

Nâng cao hiệu năng xử lý giao tác với mô hình CSDL trong bộ nhớ

Vũ Bá Duy, Dư Phương Hạnh*

Trường Đại học Công nghệ, ĐHQGHN, 144 Xuân Thủy, Hà Nội, Việt Nam

Nhận ngày 25 tháng 7 năm 2014

Chỉnh sửa ngày 27 tháng 8 năm 2014; Chấp nhận đăng ngày 19 tháng 9 năm 2014

Tóm tắt: Các hệ quản trị cơ sở dữ liệu (CSDL) truyền thống chủ yếu được xây dựng dựa trên việc khai thác các máy tính đơn nhân CPU và thiết bị lưu trữ kiểu từ tính. Các mô hình này chưa khai thác được hết những thế mạnh công nghệ của các chip đa nhân hay những hệ thống máy tính cụm và dung lượng càng ngày càng lớn của bộ nhớ chính. Bài báo này đề cập đến việc sử dụng mô hình H-Store nhằm mục đích khai thác được những kiến trúc đa nhân cũng như việc đưa toàn bộ dữ liệu vào trong bộ nhớ, từ đó có thể nâng cao được hiệu năng xử lý các giao tác đồng thời từ phía người dùng. Phần thực nghiệm với bài toán thực tế trong Đại học Quốc gia Hà Nội liên quan đến việc quá tải hệ thống khi sinh viên đăng ký tín chỉ cho phép minh chứng được tính đúng đắn của mô hình này.

Từ khoá: CSDL trong bộ nhớ, xử lý giao tác, H-Store.

1. Giới thiệu

Với các hệ cơ sở dữ liệu truyền thống, dữ liệu được lưu trữ trên thiết bị từ tính, phổ biến vẫn là đĩa cứng. Trong quá trình xử lý, dữ liệu có thể nạp vào bộ nhớ đệm để tăng tốc độ xử lý, tuy nhiên, các phương pháp xử lý cơ bản vẫn phải dựa trên thao tác với đĩa cứng. Chính vì thế mà hiệu năng xử lý không những phụ thuộc các thao tác xử lý dữ liệu mà còn phụ thuộc rất lớn vào các thao tác đọc/ghi trên đĩa. Vì thế nếu giảm thiểu hay loại bỏ được các thao tác truy xuất đĩa sẽ có ý nghĩa với hiệu năng xử lý của hệ thống [1].

Khi bộ nhớ bán dẫn ngày càng rẻ, phương án tiếp cận đưa tất cả dữ liệu vào bộ nhớ chính trở nên khả thi hơn. Phương án này hình thành cách tiếp cận xây dựng các hệ quản trị CSDL trong bộ nhớ (In Memory Database - IMDB). Với mô hình IMDB, việc tối ưu hoá truy vấn sẽ không cần phải quan tâm nhiều đến chi phí cho phần vào/ra trên thiết bị lưu trữ, một trong những yếu tố then chốt ảnh hưởng đến hiệu năng xử lý giao tác [2]. Ưu điểm này đến từ bản chất của bộ nhớ chính: dữ liệu được truy xuất theo kiểu RAM (Random Access Memory) thay vì kiểu truy cập trực tiếp như ổ đĩa cứng.

Trong bài báo này, chúng tôi sẽ giới thiệu một trong những mô hình IMDB hiện đại hiện nay: H-Store với thể hiện cài đặt là VoltDB [3].

