1. Business Logic

Thực hiện các nghiệp vụ chính của hệ thống, sử dụng các dịch vụ do lớp Data Access cung cấp, và cung cấp các dịch vụ cho lớp Presentation. Lớp này cũng có thể sử dụng các dịch vụ của các nhà cung cấp thứ 3 để thực hiện công việc của mình(ví dụ như sử dụng dịch vụ của các cổng thanh tóan trực tuyến như VeriSign, Paypal...).

1. So sánh các ưu và nhược điểm của RMI và CORBA

* RMI
* Ưu điểm
* Đơn giản,dễ sử dụng
* Giúp cho việc truyền thông giữa các đối tượng phân tán được dễ dàng hơn.
* Không những cho phép chúng ta truyền dữ liệu giữa các đối tượng trên các hệ thống máy tính khác nhau, mà còn triệu gọi các phương thức trong các đối tượng ở xa (Remote Object).
* Nhược điểm
* Không thể sử dụng với các mã khác ngoài Java
* Tốc độ không nhanh bằng triệu gọi cục bộ hoặc sử dụng các giao thức liên kết với mục tiêu cụ thể
* Có thể không hiệu quả đối với các ứng dụng thời gian thực,như các ứng dụng Video
* CORBA
* Ưu điểm
* Client khong can biet interface cho cac server object
* Kha nang doc lap ve:Nen phan cung, ngon ngu lap trinh, he dieu hanh, moi gioi yeu cau cu the, thu bac cua phan tan doi tuong
* Cung cấp lựa chọn khi lấy kết quả trả về
* Nhược điểm
* Chương trình phức tạp
* Chi phí cao
* Dễ gây ra lỗi

1. Kiến trúc client-server 3 tầng

(Vẽ hình)

* Đối với client liên tác với Middleware(Logic tích hợp và Logic ứng dụng) gọi chung là Application Logic
* Kiến trúc client server phát triển và cài đặt các ứng dụng trong môi trường phân tán.
* Các ứng dụng mức phân tán yêu cầu các dịch vụ mức hệ thông như:

+ Quản lý giao dịch

+ Bảo mật

+ Kết nối client

+ Truy xuất dữ liệu

*Ưu điểm*:

- Dễ dàng mở rộng, thay đổi quy mô của hệ thống: Khi cần tải lớn, người quản trị có thể dễ dàng thêm các máy chủ vào nhóm, hoặc lấy bớt ra trong trường hợp ngược lại.

*Nhược điểm*:

- Việc truyền dữ liệu giữa các tầng sẽ chậm hơn vì phải truyền giữa các tiến trình khác nhau (IPC), dữ liệu cần phải được đóng gói -> truyền đi -> mở gói trước khi có thể dùng được.

- Việc phát triển ứng dụng phức tạp hơn.

1. Quá trình phát triển ứng dụng CORBA và RMI khác nhau thế nào?

CORBA dùng cùng kỹ thuật stub/skeleton như RMI, nhưng không giống như RMI, CORBA phát sinh stub và skeleton từ một mô tả giao diện độc lập với ngôn ngữ được gọi là Ngôn Ngữ Mô Tả Giao Diện (Interface Description Language – IDL) thay vì mã nguồn của ngôn ngữ.  
 IDL xác định tên phương thức, cũng như tham số gọi và trả về theo một kiểu ngôn ngữ trung lập

CORBA không có RMI Registry nên Dich Vụ Tên CORBA(CORBA Naming Service – COSNaming) cung cấp chức năng tìm kiếm vị trí đối tượng bên phía máy chủ.

Với RMI-IIOP, stub và skeleton được phát sinh ngay từ định nghĩa đối tượng Java. Thay vì sử dụng giao thức trong RMI để liên lạc giữa 2 tiến trình, RMI-IIOP sử dụng giao thức CORBA IIOP để nó có thể gọi các đối tượng không được viết bằng ngôn ngữ Java. Các bước phát triển bắt đầu với cài đặt một lớp phát sinh bằng IDL trong ngôn ngữ Java. Lớp này được biên dịch với rmic dùng cớ hiệu –iiop –d và một lớp bên phía máy chủ sử dụng COSNaming, truy cập Giao Diện Thư Mục và Tên Trong Java (Java Naming and Directory Interface – JNDI)

1. Application là gì?làm trong các mô hình ứng dụng

(Không hiểu câu hỏi:D)

