

Internal Assessment : II (Unit II).

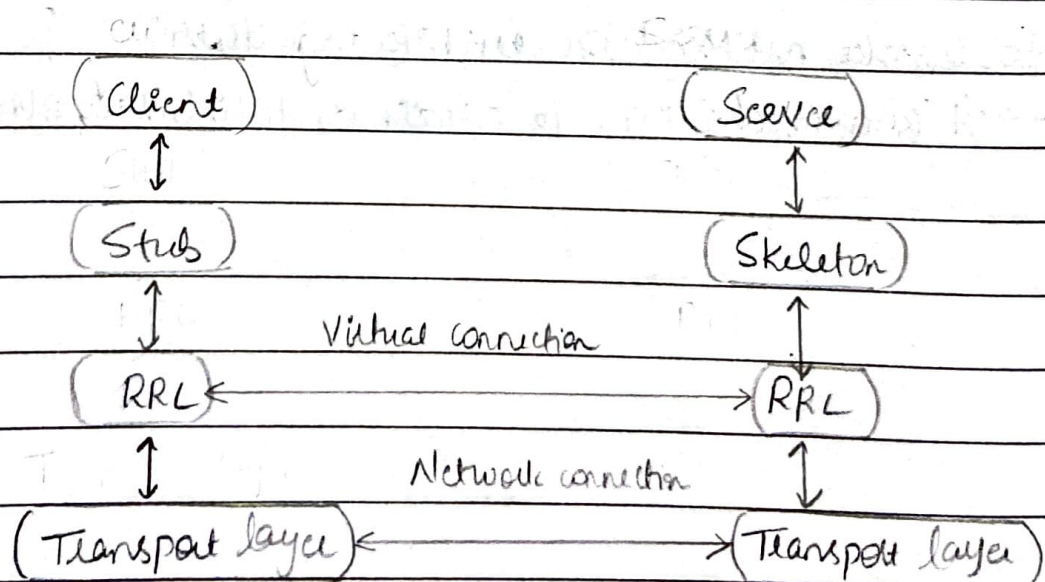
1. Communication between distributed objects by means of RMI.

→ RMI stands for Remote Method Invocation.

- It is a mechanism that allows an object residing in one system (JVM) to access/invoke an object running on another system (JVM).
- RMI is used to build distributed applications, it provides remote communication between java programs.
- It is provided in the package `java.rmi`.

Architecture of RMI application.

- In RMI application we write two programs a server program and client program.
- Inside the server program a remote object is created and lifetime of that object is made available for the client.
- The client program request the remote object on the server and tries to invoke its methods.

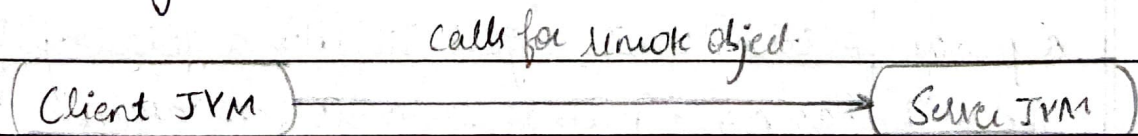


- Transport layer: Connects client to the server
- Stub: Representation (proxy) of remote object at client.
- Skeleton: Representation (proxy) of remote object at server.
- RRI (Remote Reference layer): Manages references made by client to remote object.

Working of RMI application.

