**文章编号**:1007-5321(2018)04-0016-07 **DOI**:10.13190/j.jbupt.2017-234

# 一种基于查询聚类的物化视图动态调整策略

冯 霞[1,2], 张 江[2], 左海超[1]

(1. 中国民航信息技术科研基地, 天津 300300; 2. 中国民航大学 计算机科学与技术学院, 天津 300300)

**摘要**: 为了提高数据仓库的查询响应性能,避免视图集频繁调整引发的"抖动性",提出了一种基于查询聚类的物化视图动态调整策略,运用关联规则挖掘方法计算属性字段相似性,进而计算查询语句相似性,并对一个查询周期内的查询语句集进行聚类,产生候选视图集,根据效益模型计算候选视图的效益,再运用物化视图动态调整算法生成物化视图. 在航空公司机票结算数据集上的实验结果表明,在单机环境和分布式环境下,较基准算法相比,所提出的方法均能显著提升数据仓库的查询响应性能,尤其是对高频查询语句的响应性能.

**关 键 词**: 数据仓库; 物化视图集; 动态选择; 查询聚类; 属性字段相似度

**中图分类号**: TP311 **文献标志码**: A

# A Dynamic Adjustment Strategy of Materialized Views Based on Query Clustering

FENG Xia[1,2], ZHANG Jiang[2], ZUO Hai-chao[1]

(1. Information Technology Research Base of Civil Aviation Administration of China, Tianjin 300300, China;
2. College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

**Abstract**: In order to improve performance of query response of data warehouse, and avoid the frequent "jitter" phenomenon for materialized views set caused by immediate adjustment algorithm, a dynamic adjustment strategy of materialized views based on query clustering is presented. Firstly, attribute similarity can be calculated based on method of mining association rules, then queries similarity can be calculated and candidate views set can be generated by clustering the queries set during a statistical time, and then the benefits of candidate views can be calculated according to benefit model. Finally, the latest materialized views can be selected using dynamic management algorithm of materialized views. Based on the experimental results with data of air ticket settlement recorded by airlines. Whether in single-machine environment or distributed environment, compared to other benchmark algorithms, the overall performance of query response of data warehouse has been improved greatly, especially for high frequency queries.

**Key words**: data warehouse; materialized views set; dynamic selection; query clustering; attribute similarity

通过动态调整物化视图,提高数据仓库的查询效率一直是研究人员关注的热点. Kumar 等[1-2]分别借助遗传算法等智能仿生方法,给出了物化视图选择问题的求解策略. 但上述静态物化视图选取策略很难使数据仓库长期保持较高的查询性能. 为此,谭红星等[3]提出了一种实时对物化视图进行调

**收稿日期**: 2017-11-08
**基金项目**: 国家自然科学基金项目(61502499); 中央高校科研业务费专项资金项目(3122015z007)
**作者简介**: 张 江(1992—), 男, 硕士生, E-mail:jzhangmike@163.com; 冯 霞(1970—), 女, 教授, 硕士生导师.

整的算法 FPUS，但该算法会导致物化视图集的“抖动性”. 此外，现有研究都没有关注查询语句集中不同属性字段之间的关联关系，难以取得更好的查询效率.

笔者面向运用广泛的立方体格模型，提出了一种基于查询聚类的物化视图动态选取策略（DSMVQC，dynamic adjustment strategy of materialized views based on query clustering），借助关联规则挖掘方法 Apriori[4] 计算不同属性字段之间的相似度，进而对查询语句聚类，产生候选物化视图集. 基于效益模型，运用物化视图动态调整算法（DMAMV，dynamic management algorithm of materialized views）从候选物化视图集中确定最终需物化的视图. 实验结果表明，无论在单机环境还是分布式环境下，用 DSMVQC 策略能较大程度改善数据仓库的查询响应性能，对于提升高频查询语句的响应性能效果更明显.

## 1　物化视图候选集的生成

查询语句直观体现了用户对数据的关注点. 基于此，提出一种基于查询聚类的候选物化视图生成方法，利用频繁项集挖掘算法计算属性字段的相似度，进而计算查询语句的相似度，并对查询语句进行聚类，最后基于查询语句聚簇生成候选物化视图.

