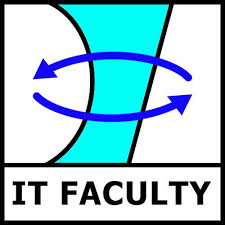
**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN**

**HỆ ĐIỀU HÀNH VÀ MẠNG MÁY TÍNH**

**Đề tài:**

**CHƯƠNG TRÌNH MÔ PHỎNG QUÁ TRÌNH HOẠT ĐỘNG CỦA 3 LOẠI THÔNG ĐIỆP REQ, ACQ VÀ REL**

**SINH VIÊN THỰC HIỆN:**

1. **Nguyễn Lê Tuấn Cương**
2. **Văn Ngọc Đạt**

**LỚP SINH HOẠT: 18TCLC\_DT3**

**Đà Nẵng, 10-2020**

# LỜI MỞ ĐẦU

# MỤC LỤC

[LỜI MỞ ĐẦU 2](#_Toc56444261)

[MỤC LỤC 3](#_Toc56444262)

[DANH MỤC HÌNH VẼ 4](#_Toc56444263)

[**1.**  **Cơ sở lý thuyết** 6](#_Toc56444264)

[1.1. Hệ phân tán 6](#_Toc56444265)

[1.1.1. Khái niệm hệ phân tán 6](#_Toc56444266)

[1.1.2. Đặc điểm hệ phân tán 6](#_Toc56444267)

[1.1.2.1. Tính chia sẻ tài nguyên 6](#_Toc56444268)

[1.1.2.2. Tính trong suốt 6](#_Toc56444269)

[1.1.2.3. Tính mở 7](#_Toc56444270)

[1.1.2.4. Tính co giãn 7](#_Toc56444271)

[1.2. Giải thuật loại trừ tương hỗ 7](#_Toc56444272)

[1.2.1. Mô tả thuật toán loại trừ tương hỗ: 8](#_Toc56444273)

[1.2.2. Kiểm tra giải thuật 9](#_Toc56444274)

[1.2.3. Phản ứng sự cố 9](#_Toc56444275)

[1.2.4. Thuyết minh 10](#_Toc56444276)

[1.2.5. Kết luận 11](#_Toc56444277)

[1.3. Kiến trúc và kỹ thuật RMI 11](#_Toc56444278)

[1.3.1. Tổng quan về RMI 11](#_Toc56444279)

[1.3.2. Kiến trúc RMI 12](#_Toc56444280)

[1.3.3. Kỹ thuật RMI 13](#_Toc56444281)

[**2.** **Xây dựng chương trình** 14](#_Toc56444282)

[2.1. Yêu cầu bài toán 14](#_Toc56444283)

[2.2. Các Modules chương trình 14](#_Toc56444284)

[Tài liệu tham khảo 14](#_Toc56444285)

# 

# DANH MỤC HÌNH VẼ

[Hình 1. Luồng dữ liệu giữa Stub và Skeleton 12](#_Toc56379160)

[Hình 2. Kiến trúc chương trình kiểu RMI 12](file:///C:\Users\DELL\Desktop\ĐA%20Hệ%20điều%20hành%20&%20Mạng%20máy%20tính\Báo%20cáo%20đồ%20án%20PBL%204.docx#_Toc56379161)

[Hình 3. Nguyên lý hoạt động của RMI 13](file:///C:\Users\DELL\Desktop\ĐA%20Hệ%20điều%20hành%20&%20Mạng%20máy%20tính\Báo%20cáo%20đồ%20án%20PBL%204.docx#_Toc56379162)

