

XÂY DỰNG CHƯƠNG TRÌNH THỬ NGHIỆM 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

CONSTRUCTING A TESTING PROGRAM FOR THE METHOD OF ASSURING DATA COHERENCE IN FAILURE CONDITION OF THE REGISTRY INFORMATION SYSTEM ON THE INTERNET/INTRANET

NGUYỄN HỮU HẢI

Trường TH Kinh tế Kỹ thuật Phương Đông

TÓM TẮT

Trong một bài viết trước, các tác giả đã hoàn thành việc 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. Để chứng minh tính đúng đắn của phương pháp đã trình bày, trong bài viết này, các tác giả tiếp tục xây dựng chương trình ứng dụng thử nghiệm. Chương trình là một hệ thống ứng dụng Web được viết bằng Java, các ứng dụng chạy trên các Server được cài đặt bằng công nghệ Java RMI và cơ sở dữ liệu được tổ chức trên SQL Server đảm bảo được tính gắn bó dữ liệu trong điều kiện có sự cố của hệ thống.

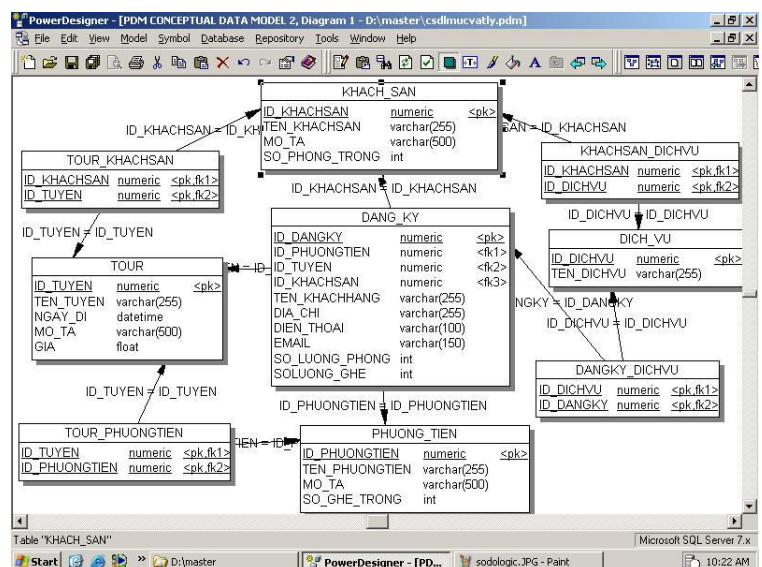
ABSTRACT

In the previous research paper, the researchers completed the development of the method of assuring data coherence in failure condition of the registry information system on the Internet/Intranet. In order to prove the correctness of the presented method, in this research paper, the researchers continue constructing the testing application. This program is a web-applied system written by Java. The applications running on servers are installed by Java RMI technology and the database is organized on the SQL Server assuring data coherence in failure condition of the system.

1. Đặt vấn đề

Với mục đích là chương trình ứng dụng thử nghiệm 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 nên bài toán được xét đến trong bài viết là bài toán đăng ký tua du lịch. Cụ thể, chúng ta xây dựng một hệ thống ứng dụng Web cho phép khách hàng đăng ký tour du lịch trên mạng Internet trong các điều kiện sau:

- Hệ thống ứng dụng bao gồm ba cơ sở dữ liệu:



Hình 1. Mô hình cơ sở dữ liệu mức vật lý

tuyến du lịch, khách sạn, phương tiện và ba cơ sở dữ liệu này được đặt tại ba server khác nhau trên mạng.

- Hệ thống đăng ký phải đảm bảo tính gắn bó dữ liệu trong từng cơ sở dữ liệu và giữa các cơ sở dữ liệu trong hệ thống.
- Trong quá trình đăng ký, các sự cố có thể xảy ra.

2. Xây dựng mô hình hệ thống

2.1 Xây dựng mô hình cơ sở dữ liệu

Dùng công cụ Power Designer để thiết kế mô hình cơ sở dữ liệu mức vật lý được thể hiện ở hình 1.