### 1.1　属性字段相似度

查询语句集中，2 个属性字段同时出现在同一条查询语句中的次数越多，则其越相似，同时，也表明用户对其的关注度越高.

将每条查询语句看作一个事务，将查询语句中包含的属性字段集看作项集，借助频繁项集挖掘算法 Apriori，计算查询语句集中属性字段间的相似度. 具体算法描述如算法 1 所示.

**算法 1**　属性字段相似性计算算法.

输入：属性字段 $a$ 和 $b$，查询语句集 $Q$，最小支持度 min_sup，最小置信度 min _conf.

输出：$a$ 和 $b$ 的相似度 $S_{a,b}$.

初始化：$S_1=0$；$S_2=0$　/* $S_1$，$S_2$ 分别为不同情况下的置信度累加值 */

$T=\text{Apriori}(Q,\text{min\_sup},\text{min\_conf})$；/* $T$ 为利用关联规则挖掘算法生成的关联规则表 */

FOR EACH $r_i \in T$ DO

　IF $a \in \text{cond}(r_i)$ && $b \in \text{concl}(r_i)$ THEN　/* $\text{cond}(r_i)$ 返回 $r_i$ 的条件项，$\text{concl}(r_i)$ 返回 $r_i$ 的结论项 */

　　$S_1=S_1+\text{conf}(r_i)$；　/* $\text{conf}(r_i)$ 返回 $r_i$ 的置信度，$S_1$ 为该情况下的置信度累加值 */

　END IF

　IF $b \in \text{cond}(r_i)$ && $a \in \text{concl}(r_i)$ THEN

　　$S_2=S_2+\text{conf}(r_i)$；/* $S_2$ 为该情况下的置信度累加值 */

　END IF

END FOR

$S_{a,b}=(S_1+S_2)/2$；

RETURN $S_{a,b}$；

从算法 1 可看出，可分别计算 2 个字段在关联规则条件列和结论列时所生成规则的置信度来计算属性字段的相似度. 属性字段间关联规则置信度越高，则属性字段越相似.

需要说明的是，传统 Apriori 算法用于发现项集间的强关联规则，在使用时通常根据问题背景设置较高的支持度和置信度阈值. 笔者借助 Apriori 算法计算任意属性字段间的相似度，而属性间的强关联规则和弱关联规则都能在某种程度上说明属性间的相似程度. 因此，设置支持度计数为 1，置信度阈值为 0，通过“频繁项集”找出属性间所有可能的“关联规则”，并计算其置信度，最终借助置信度计算属性字段之间的相似度.

### 1.2　查询语句相似度

记查询语句 $q=\{a_1,a_2,\cdots,a_n\}$，其中，$a_j(j=1,2,\cdots,n)$ 为 $q$ 中出现的属性字段，$a'$ 为查询语句集中任意一个属性字段，记 $q$ 和 $a'$ 的相似度为 $S_{a',q}$，其计算方法为

$$S_{a',q}=W_{a_j,q}\sum_{j=1}^{n}S_{a',a_j} \tag{1}$$

其中：$W_{a_j,q}$ 表示属性字段 $a_j$ 在查询语句 $q$ 中的权重，此处通过计算 $a_j$ 在 $q$ 中出现的频率得到，$S_{a',a_j}$ 表示属性字段 $a'$ 和 $a_j$ 的相似度.

记查询语句 $q'=\{a'_1,a'_2,\cdots,a'_m\}$，记 $q$ 和 $q'$ 的相似度为 $S_{q,q'}$，则其计算方法为

$$S_{q,q'}=\frac{\sum_{j=1}^{n}S_{a_j,q'}+\sum_{i=1}^{m}S_{a'_i,q}}{2} \tag{2}$$

其中：$S_{a_j,q'}$ 表示属性字段 $a_j$ 与查询语句 $q'$ 的相似度，$S_{a'_i,q}$ 表示属性字段 $a'_i$ 与查询语句 $q$ 的相似度.

采用式(2)计算查询语句相似度，而不是常用的欧氏距离，既克服了数据集可能存在的稀疏性问

题,又巧妙地识别出语句的相似程度.

