

CẬP NHẬT KHÔNG ĐỒNG BỘ CÁC KHUNG NHÌN THỰC ASYNCHRONIZED ACTUALIZATION FOR THE MATERIALIZED VIEWS

Nguyễn Trần Quốc Vinh

Trường Đại học Kinh tế, Đại học Đà Nẵng

TÓM TẮT

Để duy trì các bảng khung nhìn thực (materialized view, KNT) trong trạng thái thực tiễn, cần phải cập nhật chúng mỗi khi có sự thay đổi dữ liệu trong các bảng gốc. Phụ thuộc vào thời hạn đưa các thay đổi vào các bảng KNT, các cơ chế cập nhật được phân ra đồng bộ và không đồng bộ. Trên thực tế có các bài toán của hệ thống thông tin đòi hỏi phải tính đến một cách không chậm trễ chỉ một phần trong số các thay đổi dữ liệu trong các bảng gốc. Từ đó xuất hiện khả năng trì hoãn cập nhật KNT. Khả năng đó cho phép giảm số lượng các cập nhật KNT, thông qua đó nâng cao đáng kể năng suất tổng thể của hệ thống. Bài viết phân tích về cập nhật không đồng bộ và đề nghị thuật toán xác định các thời điểm cập nhật không đồng bộ các KNT.

ABSTRACT

To keep the tables – Materialized Views (MV) to be in the actual state, they must be updated upon the changes of the data in the source tables. Following to the deadline to apply those changes to the materialized views, the mechanisms of the MV can be classified by synchronized and asynchronized updates. In practice, there are tasks solved in the informations systems, which request to apply immediately only part of the changes made in the source tables. The feasibility to defer the MV updates allows to decrease the number of the MV updates and the global performance of the information systems can be significantly improved. This paper analyzes the mechanism of the asynchronized MV updates and suggests the algorithm to determine the moments to asynchronizedly update the MV.

I. CÁC CƠ CHẾ CẬP NHẬT

Cứ mỗi khi dữ liệu trong các bảng gốc (base tables, BG) được cập nhật, các KNT sử dụng các dữ liệu đó trở nên không thực tiễn. Để duy trì các bảng KNT trong trạng thái thực tiễn (actual state), cần phải cập nhật chúng mỗi khi có sự thay đổi dữ liệu (insert, update, delete) trong các BG. Quá trình làm cho dữ liệu trong KNT tương ứng với dữ liệu trong các bảng gốc đó được gọi là sự thực tiễn hoá (actualization, cập nhật).

Tùy thuộc vào cách thức thực hiện, cập nhật được chia thành ba phương pháp chính, đó là hoàn toàn (COMPLETE), gia tăng (FAST hay còn gọi là INCREMENTAL) và ép buộc (FORCE). Cập nhật hoàn toàn thực tế là tạo lại KNT bằng cách thực thi truy vấn trên cơ sở KNT đã được tạo ra. Cập nhật gia tăng chỉ sửa đổi phần dữ liệu trong KNT liên quan đến các thay đổi dữ liệu trong các bảng gốc. Cập nhật ép buộc nghĩa là khi có khả năng thì thực hiện cập nhật gia tăng, còn nếu không thì sử dụng cập nhật hoàn toàn.

Phụ thuộc vào thời hạn đưa các thay đổi vào các bảng KNT, các cơ chế cập nhật được phân ra đồng bộ [1÷4] và không đồng bộ [5]. Cập nhật đồng bộ được thực thi không chậm trễ ngay khi có thay đổi dữ liệu trong bảng gốc như một phần của giao tác thực hiện thay đổi đó. Ngược lại, cập nhật không đồng bộ được thực hiện vào một thời điểm nào đó sau khi các giao tác sửa đổi dữ liệu trong bảng gốc đã được cố định. Cập nhật không đồng bộ khai thác thời gian trì hoãn cho phép trong cập nhật dữ liệu được xác định bởi công nghệ xử lý dữ liệu trong các hệ thống thông tin và ý nghĩa của các bài toán giải quyết trong đó.

II. ĐẶT VẤN ĐỀ

Hiện tại KNT được hỗ trợ trong ba hệ quản trị cơ sở dữ liệu thương mại là Microsoft SQL Server, Oracle và IBM DB2. Tuy nhiên, bài viết [6] đưa ra khả năng khai thác ý tưởng KNT trong các hệ quản trị cơ sở dữ liệu kể cả thương mại và phi thương mại không hỗ trợ KNT. Trong SQL Server (từ phiên bản 2000), KNT được gọi là khung nhìn chỉ mục hoá

