维普资讯 http://www.cqvip.com

基于连续度聚类和动态ARMA时间序列预测I/O区域*)

李怀阳 谢长生 刘 艳 赵 振

(华中科技大学 计算机学院 信息存储系统教育部重点实验室 武汉 430074)

摘 要 为了动态优化存储系统中数据的分布,存储系统需要能动态发现密集I/O区域和预测未来密集I/O访问的区域,并根据发现和预测的结果来指导存储系统的优化。为此,本文根据存储系统的特点提出了实用且高效的基于连续度的聚类算法来发现密集I/O访问的区域,并采用ARMA时间序列模型来预测密集I/O可能访问的区域。为提高预测的准确性,采用了动态参数估计的策略。通过大量实验的结果验证了这两种算法的正确性和预测的准确性,对存储系统的优化具有较好的指导作用。

关键词 聚类,预测,连续度,ARMA

A New Sequence Degree-based Clustering Algorithm and Dynamical ARMA Time Series Forecasting for I/O Requests

LI Huai-Yang XIE Chang-Shen LIU Yan ZHAO Zhen

(Key Laboratory of Data Storage System, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Storage system could optimize data distribution accordingly, on condition that it dynamically finds and predicts the storage areas requested frequently. So, this paper not only introduces a new sequence degree-based clustering algorithm to find the frequently accessed storage areas, but also adopts ARMA time series model to forecast the storage areas requested frequently by future I/O requests. To address the problem of accurate forecast, this paper adopts dynamic parameter estimation policy to ARMA model. The results of a large number of simulations validate the accuracy of the clustering algorithm and the preciseness of the ARMA time series model of dynamic parameter estimation policy.

Keywords Clustering, Prediction, Sequence degree, ARMA

1 引言

随着计算机体系结构和半导体工业的发展,I/O成了计算机系统的主要瓶颈,将请求的数据分布到多个磁盘上并行工作,改善I/O性能的磁盘阵列(RAID)[1]被广泛使用。但由于在实际工作环境中,工作负载变化的多样性和复杂性,所配置的RAID性能难以达到最优:不同的RAID有不同的适用范围,只能对分布范围比较窄的工作负载达到较好的性能。为了克服这种缺陷,我们提出了"逻辑进化存储系统"[11]这一概念。进化是指系统自身相应的结构和属性不断适应环境变化,通过选择向自身最佳组织结构方向发展的过程。逻辑组织结构进化,即存储系统中数据的组织以及整个系统中各种不同属性数据的分布可以随着外在的数据流输入输出的变化而相应改变,以期获得最好的系统性能。并由此设计了一种适合不同RAID之间动态转换的数据分布策略,其基本思想是系统根据I/O请求的负载特性动态地调节数据分布形式,如RAID级别、条带单元大小、配置cache的读写策略等。为了实现此目标,需要获得在某一段时间内工作负载的分布形式,找到密集I/O访问的区域,根据以往的工作负载特性来预测未来某一段时间密集I/O可能访问的区域,由此来调节数据的分布形式。为解决这个问题,本文从两个方面入手:一是采用基于连续度的I/O聚类算法,它能高效、可靠地发现密集I/O访问的区域;其次是根据I/O请求局部性的特征,利用ARMA时间序列模型来预测未来密集I/O可能访问的区域。为提高预测的准确性,提出了动态参数估计的策略,使得预测值的匹配率达到很高的比例,能充分指导对存储系统的调节。

