RPC

What is RPC?

❖ RPC, or Remote Procedure Call, is a communication protocol that allows a program on one computer to execute a function or procedure on a different computer, making it seem like a local call

❖ It abstracts away the network complexities, letting developers focus on application logic rather than network communication details

❖ A client sends a request for a procedure to a server, which executes it and returns the results, forming the basis of a client-server model in distributed systems
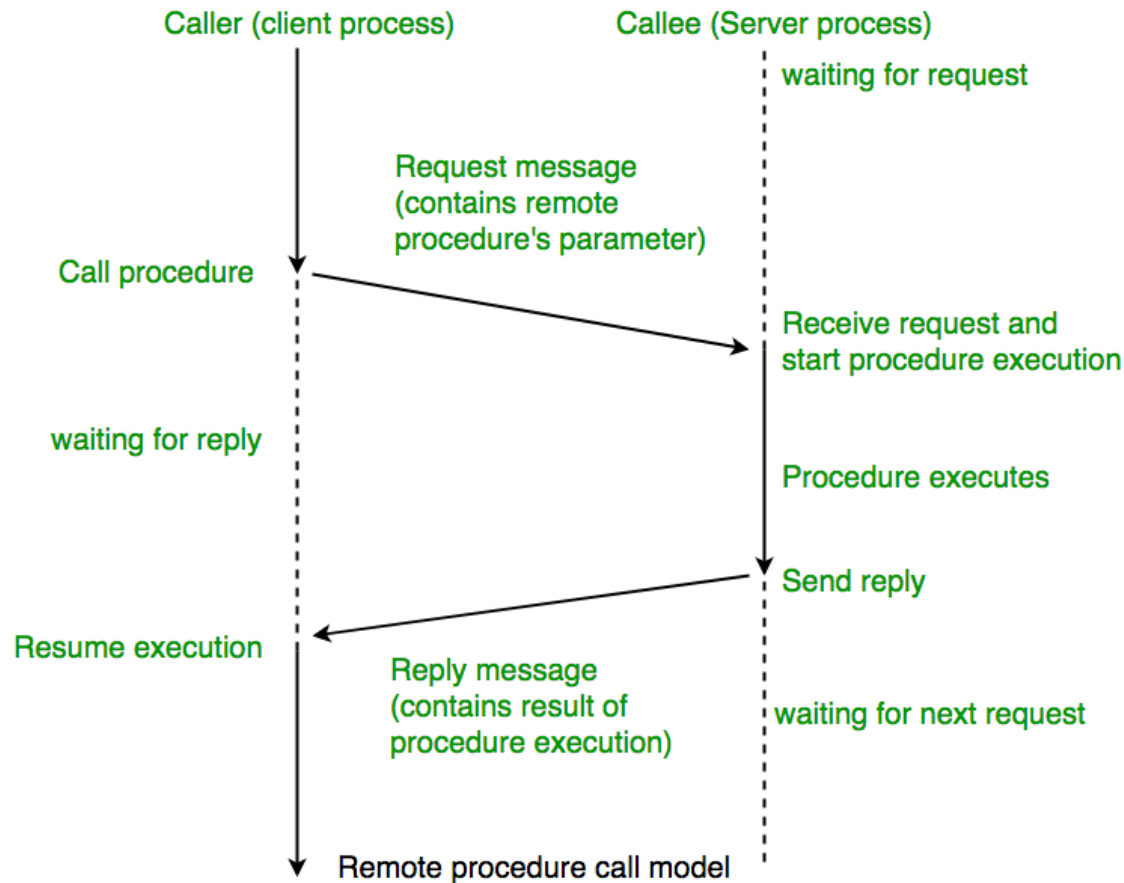
How RPC works?
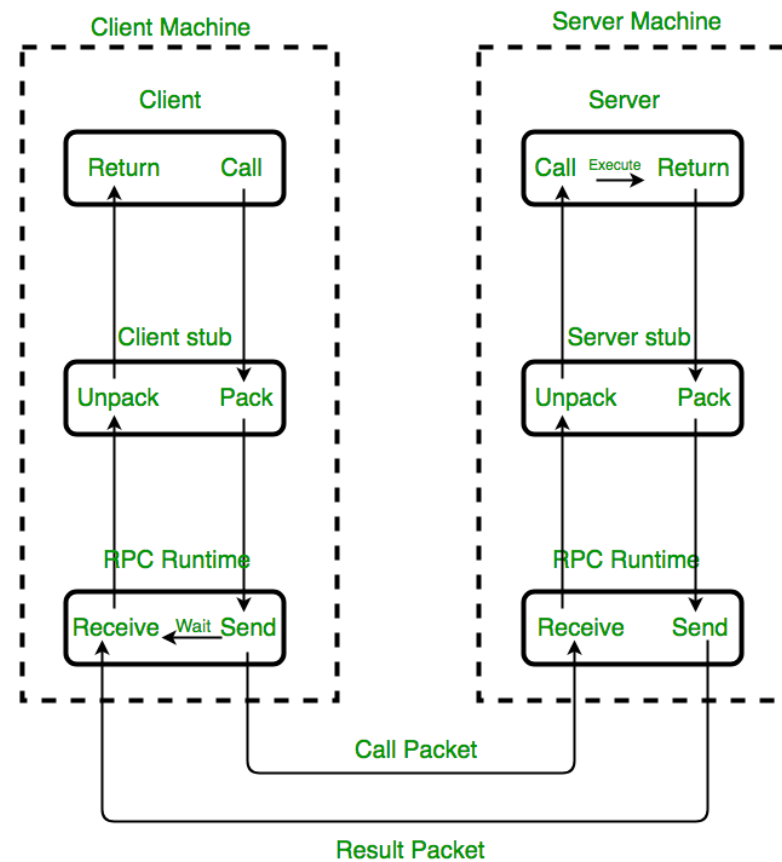
Implementation of RPC mechanism