**GIỚI THIỆU ĐỀ TÀI**

Ngày nay, trong quá trình triển khai ứng dụng Tin học vào đời sống, các mạng máy tính được phát triển không ngừng, các tài nguyên của các máy tính trong mạng (phần cứng, phần mềm) ngày càng được mở rộng và nâng cấp, giá trị các tài nguyên này ngày càng tăng nhanh dẫn đến sự tăng trưởng vượt bậc nhu cầu chia sẻ tài nguyên và thông tin trong hệ thống thống nhất. Hệ điều hành tập trung và hệ điều hành mạng thuần túy không đáp ứng được nhu cầu đối với sự tăng trưởng đó. Việc giảm giá các trạm làm việc cũng tăng không ngừng mà từ đó làm tăng yêu cầu xử lý phân tán. Điều này tạo ra nhiều vị trí có khả năng xử lý và lưu trữ thông tin hơn mà từ đó cần thiết phải phối hợp để chia sẻ tốt tiềm năng lưu trữ và xử lý các vị trí đó. Trên cơ sở việc kết nối mạng để triển khai hệ điều hành mạng tạo nên một cơ sở kỹ thuật hạ tầng làm nền tảng phát triển hệ điều hành phân tán.

Dựa vào đó, nhóm chúng em chọn đề tài như sau: “Viết chương trình mô phỏng quá trình hoạt động của ba loại thông điệp REQ, REL, ACQ trong hệ tin học phân tán. Xây dựng hệ thống ba server thể hiện khả năng phát nhận thông điệp.”

Bố cục báo cáo được chia thành 3 phần chính, gồm:

Giới thiệu đề tài

1. Cơ sở lý thuyết
2. Xây dựng chương trình

Kết luận

Tài liệu tham khảo

# **Cơ sở lý thuyết**

## Hệ phân tán

### Khái niệm hệ phân tán

Hệ phân tán - hay hệ điều hành phân tán, là tổ hợp bao gồm các máy tính độc lập, kết nối lẫn nhau bằng mạng máy tính. Các phần mềm trên các máy tính khác nhau có khả năng phối hợp, chia sẻ tài nguyên. Hệ phân tán thực hiện một nhiệm vụ chung, cung cấp dịch vụ một cách thống nhất về giao diện, cách thức truy cập dịch vụ mà người sử dụng không cần phải quan tâm tới các chi tiết của hệ thống.

### Đặc điểm hệ phân tán

Hệ phân tán có các đặc điểm cơ bản là:

* Tính chia sẻ tài nguyên
* Tính trong suốt
* Tính mở
* Tính co giãn

#### Tính chia sẻ tài nguyên

Thuật ngữ tài nguyên được dùng để chỉ tất cả mọi thứ có thể được chia sẻ trong hệ phân tán, bao gồm từ các thiết bị phần cứng (Đĩa, máy in...) tới các đối tượng (file, các cửa sổ, cơ sở dữ liệu và các đối tượng dữ liệu khác).

Trong hệ phân tán, chia sẻ tài nguyên được hiểu là tài nguyên của hệ thống được các quản trị chia sẻ mà không bị hạn chế bởi tình trạng phân tán tài nguyên theo vị trí địa lý.

Việc chia sẻ tài nguyên trên hệ phân tán - trong đó tài nguyên bị lệ thuộc về mặt vật lý với một máy tính nào đó được thực hiện thông qua truyền thông. Để chia sẻ tài nguyên một cách hiệu quả thì mỗi tài nguyên cần phải được quản lý bởi một chương trình có giao diện truyền thông, các tài nguyên có thể truy nhập, cập nhật được một cách tin cậy và nhất quán. Quản lý tài nguyên ở đây bao gồm lập kế hoạch và dự phòng, đặt tên các lớp tài nguyên, cho phép tài nguyên được truy cập từ nơi khác, ánh xạ tên tài nguyên vào địa chỉ truyền thông….

#### Tính trong suốt

Đây là tính chất căn bản của hệ phân tán. Tính trong suốt của hệ phân tán được hiểu như là sự che khuất đi các thành phần riêng biệt của hệ thống máy tính (phần cứng và phần mềm) đối với người sử dụng và những người lập trình ứng dụng. Người sử dụng có quyền truy cập đến dữ liệu mà không cần biết đến sự phân tán của tất cả dữ liệu trên mạng. Hệ thống tạo cho người dùng cảm giác là dữ liệu được coi như đặt tại máy tính cục bộ của mình.