2 相关工作

聚类,就是把大量的m维n个数据样本聚集成k个类,使同一类中样本的相似性最大,不同类中样本的相似性最小。主要的聚类算法可以分为如下几类[2]:划分的方法、层次的方法、基于密度的方法、基于网格的方法和基于模型的方法。而这些算法主要来源于数据库方面的应用,算法复杂度较高,不太适合存储系统的实时调节。文[3]利用NAS服务器从应用程序获取数据间的互连列表,并发送到磁盘控制器端,在磁盘控制器级实现无浮点运算的聚类和缓存算法。其关键的互连列表是通过Web页的超级链接和XML链接特性来获取文档之间的互连特性。文[4]利用决策树来聚类和预测文件的特性,如文件大小、读写方式及生命周期等特性,利用获取的特性来实现文件的分布策略。其基本思想是利用文件在创建时的属性(文件名、用户和组ID、文件类型、权限等)来聚类和预测。这些方法主要是针对文件的聚类和预测。而在存储系统的设备层,只有物理块的信息,要获取足够多的信息来对负载进行聚类和预测是比较困难的。文[5]在应用层利用k-平均值算法聚类ftp服务器的I/O负载特性,其最大问题在于需要事先确定聚类簇的数量,而对动态变化的负载来说,是比较困难的。文[6]采用时间序列模型来预测应用程序的I/O请求空闲时间,并在空闲时间内对数据进行预取。这项工作对本文的研究有很好的借鉴作用。文[7]利用马尔可夫模型预测应用程序要访问的块号,但需要建立很大的转移矩阵并进行复杂的矩阵运算。

*)国家自然科学基金(项目编号:60273073)和"973"(项目编号:2004CB318203)资助项目。**李怀阳** 博士研究生,主要研究方向为大规模存储、网络存储系统;**谢长生** 教授,博士生导师,主要研究方向为计算机体系结构、网络存储系统、采用新原理的超高密度、超高速存储技术。

维普资讯 http://www.cqvip.com

3 概念及聚类算法

由于本系统是实时的监控系统，要求算法尽可能简单。同时本系统并不是对长时间段的所有数据进行聚类，而只是聚类某一时间段密集 I/O 的数据。如图 1 所示，只需要聚类 1～10 或 21～30 时间段的数据，现有聚类算法均不太合适。因此我们基于存储系统的特点提出了连续度的定义，并根据连续度进行聚类。基本思想是在很小的时间段内发现密集 I/O 访问的区域，而忽略少量 I/O 访问的区域。为了方便起见，做以下假定：

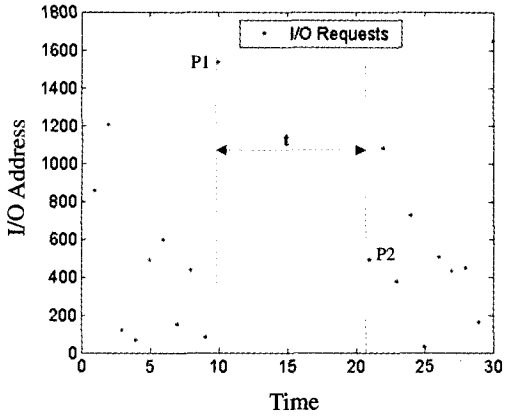


图 1 数据采样时间间隔说明

(1)将 I/O 请求称之为对象。

(2)根据存储系统中 RAID 的特点及进化存储系统中数据转换的需要，I/O 请求的基本单位是 RAID 中的条带大小。

(3)对 I/O 请求的采样时间段是以两个 I/O 请求的时间间隔超过阈值 H_t 为标准。如图 1 所示，若请求 $P1$ 与 $P2$ 的时间间隔 $t>H_t$，则可认为时间段 1～10 和 21～30 的数据为两个聚类区域。

3.1 连续度定义

定义 1 连续度 $S_w(p)$ 表示对象的连续程度，与其长度成正比。对一个对象 p 来说，若长度为 1，其左侧或右侧无相连的对象，连续度为 a_1；其左侧或右侧有一个相连的对象，连续度为 a_2；两侧均有相连的对象，连续度为 a_3，$a_3>a_2>a_1$。

之所以这样定义，是源于存储系统中，如果请求是连续的，系统就有很高的存储效率。对比图 2(P_i 为请求，l_i 为请求长度，$i=1,2,3$)中的状态(1)、(2)和(3)，处于状态(2)时的存储效率要比状态(1)高，与状态(3)相同。若以连续度来计算，则

