DBMS OPERATING SYSTEM

Tech Differences
789 likes

Like Page Send Messa

Be the first of your friends to like this

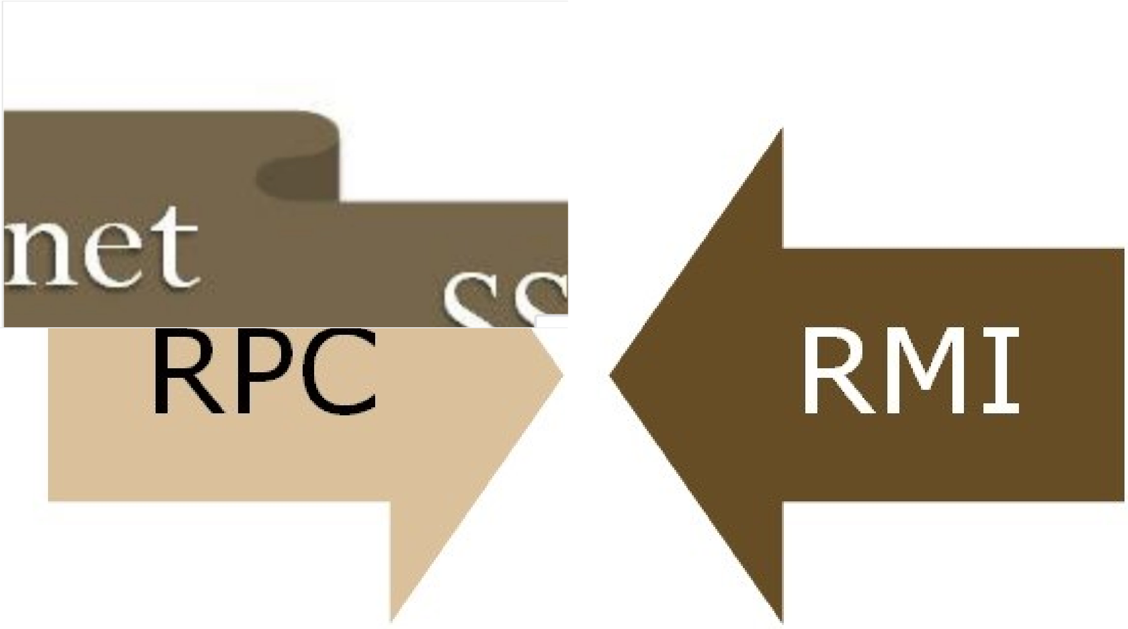
Tech Differences
about a week ago

Get the differences between two client-server application program, Telnet and SSH....... https://techdifferences.com/difference-between telnet-and-s…

es

t Certified by Google!

C and RMI



RPC and RMI are the mechanisms which enable a client to invoke the procedure or method from the server through establishing communication between client and server. The common difference between RPC and RMI is that RPC only supports **procedural programming** whereas RMI supports **object-oriented programming**.

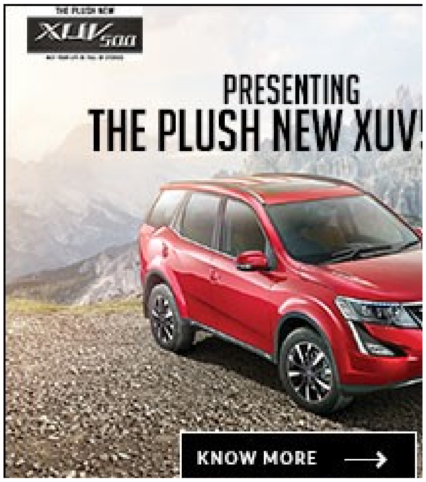
Another major difference between the two is that the parameters passed to remote procedures call consist of **ordinary data structures**. On the other hand, the parameters passed to remote method consist of **objects**.

Content: RPC Vs RMI

Comparison Chart

BASIS FOR COMPARISON	RPC	RMI
Supports	Procedural programming	Object-oriented programming
Parameters	Ordinary data structures are passed to remote procedures.	Objects are passed to remote methods.
Efficiency	Lower than RMI	More than RPC and supported by modern programming approach (i.e. Object-oriented paradigms)

BASIS FOR COMPARISON	RPC	RMI
Overheads	More	Less comparatively
In-out parameters are mandatory.	Yes	Not necessarily
Provision of ease of programming	High	low