#### Tính mở

Tính mở của hệ phân tán là tính dễ dàng mở rộng phần cứng (thiết bị ngoại vi, bộ nhớ, các giao diện truyền thông ...) và phần mềm (các mô hình hệ điều hành, các giao thức truyền thông, các dịch vụ chia sẻ tài nguyên...) của nó. Nói một cách khác, tính mở của hệ thống phân tán mang ý nghĩa bao hàm tính dễ dàng cấu hình cả phần cứng lẫn phần mềm của nó.

Tỉnh mở của hệ phân tán được thể hiện là hệ thống có thế được tạo nên từ nhiều loại phần cứng và phần mềm của nhiều nhà cung cấp khác nhau với điều kiện các loại thành phần này phải theo một tiêu chuẩn chung.

Tính mở của hệ phân tán được thi hành dựa trên việc cung cấp cơ chế truyền thông giữa các quản trị và công khai các giao diện được dùng để truy cập tài nguyên chung.

#### Tính co giãn

Tính co giãn là khả năng của hệ thống có thể đáp ứng được các thay đổi của hạ tầng của môi trường xung quanh. Tính co giãn thường được xem xét dưới 3 góc độ: “co giãn về mặt quy mô, co giãn về mặt địa lý, co giãn về mặt tổ chức”.

* **Co giãn về mặt quy mô**: đảm bảo đáp ứng của hệ thống khi số lượng máy tính, số lượng người sử dụng, số lượng yêu cầu của người sử dụng gửi giữa các máy tính với nhau tăng.
* **Co giãn về mặt địa lý**: đảm bảo trao đổi thông tin trên mạng diện rộng như với mạng cục bộ.
* **Co giãn về mặt tổ chức**: khi tổ chức thay đổi hay máy tính dịch chuyển từ cùng tổ chức này sang cùng tổ chức khác. Khi đó tổ chức hệ thống thành các domain để khi cần thay đổi tổ chức ta chỉ cần thay đổi domain và thay đổi độ tin cậy giữa các domain đó.

## Giải thuật loại trừ tương hỗ

Loại trừ tương hỗ (Mutual Exclusion): là quá trình truy cập đồng thời của các tiến trình với một tài nguyên hoặc dữ liệu được chia sẻ, thực hiện theo cách loại trừ lẫn nhau. Trong hệ thống phân tán không có các biến chia sẻ có thể sử dụng để thực hiện loại trừ lẫn nhau.

Loại trừ tương hỗ có thể được thực hiện trên một trạm trung tâm có nhiệm vụ nhận tất cả các thông điệp và khuyến nghị giải phóng. Trạm này duy trì một hàng đợi, sắp xếp các yêu cầu theo trật tự đến và phục vụ cho từng thông điệp một trong trật tự này.

Phân tán giải thuật này kéo theo việc phân tán các chức năng cung cấp mà cần phải điều khiển hàng đợi trên trạm. Do vậy, một trạm chuyên cho việc nhận các yêu cầu và khuyến nghị giải phóng trên tất cả các trạm còn lại. Một trật tự giống nhau trên các trạm chỉ đạt được nếu ta áp dụng dấu trong các thông điệp bởi các đồng hồ logic truyền và đánh số các trạm. Quan hệ trật tự toàn bộ được định nghĩa, thêm vào đó, để cho một trạm có thể ra quyết định bằng việc tham chiếu duy nhất vào hàng đợi của mình, nó còn cần phải nhận một thông điệp của từng trạm khẳng định rằng không có thông điệp nào trước các thông điệp khác mà còn đang quá cảnh trên đường.