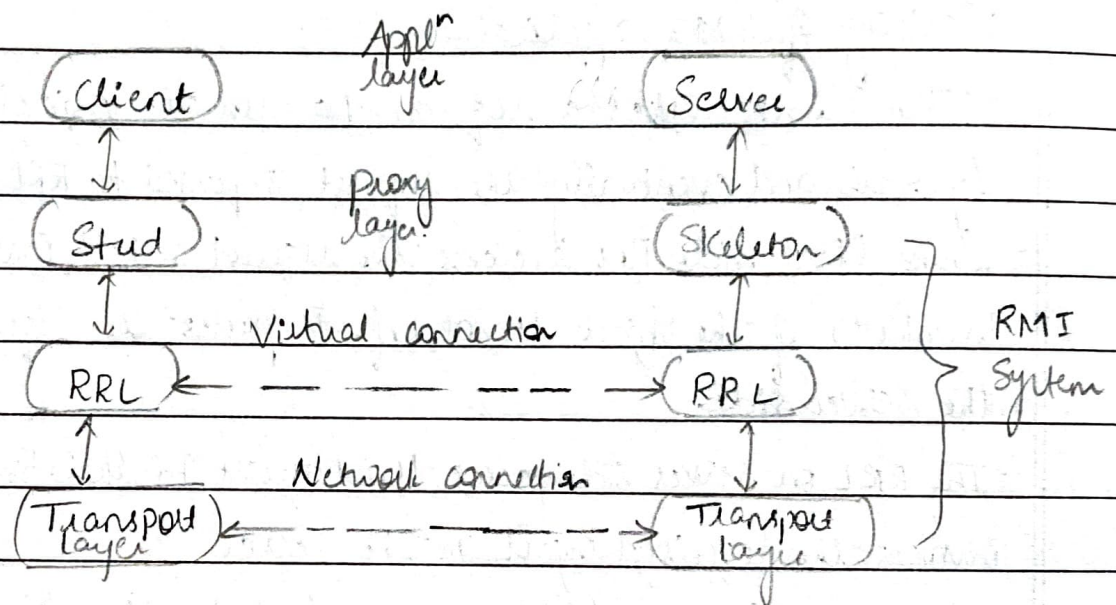
- When client makes a request to remote object it is received by stub and eventually the request is passed to RRI.
- When client side RRI receives the request and it invokes the method `invoke()` of the object `RemoteRef`. It passes the request to RRI on the server side.
- The RRI on server side passes the request to Skeleton which finally invokes the required object on the server.
- The result is passed all the way back to the client.

RMI design:



2 With neat labelled diagram explain the role of stub (proxy) and skeleton in RMI.

- In RMI application we write two types of programs one is client program and the other is server program.
- Inside the server program remote object is created and the client request the remote object in order to invoke its methods.



(i) Stub:

- It is an object that resides on the client machine and acts as a proxy for the remote object. It's like a gateway for client program.
- The stub has same method as a remote object.
- When client calls on stub object, it forwards this request to the remote object via RMI infrastructure which is then executed on the server side.
- Stub performs the following events.
 - Initiates connection with remote JVM
 - Writes and transmits (marshals) parameters to remote JVM

- Waits for the result
- Reads (Unmarshals) the returned result
- Passes the received result to the caller.

(ii) Skeleton :

- It is the server object which resides on the server machine.
- Stub communicates with server application with the help of an intermediate skeleton object.
- Main responsibility of skeleton object is to send parameters to method implementation and send the results back to client.
- Skeleton performs the following events
 - Reads the parameters passed by client
 - Invokes the method on actual remote object
 - Transmits / passes the result to the client.

3. With a neat diagram explain the role of fundamental concepts of the distributed object model.

- There are two fundamental concepts

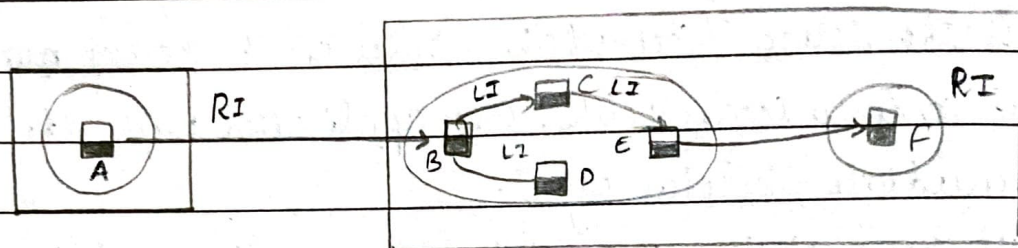
(i) Remote object reference

(ii) Remote interface

- Each process contains objects, some of which can receive remote invocation and are called as remote objects and others receive the local invocation and are called local objects.
- Objects need to know remote object reference of an object in another process in order to invoke its methods, called remote method invocation.

- Every remote object has a remote interface that specifies which of its methods can be invoked remotely.

Remote and local method invocation



RI - Remote Invocation

LI - local invocation

- Remote object reference
 - accessing the remote object
 - identifier throughout distributed system
 - can be passed as an argument
- Remote interface
 - specifies which methods can be invoked remotely
 - name, arguments, return type
 - interface definition language used for defining remote interface.

