

PHÁT TRIỂN PHƯƠNG PHÁP ĐẢM BẢO GẮN BÓ DỮ LIỆU TRONG ĐIỀU KIỆN CÓ SỰ CỐ CỦA CÁC HỆ THỐNG THÔNG TIN ĐĂNG KÝ TRÊN MẠNG INTERNET/INTRANET

DEVELOPING DATA CONSISTENT ENSURING METHOD IN FAILURE CONDITION OF REGISTRY INFORMATION SYSTEM IN INTERNET/INTRANET

LÊ VĂN SƠN - NGUYỄN HỮU HẢI
Trường Đại học Sư phạm, Đại học Đà Nẵng

TÓM TẮT

Hiện nay, việc nghiên cứu phát triển các giải pháp kỹ thuật cơ sở cho các ứng dụng phân tán đã đạt được những thành công nhất định và thể hiện trong các công bố mới nhất. Tuy nhiên, để có được một giải pháp hữu hiệu đáp ứng các yêu cầu đặt ra của việc gắn bó dữ liệu trong môi trường phân tán như Internet/Intranet, thì đòi hỏi phải tiếp tục nghiên cứu hoàn thiện các giải pháp hiện hành. Với mục tiêu đó, trong bài viết này, các tác giả trình bày phương pháp thể hiện dưới dạng các bước cơ bản đảm bảo đặc tính gắn bó dữ liệu cho các hệ thống đăng ký trong điều kiện hệ thống xảy ra sự cố.

ABSTRACT

The research and development of basic technical methods for distributed applications have currently achieved specific successes and showed in the latest announcements. However, in order to have an effective method meeting the demands of data consistent in distributed such environments as the Internet/Intranet, further researches for the perfection of existing methods are required. With such an objective, in the research paper, we present a method describing basic steps to ensure the data consistent feature of registry system in failure conditions.

1. Đặt vấn đề

Vấn đề gắn bó dữ liệu trong các hệ thống phân tán nói chung và các hệ thống thông tin đăng ký trên mạng nói riêng như việc đăng ký các tua du lịch, mua bán trong các giao dịch thương mại điện tử, đăng ký giữ chỗ trong giao thông vận tải, đăng ký dự thi trong các hệ thống thông tin đào tạo,... đã trở thành một lĩnh vực thu hút nhiều sự quan tâm của các nhà nghiên cứu lý thuyết lẫn các nhà sản xuất phần mềm nhằm tìm ra các giải pháp hữu hiệu đảm bảo tính gắn bó thông tin trong các cơ sở dữ liệu truy cập ngẫu nhiên với số lượng truy cập lớn [6,5,4,1].

Việc nghiên cứu, phát triển các phương pháp đảm bảo gắn bó dữ liệu trong môi trường phân tán đến nay đã có một số thành công đáng kể. Trước hết, phải nói đến sự tích hợp hàng loạt các hàm vào các hệ điều hành, hệ quản trị cơ sở dữ liệu, ngôn ngữ lập trình với phương thức riêng rẽ của từng hệ cho phép đảm bảo phần cơ bản của việc gắn bó thông tin nói chung trong quá trình đa truy cập thông qua hệ thống viễn thông. Việc triển khai phương pháp cho phép gắn bó dữ liệu trong môi trường phân tán trong điều kiện lý tưởng (không có bất kỳ sự cố nào) đã được áp dụng và mang lại những hiệu quả nhất định cho các hệ thống đăng ký [4,5,1]. Tuy nhiên, trong quá trình triển khai lập trình và vận hành các hệ thống đăng ký, vấn đề gắn bó dữ liệu trong các cơ sở dữ liệu khi hệ thống bị sự cố đang đặt ra như là vấn đề quan trọng và có ý nghĩa sống còn trong việc phát triển toàn hệ nói chung. Nói tóm lại, sự cố trong các hệ thống nói chung, hệ thống đăng ký nói riêng có thể xảy ra và là nguyên nhân dẫn đến hệ thống cơ sở dữ liệu không thể đảm bảo tính gắn bó được nữa.

Trong phạm vi của bài viết này, chúng tôi giới thiệu một giải pháp được phát triển dựa trên giải thuật hai pha tuyến tính [4] và giải thuật MAONT [1] kết hợp với mô hình xử lý phân tán JavaRMI [9,8,2] và các kỹ thuật chuyên tác trong SQL Server 2000 [3] nhằm đảm bảo tính gắn bó dữ liệu phân tán trong điều kiện có sự cố phục vụ cho các hệ thống thông tin đăng ký.