### Mô tả thuật toán loại trừ tương hỗ:

1. Trạm i của mạng có thể gửi thông điệp cho các trạm khác thông điệp có dạng (T, Hi, i), trong đó Hi là dấu của thông điệp có nghĩa là đồng hồ logic của nó và T có thể nhận một trong 3 giá trị REQ, REL, ACQ.

|  |  |  |
| --- | --- | --- |
| STT | Tên thông điệp | Chức năng |
| 1 | REQ | Thông điệp REQ gửi yêu cầu đến các trạm, khi trạm i muốn vào trong đoạn găng |
| 2 | REL | Thông điệp REL giải phóng được phát đi cho tất cả các trạm, khi trạm i đã rời khỏi đoạn găng |
| 3 | ACQ | Thông điệp ACQ phản hồi được gửi bởi trại j cho trạm i, khi trạm j đã nhận được từ trạm i thông điệp REQ |

1. Mỗi trạm quản lý một hàng đợi các thông điệp được sắp xếp theo quan hệ <thời gian, số> của từng thông điệp. Theo cấu trúc, hàng đợi luôn luôn chứa một thông điệp và chỉ một thường trực trên mỗi trạm, bao gồm trạm cục bộ. Khi có một thông điệp được gửi đi bởi trạm i, đồng thời nó cũng được ghi trong hàng đợi của trạm này.

Giả sử rằng mỗi hàng đợi bann đầu chứa các thông điệp:

Mi = (REL, Hinit, i)

Trong đó, i, Hinit là thời điểm khởi sự giống nhau cho tất cả các trạm.

1. Trên mỗi trạm, khi nhận được một thông điệp dạng (REQ, Hi, i) hay(REL, Hi, i), thông điệp này thay thế thông điệp Mi bất chấp nó là gì. Khi nhận thông điệp loại (ACQ, Hi, i), thông điệp này thay thế Mi ngoại trừ Mi là một yêu cầu mà trong trường hợp đó ACQ bị bỏ qua. Do vậy, ta có thể tiết kiệm việc gửi thông điệp ACQ cho trạm i khi trạm này đã gửi một thông điệp REQ và không còn thông điệp REL.
2. Trạm i được quyền vào đoạn găng khi thông điệp REQ của nó đến trước theo nghĩa của quan hệ <thời gian, số> của tất cả các thông điệp khác trong hàng đợi của nó.

### Kiểm tra giải thuật

Các yêu cầu vào đoạn găng được xử lý theo trật tự FIFO và theo quan hệ =>. Để chứng minh điều đó, ta lưu ý rằng khi trạm i quyết định vào đoạn găng, nó không thể đưa vào trong mạng yêu cầu REQ nào trước bối cảnh sau. Thực tế, khi trạm i vào đoạn găng có nghĩa là nó đã nhận thông điệp từ tất cả các trạm khác và tất cả các thông điệp đều sau thông điệp REQ của riêng nó.

Ta kiểm tra các đặc tính sau đây :

1. Trạm i đang ở trong đoạn găng là trạm duy nhất nằm trong đoạn găng ấy. Thực tế cho thấy thông điệp REQ được phát bởi i vẫn tiếp tục tồn tại trong tất cả các hàng đợi cho đến khi nó được thay thế bởi thông điệp REL.

2. Trạm đã yêu cầu vào đoạn găng phải đảm bảo thời hạn và phải ra khỏi đoạn găng sau một khoản thời gian xác định. Thuật toán thể hiện tính đồng đều và tránh được tổn thất.

