

支持 What-if 分析的 OLAP 多版本管理机制

张宇^{1,3}, 张延松^{1,2,3}

(1. 中国人民大学数据工程与知识工程教育部重点实验室, 北京 100872; 2. 中国人民大学信息学院, 北京 100872; 3. 哈尔滨金融高等专科学校计算机系, 哈尔滨 150030)

摘要: 针对 What-if 分析中的多版本更新管理需求, 根据不同的假设更新实现技术, 提出支持 What-if 分析的内存 OLAP 系统中的多版本更新策略, 其中包括常规的递归假设更新和基于差值的多版本假设更新, 采用预合并机制的 delta 记录合并算法来提高级联假设更新模式下的多版本 What-if 查询性能。实验表明, 与常规的递归假设更新相比, 该算法无论在减少多版本更新代价方面还是在减少整体 What-if 查询时间方面均有效。

关键词: What-if 分析; 假设更新; 假设更新数据视图; 级联假设更新

Multi-version Management Mechanism in OLAP with What-if Analysis Support

ZHANG Yu^{1,3}, ZHANG Yan-song^{1,2,3}

(1. Key Laboratory of Ministry of Education for Data Engineering and Knowledge Engineering, Renmin University of China, Beijing 100872; 2. School of Information, Renmin University of China, Beijing 100872; 3. Department of Computer Technology, Harbin Financial College, Harbin 150030)

【Abstract】 This paper focuses on requirements of multi-version management in What-if analysis. Multi-version update policy is presented in main-memory OLAP system with What-if analysis support based on different hypothetical update techniques which includes traditional cascade multi-version update and multi-version update based on difference value delta tuples. Delta tuple pre-merge mechanism is proposed to improve the performance of cascade hypothetical version merging procedure. Performance experiment shows that the pre-merge hypothetical update can reduce both time cost of multi-version updates and total time cost of What-if queries remarkably compared with cascade multi-version update.

【Key words】 What-if analysis; hypothetical update; hypothetical update data view; cascade hypothetical update

What-if query 是基于假设场景的查询, 在实际的应用场景中, What-if analysis 往往是基于多用户协同分析机制, 即多个不同管理层次的用户在不同的数据维度或不同的数据空间上根据商业处理逻辑进行假设分析, 并通过不同用户之间的协同完成企业级决策的假设分析。在 What-if analysis 的假设更新策略中, 典型的方法包括 2 类: 基于 delta 表^[1-2]和基于查询重写^[3]。

1 多版本假设更新

在 What-if analysis 的应用场景中, 本文的研究以基于值的假设更新为目标, 分析基于事实表、维表或重要的数据视图上的多版本假设更新管理机制^[4-6]。假设更新版本和假设更新数据视图见图 1。

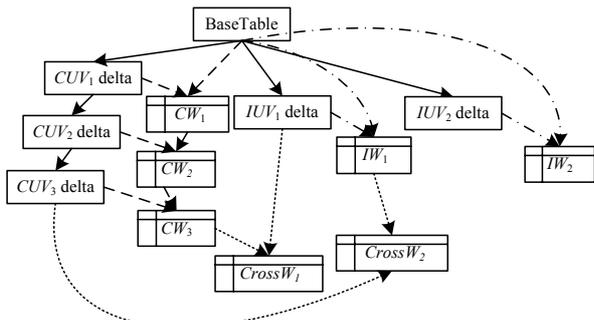


图 1 假设更新版本和假设更新数据视图

如图 1 所示, BaseTable 为假设更新所依据的事实表或数据视图, CUV 代表级联假设更新, 即当前的假设更新基于前一个假设更新的数据视图; IUV 代表独立假设更新, 即假设更新的基础是基表, 与其他的假设更新无关; CW, IW 和 CrossW 是基于不同假设更新类型的假设数据视图, CrossW 是一种交叉类型, 在一种假设更新类型的基础上与其他类型的假设更新版本进行合并。

2 基于 delta 表的假设更新版本管理

在 delta 表模式下, 假设更新所影响的基表记录被存储在独立的 delta 表中, delta 表的结构是在基表结构基础上增加若干用于管理的附加字段(如版本号 version、类型标识 flag 等), 基于值的假设更新只改变度量字段的内容, 在存储时可以在相应的度量字段中存储假设更新后的新值, 也可以存储假设更新后与假设更新前该字段的差值, 本文采用差值存储模式。当需要生成假设更新数据视图时, 根据 delta 记录的标识类型采取如下策略: 当类型为 U 时, 需要将对应的基表记

