音频数据	实际值	181.76	169.33	143.35	122.67	114.06	109.37
	预测值	187.38	165.56	140.09	123.78	112.51	107.66
视频数据	实际值	107.65	119.34	124.71	136.92	146.19	151.25
	预测值	111.24	121.96	126.27	135.43	145.62	149.83
文本数据	实际值	203.36	209.51	211.58	207.82	205.64	209.91
	预测值	205.11	208.42	209.67	205.76	207.42	210.67

从表 1 和表 2 可以看出,四种数据集的实际值与预测值之间的差异很小。图 3 为四种数据类型的数据流体积和速度预测的平均绝对预测误差(MAPE) 。

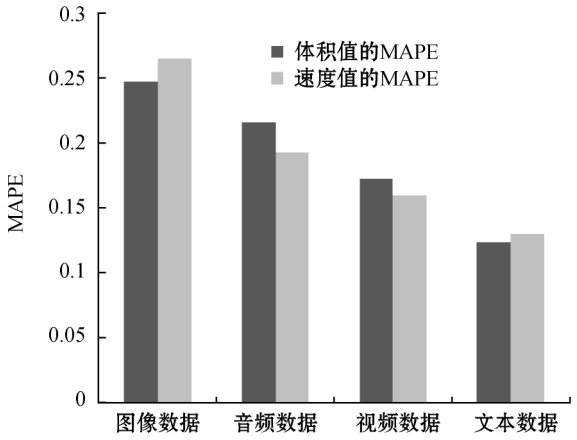

图 3　平均绝对预测误差

从图 3 可以看出,MAPE 随着数据占比的增加而减少。如上文所述,数据流由 12% 的图像数据,21% 的音频数据,24% 的视频数据和 43% 的文本数据组成,而预测图像、音频、视频和文本数据体积的 MAPE 分别为 0. 247、0. 216、0. 172 和 0. 123。图像数据占比是最小的,但它的 MAPE 是最高的; 文本数据的占比是最大的,而它的 MAPE 是最低的。因此,对于占比更高的数据类型,其 MAPE 会更低,并且这一数值随着数据占比的减少而增加。从数据速度的角度上可以观察到类似的现象,这符合大数定律,也就是说,误差随着数据大小的增加而减小。

从以上结果可知,本文算法能有效地预测图像、音频、视频和文本数据的体积和速度。

4. 3　算法对比

将本文提出的算法与文献 [5] 提出的基于 QoS 的资源管理算法、文献 [8] 提出的基于马尔可夫链的资源分配算法进行实验比较。本次实验中生成具有不同体积和速度占比的 5 个大数据流集,流生成的过程与第 4. 1 节相同。表 3 为 5 个数据流集中各种数据的体积和速度的占比,每隔十分钟将一个流送入算法中。

表 3　生成的流中不同数据的体积和速度的百分比

数据流量	体积/%				速度/%			
	图像	音频	视频	文本	图像	音频	视频	文本
流 1	17	21	22	40	22	25	19	34
流 2	17	19	29	35	33	18	26	23
流 3	27	26	22	25	28	27	21	24
流 4	33	38	0	29	31	35	0	34
流 5	4	33	31	32	6	37	28	29

分别使用上述三种算法,从 Amazon 弹性计算云 Elastic Compute Cloud(EC2) 中选择虚拟机(VM) 作为资源,分配给适当的数据流。本文中选择 10 个 VM 实例来进行资源管理,每个实例都来自于 EC2 中针对计算密集型工作负载优化的 c4. large 实例,配备 2. 9 GHz Intel Xeon E5-2666 v3 处理器。虚拟机处理生成的流,并在每个流上运行 Alon-Matias-Szegedy(AMS) 算法。AMS 用于确定流中不同元素的频率。整个实验的执行时间为一个小时。

图 4 为三种算法的资源利用率的变化。在文献 [5] 提出的基于 QoS 的算法中,算法的处理能力、GPU 功率、RAM 和输入数据的大小由用户提供。这些用户需求用于为输入请求分配资源。用户需求取决于数据特征,在大数据流的情况下,用户通常不知道数据特征。因此,在基于 QoS 的方法中,用户可能无法为大数据流确定合适的资源,并且数据流在整个时间段内运行在相同的资源上。文献 [8] 提出的算法能有效地预测数据流的大小,以选择合适资源来处理数据,但是算法没有考虑大数据资源分配的高速性、多样性和可变性等因素。本文提出的算法能够预测流的 4V,并且能够随流中数据的特征变化恰当的分配资源。因此,在图 4 中,与文献 [5]、文献 [8] 提出的算法相比,本文提出的算法资源利用率更高。