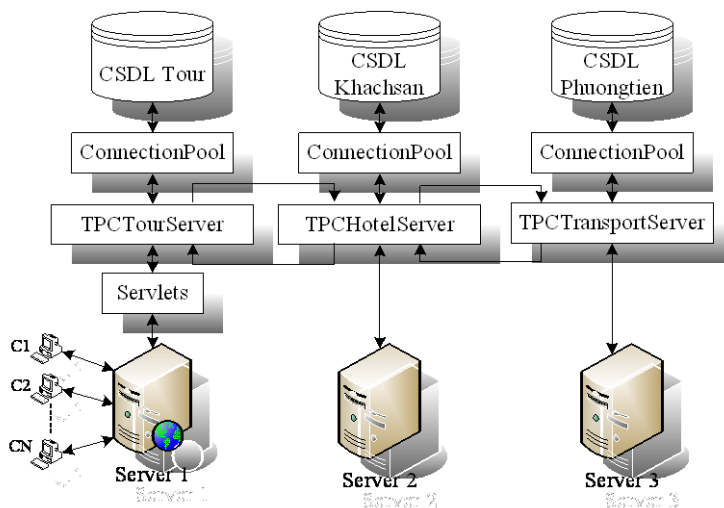
2.2 Xây dựng mô hình ứng dụng

Từ mô hình cơ sở dữ liệu mức vật lý, chúng ta phân thành ba cơ sở dữ liệu để cài đặt trên ba database SQL Server chạy trên ba server khác nhau [3, 4, 6]. Hệ thống ứng dụng được xây dựng thỏa các yêu cầu trong bảng 1.

Bảng 1. Các yêu cầu của hệ thống ứng dụng

TT	Yêu cầu
1	Thông tin đăng ký tại các database server phải nhất quán.
2	Việc truy xuất các cơ sở dữ liệu được thực hiện bởi ứng dụng đang chạy trên cùng server với cơ sở dữ liệu.
3	Thao tác đăng ký thành công nếu các thao tác truy vấn cơ sở dữ liệu tại các server đều thành công và ngược lại.
4	Một yêu cầu đăng ký được xử lý phân tán tại các server đang chạy database server.
5	Cho phép nhiều người dùng đăng ký tại một thời điểm [1, 5].

Mô hình triển khai của ứng dụng được mô tả như hình 2.



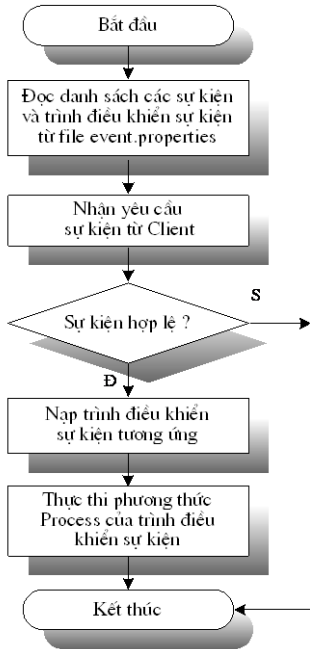
Hình 2. Mô hình triển khai của ứng dụng

Ứng dụng web được cài đặt bằng Java Servlet và JSP. Tất cả các yêu cầu của client được xử lý bởi servlet Controller. Tùy theo yêu cầu của người dùng, Controller sẽ nạp trình điều khiển sự kiện tương ứng để xử lý yêu cầu của người dùng. Các sự kiện và trình điều khiển sự kiện được cấu hình trong file EVENT.properties, cấu trúc của nó là một dãy các cặp tên sự kiện, tên lớp điều khiển sự kiện.

REGISTER=RegisterEventHandler
 RS_REGISTER=QueryRegisterEventHandler

Với dữ liệu này, khi người dùng gửi yêu cầu có tham số event là REGISTER thì lớp RegisterEventHandler sẽ xử lý yêu cầu này [2, 7]

Mô hình hoạt động của Controller được minh họa theo sơ đồ hình 3.