$$S_w(p_1)=a_3l_1-2(a_3-a_2)$$

$$S_w(p_2)=a_3l_2-2(a_3-a_2)$$

其中，$l_3=l_1+l_2$

状态(1)的连续度之和为：

$$S_{w2}(p_1+p_2)=a_3l_3-4(a_3-a_2)$$

而状态(2)的连续度之和为：

$$S_{w2}(p_1+p_2)=a_3l_3-2(a_3-a_2)$$

即 $S_{w2}(p_1+p_2)>S_{w1}(p_1+p_2)$。同样可得到状态(2)的连续度等于状态(3)的连续度。这样，我们可以推出定理 1。

定理 1 如果两个对象的物理地址相连，两个对象可以合并成一个对象。

在存储系统中，状态(3)的存储效率与状态(4)相同，则可推出定理 2。

定理 2 如果两个对象的物理地址有重叠，其重叠部分的连续度为 a_3，两个对象可合并成一个对象。

若假定重叠部分长度为 m，则状态(4)的连续度为：

$$S_{w4}(p_1+p_2)=a_3(l_3-m)-2(a_3-a_2)+a_3m$$

$$=a_3l_3-2(a_3-a_2)=S_{w3}(p_3)$$

定义 2 核心对象：如果一个对象的连续度大于或等于阈值，这个对象称作核心对象。

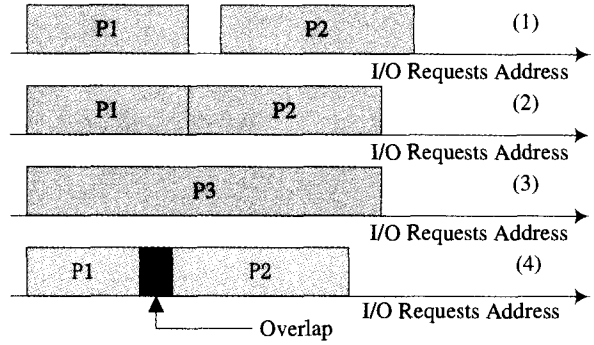


图 2 连续度示意图

定义 3 邻域：以核心对象 o 为中心，左右两侧距离 d 为半径的区域称为 o 的 d-邻域。邻域内对象的连续度之和称作 o 在 d-邻域内的连续度。

定义 4 直接可达：D 是对象集合，$p\in D$，$o\in D$。若 o 是核心对象，p 在 o 的 d-邻域内，则称 p 是从 o 出发关于 d 直接可达。

定义 5 可达性：D 是对象集合，若存在一个对象链 p_1，p_2，…，p_n；$p_i\in D(1\leqslant i\leqslant n-1)$；$p_{i+1}$ 是从 p_i 出发，关于 d 直接可达，则 p_n 是从 p_1 出发关于 d 可达。

定义 6 相连：D 是对象集合，$o\in D$，$p\in D$，$q\in D$，若存在 p 和 q 都是从 o 出发关于 d 的可达，则称 p、q 是关于 d 的相连。

定义 7 聚内：D 是对象集合，簇 C 是 D 的一个非空集合，如果 C 满足下面 3 个条件，则 C 为有效结果簇。

(1) $\forall p,q$：如果 $p\in C$ 且 q 是从 p 出发关于 d 的可达，则 $q\in C$。

(2) $\forall p,q\in C$：p、q 是关于 d 的相连，则 $p\in C$。

(3) $\forall p\in C$，$\sum_{i=1}^{k}S_w(p_i)\geqslant H_c$，$i=1,\cdots,k$，$H_c$ 为簇阈值。

3.2 参数选择

由前面的定义可知，基于连续度的聚类算法受参数的影响很大，不同的参数选择可能会导致不同的聚类结果，特别是在实时系统中，如何动态调节这些参数是必须要解决的问题。

对定义 1 中的 a_1、a_2 和 a_3，采用模糊定量的方法，将其赋值为 1、2 和 3。即请求处于 a_3 状态，其存储效率最好；请求处于 a_1 状态，其存储效率最差。核心对象阈值 H_o 的取值是平均连续度，即

$$H_o=\frac{\sum_{i=1}^{k}S_w(p_i)}{k},\ i=1,\cdots,k \tag{1}$$

当对象大于或等于阈值 H_o，可认为这个对象的长度较大或是密集 I/O 的区域。邻域 d 的取值是为了找到核心对象附近的密集 I/O，同时需要考虑存储系统的特点。若假定在存储系统中有 3 个对象 p_1、p_2 和 p_3，其中 p_2 是核心对象，l_i($i=1,2,3$)为 3 个对象的长度，d_1 和 d_2 是 3 个对象之间的间隔，如果核心对象 p_2 的 d-邻域包含 p_1 和 p_3 两个对象，那么 p_2 的 d-邻域所包含的存储空间为 $(l_1+l_2+l_3+d_1+d_2)k$，k 为一个条带的大小，有效空间为 $(l_1+l_2+l_3)k$，则有效存储空间

