

MỤC LỤC

1. A NEW APPROACH FOR CLUSTERING ALGORITHMS IN PROCESS MINING	
<i>Hong-Nhung Bui, Tri-Thanh Nguyen, Quang-Thuy Ha</i>	5
2. LƯỢC ĐỒ THỦY VĂN RỒNG TRÊN ẢNH SỐ DỰA VÀO PHÂN TÍCH SVD	
<i>Nguyễn Hiếu Cường, Nguyễn Đình Thi, Cao Thị Luyên</i>	13
3. NÂNG CAO CHẤT LƯỢNG ẢNH TRONG GIẤU TIN THUẬN NGHỊCH DỰA TRÊN PHÉP BIẾN ĐỔI HISTOGRAM	
<i>Nguyễn Kim Sao, Mai Thanh Hồng.....</i>	20
4. CẢI TIẾN THUẬT TOÁN TỐI ƯU ĐÀN KIẾN GIẢI BÀI TOÁN MLP	
<i>Đặng Thị Thu Hiền</i>	29
5. VỀ TRƯỜNG HỮU HẠN $GF(2^N)$ DỰA TRÊN PHÉP TOÁN MODULAR ĐA THỨC	
<i>Đỗ Văn Đức.....</i>	37
6. SỬ DỤNG MÔ HÌNH CRFS XÁC ĐỊNH Ý ĐỊNH NGƯỜI DÙNG TRONG LĨNH VỰC DU LỊCH	
<i>Lương Thái Lê</i>	44
7. MỘT PHƯƠNG PHÁP TRÍCH CHỌN MẪU DỮ LIỆU GIẢI BÀI TOÁN HỒI QUY	
<i>Nguyễn Đức Dư, Hoàng Văn Thông.....</i>	51
8. ƯỚC LƯỢNG THÔNG TIN VỀ LƯỒNG GIAO THÔNG	
<i>Lại Mạnh Dũng</i>	62
9. CÁC GIẢI THUẬT PHÂN CỤM TRONG PHÂN ĐOẠN ẢNH	
<i>Nguyễn Thị Hồng Hoa</i>	70
10. ỨNG DỤNG BỘ LỌC KALMAN GIẢM THIỂU SAI SỐ VẬN TỐC TRONG DỮ LIỆU GPS	
<i>Nguyễn Thị Yến Liên, Nghiêm Trung Dũng.....</i>	76
11. ỨNG DỤNG PHẦN MỀM VISUM DỰ BÁO NHU CẦU LOGISTICS TRÊN CÁC HÀNH LANG VẬN TẢI CHIẾN LƯỢC CỦA VIỆT NAM- NGHIÊN CỨU TRƯỜNG HỢP CỤ THỂ HÀNH LANG HÀ NỘI - LÀO CAI	
<i>Nguyen Thanh Tu, Nguyen Thi Binh</i>	84

12. CƠ SỞ DỮ LIỆU KHÔNG GIAN TRONG HỆ THỐNG CẢNH BÁO THIÊN TAI	
<i>Trương Xuân Quang</i>	93
13. CƠ CHẾ ĐỒNG BỘ HOÁ LIÊN LƯỢC ĐỒ CHO CÁC ỨNG DỤNG CƠ SỞ DỮ LIỆU PHÂN TÁN	
<i>Hoàng Xuân Tùng</i>	99
14. ƯU TIÊN ĐỀN TÍN HIỆU CHO XE BUÝT THEO ASP	
<i>Nguyễn Thị Thanh Hương.....</i>	107
15. ỨNG DỤNG KỸ THUẬT XỬ LÝ ẢNH HÌNH THÁI VÀ LẬP TRÌNH ANDROID PHÁT HIỆN VÀ ĐỊNH VỊ VẾT NÚT MẶT ĐƯỜNG BÊ TÔNG NHỰA	
<i>Lê Nhật Tùng.....</i>	117