### 1.3 候选物化视图集的生成

考虑到层次聚类能灵活控制聚类粒度,采用层次聚类算法 AGNES(AGglomerative NEsting)对查询语句进行聚类[5-6],算法描述如算法 2 所示.

**算法 2** 查询语句聚类算法.

输入:查询语句集 $Q$,聚类簇数 $g$.

输出:查询语句簇划分 $C=\{c_1,c_2,\cdots,c_g\}$.

init($Q$); /* 将查询语句集中每个查询语句初始化为一个簇 */

WHILE 当前聚簇数 $>g$ DO

从聚簇组中找出 2 个相似度最高的簇,并合并生成新的簇;

END WHILE

RETURN $C$;

算法 2 中,第 3 步需计算任意 2 个聚簇的相似度,若每个聚簇中只包含一条查询语句,则 2 个聚簇之间的相似度即为 2 条语句之间的相似度,若每个聚簇中的查询语句数大于 1 时,则簇 $c_i$ 和 $c_j$ 的相似度 $S_{c_i,c_j}$ 的计算方法为

$$S_{c_i,c_j}=\frac{1}{|c_i||c_j|}\sum_{q\in c_i}\sum_{q'\in c_j}S_{q,q'} \tag{3}$$

其中:$|c_i|$ 和 $|c_j|$ 分别表示簇 $c_i$ 和 $c_j$ 中查询语句数.

聚类完成后,对于每个聚簇 $c_i\in C$ 中的所有属性字段,统计其在簇中出现的频数. 将簇 $c_i$ 中出现频次大于某个阈值的属性字段筛选出来,即可形成一个候选物化视图 $v_i$. 整个查询语句集共生成 $g$ 个候选物化视图,记为候选视图集 $V_{CS}=\{v_1,v_2,\cdots,v_g\}$.

## 2 物化视图动态选择

### 2.1 物化视图的生成

受存储空间限制,当候选物化视图集较大或记录数较多时,无法将候选视图全部物化. 事实上,由于多维数据集中普遍存在的数据稀疏性,一些候选视图的记录数几乎与其父视图相同,物化此类候选视图也几乎不能带来查询性能的提升[7]. 基于此,提出一种物化效益模型,根据该模型可计算各个候选视图效益值,进而决定是否需要物化该视图.

#### 2.1.1 物化视图效益模型

参照文献[1],给出多维数据格的相关定义:

**定义 1** 多维数据格图. 记 MDDB $=\{d_1,d_2,\cdots,d_g\}$ 为多维数据集合,由数据集合中不同维度所有可能的组合作为结点,构成的具有偏序关系的有向图称为多维数据格图.

结点偏序关系:给定结点 $\text{Node}_1$ 和 $\text{Node}_2$,$\text{Node}_1$ 包含的维度都在 $\text{Node}_2$ 中出现,即由 $\text{Node}_1$ 响应的查询都可以由 $\text{Node}_2$ 响应,则 $\text{Node}_1$ 偏序于 $\text{Node}_2$,记为 $\text{Node}_1<\text{Node}_2$.

基结点视图:包含多维数据集中所有维结点所对应的视图,多维数据格图中的其他任意结点皆可由该结点生成.

**定义 2** 比较视图. 在物化视图集中,当可以响应查询 $q$ 的视图个数大于等于 1 时,记录数最小的视图称为比较视图.

以视图 $v$ 的记录数 $|v|$ 作为其查询代价,则其效益模型 $B(v,u)$ 的计算方法为

$$B(v,u)=\frac{\sum_{u<v}b(u)}{|v|},v\in V_{CS} \tag{4}$$

其中:$\sum_{u<v}b(u)$ 代表视图 $v$ 的所有子视图的效益贡献 $b(u)$ 之和,$b(u)$ 的计算方法为

$$b(u)=W_{av}+f_u*\text{AUX\_VAL},u<a \tag{5}$$

其中:为了计算 $b(u)$,将子视图 $u$ 视为查询 $u$,$a$ 为 $u$ 的比较视图;$f_u$ 表示 $u$ 对应的查询频率,当不存在查询 $u$ 时,$f_u$ 取值为 0;为了对 $f_u$ 进行规范化,加入调整系数 AUX_VAL,使 $f_u$ 的值与 $W_{av}$ 在同一个数量级,实际应用中,AUX_VAL 的取值可参考视图记录数;$W_{av}$ 表示视图集为响应查询 $u$ 所产生的查询代价差值,计算方法为

$$W_{av}=\begin{cases}|a|-|v|, & |a|>|v|\\ 0, & |a|\leqslant|v|\end{cases} \tag{6}$$

