The following paper was originally published in the
Proceedings of the Fifth Annual Tcl/Tk Workshop
Boston, Massachusetts, July 1997

TxRx: An ONC RPC Interpreter

Cristian S. Mata
Department of Computer Science, State University of New York
Stony Brook, NY

TxRx : An ONC RPC Interpreter

Cristian S. Mata*
Department of Computer Science
State University of New York
Stony Brook NY 11794-4400
cristian@cs.sunysb.edu

1 Introduction

TxRx is a run-time environment for ONC RPC tightly integrated with the Tcl language. Open Network Computing Remote Procedure Call [RFC1831] is a widely supported interprocess communication protocol currently on the standards track of the IETF. The eXternal Data Representation (XDR) is the language used by ONC RPC to describe inter-application communication. *TxRx* provides an interpreter for XDR and support for the features of ONC RPC.
TxRx improves the process of building distributed applications by eliminating most of the steps required to develop such a program. *TxRx* makes Tcl scripts compatible with existing client-server applications and facilitates access to system services like NIS and NFS.
RPC uses the *procedure call* abstraction to model the process of sending a request to, and receiving a reply from, a remote computer. The message sent to the remote is an encoding of the parameters of the procedure. The return value from the procedure call is the reply message received from the server. *TxRx* is intended for use in instances where compatibility with existing systems is important e.g. when existing client-server systems need to be upgraded.

2 Implementation

TxRx is a Tcl dynamically loadable package. It consists of a compiler/interpreter for XDR and code that implements RPC communication. A developer defines the communication protocol between two applications by specifying the remote procedures, parameters and return values in XDR. The next step – at run-time – is to load the protocol description using a *TxRx* command. The file with the protocol description is compiled by *TxRx* into bytecode.
In ONC RPC the internal data structures used by the protocol – the RPC headers – are defined using XDR. *TxRx* takes advantage of this feature by using a data driven approach: RPC headers are bytecode-compiled with the user protocol. When a message arrives from the remote computer, the incoming data stream is parsed according to the instructions stored in the bytecode. Conversely, outgoing data, including RPC headers, are converted from Tcl data into binary data by interpreting the bytecode program.
TxRx is layered, with different abstractions implemented at each level. The base level is the XDR interpreter. Its function is to create and manage objects that encapsulate the communication protocol. The level above it implements remote procedure call functionality and handles data buffer management, timeout and network transport semantics. The user level deals with authentication and security issues.
RPC processing in *TxRx* is independent of the communication channel. One advantage is that connection setup between client and server is done separately from data transfer.

3 Experimental results

The goal of *TxRx* is to reduce development time and improve application portability. The idea behind the experiments is that local processing times are small compared to network latency and throughput. The typical workload consists of RPC calls with variable payload size with little processing on both the server and the client. The time performance of *TxRx* is compared to the performance of a C program generated with *rpcgen*. All experiments were executed on Sun workstations running Solaris. The data in Figure 1 reflects the difference in speed between *TxRx* and C code. The times are the av-

*Supported by NSF grants CCR-9201585 and CCR-9501192 and by a grant from Hughes Aircraft.

erage times in seconds required to execute a given number of iterations. The client and server are on the same local network. The C code implementation is about 4 times faster than *TxRx* .

Figures 2 and 3 graph the ratio of times between *TxRx* and C clients with respect to the number of RPC calls respectively data transfer size between client and server. Somewhat surprisingly, there are no major differences in behavior when client and server are located on the same computer – label "Local" – or on the same Ethernet segment – label "Ether". The increase in the ratio of execution times can be attributed to the interpreted approach – Figure 2 – of *TxRx* . When client and server are hosts on the Internet – one host in *edu*, the other in *com* – performance becomes a function of overall data traffic on the Internet.

Figure 3 is the time ratio between *TxRx* and C code with respect to the size of the data transfer. In this case the key to understanding the graph is the structure of the transferred data – in this case a linked list. The performance ratio between *TxRx* and C-code decreases when the complexity of the data structure increases.