* Tác giả liên hệ. ĐT: 84-913365832.
E-mail: hanhdp@vnu.edu.vn

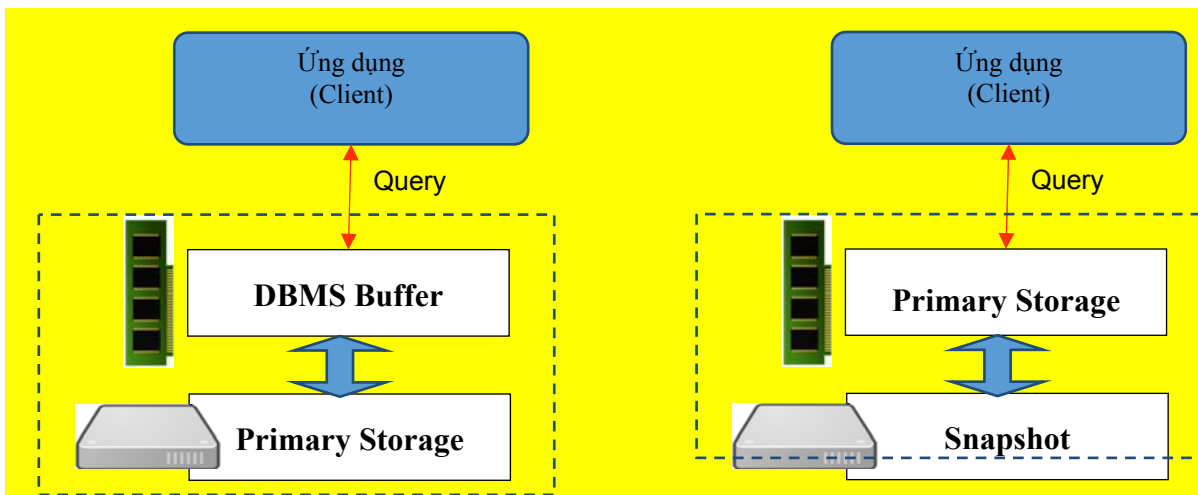
Dựa trên hệ quản trị này, chúng tôi cũng đã tiến hành thử nghiệm với bài toán thực tế đăng ký lớp tín chỉ trong Đại học Quốc gia Hà Nội (ĐHQGHN). Kết quả thực nghiệm cho thấy hệ thống mô phỏng cho hiệu năng của mô hình IMDB luôn vượt trội so với mô hình CSDL truyền thống và cho phép giải quyết được vấn đề quá tải hệ thống.

Phần còn lại của bài báo được tổ chức như sau: mục 2 giới thiệu về mô hình CSDL trong bộ nhớ, mục 3 trình bày về kiến trúc hệ thống H-Store và thể hiện cụ thể của H-Store với hệ quản trị VoltDB; mục 4 được giành để giới thiệu bài toán đăng ký lớp tín chỉ trực tuyến

trong ĐHQGHN và mục 5 tổng hợp những kết quả thực nghiệm mô hình giải pháp và những đánh giá, so sánh; cuối cùng là phần kết luận và một số định hướng nghiên cứu tiếp.

2. Mô hình cơ sở dữ liệu trong bộ nhớ

Cơ sở dữ liệu trong bộ nhớ (IMDB) là mô hình quản lý CSDL mà toàn bộ dữ liệu được xử lý trong bộ nhớ chính của hệ thống tính toán [1]. Mô hình này khác biệt so với mô hình truyền thống, ở đó chỉ một phần dữ liệu được xử lý trong bộ nhớ.



Hình 1. Mô hình IMDB.

Bộ nhớ chính của một máy tính có các tính chất khác nhau so với các đĩa từ, và sự khác biệt này có ý nghĩa lớn về thiết kế và hiệu năng của hệ quản trị cơ sở dữ liệu khi thi hành các giao tác trực tuyến (Online Transaction Processing – OLTP). Sự khác biệt này được thể hiện qua:

* Thời gian truy xuất (đôi khi sử dụng thuật ngữ latency) từ bộ nhớ chính nhỏ hơn so với đĩa do bộ nhớ chính có phương thức truy cập

ngẫu nhiên còn đĩa từ có phương thức truy cập trực tiếp.

