

位置相关查询中基于最小访问代价的缓存替换方法

卢秉亮,梅义博,刘娜

(沈阳航空航天大学 计算机学院,沈阳 110136)

(lubing_liang@163.com)

摘要:在位置相关查询(LDQ)中由于用户的移动性和数据的位置相关性,给缓存替换策略带来了新的挑战。在详细分析位置相关数据(LDD)的空间位置特性和几种典型的位置相关缓存替换策略的基础上,提出一种基于最小访问代价的缓存替换策略(PLAC),一些重要的缓存替换因素如访问概率、更新频率、数据距离和有效范围等都包含在代价函数里,PLAC根据代价函数值的大小来决定被替换的数据,由此来保证有限缓存的最大利用率。通过实验对比,PLAC比其他位置相关缓存替换策略更为有效地提高了缓存命中率,缩短了查询平均响应时间。

关键词:移动计算;位置相关数据;位置相关查询;缓存替换

中图分类号: TP311.13 **文献标志码:** A

Cache replacement method based on lowest access cost for location dependent query

LU Bing-liang, MEI Yi-bo, LIU Na

(School of Computer, Shenyang Aerospace University, Shenyang Liaoning 110136, China)

Abstract: Because of the user's mobility and the location dependency of data, new challenge has been brought to cache replacement strategy for Location Dependent Query (LDQ). Based on the detailed analysis of the space location characteristics of Location Dependent Data (LDD) and several typical replacement strategies of location dependent cache, the authors proposed a prioritized approach cache replacement based on the lowest access cost (PLAC), the PLAC took some important factors into account such as access probabilities, update rates, data distance, valid scope. To ensure the maximum utilization of limited cache, the PLAC cache replacement strategy decided which data would be replaced according to the value of the lowest cost function. The contrast experiments show that the PLAC increases cache hit rate and shortens query average response time more effectively than other location dependent cache replacement strategies.

Key words: mobile computing; Location Dependent Data (LDD); Location Dependent Query (LDQ); cache replacement

0 引言

移动计算环境中移动客户机(Mobile Client, MC)受低带宽、弱连接、网络断接性和有限本地资源的困扰,使得其执行位置相关查询(Location Dependent Query, LDQ)时延长,服务质量不高。一种有效的办法就是把用户希望得到的位置相关数据(Location Dependent Data, LDD)预取到MC的本地缓存中^[1],提供高质量的服务,然而受本地缓存容量的限制,常常没有足够的空间来存储新的数据,这就需要替换缓存中那些不经常被使用的LDD,以提高缓存的利用效率。

但是由于LDD的空间位置特性给数据缓存研究带来了新的挑战:首先,一次查询(例如:查找最近的饭店)的缓存结果可能由于用户从一个位置移动到另一位置而变得无效;其次,缓存替换策略必须考虑这些缓存结果的有效范围区域。当数据值的有效范围较大时,用户在有效范围内执行相同查询的机会也较大,也就是产生的缓存命中率较大。所以,缓存替换策略必须尽量保留有效范围较大的数据在缓存中。例如:一个移动的用户查找所在区域的天气情况。如果返回有效范围为10 km²的天气预报数据,也就是说,在这个区域内,任何查找所在区域内天气预报数据的查询都将返回相同的结

果。跟有效范围只有1 km²的情形相比较,意味着很短时间内又移动至新的区域,明显前者的结果在用户执行相同查询后有较大的可能成为有效答案,因此需要保存在缓存中。

一个MC在移动中不断提交与自己位置相关的查询,那么不同时刻的查询结果有很大一部分是位置相关的^[2],这就要求在充分考虑客户机的移动特性同时,还要考虑LDD的空间位置特性,探索反映该特性的缓存替换策略,以适应LDQ。

1 相关定义和已有的位置相关缓存替换策略

1.1 相关定义

定义1 位置相关数据(LDD)。是指其值取决于具体地理位置的数据。

例如:本地黄页、事件、宾馆、酒店等具有位置相关属性的数据都属于LDD。设 D_i 为任意一个LDD,定义 $Val(D_i, L)$ 为 D_i 在位置 L 处的值,可见LDD在不同的位置可能有不同的值。