CƠ CHẾ ĐỒNG BỘ HOÁ LIÊN LƯỢC ĐỒ CHO CÁC ỨNG DỤNG CƠ SỞ DỮ LIỆU PHÂN TÁN

A FLEXIBLE CROSS-SCHEMA SYNCHRONIZATION MECHANISM FOR DISTRIBUTED DATABASE SYSTEMS

HOÀNG XUÂN TÙNG¹

¹Khoa Công nghệ Thông tin, Trường Đại học Công nghệ, ĐHQG HN

Email: tungxh@vnu.edu.vn

Tóm tắt: Đồng bộ hoá cơ sở dữ liệu là một trong những tính năng quan trọng trong các hệ thống tính toán phân tán. Đồng bộ hoá liên lược đồ (cross-schema) cho phép việc đồng bộ hoá có thể thực hiện được trong môi trường cơ sở dữ liệu không đồng nhất nơi mà các cơ sở dữ liệu có cấu trúc khác nhau vẫn thực hiện đồng bộ dữ liệu được với nhau. Trong bài báo này, chúng tôi giới thiệu XSync, một middleware được thiết kế nhằm phục vụ thực hiện việc đồng bộ hoá liên lược đồ một cách hiệu quả cho các hệ thống sử dụng cơ sở dữ liệu. Với XSync, nhà phát triển phần mềm có thể dễ dàng tích hợp khả năng đồng bộ dữ liệu vào hệ thống IT của mình một cách linh hoạt với chi phí phát triển thấp nhưng vẫn đem lại nhiều lợi ích lớn về mặt tính năng cũng như hiệu năng.

Từ khóa: cross-schema, database, synchronization, ETL

Abstract: Database Data synchronization between distributed databases is an essential feature of computation systems. Cross-schema database synchronization allows data to be transferred among heterogeneous database environments, in which databases are different from each other not only in database platform but also database schema and its internal data. In this paper, we introduce XSync, a middleware that is specially designed to solve the problem of cross-schema synchronization in an efficient way. Using XSync, system developers can easily integrate advance synchronization capabilities into their systems. XSync brings flexibility and greatly extends the systems with low development cost.

Keywords: cross-schema, database, synchronization, ETL

1. ĐẶT VẤN ĐỀ

Cơ sở dữ liệu quan hệ (CSDL hay database) là một phần không thể thiếu trong các hệ thống ứng dụng và dịch vụ hiện nay. Tùy thuộc vào yêu cầu hiệu năng và khả năng mở rộng của ứng dụng, hệ thống có thể được kiến trúc và thiết kế với nhiều CSDL được đồng bộ hoá với nhau dưới nhiều hình thức. Các kịch bản ứng dụng điển hình của đồng bộ hoá CSDL có thể kể đến là đồng bộ hoá giữa local database và cloud database, hay việc nạp dữ liệu vào data warehouse [5]. Hình thức đồng bộ hoá đơn giản nhất có thể kể đến là nhân bản (replication) 0. Khi đó, các CSDL có cấu trúc cũng như là dữ liệu giống hệt nhau. Tuy nhiên

việc đồng bộ hoá sẽ trở nên mạnh mẽ và linh hoạt hơn nếu cho phép các CSDL tham gia đồng bộ hoá được phép khác nhau về cấu trúc (schema) cũng như là dữ liệu bên trong. Nói cách khác, cơ chế đồng bộ hoá được mong chờ là "cross-schema" thay vì cơ chế nhân bản (replication).

Nếu các cơ chế đồng bộ hoá nhân bản là được thực hiện theo hướng "database approach" trong đó đồng bộ hoá được tích hợp thành chức năng của các hệ quản trị cơ sở dữ liệu 0, thì đồng bộ hoá cross-schema là thách thức hơn và thường được thực hiện theo hướng "application approach" [4][6]. Đồng bộ hoá cross-schema do phụ thuộc chặt chẽ vào CSDL của ứng dụng nên việc xây dựng đồng bộ hoá như thế nào sẽ tùy thuộc vào từng ứng dụng. Việc thực hiện triển khai đồng bộ hoá cross-schema cho một hệ thống phân tán do vậy là kém linh hoạt và có chi phí cao do gần như là phải thực hiện lại việc phát triển từ đầu cho từng trường hợp khác nhau. Trong bài báo này, chúng tôi đề xuất một giải pháp hỗ trợ đồng bộ hoá CSDL, tên là XSync, cho các ứng dụng nhằm giúp việc triển khai cross-schema synchronization trở nên linh hoạt và hiệu quả hơn. XSync giải quyết bài toán đồng bộ CSDL theo hướng "semi-application approach" trong đó người dùng thay vì phải phát triển lại chức năng đồng bộ cho ứng dụng của mình sẽ chỉ cần soạn một file cấu hình trong đó định nghĩa chính xác các thuộc tính chuyển đổi giữa CSDL nguồn và CSDL đích. Dựa vào thông tin cấu hình đó, hệ thống XSync sẽ tự động trích xuất dữ liệu từ CSDL nguồn, thực hiện việc chuyển đổi dữ liệu và nạp vào các CSDL đích ở xa qua nhiều các cơ chế kết nối mạng linh hoạt khác nhau như request/reply hay push/pull.