* So với IMDB, việc lưu trữ trên đĩa từ thuận lợi hơn việc đảm bảo tránh mất dữ liệu đối với những sự cố kỹ thuật. Tuy nhiên, với những cơ chế đồng bộ và lưu vết thích hợp, nhược điểm này của IMDB hoàn toàn có thể được khắc phục [4].

* IMDB cho phép nâng cao hiệu năng xử lý giao tác dựa trên bản chất của phương thức truy cập ngẫu nhiên: giảm thời gian tính toán tối ưu

xử lý truy vấn, giảm thiểu thời gian truy xuất đến dữ liệu, ...

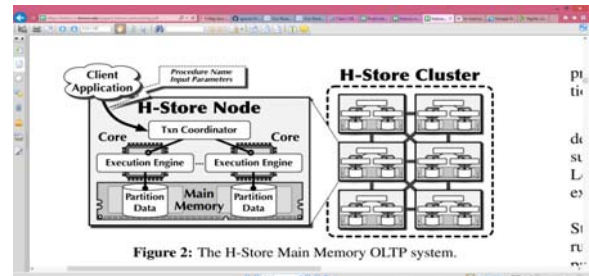
Hiện nay, mô hình IMDB đã được cài đặt trên nhiều hệ quản trị CSDL, cả thương mại lẫn mã mở hay trong nghiên cứu thực nghiệm. Các hệ quản trị thương mại có thể kể đến giải pháp TimesTen của Oracle [5], DB2 BLU của IBM [6], Hekaton của Microsoft [7]... Một số hệ quản trị được xây dựng phục vụ nghiên cứu và cộng đồng mã mở có thể kể đến như BigMemory của hãng Terracotta [8], H-Store [9], VoltDB [10], ... Trong số các sản phẩm trên, hệ VoltDB là hệ quản trị được cài đặt trực tiếp từ mô hình kiến trúc H-Store, sản phẩm nghiên cứu của ba đơn vị: trường Đại học Brown, Carnegie Mellon và viện MIT, Mỹ. Đây là hệ được đánh giá có hiệu năng xử lý giao tác vượt trội, cho phép khai thác được tối đa những thế mạnh về công nghệ tính toán như bộ nhớ chính dung lượng lớn, đa lõi, đa chip, ... [4].

3. Kiến trúc H-Store và VoltDB

H-Store là hệ quản trị được xây dựng theo mô hình quan hệ, lưu trữ theo bộ hoàn toàn trên bộ nhớ chính và có tính phân tán cao [3,4]. Hệ quản trị này có thể chạy trên môi trường cụm máy tính theo mô hình không chia sẻ.

Một thể hiện H-Store được thi hành trên một cụm máy tính với 2 hoặc nhiều nốt (nodes), mỗi nốt là một hệ thống máy tính vật lý đơn. Mỗi nốt có thể chứa một hoặc nhiều *sites* thi hành và một bộ điều phối giao tác (transaction coordinator). *Site* là một thực thể logic hoạt động độc lập tương ứng một luồng (thread) trong hệ thống, xử lý giao tác OLTP từ phía người dùng dưới sự điều khiển của bộ điều phối giao tác. Bộ điều phối này có nhiệm vụ đảm bảo tính có thể tuần tự được của các giao tác trong những bộ điều phối từ các nốt khác.

Trong H-Store, mỗi site sẽ được cài đặt trực tiếp trên một lõi (core) của bộ vi xử lý và độc lập so với các sites khác [8].



Hình 2. Mô hình hệ thống H-Store.

Mọi quan hệ/bảng trong cơ sở dữ liệu của H-Store được phân đoạn ngang thành nhiều đoạn (fragments, hay đôi khi còn gọi là shards) dựa trên thuộc tính phân mảnh. Những đoạn liên quan với nhau từ nhiều quan hệ có thể nhóm lại thành một phân vùng (partition). Mỗi phân vùng chỉ được chứa một *site*[11].