维普资讯 http://www.cqvip.com

比例为：

$$\eta=\frac{(l_1+l_2+l_3)k}{(l_1+l_2+l_3+d_1+d_2)k} \tag{2}$$

根据有效性的原则，其最少应大于50%，那么可推导出：$\frac{d_1+d_2}{l_1+l_2+l_3}<1$。如果取极限状态值：$d_1=d_2=d$ 且 $l_1=l_2=1$，那么

$$d<\frac{l_2}{2}+1 \tag{3}$$

因此，我们将 d 的取值为核心对象的长度的一半。对聚类簇来说，假定簇所占空间长度为 l，如果存在一个理想状态（簇中的存储空间为一个对象全部拥有），那么簇的连续度为：

$$S_w(C)=a_3 l-2(a_3-a_2) \tag{4}$$

而实际簇的连续度为 $\sum_{i=1}^{k}S_w(p_i)$，$i=1,\cdots,k$。同样，为了达到有效的存储空间利用率，

$$\sum_{i=1}^{k}S_w(p_i)\geqslant\frac{1}{2}S_w(C) \tag{5}$$

所以我们将簇阈值 H_c 设为：

$$H_c=\frac{1}{2}S_w(C)\text{且 }H_c\geqslant 60 \tag{6}$$

之所以 H_c 要大于等于60，是因为若存储系统对存储空间中的少量数据进行频繁的调节，既缺乏效率，又会造成系统的瓶颈。通过后面实验表明，这样的取值是可行的。

3.3 连续度算法

算法1 基于连续度的聚类算法

输入：I/O流 D

输出：以簇集合形式描述的聚类结果

(1)根据I/O流 D 的物理块地址排序，形成一个对象链表。

(2)合并有关对象，根据定义计算出所有对象的连续度。

(3)标记连续度大于阈值 H_o 的对象作为核心对象。

(4)计算出与核心对象相连的对象形成的所有集合，形成簇。

(5)从形成的结果簇中选择出连续度大于阈值 H_c 的簇，形成有效结果簇。

3.4 算法时间复杂度

本算法采用快速排序算法对I/O请求的物理地址进行排序，算法时间复杂度为 $O(N\log_2^N)$，其中 N 为某一段时间内I/O请求数。合并对象并计算对象的连续度的时间复杂度为 $O(N)$。标记核心对象的时间复杂度为 $O(M)$，其中 M 为合并后的对象数量，$M<N$。形成以核心对象为主的簇，它的时间复杂度是 $O(M)$，形成有效结果簇的时间复杂度为 $O(K)$，K 是簇的数量，$K<<N$。因此整个算法的时间复杂度是 $O(N\log_2^N)$。基于连续度的算法具有线性的时间复杂度，执行效率很高。其次它不是递归算法，不需要大量的堆栈操作，实际效率会更高。

4 聚类区域的预测

预测的关键是根据前面所获得的多个时间段密集I/O访问区域来预测在下一个时间段密集I/O可能访问的区域，因而需要获取I/O访问区域的特征值，并采用合适的算法来预测密集I/O未来访问区域的特征值。

4.1 特征提取

根据连续度算法，可以获取具有不同I/O特性的簇，而数据分布的调节依赖于簇的I/O特征。因此我们需要获取簇的平均请求大小、读写率等数据。同时为了预测聚类区域的变化趋势，我们也需要获取聚类区域的中心点及区域半径。参数说明如表1。