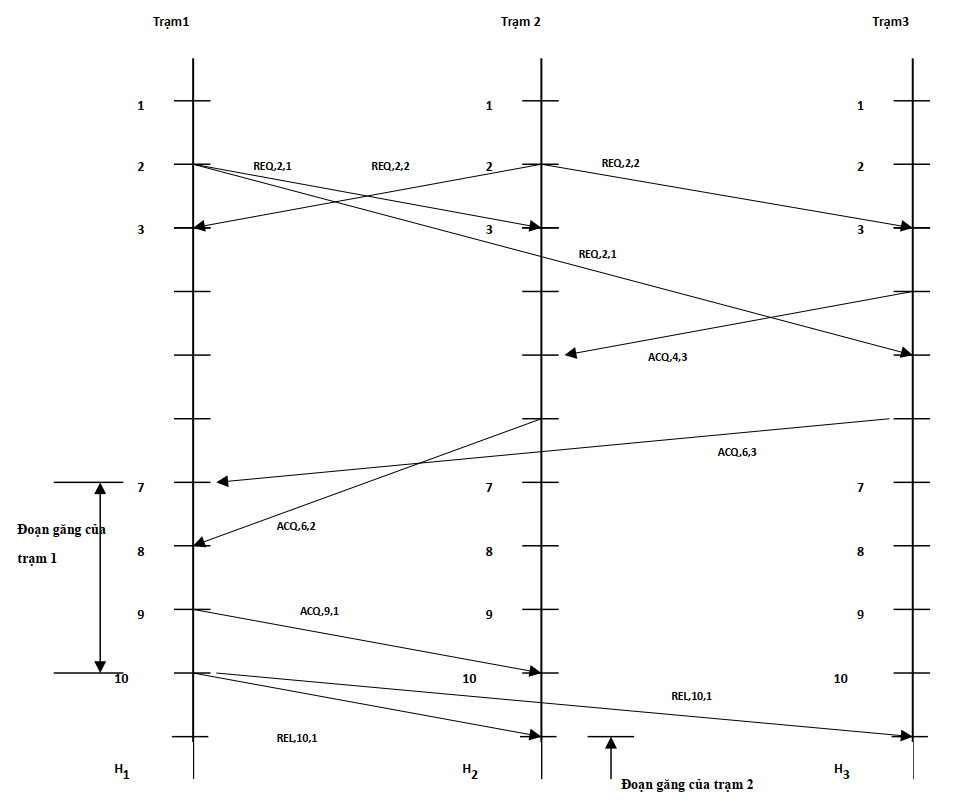
### Phản ứng sự cố

Sự cố xảy ra với trạm chưa vào được trong đoạn găng không làm rối loạn hoạt động của giải thuật với điều kiện là nó gây ra việc truyền thông điệp đặc biệt vang\_mat (vắng) cho việc chuyển tải ở mạng giao vận. Do có trang bị như thế, việc vào đoạn găng trở nên không được nhanh chóng và dễ dàng cho các trạm khác. Nếu trạm có sự cố đã gửi yêu cầu, thì nó kết thúc với lý do trở thành trước đối với tất cả các trạm khác.

Khi một trạm lại được đưa vào trong mạng sau khi đã có sự cố và đã khắc phục, nó cần phải kiến tạo lại trạng thái hiện hành của các yêu cầu. Để đảm bảo điều đó, nó phát đi thông điệp vao\_lai (xin vào lại) và để trả lời, các trạm gửi hoặc thời gian của yêu cầu cuối cùng của nó REQ không được thõa mãn (nếu có tồn tại một REQ) hoặc một thông điệp REL. Mạng cần phải bổ khuyết cho các trạm bị sự cố bằng cách gửi thông điệp vang\_mat. Khi nó đã nhận tất cả các trả lời cho thông điệp vao\_lai, trạm vừa đưa vào đó có thể bắt đầu lại bằng các yêu cầu.

### Thuyết minh

Để mô tả cho giải thuật này, điều kiện đủ là một trạm cần biết gần đúng trạng thái của các trạm khác trên mạng. Sự xấp xỉ này không đặt ra trạng thái mặc định cho loại trừ tương hổ. Nó có thể làm chậm một tí việc vào đoạn găng của một trạm.