6. So sánh CORBA và RMI

|  |  |
| --- | --- |
| Corba | RMI |
| -Doc lap ve Platform  -Doc lap ve giao thuc (IIOP)  -Doc lap ve ngon ngu (DDL)  -Cac doi tuong phan tan (gia tri)  -Triệu gọi đối tượng từ xa | -Chi ro JVM (may ao java)  -Giao thuc thuoc tinh (JRMP)  -Ngon ngu java  -Cac doi tuong phan tan (gia tri)  - Triệu gọi đối tượng từ xa |

Sự khác nhau giữa RMI và CORBA  
  
· RMI là một phần của bộ J2SDK và là các hàm thư viện hỗ trợ các lời gọi phương thức từ xa và trả về giá trị cho các ứng dụng tính toán phân tán. Chúng ta giả sử rằng ngôn ngữ Java được sử dụng ở cả phía gọi và phía bên phương thức được gọi  
  
· CORBA là một chuẩn công nghiệp cho phép gọi các phương thức từ xa và nhận kết quả trả về, nhưng không giống như RMI, nó có thể được sử dụng khi bên phía gọi và bên phía phương thức được gọi có thể sử dụng các ngôn ngữ lập trình khác nhau, bao gồm cả trường hợp là cả 2 bên đều không sử dụng ngôn ngữ Java  
  
· RMI là một tập các hàm thư viện đơn giản vì cả 2 bên đều sử dụng cùng môt ngôn ngữ lập trình và kiến trúc máy. Điều này sẽ làm cho vấn để triệu gọi phương thức từ xa dễ giải quyết hơn.  
  
· Bộ phát triển J2SDK cũng hỗ trợ RMI-CORBA, cho phép một đối tượng Java gọi một đối tượng CORBA sử dụng hai cách tiếp cận khác nhau.

7. Trong RMI, lam the nao de Client tim duoc dich vu tu xa?

Để Client tìm được dịch vụ từ xa,trước hết Server RMI phải đăng ký với một dịch vụ tra tìm và đăng ký tên. Dịch vụ này cho phép các client truy tìm chúng, hoặc chúng có thể tham chiếu tới dịch vụ trong một mô hình khác. Một chương trình đóng vai trò như vậy có tên là rmiregistry, chương trình này chạy như một tiến trình độc lập và cho phép các ứng dụng đăng ký dịch vụ RMI hoặc nhận một tham chiếu tới dịch vụ được đặt tên. Mỗi khi server đựơc đăng ký, nó sẽ chờ các yêu cầu RMI từ các client. Gắn với mỗi đăng ký dịch vụ là một tên được biểu diễn bằng một xâu ký tự để cho phép các client lựa chọn dịch vụ thích hợp. Nếu một dịch vụ chuyển từ server này sang một server khác, client chỉ cần tra tìm trình đăng ký để tìm ra vị trí mới. Điều này làm cho hệ thống có khả năng dung thứ lỗi-nếu một dịch vụ không khả dụng do một máy bị sập, người quản trị hệ thống có thể tạo ra một thể hiện mới của dịch vụ trên một hệ thống khác và đăng ký nó với trình đăng ký RMI.

8.So sanh nhung dac trung giong nhau va khac nhau giua RMI va CORBA

* Giống nhau

Là tầng giữa hỗ trợ việc gọi phương thức phân tán, trong đó mã lệnh không tập trung trên một máy mà được phân tán trên nhiều máy

Trong một ứng dụng không phân tán, đoạn mã trong một đối tượng có thể gọi phương thức của một đối tượng khác và máy ảo Java phân giải địa chỉ và truyền tham số từ đối tượng gọi đến phương thức được gọi, ngoài ra nó cũng trả về các giá trị cho đối tượng gọi thực thi phương thức

 Lời triệu gọi các phương thức đều thông qua lớp trung gian. Lớp trung gian tồn tại cả trên Server và Client. Phía **Client**lớp trung gian gọi là **stub**(lớp móc). Phía **Server**lớp trung gian gọi là **skeleton**(lớp nối).

* Khác nhau

**Đặc trưng RMI:**

* Chỉ cho phép một đối tượng đang chạy trên một máy ảo Java này ( Java Virtual Machine) gọi các phương thức của một đối tượng đang tồn tại trên một máy ảo Java khác (JVM).

**Đặc trưng CORBA:**

* Họat động với nhiều ngôn ngữ lập trình khác nhau và không chỉ với nền Java
* Khi nói đến server CORBA ta phải rằng đang nói đến quy trình trên server chứ không phải máy server.
* Một hoặc nhiều quá trình server CORBA có thể chạy trên cùng một máy.
* Mỗi quá trình server CORBA chứa một hoặc nhiều trường hợp về đối tượng CORBA, một hoặc nhiều giao diện CORBA

9.Trình bày và nhận xét về kiến trúc client-server 4 tầng

(Slide)