2. Tác động của sự cố đối với việc gắn bó dữ liệu

Nhằm nghiên cứu và phát triển phương pháp đảm bảo gắn bó dữ liệu một cách khả thi trong điều kiện có sự cố cho các hệ thống thông tin đăng ký nói riêng, các hệ thống tin học phân tán nói chung, bản chất của vấn đề gắn bó dữ liệu cũng như nguyên nhân dẫn đến sự cố làm cho dữ liệu không còn gắn bó được chỉ ra và nhóm ghép trong quá trình phân tích, thiết kế hệ thống.

Một cơ sở dữ liệu nào đó được gọi là gắn bó, nếu nó thỏa mãn một tập các ràng buộc về toàn vẹn ngữ nghĩa. Để đảm bảo tính gắn bó dữ liệu nhiều cơ chế khác nhau như điều khiển hoạt động đồng thời [5], kiểm soát tính toàn vẹn ngữ nghĩa, ... được sử dụng [4].

Việc kiểm soát tính toàn vẹn ngữ nghĩa tốt sẽ đảm bảo được tính gắn bó dữ liệu của hệ thống thông tin. Hiện nay, người ta đang áp dụng hai phương pháp chủ yếu :

1. Loại bỏ các chương trình/thủ tục cập nhật có thể dẫn đến trạng thái không gắn bó dữ liệu trong các cơ sở dữ liệu [CKP₁]

2. Triệu gọi các chương trình/thủ tục đặc biệt đã được cài đặt trên hệ thống nhằm khôi phục trạng thái ban đầu trước khi cập nhật [CKP₂].

Các ràng buộc toàn vẹn được phân làm hai loại chủ yếu :

1. Ràng buộc cấu trúc (Structural Constraint) diễn tả những đặc tính ngữ nghĩa cơ bản vốn có trong mô hình. Ví dụ như ràng buộc thể hiện bằng khóa duy nhất trong mô hình quan hệ hoặc các liên kết theo kiểu $1 - n$, ($n > 1$) giữa các đối tượng trong mô hình mạng

2. Ràng buộc hành vi (Behavioral Constraint) nhằm điều hòa các hoạt động của các ứng dụng.

Trong quá trình nghiên cứu bài toán, các tác giả vận dụng phương pháp tổng quát đảm bảo gắn bó dữ liệu và các ràng buộc toàn vẹn thông tin phục vụ công tác phân tích, thiết kế hệ quản trị cơ sở dữ liệu cho hệ thống cơ sở dữ liệu đăng ký với thông tin gắn bó trong điều kiện phân tán. Những thông tin chi tiết có thể tham khảo trong tài liệu [7].

Để có thể khôi phục lại dữ liệu và trạng thái gắn bó thông tin của toàn hệ thống khi có sự cố diễn ra, một trong những vấn đề quan trọng hàng đầu là cần phải xác định được loại, bản chất và vị trí diễn ra sự cố mà từ đó nhận biết một cách tự động và chuyển đến phương án giải quyết nào cho phù hợp. Bản chất (mã) sự cố và thủ tục giải quyết sự cố có liên quan đến gắn bó được lưu trữ tại trung tâm hoặc phân tán trên các trạm của hệ đăng ký phân tán. Trong trường hợp các thủ tục được phân tán, thì bản thân các thủ tục đó cũng đòi hỏi phải có giải pháp đảm bảo gắn bó giống như dữ liệu bình thường [5].

Về tổng quan, các sự cố được chia làm hai loại chính là sự cố xảy ra do các trạm trên hệ thống phân tán và sự cố do hệ thống viễn thông gây ra [4,5]. Song để tiện lập trình giải pháp, người ta cụ thể hoá ra thành bốn mức với các nguyên nhân tương ứng thể hiện trong bảng 1.