2. CÁC NGHIÊN CỨU LIÊN QUAN

Nhân bản cơ sở dữ liệu hay phân cụm cơ sở dữ liệu [8] là các phương án đồng bộ database nhằm tăng cường khả năng mở rộng của hệ thống. Các phương án đồng bộ này là thuộc nhóm giải pháp dựa trên chức năng của hệ quản trị cơ sở dữ liệu (database-based approach). Các cơ chế này là hữu dụng cho các kịch bản ứng dụng như backup dữ liệu, tăng tốc độ truy xuất database và thường được triển khai trong môi trường mạng nội bộ hoặc data center. Tuy nhiên mô hình này hiếm khi được triển khai thực hiện trong môi trường mạng diện rộng hay đồng bộ giữa cơ sở dữ liệu địa phương và cơ sở dữ liệu đám mây do: (1) thiếu tính linh hoạt trong đồng bộ dữ liệu, (2) lượng dữ liệu đồng bộ lớn và (3) việc mất mát dữ liệu trong quá trình đồng bộ có thể khiến quá trình đồng bộ không còn làm việc được.

Bên cạnh các giải pháp dựa trên hệ quản trị cơ sở dữ liệu, các giải pháp dựa trên ứng dụng (application-based approach) là linh hoạt hơn và có thể được triển khai trong môi trường mạng diện rộng hoặc mạng Internet cho các ứng dụng như local-cloud synchronization. McCormick, Zach, và Douglas C. Schmidt [4] đưa ra một vài kiến trúc cho việc đồng bộ dữ liệu. Các mẫu kiến trúc được trình bày mô tả ra các cấu trúc tổ chức cơ bản của hệ thống đồng bộ tầng ứng dụng, các chức năng và nhiệm vụ của từng hệ thống con, cũng như các luật và các hướng dẫn cho việc tổ chức phối hợp giữa các hệ thống con. Hai kiến trúc tiêu biểu được đưa ra là Asynchronous Data Synchronization and Synchronous

Data Synchronizaton, cho các ứng dụng mobile. Mặc dù các giải pháp trong [4] là linh hoạt và thực tiễn, tuy nhiên các đề xuất này không cung cấp được khả năng chuyển đổi schema cũng như là dữ liệu giữa các database.

Với sự phát triển ngày càng nhanh của mạng internet, việc có được các cơ chế đồng bộ dữ liệu không đồng nhất cho môi trường doanh nghiệp là rất quan trọng. Khá nhiều nghiên cứu (ví dụ như [8][9]) đã được đưa ra nhằm cung cấp các giải pháp dựa trên middleware tương tự như XSync để cung cấp khả năng đồng bộ dữ liệu cho ứng dụng trong nhiều hoàn cảnh và tình huống khác nhau. Tuy nhiên, tương tự như [4] các giải pháp này không chú trọng đến vấn đề chuyển đổi cấu trúc dữ liệu cũng như dữ liệu bên trong giữa các cơ sở dữ liệu.

Gần nhất với nghiên cứu của bài báo này, theo những gì chúng tôi được biết, có lẽ là mô hình đồng bộ ETL [7]. ETL là viết tắt của Extract-Transform-Load, là 3 giai đoạn của mô hình và được thực hiện liên tiếp nhau cho một chu trình đồng bộ dữ liệu:

- + Giai đoạn Extraction: Hệ thống ETL truy xuất vào hệ thống nguồn và trích xuất dữ liệu. Mục đích của giai đoạn này là đưa dữ liệu về dạng phù hợp cho giai đoạn sau là giai đoạn Transform.