定义2 给出一组数据实例 $D = \{D_1, D_2, \dots, D_n\}$ 及其相对应的一组有效范围 $R = \{R_1, R_2, \dots, R_n\}$,LDD查询是指对于查询 Q ,当且仅当 Q 落在 R_i 时,从 D 中返回数据实例 D_i ,其中 n 为数据实例的总数。

如图1所示,当查询 Q 落在范围 R_4 时返回的数据实例是

收稿日期:2010-09-10;修回日期:2010-10-30。

作者简介:卢秉亮(1953-),男(满族),辽宁本溪人,副教授,硕士,主要研究方向:数据库系统;梅义博(1982-),男,陕西山阳人,硕士研究生,主要研究方向:数据库系统;刘娜(1984-),女,山东泰安人,硕士研究生,主要研究方向:数据库系统。

D_4 。

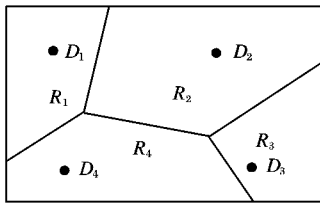


图1 位置相关数据及其数据区域

定义3 数据区域(Data Region)。它是一个数据实例有效范围的空间表示。

对于整个地理区域 G 和一个数据区域 R_i , 若以下两个条件: 1) $\bigcup_{i=1}^N R_i = G$; 2) $\forall i, j (i \neq j), R_i \cap R_j = \emptyset$ 成立, 则称 R_i 为 G 的一个数据区域, 可见一个数据区域对应一个数据实例。

定义4 数据的有效范围区域(Valid Scope Area)。指数数据实例有效范围的几何区域。

每个 LDD 实例有一个特定的有效范围, 只有在此有效范围之内, 该实例才是正确的。可以在没有先验的用户移动模式的前提下, 对同一项的不同数据实例最精确地估计出被访问的可能性, 也就是说, 数据的有效范围区域越大, 用户访问该数据的可能性也就越高。因为, 通常相对于较小的区域, MC 有更大可能性位于某个较大的区域。因此, 若仅仅考虑这个因素, 拥有更大有效范围的数据实例是较好的选择。

定义5 数据距离(Data Distance)。指 MC 当前位置和数据实例有效范围之间的距离。

当一个数据实例的有效范围距 MC 当前位置还很远时, 数据将有很小的机会被重用, 因为 MC 还要经过一段时间才能“再次”进入有效范围区域, 而在进入有效范围区域之前, 数据实例对 MC 来说是无效的。从这方面考虑, 应该倾向于替换掉“最远”的数据。

1.2 已有的位置相关缓存替换策略分析

基于时态位置的缓存替换策略^[3], 如最久未使用(Least Recently Used, LRU)策略、最不经常使用(Least Frequently Used, LFU)策略和最近 k 次未使用策略(LRU- k)^[4], 已经被广泛地研究。这些方法都是基于以下假设: 最近被访问过的数据项在未来很有可能被再次访问, 过去被频繁访问过的数据项将来也会继续被频繁访问。但是这些方法都未考虑到数据的空间位置特性和 MC 的移动性, 因此它们对于 LDQ 中的缓存管理, 实用性不大。为了解决这个问题, 很多学者已经提出了适用于位置相关服务的缓存替换策略^[5-6]。

基于数据距离的位置相关缓存替换策略^[7], 如 Manhattan Distance 策略和 FAR(Further Away Replacement)策略^[8], 这两种方法在替换时删除离 MC 当前位置最远的数据。Manhattan Distance 策略只考虑了数据距离, 但是其忽略了 MC 的当前位置和移动方向。FAR 策略考虑了 MC 的当前位置和移动方向, 将数据分为 In-direction 和 Out-direction 两个集合, 每次优先替换 Out-direction 集合中离 MC 距离最远的数据, 但是忽略了 MC 访问数据项的其他因素(比如访问概率), 当 MC 移动方向频繁变化时, 使得需要访问的数据在两个集合频繁被替换, 导致替换效率低下。

基于代价的位置相关缓存替换策略, 如概率区域(Probability Area, PA)策略和概率区域反距离(Probability Area Inverse Distance, PAID)策略^[9], 当缓存空间满时, 将计算出缓存中每个数据项的代价值, 值最小的数据项将被替换。