#### 2.1.2 基于贪心策略的物化视图生成

物化视图的选择问题已经被证明是 NP-hard 问题[8],以式(4)给出的效益模型为依据,基于贪心策略,选取候选视图集中效益值较大的视图进行物化.

基于贪心策略的物化视图选取算法描述如算法 3 所示.

**算法 3** 物化视图选取算法.

输入:候选视图集 $V_{CS}$,存储空间阈值 Space.

输出:已物化视图集 $V_{MS}$,查询语句集 $Q$,聚类簇数 $g$.

sort($V_{CS}$); /* 将 $V_{CS}=\{v_1,v_2,\cdots,v_g\}$ 中各候选视图按效益值降序排列 */

FOR $i=1,2,\cdots,g$ DO /* 遍历排序后的 $V_{CS}$ 中每个视图 */

IF Space≥|$v_i$| THEN　/* $v_i$ 为 $V_{CS}$中效益值排第 $i$ 的视图 */

$V_{MS} = V_{MS} \cup \{v_i\}$;

Space = Space − |$v_i$|;

END IF

END FOR

RETURN $V_{MS}$;

### 2.2　物化视图的动态选取

由于数据仓库具有时变性的特点,随着时间的变化,用户的查询偏好也会发生变化,为保持系统的响应性能,需要定期对物化视图进行动态调整. 在算法 3 的基础上,提出的物化视图动态调整算法 DMAMV 如算法 4 所示.

**算法 4**　DMAMV 算法.

输入:候选视图集 $V_{CS}$,存储空间阈值 Space.

输出:已物化视图集 $V_{MS}$.

初始化:$V_{MS\_NEW} = \varnothing$; $V_{TS} = \varnothing$; $V_{MS\_OLD} = V_{MS}$/* $V_{MS_NEW}$为新增物化视图队列,$V_{MS_OLD}$为旧物化视图队列,$V_{TS}$用于存放预删除的视图 */

sort($V_{CS}$);　/* 将 $V_{CS} = \{v_1, v_2, \cdots, v_g\}$ 中各候选视图按效益值降序排列 */

FOR $i = 1, 2, \cdots, g$　DO　/* 按照效益值从高到低遍历 $V_{CS}$中的视图 */

IF Space≥|$v_i$| THEN　/* $v_i$ 为 $V_{CS}$中效益值排第 $i$ 的视图 */

$V_{MS\_NEW} = V_{MS\_NEW} \cup \{v_i\}$;

Space = Space − |$v_i$|;

ELSE /* 剩余空间 Space 不能存放视图 $v_i$ */

WHILE Space < |$v_i$| DO　/* |$v_i$|为视图 $v_i$ 的记录数 */

IF $V_{MS\_OLD} = = \varnothing$ THEN　/* 若将 $V_{MS_OLD}$中的视图全部删除后 Space 仍然小于|$v_i$|,则将预删除的视图恢复到 $V_{MS\_OLD}$ */

$V_{MS\_OLD} = V_{MS\_OLD} \cup V_{TS}$;

BREAK;

END IF

$V_{MS\_OLD} = V_{MS\_OLD} - \{w_{least}\}$;

/* $V_{MS_OLD}$中删除当前查询频率最少的视图 $w_{least}$ */

$V_{MS} = V_{MS} - \{w_{least}\}$;

$V_{TS} = V_{TS} \cup \{w_{least}\}$; /* 将 $w_{least}$放入预删除视图集 $V_{TS}$中 */

Space = Space + |$w_{least}$|;

/* |$w_{least}$|为视图 $w_{least}$的记录数 */

END WHILE

IF Space < |$v_i$| THEN /* 若对 $V_{MS_OLD}$中的视图进行预删除后依然不能存放视图 $v_i$,则检验 $V_{CS}$中第 $i+1$ 个视图 */