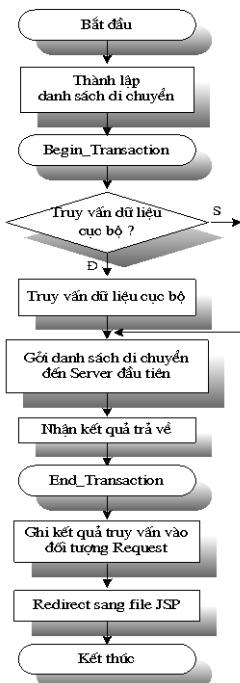


Hình 3. Mô hình hoạt động của Controller

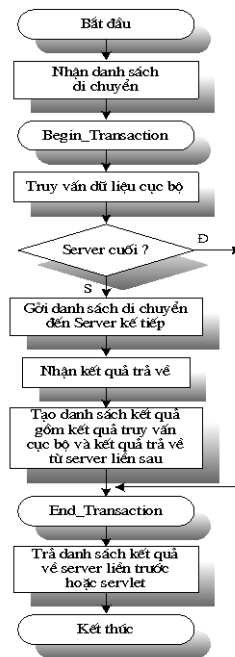
Các nhiệm vụ chính của trình điều khiển sự kiện:

- Xác định danh sách các server tham gia vào truy vấn dữ liệu.
- Thành lập danh sách các câu lệnh truy vấn tại mỗi server.
- Tạo danh sách di chuyển, mỗi mục trong danh sách di chuyển bao gồm địa chỉ server tham gia truy vấn và danh sách các câu lệnh truy vấn tại server.
- Mở một giao dịch (transaction) mới, truy vấn dữ liệu tại server cục bộ nếu có.
- Chuyển danh sách di chuyển cho server đầu tiên trong danh sách thông qua lời gọi phương thức từ xa RMI.
- Nhận kết quả trả về, kết thúc giao dịch.
- Ghi kết quả xử lý vào đối tượng request, redirect sang trang JSP để hiển thị kết quả trong đối tượng request.

Sơ đồ thuật toán cho một servlet điều khiển sự kiện được mô tả qua hình 4.



Hình 4. Sơ đồ thuật toán một Servlet điều khiển sự kiện



Hình 5. Sơ đồ thuật toán TPCServerApp

Nhiệm vụ của các TPCServerApp chạy tại các các server là truy vấn cơ sở dữ liệu cục bộ, gửi danh sách di chuyển đến server kế tiếp trong danh sách (nếu nó chưa là server cuối cùng trong danh sách di chuyển) và trả kết quả truy vấn về server liền trước hoặc servlet (nếu nó là server đầu tiên trong danh sách di chuyển). Khi TPCServerAPP nhận được danh sách di chuyển, chúng tiến hành thực hiện các thao tác theo sơ đồ được minh họa bằng hình 5.

3. Triển khai xây dựng hệ thống

Trong phần này, chúng ta sẽ tiến hành cài đặt mô hình ứng dụng bằng ngôn ngữ lập trình Java.

3.1 Controller

Controller là trình điều khiển sự kiện. Controller tiếp nhận yêu cầu của client, nạp trình điều khiển sự kiện tương ứng và gọi phương thức process của trình điều khiển sự kiện. Các phương thức chính của Controller được mô tả qua bảng 2.

Bảng 2. Các phương thức chính của Controller

Tên phương thức	Mô tả
init()	Khởi tạo các thông tin ban đầu bằng cách đọc file EVENT.properties và tạo thể hiện cho các trình điều khiển sự kiện.
public void doPost(HttpServletRequest request, HttpServletResponse response)	Xử lý các yêu cầu sử dụng phương thức POST
public void doGet(HttpServletRequest request, HttpServletResponse response)	Xử lý các yêu cầu sử dụng phương thức GET

3.2 Các trình điều khiển sự kiện

Tất cả các lớp điều khiển sự kiện đều kế thừa từ lớp trừu tượng EventHandlerBase, lớp EventHandlerBase định nghĩa các phương thức được mô tả qua bảng 3.

