Benjamin Prosnitz
EECS443
01/19/09

Title: Implementing Remote Procedure Calls
Authors: Birrell, Nelson

Summary: An overview of their implementation of RPC, including methods used to improve performance.

Ideas: Remote Procedure Calls were implemented by inserting stubs in both the client and server which act as normal functions. These stubs perform network communication without knowledge of the program that is using them. They created a new transport-level protocol to improve performance and minimize latency. The protocol also included exception-handling capabilities which allow it to send an exception back to the client rather than a typical return value. For security, they sent the call values in an encrypted format.

Flaws: RPC can't really be treated as a normal procedure call because of the huge delay when waiting for data. These procedure calls would either be used for querying data or starting a remote process. So a normal local procedure wouldn't typically be just replaced with a remote procedure, and because of this the transparent programming call could be replaced with something more explicit.

Relevance/Future: Web services and distributed computing are increasingly used and remote procedure calls provide an easy mechanism to access data from these services. Some applications operating on paradigms like Model-View-Controller could use RPC to transparently move one of these components between a client and server. Newer networks may make the latency of RPCs less of a factor.