Ví dụ: Xét một mạng gồm 3 trạm, trong số đó trạm 1 và trạm 2 yêu cầu vào đoạn găng tại thời điểm 2 của đồng hồ Logic của chúng. Tập hợp các thông điệp được truyền đi giữa chúng với nhau được mô tả theo hình vẽ sau và ta cũng có thể tìm thấy một số trạng thái của các hàng đợi thông điệp:

Ta nhận thấy rằng:

* Trạm 1 vào đoạn găng tại = 7
* Trạm 2 vào đoạn găng tại = 11

### Kết luận

Các dấu được cung cấp bởi đồng hồ Logic cho phép đánh dấu các sự kiện và xác định một tổng quát chặt chẽ. Nhưng tại đây không có quan hệ nào giữa các sự kiện và các giá trị của dấu. Trên một trạm cho trước, việc nhận một thông điệp có đóng dấu không thể cho nó biết còn sự kiện nào trước sự kiện đó đang ở trên đường. Như thế, ta còn phải nhận thông điệp từ các trạm khác còn lại.

## Kiến trúc và kỹ thuật RMI

### Tổng quan về RMI

Thông thường các chương trình của chúng ta được viết dưới dạng thủ tục hàm và việc các hàm gọi lẫn nhau và truyền tham số chỉ xảy ra ở máy cục bộ. Kỹ thuật RMI - mang ý nghĩa là triệu gọi phương thức từ xa là cách thức giao tiếp giữa các đối tượng trong Java có mã lệnh cài đặt nằm ở trên các máy khác nhau có thể triệu gọi lẫn nhau.

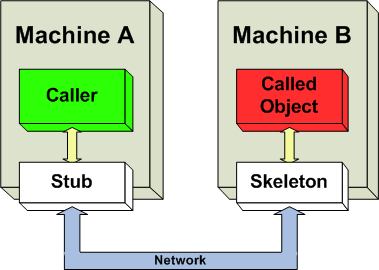
Các đặc tính của RMI:

* RMI là mô hình đối tượng phân tán của Java, giúp cho việc truyền thông giữa các đối tượng phân tán được dễ dàng hơn.
* RMI là API bậc cao được xây dựng dựa trên việc lập trình socket.
* RMI không những cho phép chúng ta truyền dữ liệu giữa các đối tượng trên các hệ thống máy tính khác nhau mà còn gọi được các phương thức trong các đối tượng ở xa.
* Việc truyền dữ liệu giữa các máy khác nhau được xử lý một cách trong suốt bởi máy ảo Java(Java virtual machine)
* RMI cung cấp cơ chế callback, nó cho phép Server triệu gọi các phương thức ở Client

Để giải quyết một số vấn đề trong việc truyền thông giữa client và server. RMI không gọi trực tiếp mà thông qua lớp trung gian. Lớp này tồn tại ở cả 2 phía client(Stub) và server(Skeleton).

Stub sẽ gọi kiến trúc RMI bên phía client và truyền dữ liệu đến kiến trúc RMI trên server, khi đến được nó sẽ gọi thực thi một đối tượng bên phía server(Skeleton).

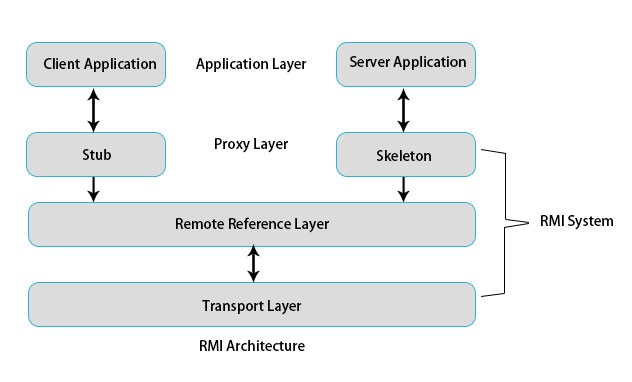
Đối tượng Skeleton sẽ gọi phương thức của đối tượng thất sự bên phía server. Đến khi trả lại kết quả thì một cơ chế giống hệt như trên sẽ được gọi thực thi nhưng theo thứ tự ngược lại, khi đó kết quả trả về sẽ được truyền cho đối tượng Skeleton trên máy chủ, và được đối tượng này truyền theo đường mạng sử dụng kiến trúc RMI, tiếp đến nó sẽ gọi đối tượng Stub phía client, và trả về giá trị cho đối tượng gọi phương thức.