4. Discusses RMI Invocation Semantics and elaborate failure handling mechanism for each.

- Error handling for delivery guarantees
 - Retry request message: whether to retransmit the request message until either a reply is received or server is assumed to have failed.
 - Duplicate filtering: when retransmissions are used whether to filter out the duplicate requests at the server.
 - Retransmission of results: whether to keep a history of result messages to enable lost results to be retransmitted without re-executing operations.
- Choices of invocation semantics:
 - Maybe: a method is executed once or not at all
 - At-least-once: a method is executed at least one
 - At-most-once: method is executed exactly once.

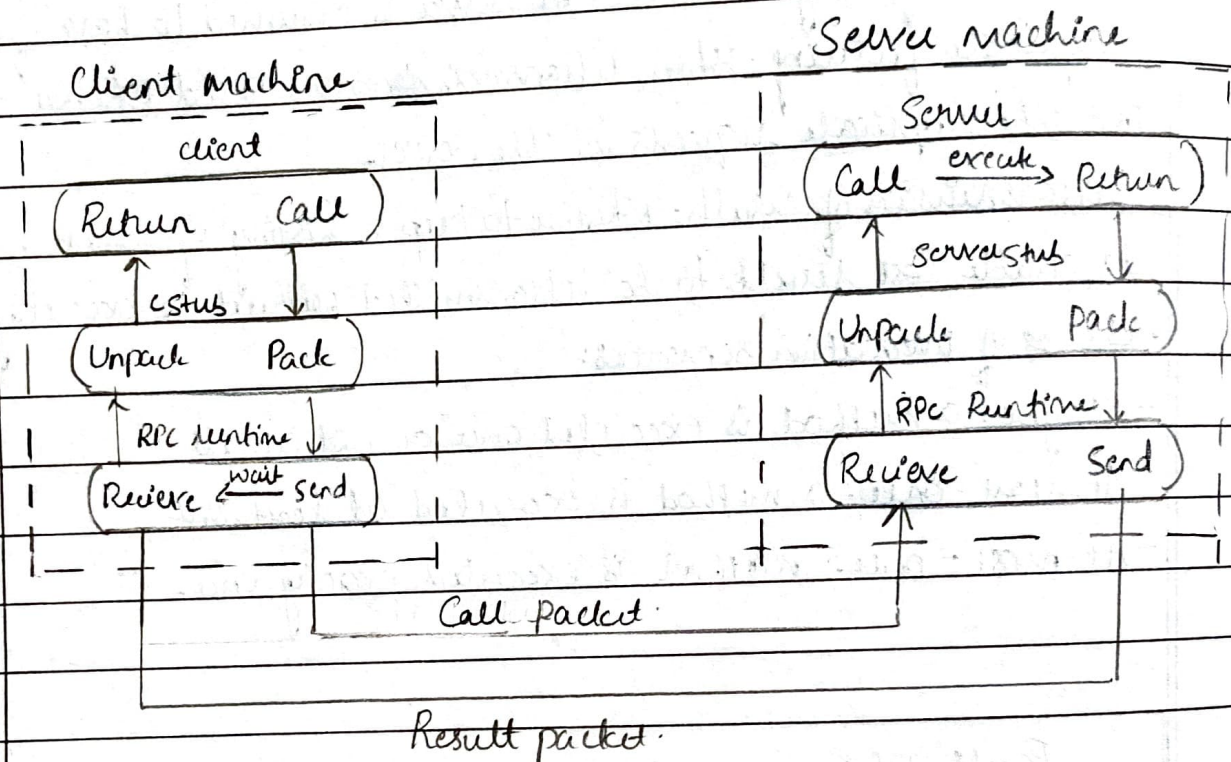
Fault Tolerance measure		Invocation semantics.	
Retry request message	Duplicate filtering	Retransmission of results.	
No	Not applicable	Not applicable	Maybe
Yes	No	Reexecute the procedure	At least once
Yes	Yes	Retransmit reply	At most once.

Invocation Semantics: choice of intent.

5.

Define and explain the implementation of RPC.

- RPC stands for Remote Procedure Call.
- It is a powerful technique for constructing client-server based applications.



Working of RPC.

- Client's stub residing in client machine invokes client stub procedure by passing the parameters in usual way.
- The client stub marshals the parameters into a message. This message is passed to transport layer which sends it to remote server machine.
- On server side, server stub passes the message to server stub which demarshals the parameters.
- After the completion of server side procedure, the values

are passed to the server stub which marshals the values into a message and passes it to transport layer.

- Transport layer sends the result message back to client stub
- Client stub demarshals the result and returns the values to the caller.