表1 参数说明

参数	说明
P_{BA}	簇中心点
R	簇半径
$\bar{R}$	平均请求大小
N_w	请求数
P_r	读请求比例
S_w	连续度

由于在一个时间段内聚类簇的数量可能不止一个，这里我们用曼哈坦距离来判断下一个时间段与本时间段多个簇之间的相似度。由第3节的定义可以知道，平均请求大小 $\bar{R}$ 和请求数 N_w 与连续度成正比，因此可用连续度代替这两个值。则簇之间的相似度可由下面公式给出：

$$\phi=|P_{BAi}-P_{BAj}|+|R_i-R_j|+|S_{wi}-S_{wj}|+|P_{ri}-P_{rj}| \tag{7}$$

当 ϕ 超过阈值 H_k 时，可认为两个簇无相似性。

4.2 预测模型

文[8]研究中发现，在存储系统中，I/O请求具有局部性的特征：空间局部性和时间局部性，即在某一段时间内，I/O请求密集地访问某一个或多个存储空间，称之为空间局部性。时间局部性表现为I/O访问的突发性上，即在某一时刻，I/O请求非常密集，而在其它时刻I/O请求数很少。同时在存储系统中，I/O访问具有很强的相关性。因此我们可以认为聚类区域也具有相关性，即在 T 时刻以前的聚类区域影响 T 时刻的聚类区域。我们可以有以下假定：

$$P_{BA(t)}=\phi_1 P_{BA(t-1)}+\phi_2 P_{BA(t-2)}+\cdots+\phi_k P_{BA(t-p)}+a_t \tag{8}$$

其中 ϕ_k 为待定系数，a_t 为误差系数。如果将 a_t 看成是白噪声序列，扩展为 $a_t-\theta_1 a_{t-1}-\cdots-\theta_q a_{t-q}$，则等式写成：

$$P_{BA(t)}-\phi_1 P_{BA(t-1)}-\cdots-\phi_k P_{BA(t-p)}=a_t-\theta_1 a_{t-1}-\cdots-\theta_q a_{t-q} \tag{9}$$

这就是一个典型的 $ARMA(p,q)$（自回归滑动平均）模型，即

$$\phi(B)P_{BA(t)}=\theta(B)a_t \tag{10}$$

其中 B 为一步延迟算子，$BP_{BA(t)}=P_{BA(t-1)}$，$\phi(B)=1-\phi_1 B-\cdots-\phi_p B^p$，$\theta(B)=1-\theta_1 B-\cdots-\theta_q B^q$。

同样，簇半径、平均请求大小、请求数和读请求比例一样可以用相同的方法建立对应的预测模型。

现在对 $ARMA(p,q)$ 模型有大量的研究，解决方法也非常成熟，本文不再阐述。由于系统采集的数据序列是一个动态的过程，对模型的建立也是一个动态的过程，因而本文阐述的关键是如何动态地识别、建立动态的 $ARMA(p,q)$ 模型，以便更好地预测聚类区域。

4.3 模型的鉴别

$ARMA(p,q)$ 模型要求数据序列是零均值的平稳时间序列，通常是用自相关函数（ACF）和偏相关函数（$PACF$）对序列进行鉴别。如果序列的自相关函数和偏相关函数呈现出缓慢衰减的趋势，那么可认为序列是一个非平稳的时间序列，即是 $ARIMA$ 模型。文[10]提出了解 $ARIMA$ 模型的方法：首先需要对时间序列进行平稳化处理，构造出零均值的平稳时间序列，即 $ARMA$ 模型，然后可用对应的方法求解。平稳化的方法一般是对时间序列进行差分，形成一个平稳时间序列，然后对这个平稳时间序列去均值化处理，形成零均值的平稳时间序列。如果经过平稳化后的时间序列的自相关函数呈有规律的时间间隔 S，那么认为这个序列具有季节性。为了消除季节性趋势，时间序列必须再经过 D 阶差分（D 为 S 的倍

维普资讯 http://www.cqvip.com

数)。这时序列可用 $ARIMA(p,d,q)\times(P,D,Q)^S$ 模型来描述。

首先,必须判断出时间序列是平稳的还是非平稳的,这可以转换为判断序列的自相关函数和偏相关函数的衰减是否缓慢。判断函数的变化趋势、快慢,最简单的方法就是函数的斜率,即

$$p=\frac{1}{J}\sum_{k=0}^{J}\frac{|P_k|-|P_{k+1}|}{|P_k|} \tag{11}$$

其中 P_k 表示序列的自相关函数值,J 表示最后一个超过自信区间的自相关函数值。通过后面的试验表明,当变化率小于15%时,基本上就可以认为这个时间序列是非平稳的。