(indexed views) và chúng được tự động cập nhật đồng bộ với các thay đổi dữ liệu. Các tài liệu hướng dẫn của IBM DB2, Oracle (từ phiên bản 8.15) và các công trình nghiên cứu khác đề cập đến khả năng cập nhật không đồng bộ các KNT. Tuy nhiên, khả năng cập nhật không đồng bộ đó được các tài liệu mô tả thuần túy về mặt kỹ thuật cài đặt [2] và các thuật toán [5] thực hiện cập nhật không đồng bộ KNT trên cơ sở không thể truy cập các dữ liệu cần thiết hoặc hệ quản trị cơ sở dữ liệu không biết về sự tồn tại của các KNT để thực hiện cập nhật tức thời như một phần của giao tác thay đổi dữ liệu các bảng gốc trong những khoảng thời gian nào đó. Tác giả chưa gặp tài liệu nào phân tích lý do, ý nghĩa của cập nhật không đồng bộ và khi nào có thể chọn nó.

III. CẬP NHẬT KHÔNG ĐỒNG BỘ

3.1 Phân tích về cập nhật không đồng bộ

Ngoài các lý do thiếu thông tin nêu trên, khả năng ứng dụng cơ chế cập nhật không đồng bộ còn dựa trên cơ sở trên thực tế có các bài toán của hệ thống thông tin đòi hỏi chỉ tính đến một phần trong số các thay đổi dữ liệu trong các bảng gốc một cách không chậm trễ. Từ đó cũng xuất hiện khả năng trì hoãn cập nhật KNT, rõ ràng, khả năng đó cho phép giảm số lượng các cập nhật KNT.

Không chỉ mỗi sự kiện sửa đổi dữ liệu trong các bảng gốc ảnh hưởng đến giới hạn (thời lượng) trì hoãn có thể trong cập nhật KNT, mà cả cách thức khai thác các dữ liệu đó. Chẳng hạn, khai báo thêm điểm một môn học nào đó cho một sinh viên hầu như không gây ra một thay đổi đáng kể nào đến điểm trung bình cho toàn khoa, toàn ngành, có thể, thậm chí là môn học đó; bán thêm một vài món hàng không ảnh hưởng đến doanh thu toàn cầu, doanh thu theo châu lục, doanh thu theo từng vùng, hoặc thậm chí doanh thu theo quốc gia. Tuy nhiên, cũng tồn tại nhiều dữ liệu, mà sự trì hoãn cập nhật chúng vào các KNT có thể dẫn đến các hậu quả nghiêm trọng. Chẳng hạn, trì hoãn trong việc cố định sự thay đổi của số dư tài khoản ngân hàng của có thể cho phép khách hàng rút từ tài khoản số tiền lớn hơn số dư thực tế.

Các ví dụ trên đây cho thấy sự cần thiết của việc phân tích dữ liệu để đưa ra cơ chế cập nhật các KNT. KNT không chỉ mang lại lợi ích,

mà nó còn gây ra một số tác hại là chi phí tài nguyên để duy trì nó. Số lượng KNT trong hệ thống có thể bị giới hạn dựa trên chi phí tổng thể để duy trì các KNT trong hệ thống, chẳng hạn, chi phí về ổ cứng, bộ nhớ, thời gian CPU. Danh sách và số lượng các truy vấn được tạo KNT sẽ giảm đáng kể nếu chúng ta phân tích hệ thống thông tin và nhóm các truy vấn [7] nhằm tạo ra các KNT, mà mỗi KNT được sử dụng để trả lời nhiều truy vấn khác nhau [8–9]. Ngoài ra, danh sách đó có thể sẽ được điều chỉnh vì số lượng cập nhật KNT giảm. Từ đây xuất hiện khả năng tạo KNT cho nhiều truy vấn khác. Đặc biệt, trì hoãn các cập nhật KNT có thể giúp giảm tải hệ thống trong các giai đoạn cao điểm tải [7]. Thông qua các điểm đó, cơ chế cập nhật không đồng bộ có thể cho phép nâng cao rõ rệt năng suất tổng thể của hệ thống.