PA 策略中的代价模型考虑了访问概率和数据有效范围的面积这两个因素, 将代价函数定义为: $Cost(i) = P_i \times A(i)$, 其中: P_i 为访问概率, $A(i)$ 为数据的有效范围, PA 策略选择代价最小的数据作替换。而 PAID 策略将 PA 策略进行扩展, 引入了 MC 到数据有效范围的距离, 其代价函数定义为: $Cost(i) = P_i \times A(i)/D(i)$, 其中 $D(i)$ 为 MC 当前位置到数据有效范围的距离, 与 PA 相同, PAID 在替换中选择最小代价的数据。虽然实验结果表明 PAID 策略胜过现有的同类方法, 但是它和 PA 一样, 在访问概率中加入了访问时间因素的考虑。但是其他的因素, 如重新获取数据项的开销、数据项的大小、数据的更新频率都没有考虑, PAID 策略的一个很大缺陷是因有效范围比较小而导致离 MC 近的数据常常被替换掉。

在 LDQ 中, 由于提交的查询是与当前的位置密切相关的, 因此需要将这些可能被再次访问的 LDD 保留在缓存中, 提高缓存利用率。一个好的替换方法能保证较高的缓存命中率和较短的响应时间, 应考虑以下影响缓存利用效率的关键因素: 访问概率, 更新频率, 数据项大小, 重取延迟, 缓存保鲜代价等。综合考虑以上因素, 本文提出一种基于最小访问代价优先的缓存替换策略(prioritized lowest access cost based cache replacement policy, PLAC)。

2 基于最小访问代价优先的位置相关缓存替换

2.1 位置模型

本文采用几何模型表示位置^[10], 位置用二维坐标来确定, 例如由全球定位系统(Global Positioning System, GPS)来返回经纬度坐标, 或者用于描述区域边界几何形状的坐标集合, 系统根据精确的表示(例如用圆心和半径来表示区域)来计算规则的几何形状大小。MC 用自带的 GPS 来确定自己的位置, 可以在无线网络覆盖的区域自由移动, 用 $L_{MC} = (l_x, l_y)$ 表示 MC 机的位置, $v_{MC} = (v_x, v_y)$ 表示 MC 的移动速度, 其中 v_x/v_y 为速度在水平、垂直方向上的分量。

数据的有效范围表示为数据项值的有效范围的几何区域, 数据项值不同于数据项, 数据项值是某一项数据实例在特定区域的有效值。例如, “最近的餐馆”就是一个数据项, 那么其值随着查询位置(区域)的不同而不同。本文用近似圆来描述数据的有效范围, 数据项 d_i 的有效范围 $vs(d_i)$ 用内切圆的中心和半径近似表示为 $vs(d_i) = (L_i, Range_i)$, 其中: L_i 为数据有效范围的中心位置, $Range_i$ 为半径, $Range_i$ 是有效范围边界到中心位置的最大距离。数据有效范围区域的大小表示为 $Area(vs(d_i)) = \pi \times (Range_i)^2$, 当一个数据从服务器发送给用户时包含了该数据的有效范围, 系统可以针对自身位置的变化来验证数据的有效范围区域。

2.2 系统模型

本文使用基于请求的数据广播模型, 服务器上的数据库由 N 个数据 d_1, d_2, \dots, d_N 组成, 服务器负责维护数据库和响应 MC 的访问请求。一个 MC 的缓存存放频繁访问的数据, 当有 LDQ 时首先检查 MC 缓存是否有可用数据, 当需要访问的数据在本地缓存中找不到时, 就通过上行带宽向服务器发一个请求, 服务器一旦接收到 MC 的请求, 就通过广播通道推送数据。假设数据只在服务器端更新, 用 μ 表示单位时间内数据更新的频率, 用 λ 来表示单位时间内 MC 对服务器中 LDD 的访问频率。本文假设 MC 对服务器中数据项 d_i 的访问模型类似服从 Zipf 分布^[3], MC 访问数据项 d_i 的概率为 λ_i ; 而服务器中每个数据的更新是随机的, 假设数据更新模型服从均匀分布,

d_i 在单位时间内的更新频率为 μ_i 。在这个系统模型中缓存替换仅考虑两个方面的情况:首先,数据失效的验证报告通过广播通道,以一定的时间间隔定期地向客户端广播,MC 根据失效报告内容来决定替换那些过时失效的数据;其次,当 MC 的缓存因预取一个数据而满时,就根据 PLAC 来替换掉访问代价最小的数据项,以满足预取数据的缓存空间。