Bảng 1. Các loại sự cố và nguyên nhân gây ra sự cố

TT	Loại sự cố	Tiếng Anh	Nguyên nhân
1	Sự cố giao dịch	Transaction Failure	Do một lỗi nào đó trong bản thân giao dịch gây nên. Ví dụ như dữ liệu nhập không đúng hoặc do phát hiện ra một khóa gài tiềm tàng hoặc hiện hữu
2	Sự cố vị trí	Site Failure	Do một lỗi phát sinh trong quá trình vận hành hệ thống. Lỗi này có thể bắt nguồn từ phần thiết bị như bộ xử lý/bộ vi xử lý, bộ nhớ, các thiết bị ngoại vi,... bị sự cố. Khi bị sự cố, hệ thống lập tức bị ngừng hoạt động. Hệ thống chương trình, đặc biệt là các chương trình điều khiển cũng có thể sinh lỗi. Đó là các lỗi do thuật toán, do lệnh viết sai, do phần lưu trữ chương trình hay do virus. Các lỗi này thường là ở các chương trình và cơ sở dữ liệu
3	Sự cố phương tiện	Media Failure	Do sự cố của các thiết bị lưu trữ thứ cấp dùng để lưu cơ sở dữ liệu. Khi có sự cố này thì một phần hoặc tất cả cơ sở dữ liệu trên thiết bị đó được xem như bị hủy hoại hoặc không thể truy cập một cách bình thường được
4	Sự cố đường truyền	Transmission Failure	Do lỗi trong các thông điệp, các thông điệp vô trật tự, thông điệp bị thất lạc hoặc không phân phối thông điệp và sự cố khác liên quan đến đường truyền.

Phương pháp tổng quát cho việc khắc phục bốn loại sự cố này được trình bày trong bảng 2 [4].

Bảng 2. Phương pháp tổng quát khắc phục sự cố

TT	Loại sự cố	Cách khắc phục
1	Sự cố giao dịch	[CKP ₁] + [CKP ₂] Hủy bỏ giao dịch, sau đó đặt lại cơ sở dữ liệu về trạng thái của nó trước khi khởi động giao dịch này
2	Sự cố vị trí	[CKP ₂] Thiết kế nghi thức ủy thác nguyên tử không bị phong tỏa
3	Sự cố phương tiện	[CKP ₀] Được xem như những vấn đề cục bộ của một vị trí, vì vậy không xem xét trong trường hợp các hệ thống phân tán
4	Sự cố đường truyền	[CKP ₂] Sử dụng bộ đếm thời gian và cơ chế quá hạn để theo dõi xem đã qua bao lâu kể từ khi vị trí gửi không nhận được thông điệp trả lời của vị trí đến.

3. Phát triển giải pháp kỹ thuật đảm bảo gắn bó dữ liệu

Để dễ dàng mô tả các giải thuật đảm bảo gắn bó dữ liệu phân tán, chúng ta giả thiết rằng, tại vị trí nguồn của giao dịch một tiến trình thực hiện các thao tác của nó, tiến trình này

được gọi là điều phối viên (Coordinator). Điều phối viên trao đổi với các thành viên (Participant) tại những vị trí có tham gia vào việc thực hiện các thao tác của giao dịch.

Cải tiến giải thuật hai pha tuyến tính (Linear 2PC) [4], giải thuật MAONT [1], ta thiết kế giải thuật mà trong đó các thành viên có thể trao đổi với nhau.

Có một thứ tự giữa các vị trí trong hệ thống dành cho việc giao tiếp. Chúng ta hãy giả thiết rằng thứ tự giữa các vị trí có tham gia vào việc thực hiện một giao dịch là $1, 2, \dots, N$ với điều phối viên là vị trí đầu tiên trong thứ tự này (Giải thuật được minh họa bằng hình 1).