Chẳng hạn, dữ liệu mà KNT_01 sử dụng thay đổi n ($n > 1$) lần trong một khoảng thời gian nào đó. Có ba trường hợp có thể xảy ra. A) Nếu KNT_01 được sử dụng thường xuyên sau mỗi thay đổi, thì cập nhật đồng bộ KNT_01 sau mỗi thay đổi mang lại lợi ích tuyệt đối là điều hiển nhiên. B) Nếu KNT_01 chỉ được sử dụng sau lần thay đổi thứ n và sau một thời gian ít tải của hệ thống, thì cập nhật không đồng bộ vào thời ít tải đó thường có lợi nhất. C) Bởi vì, cập nhật một lần cho n thay đổi không phải bao giờ cũng tốt hơn cập nhật n lần. Nếu có thể cập nhật gia tăng và cập nhật gia tăng n lần tương ứng với n thay đổi dữ liệu gốc có chi phí cao hơn một lần cập nhật hoàn toàn hoặc cập nhật gia tăng 1 lần cho n thay đổi, thì thực hiện cập nhật một lần có lợi hơn, ngược lại, cập nhật gia tăng n lần có lợi hơn. Trong trường hợp cập nhật một lần được lựa chọn, thông thường, nếu có khả năng cập nhật gia tăng thì cập nhật gia tăng cho n thay đổi đòi hỏi chi phí thấp hơn cập nhật hoàn toàn và thông thường nó được sử dụng. Ngược lại, sử dụng cập nhật hoàn toàn. Nếu không có khả năng cập nhật gia tăng, dễ thấy cập nhật hoàn toàn n lần cho n lần thay đổi dữ liệu gốc đòi hỏi chi phí gấp n lần so với chi phí một lần cập nhật hoàn toàn sau thay đổi thứ n . Rõ ràng, khi đó cập nhật không đồng bộ một lần cho n thay đổi có lợi tuyệt đối so với cập nhật đồng bộ n lần.

Tùy thuộc vào sự trì hoãn cho phép trong cập nhật dữ liệu trong các KNT, có thể phân loại dữ liệu theo 4 loại sau:

a) Dữ liệu đòi hỏi phải cập nhật KNT không chậm trễ khi có sự thay đổi.

b) Dữ liệu cho phép trì hoãn điều chỉnh KNT trong một giới hạn thời gian T_i nào đó khi có sự thay đổi.

c) Dữ liệu đòi hỏi phải cập nhật KNT vào những thời điểm nhất định trong quá trình hoạt động của hệ thống. Chẳng hạn, thời điểm cập nhật có thể là tính doanh thu trong ngày vào cuối ngày, thời điểm bắt đầu xếp học bổng cho sinh viên, giờ giải lao, giờ nghỉ trưa, cuối ngày làm việc, giờ thay ca,...

d) Các dữ liệu được xử lý bởi các hàm thống kê có thể cho phép không tính lại KNT cho đến khi chưa tích lũy những thay đổi đến một tổng tới hạn nào đó.

3.2 Thuật toán xác định thời điểm thực hiện cập nhật không đồng bộ

Cơ chế cập nhật không đồng bộ tính đến việc phân tích dữ liệu trên quan điểm yêu cầu tính thực tiễn của chúng cho từng người sử dụng, cũng như thời gian biểu hằng ngày của tổ chức kinh tế - xã hội. Sau đây là các bước của thuật toán đơn giản xác định thời điểm cập nhật không đồng bộ các KNT.

Giả sử ta có tập hợp các truy vấn $Z = \{Z_i | i=1,2,...,N\}$, với Z_i - truy vấn đại diện cho nhóm truy vấn G_i , N - tổng số nhóm.

Bước 1. Đối với mỗi tập hợp các truy vấn $Z_i^G = \{Z_{i_j}^G | j=1,2,...,N_{i_j}^G\}$ tạo nên nhóm G_i , xác định các máy trạm làm việc của người dùng. Mỗi truy vấn $Z_{i_j}^G$ tương ứng với một tập hợp nào đó các trường bảng $F_{i_j} = \{F_{i_{jl}} | l=1,2,...,N_{F_j}\}$. Trong đó, $N_{i_j}^G$ - số truy vấn trong nhóm G_i , N_{F_j} - số trường chọn lựa của truy vấn $Z_{i_j}^G$.

Bước 2. Bằng cách điều tra người dùng cho mỗi trường trong $F_i = \bigcup_{j=1}^{N_{i_j}^G} F_{i_j}$ xác định khoảng thời gian cho phép tối đa $T_{i_{j_l}}^F$ giữa thời điểm thay đổi dữ liệu và thời điểm cần phải cập nhật thay đổi đó trong KNT. Gọi nó là khoảng

thời gian cập nhật trường (dữ liệu). Khi đó tập hợp các khoảng thời gian cập nhật các trường cho nhóm sẽ là $T_i^F = \bigcup_{j=1}^{N_{i_j}^G} T_{i_j}^F$.

Bước 3. Tìm giá trị nhỏ nhất trong tập hợp $T_{i_j}^F$ xác định ranh giới cao nhất của khoảng thời gian cập nhật KNT $T_{i_{\min}}^F = \min(T_i^F)$.

Bước 4. Phân tích chế độ làm việc của các nhân viên tại trạm làm việc và sử dụng những KNT cụ thể nhằm mục đích làm rõ các thời gian nghỉ có thể trong công việc $T_{i_x}^R = \{T_{i_x}^R | x=1,2,...,N_{T_x}^R\}$. Với $N_{T_x}^R$ là số lượng các giờ nghỉ có thể.

Bước 5. Tìm giá trị nhỏ nhất trong tập hợp $T_{i_{\min}}^R = \min(T_i^R)$.