Các bộ trong H-Store được lưu trữ trong bộ nhớ chính trên mỗi nốt. Ứng dụng OLTP từ phía người dùng thực hiện các truy vấn tới hệ thống H-Store để thực thi các thủ tục lưu trữ (*stored procedures*) đã được định nghĩa trước. Mỗi thủ tục được xác định bởi một tên duy nhất và chứa mã điều khiển có cấu trúc pha trộn với ngôn ngữ SQL (Structure Query Language) được tham số hóa. Một thể hiện của thủ tục được khởi tạo bởi ứng dụng OLTP là một giao tác (transaction). Giao tác trong H-Store đảm bảo đầy đủ tính ACID (Atomicity, Consistency, Isolation, Durability) đối với một hệ quản trị truyền thống.

Hiện nay, ngoài việc đưa ra hệ thống H-Store phục vụ nghiên cứu và đánh giá, nhóm nghiên cứu H-Store đã đưa ra bản thương mại VoltDB [9] để có thể triển khai các ứng dụng thực tiễn. Dựa trên mô hình H-Store, chúng tôi sẽ tiến hành thử nghiệm với bài toán đăng ký lớp môn học tin chỉ tại ĐHQGHN trên VoltDB.

4. Bài toán đăng ký lớp môn học tại ĐHQGHN

Hiện nay ĐHQGHN đã cho phép các sinh viên đăng ký trực tuyến các lớp môn học đầu mỗi kỳ dựa vào danh sách các lớp môn học được nhà trường xây dựng. Tuy nhiên, hệ thống đăng ký này được xây dựng dựa trên hệ quản trị cơ sở dữ liệu truyền thống. Trong thực tế, vào thời điểm mở thông báo, hầu hết các sinh viên đều thực hiện đăng ký các môn học mình dự kiến học đồng thời, dẫn đến tình trạng thường xuyên quá tải cho hệ thống.

Trong bài báo này, như đã trình bày ở trên, chúng tôi nghiên cứu các đặc tính và hiệu suất của hệ thống theo hướng đưa tất cả dữ liệu vào bộ nhớ chính, trên cơ sở đó thiết kế các bảng dữ liệu và cài đặt thử nghiệm một số mô đun cơ bản của hệ thống quản lý đào tạo (đăng ký lớp môn học, cập nhật điểm) trên 2 hệ quản trị VoltDB và MySQL để so sánh hiệu năng. Trong thực nghiệm này, dữ liệu về môn học, ngành, chương trình đào tạo được chúng tôi sử dụng toàn bộ những dữ liệu tại trường Đại học Công nghệ - ĐHQGHN.

Quy trình đăng ký lớp môn học được xác lập như sau:

1. Đầu học kỳ phòng đào tạo lập danh sách lớp môn học đối với mỗi chương trình đào tạo

- tên lớp,
- tên môn học
- giáo viên
- phòng học
- thời gian
- Số sinh viên tối đa

2. Sinh viên thuộc chương trình nào thì có thể đăng ký lớp học thuộc chương trình đó với điều kiện

- Lớp đó chưa quá số tối đa,

- Sinh viên không vi phạm môn điều kiện tiên quyết đối với môn học,

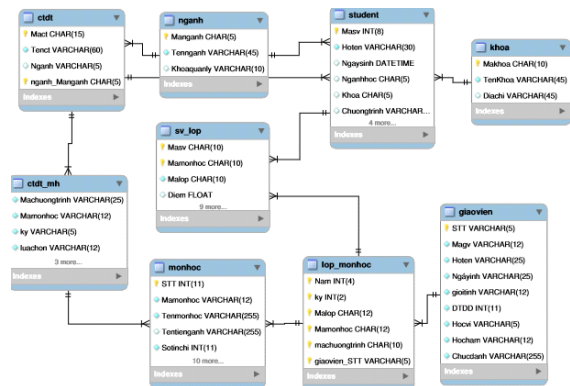
- Sinh viên chưa đăng kí quá số tín chỉ đã được phép trong 1 kỳ.

