

Research on the Method of Neighbour Storage Based on Data Mapping Algorithm

Shanshan Li

School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu

Email: 18252063532@163.com

Received: Aug. 15th, 2016; accepted: Oct. 7th, 2016; published: Oct. 10th, 2016

Copyright © 2016 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With the high-speed development of the Internet, processing of high-dimensional and massive amounts of data for querying is a key challenge. However, for the traditional distributed storage scheme, such as the P2P network and Chord, the data storage capacity and the switch overheads from the nodes are increasing, thus decreasing the storage efficiency and data query efficiency continuously. In this article, a neighbor data storage approach based on data mapping algorithm is proposed. The experiment results show that the proposed method can improve the query accuracy rate and reduce network bandwidth through relevant query.

Keywords

P2P, Distributed-Memory, Neighbor Data Storage, Relevant Query

基于数据映射算法的近邻存储方法研究

李姗姗

南京邮电大学通信与信息工程学院, 江苏 南京

Email: 18252063532@163.com

收稿日期: 2016年8月15日; 录用日期: 2016年10月7日; 发布日期: 2016年10月10日

摘要

随着互联网的高速发展, 如何有效地存储海量数据以提供高效的查询效率是一项亟待解决的关键问题。

文章引用: 李姗姗. 基于数据映射算法的近邻存储方法研究[J]. 数据挖掘, 2016, 6(4): 139-147.

<http://dx.doi.org/10.12677/hjdm.2016.64016>

然而，采用现有的例如Chord和P2P等分布式存储方案，面对高维、海量的存储数据时，数据存储规模和开销不断增加，造成存储效率以及数据查询效率不断降低。本文提出了基于数据映射算法的近邻存储方法。实验表明当进行相关性查询时，提高了查询准确率，同时显著降低了网络带宽。

关键词

P2P, 分布式存储, 近邻存储, 相关性查询

1. 引言

随着互联网的发展越来越快，海量数据指数级增长的时代已经到来。海量资源的爆炸式的增长，对于现有的存储系统是一个极大的考验。面对高维、海量的数据，如何有效地存储海量数据同时能够高效的查询到有效的数据信息是一个值得研究的问题。

为了更高效的解决海量数据下存储问题，分布式存储应运而生。主流的分布式存储技术是 P2P 系统 [1]。根据 P2P 技术的点对点和分布式特点，将其加入到云存储系统中，有效地解决集中式云存储系统的中心服务器瓶颈问题。将数据分布式的存放在各个节点服务器上，减轻了节点服务器的负载，实现了云存储系统的负载均衡，从而提高了系统硬件的使用率，大大提高了系统的存储性能。在主流的 P2P 存储网络中，存储数据在 DHT 中的哈希值 Key 由相应的哈希函数获得，然而哈希函数为了保持网络的负载均衡会破坏资源的语义相关性 [2] 信息，如 MD5，因此大部分 P2P 系统只支持单关键字的精确查询 [3]，不能支持语义相关性查询 [4]。面对海量数据时，查全率低，节点切换开销变大，导致网络拥堵，造成系统的存储效率和查询效率会大大降低。为了解决分布式存储系统中存储数据语义相关性低和查询效率低等问题，本文提出了基于数据映射算法的近邻存储方法 [5]。

2. 相似性度量方式

无论在原始欧式空间还是在汉明空间都需要去衡量两个数据之间的相似性 [6]。数据间的相似性需要通过一定的方法进行度量，相似度是衡量两个对象之间相似性的指标 [7]，取值在 [0,1] 之间，目前常用的相似性度量方法主要有余弦相似度、海明距离、Jaccard 相似度、Minkowski 距离、马氏距离、负指数法及正切值法：

假定有一些点组成的集合称为空间(space)，该空间下的距离测度(similarity measure)是一个函数 $d(x, y)$ ，以空间两个点 x, y 作为参数，输出是一个实数值。该函数必须满足如下准则：

- (1) $d(x, y) \geq 0$ (距离非负)；
- (2) $d(x, y) = 0$ 当且仅当 $x = y$ (只有点到自身的距离为 0，其他距离都大于 0)；
- (3) $d(x, y) = d(y, x)$ (距离具有对称性)；
- (4) $d(x, y) \leq d(x, z) + d(z, y)$ (三角不等式)。