RPC operates in a client-server model with the following steps:

- ❑ 1. Client Calls Stub:
- ❑ The client program calls a local procedure, known as a stub, just like any other local function.
- ❑ 2. Marshalling:
- ❑ The client stub packs the input parameters into a message.
- ❑ 3. Send to Server:
- ❑ The message is sent across the network to the server using a network protocol like TCP or UDP.
- ❑ 4. Server Stub Unpacks:
- ❑ The server stub receives the message, unpacks it, and calls the actual server procedure with the extracted parameters.
- ❑ 5. Execution and Return:
- ❑ The server executes the procedure and sends the result back to the server stub.
- ❑ 6. Return to Client:
- ❑ The server stub sends the result back across the network to the client stub, which then unpacks it, and the client application continues its execution with the result.

How to make an RPC call?

Caller (client process)   Callee (Server process)

waiting for request

Request message
(contains remote
procedure's parameter)

Call procedure

Receive request and
start procedure execution

waiting for reply

Procedure executes

Send reply

Resume execution

Reply message
(contains result of
procedure execution)

waiting for next request

Remote procedure call model

Types of RPC

❖ **Callback RPC**: Both client and server can act as each other. The server can call back the client, useful in interactive applications. Handles deadlocks and supports peer-to-peer communication.

❖ **Broadcast RPC**: The client's request is broadcast to all servers. Special broadcast ports are used. Useful when multiple servers can handle the request.

❖ **Batch-mode RPC**: Groups multiple client requests and sends them together to the server, reducing network overhead. Best for applications with infrequent calls.

What Does RPC do?

➢ RPC (Remote Procedure Call) lets a program use code on another computer as if it were local.

➢ A stub acts as a placeholder for the remote code in the client program.

➢ When the program calls the remote code, the stub sends the request to a local helper.

➢ The helper finds the remote server and sends the request over the network.

➢ The server runs the code and sends the result back through its own helpers.

➢ The client receives the result, making it look like the code ran locally.

Issues

❖ **RPC Runtime:** A library that manages the communication in RPC. It handles binding, sending/receiving data, choosing the protocol, and dealing with errors.

❖ **Stub:** A helper code that hides the complexity from the programmer. On the client side, it converts function calls into messages (marshalling/unmarshalling) and works with the runtime to connect to the server.

❖ On the server side, the stub provides a similar interface between the run-time system and the local manager procedures that are executed by the server.

❖ **Binding:** The most flexible solution is to use dynamic binding and find the server at run time when the RPC is first made. The first time the client stub is invoked, it contacts a name server to determine the transport address at which the server resides. Binding consists of two parts

Call semantics

- ✓ **Retry Request Message:** Whether to retry sending a request message when a server has failed or the receiver didn't receive the message.

- ✓ **Duplicate Filtering:** Remove the duplicate server requests.

- ✓ **Retransmission of Results**: To resend lost messages without re-executing the operations at the server side.

Advantages

❖ **Easy Communication:** RPC lets clients talk to servers using normal procedure calls in <u>high-level programming languages</u>. This makes it simple for programmers to work with.

❖ **Hidden Complexity:** RPC hides the details of how messages are sent between computers. This means programmers don't need to worry about the underlying network communication.

❖ **Flexibility:** RPC can be used in both local and distributed environments. This makes it versatile for different types of applications.

Compare RPC and REST

Disadvantages

Limited Parameter Passing: RPC can only pass parameters by value. It can't pass pointers, which limits what can be sent between computers.

Slower Than Local Calls: Remote procedure calls take longer than local procedure calls because they involve network communication.

Vulnerable to Failures: RPC depends on network connections, other machines, and separate processes. This makes it more likely to fail than local procedure calls.

SUN RPC

➢ Sun RPC, also known as Open Network Computing (ONC) RPC, is a remote procedure call system developed by Sun Microsystems that allows a program on one computer to execute a procedure in a program on another computer.

➢ It uses XDR (External Data Representation) to format data for transmission and can transport data over either TCP/IP or UDP/IP.

➢ A key component is the port mapper daemon, which listens on port 111 and registers RPC services, allowing clients to find and connect to them

Working of SUN RPC

❖ **Client initiates call:** A client program makes a "call" to a procedure on a remote server.

❖ **Data serialization:** The client's data parameters are serialized into XDR format.

❖ **Transport:** The XDR-encoded data is sent to the server using TCP or UDP.

❖ **Server execution:** The server receives the data, deserializes it, executes the procedure, and then serializes the results.

❖ **Reply to client:** The server sends the results back to the client in XDR format, which then deserializes the data.

Key components

➢ **RPC Interface Language:**

Sun RPC uses an interface definition language to specify the procedures and data types that can be called remotely.

➢ **rpcgen compiler:**

This C-based compiler processes the interface definition language to automatically generate client and server "stub" code, which handles the details of RPC communication.

➢ **External Data Representation (XDR):**

A language-independent data serialization method for translating data between different platforms.

➢ **Port Mapper (Binder):**

A daemon that acts as a central registry. It listens on a well-known port (port 111) and provides a way for clients to look up the dynamic TCP or UDP port numbers where specific RPC services are running.

Use cases

❑ **Sun NFS (Network File System):**

Sun RPC was initially designed for the Sun NFS project, a distributed file system.

❑ **Distributed Applications:**

It provides a foundational framework for building distributed applications where processes on different machines need to communicate and share services.