BREAK;

END IF

$V_{MS\_NEW} = V_{MS\_NEW} \cup \{v_i\}$;　/* 若对 $V_{MS_OLD}$中的视图进行预删除后可以存放视图 $v_i$,则将 $v_i$ 加入 $V_{MS_NEW}$中 */

Space = Space − |$v_i$|;

END IF

END FOR

IF $V_{MS\_OLD} = = \varnothing \&\& V_{MS\_NEW} = = \varnothing$ THEN　/* 若 $V_{MS_OLD}$中所有视图全部删除后,存储空间依然不能存放 $V_{CS}$中至少一个视图,则将 $V_{TS}$中的视图恢复到 $V_{MS}$中 */

$V_{MS} = V_{TS}$;

ELSE

$V_{MS} = V_{MS\_NEW} \cup V_{MS}$;　/* 将新增物化视图集放入 $V_{MS}$中 */

END IF

RETURN $V_{MS}$;

### 2.3　物化视图动态选取策略–DSMVQC

综上所述,提出的物化视图动态选取策略 DSMVQC 简述如下:首先,利用 1.1 节提出的属性相似度计算方法获得属性间的相似度,并以此为基础利用 1.2 节提出的查询语句相似度计算方法获得查询语句间的相似度,并对查询语句聚类,通过聚类产生候选物化视图集;接着,利用 2.1 节提出的效益模型计算候选物化视图的效益值,结合贪心策略选出需要被物化的视图;最后,考虑用户查询分布的时变性,利用 2.2 节提出的 DMAMV 算法对已物化视图进行动态选取,使数据仓库长期保持较高的查询响应性能.

## 3　实验

### 3.1　实验环境

为了验证 DSMVQC 策略在不同数据规模上的有效性和一致性,且考虑实际应用中数据规模的多样性,分别在单机环境和分布式环境中进行了实验. 其中,单机环境中硬件平台为 Dell OptiPlex 7010 Mini Tower,数据仓库平台为 MySQL 5.6. 分布式环

境中硬件平台配置由6台Dell PowerEdge R210 II服务器组成,数据仓库平台为Hive 0.14.0.

### 3.2 实验数据

实验数据集来源于国际航空运输协会(IATA, international air transport association)的商业智能系统(BI,business intelligent system),为某年连续3个月的全国代理点机票结算数据,共包含1个事实表和8个维表.

实验所用查询语句集包括两类,第1类来自BI系统真实用户查询共118条语句,按照一个查询周期内查询语句出现频数将其分为4组,如表1所示.

**表1 第1类真实查询语句**

| 组 | 查询频数 | 查询语句条数 |
|---|---|---|
| 1 | ≤4 | 26 |
| 2 | (4,7] | 42 |
| 3 | (7,11] | 38 |
| 4 | >11 | 12 |

由表1可见,118条语句中,查询频数大于7的语句共50条,超过查询语句总数的40%,这表明用户针对机票结算数据集的查询偏好较明显.

为了验证新策略在更大规模查询样本集上的有效性,结合BI系统数据集与查询语句特点,仿真生成1 000条查询语句,作为第2类查询语句集.

### 3.3 实验参数设置

采用新策略生成物化视图,需要预先设定部分实验参数,如属性筛选阈值、聚簇数和空间约束值等. 设置原则如下.

属性筛选阈值. 一般来讲,查询语句集的语句数量与其涵盖的属性字段个数成正比. 由此,属性筛选阈值可定义为相异属性字段数目与语句数目的比值. 此外,也可通过多组实验确定恰当阈值.

聚簇数. 同一聚簇内的语句相似性高,不同聚簇间的语句相似性低. 因此,当查询语句覆盖的业务面较广、数量较多,或查询语句的复杂程度较高、数据仓库中基本表集的规模较大时,可增大聚簇数的值;反之,可减少聚簇数的值.

空间约束值. 实际应用中,可根据存储空间大小确定空间约束值.

结合本文实验环境,实验参数的设置如下:查询语句聚簇数为4,属性字段出现频次筛选阈值为5,单机环境下空间约束值为$1.2\times10^5$行,分布式环境下空间约束值为$1.6\times10^8$行.