三角不等式是上述条件中最复杂的条件。它的直观意义是，如果从 x 点行进到 y 点，那么一定不存在第三点 z 使得途径 z 的距离更近。三角不等式准则使得所有的距离测度表现如同其描述，为从一个点到另一个点的最短路径长度。

(1) 海明距离

海明距离(Hamming Distance)，是一种衡量两个二进制编码序列之间的相似度的指标。给定 v_1^b, v_2^b 两个长度为 L 且是任意的二进制向量，其海明距离定义为两者对应位置上的不同编码值的个数，即式(2.1)：

$$\text{dist}(v_1^b, v_2^b) = \sum_{i=1}^L v_1^b[i] \neq v_2^b[i] \quad (2.1)$$

海明距离直接反映的是二进制序列的差异程度，由于海明距离越大两个序列的相似度越低，所以实际用于衡量相似度的时候需要进行适当的变换。

(2) 范数距离

对两个向量 $x, y \in \mathbb{R}^d$ ，其范数距离(又称闵可夫斯基距离(Minkowski Distance)，简记为 L_p)定义为式(2.2):

$$L_p(x, y) = \left(\sum_{i=1}^d |x_i - y_i|^p \right)^{1/p} \quad (2.2)$$

当 $p=1$ ， L_1 范数称为曼哈顿距离，实际上是两个向量每维距离绝对值之和；当 $p=2$ 时， L_2 范数称为欧氏距离，广泛应用在各类数据挖掘任务的相似性计算当中；当 $p=\infty$ ， L_∞ 范数是对象属性之间的差值最大那一维的差值的绝对值，定义为式(2.3):

$$L_\infty(x, y) = \max_{1 \leq i \leq d} |x_i - y_i| \quad (2.3)$$

3. 基于数据映射算法的近邻存储方法的提出

3.1. 理论基础

本文所要解决的技术问题在于克服现有技术的不足，提供一种基于数据映射算法的近邻存储方法。该方法将谱哈希算法与数据映射算法相结合，将数据映射到分布式存储网络中，解决在大规模数据场景下分布式存储和相关性查询时系统查询开销过大的问题，提高数据的查询效率。如图 1 为映射流程图。

本方法主要分三个层次，分别为：利用谱哈希进行哈希映射，数据映射算法，Chord 的构建，具体实施步骤包括如下：

步骤 1：利用谱哈希算法对高维数据样本哈希映射获得每个高维数据项的 k 维二进制哈希码；

步骤 2：根据步骤 1 得到哈希桶编号 v ，根据 Z-curve 方法计算出 Z-Value 值；

步骤 3：利用 Chord [8] 方法构建分布式节点网络，所述分布式节点网络由 Chord 环和分布于 Chord 环上的 m 个节点服务器构成；将所得每个高维数据项的 Z-Value 值映射和存储至 Chord 环上的节点服务器，并更新获得已存储高维数据项的 Z-Value 值的节点服务器路由表。

3.2. 实现方法

根据上文描述的流程可知，谱哈希先将 d -维数据映射到各个桶中，然后利用 Z-curve 将各个桶映射到各个分布式网络节点上。这样既能保证拥有相似数据的桶映射到同一节点上，也能保证各个节点之间负载均衡。

由谱哈希定义[4]可知，相似的数据极有可能映射到相近哈希桶中，再通过使用 Z-curve 函数，相似的桶编号有极大概率被映射为相近的整数，即这些整数之间拥有较小的 l_1 距离。这样可以认为 Z-curve 函数能够表征哈希桶之间的 l_1 距离，即两个哈希桶中包含相似的高维数据点，那么对应哈希桶在分布式存储中的位置越相近，从而实现近邻存储。

(1) 利用谱哈希进行哈希映射

谱哈希算法是将谱分析技术融合进哈希函数中，将构造的新的哈希函数用于高维数据的处理上。谱哈希首先需要对高维数据样本进行谱分析，利用特征函数,通过分析高维数据样本的平均分布，生成一个哈希函数。因此谱哈希算法的动机就是给定一个高维样本数据库进行训练，找到一个理想的哈希函数。