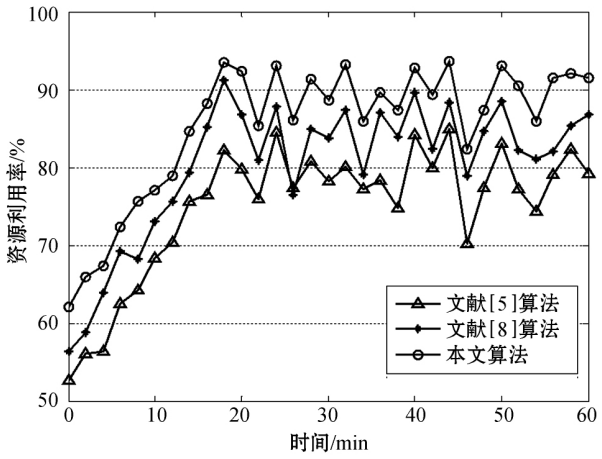

图 4　资源利用率

图 5 为三种算法的整体执行延迟的比较。在文献 [5] 算法中,整个执行期间执行延迟几乎相同,只是在实验结束时执行延迟略有增加。这是因为随着更多流添加到算法中,算法所需集群的不可用的概率逐渐增加(所请求的资源可能已被其他流所占用) ,因此,算法需要更多的时间寻找可用且最近的拓扑有序集群。在文献 [8] 算法中,执行延迟随着执行时间的增加慢慢增大。在本文提出的算法中,实验开始时,执行延迟较高,这是因为本文算法在分配适当集群之前就预测了流的 CoD,这就减少了执行延迟,并且使合适资源可以更快地处理数据流。此外,从图 5 可以看出,自从

10 min 后在系统中添加新流时，每隔 10 min 执行延迟会出现周期性峰值。这是因为，新流添加后需要重新计算所有流的 CoD，这就导致执行延迟的周期性增加。尽管如此，本文提出算法的整体执行延迟小于其余两种算法。

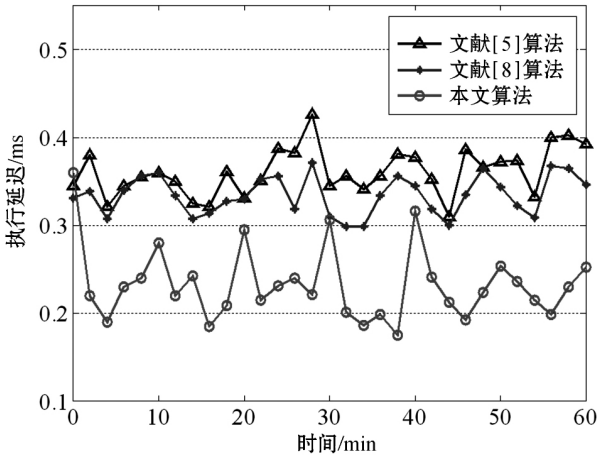

图 5 执行延迟

图 6 为三种算法的的响应时间比较。本文提出的算法的响应时间在整个实验中几乎保持相同。这是因为无论数据流中的数据特征发生何种变化，在本文提出的算法中始终能找到更合适的资源集群来处理该数据流，这带来了稳定的响应时间。对于文献 [5] 算法，在整个执行时间段内，无论数据特性如何变化，流始终在相同的资源上运行，响应时间就会随着执行时间而增加。文献 [8] 算法仅仅根据数据流的大小来调整资源，没有考虑资源速度的可变性，所分配的资源不一定最合适。