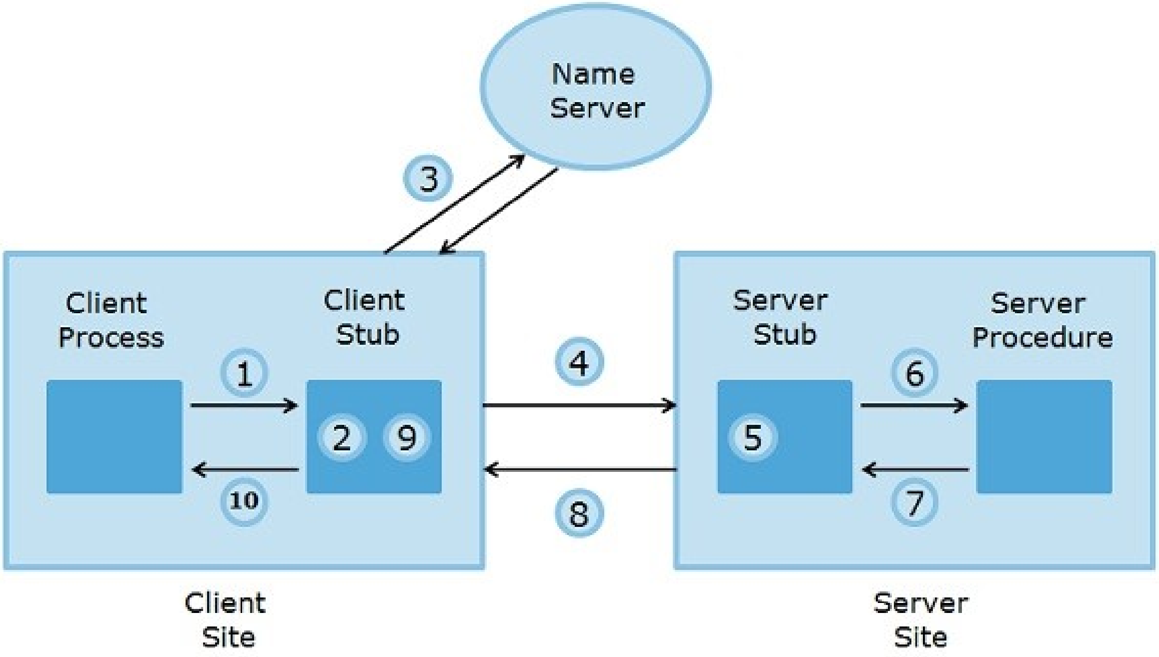
Definition of RPC

Remote Procedure Call (RPC) is a programming language feature devised for the distributed computing and based on semantics of **local procedure** calls. It is the most common forms of remote service and was designed as a way to abstract the procedure call mechanism to use between systems connected through a network. It is similar to IPC mechanism where the operating system allows the processes to manage shared data and deal with an environment where different processes are executing on separate systems and necessarily require message-based communication.

Let's understand how RPC is implemented through the given steps:

- The client process calls the client stub with parameters, and its execution is suspended until the call is completed.

- The parameters are then translated into machine-independent form by marshalling through client stub. Then the message is prepared which contain the representation of the parameters.
- To find the identity of the site the client stub intercommunicate with name server at which remote procedure exists.
- Using blocking protocol the client stub sends the message to the site where remote procedure call exists. This step halt the client stub until it gets a reply.