基金项目: 国家自然科学基金资助项目(60473069, 60496325); 国际合作(HP Lab)基金资助项目“Large Scale Data Management”; 北京市教委产学研合作基金资助项目“基于内存的联机分析处理系统”

作者简介: 张宇(1977-), 女, 副教授、硕士, 主研方向: 数据仓库, 电子商务; 张延松, 博士研究生

收稿日期: 2009-10-10 E-mail: zhangys_ruc@hotmail.com

录和 delta 表记录的度量值相加后输出；当类型为 D 时，不输出当前基表记录；当类型为 I 时，直接输出该 delta 表记录。不同假设更新版本的 delta 记录存储在相同的 delta 表中，delta 记录之间通过版本号标识。

2.1 递归多版本假设更新

基于递归模式的多版本假设更新是指根据假设更新的级联关系，在前一个假设更新数据视图的基础上合并当前假设更新的 delta 记录集，生成新的假设更新数据视图。

用符号“ \bowtie ”表示数据视图与当前假设更新记录集之间的合并操作，则在多版本级联假设更新模式下的假设数据视图可以表示为

$$CW_1 = \text{BaseTable} \bowtie CUV_1$$

$$CW_i = CW_{i-1} \bowtie CUV_i \quad i \in N$$

算法描述如下：

算法 1 cascade merge

输入 基表 B ，delta 表 D ，级联更新版本序列 L

输出 假设更新数据视图 What-ifView

```
B → tmpView;
for each 更新版本 v in L
    tmpView = tmpView  $\bowtie$  CUVv;
end for
What-ifView = tmpView;
return What-ifView;
```

当系统中存在多个级联假设更新版本时，需要进行多次的合并操作，当基表的记录数量很大时，级联假设数据视图的生成代价非常大。

2.2 预合并多版本假设更新

以 2 个版本的级联假设更新为例分析级联更新 delta 记录之间的数据关系。 v_1 假设更新版本对应的数据视图的记录包括：基表中没有被影响的记录 t 和 3 种假设更新类型的 delta 记录 d ，数据视图 What-ifView $v_1 = \{t\} \cup \{d_{U1}\} \cup \{d_{D1}\} \cup \{d_{I1}\}$ 。在 v_2 假设更新版本中，除了产生新的基于基表记录的 delta 记录之外，还会在 v_1 版本的 delta 记录之上生成新的 delta 记录，类型对应关系为： $U \rightarrow U, D; I \rightarrow U, D; D \rightarrow \emptyset$ 。 Δm 与 $\Delta m'$ 分别为假设更新版本 1 和更新版本 2 中的度量值差值，以 U 操作为例，在 v_1 假设更新版本中，设更新后的值为 V ，则 $\Delta m = V - m$ ，在 v_2 假设更新版本中 $\Delta m' = V - (m + \Delta m)$ 。

多版本的 delta 记录合并满足结合律，可以在多版本假设更新视图生成的过程中先对多版本的 delta 记录进行合并，最后再与基表进行合并，从而减少假设更新数据视图多版本合并时的数据处理代价。算法描述如下：

算法 2 pre-merge

输入 基表 B ，delta 表 D ，级联更新版本序列 L

输出 假设更新数据视图 What-ifView

```
tmpdeltaView = deltaSet1;
for each 更新版本 v in L
    tmpdeltaView = tmpdeltaView  $\bowtie$ 
        deltaSetv+1;
end for
What-ifView = B  $\bowtie$  tmpdeltaView;
return What-ifView;
```

3 基于查询重写机制的假设更新版本管理

在查询重写机制中，假设更新所影响的记录直接被蕴含在重写的查询语句中，无法单独对 delta 记录进行访问，因此在多版本级联更新场景中一般采用嵌套查询的方法，即将上一个假设版本所对应的 What-if 查询作为子查询包含在当前

的查询重写语句中，通过 SQL 引擎自身的优化机制进行优化，在复杂的假设更新场景下，复杂的嵌套查询的性能会受到很大的影响。算法描述如下：

算法 3 nestedQueryRe-writing

输入 基表 B ，查询 Q

输出 假设更新数据视图 What-ifView

```
rwQuery1(Q) → hypotheticalViewv1;
rwQuery2(hypotheticalViewv2) → What-ifView
return What-ifView;
```