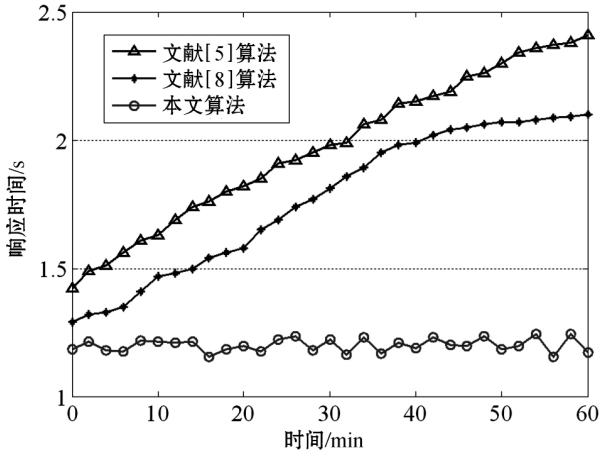

图 6 响应时间

5 结 语

本文提出了一种高效的大数据流资源管理算法。所提出的算法基于数据特性的预测来分配合适资源，从而高效的处理数据流。这种分配方式可以随着数据特性改变做出调整，使对数据流的响应时间保持稳定。此外，由改进型 SOM 形成的簇的拓扑排序可以减少了流的等待时间。实验结果表明，和常用的分配算法相比，本文提出的算法能有效提高云资源的利用率。

参 考 文 献

[1] Liu C, Yang C, Zhang X, et al. External integrity verification for outsourced big data in cloud and IoT: A big picture [J]. Future Generation Computer Systems, 2015, 49(C): 58 – 67.

[2] 王元卓，靳小龙，程学旗. 网络大数据: 现状与展望 [J]. 计算机学报，2013，36(6): 1125 – 1138.

[3] 郑宇超，夏学文，艾冬梅. 基于队列理论的云资源分配收益最大化算法 [J]. 计算机应用与软件，2017，34(11): 252 – 257.

[4] Vasile M A, Pop F, Tutueanu R I, et al. Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing [J]. Future Generation Computer Systems, 2015, 51(C): 61 – 71.

[5] Sandhu R, Sood S K. Scheduling of big data applications on distributed cloud based on QoS parameters [J]. Cluster Computing, 2015, 18(2): 817 – 828.

[6] Sun D, Zhang G, Yang S, et al. Re-Stream: Real-time and energy-efficient resource scheduling in big data stream computing environments [J]. Information Sciences, 2015, 319: 92 – 112.

[7] Rahman M M, Graham P. Responsive and efficient provisioning for multimedia applications [J]. Computers & Electrical Engineering, 2016, 53(C): 458 – 468.

[8] Zhang Q, Chen Z, Yang L T. A nodes scheduling model based on Markov chain prediction for big streaming data analysis [J]. International Journal of Communication Systems, 2015, 28(9): 1610 – 1619.

[9] Castiglione A, Pizzolante R, Santis A D, et al. Cloud-based adaptive compression and secure management services for 3D healthcare data [J]. Future Generation Computer Systems, 2015, 43/44: 120 – 134.

[10] Baughman A K, Bogdany R J, Mcavoy C, et al. Predictive Cloud Computing with Big Data: Professional Golf and Tennis Forecasting [Application Notes] [J]. Computational Intelligence Magazine IEEE, 2015, 10(3): 62 – 76.

[11] 王旭，王国中，范涛. 深度图像的分块自适应压缩感知 [J]. 计算机应用研究，2016，33(3): 903 – 906.

[12] Zhou A C, He B, Liu C. Monetary Cost Optimizations for Hosting Workflow-as-a-Service in IaaS Clouds [J]. IEEE Transactions on Cloud Computing, 2016, 4(1): 34 – 48.

[13] 张蕾，章毅. 大数据分析的无限深度神经网络方法 [J]. 计算机研究与发展，2016，53(1): 68 – 79.

（下转第 280 页）

5 结 语

智能手机具有多品牌竞争扩散的特点。而目前，我们尚未见到具备强大销量预测能力的多品牌产品扩散模型出现。消费者的购买决策受产品、消费者及其所处社会环境中众多因素的影响，找到能将这些因素有机结合的一套建模理论和方法，以确保所构建的模型能准确刻画消费者的决策逻辑及其对产品竞争与扩散过程的影响逻辑，是构建具备强大销量预测能力的多品牌产品扩散模型的关键。本文运用多 Agent 理论和模糊推理技术，综合产品、消费者和社会环境中的重要因素所构建的多品牌产品扩散模型，被大量实验证明能较好地预测多品牌智能手机的销量走势。本文的研究不仅丰富和发展了创新扩散理论，而且还可为相关企业的产品营销提供决策参考。

参 考 文 献

[1] 艾兴政，唐小我. 广告媒介下两种产品竞争与扩散模型研究 [J]. 管理工程学报，2000，14(3)：19－22.