### 3.4 实验过程

实验基准算法选用较经典的BPUS(benefit per unit space)[1]及其改进算法FPUS(frequency per unit space)[7],采用Java语言实现.

**实验1** 不同查询频数查询语句响应时间比较. 从表1所示的第1类查询语句集中,每组查询语句随机抽取50%作为测试语句集,以验证物化视图集对不同查询频数的查询语句的响应性能. 此外,又从第1类查询语句集中随机抽取50%的查询语句作为第5组测试语句集,以验证物化视图集对整体查询语句集的响应性能. 图1给出了不同查询频数查询语句的响应性能比较结果.

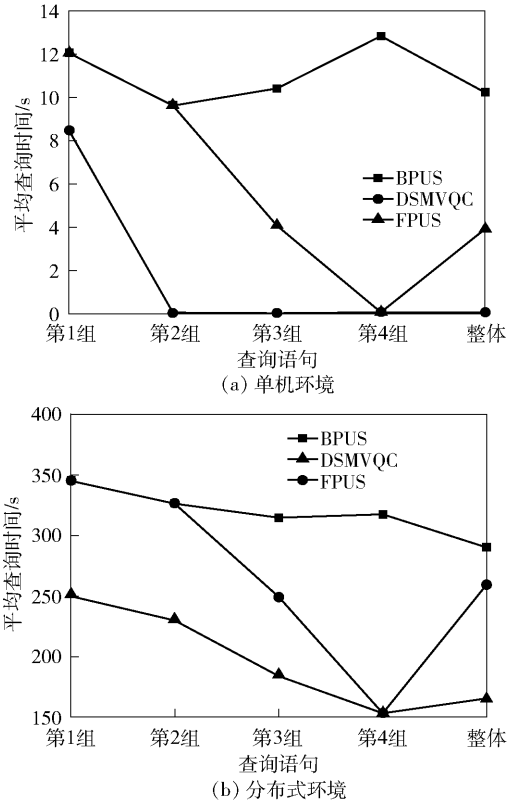


图1 不同查询频数查询语句响应时间比较

由图1可看出:1)在单机环境和分布式环境下,对于不同查询频数的查询语句,DSMVQC策略选择视图集的查询响应性能均优于BPUS算法和FPUS算法;2)BPUS算法对不同查询频数的查询语句响应性能没有太大差异,其原因主要是BPUS算法仅以视图记录数为查询代价来衡量视图效益,没有考虑用户对数据集的查询偏好. 而FPUS算法虽考虑到用户的查询偏好,但在有限的空间内,个别查询频率特别高且属性字段重复度较大的视图占据整个空间,导致物化视图集整体的多样性较差.

随着查询频数的增加，与基准算法相比，DSMVQC 策略的优势更为明显，这是因为 DSMVQC 策略会随着查询频率的增加，对用户查询语句的关注度依次增加，保证了视图集多样性的同时对有查询偏好的数据集有更好的适用性.

**实验 2**　不同查询阶段查询语句响应时间比较. 从第 2 类查询语句集中采用无放回分层抽样方法抽取 4 组查询语句，每组 100 条，分别作为 4 个不同阶段的查询语句集. 从每组查询语句集中随机抽取 50% 查询语句作为测试语句集，以验证物化视图集在不同阶段的响应性能. 图 2 给出了不同阶段查询语句的响应性能比较结果.