2.3 PLAC

需要对缓存进行替换时,根据提供的代价函数进行计算,选择代价最小数据项进行替换。在本策略中,选择那些最久未被访问,有效范围区域最小,距离最远,获取花费最小的数据项进行替换,为此本文引入一个代价函数来评判,函数值最小的数据项将被替换。这个代价函数考虑了三方面的因素,分别是时间代价,空间代价和获得该数据项的花费。用式(1)表示为:

$$Cost(i) = Score_{temp}(i) \times Score_{spat}(i) \times O_i \quad (1)$$

其中: $Score_{temp}(i)$ 为数据项 d_i 的时间代价, $Score_{spat}(i)$ 为数据项 d_i 的空间代价, O_i 为从服务器获取数据项 d_i 花费的代价。

PLAC 的目标 对于 MC 缓存中 LDD 集合 S ,寻找一个访问代价最小的数据集替换掉。用目标函数(2)描述:

$$\begin{cases} \min(\sum_{d_i \in S} Cost(i)) \\ \sum_{d_i \in S} DataSize(i) \geq NewDataSize(j) \end{cases} \quad (2)$$

式(2)的目的是使 $\sum_{d_i \in S} Cost(i)$ 取得最小值,且满足

$\sum_{d_i \in S} DataSize(i) \geq NewDataSize(j)$ 的数据项 d_i 替换掉,其中: $\sum_{d_i \in S} DataSize(i)$ 为替换掉的缓存空间, $NewDataSize(j)$ 为新取入的数据项 d_j 的大小。

2.3.1 时间代价

从时间的角度来考虑,越久未被使用的数据就越陈旧,被再次访问的可能性就越小;而越久未被更新的数据,说明其因服务器端的数据更新而导致缓存失效的可能性越小。因此,采用最后一次更新到现在的时间间隔与最后一次查询到现在的时间间隔的比值,来反映数据最近被访问且不易更新的时间代价,如式(3)所示:

$$Score_{temp}(i) = \frac{t_{current} - t_{u,i}}{t_{current} - t_{q,i}} \times \frac{\lambda_i}{\mu_i} \quad (3)$$

其中: $t_{current}$ 是系统当前时间, $t_{u,i}$ 是数据项 d_i 最后一次被更新的时间, $t_{q,i}$ 是 d_i 最后一次被查询的时间, λ_i 为访问 d_i 的频率, μ_i 为数据更新的频率。式(3)中,若 $t_{current} - t_{q,i} = 0$,则令 $(t_{current} - t_{u,i}) / (t_{current} - t_{q,i}) = 1$,即刚被访问过的数据很又可能被再次访问。若 $\mu_i = 0$,则令 $\mu_i / \lambda_i = 1$,即很久未被更新的数据在将来较短的时间内也不会被再次更新。

2.3.2 空间代价

从空间的角度来讲,根据文献[9],在 LDQ 时,数据有效范围的面积越大,距离用户当前位置越近,且越靠近 MC 当前移动方向或路径上的 LDD 越容易被再次访问。根据 LDD 的这一特性,本文以下三个条件来评判空间代价:

- 1) $|L_{MC} - L_i| < |L_{MC} - L_j|$, d_i 比 d_j 到 MC 的数据距离小;
- 2) $Area(vs(d_i)) > Area(vs(d_j))$, d_i 的有效范围比 d_j 大;
- 3) d_i 在 MC 的移动路径或方向上,而 d_j 不在。

若以上三个条件都成立,说明 MC 访问数据项 d_i 的可能性大于 d_j ,访问 d_i 的空间代价用式(4)表示:

$$Score_{spat}(i) = \frac{1}{|L_{MC} - L_i|} \times \frac{1}{|(v_{MC}/\lambda_i + L_{MC}) - L_i|} \times Area(vs(d_i)) \quad (4)$$

其中: $1/|L_{MC} - L_i|$ 为 MC 当前的位置到 d_i 有效范围的反距离,若 $|L_{MC} - L_i| = 0$,则令 $1/|L_{MC} - L_i| = 1$; $|(v_{MC}/\lambda_i + L_{MC}) - L_i|$ 为 MC 预定的位置到 d_i 有效范围的距离,若 $|(v_{MC}/\lambda_i + L_{MC}) - L_i| < |L_{MC} - L_i|$,说明 d_i 在 MC 的移动路径或方向上,且数据距离越来越小,反之说明 MC 将远离 d_i 。