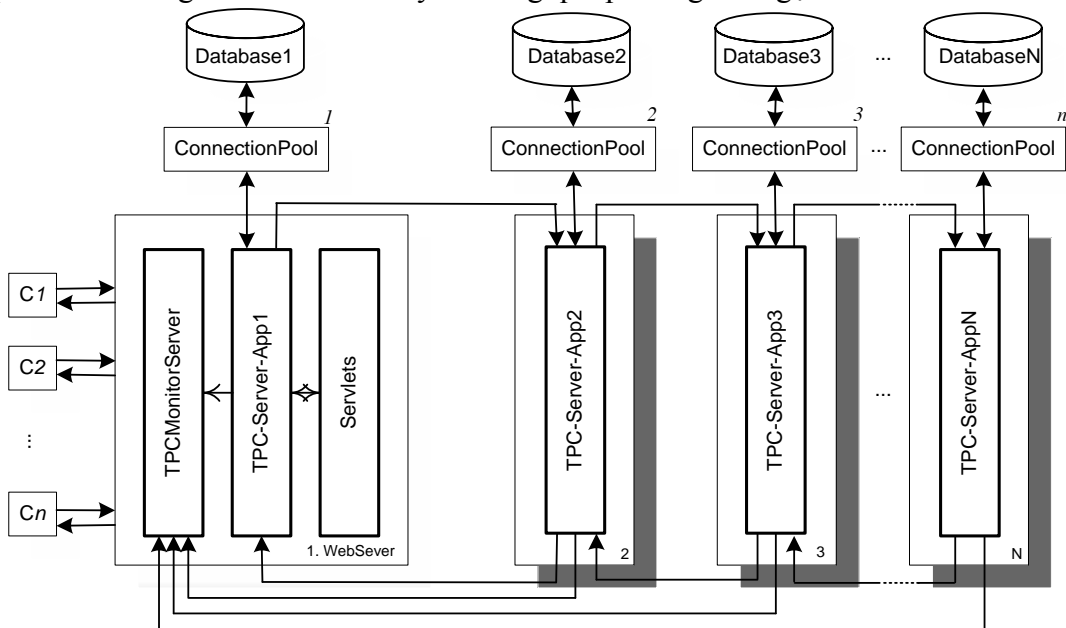
Theo mô hình trong hình vẽ, ta có các đối tượng sau :

1. C_1, C_2, \dots, C_n là các Client truy cập Web Server bằng trình duyệt Web
2. Servlets là các đối tượng xử lý yêu cầu được gửi từ các $C_i, i=1, n$
3. TPC-Server-App1, TPC-Server-App2, ..., TPC-Server-AppN là các RMI Server cài đặt thuật toán 2PC tuyến tính (Linear Two Phase Commit - TPC)
4. TPCMonitorServer là một trình giám sát cho phép hiển thị quá trình dịch chuyển của danh sách di chuyển [6] trong quá trình xử lý của các TPC-Server-Appi, $i = 1, N$
5. ConnectionPool là chương trình điều khiển các liên kết cơ sở dữ liệu dùng chung
6. Database1, Database2, ..., DatabaseN là các cơ sở dữ liệu quan hệ phân tán trên mạng.

Mô tả hoạt động :

Các C_1, C_2, \dots, C_n là các Client truy cập vào Web Server bằng trình duyệt Web. Các yêu cầu này đòi hỏi việc xử lý phải truy vấn đến các cơ sở dữ liệu phân tán. Do đó việc xử lý thành công hay không phụ thuộc vào kết quả của tất cả các truy vấn này. Điều đó có nghĩa là, xử lý sẽ thành công nếu tất cả các truy vấn đều thành công, ngược lại, xử lý sẽ không thành công nếu có bất kỳ một truy vấn nào bị lỗi hoặc không thực hiện được.

Khi nhận được một yêu cầu từ các $C_i, i=1, n$, các Servlet sẽ dựa vào yêu cầu này để thành lập nên một danh sách các Database Server sẽ truy vấn và các câu lệnh SQL tương ứng được thực hiện tại mỗi Server. Việc thành lập danh sách các câu lệnh SQL truy vấn tại mỗi Database Server được dựa theo thứ tự ưu tiên SELECT, DELETE, UPDATE, INSERT. Sau khi thành lập danh sách di chuyển, Servlet chuyển danh sách di chuyển này cho TPC-Server-App đầu tiên trong danh sách di chuyển thông qua phương thức gọi từ xa RMI.



Hình 1. Sơ đồ hoạt động của giải pháp kỹ thuật đảm bảo gắn bó dữ liệu

Mỗi TPC-Server-App cài đặt một giao diện cung cấp phương thức xử lý theo thuật toán 2PC tuyến tính với tham số là danh sách di chuyển, chỉ mục hiện tại của danh sách và trả về kết quả là danh sách kết quả truy vấn của chính nó và các Server đứng phía sau nó trong danh sách di chuyển.

```
public interface TPCApp extends Remote  
{  
    public ResultQueryList queryApp(MovableList movableList,int index) throws  
RemoteException;  
}
```