4 实验与性能分析

实验的硬件环境是：HP Integrity rx2620-2 1.6 GHz, 4 GB 内存, 160 GB 硬盘。测试数据集为 FoodMart，笔者自己设计了测试数据生成工具，测试中的事实表记录量为 800 万条，delta 记录占事实表记录的比率 η 从 1%~10%，其中，I 类型的 delta 记录为 30%；D 类型的 delta 记录为 30%；U 类型的 delta 记录为 40%。测试查询为 1 个事实表与 4 个维表连接后进行分类汇总计算。

4.1 实验设计

在实验中对比了前文所述的 3 种算法，算法基于 SQL 语句实现，实验的数据库平台是开源的内存数据库 MonetDB，它是一种基于列存储模式的高性能的内存数据库，具有良好的 OLAP 查询处理性能。

4.1.1 delta 表合并算法设计

在笔者前期的研究工作中，实现并测试了多种基于 delta 表合并机制的算法，本文采用基于 full join 模式的基表与 delta 表全映射匹配算法实现假设更新数据视图的生成算法。实验中测试了具有 2 个和 3 个级联假设更新版本情况下的假设数据视图生成算法。

cascadeMerge 算法对应的查询处理过程如下：

```
What-ifViewv1 = B  $\bowtie$  D1
What-ifViewv2 = What-ifViewv1  $\bowtie$  D2
What-ifViewv3 = What-ifViewv2  $\bowtie$  D3
```

pre-Merge 算法对应的查询处理过程如下：

```
deltaViewv2 = D1  $\bowtie$  D2, What-ifViewv2 = B  $\bowtie$  deltaViewv2
deltaViewv3 = deltaViewv2  $\bowtie$  D3, What-ifViewv3 = B  $\bowtie$  deltaViewv3
```

4.1.2 查询重写算法设计

以 sales 表中的 product_id 为基准，选出 $\eta\%$ 的事实表记录作为查询重写的假设更新记录，并按照 U, I, D 这 3 种类型的比例设置了假设更新对应的查询语句，并以此为基础重写整个查询语句。

在处理级联假设更新时，将上一个版本的假设数据视图合并的查询语句存储为一个视图，并在当前查询中用该视图替换原始的事实表。

4.2 实验结果

4.2.1 相同 delta 记录比率时的假设更新算法性能

在进行算法性能测试时，分别测试了假设数据视图合并时间和查询执行整体时间。随着级联假设更新版本的增加，查询处理时间随之增加，但不同算法中，查询执行时间增长的幅度有很大的差异：pre-merge 算法查询时间增长的幅度最小，而级联查询重写算法的查询执行时间成倍数增长。

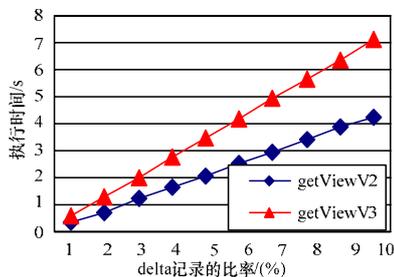
考查假设更新数据视图合并过程占查询总时间的比例，发现在 pre-merge 算法中，级联假设更新版本的层次为 2 和 3 时，其比率分别为 0.73% 和 1.36%；cascade merge 算法中相应的比例为 45.07% 和 64.10%；而级联查询重写算法中比例为 23.72% 和 11.87%，虽然合并过程的时间比例在减少，但

整个查询的执行时间分别增长了 1 倍和 7 倍。假设更新会导致查询重写后的 SQL 语句的复杂度大大提高,在级联假设更新时再进行复杂的嵌套转换将导致查询语句的复杂度进一步增加。SQL 查询引擎只能基于操作进行优化,而在 What-if 查询中需要根据假设更新的语义信息进行优化才能更好地提高查询的执行性能,目前还没有专门针对 What-if 应用的查询优化器,因此在级联假设更新中采用查询重写算法难以获得良好的查询处理性能。

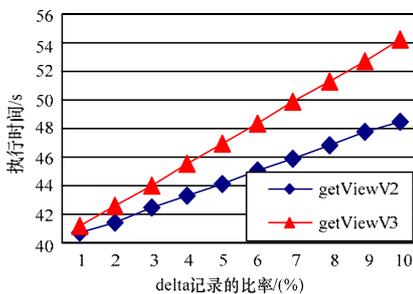
4.2.2 在不同的 delta 记录比率时 pre-merge 算法的性能

delta 记录预合并机制是基于差值存储的 delta 表存储策略,可以将多版本级联假设更新的过程分解执行,先对数量较少的 delta 记录进行合并,再与事实表进行代价高昂的合并操作。