- + Giai đoạn Transform: Trong giai đoạn này, một loạt các bước chuyển đổi dữ liệu được áp dụng vào dữ liệu có được từ giai đoạn extract. Một vài dạng chuyển đổi dữ liệu điển hình của giai đoạn này gồm: "lấy một trường dữ liệu cụ thể", "translate giá trị", "tính ra giá trị mới từ giá trị đầu vào", hay "phân tách dữ liệu đầu vào thành các mảnh dữ liệu con".

- + Giai đoạn Load: Ghi dữ liệu sau giai đoạn Transform vào hệ thống đích.

Như vậy, ETL là hỗ trợ đồng bộ cross-schema. Tuy nhiên, ETL là không hoàn thiện cho các chức năng đồng bộ hoá dữ liệu do không hỗ trợ tải nạp dữ liệu dần dần (incremental update), không quy định được rõ ràng cơ chế truyền thông giữa nguồn, đích, và không được tối ưu cho các tính chất bảo mật và thời gian thực của hệ thống.

3. PHƯƠNG PHÁP ĐỀ XUẤT

Một cách ngắn gọn XSync tiếp nhận cách tiếp cận của ETL cho chuyển đổi dữ liệu và cả cách tiếp cận của [4][8][9] cho việc kiến trúc đồng bộ hoá. Ngoài ra XSync chú trọng giải quyết các yêu cầu cốt lõi sau: (1) Change Data Capture: nhận biết và thu gom các thay đổi của database nguồn để hỗ trợ incremental synchronization; (2) ETL-based Data Transformation: cơ chế chuyển đổi tối ưu dựa trên ETL; (3) Bidirectional Synchronization: cơ chế đồng bộ 2 chiều, cho phép các cơ sở dữ liệu đồng bộ lẫn nhau.

a) Change Data Capture (CDC)

Có thể có hai phương án đồng bộ hoá: đồng bộ hoá toàn bộ (complete synchronization) và đồng bộ hoá dần dần (incremental synchronization). Với đồng bộ hoá toàn bộ, toàn bộ nội dung của database nguồn được sao chép sang database đích. Để cập nhật dữ liệu mới được

sửa đổi ở database nguồn sang database đích, toàn bộ quá trình đồng bộ phải được thực hiện lại từ đầu. Với đồng bộ hoá dần dần, chỉ những phần dữ liệu khác biệt ở cơ sở dữ liệu nguồn là được chuyển sang cơ sở dữ liệu đích. Để thực hiện đồng bộ dần dần, các cơ chế có thể được sử dụng bao gồm: sử dụng trigger gắn vào trước các sự kiện sửa đổi ở CSDL nguồn, hoặc đánh nhãn thời gian (timestamping) và các bản ghi ở CSDL nguồn.

Với XSync, cơ chế Change Data Capture sử dụng cơ chế timestamping đánh nhãn thời gian cho các bản ghi. Mỗi bản ghi trong cơ sở dữ liệu sẽ được bổ sung hai trường nhãn thời gian là "updatedAt" và "deletedAt". Hai trường nhãn dữ liệu này sẽ được cập nhật thời gian khi bản ghi tương ứng được tạo ra hay sửa đổi. Với xu hướng phát triển phần mềm hiện nay, hầu hết các ứng dụng cơ sở dữ liệu đều sử dụng các thư viện ORM (Object Relation Mapping) và việc đánh timestamp vào cơ sở dữ liệu là đã được tích hợp sẵn trong ORM hoặc có thể được bổ sung một cách tương đối đơn giản.

Dựa trên các nhãn thời gian, việc lọc bỏ các dữ liệu đã được đồng bộ, giữ lại các dữ liệu mới được sửa đổi là tương đối đơn giản. Tiến trình XSync gắn với cơ sở dữ liệu nguồn sẽ được chạy ngầm và thực hiện đồng bộ định kỳ sau những khoảng thời gian T cố định. Sau mỗi phiên đồng bộ thành công XSync lưu lại nhãn thời gian của phiên đồng bộ đó. Những bản ghi nào trong cơ sở dữ liệu đích có nhãn thời gian lớn hơn nhãn thời gian lưu lại của XSync sẽ được đưa vào phiên đồng bộ tiếp theo.

b) ETL-based Data Transform (EDT)