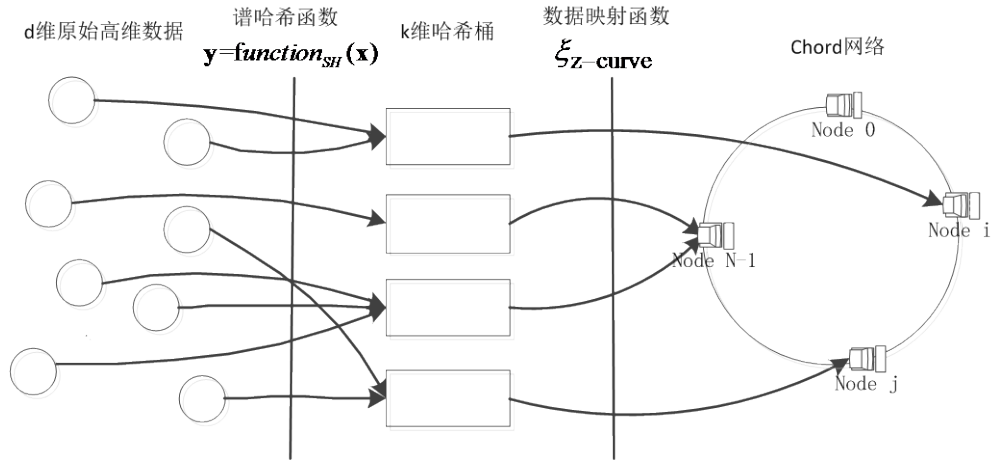


Figure 1. The flow chart of the mapping
图 1. 映射流程图

这个哈希函数可以将相似的高维数据点哈希到一个相似的值上。如下即为谱哈希训练得到的函数：

根据假定有个训练集 S ，它有 N 个 d 维的高维原始数据，这里用 $\mathbf{X} \in \mathbb{R}^{N \times d}$ 来表示训练集在原始欧式空间中的表达形式，相对应的将所有高维原始数据映射到汉明空间的结果为 $\mathbf{Y} \in \{-1, 1\}^{N \times m}$ ，即哈希码。

对于每一个样本 $\mathbf{x} \in \mathbb{R}^d$ 而言，其 k 维哈希码 $\Psi(\mathbf{x})$ 可由如下哈希函数得到：

$$\Psi(\mathbf{x}) = \left[\text{sgn}\left(\Phi_{(i_1, \theta_1)}(x_{i_1})\right), \dots, \text{sgn}\left(\Phi_{(i_k, \theta_k)}(x_{i_k})\right) \right]^T \quad (3.1)$$

其中 $\lambda_{(i_1, \theta_1)} \leq \dots \leq \lambda_{(i_k, \theta_k)} \leq \dots \leq \lambda_{(i_k, \theta_k)}$ ， x_{i_1} 是 \mathbf{x} 的第 i_1 个元素。

因此对于新来的数据项 $\mathbf{x}_{n+j} \in \mathbb{R}^d$ ，该数据项的 k 位哈希码 \mathbf{y}_{n+j} 可由如下函数得到：

$$\mathbf{y}_{n+j} = \Psi\left(\left(\mathbf{x}_{n+j}\right)^T \mathbf{P}\right) \quad (3.2)$$

(2) 新型数据映射算法

本文提出了一种新型数据映射算法。该算法利用 Z-curve 的方法将 k 维向量转化为一个 Z-Value 值，具体过程如图 2 所示。将通过谱哈希得到的 k 维向量中的每一个数中的第一个二进制位，按向量中的位置顺序排列。然后排列每个整数的第二个二进制位，也按照向量中的位置顺序排列，一直重复上述的过程直到向量中所有数的二进制位都被处理过之后，将这些二进制序列串联在一块就生成一个 Z-Value 值。

由相关理论可以证实 Z-curve 方法能够有效的保持高维数据的相似性。这里提出了 LLCPC 值，LLCPC 表示最长公共前缀字符串，而且可以知道两个 Z-Value 值的 LLCPC 越大，那么这两个 Z-Value 所代表的高维数据越相似。因此可以利用 Z-Value 的值将高维原始空间数据通过谱哈希生成的 k 维向量映射到分布式存储网络中。

由本文的算法可知，如果高维原始数据相似，则映射到分布式存储网络中的位置也相近，即能将相似的数据存储到相近的节点服务器。这说明用 Z-Value 表示高维数据在 Chord 环上的位置是有意义的。

(3) 分布式节点网络的构建

Chord 由麻省理工学院(MIT)在 2001 年提出，其目的是提供一种能在 P2P 网络快速定位资源的的算法。Chord 通过把 Node 和 Key 映射到相同的空间而保证一致性哈希。为了保证通过哈希函数得到的哈希码不具有重复性，选择 SHA-1 作为哈希函数，它会产生一个 2160 大小的空间，环上的每一项均为一个 16 字节的大整数。这些整数首尾相连形成一个环，称之为 Chord 环。整数在 Chord 环上按大小顺时针排