2.3.3 获取数据的花费代价

获取一个数据的难易取决于该数据项的大小和当前的上传速率和下行速率。数据量越大,提交的请求越多,下载数据的速度越慢,则获取该数据的单位时间代价就越大。用式(5)表示:

$$O_i = \frac{DelayTime(i)}{DataSize(i)/BandWidth} \quad (5)$$

其中: O_i 为获取 d_i 花费的单位时间代价, $DelayTime(i)$ 为获取 d_i 的延迟时间, $BandWidth$ 是带宽的大小, $DataSize(i)/BandWidth$ 是 d_i 所占带宽的比例。

3 实验及性能分析

3.1 模拟实验的环境和参数设置

为了验证本文中提出的 PLAC,设计模拟实验进行测试并与其他位置相关缓存替换策略(如 FAR,PAID)进行对比。实验系统中有一个服务器和多个 MC,它们之间由无线网络连接。服务器负责维护数据库并采用按需广播的方式来提供数据传送服务。MC 发出查询请求时,先在本地缓存查找,若本地缓存不能回答,再将查询提交给服务器处理。服务器的数据库中有一个表,数据项的总数为1 000,数据访问模式和更新模式应用在这些数据项上,假设在整个访问过程中,80%的更新发生在20%的数据分区上,数据的访问模式服从 Zipf 分布,数据的访问频率、更新频率和数据项的大小之间都没有明显联系。

MC 在一个 $4\,000\text{ m} \times 4\,000\text{ m}$ 的地图上移动并随时发出 LDQ 请求,MC 的每两次查询之间有一定的时间间隔,在此查询间隔中,用户将改变速度和方向进行移动。这些变化受最大/小速度和最大角度变化的限制。客户端的缓存大小为 $CacheSize$,缓存参数包括了存储数据项语义说明和相关数据的存储空间。实验系统的主要参数设置如表1所示。

表1 系统参数

参数	参数描述	值
CilentCPU	客户端 CPU 速度/Mips	500
U/Dlinkband	无线网络的上/下行带宽/(Kbps)	19.2/144
CacheSize	客户端缓存大小/KB	256
NumberData	数据项的数量	1 000
DataSize _{max/min}	数据项大小的最大/最小值/(B)	100/200
QueryRate	每秒查询次数	1
UpdateRate	每秒更新次数	0.1
ClientSpeed _{max/min}	移动客户的最大/最小速度/(m·s ⁻¹)	1/25

3.2 实验结果

由于在系统初始化阶段,MC 的缓存没有全部填满,因此

测试的工作负载为一组随机产生的查询序列,由1000个查询组成,其中前300个查询作为缓存预热不计入统计数据,后700个查询的执行结果作为评价数据。测试的主要性能指标是查询的缓存命中率和平均响应时间,实验结果如图2、3所示。

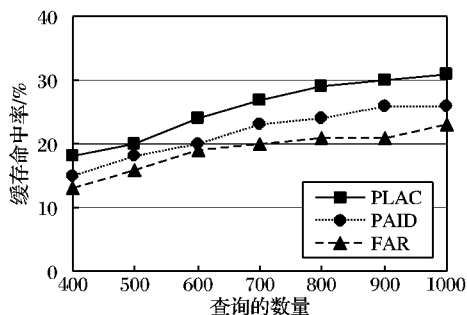


图2 查询的缓存命中率

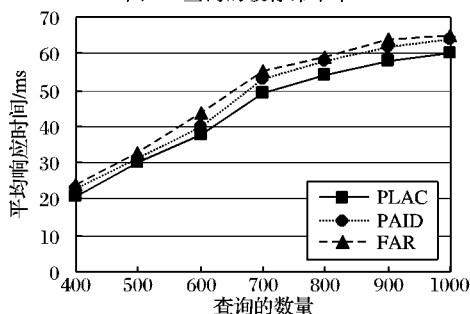


图3 查询的平均响应时间

由图2、3的实验结果可以看出,基于PLAC在查询的缓存命中率方面,相对于经典的FAR策略提高比较明显,而与PAID缓存替换策略相比,缓存命中率略高一些;查询的平均响应时间方面,比FAR策略缩小6%左右,比PAID缓存替换策略缩小3%左右。这是因为PLAC综合考虑了访问数据的时间代价、空间代价和获取数据花费的代价等缓存替换的关键因素,而FAR、PAID的策略只考虑其中的某一方面的因素。通过上面的实验数据可以看出,无论是在查询的缓存命中率还是在平均响应时间方面,PLAC都比FAR、PAID策略的性能明显提高。