Bước 6. Bằng cách so sánh tương ứng $T_{i_{\min}}^F$ và $T_{i_{\min}}^R$ xác định khả năng trì hoãn cập nhật KNT cho đến các thời gian nghỉ trong công việc.

Bước 7. Lặp lại các bước từ 1-6 cho tất cả các truy vấn trong tập hợp Z . Bằng cách phân tích tất cả $T_{\min}^F = \{T_{i_{\min}}^F | i=1,2,...,N\}$ và $T_{\min}^R = \{T_{i_{\min}}^R | i=1,2,...,N\}$ đưa ra tập hợp các truy vấn Z_{um} cần cập nhật KNT tại mỗi thời gian nghỉ thứ m .

Có thể cập nhật các KNT với các truy vấn chứa các hàm thống kê theo một chế độ đặc biệt. Thông thường, một vài sửa đổi dữ liệu làm ảnh hưởng không nhiều về lượng đối với kết quả thống kê khi số lượng dữ liệu được thống kê lớn. Vì thế, tồn tại khả năng đánh giá vùng các giá trị có thể của các dữ liệu được thống kê và sự ảnh hưởng có thể đến kết quả truy vấn từ một thay đổi dữ liệu. Điều đó cho phép trì hoãn cập nhật KNT đến thời điểm đạt được một tổng tới hạn các thay đổi của các dữ liệu được thống kê. Có thể xác định tổng đó bằng cách đếm số các thay đổi trong dữ liệu gốc.

III KẾT LUẬN

Thông qua nghiên cứu vấn đề, có thể kết luận:

- Khả năng ứng dụng cơ chế cập nhật không đồng bộ có thể dựa trên cơ sở trên thực tế có các bài toán của hệ thống thông tin đòi hỏi phải tính đến chỉ một phần trong số các thay đổi dữ liệu trong các bảng gốc một cách không chậm trễ. Từ đó xuất hiện khả năng trì hoãn cập nhật KNT.

- Cập nhật không đồng bộ cho phép giảm số lượng các cập nhật KNT. Đặc biệt, trì hoãn các cập nhật KNT có thể giúp giảm tải hệ thống

trong các giai đoạn cao điểm tải. Thông qua các điểm đó, nó cho phép nâng cao năng suất tổng thể của hệ thống thông tin.

- Bài viết phân tích về cập nhật không đồng bộ và đề nghị thuật toán xác định các thời điểm cập nhật không đồng bộ các KNT. Vì yêu cầu cập nhật xuất phát từ nhu cầu sử dụng KNT của người dùng, nên không thể loại trừ sự tham gia của người sử dụng trong thuật toán đơn giản đó thông qua các kết quả điều tra.

TÀI LIỆU THAM KHẢO

1. SQL Server 2005 manual. [http://msdn.microsoft.com/en-us/library/ms191432 \(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms191432 (SQL.90).aspx). 15/12/2008.
2. Eric Hanson and Contributors. Improving Performance with SQL Server 2005 Indexed Views. <http://www.microsoft.com/technet/prodtechnol/sql/2005/impprfiv.msp>. 20/01/2009.
3. Oracle 11g manual. http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/statements_6002.htm. 17/01/2009.
4. Thomas Kyte; Expert one-on-one Oracle. Apress, 2003.
5. L.S. Colby, T. Griffin, L. Libkin, I.S. Mumick, H. Trickey; Algorithms for deferred view maintenance; ACM SIGMOD Record V. 25 Issue 2, 06/1996.
6. Nguyễn Trần Quốc Vinh; Ứng dụng khung nhìn thực để nâng cao tốc độ thực thi truy vấn; Tạp chí Khoa học và Công nghệ - Đại học Đà Nẵng, số 1(30), 2009.
7. Quoc Vinh Nguyen Tran, A. B. Kungurtsev; The analysis of the information systems for applying the materialized views // Метод анализа информационной системы для применения материализованных представлений. Холодильна Техніка і Технологія. Одеса, 2005. – 2(94). – tr. 102-105.
8. A. B. Kungurtsev, Quoc Vinh Nguyen Tran; The analysis of feasibility of applying the materialized views in information systems // Анализ возможности применения МП в ИС. Тр. Одесск. политехн. ун-та. Одеса, 2003. – 2(20). – tr. 102-106.
9. A. B. Kungurtsev, Quoc Vinh Nguyen Tran; Data extraction from materialized views in information systems // Извлечение данных из материализованных представлений в информационных системах. Тр. Одесск. политехн. ун-та. Одеса, 2005. – 1(23). – tr. 82-87.

Địa chỉ liên hệ: Nguyễn Trần Quốc Vinh - Tel: 0914.780.898, Email: ntquocvinh@gmail.com
Trường Đại học Kinh tế, Đại học Đà Nẵng