Nếu quá trình xử lý tại bất kỳ một TPC-Server-App nào bị lỗi thì kết quả trả về là null. Dựa vào kết quả trả về này, các TPC-Server-App commit hoặc rollback transaction đang quản lý. Như vậy, khi TPC-Server-App đầu tiên nhận được danh sách di chuyển, TPC-Server-App bắt đầu một transaction để thực hiện các câu lệnh SQL truy vấn CSDL cục bộ thông qua kết nối CSDL được lấy từ ConnectionPool cục bộ. Tiếp theo, TPC-Server-App tăng chỉ mục hiện tại của danh sách di chuyển lên 1 và chuyển danh sách di chuyển này đến TPC-Server-App kế tiếp. Quá trình này được lặp lại cho đến khi kết thúc danh sách di chuyển. Tại Server cuối cùng trong danh sách di chuyển, nếu việc truy vấn CSDL cục bộ thành công, TPC-Server-App commit transaction và trả về kết quả là ResultQueryList khác null. Dựa vào kết quả trả về này, Server đứng trước trong danh sách di chuyển sẽ commit hoặc rollback transaction cục bộ và trả về kết quả cho Server liền trước. Khi Servlet nhận được kết quả là null có nghĩa là xử lý không thành công. Ngược lại, Servlet sẽ tiếp tục xử lý kết quả nhận được để trả về cho Client.

Các bước cơ bản của giải thuật đề xuất nhằm minh họa cho việc cải tiến giải thuật MAONT và giải thuật hai pha tuyến tính được mô tả qua hình 2. Việc triển khai xây dựng chương trình bằng ngôn ngữ lập trình Java cùng với thư viện RMI và tiến hành thực nghiệm sau đó đã đạt được kết quả nhất định.

Riêng những kết quả thực nghiệm sẽ được trình bày trong các phần tiếp theo của đề tài nghiên cứu.

