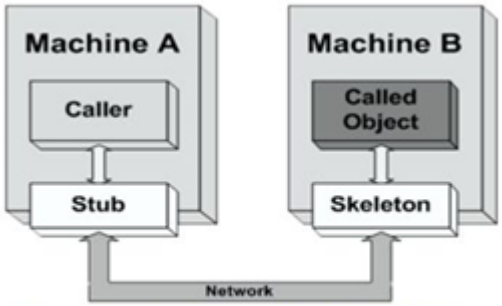
Various middleware languages like RMI required to make successful communication between distributed objects.

Stub and skeleton objects works as communication objects in distributed system.

RMI means remote method invocation. Whenever needed RMI invokes the methods at client and server side objects.



Communication between Distributed Objects

As shown in above diagram, in RMI communication follows the following steps:

1. A stub is defined on client side (machine A).
2. Then the stub passes caller data over the network to the server skeleton (machine B).
3. The skeleton then passes received data to the called object.
4. Skeleton waits for a response and returns the result to the client stub (machine A).

Stub

A stub is the client side object in distributed object communication.

The stub acts as a gateway for client side objects and all outgoing requests from client side

to the server-side objects.

The stub wraps or binds client object functionality & by adding the network logic ensures the reliable communication channel between client & server.

The stub can be manually or automatically generated depending on chosen communication protocol.

Skeleton

A skeleton is the server side object in distributed object communication.

The skeleton acts as the gateway for server side objects & for all incoming clients requests.

The skeleton wraps or binds server/called object functionality & exposes it to the clients; moreover by adding the network logic ensures the reliable communication channel between clients & server.

Skeletons can be manually or automatically generated depending on chosen communication protocol.

Related posts:

1. Advantages Disadvantages of DS
2. Distributed computing models
3. Goals of DS
4. Hardware software concepts
5. Issues in designing ds

6. Design and Implementation Issues DS
7. Structure of share memory space
8. DSM Architecture & its Types
9. File Application & Fault tolerance
10. File service architecture
11. Desirable features of good distributed file system
12. Distributed shared memory
13. RPC messages
14. Election algorithm
15. RPC Implementation
16. RPC mechanism
17. Client server communication
18. Datarepresentation and Marshalling
19. Load distributing algorithm
20. Task migration and its issues
21. Deadlock issues in deadlock detection & resolution
22. Distributed Scheduling-Issues in Load Distributing
23. Characterstics of Multimedia Data
24. Case Study of Distributed System
25. Distributed multimedia system
26. Distributed DBMS
27. Advantages of DDBMS over centralised DBMS