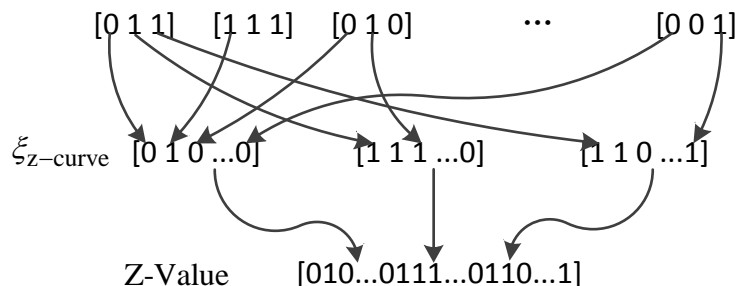


Figure 2. Z-curve method
图 2. Z-curve 方法

列, Node (机器的 IP 地址和 Port)与 Key (资源标识)都被哈希到 Chord 环上,这样就假定了整个 P2P 网络的状态为一个虚拟的环。

利用 Chord 方法构建分布式节点网络,所述分布式节点网络由 Chord 环和分布于 Chord 环上的 m 个节点服务器构成;将所得每个高维数据项的 Z-Value 值映射和存储至 Chord 环上的节点服务器,并更新获得已存储高维数据项的 Z-Value 值的节点服务器路由表。

由于 Chord 是环形构造,通过线性扫描能找到目标节点,然而这样显然开销过大,开销为 $O(N)$ 。因此 Chord 实际使用的是一种类似折半查找的非线性查找方法。

另外,由 Z-Value 生成的二进制字符串中,LLCP 的值越长,那么对应的高维数据点越相似。因此假定对于一个高维数据集而言其 Z-Value 的值分布为 $[u, l]$ 。那么可以将这个区间的值平均分配到 m_0 至 m_7 节点中,如图 3 所示,并更新获得每个存储有数据项的节点服务器路由表。

在图 3 中, Z-Value 的值表达范围为 $[0, 127]$,那么假设得到的 Z-Value 值为 Z_i ,根据 Z_i 可以对应数据按照顺时针方向分配到相近的节点服务器。即第一个索引节点负责 Z-Value 值为 $[0, 15]$ 的高维数据点的索引,即假设得到的 Z-Value 值为 Z_i , Z_i 值小于 15 的都存放于第一个节点服务器;而第二个索引节点负责 Z-Value 值为 $[16, 31]$ 的高维数据节点的索引, Z_i 值在 $[16, 31]$ 之间的都存放于第二个节点服务器;以此类推,每一个落在一定区间的高维数据,由对应的索引节点负责建立索引。当有数据映射到 Chord 环上时,可以根据 Z-Value 按顺时针顺序查找到相应的节点服务器,实现近邻存储。

根据待查询高维数据项进行相关性查询,包括:确定待查询高维数据项的 Z-Value 值,及根据所确定 Z-Value 值按顺时针查找各节点的节点服务器路由表找到高维数据项所在的节点服务器。具体如下:

从上文可知,假设得到的 Z-Value 值为 Z_i ,在这种环状结构中,随机的给 $\mathbf{a} \cdot \mathbf{y}_i$ 第 i 节点服务器分配一个随机值 r_i ($r_{\min} \leq r_i \leq r_{\max}$),这个值被称作节点编号,用来代替其在环中的位置,其中 r_{\min} 和 r_{\max} 根据测试的数据集映射后最小值和最大值来确定。所有的节点按照节点编号升序的方式组成一个环,一共有 m 个节点。同时,每个数据项 $\mathbf{x}_i \in \mathbb{R}^d$ 都有唯一标识 Z_i ,根据映射函数 $\theta(\cdot)$ 可以将标识为 Z_i 分配给节点编号为 r_i 的节点服务器,该函数定义如下:

$$\theta(Z_i) = r_i, r_{i-1} < Z_i \leq r_i \quad (3.3)$$

当进行相关性查询时,存在一个待查询高维数据项,根据一致性哈希原理,首先获得查询项唯一标识,即 Z-Value 值,然后沿着环顺时针查找,直到找到满足节点的哈希值大于或等于查询项哈希值的第 1 个节点服务器,它就是所要查询的节点。这就是说,每一个节点标号为 r_i 的节点服务器负责环上 $[r_{i-1}, r_i]$ 这个范围。

