

RPC: first parallel programming models

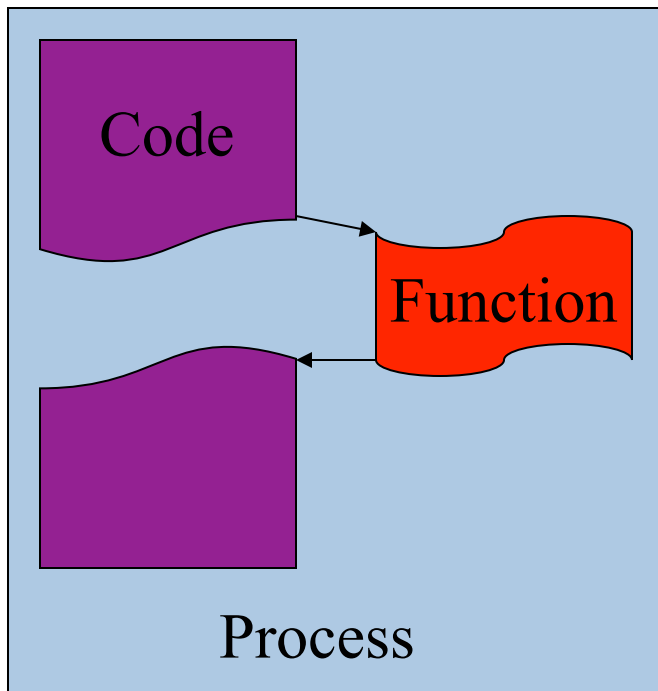
George Bosilca

bosilca@eecs.utk.edu

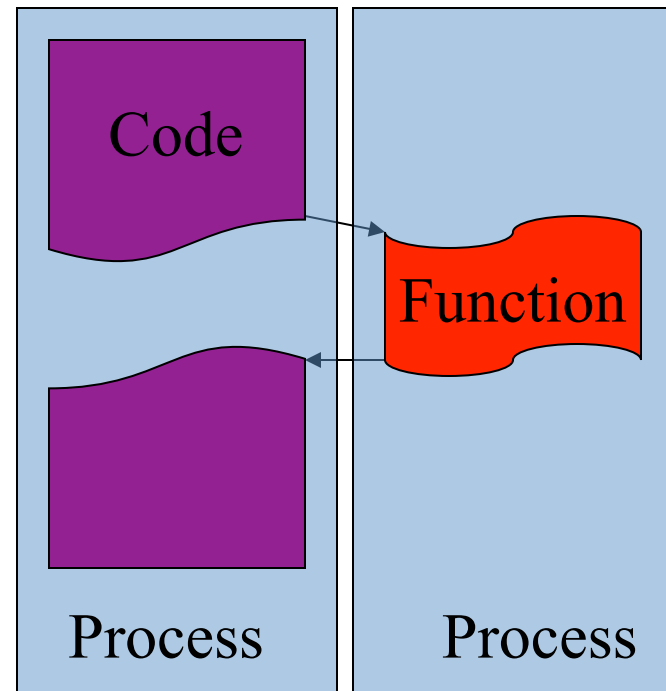
Remote Procedure Call

- Sun 1980 (?)