4 结语

本文在详细分析现有位置相关缓存替换策略的基础上,充分考虑LDD的空间特性和访问频率、更新频率等时间特性,引入缓存替换代价函数,提出基于最小代价优先的位置相关缓存替换方法,根据代价函数值的大小进行替换。从模拟

实验的结果来看,PLAC有效地提高了缓存的命中率,降低了查询的平均响应时间。

参考文献:

- [1] WANG Y, CHAN E, LI W, *et al.* Caching invalidation strategies for supporting weak location dependent queries [C]// Proceedings of the 28th International Conference on Distributed Computing Systems Workshops. Washington, DC: IEEE Computer Society, 2008: 459-464.
- [2] TABASSUM K, HIJAB M, DAMODARAM A. Location dependent query processing-issues, challenges and applications [C]// Proceedings of the Second International Conference on Computer and Network Technology. Washington, DC: IEEE Computer Society, 2010: 239-243.
- [3] BALAMASH A, KRUNZ M. An overview of Web caching replacement algorithms [J]. IEEE Communications Surveys and Tutorials, 2004, 6(2): 44-56.
- [4] O'NEIL E J, O'NEIL P E, WEIKUM G. An optimality proof of the LRU-K page replacement algorithm [J]. Journal of the ACM, 1999, 46(1): 92-112.
- [5] KUMAR A, MISRA M, SARJE A K. A weighted cache replacement policy for location dependent data in mobile environments [C]// Proceedings of the ACM Symposium on Applied Computing. New York: ACM Press, 2007: 920-924.
- [6] LAI K Y, TARI Z, BERTOK P. Location-aware cache replacement for mobile environments [J]. Proceedings of the 2004 IEEE Global Telecommunications Conference. Washington, DC: IEEE Computer Society, 2004: 3441-3447.
- [7] JANE M M, NOUH F Y, NADARAJAN R, *et al.* Network distance based cache replacement policy for location-dependent data in mobile environment [C]// Proceedings of the 9th International Conference on Mobile Data Management Workshops. Washington, DC: IEEE Computer Society, 2008: 177-181.
- [8] REN Q, DUNHAM M H. Using semantic caching to manage location dependent data in mobile computing [C]// Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. New York: ACM Press, 2000: 210-221.
- [9] ZHENG B, XU J, LEE D L. Cache invalidation and replacement strategies for location-dependent data in mobile environments [J]. IEEE Transactions on Computers, 2002, 51(10): 1141-1153.
- [10] XU J, TANG X, LEE D L. Performance analysis of location dependent cache invalidation schemes for mobile environments [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(2): 474-488.

(上接第676页)

参考文献:

- [1] WANG ZAN, TSIM Y C, YEUNG W S, *et al.* Probabilistic Latent Semantic Analysis (PLSA) in bibliometric analysis for technology forecasting [J]. Journal of Technology Management and Innovation, 2007, 41(6): 11-24.
- [2] HOFMANN T. Unsupervised learning by probabilistic latent semantic analysis [J]. Machine Learning, 2001, 42(1/2): 177-196.
- [3] PETERSEN B, WINTER O, HANSEN L K. On the slow convergence of EM and VBEM in low-noise linear models [J]. Neural Computation, 2005, 17(9): 1921-1926.
- [4] AZADI T EI, ALMASCANJ F. Using backward elimination with a new model order reduction algorithm to select best double mixture model for document [J]. Expert Systems with Applications, 2009, 36(7): 10485-10493.
- [5] TIPPING M, BISHOP C M. Probabilistic principal component analysis [J]. Journal of the Royal Statistical Society, Series B, 1999, 61(3): 611-622.
- [6] DING C H Q. A similarity-based probability model for latent semantic indexing [C]// Proceedings on the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Berkeley: ACM Press, 1999: 194-198.
- [7] CHEN WENYEN, SONG YANGQIU, BAI HONGJIE, *et al.* Parallel spectral clustering in distributed systems [EB/OL]. [2010-02-26]. <http://www.csie.ntu.edu.tw/~cjlin/papers/psc08.pdf>.