(4) 负载均衡

在分布式存储网络中,常出现系统热点问题,影响系统可靠性。因此要对分布式存储实现负载均衡。

即本文的方法还包括判断 Chord 环上各节点服务器的负载，根据判断结果对 Chord 环上的节点服务器进行负载调控步骤。这里提出了虚拟节点技术，通过添加虚拟节点的方法维持系统的负载均衡。具体做法如下：

如图 4 所示，矩形为节点服务器，该实例中包括 m1 至 m60 个节点服务器；圆点为资源标识符，其中的节点服务器直接根据其对应的资源标识符来表示；object 表示数据项，该实例包括 object1、object2、object3、object4 等若干个数据项。

若数据项 object1 存储到了节点服务器 m1 中，而 object2、object3、object4 都存储到了节点服务器 m18 中。由于过程中设置对每个节点服务器设定一个负载阈值，若该节点服务器的负载超过负载阈值，则进行调控。本实例中负载阈值设置为 2，而节点服务器 m18 的负载数量为 3 超出负载阈值 2，则出现明显过载的情况，此时需要对负载进行调控。

将实际物理上的节点服务器按照网络负载情况从逻辑上分为多个逻辑节点，每个逻辑节点可以视作一个虚拟节点。一个物理上的节点服务器可以对应 2 个甚至更多个虚拟节点。这里设置每个物理上的节点服务器分为两个虚拟节点，此时节点服务器 m1 对应有两个虚拟节点服务器 m1-1, m1-2，m18 对应有两个虚拟节点服务器 m18-1，m18-2。此时，对象到“虚拟节点服务器”的映射关系为：

object1->m1-1; object2->m1-2; object3->m18-2; object4->m18-1;

因此数据项 object1 和 object2 都被映射到了节点服务器 m1 上，而数据项 object3 和 object4 映射到了

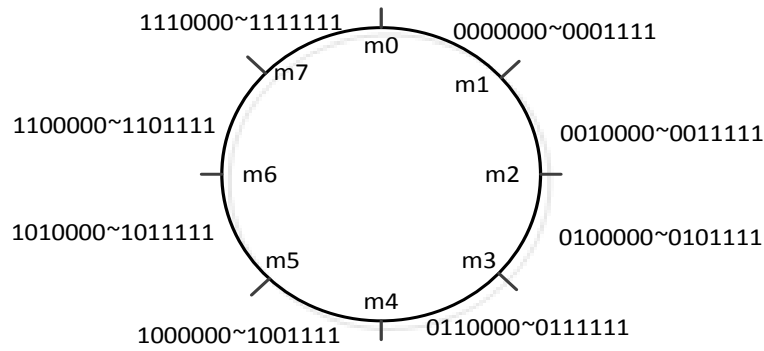


Figure 3. Z-Value distribution of the ring
图 3. Z-Value 在环上分布

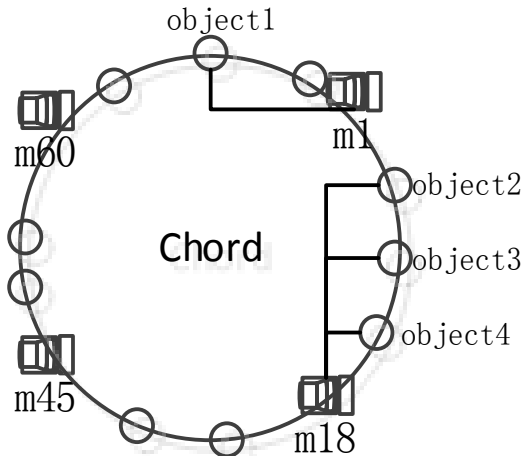


Figure 4. Distributed storage architecture diagram
图 4. 分布式存储(chord)架构图

节点服务器 m18 上，系统负载均衡性有了很大提高。

4. 实验验证

本节主要目的主要是为了验证在分布式存储环境下，本文提出的算法是否实现了数据的近邻存储。实验部分将本文提出的算法与 Chord 相比较，验证系统相似度和系统切换开销两个指标对两个算法的影响。