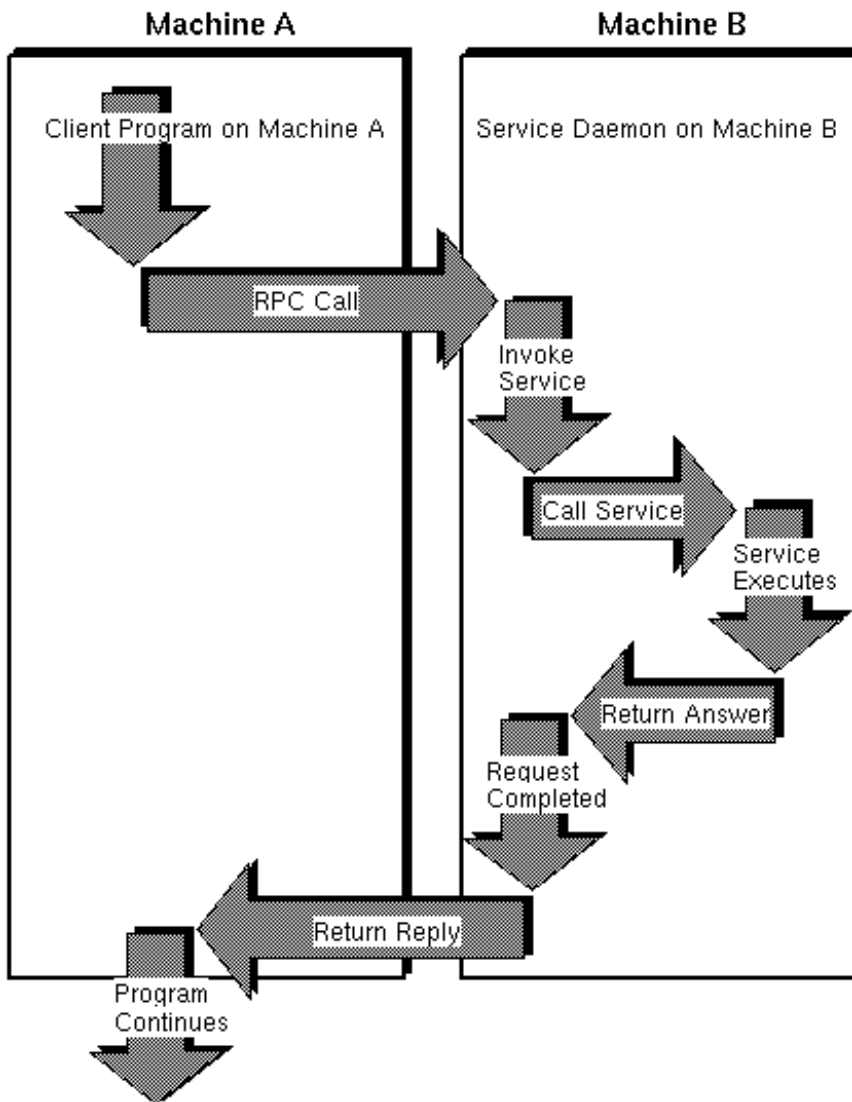
Function Call



RPC



RPC



- Hide the underlying network transport
- Mostly synchronous
- Manual data conversion XDR
- RPC reliability ?

RPC - reliability

- Again about reliability ...
- If and only if the user program is reliable
- Example:
 - On UDP (non reliable protocol) RPC retransmitted after timeout. When reply is received, the application infers that the RPC has been executed **at least 1 times**.
 - On TCP (reliable protocol) reply = only one execution, but no reply doesn't means no execution ... Still need timeout to handle server crashes.

RPC – selecting network protocol

– UDP (non reliable) if :

- procedures are idempotent (no side effects for multiple executions).
- The size of arguments or results is less than the RPC packet size (8K)
- The server should handle hundred clients.

– TCP (reliable) if :

- The application needs a reliable underlying transport
- The procedures are non-idempotent
- The size of either the arguments or the results exceeds 8K bytes

RPC – eXternal Data Representation

- Network standard representation
- Machine-independent description and encoding of data
- Both sides involved:
 - Machine format to XDR = **serializing**
 - XDR to machine format = **deserializing**
- Handle arbitrary data structures

RPC – Middle Layer

- callrpc & registerrpc (UDP)

_____ Client _____

```
if (stat = callrpc(argv[1],
    RUSERSPROG, RUSERSVERS, RUSERSPROC_NUM,
    xdr_void, 0, xdr_u_long, &nusers) != 0) { /* report error */ }
```

_____ Server _____

```
unsigned long * nuser(char* indata)
{ /* do something useful */
    return some_unsigned_long; }
registerrpc(RUSERSPROG, RUSERSVERS, RUSERSPROC_NUM,
    nuser,
    xdr_void, xdr_u_long);
svc_run(); /* Never returns */
```


RPC – Lower Layer

- It enables you to use TCP as the underlying transport instead of UDP, without restriction on the data size.
- It enables you to allocate and free memory explicitly while serializing or deserializing with XDR routines.
- It enables authentication on either the client or server side, through credential verification