- Không trùng lịch học với môn khác đã đăng kí

Trên thực tế, hệ thống quản lý đào tạo hiện nay chưa kiểm tra được hết các ràng buộc, còn dẫn đến rất nhiều tình trạng sinh viên vi phạm điều kiện tiên quyết, trùng lịch, ...

5. Thử nghiệm

Chúng tôi đã tiến hành mô hình hoá bài toán trên theo mô hình quan hệ - thực thể và tiến hành cài đặt ứng dụng thử nghiệm trên cả hai hệ quản trị CSDL: IMDB với VoltDB và truyền thống với MySQL.



Hình 3. Lược đồ CSDL thực nghiệm.

Môi trường thực nghiệm được tiến hành bước đầu trên cùng máy tính cấu hình CPU Intel(R) Core(TM) i5-3317U, 4GB RAM, cài hệ điều hành Ubuntu 13.04.3 LTS.

Toàn bộ lược đồ cũng như yêu cầu đối với hệ thống đăng ký lớp môn học tín chỉ được cài đặt trên MySQL x86-64 phiên bản 5.6 và VoltDB phiên bản 4.0. Để đánh giá hiệu năng cả hai hệ quản trị này, chúng tôi dựa trên chuẩn

công nghiệp TPC-C để đánh giá hiệu năng với các hệ quản trị CSDL (TPC-C - Transaction Processing Performance Council) [12] và tiến hành hiệu chỉnh để phù hợp với bài toán đặt ra.

Kết quả thực nghiệm với cả hai hệ quản trị nêu trên thu được như sau:

Bảng 1. Kết quả đánh giá hiệu năng tổng hợp

Số sinh viên	MySQL (giao tác/s)	VoltDB (4 partiton) (giao tác/s)
10.000	42.6	2375.5
40.000	42.8	2358.9
60.000	42.5	2350.2
100.000	30.8	2501.5

Với kết quả thực nghiệm đạt được, ta thấy: với số lượng sinh viên tăng 10 lần, hiệu suất trung bình của ứng dụng trên MySQL đã giảm đáng kể, trong khi trên VoltDB vẫn ổn định và nhanh hơn khoảng 55 lần so với MySQL. Một điểm cần nhấn mạnh thêm ở đây là hiệu năng chỉ khoảng 40 giao tác/giây với hệ quản trị truyền thống thì tại những thời điểm đăng ký lớp tín chỉ, hoàn toàn có thể xả ra tình trạng quá tải, thậm chí cả chờ đợi vô hạn (deadlock) khi có cỡ khoảng một vài trăm sinh viên truy cập đồng thời vào hệ thống.

Thực nghiệm trên cũng cho phép khẳng định, tuy có hệ thống tính toán hiệu năng vừa phải (máy tính cá nhân), VoltDB đã cho phép xử lý đồng thời truy vấn của khoảng 2300 sinh viên. Với lượng sinh viên như ở trường Đại học Công nghệ (~600 sinh viên mỗi năm), hiệu năng này cho phép đủ đáp ứng được tất cả yêu cầu đăng ký lớp tín chỉ của sinh viên toàn trường.

7. Kết luận

Mô hình H-Store thực sự khai thác được những thế mạnh về những kiến trúc đa nhân, hệ thống cluster và đặc biệt là mô hình CSDL trong bộ nhớ. Với những tính năng đó, hệ quản

trị này cho phép nâng cao được hiệu năng xử lý các giao tác đồng thời từ phía người dùng, minh chứng được ưu điểm của IMDB so với hệ quản trị truyền thống.

Thông qua thực nghiệm với bài toán sinh viên đăng ký trực tuyến lớp môn học tín chỉ, chúng tôi đã minh chứng rõ rệt sự quá tải của hệ thống này với mô hình CSDL truyền thống. Và dựa trên cài đặt của H-Store, VoltDB, kết quả thực nghiệm cho phép khẳng định được IMDB giải quyết triệt để vấn đề quá tải của hệ thống đăng ký lớp môn học trong ĐHQGHN thông qua việc nâng cao hiệu năng (khoảng 55 lần) so với hệ quản trị CSDL truyền thống.