仿真程序采用 JDK1.7，实验平台为 Eclipse，操作系统 32 位 Windows7，硬件环境为 Intel i3-3230 处理器，4G 内存。所用数据库为 Oracle 11i。

系统相似度(Degree of Similarity, S): 表示整个系统的相似度;

系统切换开销(Number of Access): 相关性查询时系统节点之间的切换开销。

这里设定谱哈希的哈希比特数为: 8,16,32。节点服务器个数为 300。

在相同的实验环境下，将本文算法与 Chord 算法比较。由前文可知，Chord 是基于 DHT 的典型 P2P 存储网络。Chord 支持大规模和分布式存储以及查询信息。

由图 5 可以看出，当把数据直接映射到 chord 环上时，由于没有考虑数据之间相关性，分布式存储系统的相似度比较低，节点服务器内的数据和节点服务器之间的数据相似度较低。而与之相反的是，在哈希比特取不同情况下，本文提出的方案要远远优于 chord。这是因为在存储时考虑了数据的相关性，通过一定的算法将数据映射到存储网络中，大大提高了数据在分布式存储网络中的相关性。

由图 6 可知，当进行相关性查询时，本文提出的方案明显优于 chord 方案。而且当查询集越大时，chord 网络中查询的节点开销剧烈增加，这是由于存储时没有考虑数据相关性。同样，本文提出的方案随着查询集变大，开销增长不多。

5. 总结

在大数据时代，面对海量数据，数据的分布式近邻存储已经成为时下热门的研究课题。本文针对海量数据分布式存储系统中数据的语义相似性较低的问题，提出了基于数据映射算法的近邻存储方法。本