Cơ chế CDC mô tả ở mục a. cung cấp dữ liệu nguồn cho khâu chuyển đổi dữ liệu ETL-based Data Transform. Ở khối chức năng này, các transformation function được định nghĩa trong file cấu hình synchronization schema. Synchronization schema là một tập hợp cấu trúc sử dụng ký pháp JSON để mô tả: (1) các bảng dữ liệu cần được đồng bộ trong cơ sở dữ liệu nguồn; (2) các ánh xạ (mapping) từ bảng nguồn sang bảng đích, cũng như các ánh xạ từ một cột trong bảng nguồn sang một cột tương ứng trong bảng đích; (3) và các hàm chuyển đổi dữ liệu của từng dữ liệu bên trong các bản ghi. Các hàm chuyển đổi dữ liệu mà XSync hỗ trợ là các hàm khả nghịch. Lý do mà chỉ có các hàm khả nghịch được hỗ trợ là XSync cần hỗ trợ đồng bộ cả hai chiều.

Để hiểu rõ hoạt động của XSync, ta hãy xem xét một ví dụ trong đó bảng "student" ở CSDL nguồn được đồng bộ sang bảng "hoc_sinh" ở CSDL đích. Ngoài ra, các cột "idStudent", "name", "address", và "class" của bảng "student" được ánh xạ tương ứng thành các cột "idHS", "ten", "dia_chi", và "lop" của bảng "hoc_sinh". Đồng thời, các giá trị của cột "name" ở bảng "student" sẽ được chuyển đổi thành chữ hoa ở cột "tên" trong bảng "hoc_sinh". Với trường hợp ví dụ trên, synchronization schema tương ứng được cho ở **Hình 1** dưới. Như có thể thấy trong ví dụ, ký pháp của synchronization schema là trực tiếp và dễ sử dụng.

```

{
  "from_table": "student",
  "to_table": "hoc_sinh",
  "mappings": {
    "idStudent": "idHS",
    "name": "ten",
    "address": "dia_chi",
    "class": "lop"
  },
  "transformation": {
    "forward": {
      "name": "UPPERCASE(%ten%)"
    },
    "backward": {
      "ten": "LOWERCASE(%name%)"
    }
  }
}

```

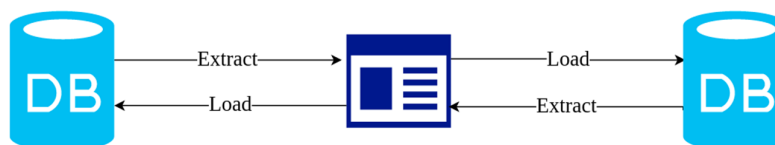
Hình 1. Ví dụ một Synchronization schema trong đó có chuyển đổi tên bảng, tên trường, và dữ liệu trong trường.

Thêm vào đó, để đảm bảo sự tin cậy và chống chịu lỗi của quá trình đồng bộ, việc đồng bộ trong XSync được thực hiện theo hình thức của các giao dịch (database transaction) ở phía CSDL đích. Việc thực hiện đồng bộ bằng giao dịch CSDL cho phép phiên đồng bộ có thể rollback về trạng thái cũ trong trường hợp có lỗi xảy ra.

c) Bidirectional synchronization

Với cơ chế CDC và EDT mô tả ở trên, đồng bộ dữ liệu một chiều (unidirectional synchronization) là khả thi với sự trợ giúp của framework truyền thông (communication frameworks) ví dụ như Webservice, Remote Procedure Call (RPC), hay messaging. Để thực hiện XSync, chúng tôi lựa chọn sử dụng ZeroMQ [10] bởi sự tin cậy, tính gọn nhẹ, và khả năng hỗ trợ nhiều phương thức giao tiếp bao gồm Request/Reply (giống RPC và Webservice), và Push/Pull (giống messaging). Hạn chế duy nhất có lẽ là ZeroMQ là cung cấp các API hỗ trợ ở mức thấp, điều đó khiến công sức và độ phức tạp tăng lên khi sử dụng. Tuy nhiên XSync chỉ cần phát triển một lần và dùng đi dùng lại cho các ứng dụng khác nhau nên tính hiệu quả của XSync cho bài toán đồng bộ dữ liệu là được bảo toàn.

Dựa trên đồng bộ 1 chiều, đồng bộ hai chiều bidirectional synchronization được thực hiện bằng cách kết hợp hai kênh đồng bộ một chiều với nhau như minh hoạ trong **Hình 2**



Hình 2. Bidirectional synchronization bằng việc kết hợp hai kênh đồng bộ unidirectional.