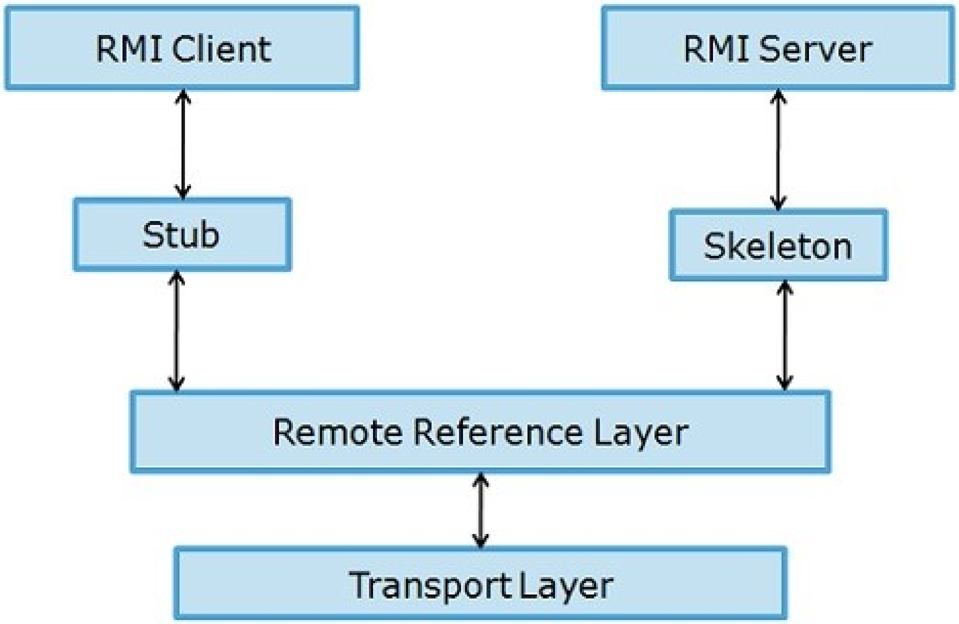


- The server site receives the message sent from the client side and converts it into machine specific format.
- Now server stub executes a call on the server procedure along with the parameters, and the server stub is discontinued till the procedure gets completed.
- The server procedure returns the generated results to the server stub, and the results get converted into machine-independent format at server stub and create a message containing the results.
- The result message is sent to the client stub which is converted back into machine specific format suitable for the client stub.
- At last client, stub returns the results to the client process.

Definition of RMI

Remote Method Invocation (RMI) is similar to RPC but is language specific and a feature of java. A thread is permitted to call the method on a remote object. To maintain the transparency on the client and server side, it implements remote object using stubs and skeletons. The stub resides with the client and for the remote object it behaves as a proxy.

When a client calls a remote method, the stub for the remote method is called. The client stub is accountable for creating and sending the parcel containing the name of a method and the marshalled parameters, and the skeleton is responsible for receiving the parcel.



The skeleton unmarshals parameters and invokes the desired method on the server. The skeleton marshals the given value (or exceptions) with the parcel and sends it to client stub. The stub reassembles the return parcel and sends it to the client.

In Java, the parameters are passed to methods and returned in the form of reference. This could be troublesome for RMI service since not all objects are possibly remote methods. So, it must determine which could be passed as reference and which could not.

Java uses process named as **serialisation** where the objects are passed as value. The remote object is localised by pass by value. It can also pass an object by reference through passing a remote reference to the object along with the URL of the stub class. Pass by reference restricts a stub for the remote object.

Key Differences Between RPC and RMI

1. RPC supports procedural programming paradigms thus is C based,

while RMI supports object-oriented programming paradigms and is java based.

2. The parameters passed to remote procedures in RPC are the ordinary data structures. On the contrary, RMI transits objects as a parameter to the remote method.
3. RPC can be considered as the older version of RMI, and it is used in the programming languages that support procedural programming, and it can only use pass by value method. As against, RMI facility is devised based on modern programming approach, which could use pass by value or reference. Another advantage of RMI is that the parameters passed by reference can be changed.
4. RPC protocol generates more overheads than RMI.
5. The parameters passed in RPC must be "**in-out**" which means that the value passed to the procedure and the output value must have the same datatypes. In contrast, there is no compulsion of passing "**in-out**" parameters in RMI.
6. In RPC, references could not be probable because the two processes have the distinct address space, but it is possible in case of RMI.

Conclusion

Both RPC and RMI serves the same purpose but are used in languages support different programming paradigms, therefore have distinct features.

Related Differences:

1. **Difference Between Call By Value and Call by Reference**
2. **Difference Between Client-Server and Peer-to-Peer network**
3. **Difference Between SMTP and POP3**
4. **Difference Between Trigger and Procedure**
5. **Difference Between ref and out in C#**

Filed Under: Operating System

Comments

Ashwin Dhakal says
March 24, 2018 at 1:48 am

beautiful, thanks for the description 🙂

Reply

Aditya Garg says
April 5, 2018 at 1:48 am

Very elaborately explained. Thanks.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

I'm not a robot

reCAPTCHA
Privacy - Terms

POST COMMENT

Search the site ...

TOP 10 DIFFERENCES

Difference Between Logical and Physical Address in Operating System

Difference Between Preemptive and Non-Preemptive Scheduling in OS

Difference between Synchronous and

Asynchronous Transmission

Difference Between Paging and Segmentation in OS

Difference Between Internal and External fragmentation

Difference Between while and do-while Loop

Difference Between LAN, MAN and WAN

Difference Between Pure ALOHA and Slotted ALOHA

Difference Between Recursion and Iteration

Difference Between Go-Back-N and Selective Repeat Protocol

RECENT ADDITION

Difference between H.323 and SIP

Difference between LAN and VLAN

Difference between PCM and DPCM

Difference between Fast Ethernet and Gigabit Ethernet

Difference between Subnetting and Supernetting

CATEGORIES

DBMS

Internet

Networking

Operating System

Programming

Software

Copyright © 2018 · TechDifferences · Contact Us