4.4 差分阶数

差分阶数的确定现有很多的方法,但由于在本系统中是一个实时的系统,要求算法尽量简单,我们采用了文[9]中的近似估计算法,它是利用 *ARIMA* 序列的平方和的渐进性质给出差分阶数的简单估计。即

$$d=[\lg\sum_{n=1}^{N}X(n)^2/2\lg N] \tag{12}$$

其中 $X(n)$表示观测的 n 个样本序列。

如果经过差分后的时间序列的自相关函数值仍有部分超过自信区间,那么我们认为这个序列具有季节性。为了发现序列的季节性的特征,本文采用简单的差分方法。如图 3 所示,将自相关函数中超过自信区间的值所对应的滞后系数组成一个时间序列,然后对这个序列进行一阶差分。如发现某个值在差分后的序列中出现频率超过门限值(一般为 50%),那么我们就可以认为这个值为季节周期数。如果在一阶差分后没有发现季节性特征,则采用两阶差分,并以此类推,直到发现其季节性特征值。

自相关函数值	0	1	2	7	9	14	15	20	25	30	35	38	40	45	50
一阶差分值		1	1	5	2	5	1	5	5	5	5	3	2	5	5

图 3 时间序列季节性周期分析示意图

4.5 动态参数估计

对 *ARMA* 模型来说,有很多成熟的参数估计方法[10]。对一个实时的监控系统来说,这些方法有一个固有的缺陷,即预测函数及参数是固定的,不能实时随系统变化而改变,会随着时间的推移产生较大的误差。这点通过我们后面的实验也可以证实。为解决这个问题,我们采用动态参数估计的策略:在系统开始一段时间后,根据采样和聚类的数据,建立一个 *ARMA* 预测函数,由此函数进行一步预测。同时将每次预测值同实际值进行对比,当它们的误差超过某一门限值时,则认为现有的预测函数已不适应系统的变化。这时由最新的时间序列重新产生 *ARMA* 函数及其参数,并由此函数来预测。其结构示意图如图 4。

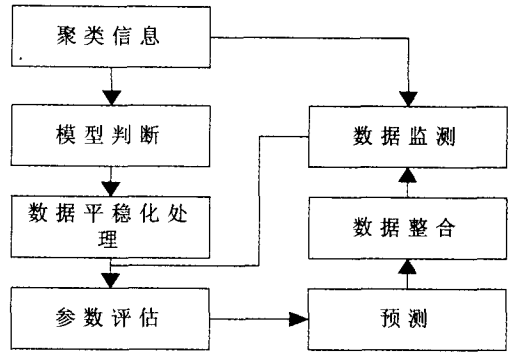


图 4 动态参数估计流程

从图 4 中可以看出,动态参数估计流程中有一个数据整合模块,它是与数据平稳化处理模块相对应的。在建立 *ARMA* 模型前,需要利用差分的方法对聚类的数据进行平稳化处理。而通过模型获得的预测数据也是针对平稳化处理后的数据,因此要获得真正的实际预测数据,需要对预测的数据进行还原,如果原有数据进行了差分,那么预测的数据需要加上被差分的数据;若原数据序列进行了零均值化处理,预测的数据需要加上原有数据的均值,具体的流程如图 5 所示。

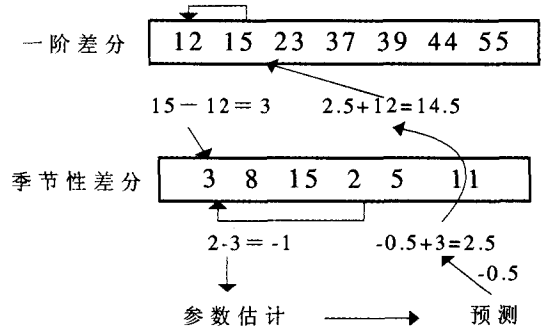
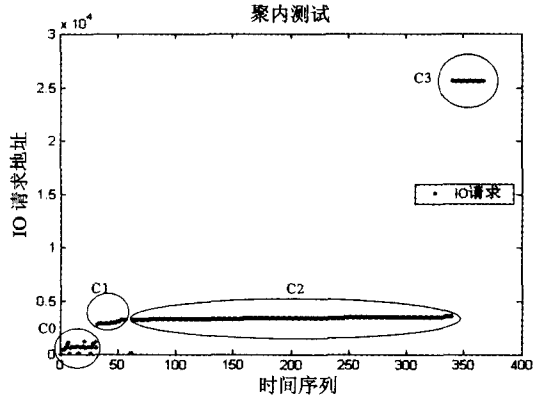