4. THÍ NGHIỆM VÀ ĐÁNH GIÁ HIỆU NĂNG

Để đánh giá tính chính xác của đồng bộ do XSync mang lại cũng như hiệu năng xử lý của XSync, một vài thí nghiệm chạy thử đã được thiết lập trong môi trường phòng thí nghiệm. Trong các thí nghiệm của chúng tôi, các cơ sở dữ liệu nguồn và đích là các CSDL MySQL. Bản phát triển (development version) hiện tại của XSync được triển khai bằng ngôn ngữ Javascript chạy trên nền NodeJS version 8. Các cơ sở dữ liệu mẫu được sử dụng để thực hiện đồng bộ hoá thử nghiệm gồm 3 bảng: bảng "city" chứa trên 4000 bản ghi; bảng "language" chứa 239 bản ghi; và bảng "countrylanguage" chứa 984 bản ghi. Hệ thống thí nghiệm được cài đặt trên máy tính 8GB RAM, bộ xử lý Intel Core i7 2.6GHz với 8 nhân, chạy hệ điều hành Ubuntu 64bit 16.04. Bản MySQL sử dụng trong các thí nghiệm của chúng tôi có phiên bản 5.7.

Đầu tiên các thí nghiệm đánh giá hoạt động của XSync ở chế độ đồng bộ 1 chiều được thực hiện. Trong các thí nghiệm này, một cơ sở dữ liệu được nạp dữ liệu thử nghiệm như mô tả ở trên rồi đồng bộ toàn bộ sang cơ sở dữ liệu đích vốn ban đầu là rỗng. Tiếp đó các bản ghi mới được thêm vào hoặc các bản ghi hiện có được sửa đổi vào CSDL nguồn theo từng batch 100 bản ghi một. Hệ thống XSync sẽ tự động đồng bộ dữ liệu sang cơ sở dữ liệu đích và thông lượng đồng bộ được đo đạc lại. Kết quả thí nghiệm được tóm tắt lại ở **Bảng 1** dưới đây

Bảng 1. Kết quả chạy thử đồng bộ một chiều ở các điều kiện đồng bộ và chuyển đổi khác nhau

Đặc điểm đồng bộ (Sync. characteristics)	Đặc điểm chuyển đổi (Transformation characteristics)	Synchronization Accuracy	Throughput (record/s)
Fresh synchronization	No transformation	Chính xác	1899
	Schema only	Chính xác	1847
	Schema & data	Chính xác	1612
Incremental sync. for new records	No transformations	Chính xác	243
	Schema only	Chính xác	192
	Schema & data	Chính xác	183
Incremental sync. for updating records	No transformations	Chính xác	287
	Schema only	Chính xác	230
	Schema & data	Chính xác	207

Sau khi hai cơ sở dữ liệu nguồn và đích đã có dữ liệu đồng bộ. Các thay đổi được thực hiện đồng thời trên cả hai cơ sở dữ liệu để đánh giá hoạt động của XSync khi đồng bộ dữ liệu hai chiều. Giống như khi thí nghiệm đánh giá đồng bộ 1 chiều, các sửa đổi hoặc thêm mới sẽ được thực hiện theo từng batch 100 bản ghi một lần. Kết quả thu được khi đánh giá hoạt động của XSync khi thực hiện đồng bộ 2 chiều được tóm tắt trong **Bảng 1**

Bảng 2. Kết quả chạy thử đồng bộ một chiều ở các điều kiện đồng bộ và chuyển đổi khác nhau

Đặc điểm đồng bộ (Sync. characteristics)	Đặc điểm chuyển đổi (Transformation characteristics)	Synchronization Accuracy	Throughput (record/s)
Incremental sync. for new records	No transformations	Chính xác	105
	Schema only	Chính xác	109
	Schema & data	Chính xác	88
Incremental sync. for updating records	No transformations	Chính xác	110
	Schema only	Chính xác	101
	Schema & data	Chính xác	89

