

Implementing Remote Procedure Calls*

Birrell, A. D. and Nelson, B. J.

Presented by Emil Constantinescu

*ACM Trans. Comput. Syst. 2, 1 (Feb. 1984), 39-59

Overview

- Brief introduction
- RPC issues
- Implementation
- Examples
- Current RPC implementations
- Review
- Conclusions

09/14/05

2

What are Remote Procedure Calls (RPCs) in a nutshell (1)

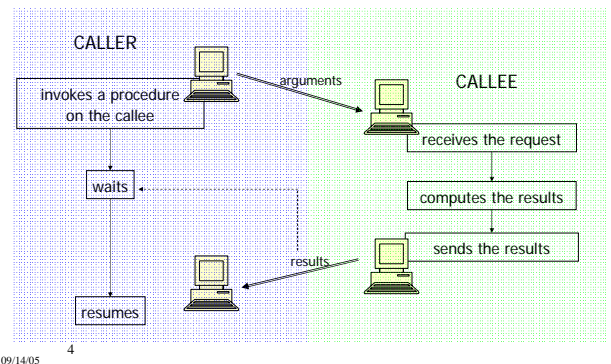
RPCs represent a set of communication paradigms that allow one procedure to call another procedure on a different machine.

1. one procedure (**caller**) calls another procedure (**callee**)
2. the **caller waits** for the result from the callee
3. the **callee receives the request**, computes the results, and then **sends them to the caller**
4. the **caller resumes** upon receiving the results from the callee

09/14/05

3

What are Remote Procedure Calls (RPCs) in a nutshell (2)



09/14/05

4

What are Remote Procedure Calls (RPCs) in a nutshell (3)

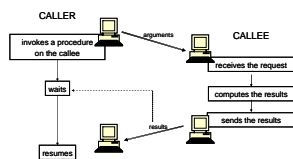
PURPOSE: Make distributed computing easy!

MAIN PRINCIPLE:

The procedure's communication patterns are transparent to the user. The procedure invocation looks just like a local procedure call.

Attractive aspects:

- simple semantics
- efficiency
- generality



09/14/05

5

RPC Issues (1)

- transparency, mimics the local procedure call
- precise semantics
 - ❖ machine failure
 - ❖ communication failure
- address based arguments
 - ❖ the address space is not shared
- programming integration
 - ❖ integration into programming environment
- data integrity

09/14/05

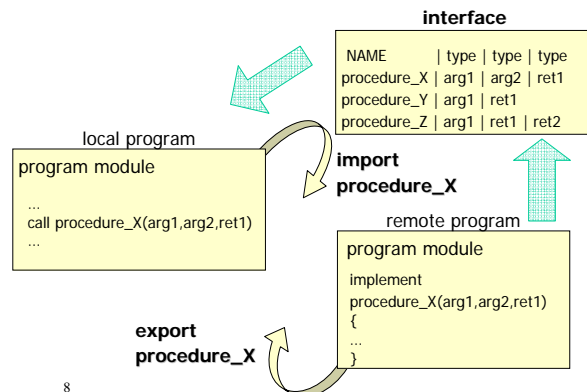
6

RPC Issues (2)

- data transfer protocols
 - ❖ network protocols
- binding
 - ❖ caller determines
 - the location
 - the identity
 - ❖ ... of the callee
- security
 - ❖ open communication network