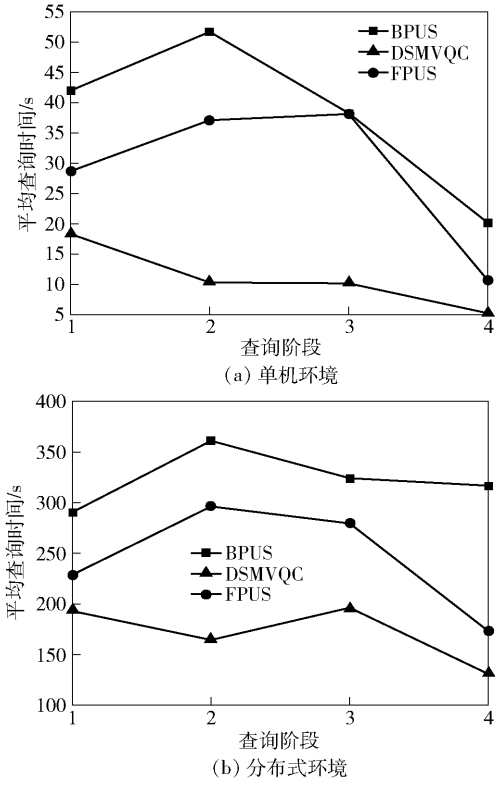


图 2　不同查询阶段查询语句的查询响应时间比较

由图 2 可看出，在单机环境和分布式环境下，对于不同阶段的查询语句，DSMVQC 策略选择视图集的查询响应性能均优于 BPUS 算法和 FPUS 算法. 这是因为 DSMVQC 策略定期收集用户查询语句，生成当前阶段用户关注度较高的新视图集，按效益值从大到小的顺序进行物化. 物化过程中，如果存储空间不足，则采用“最近最少使用”策略淘汰存储空间中旧视图. 物化结束条件是全部新视图集都被物化或者存储空间中没有可以淘汰的旧视图. 由此，DSMVQC 策略使物化视图集在不同阶段均包含价值较大的视图，使得数据仓库长期保持较高的查询响应性能. 而基准算法 BPUS 每阶段只单纯根据视图尺寸大小选取物化视图，导致其有效性较差. 基准算法 FPUS 倾向于长期保留查询频率较高且属性字段重复度较大的视图，难以适应用户查询偏好的变化. 此外，DSMVQC 策略运用聚类方法避免了视图集的抖动.

较单机环境相比，分布式环境下 DSMVQC 策略和其他 2 种基准算法对不同阶段查询语句响应性能的波动程度较小，这是因为分布式环境中查询响应时间与数据量大小呈线性关系，而单机环境中呈非线性关系.

**实验 3**　多表联合查询和单表查询时间比较. 通常情况下，将视图进行物化，可使用户查询由基本表集响应转化为由单个物化视图响应，即多表联合查询转化为单表查询. 本实验针对分布式环境，在基本表集尺寸大小恒定且基本表集中事实表行数等于物化视图行数的情况下，依次设定物化视图尺寸为 3、4、5、6、7 GB，分别测试复杂用户查询（select a1 from T1 where a2 like ‘***’ and substr(a3) = ‘***’ and month(a4) < = ‘***’ group by a1;）和简单用户查询（select a1 from T2;）的响应时间. 实验结果如图 3 所示.

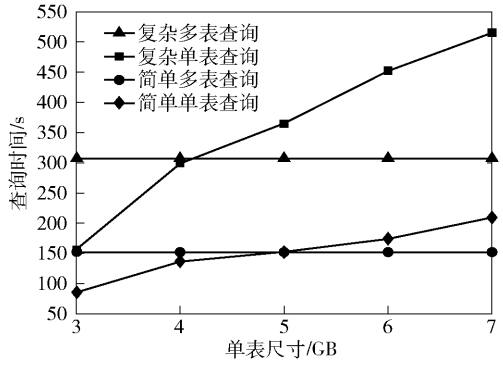


图 3　多表联合查询和单表查询时间比较

从图 3 可见，在分布式环境中，无论是复杂用户查询还是简单用户查询，查询时间和数据量呈线性关系，且随着物化视图尺寸的增加，单表查询语句的响应时间渐渐大于多表联合查询的响应时间，因此，1.3 节中聚簇的属性字段个数应限定在一定数量，否则由聚簇生成的物化视图尺寸较大，其查询响应性能反而没有基本表集的查询响应性能高.

**实验 4**　效益模型性能对比. 通常，好的效益模型应该选择物化命中率高的视图. 以第 2 类查询语

句集为样本数据,统计某一查询周期内各个视图的查询命中率,并根据命中率的不同抽取5个典型视图,如表2所示.