Hình . Luồng dữ liệu giữa Stub và Skeleton

### Kiến trúc RMI

Kiến trúc của hệ thống RMI bao gồm các tầng như sau:



Hình . Kiến trúc chương trình kiểu RMI

* Tầng ứng dụng: Bao gồm client và server, trong đó:

Server:

* Hiện thực giao diện Remote mà client dùng.
* Xuất (Exports) đối tượng mà các phương thức của nó được triệu gọi từ xa (bằng cách tạo lớp là UnicastRemoteObject)
* Đăng ký chính nó với bộ đăng ký RMI.

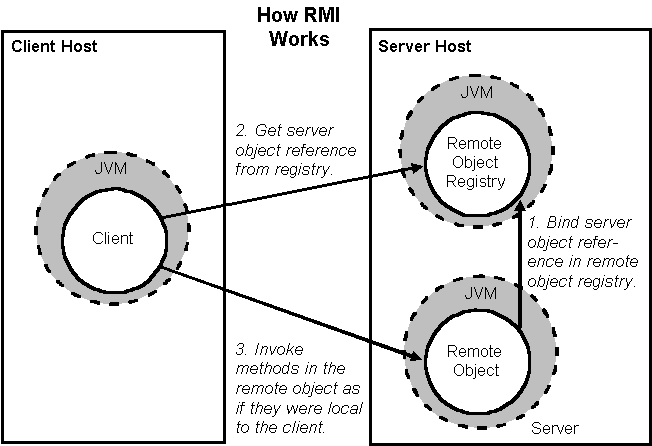
Client:

* Nhận tham chiếu đến đối tượng
* Chuyển các tham chiếu như những giao diện từ xa
* Triệu gọi các hành vi của đối tượng.
* Tầng ủy quyền: gồm Client(Stub) và Server(Skeleton)
* Tầng truy cập từ xa: Sự trừu tượng hóa giữa tầng ủy quyền và tầng vận chuyển.
* Tầng vận chuyển: Quản lý giao tiếp giữa các host với nhau trên mạng, dựa trên kết nối TCP/IP. Thực chất thực hiện kết nối giữa các máy ảo Java.

### Kỹ thuật RMI

Cơ chế làm việc của chương trình trên Client và Server khi đã được đăng ký bởi bộ đăng ký Rmiregistry như sau:

* Máy ảo Java (JVM) trên máy Server dùng hàm Bind() hoặc Rebind()của lớp Naming để đăng ký với Rmiregistry.
* Chương trình client dùng hàm Naming.Lookup() để yêu cầu bộ đăng ký cho biết tham chiếu đến đối tượng.
* Bộ đăng ký Rmiregistry trả về tham chiếu đối tượng tồn tại trên máy Server.
* Trên cơ sở tham chiếu của đối tượng do hàm Lookup() trả về, chương trình Client sử dụng để triệu gọi hành vi của đối tượng trên máy Server.



Hình . Nguyên lý hoạt động của RMI

# **Xây dựng chương trình**

## Yêu cầu bài toán

Xây dựng hệ thống 3 Server thể hiện khả năng phát/nhận thông điệp.

Viết chương trình mô phỏng quá trình hoạt động của 3 loại thông điệp cơ bản REQ, ACQ và REL trong hệ tin học phân tán.

## Các Modules chương trình

**Kết luận**

# Tài liệu tham khảo

[1]

[2]

[3]