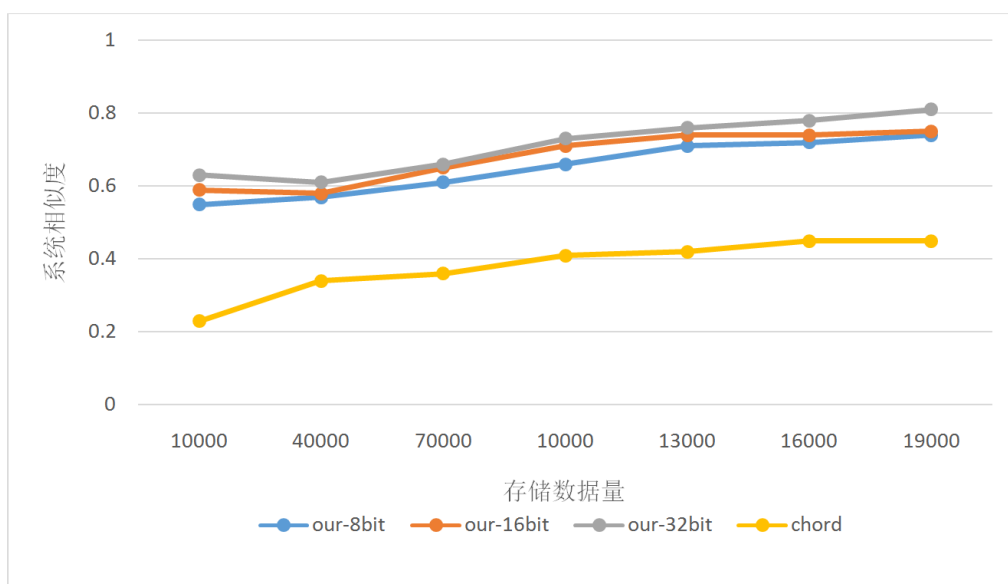


Figure 5. Similarity degree between systems

图 5. 系统相似度

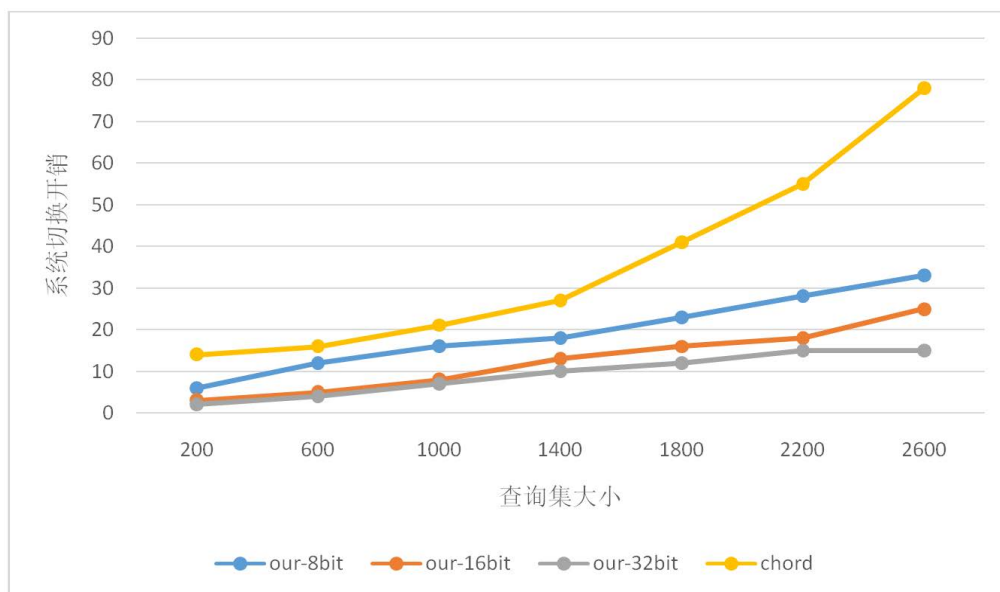


Figure 6. System switching expenditure
图 6. 系统切换开销

文主要工作总结如下:

1) 针对海量数据下分布式存储系统中相似性查询查全率较低等问题, 提出了近邻存储这一创新性概念。基于这个理念, 提出了一种基于数据映射算法的近邻存储方法, 该方法利用谱哈希哈希映射算法和数据映射算法将高维原始数据映射到分布式存储空间, 实现了分布式近邻存储。

2) 为了能够更好的将相近原始数据映射到邻近的服务器节点中, 提出了一种基于 Z-curve 的数据映射方法。该方法从原理证明了在原始空间相近的数据能够更好的映射到分布式存储系统的邻近的服务器节点, 保证了在各个空间中数据的相似性。

3) 为了防止出现系统热点问题, 影响系统可靠性, 通过虚拟节点的方法, 动态的分析和移动各个节点的负载, 有效了降低了各个节点的负载, 减轻了系统的过载情况, 实现了系统的负载均衡。

4) 最后通过实验可以证明, 在多个不同指标下, 本文提出的方案在系统相似度和系统切换开销均优于 chord。通过本文实验可以说明本方案在分布式网络中实现了分布式近邻存储, 在大数据领域具有较大的应用价值。

参考文献 (References)

- [1] Svendsen, H.B. and Erickson, M. (2016) System and Method for Identifying Music Content in a P2P Real Time Recommendation Network. http://xueshu.baidu.com/s?wd=paperuri%3A%2879696d30bc0034769b511c43747ca6ab%29&filter=sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Fwww.freepatentsonline.com%2F8422490.html&ie=utf-8&sc_us=3331819690017585512
- [2] Li, P., Wang, M., Cheng, J., et al. (2013) Spectral Hashing with Semantically Consistent Graph for Image Indexing. *IEEE Transactions on Multimedia*, **15**, 141-152. <http://dx.doi.org/10.1109/TMM.2012.2199970>
- [3] 彭良睿, 李学明. 一种基于树型结构的 P2P 系统高维数据检索方法[J]. 计算机应用研究, 2015, 32(3): 842-845.
- [4] Yao, C., Bu, J.J., Wu, C.X., Chen, G.C., et al. (2013) Semi-Supervised Spectral Hashing for Fast Similarity Search. *Neurocomputing*, **101**, 52-58. <http://dx.doi.org/10.1016/j.neucom.2012.06.035>
- [5] Washbourne, L. (2015) A Survey of P2P Network Security. arXiv:1504.01358
- [6] Shao, J., Wu, F., Ouyang, C., et al. (2012) Sparse Spectral Hashing. *Pattern Recognition Letters*, **33**, 271-277. <http://dx.doi.org/10.1016/j.patrec.2011.10.018>

-
- [7] Zou, F., Liu, C., Ling, H., *et al.* (2013) Least Square Regularized Spectral Hashing for Similarity Search. *Signal Processing*, **93**, 2265-2273. <http://dx.doi.org/10.1016/j.sigpro.2012.05.033>
- [8] Stoica, I., Morris, R., Karger, D., *et al.* (2001) Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *Proceedings of the 2001 SIGCOMM*, **31**, 149-160. <http://dx.doi.org/10.1145/383059.383071>

期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: hjdm@hanspub.org