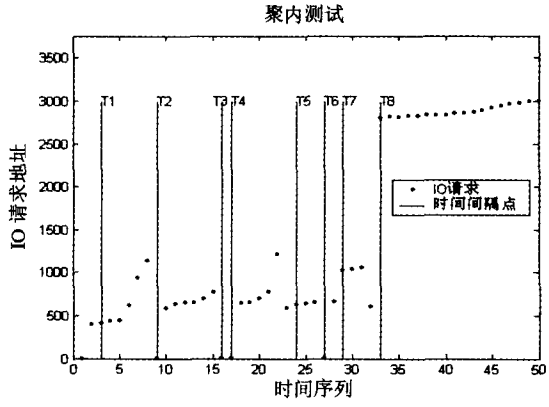
图 5 数据整合示意图

5 评测

我们的测试主要分为两个方面:一是利用本文提出的聚类算法能否正确地聚类 I/O 请求;另一部分检验动态预测算法的正确性。



(a)



(b)

图 6 聚类结果示意图

维普资讯 http://www.cqvip.com

5.1 I/O请求聚类

在本项测试中，我们利用HP公司一段trace文件，它记录的是典型的办公环境下客户端访问服务器的3个磁盘卷(卷号为21、23和35)情况，时间跨度为191.12h，有230370个请求数。设数据聚类区域的采样时间间隔为5s。

图6(a)显示的是卷21的I/O请求点和卷聚内中心结果。从图中可以看出，整个I/O请求密集区域可分为C0、C1、C2和C3共4个部分，而图的左上侧和下侧两个孤立点是噪点。通过基于连续度的算法得到的是3个有效簇，即C1、C2和C3。C0区域之所以没有成为一个有效簇，从图6(b)中可以看出，它是C0区域的详细图。最有可能形成有效簇是T2-T3和T4-T5两个时间段的数据。但根据定义及前面参数选择中所描述的情况，它们会产生聚类簇。由于簇的连续度小于60这个门限值，因此无法形成有效簇。如果将阈值60改为30(相当于10个条带长度)，有效簇的数量会增加21.3%。当阈值处于60和100之间时，有效簇的数量基本保持稳定，说明阈值为60的这个取值是可行的。

5.2 预测

在预测的测试中，本文主要利用原有的I/O请求特性来预测下一时刻的I/O请求特性，因此对*ARMA*模型来说，只是一步预测。预测的时间序列是前面测试中的聚类结果。图8(b)中的实线反映的是35号卷时间跨度为191.11h的152616个I/O请求所形成的937个聚类中心点结果，它具有线性递增的趋势，是非平稳的时间序列。利用式(12)，可计算出其差分阶数为1。图7表示的是时间序列经过1阶差分并去均值化后的自相关函数曲线图，它是一个快速的衰减过程，表明这时的序列是平稳的。图8(a)、(b)是一步预测值和实际值的对比。图8(a)是前40个数据的预测对比结果。当采用常规预测方法时，从第20个预测值开始，误差开始增大，匹配率只有67.25%；而采用动态参数估计策略后，其误差值减少很多，匹配率达到89.31%。一般认为，当匹配率达到85%以上时，预测对系统具有很好的指导性。图8(b)则是整个测试的结果对比图。在时间跨度为191.11h的测试中，采用动态参数的预测模型能很好的预测下一步聚类中心点值，预测值的匹配率达到86.45%。