Quan sát các kết quả thí nghiệm trong bảng 1 và 2 có thể cho thấy XSync là tương đối hứa hẹn để có thể áp dụng và ứng dụng thực tế. Lưu ý là hiện tại XSync mới đang trong giai đoạn phát triển và còn có khá nhiều đoạn mã chưa được tối ưu. XSync sẽ chạy nhanh nhất khi thực hiện đồng bộ dữ liệu mới (Fresh sync.). Khi đó sẽ không có overhead để nhận biết và tổng hợp những cập nhật mới. Với trường hợp đồng bộ dần dần (incremental sync.), thông lượng cập nhật là giảm đi đáng kể (khoảng 7 đến 8 lần) so với khi đồng bộ mới. Nếu so sánh thông lượng đồng bộ giữa các trường hợp khác nhau của đặc điểm chuyển đổi dữ liệu, ta có thể thấy sự khác biệt là tương đối nhỏ so với sự khác nhau giữa đồng bộ mới và đồng bộ dần dần. Như vậy chi phí lớn nhất cho quá trình đồng bộ của XSync nằm ở khâu CDC (Change Data Capture). Điều này theo hiểu biết của chúng tôi là hợp lý và lý giải được. CDC sẽ liên quan chặt chẽ đến hoạt động truy vấn database để lọc được ra các bản ghi cần đồng bộ. Đó là các câu truy vấn SELECT với các mệnh đề WHERE. Trong khi đó, chi phí để thực hiện chuyển đổi dữ liệu chỉ là các thao tác transform trên các bản ghi dữ liệu thu được và sẽ tốn ít tài nguyên tính toán hơn nhiều nếu so với các truy vấn từ CSDL.

5. KẾT LUẬN

Trong báo cáo này chúng tôi đã trình bày một giải pháp đồng bộ mới XSync giúp các ứng dụng hay dịch vụ sử dụng cơ sở dữ liệu có thể dễ dàng đồng bộ với nhau ngay cả khi cấu trúc cơ sở dữ liệu đích và cơ sở dữ liệu nguồn là khác nhau. XSync có thể thực hiện đồng bộ dữ liệu dần dần, từng phần, cũng như cho phép đồng bộ dữ liệu hai chiều. Qua các kết quả thí nghiệm chạy thử, có thể thấy là XSync dù có thể chưa thực tối ưu để có hiệu năng cao nhất nhưng hoàn toàn đáp ứng yêu cầu là một công cụ mạnh, linh hoạt trong rất nhiều tình huống cần sự đồng bộ cơ sở dữ liệu trong hệ thống.

XSync hiện mới chỉ tập trung vào đồng bộ các bảng dữ liệu mà chưa thực hiện việc đồng bộ cho các dữ liệu hỗ trợ như index, trigger, hay store procedure. Việc hỗ trợ đồng bộ các dạng dữ liệu này cũng có thể đem đến nhiều lợi ích trong nhiều tình huống ứng dụng thực tế và sẽ là mục tiêu nghiên cứu tiếp theo của chúng tôi.

Tài liệu tham khảo

- [3]. *Moiz, Salman Abdul and Sailaja, P and Venkataswamy, G and Pal, Supriya N*, Database replication: A survey of open source and commercial tools, Database, Vol. 13, 2011
- [4]. *McCormick, Zach and Schmidt, Douglas C*, Data synchronization patterns in mobile application design, Proceedings of the 19th Conference on Pattern Languages of Programs, 2012
- [5]. *Chaudhuri, Surajit and Dayal, Umeshwar*, An overview of data warehousing and OLAP technology, ACM Sigmod record, Vol. 26, 1997
- [6]. *Malhotra, Naveen and Chaudhary, Anjali*, Implementation of Database Synchronization Technique between Client and Server, International Journal Of Engineering And Computer Science, Vol. 3, 2014
- [7]. *Vassiliadis, Panos*, A survey of extract--transform--load technology, International Journal of Data Warehousing and Mining (IJDWM), Vol. 5, 2009
- [8]. *Cecchet, Emmanuel and Candea, George and Ailamaki, Anastasia*, Middleware-based database replication: the gaps between theory and practice, Proceedings of the 2008 ACM SIGMOD international conference on Management of data, Vol. , 2008
- [9]. *Cecchet, Emmanuel*, C-JDBC: a Middleware Framework for Database Clustering., IEEE Data Eng. Bull., Vol. 27, 2004
- [10]. *Hintjens, Pieter*, ZeroMQ: messaging for many applications, O'Reilly Media, Inc., Vol. , 2013