Trong thời gian tới, chúng tôi sẽ tiến hành phối hợp cùng các đơn vị khác trong ĐHQGHN để có thể thử nghiệm thực tế mô hình giải pháp nêu trên, trực tuyến trên Internet.

Lời cảm ơn

Công trình này được tài trợ một phần từ đề tài nghiên cứu mã số CN.12.02, cấp trường Đại học Công nghệ - ĐHQGHN.

Tài liệu tham khảo

- [1] J. Baulier, P. Bohannon, S. Gogate, S. Joshi, C. Gupta, A. Khivesera, H. F.Korth, P. McIlroy, J. Miller, P. P. S. Narayan, M. Nemeth, R. Rastogi, A. Silberschatz, and S. Sudarshan. Dablitiz: A high performance main-memory storage manager. VLDB, pages 701–714, 1998.
- [2] Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 6th Edition, 2011 - Addison-Wesley, ISBN-10: 0136086209, ISBN-13: 9780136086208.
- [3] R. Kallman, H. Kimura, J. Natkins, A. Pavlo, A. Rasin, S. Zdonik, E. P. C. Jones, S. Madden, M. Stonebraker, Y. Zhang, J. Hugg, and D. J. Abadi, "H-Store: a High-Performance, Distributed Main Memory Transaction Processing System," Proc.

- VLDB Endow., vol. 1, iss. 2, pp. 1496-1499, 2008.
- [4] J. DeBrabant, A. Pavlo, S. Tu, M. Stonebraker, and S. Zdonik, "Anti-Caching: A New Approach to Database Management System Architecture," Proc. VLDB Endow., vol. 6, pp. 1942-1953, 2013.
- [5] Oracle TimesTen Products and Technologies. Technical report, February 2013.
- [6] DB2 with BLU Acceleration, <http://www-01.ibm.com/software/data/db2/linux-unix-windows/db2-blu-acceleration/> retrieved on 25/12/2013.
- [7] C. Diaconu, C. Freedman, E. Ismert, P.-A. Larson, P. Mittal, R. Stonecipher, N. Verma, and M. Zwilling. Hekaton: SQL Server's Memory-Optimized OLTP Engine. In SIGMOD, pages 1-12, 2013.
- [8] BigMemory, <http://terracotta.org/products/bigmemory> retrieved on 25/12/2013.
- [9] H-Store <http://hstore.cs.brown.edu/>, last retrieved on 12/1/2014.
- [10] VoltDB, <http://voldb.com/>, last retrieved on 28/1/2014.
- [11] E. P. C. Jones, D. J. Abadi, and S. Madden (2010), "Low overhead concurrency control for partitioned main memory databases" in SIGMOD '10: Proceedings of the 2010 international conference on Management of data, New York, NY, USA, pp. 603-614.
- [12] The Transaction Processing Council. TPC-C Benchmark (Revision 5.9.0). http://www.tpc.org/tpcc/spec/tpcc_current.pdf, June 2007.

Improving the Transaction Processing Performance by Using In-memory Database Model

Vũ Bá Duy, Đur Phương Hạnh

VNU University of Engineering and Technology, 144 Xuân Thủy, Hanoi, Vietnam

Abstract: The traditional database management systems are based mainly on the exploitation of single-CPU computer and magnetic storage devices. These models do not fully exploit the technological advantages of large amount of main memory, even terabytes. This paper presents an empirical approach to use the H-Store model in order to improve the transaction processing performance and evaluate it based on the real problem at Vietnam National University: the online course registration system is always over-load. Our experiment allows us to validate the advantages of in-memory database model and to overcome the real problem of the online course registration.

Keywords: In-memory database, H-Store, transaction processing.