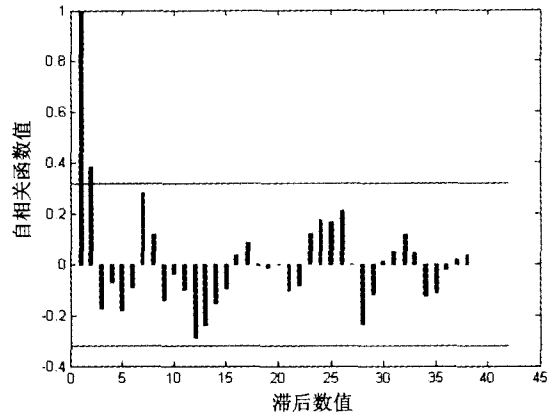


图7 数据平稳化后的自相关函数图

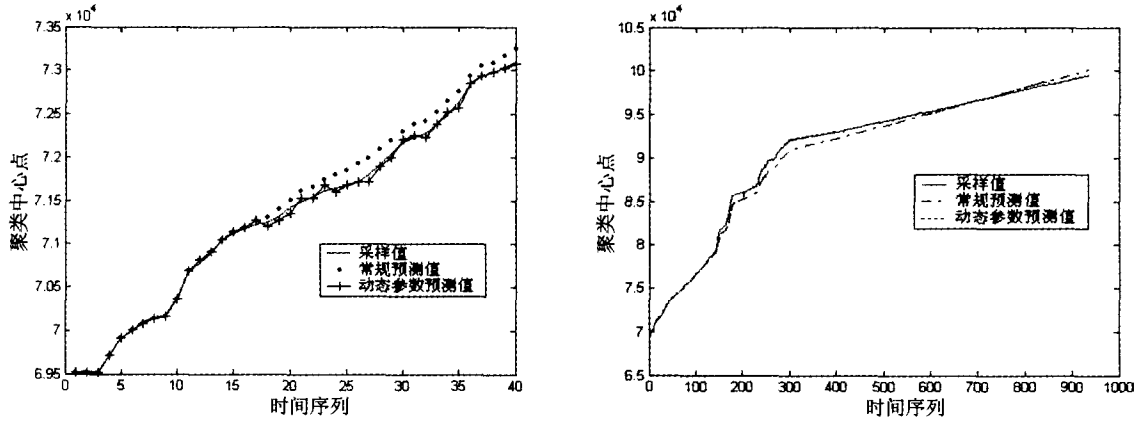


图8 聚类中心点预测结果对比图

总结和展望 本文采用基于连续度的聚类算法和动态参数估计的ARMA时间序列模型发现和预测存储系统中的密集I/O访问区域，并通过实验验证了两个算法的正确性和预测的准确性。这种算法不仅可以应用于优化存储系统的数据分布上，还可以应用于提高存储系统的缓存预取效率上，这正是我们今后的工作。

参考文献

1 Gibson D A, Katz R H. A case for redundant arrays of inexpensive disks (RAID). In: Proceedings of International Conference on Management of Data (SIGMOD), Chicago, Illinois, 1988. 109～116

2 Han Jiawei, Kamber M. Data Mining Concepts and techniques. China Machine Press, 2001

3 Georgiev I K, Georgiev I I. An Information-Interconnectivity-Based Retrieval Method for Network Attached Storage. In: CF'04, Ischia, Italy, April 2004. 268～275

4 Mesnier M, Thereska E, Ellard D, et al. File Classification in self-* storage systems. CMU-PDL-04-101, Jan. 2004

5 Pentakalos O I, Menasce D A, Yesha Y. Automated Clustering-Based Workload Characterization. In: Proc 5th NASA Goddard Conference on Mass Storage Systems and Technologies, College Park, MD, Sept. 1996. 253～263

6 Tran N, Reed D A. ARIMA Time series Modeling and Forecasting for Adaptive I/O Prefetching. ICS'01 Sorrento, Italy, 2001. 473～485

7 Oly J, Reed D A. Markov Model Prediction of I/O Requests for Scientific Applications. In: Proc of the 2002 International Conference on Supercomputing, New York, June 2002. 147～155

8 Roselli D, Lorch J R, Anderson T E. A Comparison of File System Workload. In: Proc 2000 USENIX Conference, June 2000. 41～45

9 Li Guibin. Estimation of parameters in ARIMA model. Acta Mathematics Application Sinica, 1990. 173～192

10 Box G E, Jenkins G M, Reinsel G C. Time Series Analysis Forecasting and Control. third edition. 1997

11 Li Huai Yang, Xie Chang Sheng, Bin Cai, et al. A New Technique for Eliminating Data Migration in Logistic Evolution Storage System. In: The Fifth International Conference on Computer and Information Technology, Sep. 2005. 327～331