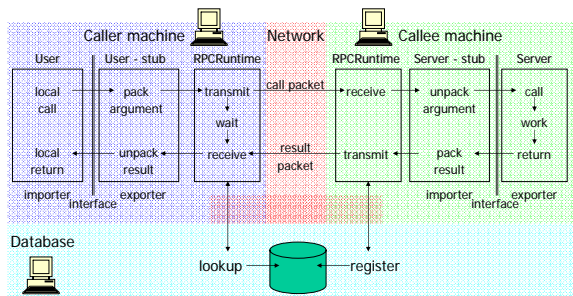
09/14/05 7

RPC Implementation (1) – Basic Concepts



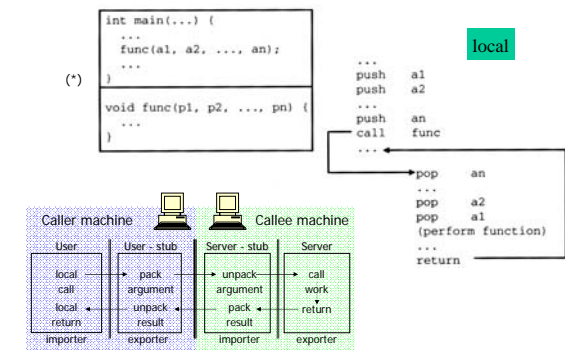
09/14/05 8

RPC Implementation (2) – Overview



09/14/05 9

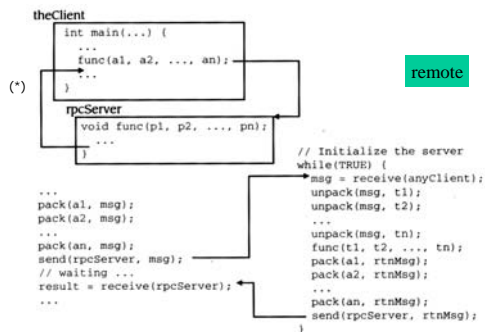
RPC Implementation (3) – Local vs. Remote



09/14/05 10

* Garry Nutt, Operating systems, 2nd Edition, Addison Wesley

RPC Implementation (4) – Local vs. Remote



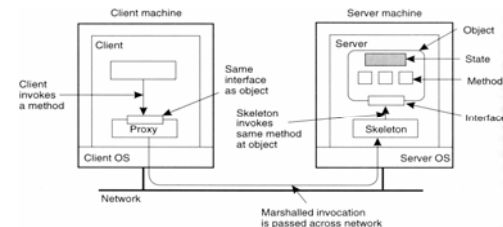
09/14/05 11

* Garry Nutt, Operating systems, 2nd Edition, Addison Wesley

RPC Implementation (5) – User/Server Stub

User/Server Stub:

- arguments are serialized/marshaled
- handles language binding (IDL, object passing)
- pass by value not by reference

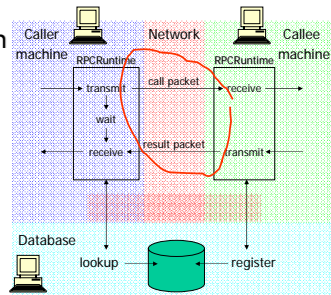


09/14/05 12

RPC Implementation (6) – RPCRuntime

RPCRuntime deals with

- data (re)transmission
- data acknowledge
- packet routing
- encryption
- exception handling
- binding



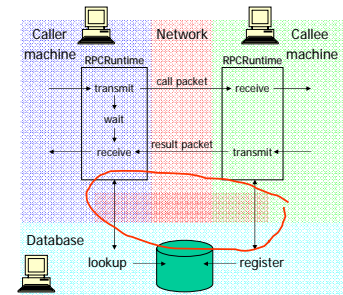
09/14/05 13

RPC Implementation (7) – Binding

Binding

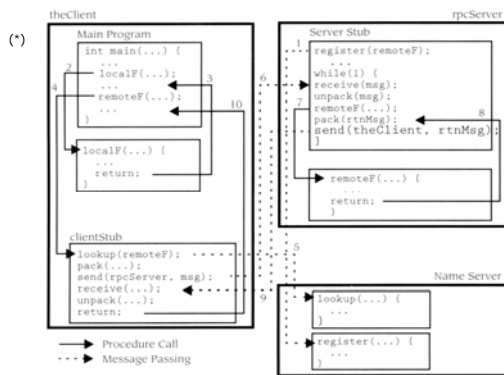
... is concerned with

- location
- identity
- ... implements
- ImportInterface
- ExportInterface
- ... has issues
- granularity
- security



09/14/05 14

RPC Implementation (8) – RPC flow

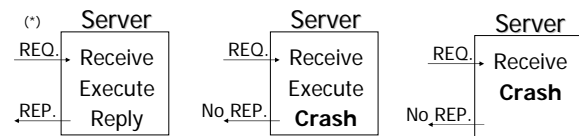


15 * Garry Nutt, Operating systems: 2nd Edition, Addison Wesley

RPC Implementation (9) – Semantics

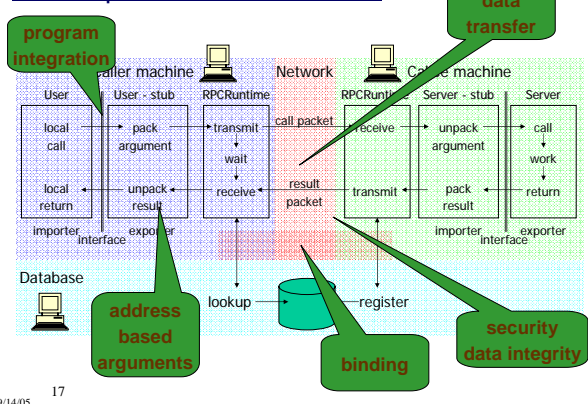
Semantics & Data Transfer issues

- scalable, connectionless scheme, reduced acknowledgements
- unique ID/incremental packets
- at least/most or exactly once semantics