**表2 物化视图查询命中率**

| 视图编号 | 命中率 |
|---|---|
| 1 | 0.025 |
| 2 | 0.02 |
| 3 | 0.13 |
| 4 | 0.29 |
| 5 | 0.31 |

实验基准模型选自BPUS算法效益模型,分别通过基准算法效益模型和本文效益模型计算表2中视图的效益值,实验结果如图4所示.

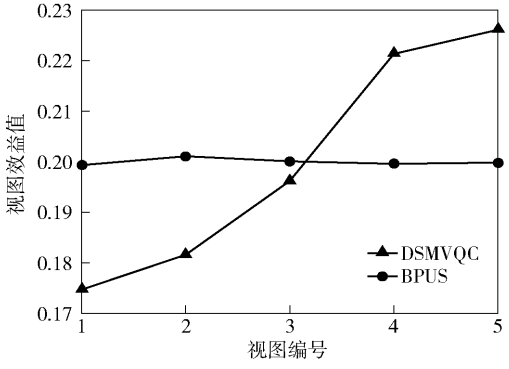


图4 效益模型性能对比

结合表2和图4可以看出,效益模型计算出的效益值和相应视图的命中率成正相关. 采用基准算法计算出的效益值和相应视图的命中率几乎不相关. 这是因为BPUS算法效益模型仅以视图尺寸为依据计算视图效益值,而DSMVQC算法效益模型在视图尺寸的基础上还考虑了用户查询语句对视图效益值的影响,因此计算所得效益值更加合理.

## 4 结束语

提出一种物化视图动态选取策略DSMVQC,实验结果表明,在单机环境和分布式环境下,该策略较其他基准算法相比,数据仓库的查询性能均有较大改善,且对于偏好查询的响应效果更优. 该策略对定期收集的用户查询语句集进行聚类,避免了物化视图集频繁调整的"抖动性"问题,且当数据仓库查询分布情况变化时,使数据仓库依然可以保持较高的查询响应性能.

**参考文献:**

[1] Kumar A, Kumar T V V. Materialized view selection using discrete genetic operators based particle swarm optimization[C]//International Conference on Inventive Systems and Control. Coimbatore: IEEE, 2017: 1-5.

[2] Kumar T V V, Arun B. Materialized view selection using marriage in honey bees optimization[J]. International Journal of Natural Computing Research, 2015, 5(3): 1-25.

[3] 谭红星, 周龙骧. 多维数据实视图的动态选择[J]. 软件学报, 2002, 13(6): 1090-1096.
Tan Hongxing, Zhou Longxiang. Dynamic selection of materialized view of multi-dimensional data[J]. Journal of Software, 2002, 13(6): 1090-1096.

[4] Sathya M, Isakki D P. Apriori algorithm on web logs for mining frequent link[C]//International Conference on Intelligent Techniques in Control, Optimization and Signal Processing(INCOS). Srivilliputhur: IEEE, 2017: 1-5.

[5] 延皓, 张博, 刘芳, 等. 基于量值的频繁闭项集层次聚类算法[J]. 北京邮电大学学报, 2011, 34(6): 64-68.
Yan Hao, Zhang Bo, Liu Fang, et al. A new method of items' quantities based closed frequent itemsets hierarchical clustering[J]. Journal of Beijing University of Posts and Telecommunications, 2011, 34(6): 64-68.

[6] 赵金东, 于彦伟, 刘惊雷. 面向实时海量数据流的数据聚类[J]. 北京邮电大学学报, 2016, 39(3): 114-119.
Zhao Jindong, Yu Yanwei, Liu Jinglei. A data clustering algorithm over real time high-volume data streams[J]. Journal of Beijing University of Posts and Telecommunications, 2016, 39(3): 114-119.

[7] 张柏礼, 孙志挥, 孙翔. 物化视图选择的预处理算法[J]. 计算机研究与发展, 2004, 41(10): 1645-1651.
Zhang Baili, Sun Zhihui, Sun Xiang. Preprocessor of materialized views selection[J]. Journal of Computer Research and Development, 2004, 41(10): 1645-1651.

[8] Yao D, Abulizi A, Hou R. An improved algorithm of materialized view selection within the confinement of space[C]//2015 IEEE Fifth International Conference on Big Data and Cloud Computing. Dalian: [s.n.], 2015: 310-313.