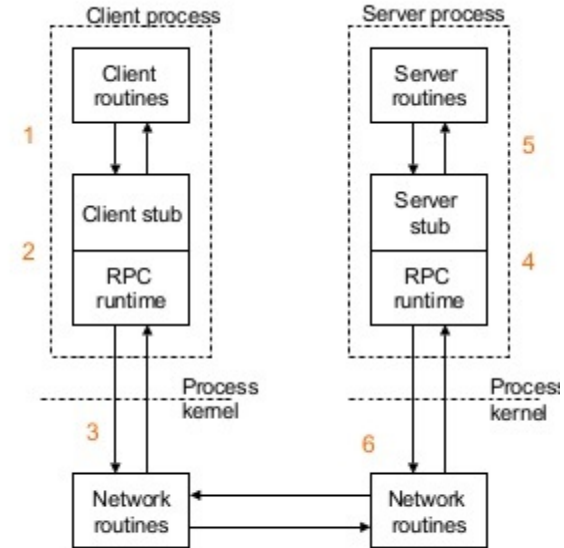# "Lightweight Remote Procedure Call"

Nayan Bhatia

What is Remote Procedure Call?

- RPC calls: Cross-machine or Cross-Domain(micro-kernel)
- Server provides interface for procedures/function to the client.
- Client makes local procedure calls.
- The local procedure hides the details of the network communication.

Source: R. Stevens, Unix Network Programming Vol 2, 1998

https://web.cs.wpi.edu/~cs4514/b98/week8-rpc/week8-rpc.html

Communication in different kernels

Monolithic kernel:
- Easy communication between kernel threads due to shared address space.

Micro kernel:
- Each module in its own "protection domain", since can only access its own address space.
- For communication between these protected domain, RPC is used(Cross domain RPC).

https://documen.site/download/pptx-5acd5f67b613b_pdf

Confusion

Why call it "Remote" Procedure Call for the same machine?
- Same RPC interface for local and remote clients
- The RPC architecture optimizes under the hood calls to local servers, so that local clients are not penalized by some remote IPC technique that would be sub-optimal for local communication (light RPC).

Problems with RPC in cross-domain

- RPC systems incur an unnecessarily high cost on the same machine
- Coalesce weakly related subsystems into the same protection domain, trading safety for performance
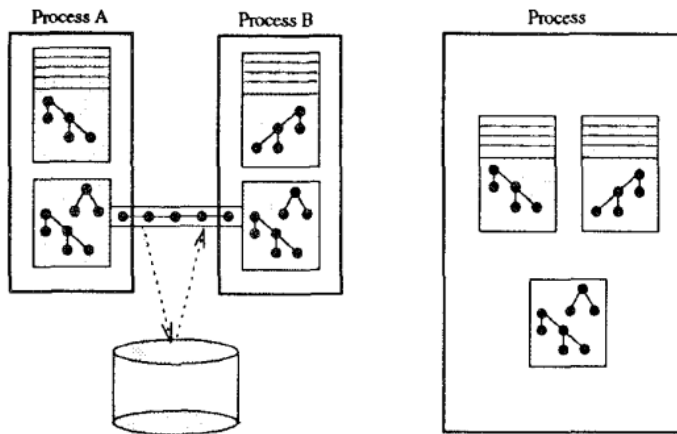
Light RPC

- Simple Control Transfer- Request on server's domain
- Simple Data Transfer- Eliminate redundant data copy
- Simple Stub-Simple and fast stub in assembly language
- LRPC on MP

USE AND PERFORMANCE OF RPC SYSTEMS

- Frequency of Cross-Machine Activity
- Parameter Size and Complexity
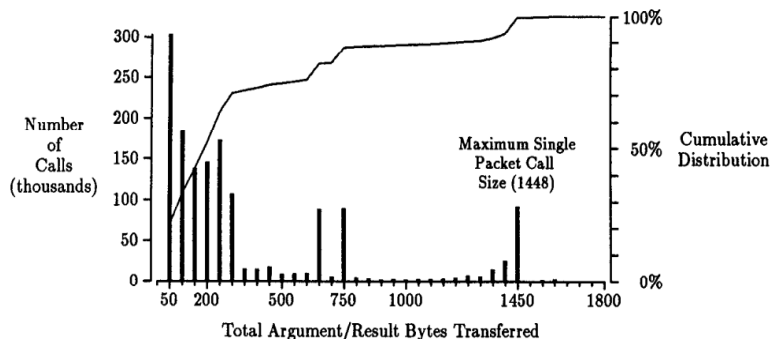- The Performance of Cross-Domain RPC

Table I. Frequency of Remote Activity

Operating system	Percentage of operations that cross machine boundaries
V	3.0
Taos	5.3
Sun UNIX+NFS	0.6

Fig. 1. RPC size distribution.

Table II. Cross-Domain Performance (times are in microseconds)

System	Processor	Null (theoretical minimum)	Null (actual)	Overhead
Accent	PERQ	444	2,300	1,856
Taos	Firefly C-VAX	109	464	355
Mach	C-VAX	90	754	664
V	68020	170	730	560
Amoeba	68020	170	800	630
DASH	68020	170	1,590	1,420

7

Design and Implementation of LRPC

- Execution Model
- Binding
- Calling
- Stub Generation
- LRPC on Multiprocessor (MP)
- Argument Copying

LightRPC Discussion

1. In the world of Unix, are LRPC still relevant today?
2. How are portals different from LightRPC?