Bảng 3. Các phương thức của lớp EventHandlerBase

Tên phương thức	Mô tả
public void process(ServletContext sc, HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException	Đây là phương thức được Controller gọi bên trong hai phương thức doPost() và doGet() để xử lý yêu cầu của người dùng. Các lớp kế thừa từ lớp này phải cài đặt phương thức này để xử lý các yêu cầu cụ thể của người dùng.
protected abstract String getURL()	Phương thức này trả về đường dẫn đến file jsp sẽ được Controller redirect đến sau khi thực hiện xong phương thức process của các trình điều khiển sự kiện.

3.3 Danh sách di chuyển

Danh sách di chuyển được cài đặt bởi lớp MovableList. Mỗi phần tử trong danh sách di chuyển là một đối tượng MovableItem, mỗi MovableItem chứa danh sách các câu lệnh truy vấn sẽ được thực hiện tại một server. Các thuộc tính và phương thức của lớp này được mô tả qua bảng 4.

Bảng 4. Các thuộc tính và phương thức của lớp MovableList

Tên phương thức	Mô tả
Private String addrHttpClient	Là thuộc tính chứa địa chỉ IP của client gửi yêu cầu.
Private Vector movableItem	Là biến chứa danh sách các MovableItem của mỗi server.
public MovableList(String addr)	Constructor tạo danh sách di chuyển cho yêu cầu được gửi từ client
public int getSize()	Phương thức này trả về kích thước của danh sách di chuyển
public String getAddressHTTIClient()	Trả về địa chỉ IP của client gửi yêu cầu.
public Vector getMovableItems()	Trả về danh sách các phần tử của danh sách di chuyển.
public void addMovableItem(MovableItem item)	Thêm một phần tử vào cuối danh sách di chuyển.
public MovableItem getMovableItemAt(int index)	Trả về phần tử có chỉ mục index trong danh sách di chuyển.

3.4 Danh sách kết quả

Sau khi nhận được kết quả truy vấn từ các server đứng sau trong danh sách di chuyển, TPCServerApp sẽ tạo ra một danh sách di chuyển mới bao gồm kết quả truy vấn cục bộ và danh sách kết quả truy vấn nhận được từ server liền sau. Danh sách di chuyển được cài đặt bởi lớp ResultQueryList. Các thuộc tính và phương thức của lớp này được mô tả trong bảng 5.

Bảng 5. Các thuộc tính và phương thức của lớp ResultQueryList

Tên phương thức	Mô tả
private Vector resultList	Chứa danh sách các kết quả đối tượng ResultQueryItem. Mỗi đối tượng ResultQueryItem chứa danh sách kết quả truy vấn tại một server.
public int getSize()	Trả về chiều dài của danh sách kết quả.
public Vector getResultQueryList()	Trả về vector chứa danh sách kết quả.
public void append(ResultQueryList rqlRemote)	Thêm danh sách kết quả vào danh sách kết quả hiện tại.
public void addResultQueryItem(ResultQueryItem rqi)	Thêm một đối tượng chứa kết quả truy vấn tại một server vào danh sách kết quả.
public ResultQueryItem getResultQueryItemAt(int index)	Lấy kết quả truy vấn tại server có chỉ mục index trong danh sách kết quả.

3.5 TPCServerApp

Các TPCServerApp được cài đặt theo công nghệ Java RMI. Mỗi TPCServerAPP bao gồm một giao diện, một lớp cài đặt giao diện và một lớp ứng dụng. Lớp giao diện định nghĩa các phương thức có thể được gọi từ các đối tượng trên một máy ảo Java khác. Lớp cài đặt các phương thức được định nghĩa trong giao diện. Lớp ứng dụng sẽ đăng ký lớp RMI với rmiregistry. Giao diện của các TPCTourServer, TPCHotelServer, TPCTransportServer định nghĩa một phương thức cài đặt thuật toán 2PC tuyến tính nhận tham số là danh sách di chuyển

và chỉ mục của server trong danh sách di chuyển [7].

```

public interface TPCTour extends Remote
{
public ResultQueryList queryTour(MovableList movableList,int index) throws RemoteException;
}
public interface TPCHotel extends Remote
{
public ResultQueryList queryHotel(MovableList movableList,int index) throws RemoteException;
}
public interface TPCTransport extends Remote
{
public ResultQueryList queryTransport(MovableList movableList,int index) throws
RemoteException;
}