4 What next?

TxRx is currently implemented as a "C code + Tcl script"[TxRx] extension. This is for convenience and for efficiency reasons. As Tcl evolves – with the introduction of the bytecode compiler and native binary data handling in version 8 – it becomes possible to implement *TxRx* as a script only extension. Efficiency aside, this makes interesting applications possible, like a WebNFS [RFC2055] client coded entirely in Tcl.

References

[RFC2055] Brent Callaghan. SUN Microsystems, Inc. WebNFS Server Specification, October 1996. http://www.sun.com/webnfs/

[RFC1831] Raj Srinivasan. SUN Microsystems, Inc. RPC: Remote Procedure Call Protocol Specification, Version 2. Request for Comments 1831, Standards Track, August 1995. http://ds.internic.net/rfc/rfc1831.txt.

[TxRx] Cristian Mata. SUNY Stony Brook. *TxRx* : An ONC RPC extension for Tcl. http://www.cs.sunysb.edu/~cristian/txrx.html

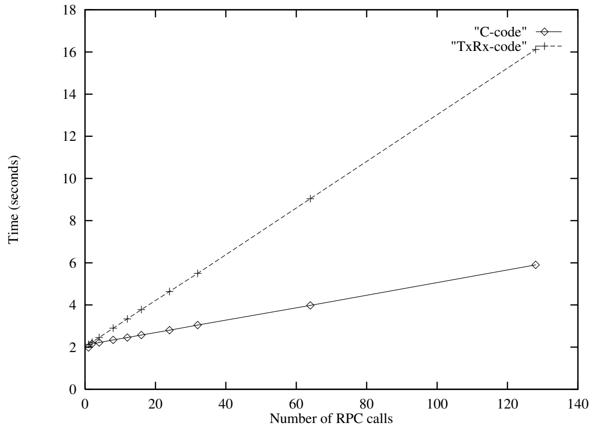

Figure 1: Execution time as a function of the number of calls

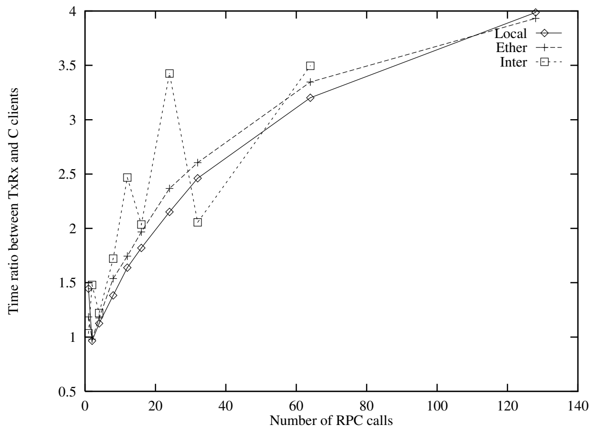

Figure 2: Performance ratio vs. number of calls

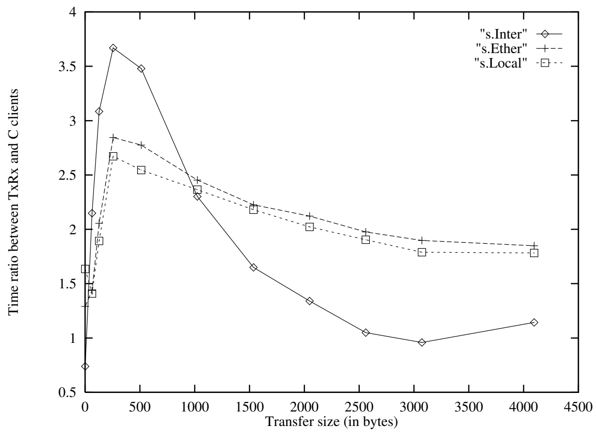

Figure 3: Performance ratio vs. transfer size