如图 2 所示,当测试使用的 delta 记录的比率从 1%变化到 10%时,pre-merge 算法的总执行时间随比率的增加而增长,图中级联假设更新版本的层次分别为 2 和 3,即 getViewV2 和 getViewV3。



(a)假设数据视图生成时间



(b)查询执行总时间

图 2 delta 记录比率为 1%时的算法性能

编辑 任吉慧

(上接第 69 页)

在实际的测试中,信号、流量、诱导、报警和 GPS 信息的服务品质主要参考每个数据包的延时情况,以及数据包的丢包率,而视频信号主要是参考视频图像的延时和图像的质量,其实也可以通过数据包的延时和丢包率来反应。具体测试情况如表 1 所示。

表 1 实时交通信息平均数据包延时

频道	发送频率 (/数据包·s ⁻¹)	期望 延时/s	实际平均 延时/s	期望 丢包率/(%)	实际 丢包率/(%)
信号频道	100	0.2	0.111 3	0	0
流量频道	200	1.0	0.936 7	1	0
视频频道	50	5.0	3.243 0	5	3
频道 4	1	0.1	0.056 4	0	0

4 结束语

本文探讨了一种针对传统消息服务改进的基于频道的消

从图中可以看到,假设数据视图的生成时间和查询总时间与 delta 记录比率成正比,级联更新的版本越多,其查询执行时间越长;假设更新级联版本的层次数增加时,假设数据视图的生成代价增大,由于执行代价最大的与事实表的合并操作被推迟到最后执行,因此查询总体执行时间增长的幅度小于假设数据视图生成时间增长的幅度。

5 结束语

本文针对 What-if analysis 应用中多版本假设更新场景下的查询处理特点提出了基于 delta 表的预合并算法来解决级联假设更新时 What-if 查询执行性能低的问题,通过预合并机制先对数据量小的多版本 delta 记录进行合并,最后与基表合并后执行当前的 What-if 查询。实验表明,基于差值存储策略的 delta 表预合并算法可以有效地解决多版本假设更新时的 What-if 查询性能问题,能够更好地应用于多用户、多版本的 What-if analysis 场景。

参考文献

- [1] Woodfill J, Stonebraker M. An Implementation of Hypothetical Relations[C]//Proc. of the 9th Int'l Very Large Data Base Conference. San Francisco, USA: Morgan Kaufmann Publishers Inc., 1983: 157-166.
- [2] Timothy G, Richard H. A Framework for Implementing Hypothetical Queries[C]//Proc. of the ACM SIGMOD International Conference on Management of Data. Tuscon, USA: [s. n.], 1997: 231-242.
- [3] Balmin A, Papadimitriou T, Papakonstantinou Y. Hypothetical Queries in an OLAP Environment[C]//Proc. of the 26th Int'l Very Large Data Bases Conf.. San Francisco, USA: Morgan Kaufmann Publishers Inc., 2000: 220-231.
- [4] 王 珊, 肖艳芹, 张延松, 等. 支持 What-if 分析的 OLAP 系统研究[J]. 计算机学报, 2008, 31(9): 1573-1587.
- [5] Bebel B, Eder J, Koncilia C. Creation and Management of Versions in Multiversion Data Warehouse[C]//Proc. of 2004 ACM Symposium on Applied Computing. Nicosia, Cyprus: [s. n.], 2004: 717-723.
- [6] Morzy T, Wrembel R. On Querying Versions of Multiversion Data Warehouse[C]//Proc. of the 7th ACM International Workshop on Data Warehousing and OLAP. Washington D. C., USA: [s. n.], 2004: 92-101.

息服务,它通过频道管理使消息传输服务与具体业务逻辑脱离开来,利用消息的频道注册和查询机制来实现与业务的动态关联,使得消息服务在建成以后拥有较多的扩展性,能够较好满足企业不同业务的分布式操作和信息交换的需求。

参考文献

- [1] 范玉顺. 企业集成系统技术[J]. 新技术新工艺, 2005, 17(7): 4-7.
- [2] 逯 鹏, 林学练, 王 斌. 可配置可扩展发布订阅系统的研究及其实现[J]. 计算机应用研究, 2007, 24(5): 160-162, 168.
- [3] 张卓彬, 张 璟, 李军怀. 一种面向消息的服务契约开发方法研究与实现[J]. 微电子学与计算机, 2008, 25(8): 99-102.
- [4] 阳王东, 何焕民, 祝 青. 基于统计分析对交通信息集成的优化[J]. 交通运输系统工程与信息, 2008, 8(4): 114-118.

编辑 任吉慧