[2] 谭建，王先甲. 替代性产品在广告媒介下的扩散动态研究 [J]. 软科学，2014，28(3)：110－113，118.

[3] Erickson G M. An oligopoly model of dynamic advertising competition [J]. European Journal of Operational Research，2009，197(1)：374－388.

[4] Eliashberg J，Jeuland A P. The impact of competitive entry in a developing market upon dynamic pricing strategies [J]. Marketing Science，1986，5(1)：20－36.

[5] 丁士海，韩之俊. 考虑竞争与重复购买因素的耐用品品牌扩散模型 [J]. 系统工程理论与实践，2011，31(7)：1320－1327.

[6] Guseo R，Mortarino C. Within-brand and cross-brand word-of-mouth for sequential multi-innovation diffusions [J]. IMA Journal of Management Mathematics，2014，25(3)：287－311.

[7] Muller E，Peres R. The effect of social networks structure on innovation performance：A review and directions for research [J]. International Journal of Research in Marketing，2018.

[8] Kiesling E，Gunther M，Stummer C，et al. Agent-based simulation of innovation diffusion：a review [J]. Central European Journal of Operations Research，2012，20(2)：183－230.

[9] Schramm M E，Trainor K J，Shanker M，et al. An agent-based diffusion model with consumer and brand agents [J]. Decision Support Systems，2010，50(1)：234－242.

[10] Kim S，Lee K，Cho J K，et al. Agent-based diffusion model for an automobile market with fuzzy TOPSIS-based product adoption process [J]. Expert Systems with Applications，2011，38(6)：7270－7276.

[11] 李英，胡剑. 基于智能体的多类新能源汽车市场扩散模型 [J]. 系统管理学报，2014，23(5)：711－716.

[12] Stummer C，Kiesling E，Günther M，et al. Innovation diffusion of repeat purchase products in a competitive market：An agent-based simulation approach [J]. European Journal of Operational Research，2015，245(1)：157－167.

[13] Jiang G，Tadikamalla P R，Shang J，et al. Impacts of knowledge on online brand success：an agent-based model for online market share enhancement [J]. European Journal of Operational Research，2016，248(3)：1093－1103.

[14] Mamdani E H，Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller [J]. International Journal of Man-Machine Studies，1975，7(1)：1－13.

[15] Barabási A，Albert R. Emergence of Scaling in Random Networks [J]. Science，1999，286(5439)：509.

(上接第 268 页)

[14] 李玮，张大方，黄昆，等. 面向大数据处理的高精度多维计数布鲁姆过滤器 [J]. 电子学报，2015，43(4)：652－657.

[15] Li K，Zhu Y，Yang J，et al. Video super-resolution using an adaptive superpixel-guided auto-regressive model [J]. Pattern Recognition，2016，51(C)：59－71.

[16] Agarwal A，Maheswaran R，Kurths J，et al. Wavelet Spectrum and Self-Organizing Maps-Based Approach for Hydrologic Regionalization -a Case Study in the Western United States [J]. Water Resources Management，2016，30(12)：4399－4413.

(上接第 274 页)

[9] Zhang J. RNN-BLSTM Based Multi-Pitch Estimation [C]//INTERSPEECH，Germany：Inter-speech，2016：1785－1789.

[10] 冯多，林政，付鹏，等. 基于卷积神经网络的中文微博情感分类 [J]. 计算机应用与软件，2017，34(4)：157－164.

[11] Graves A，Mohamed A R，Hinton G. Speech recognition with deep recurrent neural networks [C]//IEEE International Conference on Acoustics，Speech and Signal Processing. IEEE，2013：6645－6649.

[12] 谢志宁. 中文命名实体识别算法研究 [D]. 杭州：浙江大学，2017.

[13] Bengio Y，Duchme R. A neural probabilistic language model [J]. Journal of Machine Le-arning Research，2003，3(6)：1137－1155.

[14] 王蕾. 基于神经网络的中文命名实体识别研究 [D]. 南京：南京师范大学，2017.

[15] 冯艳红，于红，孙庚，等. 基于 BLSTM 的命名实体识别方法 [J]. 计算机科学，2018，45(2)：261－268.

[16] 李航. 统计学习方法 [M]. 北京：清华大学出版社，2012.