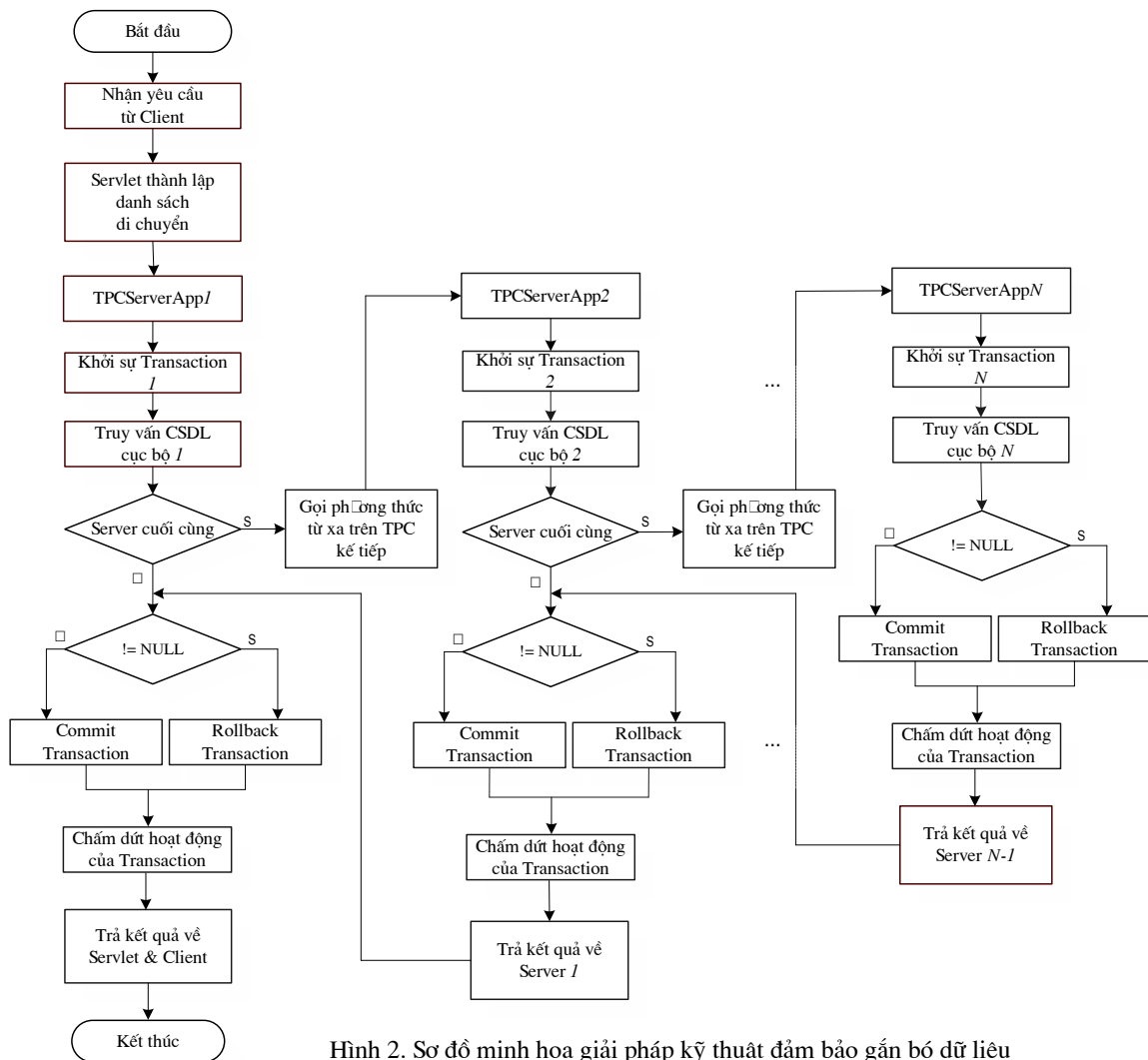
4. Kết luận

Đề tài phát triển phương pháp đảm bảo gắn bó dữ liệu trong điều kiện có sự cố của các hệ thống thông tin đăng ký trên mạng Internet/Intranet đã đi sâu nghiên cứu và đạt được những kết quả sau đây :

1. Phân tích bài toán đảm bảo gắn bó dữ liệu trên các hệ thống phân tán, làm cơ sở cho lớp các bài toán đăng ký thông tin từ xa trên mạng Internet/Intranet

2. Đưa ra giải pháp mới đảm bảo tính gắn bó dữ liệu trong điều kiện có sự cố của hệ thống thông tin phân tán. Đó là giải pháp 2PC danh sách di chuyển. Ưu điểm cơ bản của giải pháp là không chỉ cho phép xử lý dữ liệu phân tán mà còn cập nhật các chương trình trong môi trường phân tán, chia sẻ công việc cho nhiều Server. 2PC danh sách di chuyển còn có thể đáp ứng yêu cầu của cấu hình động khi tăng hoặc giảm số lượng Server trong hệ thống.

3. Tiến hành thực nghiệm giải pháp 2PC danh sách di chuyển bằng ngôn ngữ lập trình Java với hệ quản trị cơ sở dữ liệu SQL Server 2000 thông qua bài toán đăng ký các tour du lịch từ xa.



Hình 2. Sơ đồ minh họa giải pháp kỹ thuật đảm bảo gắn bó dữ liệu

TÀI LIỆU THAM KHẢO

- [1] Phạm Đình Hân, Lê Văn Sơn, *Phát triển giải pháp kỹ thuật đảm bảo gắn bó dữ liệu cho các hệ thống thông tin đào tạo trên mạng Internet/Intranet*, Tạp chí Khoa học Công nghệ, số 46-47(2004).
- [2] Nguyễn Phương Lan, Hoàng Đức Hải, *Java Lập trình mạng*, NXB Giáo dục, 2001.
- [3] Phạm Hữu Khang, *Lập trình ứng dụng chuyên nghiệp SQL Server 2000*, Tập I và II, NXB Lao động - Xã hội, 2002.
- [4] M. Tamer Ozsu, Patrick Valduriez, Trần Đức Quang biên dịch, *Nguyên lý các hệ cơ sở dữ liệu phân tán*, Tập I và II, NXB Thống Kê, 1999.
- [5] Lê Văn Sơn, *Hệ tin học phân tán*, NXB ĐH Quốc gia TP Hồ Chí Minh, 2002.
- [6] Lê Văn Sơn, Đặng Hùng Vĩ, *Nghiên cứu phát triển hệ thống thông tin dạy và học trên mạng Internet/Intranet*, Tạp chí Khoa học & Công nghệ, Đại học Đà Nẵng, số 4.2003.
- [7] E. Simon and P. Valduriez, *Integrity Control in Distributed Database Systems*, In Proc. 19th Hawaii Int. Conf. on System Sciences, January 1986.
- [8] <http://www.javasoft.com>.
- [9] <http://java.sun.com/products/jdk/rmi>.