```

3.6 ConnectionPool

ConnectionPool là thùng chứa các kết nối cơ sở dữ liệu sẵn dùng. Các phương thức của đối tượng được mô tả trong bảng 6.

Bảng 6. Các phương thức của ConnectionPool

Tên phương thức	Mô tả
public ConnectionPool(String strProperties,int initialConnections,int maxConnections,boolean waitIfBusy)	Khởi tạo thùng chứa các thông tin sau: - strProperties: file chứa thông tin để kết nối cơ sở dữ liệu - initialConnections: Số kết nối tạo sẵn lần đầu tiên. - maxConnections: Số kết nối tối đa - waitIfBusy: Đợi hay không đợi khi tất cả các kết nối đều bận
public synchronized void free(Connection connection)	Giải phóng một kết nối cơ sở dữ liệu.
public synchronized int totalConnections()	Trả về tổng số kết nối sẵn dùng hiện tại.
public synchronized void closeAllConnection()	Đóng tất cả các kết nối.
private Connection makeNewConnection() throws SQLException	Tạo một kết nối cơ sở dữ liệu mới.
public synchronized Connection getConnection()	Nhận một kết nối cơ sở dữ liệu sẵn dùng từ thùng chứa.

4. Kết quả thực hiện

4.1 Môi trường thử nghiệm chương trình

Chương trình được triển khai thử nghiệm trên mạng cục bộ. Cấu hình của các máy trạm như sau:

- Phần cứng: . Tốc độ vi xử lý từ 450 đến 733 MHz
. Bộ nhớ RAM từ 64 đến 128 MB.
- Phần mềm: . Hệ điều hành Windows 2000, SQL Server 2000.
. Công cụ lập trình Java JDK 1.4

4.2 Dữ liệu thử nghiệm

Thực hiện bài toán đăng ký các Tour du lịch của du khách gồm Tour du lịch, khách sạn, phương tiện đi lại và dịch vụ với mỗi Server lưu trữ một Database dùng SQL Server 2000 như sau:

- Server 1: Lưu trữ database về Tour, file database là Tour gồm hai bảng Tour, DangKy, DangKy_DichVu
- Server 2: Lưu trữ database về Hotel, file database là KháchSan gồm hai bảng Khách_San, Tour_KháchSan, Dich_Vu, DichVu_KháchSan
- Server 3: Lưu trữ database về Transport, file database là PhươngTien gồm hai bảng Phương_Tien, Tour_PhuongTien

4.3 Các bước chạy và nhập liệu

- **Bước 1: Khởi động các TPCServerApp tại các Server**

✎ Tại Server Tour: Giả sử là Server có chứa ứng dụng web, được xem như là điều phối viên, quản lý việc đăng ký các tua du lịch cho du khách, ta thực hiện như sau:

```
start rmiregistry
```

```
start java TPCTourServer
```

```
java TPCMonitorServer
```

✎ Tại Server Hotel: Là Server chứa cơ sở dữ liệu về khách sạn phục vụ cho du khách trong các tour du lịch, được xem như là các thành viên, ta thực hiện

```
start java TPCHotelServer
```

✎ Tại Server Transport: Là Server chứa cơ sở dữ liệu về phương tiện đi lại phục vụ cho du khách trong các tour du lịch, cũng được xem như là các thành viên, ta thực hiện