16 * A. Tanenbaum and M. v. Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2002

RPC Implementation & Issues



09/14/05 17

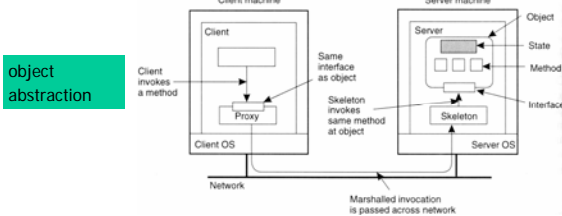
RPC Implementations (1)

- DCE RPC (Distributed Computing Environment RPC)
- SUN RPC
- DCOM (Distributed COM)
- CORBA (Common Object Request Broker Arch.)
- XML RPC
- SOAP (Simple Object Access Protocol)

09/14/05 18

RPC Implementations (2)

- DCE RPC
 - low level
 - handles the binding & transport
- Sun RPC
 - DCOM & CORBA
 - object abstractions



09/14/05 19 A. Tanenbaum and M. v. Steen; *Distributed Systems: Principles and Paradigms*, Prentice Hall, 2002

RPC Implementations (3)

- XML RPC & SOAP
 - somewhat lightweight
 - use HTTP and XML
 - Port 80

09/14/05 20

Conclusions & Future Research

- Transparency is imperative, and leads to effectiveness
- Maintain local procedure calling semantics
- Binding strategies influences efficiency
- Emulate shared address space
- Timeout implementation

09/14/05 21

About the paper (1)

Paper:

- Birrell, A. D. and Nelson, B. J.
- "Implementing Remote Procedure Calls"
- ACM Transactions on Computer Systems
- published in 1984
- first paper to formalize RPC
- concepts started to appear in 1976

09/14/05 22

About the paper (2)

- Special treatment for missing remote implementation (late binding)
- What if the client crashes?
- Machine specific binary data representation
- RPC in the HPC field?

09/14/05 23

References

Birrell, A. D. and Nelson, B. J.; **Implementing remote procedure calls**; ACM Trans. Comput. Syst. 2, 1 (Feb. 1984), 39-59

Garry Nutt; **Operating systems**; 2nd Edition, Addison Wesley

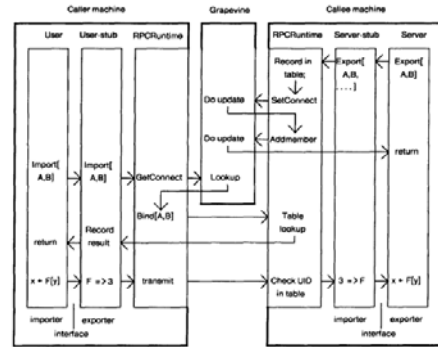
Andrew Tanenbaum and Maarten van Steen; **Distributed Systems: Principles and Paradigms**, Prentice Hall, 2002

09/14/05 24

Any Questions?

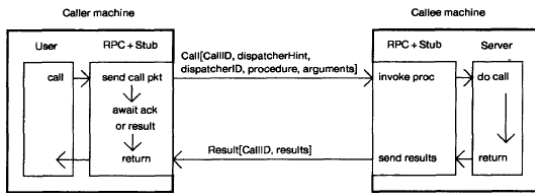
Paper Implementation (1)

Binding



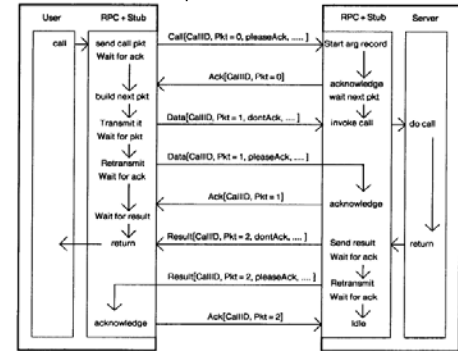
Paper Implementation (2)

Simple RPC



Paper Implementation (3)

"Complicated" RPC



Paper Implementation (4)

Results

Procedure	Minimum	Median	Transmission	Local-only
no args/results	1059	1097	131	9
1 arg/result	1070	1105	142	10
2 args/results	1077	1127	152	11
4 args/results	1115	1171	174	12
10 args/results	1222	1278	239	17
1 word array	1069	1111	131	10
4 word array	1106	1153	174	13
10 word array	1214	1250	239	16
40 word array	1643	1695	566	51
100 word array	2915	2926	1219	98
resume except'n	2555	2637	284	134
unwind except'n	3374	3467	284	196