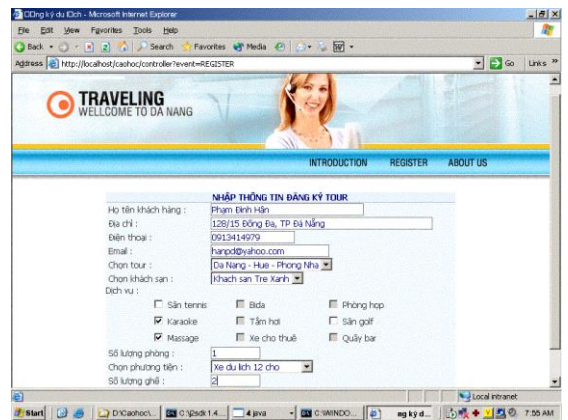
```
start java TPCTransportServer
```

- **Bước 2: Khởi động ứng dụng web tại điều phối viên**

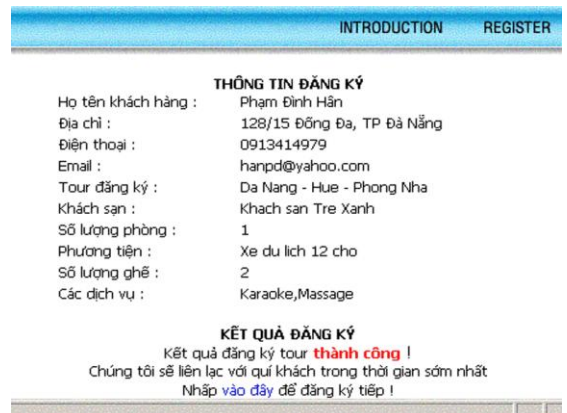
Tại Server tour ta khởi động trình duyệt Internet Explorer, gõ địa chỉ của site ứng dụng. Sau đó ta kích vào nút Register, ứng dụng sẽ đưa ta vào trang đăng ký như hình 6. Tại trang này ta thực hiện nhập, chọn các thông tin cần thiết. Sau đó ấn vào nút lệnh **Đăng ký** để thực hiện việc đăng ký. Kết quả đăng ký thành công sẽ có giao diện như hình 7.

- **Bước 3: Tạo sự cố để kiểm tra tính đảm bảo gắn bó dữ liệu**

Ta cũng thực hiện đăng ký như bước 1, nhưng trước khi ấn nút **Đăng ký**, ta tắt một Server, ví dụ Server Hotel. Điều hoàn toàn tương tự việc tắt ứng dụng TPCServerHotel. Sau đó ta ấn nút **Đăng ký** để thực hiện đăng ký, lúc này được xem như là có sự cố hệ thống, hệ thống sẽ thông báo việc đăng ký không thành công trên giao diện



Hình 6. Trang đăng ký của chương trình



Hình 7. Trang kết quả đăng ký thành công



Hình 8. Trang kết quả đăng ký không thành công do sự cố

như hình 8.

Sau đó ta khởi động lại Server Hotel. Điều này hoàn toàn tương tự việc khởi động lại ứng dụng TPCServerHotel. Ta thực hiện lại việc đăng ký với những thông tin giống hệt lần đăng ký không thành công, lúc này ta thấy dữ liệu trên các Server được rollback lại. Điều này khẳng định là dữ liệu đã được đảm bảo gắn bó trong điều kiện có sự cố của hệ thống.

5. Kết luận

Bài báo xây dựng chương trình thử nghiệm 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 được xây dựng chi tiết và đạt được những kết quả sau đây:

1. Xây dựng một ứng dụng Web cụ thể đảm bảo gắn bó dữ liệu trên các hệ thống phân tán, làm cơ sở cho lớp các ứng dụng đăng ký thông tin từ xa trên mạng Internet/Intranet trong điều kiện có sự cố của hệ thống.

2. Khẳng định được tính đúng đắn của giải pháp đề xuất bằng thực nghiệm, các tác giả đặt tên cho giải pháp 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.

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] E. Simon and P. Valduriez, *Integrity Control in Distributed Database Systems*, In Proc. 19th Hawaii Int. Conf. on System Sciences, January 1986.
- [7] <http://java.sun.com/products/jdk/rmi>.